PRESIDENCY UNIVERSITY

Presidency University Act, 2013 of the Karnataka Act No. 41 of 2013 | Established under Section 2(f) of UGC Act, 1956
Approved by AICTE, New Delhi

PRESIDENCY SCHOOL OF ARTIFICIAL INTELLIGENCE AND ADVANCED COMPUTING

Program Regulations and Curriculum

2026-2030

BACHELOR OF TECHNOLOGY (B.Tech.) in

Computer Science and Engineering
(Artificial Intelligence and Machine Learning)
B. Tech. [CAI]
based on Choice Based Credit System (CBCS) and Outcome Based Education (OBE)

Table of Contents

Clause No.	Contents	Page Number
PART A – PROGRAM REGULATIONS		
1.	Vision & Mission of the University and the School / Department	
2.	Preamble to the Program Regulations and Curriculum	
3.	Short Title and Applicability	
4.	Definitions	
5.	Program Description	
6.	Minimum and Maximum Duration	
7.	Programme Educational Objectives (PEO)	
8.	Programme Outcomes (PO) and Programme Specific Outcomes (PSO)	
9.	Admission Criteria (as per the concerned Statutory Body)	
10.	Lateral Entry / Transfer Students requirements	
11.	Change of Branch / Discipline / Specialization	
12.	Specific Regulations regarding Assessment and Evaluation	
13.	Additional clarifications - Rules and Guidelines for Transfer of Credits from MOOC, etc.	
PART B: PROGRAM STRUCTURE		
14.	Structure / Component with Credit Requirements Course Baskets & Minimum Basket wise Credit Requirements	
15.	Minimum Total Credit Requirements of Award of Degree	
16.	Other Specific Requirements for Award of Degree, if any, as prescribed by the Statutory Bodies	
PART C: CURRICULUM STRUCTURE		
17.	Curriculum Structure – Basket Wise Course List	
18.	Practical / Skill based Courses – Internships / Thesis / Dissertation / Capstone Project Work / Portfolio / Mini project	
19.	List of Elective Courses under various Specializations / Stream Basket	

20.	List of Open Electives to be offered by the School / Department (Separately for ODD and EVEN Semesters).	
21.	List of MOOC Courses	
22.	Recommended Semester Wise Course Structure / Flow including the Program / Professional Elective Paths / Options	
23.	Course Catalogue of all Courses Listed including the Courses Offered by other School / Department and Discipline / Program Electives	

PART A – PROGRAM REGULATIONS

1. Vision & Mission of the University and the School / Department

1.1 Vision of the University

To be a Value-driven Global University, excelling beyond peers and creating professionals of integrity and character, having concern and care for society.

1.2 Mission of the University

- Commit to be an innovative and inclusive institution by seeking excellence in teaching, research and knowledge-transfer.
- Pursue Research and Development and its dissemination to the community, at large.
- Create, sustain and apply learning in an interdisciplinary environment with consideration for ethical, ecological and economic aspects of nation building.
- Provide knowledge-based technological support and services to the industry in its growth and development.
- To impart globally-applicable skill-sets to students through flexible course offerings and support industry's requirement and inculcate a spirit of new-venture creation.

1.3 Vision of Presidency School of Computer Science and Engineering

To be a value based, practice-driven School of Computer Science and Engineering, committed to developing globally-competent Engineers, dedicated to developing cutting-edge technology, towards enhancing Quality of Life.

1.4 Mission of Presidency School of Computer Science and Engineering

- Cultivate a practice-driven environment, with computing-based pedagogy, integrating theory and practice.
- Attract and nurture world-class faculty to excel in Teaching and Research, in the realm of Computing Sciences.
- Establish state-of-the-art computing facilities, for effective Teaching and Learning experiences.
- Promote Interdisciplinary Studies to nurture talent for global impact.
- Instill Entrepreneurial and Leadership Skills to address Social, Environmental and Community-needs.

2. Preamble to the Program Regulations and Curriculum

This is the subset of Academic Regulations and it is to be followed as a requirement for the award of B.Tech degree.

The Curriculum is designed to take into the factors listed in the Choice Based CreditSystem (CBCS) with focus on Social Project Based Learning, Industrial Training, and Internship toenable the students to become eligible and fully equipped for employment in industries,choosehigherstudiesorentrepreneurship.

In exercise of the powers conferred by and in discharge of duties assigned under the relevant provision(s) of the Act, Statutes and Academic Regulations, of the University, the Academic Council hereby makes the following Regulations.

3. Short Title and Applicability

a. These Regulations shall be called the Bachelor of Technology Degree Program Regulations and Curriculum 2026-2030.
b. These Regulations are subject to, and pursuant to the Academic Regulations 2026.
c. These Regulations shall be applicable to the ongoing Bachelor of Technology Degree Programs of the 2026-2030batch, and to all other Bachelor of Technology Degree Programs which may be introduced in future.
d. These Regulations shall supersede all the earlier Bachelor of Technology Degree Program Regulations and Curriculum, along with all the amendments thereto.
e. These Regulations shall come into force from the Academic Year 2026-2027.

4. Definitions

In these Regulations, unless the context otherwise requires:

a. "Academic Calendar" means the schedule of academic and miscellaneous events as approved by the Vice Chancellor;
b. "Academic Council" means the Academic Council of the University;
c. "Academic Regulations" means the Academic Regulations, of the University;
d. "Academic Term" means a Semester or Summer Term;
e. "Act" means the Presidency University Act, 2013;
f. "AICTE" means All India Council for Technical Education;
g. "Basket" means a group of courses bundled together based on the nature/type of the course;
h. "BOE" means the Board of Examinations of the University;
i. "BOG" means the Board of Governors of the University;
j. "BOM" means the Board of Management of the University;
k. "BOS" means the Board of Studies of a particular Department/Program of Study of the University;
l. "CGPA" means Cumulative Grade Point Average as defined in the Academic Regulations;
m. "Clause" means the duly numbered Clause, with Sub-Clauses included, if any, of these Regulations;
n. "COE" means the Controller of Examinations of the University;
o. "Course In Charge" means the teacher/faculty member responsible for developing and organising the delivery of the Course;
p. "Course Instructor" means the teacher/faculty member responsible for teaching and evaluation of a Course;
q. "Course" means a specific subject usually identified by its Course-code and Course-title, with specified credits and syllabus/course-description, a set of references, taught by some teacher(s)/course-instructor(s) to a specific class (group of students) during a specific Academic Term;
r. "Curriculum Structure" means the Curriculum governing a specific Degree Program offered by the University, and, includes the set of Baskets of Courses along with minimum credit requirements to be earned under each basket for a degree/degree with specialization/minor/honours in addition to the relevant details of the Courses and Course catalogues (which describes the Course content and other important information about the Course). Any specific requirements for a particular program may be

brought into the Curriculum structure of the specific program and relevant approvals should be taken from the BOS and Academic Council at that time.

s. *"DAC" means the Departmental Academic Committee of a concerned Department/Program of Study of the University;*

t. *"Dean" means the Dean / Director of the concerned School;*

u. *"Degree Program" includes all Degree Programs;*

v. *"Department" means the Department offering the degree Program(s) / Course(s) / School offering the concerned Degree Programs / other Administrative Offices;*

w. *"Discipline" means specialization or branch of B.Tech. Degree Program;*

x. *"HOD" means the Head of the concerned Department;*

y. *"L-T-P-C" means Lecture-Tutorial-Practical-Credit – refers to the teaching – learning periods and the credit associated;*

z. *"MOOC" means Massive Open Online Courses;*

aa. *"MOU" means the Memorandum of Understanding;*

bb. *"NPTEL" means National Program on Technology Enhanced Learning;*

cc. *"Parent Department" means the department that offers the Degree Program that a student undergoes;*

dd. *"Program Head" means the administrative head of a particular Degree Program/s;*

ee. *"Program Regulations" means the Bachelor of Technology Degree Program Regulations and Curriculum, 2022-2026;*

ff. *"Program" means the Bachelor of Technology (B.Tech.) Degree Program;*

gg. *"PSOE" means the Presidency School of Engineering;*

hh. *"Registrar" means the Registrar of the University;*

ii. *"School" means a constituent institution of the University established for monitoring, supervising and guiding, teaching, training and research activities in broadly related fields of studies;*

jj. *"Section" means the duly numbered Section, with Clauses included in that Section, of these Regulations;*

kk. *"SGPA" means the Semester Grade Point Average as defined in the Academic Regulations;*

ll. *"Statutes" means the Statutes of Presidency University;*

mm. *"Sub-Clause" means the duly numbered Sub-Clause of these Program Regulations;*

nn. *"Summer Term" means an additional Academic Term conducted during the summer break (typically in June-July) for a duration of about eight (08) calendar weeks, with a minimum of thirty (30) University teaching days;*

oo. *"SWAYAM" means Study Webs of Active Learning for Young Aspiring Minds.*

pp. *"UGC" means University Grant Commission;*

qq. *"University" means Presidency University, Bengaluru; and*

rr. *"Vice Chancellor" means the Vice Chancellor of the University.*

5. Program Description

The Bachelor of Technology Degree Program Regulations and Curriculum 2022-2026 are subject to, and, pursuant to the Academic Regulations, 2026. These Program Regulations shall be applicable to

the following ongoing Bachelor of Technology (B.Tech.) Degree Programs of 2022-2026 offered by the Presidency School of Computer Science and Engineering (PSCS):

The Bachelor of Technology Degree Program Regulations and Curriculum 2026-2030 are subject to, and, pursuant to the Academic Regulations. These Program Regulations shall be applicable to the following ongoing Bachelor of Technology (B.Tech.) Degree Programs of 2026-2030 offered by the Presidency School of Computer Science and Engineering (PSCS):

1. Bachelor of Technology in Computer Science and Engineering, abbreviated as CSE
2. Bachelor of Technology in Computer Science and Technology (Big Data), abbreviated as CBD
3. Bachelor of Technology in Computer Science and Engineering (Block Chain), abbreviated as CBC
4. Bachelor of Technology in Computer Science and Technology (Dev Ops), abbreviated as CDV
5. Bachelor of Technology in Computer Science and Engineering (Cyber Security), abbreviated as CCS
6. Bachelor of Technology in Computer Science and Engineering (Internet of Things), abbreviated as CIT
7. Bachelor of Technology in Computer Science and Engineering (Data Science), abbreviated as CSD
8. Bachelor of Technology in Computer Science and Technology, abbreviated as CSG
9. Bachelor of Technology in Information Science and Technology, abbreviated as IST
10. Bachelor of Technology in Computer Science and Information Technology, abbreviated as CSI
11. Bachelor of Technology in Computer Science and Engineering (Networks), abbreviated as CSN
12. Bachelor of Technology in Computer Engineering, abbreviated as COM
13. Bachelor of Technology in Information Science and Engineering, abbreviated as ISE and
14. Bachelor of Technology in Computer Science and Engineering (Artificial Intelligence and Machine Learning) abbreviated as CAI
15. Bachelor of Technology in Robotics and Artificial Intelligence, abbreviated as RAI
16. Bachelor of Technology in Artificial Intelligence and Data Science, abbreviated as AID

5.1 These Program Regulations shall be applicable to other similar programs, which may be introduced in future.

5.2 These Regulations may evolve and get amended or modified or changed through appropriate approvals from the Academic Council, from time to time, and shall be binding on all concerned.

5.3 The effect of periodic amendments or changes in the Program Regulations, on the students admitted in earlier years, shall be dealt with appropriately and carefully, so as to ensure that those students are not subjected to any unfair situation whatsoever, although they are required to conform to these revised Program Regulations, without any undue favour or considerations

6. Minimum and Maximum Duration

6.1 Bachelor of Technology Degree Program is a Four-Year, Full-Time Semester based program. The minimum duration of the B.Tech. Program is four (04) years and each year comprises of two academic Semesters (Odd and Even Semesters) and hence the duration of the B.Tech. program is eight (08) Semesters.

6.2 A student who for whatever reason is not able to complete the Program within the normal period or the minimum duration (number of years) prescribed for the Program, may be allowed a period

of two years beyond the normal period to complete the mandatory minimum credits requirement as prescribed by the concerned Program Regulations and Curriculum. In general, the permissible maximum duration (number of years) for completion of Program is 'N' + 2 years, where 'N' stands for the normal or minimum duration (number of years) for completion of the concerned Program as prescribed by the concerned Program Regulations and Curriculum.

6.3 The time taken by the student to improve Grades/CGPA, and in case of temporary withdrawal/re-joining (Refer to Clause **Error! Reference source not found.** of Academic Regulations), shall be counted in the permissible maximum duration for completion of a Program.

6.4 In exceptional circumstances, such as temporary withdrawal for medical exigencies where there is a prolonged hospitalization and/or treatment, as certified through hospital/medical records, women students requiring extended maternity break (certified by registered medical practitioner), and, outstanding sportspersons representing the University/State/India requiring extended time to participate in National/International sports events, a further extension of one (01) year may be granted on the approval of the Academic Council.

6.5 The enrolment of the student who fails to complete the mandatory requirements for the award of the concerned Degree (refer Section 19.**Error! Reference source not found.** of Academic Regulations) in the prescribed maximum duration (Clauses 18.1 and 18.2 of Academic Regulations), shall stand terminated and no Degree shall be awarded.

7 Programme Educational Objectives (PEO)

After four years of successful completion of the program, the graduates shall be:

PEO 1. Demonstrate as a Computer Engineering Professional

PEO 2. Engage in lifelong learning through research and professional development

PEO 3. Serve as a leader in the profession through consultancy, extension activities and/ or entrepreneurship

8 Programme Outcomes (PO) and Programme Specific Outcomes (PSO)

8.1 Programme Outcomes (PO)

On successful completion of the Program, the students shall be able to:

PO1: Engineering Knowledge: Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization to develop to the solution of complex engineering problems.

PO2: Problem Analysis: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development.

PO3: Design/Development of Solutions: Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required.

PO4: Conduct Investigations of Complex Problems: Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions.

PO5: Engineering Tool Usage: Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems.

PO6: The Engineer and The World: Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment.

PO7: Ethics: Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws.

PO8: Individual and Collaborative Team work: Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.

PO9: Communication: Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences

PO10: Project Management and Finance: Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.

PO11: Life-Long Learning: Recognize the need for,and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies, and iii) critical thinking in the broadest context of technological change.

8.2 Program Specific Outcomes(PSOs):

On successful completion of the Program, the students shall be able to:

PSO1: Exhibit comprehensive knowledge of computer science to interpret foundational concepts, evaluate computational challenges, and utilize theoretical insights to design and implement dependable and efficient computing solutions.
PSO2: Apply programming expertise, ethical and responsible AI/ML knowledge to create practical solutions for societal benefit, enabling career readiness in software development, research, higher education, or entrepreneurship.
PSO3: Ability to understand, recognize opportunities where CSE, AI-ML concepts can be applied, and to identify the right, responsible for AI-ML techniques in such contexts.

9 Admission Criteria (as per the concerned Statutory Body)

The University admissions shall be open to all persons irrespective of caste, class, creed, gender or nation. All admissions shall be made on the basis of merit in the qualifying examinations; provided that forty percent of the admissions in all Programs of the University shall be reserved for the students of Karnataka

State and admissions shall be made through a Common Entrance Examination conducted by the State Government or its agency and seats shall be allotted as per the merit and reservation policy of the State Government from time to time**.** The admission criteria to the B.Tech. Program is listed in the following Sub-Clauses:

9.1 An applicant who has successfully completed Pre-University course or Senior Secondary School course (+2) or equivalent such as (11+1), 'A' level in Senior School Leaving Certificate Course from a recognized university of India or outside or from Senior Secondary Board or equivalent, constituted or recognized by the Union or by the State Government of that Country for the purpose of issue of qualifying certificate on successful completion of the course, may apply for and be admitted into the Program.

9.2 Provided further, the applicant must have taken Physics and Mathematics as compulsory subjects in the Pre-University / Higher Secondary / (10+2) / (11+1) examination, along with either Chemistry / Biology / Electronics / Computer Science / Biotechnology subject, and, the applicant must have obtained a minimum of 45% of the total marks (40% in case of candidates belonging to the Reserved Category as classified by the Government of Karnataka) in these subjects taken together.

9.3 The applicant must have appeared for Joint Entrance Examinations (JEE) Main / JEE (Advanced) / Karnataka CET / COMED-K, or any other State-level Engineering Entrance Examinations.

9.4 Reservation for the SC / ST and other backward classes shall be made in accordance with the directives issued by the Government of Karnataka from time to time.

9.5 Admissions are offered to Foreign Nationals and Indians living abroad in accordance with the rules applicable for such admission, issued from time to time, by the Government of India.

9.6 Candidates must fulfil the medical standards required for admission as prescribed by the University.

9.7 If, at any time after admission, it is found that a candidate had not in fact fulfilled all the requirements stipulated in the offer of admission, in any form whatsoever, including possible misinformation and any other falsification, the Registrar shall report the matter to the Board of Management (BOM), recommending revoking the admission of the candidate.

9.8 The decision of the BOM regarding the admissions is final and binding.

10 Lateral Entry / Transfer Students requirements

10.1 Lateral Entry

The University admits students directly to the second year (3rd Semester) of the B.Tech. Degree program as per the provisions and/or regulations of the Government of Karnataka pertaining to the "Lateral Entry" scheme announced by the Government from time to time.Further, the general conditions and rules governing the provision of Lateral Entry to the B.Tech. Program of the University are listed in the following Sub-Clauses:

10.1.1 Admission to 2nd year (3rd Semester) of the B.Tech. Degree program shall be open to the candidates who are holders of a 3-year Diploma in Engineering (or equivalent qualification as recognized by the University), who have secured not less than forty-five percentage (45%) marks in the final year examination (5th and 6th Semesters of the Diploma Program)

in the appropriate branch of Engineering. Provided that, in case of SC / ST and OBC candidates from Karnataka the minimum marks for eligibility shall be forty percent (40%).

10.1.2 Provided further that, candidates seeking Lateral Entry may be required to complete specified bridge Courses as prescribed by the University. Such bridge Courses, if any, shall not be included in the CGPA computations.

10.1.3 All the existing Regulations and Policies of the University shall be binding on all the students admitted to the Program through the provision of Lateral Entry.

10.1.4 The Course requirements prescribed for the 1st Year of the B.Tech. Program shall be waived for the student(s) admitted through Lateral Entry and the duration of the B.Tech. Program for such students is three (03) years, commencing from the 3rd Semester (commencement of the 2nd Year) of the B.Tech. Program and culminating with the 8th Semester (end of the 4th Year) of the B.Tech. Program.

10.1.5 Provided that, if a Lateral Entry student misses any mandatory program specific courses that are typically offered in the 1st year (1st or 2nd semesters), then those courses must be cleared by the students as soon as possible, preferably during the Summer Term.

10.1.6 The existing Program Regulations of the concerned Program to which the student is admitted through the provision of Lateral Entry shall be binding on the student with effect from the 3rd Semester of the Program. i.e., the Program Structure and Curriculum from the 3rd to 8th Semesters of the Program concerned shall be binding on the student admitted through Lateral Entry. Further, any revisions / amendments made to the Program Regulations, thereafter, shall be binding on all the students of the concerned Program.

10.1.7 All the Courses (and the corresponding number of Credits) prescribed for the 1st Year of the concerned B.Tech. Program shall be waived for the student(s) admitted to the concerned B.Tech Program through Lateral Entry. Further, the *Minimum Credit Requirements* for the award of the B.Tech. Degree in the concerned Program shall be prescribed / calculated as follows:

The ***Minimum Credit Requirements*** for the award of the Bachelor of Technology (B.Tech.) Degree prescribed by the concerned Bachelor of Technology Degree Program Regulations and Curriculum, 2024-2028, minus the number of Credits prescribed / accepted by the Equivalence Committee for the 1st Year (1st and 2nd Semesters) of the B.Tech. Program.

For instance, if the *Minimum Credit Requirements* for the award of the Bachelor of Technology (B.Tech.) Degree as prescribed by the Regulations for B.Tech. Computer Science and Engineering (Artificial Intelligence and Machine Learning) is "N" Credits, and, if the total credits prescribed in the 1st Year (total credits of the 1st and 2nd Semesters) of the Program concerned is "M" Credits, then the *Minimum Credit Requirements* for the award of the B.Tech. in Computer Science and Engineering (Artificial Intelligence and Machine Learning)for a student who joins the Program through the provision of the Lateral Entry, shall be "N – M" Credits.

10.1.8 Further, no other waiver except the Courses prescribed for the 1st year of the B.Tech. Program of the University shall be permissible for students joining the B.Tech. Program through the provision of Lateral Entry.

10.2 Transfer of student(s) from another recognized University to the 2nd year (3rd Semester) of the B.Tech. Program of the University

A student who has completed the 1st Year (i.e., passed in all the Courses / Subjects prescribed for the 1st Year) of the B.Tech. / B.E. / B.S., Four-Year Degree Program from another recognized University, may be permitted to transfer to the 2nd Year (3rd Semester) of the B.Tech. Program of the University as per the rules and guidelines prescribed in the following Sub-Clauses:

10.1.1 The concerned student fulfils the criteria specified in Sub-Clauses 10.1.1, 10.1.2 and 10.1.3

10.1.2 The student shall submit the Application for Transfer along with a non-refundable Application Fee (as prescribed by the University from time to time) to the University no later than July 10 of the concerned year for admission to the 2nd Year (3rd Semester) B.Tech. Program commencing on August 1 on the year concerned.

10.1.3 The student shall submit copies of the respective Marks Cards / Grade Sheets / Certificates along with the Application for Transfer.

10.1.4 The transfer may be provided on the condition that the Courses and Credits completed by the concerned student in the 1st Year of the B.Tech. / B.E. / B.S. Four-Degree Program from the concerned University, are declared equivalent and acceptable by the Equivalence Committee constituted by the Vice Chancellor for this purpose. Further, the Equivalence Committee may also prescribe the Courses and Credits the concerned students shall have to mandatorily complete, if admitted to the 2nd Year of the B.Tech. Program of the University.

10.1.5 The Branch / Discipline allotted to the student concerned shall be the decision of the University and binding on the student.

11 Change of Branch / Discipline / Specialization

A student admitted to a particular Branch of the B.Tech. Program will normally continue studying in that Branch till the completion of the program. However, the University reserves the right to provide the option for a change of Branch, or not to provide the option for a change of Branch, at the end of 1st Year of the B.Tech. Program to eligible students in accordance with the following rules and guidelines: framed by the University from time to time.

11.1. Normally, only those students, who have passed all the Courses prescribed for the 1st Year of the B.Tech. Program and obtained a CGPA of not less than 6.50 at the end of the 2nd Semester, shall be eligible for consideration for a change of Branch.

11.2. Change of Branch, if provided, shall be made effective from the commencement of the 3rd Semester of the B.Tech. Program. There shall be no provision for change of Branch thereafter under any circumstances whatsoever.

11.3. The student provided with the change of Branch shall fully adhere to and comply with the Program Regulations of the concerned Branch of the B.Tech. Program, the Fee Policy pertaining to that Branch of the B.Tech. Program, and, all other rules pertaining to the changed Branch existing at the time.

11.4. Change of Branch once made shall be final and binding on the student. No student shall be permitted, under any circumstances, to refuse the change of Branch offered.

11.5. The eligible student may be allowed a change in Branch, strictly in order of *inter se* merit, subject to the conditions given below:

11.5.1. The actual number of students in the 3rd Semester in any particular Branch to which the transfer is to be made, should not exceed the intake fixed by the University for the concerned Branch;

11.5.2. The actual number of students in any Branch from which transfer is being sought does not fall below 75% of the total intake fixed by the University for the concerned Branch.

The process of change of Branch shall be completed within the first five days of Registration for the 3rd Semester of the B.Tech. Program.

12. Specific Regulations regarding Assessment and Evaluation(including the Assessment Details of NTCC Courses, Weightages of Continuous Assessment and End Term Examination for various Course Categories)

12.1. The academic performance evaluation of a student in a Course shall be according to the University Letter Grading System based on the class performance distribution in the Course.

12.2. Academic performance evaluation of every registered student in every Course registered by the student is carried out through various components of Assessments spread across the Semester. The nature of components of Continuous Assessments and the weightage given to each component of Continuous Assessments (refer Clause 12.5 of academic regulations)shall be clearly defined in the Course Plan for every Courseand approved by the DAC.

12.3. Format of the End-Term examination shall be specified in the Course Plan.

12.4. Grading is the process of rewarding the students for their overall performance in each Course. The University follows the system of Relative Grading with statistical approach to classify the students based on the relative performance of the students registered in the concerned Course except in the following cases:

- Non-Teaching Credit Courses (NTCC)
- Courses with a class strength less than 30

Absolute grading method may be adopted, where necessary with prior approval of concerned DAC.

Grading shall be done at the end of the Academic Term by considering the aggregate performance of the student in all components of Assessments prescribed for the Course. Letter Grades (Clause **Error! Reference source not found.** of academic regulations) shall be awarded to a student based on her/his overall performance relative to the class performance distribution in the concerned Course. These Letter Grades not only indicate a qualitative assessment of the student's performance but also carry a quantitative (numeric) equivalent called the Grade Point.

12.5. Assessment Components and Weightage

S. No	Credit Structure [L-T-P-C]	Percentage/ Marks	CA		Mid-Term		End-term		Project	Total	Exam Conducted by
			Theory	Practical	Theory	Practical	Theory	Practical			
1	3-0-0-3	Percentage	25%	-	25%	-	50%	-	-	100 %	Mid-Term & End Term by CoE
		Marks	50	-	50	-	100	-	-	200	

2	2-0-2-3	Percentage	12.50 %	12.50 %	12.50%	12.50 %	25%	25%	-	100 %	Mid-Term & End Term by CoE * Except for full stack courses
		Marks	25	25	25	25	50	50	-	200	
3	1-0-4-3	Percentage	-	25%	10%	40%	5%	20%	-	100 %	Mid-Term & End Term by School
		Marks	-	25	10	40	5	20	-	100	
4	2-0-4-4	Percentage	12.50 %	12.50 %	10%	15%	20%	30%	-	100 %	*Mid-Term & End Term by CoE
		Marks	25	25	20	30	40	60	-	200	
5	0-0-4-2	Percentage	-	50%	-	-	-	-	50%	100 %	Project evaluated by IC at School level
		Marks	-	50	-	-	-	-	50	100	
6	0-0-2-1	Percentage	-	100%	-	-	-	-	-	100 %	Only CA at School Level
		Marks	-	100	-	-	-	-	-	100	
7	3-0-2-4	Percentage	12.50 %	12.50 %	15%	10%	30%	20%	-	100 %	Mid-Term & End Term by CoE
		Marks	25	25	30	20	60	40	-	200	
8	2-0-0-2	Percentage	25 %	-	25%	-	50%	-	-	100%	Mid-Term & End Term by CoE
		Marks	50	-	50	-	100	-	-	200	

*CSE3150-Front End Full stack development
CSE3151-Java Full Stack Development
CSE3152-.Net Full Stack development

The exact weightages of Evaluation Components shall be clearly specified in the concerned PRC and respective Course Plan.

Normally, for Practice/Skill based Courses, without a defined credit structure (L–T–P) [NTCC], but with assigned Credits (as defined in Clause **Error! Reference source not found.** of the Academic Regulations), the method of evaluation shall be based only on Continuous Assessments. The various components of Continuous Assessments, the distribution of weightage among such components, and the method of evaluation/assessment, shall be as decided and indicated in the Course Plan/PRC. The same shall be approved by the respective DAC.

12.6. Minimum Performance Criteria:

12.6.1. Theory only Course and Lab/Practice Embedded Theory Course

A student shall satisfy the following minimum performance criteria to be eligible to earn the credits towards the concerned Course:

a. A student must obtain a minimum of 30% of the total marks/weightage assigned to the End Term Examinations in the concerned Course.

b. The student must obtain a minimum of 40% of the AGGREGATE of the marks/weightage of the components of Continuous Assessments, Mid Term Examinations and End Term Examinations in the concerned Course.

12.6.2. Lab/Practice only Course and Project Based Courses

The student must obtain a minimum of 40% of the AGGREGATE of the marks/weightage of all assessment components in the concerned Course.

12.6.3. A student who fails to meet the minimum performance criteria listed above in a Course shall be declared as "Fail" and given "F" Grade in the concerned Course. For theory Courses, the student shall have to re-appear in the "Make-Up Examinations" as scheduled by the University in any subsequent semester, or, re-appear in the End Term Examinations of the same Course when it is scheduled at the end of the following Semester or Summer Term, if offered. The marks obtained in the Continuous Assessments (other than the End Term Examination) shall be carried forward and be included in computing the final grade, if the student secures the minimum requirements (as per Sub-Clause 12.6.1 and12.6.2) in the "Make-Up Examinations" of the concerned Course. Further, the student has an option to re-register for the Course and clear the same in the summer term/ subsequent semester if he/she wishes to do so, provided the Course is offered.

13. Additional clarifications - Rules and Guidelines for Transfer of Credits from MOOC, etc. – Note: These are covered in Academic Regulations

The University allows students to acquire credits from other Indian or foreign institutions and/or Massive Open Online Course (MOOC) platforms, subject to prior approval. These credits may be transferred and counted toward fulfilling the minimum credit requirements for the award of a degree. The process of transfer of credits is governed by the following rules and guidelines:

13.1. The transfer of credits shall be examined and recommended by the Equivalence Committee (Refer **Error! Reference source not found.** of Academic Regulation) and approved by the Dean - Academics.

13.2. Students may earn credits from other Indian or foreign Universities/Institutions with which the University has an MOU, and that MOU shall have specific provisions, rules and guidelines for transfer of credits. These transferred credits shall be counted towards the minimum credit requirements for the award of the degree.

13.3. Students may earn credits by registering for Online Courses offered by *Study Web of Active Learning by Young and Aspiring Minds* (SWAYAM) and *National Program on Technology Enhanced Learning* (NPTEL), or other such recognized Bodies/ Universities/Institutions as approved by the concerned BOS and Academic Council from time to time. The concerned School/Parent Department shall publish/include the approved list of Courses and the rules and guidelines governing such transfer of credits of the concerned Program from time to time. The Rules and Guidelines for the transfer of credits specifically from the Online Courses conducted by SWAYAM/ NPTEL/ other approved MOOCs are as stated in the following Sub-Clauses:

13.3.1. A student may complete SWAYAM/NPTEL/other approved MOOCs as mentioned in Clause13.3(as per Academic Regulations)and transfer equivalent credits to partially or fully complete the mandatory credit requirements of Professional Elective Courses and/or the mandatory credit requirements of Open Elective Courses as prescribed in the concerned Curriculum Structure. However, it is the sole responsibility of the student to complete the mandatory credit requirements of the Professional Elective Courses and the Open Elective Courses as prescribed by the Curriculum Structure of the concerned Program.

13.3.2. SWAYAM/NPTEL/ other approved MOOCs as mentioned in Clause 13.3 (as per Academic Regulations) shall be approved by the concerned Board of Studies and placed (as Annexures) in the concerned PRC.

13.3.3. Parent Departments may release a list of SWAYAM/NPTEL/other approved MOOCs for Pre-Registration as per schedule in the Academic Calendar or through University Notification to this effect.

13.3.4. Students may Pre-Register for the SWAYAM/NPTEL/other approved MOOCs in the respective Departments and register for the same Courses as per the schedule announced by respective Online Course Offering body/institute/ university.

13.3.5. A student shall request for transfer of credits only from such approved Courses as mentioned in Sub-Clause 13.3.2 above.

13.3.6. SWAYAM/NPTEL/other approved MOOCs Courses are considered for transfer of credits only if the concerned student has successfully completed the SWAYAM/NPTEL/other approved MOOCs and obtained a certificate of successful/satisfactory completion.

13.3.7. A student who has successfully completed the approved SWAYAM/NPTEL/ other approved MOOCs and wants to avail the provision of transfer of equivalent credits, must submit the original Certificate of Completion, or such similar authorized

documents to the HOD concerned, with a written request for the transfer of the equivalent credits. On verification of the Certificates/Documents and approval by the HOD concerned, the Course(s) and equivalent Credits shall forwarded to the COE for processing of results of the concerned Academic Term.

13.3.8. The credit equivalence of the SWAYAM/NPTEL/other approved MOOCs are based on Course durations and/or as recommended by the Course offering body/institute/university. The Credit Equivalence mapped to SWAYAM/ NPTEL approved Courses based on Course durations for transfer of credits is summarised in Table shown below. The Grade will be calculated from the marks received by the Absolute Grading Table **Error! Reference source not found.** in the Academic Regulations.

Table 2: Durations and Credit Equivalence for Transfer of Credits from SWAYAM-NPTEL/ other approved MOOC Courses

Sl. No.	Course Duration	Credit Equivalence
1	4 Weeks	1 Credit
2	8 Weeks	2 Credits
3	12 Weeks	3 Credits

13.3.9. The maximum permissible number of credits that a student may request for credit transfer from MOOCs shall not exceed 20% of the mandatory minimum credit requirements specified by the concerned Program Regulations and Curriculum forthe award ofthe concernedDegree.

13.3.10. The University shall not reimburse any fees/expense; a student may incur for the SWAYAM/NPTEL/other approved MOOCs.

13.4. The maximum number of credits that can be transferred by a student shall be limited to forty percent (40%) of the mandatory minimum credit requirements specified by the concerned Program Regulations and Curriculum for the award of the concerned Degree. However, the grades obtained in the Courses transferred from other Institutions/MOOCs, as mentioned in this Section (13.0), shall not be included in the calculation of the CGPA.

13.5. Mandatory Non-Credit Course Completion Requirements: All mandatory non-credit courses shall be satisfactorily completed by the student as part of the degree requirements. These courses will be evaluated and awarded letter grades based on the following criteria:

S (Satisfactorily Completed): Awarded when the student successfully completes all prescribed course requirements.

NC (Not Completed): Awarded when the student fails to meet the prescribed course requirements.

A student receiving an NC grade must reappear for and complete the course in accordance with the guidelines prescribed by the University.

In the case of non-taught and non-credited mandatory courses—where students are advised to undertake learning through MOOC platforms—there shall be a clearly defined Course Catalogue and a corresponding Course Plan. The Course Plan shall outline the assessment components, which will form the basis for evaluation.

14. Structure / Component with Credit Requirements Course Baskets & Minimum Basket wise Credit Requirements

The B.Tech. (Computer Science and Engineering (Artificial Intelligence and Machine Learning) Program Structure (2022-2026) totalling 160 credits. Table 7 summarizes the type of baskets, number of courses under each basket and the associated credits that are mandatorily required for the completion of the Degree.

Table 3: B.Tech. (Artificial Intelligence and Machine Learning**) 2026-2030: Summary of Mandatory Courses and Minimum Credit Contribution from various Baskets**

S.No.	Baskets	Credit Contribution
1	Humanities and Social Sciences including Management Courses (HSMC)	12
2	Basic Science Courses (BSC)	24
3	Engineering Science Courses (ESC)	20
4	Professional Core Courses (PCC)	64
5	Professional Elective Courses (PEC)	18
6	Open Elective Courses (OEC)	6
7	Project Work (PRW)	16
8	Mandatory Courses (MAC)	0
	Total Credits	**160 (Minimum)**

In the entire Program, the practical and skill based course component contribute to an extent of approximately 57% out of the total credits of 160 for B.Tech. Computer Science and Engineering (Artificial Intelligence and Machine Learning)program of four years duration.

15. Minimum Total Credit Requirements of Award of Degree

As per the AICTE guidelines, a minimum of 160 credits is required for the award of a B.Tech. degree.

16. Other Specific Requirements for Award of Degree, if any, as prescribed by the Statutory Bodies,

16.1 The award of the Degree shall be recommended by the Board of Examinations and approved by the Academic Council and Board of Management of the University.

16.2 A student shall be declared to be eligible for the award of the concerned Degree if she/he:

a. Fulfilled the Minimum Credit Requirements and the Minimum Credits requirements under various baskets;

b. Secure a minimum CGPA of 4.50 in the concerned Program at the end of the Semester/Academic Term in which she/he completes all the requirements for the award of the Degree as specified in Sub-Clause a of Academic Regulations**;**

c. No dues to the University, Departments, Hostels, Library, and any other such Centers/ Departments of the University; and

d. No disciplinary action is pending against her/him.

PART C: CURRICULUM STRUCTURE

17, Curriculum Structure – Basket Wise Course List (not Semester Wise)

List of Courses Tabled – aligned to the Program Structure

(Course Code, Course Name, Credit Structure (LTPC), Contact Hours, Course Basket, Type of Skills etc., as applicable).

Type of Skill
F - Foundation
S - Skill Development
EM – Employability
EN – Entrepreneurship

Course Caters to
GS - Gender Sensitization
ES - Environment and sustainability
HP - Human values and Professional Ethics

Baskets
HSMC - Humanities and Social Sciences (*including Management courses*) (HSMC)
BSC - Basic Science Courses
ESC - Engineering Science Courses
PCC - Program Core Course
PEC - Professional Elective Courses
OEC - Open Elective Courses
EEC - Employment Enhancement Courses
MAC - Mandatory Course

Table 3.1: List of Humanities and Social Sciences including Management Courses (HSMC)								
Sl. No.	**Course Code**	**Course Name**	**L**	**T**	**P**	**Credits**	**Contact Hours**	**Type of Skill**
1	ENG1030	Fundamentals of Communication	1	0	2	2	3	S
2	ENG2030	Communication in Practice	1	0	2	2	3	S
3	FIN1002	Essentials of Finance	3	0	0	3	3	S
4	APT4005	Aptitude for Employability	0	0	2	1	2	AT
5	PPS3018	Preparedness for Interview	0	0	2	1	2	SS
6	ECE1512	Design Thinking in Practice	2	0	2	3	4	S/EM
Total			7	0	10	12	17	

Table 3.2: List of Basic Science Courses (BSC)								
Sl. No.	**Course Code**	**Course Name**	**L**	**T**	**P**	**Credits**	**Contact Hours**	**Type of Skill**
1	MAT1211	Linear Algebra, Differential Equations and Statistical Methods	3	1	0	4	4	F
2	PHY2501	Optoelectronics and Quantum Physics	3	0	0	3	3	F
3	PHY2504	Optoelectronics and Quantum Physics Lab	0	0	2	1	2	F
4	MAT1212	Mathematical foundations for Machine Learning and Data Analysis	3	1	0	4	4	F
5	CHE2501	Chemistry of Smart Materials	3	0	0	3	3	S
6	CHE2502	Chemistry of Smart Materials Lab	0	0	2	1	2	S
7	MAT2311	Vector Calculus, Numerical Methods and Sampling Distributions	3	1	0	4	4	EM
8	MAT2321	Discrete Mathematics	3	1	0	4	4	F
Total			19	3	4	24	26	

Table 3.3 : List of Engineering Science Courses (ESC)								
Sl. No.	**Course Code**	**Course Name**	**L**	**T**	**P**	**Credits**	**Contact Hours**	**Type of Skill**
1	MEC1006	Engineering Graphics	2	0	0	2	2	S
2	CSS1500	Computational Thinking using Python	2	0	2	3	4	S
3	ECE2022	Digital Design	2	0	0	2	2	F/S
4	ECE2052	Digital Design Lab	0	0	2	1	2	F/S
5	CIV1200	Foundations of Integrated Engineering	2	0	0	2	2	S
6	EEE1200	Basics of Electrical and Electronics Engineering	3	0	0	3	3	F/S
7	EEE1251	Basics of Electrical and Electronics Engineering Laboratory	0	0	2	1	2	F/S
8	CSS2264	Essentials of AI	3	0	0	3	3	S/EM
9	CSS2265	Essentials of AI Lab	0	0	2	1	2	S/EM
10	CSS2274	Competitive Programming and Problem Solving	0	0	4	2	4	S/EM
Total			14	0	12	20	26	

Table 3.4 : List of Program Core Courses								
Sl. No.	Course Code	Course Name	L	T	P	C	Contact Hours	Type of Skills
1	CSS2200	Problem Solving using C	2	0	0	2	2	S
2	CSS2201	Problem Solving using C Lab	0	0	4	2	4	S
3	CSS2251	Data Communication and Computer Networks	3	0	0	3	3	S
4	CSS2252	Data Communication and Computer Networks Lab	0	0	2	1	2	S
5	CSS2253	Data Structures	3	0	0	3	3	S
6	CSS2254	Data Structures Lab	0	0	2	1	2	S
7	CSS2255	Object Oriented Programming Using Java	3	0	0	3	3	S
8	CSS2256	Object Oriented Programming Using Java Lab	0	0	2	1	2	S
9	CSS2257	Computer Organization and Architecture	3	0	0	3	3	S
10	CSS2260	Database Management Systems	3	0	0	3	3	S
11	CSS2261	Database Management Systems Lab	0	0	2	1	2	S

12	CAI2500	Machine Learning	3	0	0	3	3	S
13	CAI2501	Machine Learning Lab	0	0	4	2	4	S
14	CAI2511	Ethics of AI	2	0	0	2	2	S
15	CSS2262	Analysis of Algorithms	3	1	0	4	4	S
16	CSS2263	Analysis of Algorithms Lab	0	0	2	1	2	S
17	CSS2266	Theory of Computation	3	0	0	3	3	S
18	CAI2502	Deep Learning	3	0	0	3	3	S
19	CAI2503	Deep Learning Lab	0	0	4	2	4	S
20	CAI2512	Neural Networks and Fuzzy Logic	3	0	0	3	3	S
21	CAI2509	Web Technologies using AI	3	0	0	3	3	S
22	CAI2510	Web Technologies using AI Lab	0	0	2	1	2	S
23	CSS2269	Operating Systems	3	0	0	3	3	S
24	CSS2270	Operating Systems Lab	0	0	2	1	2	S
25	CAI2507	Reinforcement Learning	2	0	0	2	2	S
26	CAI2508	Reinforcement Learning Lab	0	0	2	1	2	S
27	CAI2504	Natural Language Processing	3	0	0	3	3	S
28	CAI2505	Natural Language Processing Lab	0	0	2	1	2	S
29	CSS2271	Software Design and Development	3	0	0	3	3	S
	total		48	1	30	64	79	

Table 3.5: List of course in Project Work basket (PRW)

S. No	Course Code	Course Name	L	T	P	C	Contact Hours	Type of Skills
1	CSS7000	Internship	0	0	0	2	0	S/EM
2	CSS7100	Mini Project	0	0	0	4	0	S/EM
3	CSS7300	Capstone Project	0	0	0	10	0	S/EM
Total No. of Credits						**16**		

Table 3.6 : List of Mandatory Courses (MAC)

S.No	Course Code	Course Name	L	T	P	C
1	CHE7601	Environmental Studies	1	0	0	0
2	LAW7601	Indian Constitution	1	0	0	0
3	CIV7601	Universal Human Values and Ethics	1	0	0	0
4	PPS1025	Industry Readiness Program - I	0	0	2	0
5	PPS1026	Industry Readiness Program - II	0	0	2	0
6	APT4002	Introduction to Aptitude	0	0	2	0
7	APT4004	Aptitude Training - Intermediate	0	0	2	0
8	APT4006	Logical and Critical Thinking	0	0	2	0
	Total				0	

18. Practical / Skill based Courses – Internships / Thesis / Dissertation / Capstone Project Work / Portfolio / Mini project

Practical / Skill based Courses like internship, project work, capstone project, research project / dissertation, and such similar courses, where the pedagogy does not lend itself to a typical L-T-P-C Structure as defined in Clause 5.1 of the Academic Regulations, 2021, are simply assigned the number of Credits based on the quantum of work / effort required to fulfill the learning objectives and outcomes prescribed for the concerned Courses. Such courses are referred to as Non-Teaching Credit Courses (NTCC). These Courses are designed to provide students with hands-on experience and skills essential for their professional development. These courses aim to equip students with abilities in problem identification, root cause analysis, problem-solving, innovation, and design thinking through industry exposure and project-based learning. The expected outcomes are first level proficiency in problem solving and design thinking skills to better equip B.Tech. graduates for their professional careers. The method of evaluation and grading for the Practical / Skill based Courses shall be prescribed and approved by the concerned Departmental Academic Committee (refer Annexure A of the Academic Regulations, 2021). The same shall be prescribed in the Course Handout.

18.1 Internship / In-plant Training / Skill-based Program / International Immersion (IM) / Recognition of Prior Learning (RPL)

A student may undergo an Internship / In-plant Training / Skill-based Program / IM / RPL for a period minimum 04 weeks in an industry / company, government bodies, academic / research institution or recognized online platforms offering certified skill-based programs during the semester break between 4th and 5th semesters, subject to the following conditions:

18.1.1 The Internship / In-plant Training / Skill-based Program / IM / RPL shall be conducted in accordance with the Internship Policy prescribed by the University from time to time.

18.1.2 Internship: student shall undergo internship, either in industry / company, academic / research organizations, government bodies, or international institutions. The objective is to provide practical exposure, industry insights, and real-world experience relevant to the student's field of study.

18.1.3 In-plant Training: student shall undergo training / industrial exposure program aimed at providing with practical insights into real-world working environments. The training may be conducted by industries / companies on-campus or through student visits to industries / companies, government bodies / institutions, or technical organizations.

18.1.4 Skill-based Program: student shall undergo a certified skill-based program of 30 hours / 04 weeks. Skill-based program should cater to Skill-Enhancement, Practical Focus, and Career orientation, Complementary to Curriculum or Industry relevant.

18.1.5 International Immersion (IM): student shall undergo IM aimed at providing global exposure through collaborations with foreign universities, industries, or research institutions. The International Immersion may include industry visits, expert interactions, and cultural exchange activities, enhancing students' international outlook, communication skills, and professional

readiness.

18.1.6 Recognition of Prior Learning (RPL): student who shall undergo any formally recognize relevant prior work experience, internships, or project-based learning that meet the internship learning outcomes, thereby allowing students to earn internship credit without repeating equivalent practical training.

18.1.7 The number of Internships available for the concerned Academic Term. Further, the available number of Internships / In-plant training / Skill-based Program / IM / RPL shall be awarded to the students by the University on the basis of merit using the CGPA secured by the student and as per the selection criteria. Provided further, the student fulfils the criteria, as applicable, specified by the industry / company, government bodies, academic / research or through certified courses.

18.1.8 A student may opt for Internship / In-plant Training / IM / RPL in an industry / company, government bodies / academic / research institution, international bodies of her / his choice, subject to the condition that the concerned student takes the responsibility to arrange the Internship / In-plant Training / IM / RPL on her / his own. Provided further, that the industry / company, government bodies, academic / research institution national or international offering such Internship / Training confirms to the University that the Internship shall be conducted in accordance with the Program Regulations and Internship Policy of the University / Rubrics.

18.1.9 A student undergoing RPL must submit an application with supporting documents such as experience letters, project reports, employer feedback, certifications, a self-reflection report etc. Application must be submitted before the commencement of the internship semester.

18.1.10 A student selected for an Internship / In-plant Training / Skill-based Program / IM / RPL in an industry / company, government bodies, academic / research institution shall adhere to all the rules and guidelines prescribed in the Internship Policy of the University.

The performance will be assessed and grades awardedas per the university's academic grading policy. It may be included as a non-GPA course (Grade-based Satisfactorily Completed / Not Completed). The student must secure a minimum of 50% of the total marks to be declared as having successfully completed the course.

18.2 Mini Project

A student may opt to do a MiniProject Work for a period of 6-8 weeks in an Industry / Company or academic / research institution or the University Department(s) during the 5th / 6th / 7th Semester as applicable, subject to the following conditions:

18.2.1 The Mini Project Work shall be approved by the concerned HOD and be carried out under the guidance of a faculty member.

18.2.2 The student may do the mini project work in an Industry / Company or academic / research institution of her / his choice subject to the above mentioned condition (Sub-Clause 18.2.1). Provided further, that the Industry / Company or academic / research institution offering such mini project work confirms to the University that the mini project work will be conducted in accordance with the Program Regulations and requirements of the University.

18.3 Capstone Project

A student may undergo a Capstone Project for a period of 12-14 weeks in an industry / company or academic / research institution in the 7th / 8th Semester as applicable, subject to the following conditions:

18.3.1 The Capstone Project shall be in conducted in accordance with the Capstone Project Policy prescribed by the University from time to time.

18.3.2 The selection criteria (minimum CGPA, pass in all Courses as on date, and any other qualifying criteria) as applicable / stipulated by the concerned Industry / Company or academic / research institution for award of the Capstone Project to a student;

18.3.3 The number of Capstone Project available for the concerned Academic Term. Further, the available number of Capstone Project shall be awarded to the students by the University on the basis of merit using the CGPA secured by the student. Provided further, the student fulfils the criteria, as applicable, specified by the Industry / Company or academic / research institution providing the Capstone Project, as stated in Sub-Clause 18.3.2 above.

18.3.4 A student may opt for Capstone Project in an Industry / Company or academic / research institution of her / his choice, subject to the condition that the concerned student takes the responsibility to arrange the Capstone Project on her / his own. Provided further, that the Industry / Company or academic / research institution offering such Capstone Project confirms to the University that the Capstone Project shall be conducted in accordance with the Program Regulations and Capstone Project Policy of the University.

18.3.5 A student selected for a Capstone Project in an industry / company or academic / research institution shall adhere to all the rules and guidelines prescribed in the Capstone Project Policy of the University.

18.4 Research Project / Dissertation

A student may opt to do a Research Project / Dissertation for a period of 12-14 weeks in an Industry / Company or academic / research institution or the University Department(s) as an equivalence of Capstone Project, subject to the following conditions:

18.4.1 The Research Project / Dissertation shall be approved by the concerned HOD and be carried out under the guidance of a faculty member.

The student may do the Research Project / Dissertation in an Industry / Company or academic / research institution of her / his choice subject to the above-mentioned condition (Sub-Clause 18.4.1). Provided further, that the Industry / Company or academic / research institution offering such Research Project / Dissertation confirms to the University that the Research Project / Dissertation work will be conducted in accordance with the Program Regulations and requirements of the University.

19. List of Elective Courses under various Specialisations / Stream Basket

Type of Skill		**Course Caters to**
F - Foundation		GS - Gender Sensitization
S - Skill Development		ES - Environment and sustainability
EM – Employability		HP - Human values and Professional Ethics
EN – Entrepreneurship SS-Soft Skills AT-Aptitude Training		

Table 3.7: Professional Electives Courses/Specialization Tracks – Minimum of 12 credits is to be earned by the student in a particular track and overall, 18 credits.									
Sl. No	Course code	Course Name	L	T	P	C	Contact Hour		
Specialization Track 1: Emerging AI and Computational Intelligence									
1	CAI3400	Image Processing and Analysis	2	0	2	3	4		
2	CAI3401	Big Data Analytics for AI	2	0	2	3	4		
3	CAI3402	Optimization Techniques for Machine Learning	2	0	2	3	4		
4	CAI3403	Deep Reinforcement Learning	2	0	2	3	4		
5	CAI3404	AI in Cybersecurity	2	0	2	3	4		
6	CAI3405	Explainable AI	2	0	2	3	4		
7	CAI3406	Responsible AI	2	0	2	3	4		
8	CAI3407	Agentic AI	2	0	2	3	4		
9	CAI3408	Deep Neural Networks	2	0	2	3	4		
10	CAI3409	Speech Recognition and Synthesis	2	0	2	3	4		
11	CAI3410	AI Chatbots without Programming	2	0	2	3	4		

12	CAI3411	Generative AI	2	0	2	3	4		
13	CAI3412	Machine Learning for Finance	2	0	2	3	4		
14	CAI3431	Fundamentals of Quantum Computing and Machine Learning	2	0	2	3	4		
Specialization Track 2: AI driven Autonomous Systems									
1	CAI3413	Industrial IoT	2	0	2	3	4		
2	CAI3414	Smart Farming	2	0	2	3	4		
3	CAI3415	AI for Autonomous Systems	2	0	2	3	4		
4	CAI3416	Edge Computing	2	0	2	3	4		
5	CAI3417	Cognitive Computing	2	0	2	3	4		
6	CAI3418	Geospatial Data Analytics	2	0	2	3	4		
7	CAI3419	AI for energy consumption optimization	2	0	2	3	4		
Specialization Track 3: Healthcare Data Analytics									
1	CAI3420	Bio Medical Informatics	2	0	2	3	4		
2	CAI3421	Intelligent system for disease prediction and drug discovery	2	0	2	3	4		
3	CAI3422	AI for Medical Imaging	2	0	2	3	4		
4	CAI3423	Genomic Data Science	2	0	2	3	4		
5	CAI3424	Clinical Data Science	2	0	2	3	4		
6	CAI3425	AI in Epidemiology and Public Health Analytics	2	0	2	3	4		
7	CAI3426	Time Series Analysis for Patient Monitoring	2	0	2	3	4		
Track - 4 Special Track									

1	CAI3427	Language Models for Text Mining	2	0	2	3	4		
2	CAI3428	Practical Deep Learning with TensorFlow	2	0	2	3	4		
3	CAI3429	Deep Learning Techniques for Computer Vision	2	0	2	3	4		
4	CSS3426	Front End Full Stack Development	2	0	2	3	4		
5	CSS3427	Java Full Stack Development	2	0	2	3	4		
6	CSS3428	.Net Full Stack Development	2	0	2	3	4		
7	CSSXXXX	Introduction to NVIDIA Accelerated Computing and AI	2	0	2	3	4		
8	CSSXXXX	NVIDIA Applied Artificial Intelligence and Generative AI	2	0	2	3	4		
9	CSSXXXX	Advanced AI Systems and Generative Intelligence using NVIDIA Platforms	2	0	2	3	4		
Track – 5 PayTM Courses									
1	CSSXXXX	Autonomous Learning Systems and Model Optimization	2	0	2	3	4		
2	CSSXXXX	Architectures for Deep Representation and Adaptive Learning	2	0	2	3	4		
3	CSSXXXX	Transformer-based Language Systems and Semantic Intelligence	2	0	2	3	4		
4	CSSXXXX	Cross-Modal AI: Vision, Speech, and Contextual Fusion Models	2	0	2	3	4		
5	CSSXXXX	Generative Intelligence and Structured Uncertainty Modeling	2	0	2	3	4		
6	CSSXXXX	Decentralized Data Platforms and Digital Finance Applications	2	0	2	3	4		
7	CSSXXXX	Mini Project/Hackathon: Applied AI Innovation Challenge	0	0	0	4	4		
8	CSSXXXX	Capstone Project - Integrated AI/ML Solution Development	0	0	0	10	4		
Track – 6 Agricultural Course									

1	CAI3450	IoT in Agriculture	2	0	2	3	4		CSS2264
2	CAI3451	Artificial Intelligence in Agriculture	2	0	2	3	4		CSS2264
3	CAI3452	Crop Monitoring and Disease Detection using AI	2	0	2	3	4		CSS2264
4	CAI3453	Edge AI for Crop Monitoring	2	0	2	3	4		CSS2264
5	CAI3454	Smart Agriculture and Precision Farming	2	0	2	3	4		CSS2264
6	CAI3455	Digital Agriculture with Blockchain	2	0	2	3	4		CSS2264
7	CAI3456	Food Traceability in Blockchain	2	0	2	3	4		CSS2264

Track -5 Mandatory Non-Credited Course (Offered for Lateral Entry students in higher semester wherever applicable in MOOC mode)**

Sl. No.	Course Code	Course Name	L	T	P	C	Contact Hour	Type of Skill
1	LAW7601	Indian Constitution **	1	0	0	0	1	F
2	CHE7601	Environmental Studies **	1	0	0	0	1	F
3	CIV7601	Universal Human Values and Ethics **	1	0	0	0	1	F
4	PPS1025	Industry Readiness Program - I	0	0	2	0	2	SS
5	PPS1026	Industry Readiness Program - II	0	0	2	0	2	SS
6	APT4002	Introduction to Aptitude	0	0	2	0	2	AT
7	APT4004	Aptitude Training - Intermediate	0	0	2	0	2	AT
8	APT4006	Logical and Critical Thinking	0	0	2	0	2	AT

*Mandatory for Students selected for Tech Mahindra and Capgemini

** Offered for Lateral Entry students in higher semester wherever applicable in MOOC mode

+Mandatory for Students Selected for Samsung Innovation Campus

20. List of Open Electives to be offered by the School / Department (Separately for ODD and

Type of Skill		**Course Caters to**
F - Foundation		GS - Gender Sensitization
S - Skill Development		ES - Environment and sustainability
EM – Employability		HP - Human values and Professional Ethics
EN – Entrepreneurship		

Baskets
HSMC - Humanities and Social Sciences (including Management courses) (HSMC)
BSC - Basic Science Courses
ESC - Engineering Science Courses
PCC - Professional Core Course
PEC - Professional Elective Courses
OEC - Open Elective Courses
EEC - Employment Enhancement Courses
MAC - Mandatory Course

Table 3.8: Open Elective Courses Baskets: Minimum Credits to be earned from this Basket is 06

Sl. No.	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**	**Type of Skill/ Focus**	**Course Caters to**	**Prerequisites/ Corequisites**	**Antirequisites**	**Future Courses that need this as a Prerequisite**
Chemistry Basket											
1	CHE1003	Fundamentals of Sensors	3	0	0	3	S	ES	-	-	-
2	CHE1004	Smart materials for IOT	3	0	0	3	S	ES	-	-	-
3	CHE1005	Computational Chemistry	2	0	0	2	S	ES	-	-	-
4	CHE1006	Introduction to Nano technology	3	0	0	3	S	ES	-	-	-
5	CHE1007	Biodegradable electronics	2	0	0	2	S	ES	-	-	-
6	CHE1008	Energy and Sustainability	2	0	0	2	S	ES	-	-	-
7	CHE1009	3D printing with Polymers	2	0	0	2	S	ES	-	-	-
8	CHE1010	Bioinformatics and Healthcare IT	2	0	0	2	S	ES	-	-	-

9	CHE1011	Chemical and Petrochemical catalysts	3	0	0	3	S	ES	-	-	-
10	CHE1012	Introduction to Composite materials	2	0	0	2	S	ES	-	-	-
11	CHE1013	Chemistry for Engineers	3	0	0	3	S	ES	-	-	-
12	CHE1014	Surface and Coatings technology	3	0	0	3	S	ES	-	-	-
13	CHE1015	Waste to Fuels	2	0	0	2	S	ES	-	-	-
14	CHE1016	Forensic Science	3	0	0	3	S	ES	-	-	-
Civil Engineering Basket											
1	CIV1001	Disaster mitigation and management	3	0	0	3	S	-	-	-	-
2	CIV1002	Environment Science and Disaster Management	3	0	0	3	FC	-	-	-	-
3	CIV2001	Sustainability Concepts in Engineering	3	0	0	3	S	-	-	-	-
4	CIV2002	Occupational Health and Safety	3	0	0	3	S	-	-	-	-
5	CIV2003	Sustainable Materials and Green Buildings	3	0	0	3	EM	-	-	-	-
6	CIV2004	Integrated Project Management	3	0	0	3	EN	-	-	-	-
7	CIV2005	Environmental Impact Assessment	3	0	0	3	EN	-	-	-	-
8	CIV2006	Infrastructure Systems for Smart Cities	3	0	0	3	EN	-	-	-	-
9	CIV2044	Geospatial Applications for Engineers	2	0	2	3	EM	-	-	-	-
10	CIV2045	Environmental Meteorology	3	0	0	3	S	-	-	-	-
11	CIV3046	Project Problem Based Learning	3	0	0	3	S	-	-	-	-
12	CIV3059	Sustainability for Professional Practice	3	0	0	3	EN	-	-	-	-
Commerce Basket											

1	COM2001	Introduction to Human Resource Management	2	0	0	2	F	HP/GS	-	-	-
2	COM2002	Finance for Non Finance	2	0	0	2	S	-	-	-	-
3	COM2003	Contemporary Management	2	0	0	2	F	-	-	-	-
4	COM2004	Introduction to Banking	2	0	0	2	F	-	-	-	-
5	COM2005	Introduction to Insurance	2	0	0	2	F	-	-	-	-
6	COM2006	Fundamentals of Management	2	0	0	2	F	-	-	-	-
7	COM2007	Basics of Accounting	3	0	0	3	F	-	-	-	-
Computer Science Basket											
(not to be offered for Computer Science and Engineering students)											
1	CSS2002	Programming in Java	2	0	2	3	S/EM	-	-	-	-
2	CSS2003	Social Network Analytics	3	0	0	3	S	GS	-	-	-
3	CSS2004	Python Application Programming	2	0	2	3	S/ EM	-	-	-	-
4	CSS2005	Web design fundamentals	2	0	2	3	S/ EM/EN	-	-	-	-
5	CSS3111	Artificial Intelligence : Search Methods For Problem Solving	3	0	0	3	S/ EM/EN	-	-	-	-
6	CSS3112	Privacy And Security In Online Social Media	3	0	0	3	S/ EM/EN	-	-	-	-
7	CSS3113	Computational Complexity	3	0	0	3	S/ EM/EN	-	-	-	-
8	CSS3114	Deep Learning for Computer Vision	3	0	0	3	S/ EM/EN	-	-	-	-
9	CSS3115	Learning Analytics Tools	3	0	0	3	S/ EM/EN	-	-	-	-
Design Basket											
1	DES1001	Sketching and Painting	0	0	2	1	S	-	-	-	-
2	DES1002	Innovation and Creativity	2	0	0	2	F	-	-	-	-
3	DES1121	Introduction to UX design	1	0	2	2	S	-	-	-	-
4	DES112	Introduction to	1	0	2	2	S	-	-	-	-

	2	Jewellery Making									
5	DES1124	Spatial Stories	1	0	2	2	S	-	-	-	-
6	DES1125	Polymer Clay	1	0	2	2	S	-	-	-	-
7	DES2001	Design Thinking	3	0	0	3	S	-	-	-	-
8	DES1003	Servicability of Fashion Products	1	0	2	2	F	ES	-	-	-
9	DES1004	Choices in Virtual Fashion	1	0	2	2	F	ES, GS, HP	-	-	-
10	DES1005	Fashion Lifestyle and Product Diversity	1	0	2	2	F	ES, GS, HP	-	-	-
11	DES1006	Colour in Everyday Life	1	0	2	2	F	ES	-	-	-
12	DES2080	Art of Design Language	3	0	0	3	S	-	-	-	-
13	DES2081	Brand Building in Design	3	0	0	3	S	-	-	-	-
14	DES2085	Web Design Techniques	3	0	0	3	S	-	-	-	-
15	DES2089	3D Modeling for Professionals	1	0	4	3	S	-	-	-	-
16	DES2090	Creative Thinking for Professionals	3	0	0	3	S	-	-	-	-
17	DES2091	Idea Formulation	3	0	0	3	S	-	-	-	-
Electrical and Electronics Basket											
1	EEE1002	IoT based Smart Building Technology	3	0	0	3	S	-	-	-	-
2	EEE1003	Basic Circuit Analysis	3	0	0	3	S	-	-	-	-
3	EEE1004	Fundamentals of Industrial Automation	3	0	0	3	S	-	-	-	-
4	EEE1005	Electric Vehicles & Battery Technology	3	0	0	3	S	-	-	-	-
5	EEE1006	Smart Sensors for Engineering Applications	3	0	0	3	S	-	-	-	-
Electronics and Communication Basket											
1	ECE1003	Fundamentals of Electronics	3	0	0	3	F	-	-	-	-

2	ECE1004	Microprocessor based systems	3	0	0	3	F	-	-	-	-
3	ECE3089	Artificial Neural Networks	3	0	0	3	S	-	-	-	-
4	ECE3097	Smart Electronics in Agriculture	3	0	0	3	F/EM	-	-	-	-
5	ECE3098	Environment Monitoring Systems	3	0	0	3	F/EM	-	-	-	-
6	ECE3102	Consumer Electronics	3	0	0	3	F/EM	-	-	-	-
7	ECE3103	Product Design of Electronic Equipment	3	0	0	3	S/F/ EM / EN	-	-	-	-
8	ECE3106	Introduction to Data Analytics	3	0	0	3	F/EM	-	-	-	-
9	ECE3107	Machine Vision for Robotics	3	0	0	3	F/EM	-	-	-	-
English Basket											
1	ENG1008	Indian Literature	2	0	0	2	-	GS/ HP	-	-	-
2	ENG1009	Reading Advertisement	3	0	0	3	S	-	-	-	-
3	ENG1010	Verbal Aptitude for Placement	2	0	2	3	S	-	-	-	-
4	ENG1011	English for Career Development	3	0	0	3	S	-	-	-	-
5	ENG1012	Gender and Society in India	2	0	0	2	-	GS/ HP	-	-	-
6	ENG1013	Indian English Drama	3	0	0	3	-	-	-	-	-
7	ENG1014	Logic and Art of Negotiation	2	0	2	3	-	-	-	-	-
8	ENG1015	Professional Communication Skills for Engineers	1	0	0	1	-	-	-	-	-
DSA Basket											
1	DSA2001	Spirituality for Health	2	0	0	2	F	HP	-	-	-
2	DSA2002	Yoga for Health	2	0	0	2	S	HP	-	-	-
3	DSA2003	Stress Management and Well Being	2	0	0	2	F	-	-	-	-
Kannada Basket											
1	KAN1001	Kali Kannada	1	0	0	1	S	-	-	-	-
2	KAN100	Kannada Kaipidi	3	0	0	3	S	-	-	-	-

	3										
3	KAN2001	Thili Kannada	1	0	0	1	S	-	-	-	-
4	KAN2003	Pradharshana Kale	1	0	2	2	S	-	-	-	-
5	KAN2004	Sahithya Vimarshe	2	0	0	2	S	-	-	-	-
6	KAN2005	Anuvadha Kala Sahithya	3	0	0	3	S	-	-	-	-
7	KAN2006	Vichara Manthana	3	0	0	3	S	-	-	-	-
8	KAN2007	Katha Sahithya Sampada	3	0	0	3	S	-	-	-	-
9	KAN2008	Ranga Pradarshana Kala	3	0	0	3	S	-	-	-	-
Foreign Language Basket											
1	FRL1004	Introduction of French Language	2	0	0	2	S	S	-	-	-
2	FRL1005	Fundamentals of French	2	0	0	2	S	S	-	-	-
3	FRL1009	Mandarin Chinese for Beginners	3	0	0	3	S	S	-	-	-
Law Basket											
1	LAW1001	Introduction to Sociology	2	0	0	2	F	HP		-	-
2	LAW2001	Indian Heritage and Culture	2	0	0	2	F	HP/GS		-	-
3	LAW2002	Introdcution to Law of Succession	2	0	0	2	F	HP/GS		-	-
4	LAW2003	Introduction to Company Law	2	0	0	2	F	HP		-	-
5	LAW2004	Introduction to Contracts	2	0	0	2	F	HP	-	-	-
6	LAW2005	Introduction to Copy Rights Law	2	0	0	2	F	HP	-	-	-
7	LAW2006	Introduction to Criminal Law	2	0	0	2	F	HP	-	-	-
8	LAW2007	Introduction to Insurance Law	2	0	0	2	F	HP	-	-	-
9	LAW2008	Introduction to Labour Law	2	0	0	2	F	HP	-	-	-
10	LAW2009	Introduction to Law of Marriages	2	0	0	2	F	HP/GS	-	-	-
11	LAW2010	Introduction to Patent Law	2	0	0	2	F	HP	-	-	-
12	LAW201	Introduction to	2	0	0	2	F	HP	-	-	-

	1	Personal Income Tax									
13	LAW2012	Introduction to Real Estate Law	2	0	0	2	F	HP	-	-	-
14	LAW2013	Introduction to Trademark Law	2	0	0	2	F	HP	-	-	-
15	LAW2014	Introduction to Competition Law	3	0	0	3	F	HP	-	-	-
16	LAW2015	Cyber Law	3	0	0	3	F	HP	-	-	-
17	LAW2016	Law on Sexual Harrassment	2	0	0	2	F	HP/GS	-	-	-
18	LAW2017	Media Laws and Ethics	2	0	0	2	F	HP/GS	-	-	-
Mathematics Basket											
1	MAT2008	Mathematical Reasoning	3	0	0	3	S	-	-	-	-
2	MAT2014	Advanced Business Mathematics	3	0	0	3	S	-	-	-	-
3	MAT2041	Functions of Complex Variables	3	0	0	3	S	-	-	-	-
4	MAT2042	Probability and Random Processes	3	0	0	3	S	-	-	-	-
5	MAT2043	Elements of Number Theory	3	0	0	3	S	-	-	-	-
6	MAT2044	Mathematical Modelling and Applications	3	0	0	3	S	-	-	-	-
Mechanical Basket											
1	MEC1001	Fundamentals of Automobile Engineering	3	0	0	3	F	-	-	-	-
2	MEC1002	Introduction to Matlab and Simulink	3	0	0	3	S/EM	-	-	-	-
3	MEC1003	Engineering Drawing	1	0	4	3	S	-	-	-	-
4	MEC2001	Renewable Energy Systems	3	0	0	3	F	ES	-	-	-
5	MEC2002	Operations Research & Management	3	0	0	3	F	-	-	-	-
6	MEC2003	Supply Chain Management	3	0	0	3	S/ EM/ EN	-	-	-	-
7	MEC2004	Six Sigma for Professionals	3	0	0	3	S/EM	-	-	MEC 2008	-
8	MEC200	Fundamentals of	3	0	0	3	F	-	-	-	-

	5	Aerospace Engineering									
9	MEC2006	Safety Engineering	3	0	0	3	S/EM	ES	-	-	-
10	MEC2007	Additive Manufacturing	3	0	0	3	F/EM	-	-	-	-
11	MEC3069	Engineering Optimisation	3	0	0	3	S/EM	-	-	-	-
12	MEC3070	Electronics Waste Management	3	0	0	3	F/S	ES	-	-	-
13	MEC3071	Hybrid Electric Vehicle Design	3	0	0	3	S/EM	ES	-	-	-
14	MEC3072	Thermal Management of Electronic Appliances	3	0	0	3	S/EM	-	-	-	-
15	MEC3200	Sustainable Technologies and Practices	3	0	0	3	S/EM	-	-	-	-
16	MEC3201	Industry 4.0	3	0	0	3	S/EM	-	-	-	-
Petroleum Basket											
1	PET1011	Energy Industry Dynamics	3	0	0	3	FC	ES	-	NIL	-
2	PET1012	Energy Sustainability Practices	3	0	0	3	FC	ES	-	NIL	-
Physics Basket											
1	PHY1003	Mechanics and Physics of Materials	3	0	0	3	FC / SD				
2	PHY1004	Astronomy	3	0	0	3	FC				
3	PHY1005	Game Physics	2	0	2	3	FC / SD				
4	PHY1006	Statistical Mechanics	2	0	0	2	FC				
5	PHY1007	Physics of Nanomaterials	3	0	0	3	FC				
6	PHY1008	Adventures in nanoworld	2	0	0	2	FC				
7	PHY2001	Medical Physics	2	0	0	2	FC	ES			
8	PHY2002	Sensor Physics	1	0	2	2	FC / SD				
9	PHY2003	Computational Physics	1	0	2	2	FC				
10	PHY2004	Laser Physics	3	0	0	3	FC	ES			
11	PHY200	Science and	3	0	0	3	FC	ES			

	5	Technology of Energy									
12	PHY2009	Essentials of Physics	2	0	0	2	FC				
Management Basket- I											
1	MGT2007	Digital Entrepreneurship	3	0	0	3	S/EM/ EN	-	-	-	-
2	MGT2015	Engineering Economics	3	0	0	3	S	-	-	-	-
3	MGT2023	People Management	3	0	0	3	S/EM/ EN	HP	-	-	-
Management Basket- II											
1	MGT1001	Introduction to Psychology	3	0	0	3	F	HP	-	-	-
2	MGT1002	Business Intelligence	3	0	0	3	EN	-	-	-	-
3	MGT1003	NGO Management	3	0	0	3	S	-	-	-	-
4	MGT1004	Essentials of Leadership	3	0	0	3	EM/ EN	GS/ HP	-	-	-
5	MGT1005	Cross Cultural Communication	3	0	0	3	S/EM/ EN	HP	-	-	-
6	MGT2001	Business Analytics	3	0	0	3	S/ EM/EN	-	-	-	-
7	MGT2002	Organizational Behaviour	3	0	0	3	F	HP	-	-	-
8	MGT2003	Competitive Intelligence	3	0	0	3	S	-	-	-	-
9	MGT2004	Development of Enterprises	3	0	0	3	S/EM/ EN	-	-	-	-
10	MGT2005	Economics and Cost Estimation	3	0	0	3	S/EM	-	-	-	-
11	MGT2006	Decision Making Under Uncertainty	3	0	0	3	S	-	-	-	-
12	MGT2008	Econometrics for Managers	3	0	0	3	S	-	-	-	-
13	MGT2009	Management Consulting	3	0	0	3	S/EM/ EN	-	-	-	-
14	MGT2010	Managing People and Performance	3	0	0	3	S/EM/ EN	HP/GS	-	-	-
15	MGT2011	Personal Finance	3	0	0	3	F	-	-	-	-
16	MGT2012	E Business for Management	3	0	0	3	S/EM	-	-	-	-
17	MGT2013	Project Management	3	0	0	3	EN / EM	GS/HP/ES	-	-	-
18	MGT2014	Project Finance	3	0	0	3	EN / EM	HP	-	-	-

19	MGT2016	Business of Entertainment	3	0	0	3	EM/ EN	-	-	-	-
20	MGT2017	Principles of Management	3	0	0	3	S/EM/ EN	-	-	-	-
21	MGT2018	Professional and Business Ethics	3	0	0	3	S/EM/ EN	HP	-	-	-
22	MGT2019	Sales Techniques	3	0	0	3	S/EM/ EN	HP	-	-	-
23	MGT2020	Marketing for Engineers	3	0	0	3	S/EM/ EN	HP	-	-	-
24	MGT2021	Finance for Engineers	3	0	0	3	S/EM/ EN	HP	-	-	-
25	MGT2022	Customer Relationship Management	3	0	0	3	S/EM/ EN	HP	-	-	-
26		Introduction to Economics	3	0	0	3	S/EM/ EN	HP	-	-	-
Media Studies Basket											
1	BAJ3050	Corporate Filmmaking and Film Business	0	0	4	2	EM	HP	-	-	-
2	BAJ3051	Digital Photography	2	0	2	3	EM	HP	-	-	-
3	BAJ3055	Introduction to News Anchoring and News Management	0	0	2	1	EM	-	-	-	-

21. List of MOOC Courses

21.1 Presidency University students are given the opportunity to study abroad in International Universities through a selection process coordinated by the Office of International Affairs (OIA). Such selected students need to complete their credits for the semester that they are abroad in the following way:

21.1.1. The student needs to study and complete School Core and Program Core Courses in offline mode only.

21.1.2. Massive Open Online Course (MOOC) courses maybe given for Open Elective and Discipline Elective Courses. These courses need to be approved by the concerned BOS and Academic Council from time to time.

21.1.3. SWAYAM/NPTEL/ other approved MOOCs shall be approved by the concerned Board of Studies and placed in the concerned PRC.

21.1.4. Student shall register for these courses in the ERP of Presidency University.

21.1.5. For these MOOC courses faculty coordinators are identified. These faculty should have undergone similar MOOC courses and therefore should be familiar with the mode of class conduction, types of assessments and evaluation procedures.

21.1.6. Study materials shall be provided to the students as video lectures shared by the MOOCs Coordinator(s), or the students may access the approved MOOCs Portal directly. The mode of class conduction is determined by the MOOCs coordinator(s) as detailed in the Course Catalogue and Course Plan.

21.1.7. The question paper shall be prepared by the MOOCs coordinator(s).

21.1.8. Students write the exams in online mode. These exams are scheduled and conducted by the

School.

21.1.9. Results are evaluated by School and given to the Office of the Controller of Examinations (CoE).

21.1.10. The details of the duration, credits and evaluation are given below:

Sl#	Duration	Credits	Evaluation
1.	12 weeks	3	Continuous Assessment –50 Marks Mid Term –50 Marks End Term-100 Marks
2.	8 weeks	2	Mid Term-50 Marks End Term-100 Marks
3	4 weeks	1	End Term-100 Marks

21.2 List of MOOC – B.Tech. Computer Science and Engineering Program (Artificial Intelligence and Machine Learning).

Table 3.9 : MOOC Discipline Elective Courses				
Sl.No	**Course Code**	**Course Name**	**Credits**	**L-T-P-C**
1	CSE3111	Artificial Intelligence: Search Methods for Problem Solving	3	3-0-0-3
2	CSE3112	Privacy and Security in Online Social Media	3	3-0-0-3
3	CSE3113	Computational Complexity	3	3-0-0-3
4	CSE3114	Deep Learning for Computer Vision	3	3-0-0-3
5	CSE3115	Learning Analytics Tools	3	3-0-0-3
6	CSE502	Technical Skills in JAVA	3	0-0-6-3
7	CSE503	Technical Skills in Python	3	0-0-6-3
8	CSE504	Comprehensive Technical Skills	5	0-0-10-5
9	CSE505	The Joy of Computing Using Python	3	3-0-0-3
10	CSE3119	Coding Skills in Python	3	3-0-0-3
11	CSE3121	Parallel Computer Architecture	3	3-0-0-3
12	CSE3124	Games and Information	3	3-0-0-3

13	CSE3140	Introduction to Industry 4.0 and Industrial Internet of Things	3	3-0-0-3
14	CSE3142	Affective Computing	3	3-0-0-3
15	CSE3196	Foundations of Cyber Physical Systems	3	3-0-0-3
16	CSE3197	Getting Started with Competitive Programming	3	3-0-0-3
17	CSE3198	GPU Architectures and Programming	3	3-0-0-3
18	CSE3199	Artificial Intelligence: Knowledge Representation and Reasoning	3	3-0-0-3
19	CSE3200	Programming in Modern C++	3	3-0-0-3
20	CSE3201	Circuit Complexity Theory	3	3-0-0-3
21	CSE3202	Basics of Computational Complexity	3	3-0-0-3
22	CSE3212	Introduction to Computer and Network Performance Analysis using Queuing	1	1-0-0-1
23	CSE3213	C Programming and Assembly Language	1	1-0-0-1
24	CSE3214	Python for Data Science	1	1-0-0-1
25	CSE3215	Software Conceptual Design	1	1-0-0-1
26	CSE3117	Industrial Digital Transformation	3	3-0-0-3
27	CSE3118	Blockchain for Decision Makers	3	3-0-0-3
28	CSE3349	Technology for Lawyers	3	3-0-0-3
29	CSE3430	Deep Learning for Natural Language Processing	3	3-0-0-3
30	CSE3431	Machine Learning for Engineering and Science Applications	3	3-0-0-3
31	CSE3432	Algorithms in Computational Biology and Sequence Analysis	3	3-0-0-3
32	CSE3433	Introduction to Large Language Models (LLMs)	3	3-0-0-3
33	CSE3434	Quantum Algorithms and Cryptography	3	3-0-0-3
34	CAI3430	Responsible & Safe AI Systems	3	3-0-0-3
35	CCS3416	Practical Cyber Security for Cyber Security Practitioners	3	3-0-0-3
36	IST3409	Design & Implementation of Human-Computer Interfaces	3	3-0-0-3

21.3 MOOC - Open Elective Courses for B. Tech. (Computer Science and Engineering- Artificial Intelligence and Machine Learning)

Table 3.10: MOOC Open Elective Courses Courses duration is 4 weeks (01 credit)/ 8 weeks (02 credits)/ 12 weeks (03 credits)				
Sl. No.	**Course Code**	**Course Name**	**Total Credits**	**L-T-P-C**
1	BBA2022	Supply Chain digitization	3	3-0-0-3
2	BBA2021	E Business	3	3-0-0-3
3	BBB2016	Business Analytics for Management Decisions	3	3-0-0-3
4	BBB2015	Artificial Intelligence for Investments	3	3-0-0-3
5	MEC3001	Design and Development of Product	1	1-0-0-1
6	ENG3004	Perspectives of Neurolinguistics	1	1-0-0-1
7	PPS4009	Working in Contemporary Teams	1	1-0-0-1
8	MGT3001	Data Analysis and Decision Making	3	3-0-0-3
9	MEC3001	Design and Development of Product	1	1-0-0-1
10	EEE3105	Microsensors and Nanosensors	3	3-0-0-3
11	CIV3065	Drone Systems and Cont¯rol	3	3-0-0-3
12	ECE3183	Neural Networks for Signal Processing - I	3	3-0-0-3
13	CIVXXXX	Disaster Management	3	3-0-0-3

22. Recommended Semester Wise Course Structure / Flow including the Programme / Professional Elective Paths / Options

Sl. No.	**Course Code**	**Course Name**	**L**	**T**	**P**	**Credits**	**Contact Hours**	**Type of Skill**	**Basket**
Semester 1 - PHY Cycle			**15**	**1**	**12**	**21**	**28**		
1	MAT1211	Linear Algebra, Differential Equations and Statistical Methods	3	1	0	4	4	F	BSC
2	PHY2501	Optoelectronics and Quantum Physics	3	0	0	3	3	F	BSC
3	MEC1006	Engineering Graphics	2	0	0	2	2	S	ESC
4	ENG1030	Fundamentals of Communication	1	0	2	2	3	S	HSMC
5	CSS1500	Computational Thinking using Python	2	0	2	3	4	S	ESC
6	ECE2022	Digital Design	2	0	0	2	2	F/S	ESC
7	PHY2504	Optoelectronics and Quantum Physics Lab	0	0	2	1	2	F	BSC

8	PPS1025	Industry Readiness Program – I	0	0	2	0	2	SS	MAC
9	ECE2052	Digital Design Lab	0	0	2	1	2	F/S	ESC
10	ECE1512	Design Thinking in Practice	2	0	2	3	4	S/EM	HSMC
Semester 2 - CHE Cycle			**14**	**1**	**12**	**20**	**27**		
1	MAT1212	Mathematical foundations for Machine Learning and Data Analysis	3	1	0	4	4	F	BSC
2	CIV1200	Foundations of Integrated Engineering	2	0	0	2	2	S	ESC
3	CHE2501	Chemistry of Smart Materials	3	0	0	3	3	S	BSC
4	CHE2502	Chemistry of Smart Materials Lab	0	0	2	1	2	S	BSC
5	CSS2200	Problem Solving using C	2	0	0	2	2	S	PCC
6	ENG2030	Communication in Practice	1	0	2	2	3	S	HSMC
7	EEE1200	Basics of Electrical and Electronics Engineering	3	0	0	3	3	F/S	ESC
8	CSS2201	Problem Solving using C Lab	0	0	4	2	4	S	PCC
9	PPS1026	Industry Readiness Program – II	0	0	2	0	2	SS	MAC
10	EEE1251	Basics of Electrical and Electronics Engineering Laboratory	0	0	2	1	2	F/S	ESC
Semester 3			**19**	**1**	**10**	**23**	**30**		
1	MAT2311	Vector Calculus, Numerical Methods and Sampling Distributions	3	1	0	4	4	EM	BSC
2	CSS2251	Data Communication and Computer Networks	3	0	0	3	3	S	PCC
3	CSS2252	Data Communication and Computer Networks Lab	0	0	2	1	2	S	PCC
4	CSS2253	Data Structures	3	0	0	3	3	S	PCC
5	CSS2254	Data Structures Lab	0	0	2	1	2	S	PCC

6	CSS2255	Object Oriented Programming Using Java	3	0	0	3	3	S	PCC
7	CSS2256	Object Oriented Programming Using Java Lab	0	0	2	1	2	S	PCC
8	CSS2257	Computer Organization and Architecture	3	0	0	3	3	S	PCC
9	CSS2264	Essentials of AI	3	0	0	3	3	S/EM	ESC
10	CSS2265	Essentials of AI Lab	0	0	2	1	2	S/EM	ESC
11	APT4002	Introduction to Aptitude	0	0	2	0	2	AT	MAC
12	CIV7601	Universal Human Values and Ethics	1	0	0	0	1	S	MAC
Semester 4			**19**	**1**	**10**	**23**	**30**		
1	MAT2321	Discrete Mathematical Structures	3	1	0	4	4	F	BSC
2	CSS2260	Database Management Systems	3	0	0	3	3	S	PCC
3	CSS2261	Database Management Systems Lab	0	0	2	1	2	S	PCC
4	CAI2500	Machine Learning	3	0	0	3	3	S	PCC
5	CAI2501	Machine Learning Lab	0	0	4	2	4	S	PCC
6	CAI2511	Ethics of AI	2	0	0	2	2	S	PCC
7	CSS2262	Analysis of Algorithms	3	1	0	4	4	S	PCC
8	CSS2263	Analysis of Algorithms Lab	0	0	2	1	2	S	PCC
9	APT4004	Aptitude Training-Intermediate	0	0	2	0	2	AT	MAC
10	CHE7601	Environmental Studies	1	0	0	0	1	F	MAC
11	FIN1002	Essentials of Finance	3	0	0	3	3	S	HSMC
Semester 5			**17**	**0**	**12**	**24**	**29**		
1	CSS2266	Theory of Computation	3	0	0	3	3	S	PCC
2	CAI2502	Deep Learning	3	0	0	3	3	S	PCC
3	CAI2503	Deep Learning Lab	0	0	4	2	4	S	PCC
4	CAI2512	Neural Networks and Fuzzy Logic	3	0	0	3	3	S	PCC

5	CAI2509	Web Technologies using AI	3	0	0	3	3	S	PCC
6	CAI2510	Web Technologies using AI Lab	0	0	2	1	2	S	PCC
7	CSS2269	Operating Systems	3	0	0	3	3	S	PCC
8	CSS2270	Operating Systems Lab	0	0	2	1	2	S	PCC
9	CSSXXXX	Professional Elective – I	2	0	2	3	4	S/EM	PEC
10	CSS7000	Internship	0	0	0	2	0	S/EM	PRW
11	APT4006	Logical and Critical Thinking	0	0	2	0	2	AT	MAC
Semester 6			**16**	**0**	**14**	**22**	**30**		
1	CAI2507	Reinforcement Learning	2	0	0	2	2	S	PCC
2	CAI2508	Reinforcement Learning Lab	0	0	2	1	2	S	PCC
3	CAI2504	Natural Language Processing	3	0	0	3	3	S	PCC
4	CAI2505	Natural Language Processing Lab	0	0	2	1	2	S	PCC
5	CSS2271	Software Design and Development	3	0	0	3	3	S	PCC
6	CSS2274	Competitive Programming and Problem Solving	0	0	4	2	4	S/EM	ESC
7	CSSXXXX	Professional Elective – II	2	0	2	3	4	S/EM	PEC
8	CSSXXXX	Professional Elective – III	2	0	2	3	4	S/EM	PEC
9	XXXXXXX	Open Elective – I	3	0	0	3	3	S	OEC
10	LAW7601	Indian Constitution	1	0	0	0	1	F	MAC
11	APT4005	Aptitude for Employability	0	0	2	1	2	AT	HSMC
Semester 7			**9**	**0**	**8**	**17**	**17**		
1	CSSXXXX	Professional Elective – IV	2	0	2	3	4	S/EM	PEC
2	CSSXXXX	Professional Elective – V	2	0	2	3	4	S/EM	PEC
3	CSSXXXX	Professional Elective – VI	2	0	2	3	4	S/EM	PEC
4	XXXXXXX	Open Elective – II	3	0	0	3	3	S	OEC
5	PPS3018	Preparedness for Interview	0	0	2	1	2	SS	HSMC
6	CSS7100	Mini Project	0	0	0	4	0	S/EM	PRW
Semester 8			0	0	0	10	0		
1	CSS7300	Capstone Project	0	0	0	10	0	S/EM	PRW

Open Elective-VI**

Students who have not earned the 15 credits of Open Elective until 7th semester are eligible towards the registration and completion of the Open Elective VI course under NPTEL MOOC Swayam

23. Course Catalogue

Course Catalogue of all Courses Listed including the Courses Offered by other School / Department and Discipline / Programme Electives – Course Code, Course Name, Prerequisite, Anti-requisite, Course Description, Course Outcome, Course Content (with Blooms Level, CO, No. of Contact Hours), Reference Resources.

Course Catalogues:

Course Code: MAT1211	Course Title: Linear Algebra, Differential Equations and Statistical Methods Type of Course:1] School Core	L-T- P- C	3	1	0	4
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-requisites	NIL					
Course Description	This course introduces fundamental mathematical concepts from linear algebra, differential calculus, ordinary differential equations and statistics. The content includes matrices, systems of linear equations, eigenvalues and eigenvectors, functions of several variables, differential equations, correlation and regression analysis, and curve fitting techniques. The course combines theoretical developments with illustrative applications drawn from scientific and engineering settings. The course emphasizes the application of these mathematical tools in solving problems in science and engineering. It extends student's problem-solving abilities through analytical and computational approaches, with focus on both theoretical understanding and applications.					
Course Objective	The objective of this course is to provide students with a strong mathematical foundation required for engineering and scientific studies. The course introduces fundamental concepts from linear algebra, differential calculus, ordinary differential equations, and statistical methods that are widely used in modeling, analysis and problem solving. It aims to familiarize students with mathematical techniques for representing engineering systems, studying relationships among variables, describing dynamic processes and interpreting data. The course serves as a foundation to prepare students for advanced engineering subjects that rely on mathematical modeling, computation and analytical methods.					

Course Out Comes	On successful completion of the course the students shall be able to: CO1: Use linear algebra concepts to evaluate properties of matrices and quadratic forms. (L3) CO2: Apply matrix-based methods to solve systems of linear equations. (L3) CO3: Use fundamental theorems and analytical techniques of differential calculus to single variable functions. (L3) CO4: Apply the concepts of partial derivatives to multivariable functions. (L3) CO5: Solve first order ordinary differential equations using analytical methods. (L3) CO6: Employ statistical techniques for data analysis and to interpret relationships between variables. (L3)	
Course Content:		
Module 1	Fundamentals of Linear Algebra	(15L+5T=20 Sessions)
Matrices, echelon forms, rank of matrix, Systems of linear equations: Consistency, solution to system of linear homogeneous and non-homogeneous equations- Gauss elimination method, Gauss-Jordan method (Self-study), LU decomposition method, Network analysis using linear systems, Design of traffic patterns, Electrical circuits. Eigenvalue and eigenvector, diagonalization of symmetric matrices, quadratic forms.		
Module 2	Differential Calculus	(15L+5T=20 Sessions)
Introduction, limit, continuity, Rolle's theorem, mean value theorems, Indeterminate forms and L'Hôpital's rule. Partial derivatives, Euler's theorem, total derivative, Jacobians, Taylor's and Maclaurin's theorems, maxima, minima and saddle points, method of Lagrange multipliers, Current Growth in an RL Circuit, Velocity Estimation, Fuel consumption optimization, Cost function optimization.		
Module 3	Ordinary Differential Equations	(6L+2T=8 Sessions)
Definition, types of differential equations-variable separable, homogeneous, exact, linear and Bernoulli's equations, Equations not of first degree: equations solvable for p, equations solvable for y, equations solvable for x and Clairaut's type, Newton's law of cooling, Current in RL circuit.		
Module 4	Statistical Methods and Curve Fitting	(9L+3T=12 Sessions)
Measures of central tendency, moments, skewness, and kurtosis. Correlation - Karl Pearson's coefficient of correlation, rank correlation, multiple correlation. Regression analysis - lines of regression, multiple regression. Principle of least square, curve fitting- straight line, parabola and exponential curves, Stress and strain curve fitting, Feature selection and Prediction models.		
Targeted Applications & Tools that can be used: • Linear Algebra is applied to solve systems of equations in circuits, traffic flow, and network analysis, with eigenvalues and quadratic forms aiding stability and optimization studies. • Differential Calculus supports process speed analysis, velocity estimation, and optimization		

techniques such as minimizing fuel consumption and cost functions.

- Differential Equations model dynamic behaviours like current in RL circuits, heat transfer in materials, and growth or decay in environmental and nuclear engineering.
- Statistical Methods and Curve Fitting enable effective data analysis through correlation, regression, and curve fitting, supporting predictive modeling and interpretation of experimental data such as stress-strain relationships.

Software: MATLAB, Python, R Programming, Mathematica, Scilab, Octave

Assignment:

Select any one simple differential equation pertaining to the respective branch of engineering, identify the dependent and independent variable – obtain the solution and compare the solution sets by varying the values of the dependent variable.

Text Book

1. B. S. Grewal, *Higher Engineering Mathematics*, 44th ed. New Delhi, India: Khanna Publishers, 2017.
2. D. C. Montgomery and G. C. Runger, *Applied Statistics and Probability for Engineers*, 7th ed. Hoboken, NJ, USA: Wiley, 2018.

References:

1. P. V. O'Neil, *Advanced Engineering Mathematics*, 8th ed. Boston, MA, USA: Cengage Learning, 2018.
2. R. I. Miller and J. E. Freund, *Probability and Statistics for Engineers*, 9th ed. Noida, India: Pearson Education Ltd., 2017.
3. E. Kreyszig, *Advanced Engineering Mathematics*, 10th ed. Hoboken, NJ, USA: John Wiley & Sons, 2011.
4. V. Henner, T. Belozerova, and M. Khenner, *Ordinary and Partial Differential Equations*. Boca Raton, FL, USA: CRC Press, 2013.

E-resources/ Web links:

1. https://nptel.ac.in/courses/111105121
2. https://nptel.ac.in/courses/111106146
3. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_106839
4. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_61605
5. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_134719
6. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=TEXTBOOK_LIBRARY01_06082022_174

Topics relevant to SKILL DEVELOPMENT: The course focuses on the concepts of differential calculus,

differential equations and statistical methods with reference to specific engineering problems. The course is of both conceptual and analytical type in nature through Problem solving. This is attained through the assessment component mentioned in course handout.	
Catalogue prepared by	Dr. Kavita Permi, Dr. Rahul M. P. and Dr. Athmakoori Prashant
Recommended by the Board of Studies on	
Date of Approval by the Academic Council	

Course Code: PHY2501	Course Title: Optoelectronics and Quantum Physics Type of Course: School Core	L-T-P-C	3-0-0-3
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	The purpose of this course is to enable the students to understand the fundamentals, working and applications of optoelectronic devices and to develop the basic abilities to appreciate the applications of advanced microscopy and quantum computers. The course develops the critical thinking and analytical skills.		
Course Out Comes	On successful completion of the course the students shall be able to: **CO1:** Understand the concepts of electrical conducting properties of metal, semiconductor, and superconductivity. — Level 2 – Understand **CO2:** Understand the principles of quantum mechanics. — Level 2 – Understand **CO3:** Discuss the quantum concepts used in quantum computers. — Level 2 – Understand **CO4:** Explain the applications of lasers and optical fibers in various technological fields. — Level 2 – Understand		
Course Objective	The objective of the course is to familiarize the learners with the concepts of "Applied Physics for Computer Science Cluster "and to attain the basic knowledge related to quantum mechanics and computation.		
Course Content:			
Module 1	Electrical Conductivity Of Solids And Semiconducting Devices	Assignment	11 Sessions
Topics: Classification of materials based on bandgap, Fermi energy and Fermi level, Fermi level in			

semiconductors,

Law of mass action, Electrical conductivity of a semiconductor, Hall effect, Superconductivity, p-n junctions, Zener diode, Solar cells, I-V characteristics, and LEDs

Module 2	Quantum Mechanics	Assignment	11 Sessions

Topics: Introduction, de-Broglie hypothesis, Heisenberg's uncertainty principle- statement and physical significance. Wave function-properties and physical significance. Schrodindger's time independent wave equation, Probability density and normalization of wave function. Wave Function in Ket Notation: Matrix form of wave function, Identity operator, Determination of I|0> and I|1>, Pauli Matrices and its operations on 0 and 1 states, Mention of Conjugate and Transpose, Unitary Matrix U, Examples: 2x2 Matrices and their multiplication (Inner Product), Probability, Orthogonality

Module 3	Quantum Computing	Term paper	12 Sessions

Topics: Introduction to quantum computing, Moore's law & its end, Differences between classical and quantum computing, Concept of Qubit and its properties, . Representation of qubit by Bloch sphere, Quantum Gates: Single Qubit Gates: Quantum Not Gate, Pauli Z Gate, Hadamard Gate, Phase Gate (or S Gate), T Gate. Multiple Qubit Gates: Controlled gate - CNOT Gate, (Discussion for 4 different input states). Representation of Swap gate, Controlled - Z gate, Toffoli gate. Problems.

Module 4	Lasers And Optical Fibers	Term paper	11 Sessions

Topics: Interactions of radiations with matter, expression for energy density of a system under thermal equilibrium in terms of Einstein's coefficients, conditions for LASER action using Einstein's coefficients, Characteristics of laser, conditions and requisites of laser, Principle of optical fibers, Numerical aperture and acceptance angle (Qualitative), Attenuation, Applications: Point to point communication with block diagram, application of optical fibers in endoscopy.

Targeted Application & Tools that can be used:

Areas of application are optoelectronics industry, Solar panel technologies, quantum computing software, electronic devices using transistors and diodes, memory devices, endoscopy, SQUIDS in MRI, Advanced material characterizations using SEM and STM.

Origin, excel and Mat lab soft wares for programming and data analysis.

Project work/Assignment: Mention the Type of Project /Assignment proposed for this course

Assessment Type

Midterm exam

Assignment (review of digital/ e-resource from PU link given in references section - mandatory to submit screen shot accessing digital resource.)

Quiz

End Term Exam

Self-Learning 1. Prepare a comprehensive report on non-conventional energy resources in Karnataka and their pros and cons. 2. Write a report on importance of quantum entanglement in supercomputers.
Text Book Engineering Physics by Avadhanalu, Revised edition, S. Chand Publications,2024. Quantum Computation and Quantum Information, Michael A. Nielsen & Isaac L. Chuang, Cambridge Universities Press, 2010 Edition
References: 1. Elementary Solid state Physics: Principles and Applications by M.A. Omar, 1st Edition, Pearson Publications, 2002. 2. Principles of Quantum Mechanics by R Shankar, 2nd edition, springer Publications, 2011. 3. Optoelectronics: An Introduction by John Wilson and John Hawkes, 3rd edition, Pearson Publications, 2017. 4. Engineering Physics by Gaur and Gupta, Dhanpat Rai Publications, 2012. 5. Introduction to Quantum Mechanics, David J Griffiths, Cambridge University Press, 2019
E-Resourses: https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=553045&site=ehost-live https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=833068&site=ehost-live https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=323988&site=ehost-live https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1530910&site=ehost-live https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=486032&site=ehost-live
Topics relevant to “SKILL DEVELOPMENT”: Fundamentals of materials, Lasers and optical fibers. for Skill Development through Participative Learning Techniques. This is attained through the Assignment/ Presentation as mentioned in the assessment component in course handout.

Course Code: MEC1006	Course Title: Engineering Graphics Type of Course: School Core & Theory Only	L- T- P- C	2-0-0-2
Version No.	1.2		
Course Pre-Requisites	NIL		
Anti-requisites	NIL		

Course Description	The course is designed with the objective of giving an overview of engineering graphics. It is introductory in nature and acquaints the students with the techniques used to create engineering drawings. The course emphasizes on projection of points, lines, planes and solids and isometric projections.
Course Objective	The objective of the course is to familiarize the learners with the concepts of "Engineering Graphics" and attain SKILL DEVELOPMENT through Problem solving methodologies.
Course Outcomes	On successful completion of this course the students shall be able to: **CO1:** Demonstrate competency in Engineering Graphics as per BIS conventions and standards. — Level 3 – Apply **CO2:** Comprehend the theory of projection for drawing projections of points, lines, and planes under different conditions. — Level 2 – Understand **CO3:** Prepare multiview orthographic projections of solids by visualizing them in different positions. — Level 3 – Apply **CO4:** Prepare pictorial drawings using the principles of isometric projections to visualize objects in three dimensions. — Level 3 – Apply

Course Content:

Module 1	Introduction to Drawing	Assignment	Standard technical drawing	7 Sessions

Topics:

Introduction, drawing instruments and their uses, relevant BIS conventions and standards, Lettering, Line conventions, dimensioning, Selection of drawing sheet size and scale.

[02 Hours: Comprehension Level]

Module 2	Orthographic projections of Points, Straight Lines and Plane Surfaces	Assignment	Projection methods Analysis	7 Sessions

Topics:

Introduction, Definitions – Elements of projection and methods of projection, Planes of projection, reference line and conventions adopted. First angle and third angle projections. Projection of Points in all 4 quadrants.Projections of Straight Lines (located in first quadrant/first angle projection only): True and apparent lengths, true and apparent Inclinations to reference planes. (No application problems). Projection of Plane surfaces (First angle projection): Regular plane surfaces – triangle, square, rectangle, pentagon, hexagon and circle – in different positions inclined to both the planes using change of position method only.

[10 Hours: Application Level]

Module 3	Orthographic Projections of Solids	Assignment	Multi-view drawing Analysis	8 Sessions

Topics: Introduction, Projection of right regular prisms, pyramids, cone, hexahedron and tetrahedron in different positions (Problems resting on HP only and First angle projection). [10 Hours: Application Level]				
Module 4	Isometric Projections of Solids (Using isometric scale only)	Assignment	Spatial Visualization	8 Sessions
Topics: Introduction, Isometric scale, Isometric projections of right regular prisms, cylinders, pyramids, cones and their frustums, spheres and hemispheres, hexahedron (cube), and combination of 2 solids, conversion of orthographic view to isometric projection of simple objects. [8 Hours: Application Level]				
Text Book: 1.N. D. Bhatt, "Engineering Drawing: Plane and Solid Geometry," Charotar Publishing House Pvt. Ltd.				
References: K.R. Gopalakrishna, "Engineering Graphics", Subhash Publishers, Bangalore. D. M. Kulkarni, A. P. Rastogi, A. K. Sarkar, "Engineering Graphics with AutoCAD," Prentice Hall. D. A. Jolhe, "Engineering Drawing with Introduction to AutoCAD," Tata McGraw Hill. Web resources: https://nptel.ac.in/courses/112103019				
Topics relevant to "SKILL DEVELOPMENT": Projection in first and third angle for SKILL DEVELOPMENT through Problem Solving methodologies. This is attained through the assessment component mentioned in the course handout.				

Course Code: ENG1030	**Course Title: Fundamentals of Communication Type of Course: Integrated**	**L-T-P-C**	**1**	**0**	**2**	**2**
Version No.	1.0					
Course Pre-	NIL					

requisites	
Anti-requisites	NIL
Course Description	Fundamentals of Communication is a lab intensive, practice first integrated course designed to introduce first-year undergraduate engineering and computer science students to the foundational principles and advanced applications of effective communication across interpersonal, spoken, written, and digital contexts. Anchored in real world scenarios directly relevant to college life, cross-functional technical teams, and digital workplaces, the curriculum develops students' abilities to navigate cultural differences, speak with clarity, read texts critically, and write with precise audience awareness. The course explicitly relies on scenario-based learning, interactive case studies, and hands-on lab applications to ensure that theoretical communication frameworks are immediately actionable. By guiding students from reflective self-awareness to complex digital and AI-mediated content creation, the course builds an enduring communicative foundation optimized for performance-oriented professional practice.
Course Objective	The course aims to explain the key principles of interpersonal, intercultural, and professional communication and apply them to real-world scenarios. It also focuses on developing spoken communication strategies, including structured delivery and impromptu responses, for academic and group contexts. Furthermore, it aims to develop effective writing skills to produce clear, coherent, and audience-appropriate texts across formal and professional contexts.
Course Outcomes	**On successful completion of this course the students shall be able to:** 1. **Understand** the principles of interpersonal, intercultural, and professional communication in academic, social, and technical team contexts. 2. **Demonstrate** audience-centered spoken communication in individual presentations, group discussions, and simulated professional scenarios through vocal control, adaptive register, and collaborative engagement. 3. **Interpret** written, multimodal, and AI-generated texts to identify underlying assumptions, rhetorical strategies, logical errors, and credibility markers across diverse media and professional contexts. 4. **Utilize** plain-language principles and professional conventions to produce clear, purposeful, and appropriately formatted written and multimodal communication, including technical content adapted for non-specialist audiences. 5. **Employ** responsible and critical practices in AI-assisted communication by designing iterative prompts, evaluating generated outputs, and applying ethical frameworks in academic and professional contexts.

Course Content: Fundamentals of Communication				
Module 1	**Communication in a New World**	**CA1 + Project Milestone**	**Deep Listening and Contextual Response Lab**	**06 Sessions**

The topics will be discussed through case studies:

- The Communication Process: How Messages Are Sent, Received, and Misread
- Verbal, Non-Verbal, and Vocal Communication in Everyday Contexts
- Communication Breakdown: Recognising Barriers Across Psychological, Cultural, and Semantic Dimensions
- Active Listening: Attention, Paraphrasing, and Purposeful Response; asking for clarification
- Interpersonal Communication and Self-Awareness *(including cross-cultural dimensions)*

Module 2	**Contemporary Speaking Practices**	**CA2, CA3, CA4**	**Narrative and Adaptive Oral Delivery** **Facilitated Stakeholder Group Discussion** **Simulated Workplace Crisis Briefing**	**15 Sessions**

The topics will be discussed through structured, individual presentations and group discussions:

- Elements of Spoken Communication: Voice, Clarity, and Presence
- Storytelling and Narrative in Spoken Contexts
- Structuring Spoken Messages: From Idea to Delivery
- Speaking with Confidence: Managing Anxiety and Building Presence
- Communicating in Groups: Participation, Productive Disagreement, and Turn-Taking
- Giving and Receiving Feedback Effectively
- Presentation Design and Delivery: Structure, Evidence, and Audience Engagement; eliciting responses
- Speaking for Technical and Non-Technical Audiences

Module 3	**Critical Reading**	**Project Milestone**	**Source Audit and Fallacy Brief**	**09 Sessions**

The topics will be discussed through case studies and real-world scenarios:

- Critical Reading: Purpose, Context, and Analytical Approach
- Perspective and Persuasion: Identifying Tone, Bias, and the Fact-Opinion Distinction
- Understanding Flawed Reasoning *(patterns of weak argumentation in everyday contexts)*
- Evaluating Sources: Credibility, Reliability, and AI-Generated Content; accuracy of the content
- Reading Multimodal Texts: Images, Infographics, and Digital Media

Module 4	**Writing with Purpose**	**CA 5 + Project Milestone**	**Technical Translation and Annotated Script**	**09 Sessions**

The topics will be discussed through case studies:

- Principles of Effective Writing: Clarity, Coherence, Precision and Concise; audience analysis

- Plain Language and Technical Communication: Making Complex Ideas Accessible
- Writing for Professional Contexts: Email, Formal Reports, and Correspondence
- Argumentative Writing: Claim, Evidence, and Logical Progression
- Summary Writing and Short-Form Academic Response

Module 5	**Communication in the Age of AI**	**CA 6 – Project Milestone**	**CA 6: Cross-Disciplinary Case Analysis and Defence**	**06 Sessions**

The topics will be discussed through case studies featuring real-world examples:

- AI as a Communication Tool: Capabilities, Limitations, and the Hallucination Problem
- Prompt Engineering: Communicating Effectively with AI Systems
- AI-Assisted Writing: Drafting, Refining, and Maintaining Your Voice
- Ethical and Legal Dimensions of AI in Communication *(includes India's Digital Personal Data Protection Act, 2023)*
- AI, Bias, and Critical Evaluation: Reading AI Outputs Responsibly

Targeted Application & Tools that can be used: Otter.ai, Microsoft Word/PowerPoint/Teams/Clipchamp/Whiteboard/OneNote/Forms/Copilot, Miro, Padlet, Canva, Mentimeter, AllSides Media Bias Ratings, Ad Fontes Media Bias Chart, Snopes/factcheck.rog, Zotero, NotebookLM, Canva, Adobe Express, CapCut, OBS Studio, Claude, ChatGPT, Gemini, DeepSeek, Google Docs

Project Work / Assignment:

1. **Multimodal Communication Portfolio** (Project) with multiple milestones: Continuous, milestone based. Developed across all five modules. Final consolidated submission at the end of the course. Develop a progressive, publicly shareable communication portfolio that demonstrates the student's ability to move a technical concept across multiple audiences, formats, and media.
2. **Active Listening and Contextual Response Lab** (Individual): Students listen to an unedited, multi-speaker audio recording of a real or simulated professional scenario. Under timed conditions, students produce a structured written brief consisting of four parts. Assessed on comprehension accuracy, analytical precision, quality of questioning, and practicality of the proposed strategy.
3. **Narrative and Adaptive Oral Delivery** (Individual): Each student delivers a 4-minute structured presentation. At the 3-minute mark, the faculty assessor introduces a counter-perspective or an unexpected constraint, and the student must adapt their argument in real time for the remaining minute. The assessment evaluates: the quality of the prepared 3 minutes; the quality of the improvised adaptation; and the degree to which the full 4 minutes function as a coherent whole.
4. **Facilitated Stakeholder Group Discussion** (Group): Groups of 5–6 students are each assigned a distinct stakeholder position within a shared scenario. and must engage in a live, facilitated discussion to resolve a stated policy impasse. No prepared scripts are permitted; students receive their role brief 15 minutes before the discussion begins. Assessment is individual, not group-level.
5. **Simulated Workplace Crisis Briefing** (Group)**:** Groups of 4–5 students receive an unexpected corporate or institutional emergency brief. Teams have exactly 30 minutes to read the brief, distil the critical information, assign speaking roles, and prepare a 5-minute status update for a simulated leadership panel.

6. **Technical Translation and Annotated Script** (Individual)**:** Students receive a dense, jargon-heavy technical document. Students must: (a) read and fully comprehend the document; (b) identify the core idea that a non-specialist audience needs to understand; and (c) produce an annotated script for a 90-second public explainer.
7. **Cross-Disciplinary Case Analysis and Defence** (Group)**:** Groups of 3–4 students select one case from a faculty-curated shortlist of five to six approved cases.The group delivers a 10-minute structured presentation that goes beyond narrative description - identify the specific communication failures, explain the ethical or regulatory dimensions of those failures, and propose what a communication-competent response would have looked like. This is followed immediately by a 5-minute live Q&A in which individual group members must defend their analytical assertions against probing questions from both faculty and peers.

Text Book(s):

1. Markel, M., & Selber, S. A. (2021). *Technical Communication* (13th ed.). Bedford/St. Martin's.

Reference Book(s):

R1. Carnegie, D. (1936/1998). *How to Win Friends and Influence People*. Simon & Schuster.

R2. Orwell, G. (1946). *Politics and the English Language*.

R3. Strunk, W., & White, E. B. (1999). *The Elements of Style* (4th ed.). Longman.

R4. Kahneman, D. (2011). *Thinking, Fast and Slow*. Farrar, Straus and Giroux.

R5. O'Neil, C. (2016). *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. Crown Publishers.

R6. Adler, R. B., Rodman, G., &DuPré, A. (2019). *Understanding Human Communication* (14th ed.). Oxford University Press.

Web Sources:

- owl.purdue.edu
- plainlanguage.gov
- allsides.com
- adfontesmedia.com
- ted.com
- economist.com/styleguide
- factcheck.org
- ailiteracy.org
- learnprompting.org
- unesco.org/en/artificial-intelligence/ethics

Topics relevant to "SKILL DEVELOPMENT": Active listening, verbal and non-verbal communication, interpersonal communication and self-awareness, elements of spoken communication — voice, clarity, and presence, storytelling and narrative in spoken contexts, structuring spoken messages, speaking with

confidence, group communication and productive disagreement, giving and receiving feedback, presentation design and delivery, speaking for technical and non-technical audiences, critical reading, identifying tone and bias, understanding flawed reasoning, source credibility and information evaluation, reading multimodal texts, principles of effective writing, plain language and technical communication, professional writing — email, reports, and correspondence, argumentative writing, summary writing, prompt engineering, AI-assisted writing, ethical AI use in academic and professional communication, AI bias and critical evaluation.	
Catalogue prepared by	Dr Ajith K and Dr ShahanasPunnilathShanavas
Recommended by the Board of Studies on	
Date of Approval by the Academic Council	

Course Code: **CSS1500**	**Course Title: Computational Thinking using Python** **Type of Course: Lab Integrated Course (ESC)**	**L-T-P-C**	**2-0-2-3**
Version No.	1.0		
Course Pre-requisites	Nil		
Anti-requisites	NIL		
Course Description	This course efficiently introduces fundamental programming concepts including conditionals, loops, functions, lists, strings, tuples, and dictionaries through practical examples and problem-solving techniques. It also covers advanced programming concepts such as exception handling, file processing, modules, and packages using Python. The course emphasizes computational thinking, algorithm development, and implementation of searching and sorting techniques using Python data structures. Additionally, it introduces secure file handling, intelligent input validation, and AI-assisted data processing concepts for real-world applications.		
Course Objective	To develop computational thinking and problem-solving skills using Python by enabling students to design algorithms, implement programs using core programming constructs, apply data structures and file handling techniques, and develop secure and intelligent solutions for real-world applications.		

Course Outcomes	On successful completion of this course, students will be able to: CO1: Apply computational thinking principles to design algorithms for solving basic and intermediate computing problems. — Understand CO2: Implement Python programs using data types, operators, and expressions to perform computational tasks. — Apply CO3: Develop programs using control structures, functions, and recursion to solve structured problems. — Apply CO4: Utilize data structures such as lists, tuples, and dictionaries for efficient data representation and manipulation. — Apply CO5: Design and implement Python programs for secure file handling, exception management, and intelligent data validation for real-world applications. — Apply	
Course Content:		
Module 1	**Computational Thinking And Problem Solving**	**6 Sessions**
Topics: Fundamentals of Computing– Identification of Computational Problems Algorithms, building blocks of algorithms (statements, state, control flow, functions), notation (pseudo code, flow chart, programming language), algorithmic problem solving, simple strategies for developing algorithms (iteration, recursion). Illustrative problems: find minimum in a list, insert a card in a list of sorted cards, guess an integer number in a range, Towers of Hanoi		
Module 2	**Datatypes, Expressions, Statements**	**6 Sessions**
Topics: Python interpreter and interactive mode,debugging; values and types: int, float, boolean, string , and list; variables, expressions, statements, tuple assignment, precedence of operators, comments; Illustrative programs: exchange the values of two variables, circulate the values of n variables, distance between two points.		
Module 3	**Control flow, Functions, Strings**	**6 Sessions**
Topics: Conditionals:Boolean values and operators, conditional (if), alternative (if else),chained conditional (if-elif-else);Iteration: state, while, for, break, continue, pass; Fruitful functions: return values,parameters, local and global scope, function composition, recursion; Strings: string slices,immutability, string functions and methods, string module; Lists as arrays. Illustrative programs: square root, gcd, exponentiation, sum an array of numbers, linear search, binary search.		
Module 4	**Lists, Tuples, Dictionaries**	**6 Sessions**

Topics:

Lists: list operations, list slices, list methods, list loop, mutability, aliasing, cloning lists, list parameters; Tuples: tuple assignment, tuple as return value; Dictionaries: operations and methods; advanced list processing- list comprehension; Illustrative programs: simple sorting, histogram, Students marks statement, Retail bill preparation.

Module 5	**Files**	**6 Sessions**

Files and exceptions: text files, reading and writing files, format operator; command line arguments, errors and exceptions, handling exceptions, modules, packages; Illustrative programs: word count, copy file, Voter's age validation, Marks range validation (0-100), Secure File Handling, Intelligent Error Detection using AI Concepts, Smart Input Validation Techniques, Automated Data Validation Applications.

List of Practical Tasks **30 Sessions**

WEEK 1: Computational Thinking & Algorithms

Lab Sheet 1: Problem Solving & Algorithm Design

Level 1: Write an algorithm and Python program to find the minimum element in a list of numbers.

Level 2: Write an algorithm and Python program to insert an element into a sorted list at the correct position.

Lab Sheet 2: Algorithm Strategies (Iteration & Recursion)

Level 1: Write a Python program to guess a number within a given range using iteration.

Level 2: Write a Python program to solve the Tower of Hanoi problem using recursion.

WEEK 2: Data Types & Expressions

Lab Sheet 3: Data Types & Variables

Level 1: Write a Python program to demonstrate different data types (int, float, boolean, string, list).

Level 2: Write a Python program to perform type conversions and display results.

Lab Sheet 4: Expressions & Operators

Level 1: Write a Python program to swap two variables using tuple assignment.

Level 2: Write a Python program to circulate the values of n variables.

WEEK 3: Expressions & Problem Solving

Lab Sheet 5: Mathematical Expressions

Level 1: Write a Python program to calculate the distance between two points.

Level 2: Write a Python program to evaluate complex arithmetic expressions with operator precedence.

Lab Sheet 6: Debugging & Interactive Mode

Level 1: Write a Python program with intentional errors and debug it.

Level 2: Write a Python program to demonstrate interactive mode operations.

WEEK 4: Conditional Statements

Lab Sheet 7: Decision Making

Level 1: Write a Python program to check whether a number is positive, negative, or zero.

Level 2: Write a Python program to find the largest of three numbers using if-elif-else.

Lab Sheet 8: Advanced Conditions

Level 1: Write a Python program to check whether a number is even or odd.

Level 2: Write a Python program to implement a grading system using chained conditions.

WEEK 5: Iteration (Loops)

Lab Sheet 9: Looping Basics

Level 1: Write a Python program to print numbers from 1 to N using a loop.

Level 2: Write a Python program to compute factorial using loops.

Lab Sheet 10: Loop Control

Level 1: Write a Python program demonstrating break and continue.

Level 2: Write a Python program to generate number patterns using nested loops.

WEEK 6: Functions

Lab Sheet 11: Functions Basics

Level 1: Write a Python function to calculate square root of a number.

Level 2: Write a Python function to find GCD of two numbers.

Lab Sheet 12: Recursion & Scope

Level 1: Write a recursive function to calculate exponentiation.

Level 2: Write a Python program demonstrating local and global variables.

WEEK 7: Strings

Lab Sheet 13: String Operations

Level 1: Write a Python program to demonstrate string slicing.

Level 2: Write a Python program to check whether a string is a palindrome.

Lab Sheet 14: String Methods

Level 1: Write a Python program using string functions (upper, lower, replace).

Level 2: Write a Python program to count vowels and consonants in a string.

WEEK 8: Lists

Lab Sheet 15: List Basics

Level 1: Write a Python program to create and display a list.

Level 2: Write a Python program to find the sum of elements in a list.

Lab Sheet 16: List Operations

Level 1: Write a Python program to demonstrate list slicing.

Level 2: Write a Python program to perform linear search on a list.

WEEK 9: Advanced Lists

Lab Sheet 17: List Methods

Level 1: Write a Python program to demonstrate list methods (append, remove).

Level 2: Write a Python program to sort a list.

Lab Sheet 18: List Comprehension

Level 1: Write a Python program using list comprehension to generate squares.

Level 2: Write a Python program to create a histogram of list values.

WEEK 10: Tuples

Lab Sheet 19: Tuple Basics

Level 1: Write a Python program to create and access tuple elements.

Level 2: Write a Python program to demonstrate tuple assignment.

Lab Sheet 20: Tuples as Return Values

Level 1: Write a Python function returning multiple values using tuple.

Level 2: Write a Python program to swap values using tuples.

WEEK 11: Dictionaries

Lab Sheet 21: Dictionary Basics

Level 1: Write a Python program to create and access dictionary elements.

Level 2: Write a Python program to count frequency of elements using dictionary.

Lab Sheet 22: Dictionary Applications

Level 1: Write a Python program to store student marks using dictionary.

Level 2: Write a Python program to generate a retail bill using dictionary.

WEEK 12: Advanced List Processing

Lab Sheet 23: Sorting & Searching

Level 1: Write a Python program to implement simple sorting.

Level 2: Write a Python program to implement binary search.

Lab Sheet 24: Data Processing

Level 1: Write a Python program to generate student marks statement.

Level 2: Write a Python program to analyze list data (min, max, average).

WEEK 13: File Handling Basics

Lab Sheet 25: File Operations

Level 1: Write a Python program to create and write to a text file.

Level 2: Write a Python program to read and display file contents.

Lab Sheet 26: File Processing

Level 1: Write a Python program to count words in a file.

Level 2: Write a Python program to copy contents from one file to another.

WEEK 14: Exceptions & Validation

Lab Sheet 27: Exception Handling

Level 1: Write a Python program using try-except.

Level 2: Write a Python program to handle multiple exceptions.

Lab Sheet 28: Validation Programs

Level 1: Write a Python program to validate voter's age.

Level 2: Write a Python program to validate marks range (0–100).

WEEK 15: Modules & Advanced Concepts

Lab Sheet 29: Modules & Packages

Level 1: Write a Python program to create and use a module.

Level 2: Write a Python program to use built-in modules.

Lab Sheet 30: Command Line Arguments

Level 1: Write a Python program to accept command-line arguments.

Level 2: Write a Python program to perform operations using command-line inputs

WEEK 15.1: Secure and Intelligent Python Applications

Lab Sheet 31: Secure File Handling and Validation

Level 1: Write a Python program for secure file reading and writing with exception handling.
Level 2: Implement intelligent input validation for user data using exception handling techniques.

Lab Sheet 32: AI-based Python Applications

Level 1: Develop a Python program for automated data validation or log analysis.
Level 2: Implement a simple intelligent application using Python modules and file processing concepts.

Project work/Assignment:

1. Assignment 1 on (Module 1 and Module 2)
2. Assignment 2 on (Module 3 and Module 4 & 5)

Text Book

1. Paul Deitel and Harvey Deitel, "Python for Programmers", Pearson Education, 1st Edition, 2021
2. Eric Matthes, Python Crash Course,: A Hands-On, Project-Based Introduction to Programming, 3rd Edition, 2023

References

1. Allen B. Downey, "Think Python: How to Think like a Computer Scientist", 2nd Edition, O'Reilly Publishers, 2016.
2. Karl Beecher, "Computational Thinking: A Beginner's Guide to Problem Solving and Programming", 1st Edition, BCS Learning & Development Limited, 2017.

Web Resources

W1. https://onlinecourses.nptel.ac.in/noc20_cs70/preview

1. **Topics relevant to development of "Employability":** Data structures using python.
2. **Topics relevant to "PROFESSIONAL ETHICS":** Naming and coding convention for simple programs using python.

Course Code: ECE2022	Course Name: Digital Design Type of Course: Theory Course	L- T-P- C	2-0-0-2
Version No.	1.0		
Course Pre-requisites	NIL		
Co-requisites	ECE2052		
Anti-requisites	NIL		
Course Description	The purpose of this course is to enable the students to appreciate the fundamentals of digital logic circuits and Boolean algebra. Further to focusing on combinational and sequential digital logic circuits. This course is analytical in nature and needs a fundamental knowledge on logical computation with Boolean algebra. The focus of the course will be to discuss the minimization techniques for making low-cost canonical digital circuit implementations. In this course, we emphasize design procedures of digital electronic circuits. Additionally, this course will create a foundation for future courses that includes Computer Architecture, Microprocessors, Microcontrollers and Embedded Systems etc.		
Course Objective	The objective of the course is to familiarize the learners with Design and Analysis of Digital Logic Circuits concepts and to improve the learner's SKILL DEVELOPMENT by using PARTICIPATIVE LEARNING TECHNIQUES through Assessments.		

Course Outcomes	On successful completion of this course the students shall be able to: CO1. Explain the fundamental concepts of Number system, Boolean algebra and Logic gates CO2. Determine the minimum literal Boolean Expression using K-Map minimization techniques CO3. Make use of relevant logic gates for constructing the combinational circuit. CO4. Produce standard and real world combinational logic circuits using minimum number of logic gates CO5. Demonstrate the working principles of flip-flops and latches within the context of sequential logic circuits. CO6. Design synchronous and asynchronous logic circuit using Flip-flops		
Course Content:			
Module 1	Number System, Logic Gates & Boolean function simplification (Comprehension Level)	Assessment	10 Sessions
Topics: Review of Number systems -Overview of Boolean Algebra- Boolean Laws, Rules & Theorems. Boolean function Standard form and Canonical form. Review of logic gates -Basic gates, Universal gates & Special Gates- Boolean functions simplifications using Algebraic functions & Karnaugh-map. SOP and POS simplifications & Implementations.			
Module 2	Combinational Logic circuits: (Application Level)	Assessment	10 Sessions
Topics: Digital logic circuit Introduction –Classification of circuit- Analysis and Design procedure of Combinational Logic circuits - Binary Adder and Subtractor- Magnitude comparator- , Multiplexers - DeMultiplexers, Decoders, Encoders and Priority Encoders-Real world scenario based application design using logic gates.			
Module 3	Sequential logic circuits (Application Level)	Quiz	10 Sessions
Topics: Introduction to sequential circuits- Storage elements: latches and flip flops- Characteristic tables and equations, excitation table- Analysis of clocked sequential circuits- state diagram-state table and design of finite state machines - Registers & Counters.			
Text Book **T1:** Thomas L Flyod, "Digital Fundamentals", Eleventh Edition, Pearson Education. ISBN-10: 132737965. (2014).eBook-[PDF] DIGITAL LOGIC DESIGN FOURTH EDITION FLOYD | abri.engenderhealth.org **T2:** Mano, M. Morris and Ciletti Michael D., "Digital Design", Pearson Education, 6th edition			

., [PDF] Digital Design By M. Morris Mano, Michael D Ciletti Book Free Download (studymaterialz.in)
References **R1:** Jain, R. P., "Modern Digital Electronics", McGraw Hill Education (India). **R2:** Roth, Charles H., Jr and Kinney Larry L., "Fundamentals of logic Design", Cengage Learning
E-Resources/Web Resources 1. https://openFullText.html?DP=https://nptel.ac.in/courses/108105132 2. https://openFullText.html?DP=https://ocw.mit.edu/courses/6-111-introductory-digital-systems-laboratory-fall-2006/ 3. https://openFullText.html?DP=https://www.javatpoint.com/digital-electronics 4. https://openFullText.html?DP=https://www.geeksforgeeks.org/digital-electronics-logic-design-tutorials/ 5. https://openFullText.html?DP=https://circuitverse.org

Course Code: PHY2504	Course Title: Optoelectronics and Quantum Physics Lab Type of Course: 1] School Core	L-T-P-C	0-0-2-1
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	The laboratory provides an opportunity to validate the concepts taught and enhances the ability to use the concepts for technological applications. The laboratory tasks aim to develop following skills: An attitude of enquiry, confidence and ability to tackle new problems, ability to interpret events and results, observe and measure physical phenomena, select suitable equipment, instrument and materials, locate faults in systems.		
Course Out Comes	On successful completion of the course the students shall be able to: **CO1:** Understand electrical and optical properties of materials. — Level 2 – Understand **CO2:** Interpret the results of various experiments to verify the concepts used in optoelectronics and advanced devices. — Level 2 – Understand		
Course Objective	The objective of the course is to familiarize the learners with the concepts of " Applied Physics for Computer Science Cluster "and attain Skill Development through Experiential Learning techniques		
List of Laboratory Tasks: Experiment No. 1: Experimental errors and uncertainty using excel			

Level 1: Calculation of accuracy and precision of a given data

Level 2: propagation of errors in addition, subtraction, multiplication and division.

Experiment N0 2: To determine the wavelength of semiconductor diode Laser and to estimate the particle size of lycopodium powder using diffraction.

Level 1: Determination of Wavelength of Laser

Level 2: Finding the particle size of lycopodium powder.

Experiment No. 3: To determine the proportionality of Hall Voltage, magnetic flux density and the polarity of Charge carrier.

Level 1: To determine the proportionality of Hall Voltage and magnetic flux density

Level 2: To determine the polarity of Charge carrier.

Experiment No. 4: To study the I-V characteristics of a given zener diode in forward and reverse bias conditions.

Level 1: To study I –V characteristics of the given Zener diode in reverse bias and to determine break down voltage.

Level 2: To study I –V characteristics of the given Zener diode in forward bias and to determine knee voltage and forward resistance.

Experiment No. 5: To study input and output characteristics of a given Transistor.

Level 1: To determine the input resistance of a given transistor.

Level 2: To determine current transfer characteristics and transistor parameters of a given transistor.

Experiment No. 6: Determination of Fermi energy and Fermi temperature of a given metal and bimetallic wire.

Level 1: Determination of Fermi energy and Fermi temperature of given metal wire.

Level 2: Determination of Fermi energy and Fermi temperature of given bimetallic wire.

Experiment No. 7: To study the I-V characteristics and I-R characteristics of a solar cell as a function of the irradiance.

Level 1: To study the I-V characteristics

Level 2: I-R characteristics of a solar cell as a function of the irradiance.

Experiment No. 8: Calculate the numerical aperture and study the losses that occur in optical fiber cable. .

Level 1: Calculate the numerical aperture.

Level 2: study the losses that occur in optical fiber cable.

Experiment No. 9: Plotting I-V characteristics in forward and reverse bias for LEDs

and Determination of knee voltage.

Level 1: Plotting I-V characteristics in forward and reverse bias for LEDs

Level 2: Determination of knee voltage.

Experiment No. 10: Determination of Stefan's constant and verification of Stefan-Boltzmann Law.

Level 1: Determination of Stefan's constant

Level 2: Verification of Stefan-Boltzmann Law.

Experiment No. 11: : Dielectric constant

Level 1: Determination of Dielectric constant of given material

Level 2: compare the obtain results with other materials

Experiment No. 12: determine the wavelength of monochromatic light, such as sodium light, using Newton's rings.

Level 1: Determination of wavelength

Level 2: determine the radius of curvature of the Plano-convex lens.

Targeted Application & Tools that can be used:

Areas of application are optoelectronics industry, Solar panel technologies, quantum computing software, electronic devices using transistors and diodes, memory devices, endoscopy, SQUIDS in MRI, Advanced material characterizations using SEM and STM.

Origin, excel and Mat lab soft wares for programming and data analysis.

Project work/Assignment: Mention the Type of Project /Assignment proposed for this course

Assessment Type Midterm exam Assignment (review of digital/ e-resource from PU link given in references section - mandatory to submit screen shot accessing digital resource.) Quiz End Term Exam Self-Learning 1. Prepare a comprehensive report on non-conventional energy resources in Karnataka and their pros and cons. 2. Write a report on importance of quantum entanglement in supercomputers.

Course Code: PPS1025	Course Title: Industry Readiness Program – I Type of Course: Practical Only Course	L-T- P - C	0	0	2	0
Version No.	1.0					
Course Pre-requisites	Students are expected to understand basic English. Students should have desire and enthusiasm to involve, participate and learn.					
Anti-requisites	NIL					
Course Description	This course is designed to enable students to set SMART goals, form professional & personal ethics for success, and learn various email writing techniques. The course will benefit learners in presenting themselves effectively through various activities and learning methodologies.					
Course Objective	The objective of the course is to familiarize the learners with the concepts of "Employability for Young Professionals" and attain SKILL DEVELOPMENT through PARTICIPATIVE LEARNING techniques.					
Course Outcomes	On successful completion of the course the students shall be able to: **CO1:** Define their career goals [Level 1 - Remember] **CO2:** Practice ethical habits for better career success [Level 3 - Apply] **CO3:** Demonstrate effective email writing techniques [Level 3 - Apply]					
Course Content:						
Module 1	Goal Setting & Grooming	Classroom activities	10 Sessions			
Topics: SMART Goals, formal grooming through self-introduction activity Activity: Real world scenarios						
Module 2	Habit Formation	Role plays	10 Sessions			
Topics: Professional and Personal ethics for success and activity-based practice Activity: Students to present 2 min video on building professional ethics						
Module 3	Email Etiquettes	Individual and group presentation	10 Sessions			
Topics: Types of prompts to generate effective or desired results for email etiquettes Activity: Individual student presenting various search prompts						

Targeted Applications & Tools: TED Talks; You Tube Links; Activities
Assignment proposed for this course Assignment 1: SMART Goal Assignment 2: AI tools for prompt search
Continuous Individual Assessment Module 1: Presentation Module 2: Activity based assessment Module 3: Class assessment

Course Code: ECE2052	Course Name: Digital Design Lab Type of Course: Laboratory Course	L- T-P- C	0-0-2-1
Version No.	1.0		
Course Pre-requisites	NIL		
Co-requisites	ECE2022		
Anti-requisites	NIL		
Course Description	This course gives knowledge through the simulation software for design and implementation of various digital logic circuit. The theoretical concepts and working principles of combinational and sequential logic circuits are verified through implementation using openCV. This course provides necessary learning for fundamental concepts of computer and communication engineering. The purpose of this course is to support the students to exhibit the design and analysis skills. The course enhances the Design and Implementation abilities through laboratory tasks. Additionally, it provides an opportunity to verify the theoretical knowledge.		
Course Objective	The objective of the course is to familiarize the learners with the design concepts of digital logic circuit and attain Skill Development through Experiential Learning by simulation-based laboratory Assessments.		
Course Outcomes	On successful completion of the course the students shall be able to CO1. Interpret Boolean laws and the logic gates functionality using available circuit elements in simulator platforms. CO2. Apply the fundamental laws and rules of Boolean algebra for simplified circuit simulation using any simulator tool. CO3. Develop combinational logic circuits by connecting offered gates for basic arithmetic operation CO4. Make use of circuit elements for ensuring the digital logic circuit's truth table perfection. CO5. Examine the performance of sequential logic circuit simulation using flip-		

	flop and latch elements CO6. Demonstrate the operation of clocked and edge-triggered sequential components in an open-source simulation environment

Course Content:

List of Laboratory Tasks:

- **Experiment 1: Logic Gates**.
  Study and verify the operation of basic logic gates such as AND, OR, NOT, NAND, NOR, XOR, and XNOR.

- **Experiment 2: Boolean Algebra**

  Implement and verify Boolean algebra laws and simplification techniques using logic circuits.

- **Experiment 3: Adder**

  Design and analyze half adder and full adder circuits for binary addition.

- **Experiment 4: Subtractor**

  Design and verify half subtractor and full subtractor circuits for binary subtraction.

- **Experiment 5: Magnitude Comparator**

  Design a circuit to compare two binary numbers and determine equality or greater/less conditions.

- **Experiment 6: Design of Combinational Circuits**

  Design and implement combinational logic circuits using Boolean expressions and K-maps.

- **Experiment 7: Multiplexer, Demultiplexer& Applications of MUX**

  Study the working of multiplexers and demultiplexers and implement logic functions using MUX.

- **Experiment 8: Flip-Flops**

  Implement and verify the operation of SR, JK, D, and T flip-flops.

- **Experiment 9: Shift Registers**

  Design and analyze serial and parallel shift registers for data storage and transfer.

- **Experiment 10: Counter Design**

  Design and implement synchronous and asynchronous counters using flip-flops.

Targeted Applications & Tools:

CIRCUIT VERSE	An online digital circuit simulation platform used for designing and testing logic circuits and combinational/sequential circuits.
HDL	Hardware Description Language used to model, design, and

	simulate digital electronic systems and circuits.
VHDL	Very High-Speed Integrated Circuit Hardware Description Language used for designing, simulating, and implementing digital systems in FPGA and ASIC design.

Skill-Building Activities through Experiential Learning:

- Design and simulation of digital circuits using CircuitVerse/HDL tools.
- Hands-on implementation of combinational and sequential circuits.
- Mini exercises on logic simplification and circuit optimization.
- Team-based circuit design and debugging activities.
- Real-time verification of flip-flops, counters, and shift registers.

Text Book

T1: Mano, M. Morris and Ciletti Michael D., "Digital Design", Pearson Education, 6th edition

T2: Thomas L. Floyd "DIGITAL LOGIC DESIGN" ,Pearson Education, fourth edition.

References

R1: Jain, R. P., "Modern Digital Electronics", McGraw Hill Education (India), Fourth Edition

R2: Roth, Charles H., Jr and Kinney Larry L., "Fundamentals of logic Design", Cengage Learning, Seventh Edition

E-Resources/Web Resources

1. https://circuitverse.org/
2. https://ocw.mit.edu/courses/6-111-introductory-digital-systems-laboratory-fall-2006/
3. https://nptel.ac.in/courses/108105132
4. https://www.geeksforgeeks.org/digital-electronics-logic-design-tutorials/
5. https://www.tutorialspoint.com/digital_electronics/index.htm
6. https://github.com/logisim-evolution/logisim-evolution
7. https://www.javatpoint.com/digital-electronics

Course Code: ECE1512	**Course Title: Design Thinking in Practice**	**L- T-P- C**	**2-0-2-3**
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course introduces students to Design Thinking frameworks for identifying real-world user needs and transforming them into innovative technology solutions using embedded systems, sensors, IoT platforms, and AI-enabled computing devices. The course blends human-centered innovation methodologies with hands-on		

	development using Arduino, Aries, ESP32, Raspberry Pi, NVIDIA Jetson, and GPU-based embedded platforms. Students will develop problem-solving skills while learning problem identification, ideation, rapid prototyping, sensor integration, edge AI, and intelligent system development for applications in smart automation, healthcare, agriculture, robotics, and Industry 5.0. The course emphasizes experiential and project-based learning to build future-ready innovators capable of developing scalable, intelligent, and socially impactful products.	
Course Objective	The objective of the course is to enhance students' **employability skills** through **participative learning techniques** by integrating Design Thinking, embedded systems, IoT, and AI technologies to develop innovative, intelligent, and socially impactful real-world solutions.	
Course Outcomes	**On successful completion of the course, the students shall be able to** 1. **Explain** the principles, framework, and process of design thinking 2. **Formulate** Engineering Problem from user challenges, customer expectations, and market opportunities using design thinking process. 3. **Analyze** the Engineering Problem using the basic concepts of microcontroller and sensory devices. 4. **Find** alternate methods to solve the engineering problem by working with embedded computing platforms. 5. **Develop** functional prototypes using embedded boards, sensors, and an intelligent system for the chosen method 6. **Validate** prototype performance through testing, user feedback, and iterative improvements.	
Course Content		
Module 1	**Introduction to Design thinking**	**6 Sessions**
Introduction to Design Thinking Importance of Design Thinking History of Design Thinking- Design Thinking Framework - Design Thinking process - Empathise Define Understanding Purpose - Ideate - What is Creativity? -Thinking Differently Basic Rules of Design Thinking The Creativity Heart Set Everyone is a Designer - Activity-based with case studies		
Module 2	**Basic Concepts of microcontrollers and sensory devices**	**8 Sessions**

Introduction to Arduino, ESP and NodeMCU - Pin Configuration and Architecture - Device and Platform Features - Concept of Digital and Analog Ports - Familiarizing with Arduino Interfacing Board Programming using Arduino IDE and Embedded C - Datatypes and Variables - I/O Functions Sensors, types and its applications - Humidity Sensor - Temperature Sensor - Water Detector / Sensor - PIR Sensor - Ultrasonic Sensor
- Connecting Switches and Actuators - Sensor Interface with ESP Microcontroller - Various Cloud Platforms

Module 3	**Working with embedded computing platforms and 3D printing technology**	**8 Sessions**

Introduction to Raspberry Pi boards, pin diagram, different types of Raspberry Pi boards and their applications, Configuration of Raspberry Pi boards - LED and switch control Exploring various libraries and their functions. GPU fundamentals Configuration of Jetson Nano board, Working with H200 GPU Servers. Introduction to 3D Printer: 3D Printer technology and its working Principles, Applications. Introduction to online Simulators:
Working with Tinker CAD Simulator.

Module 4	**Prototype development and validation techniques**	**8 Sessions**

Importance of Prototyping in Design Thinking Concept Refinement Using Prototypes From Idea to Functional Prototype User Testing and Validation Iteration and Design Optimization Prototype Readiness
for Product Development (DFM & Market Feasibility overview) Activity-based with case studies.

Prototype Development Tasks:
Total Sessions: 30

1. Smart illumination system using Arduino
2. Intelligent Plant Health Monitoring System.
3. Aries Rover Drive and Control System Development
4. IoT-Based Environmental Intelligence System using ESP32
5. Edge Computing Platform Setup using Raspberry Pi
6. Raspberry Pi-based smart lighting Controller
7. VoiceX: AI-Powered Smart Home Automation System
8. VisionAI: Intelligent Object Recognition System.
9. Smart Traffic Management System using NVIDEA Jetson Nano
10. High-Performance AI Vision Analytics using H200-GPU Computing.
11. Rapid Prototype Development using 3D Design and printing

Topics: Types of embedded boards, Thonny Python, Python IDLE, sensors, 3D Printer, Jetson Nano board, GPUs

Targeted Application & Tools that can be used:

Application Area:

Home Automation, Environmental Monitoring, Agriculture and Farming, Industrial Automation, Internet of Things (IoT), Robotics, Wearable Devices, Security Systems, Education and Learning. These are just a few examples of the many application areas where Arduino, Aries, ESP32, Raspberry Pi, and sensors can be applied. The flexibility and affordability of all embedded boards, combined with the wide range of sensors available, allow for endless possibilities in creating innovative projects.

Professionally Used Software: Students can use open-source software, such as Arduino IDE, Blynk console, Tinker CAD, Thonny Python, Python IDLE, etc.

Project work/Assignment:

Projects: At the end of the course, students will be completing the project work on solving many real-time problem statements on different domains, such as women-centric, agritech, medtech, climate cell, robotics, drones, and smart cities etc.

Book/Article review: At the end of each module, a book reference or an article topic will be given to an individual or a group of students. They need to refer to the library resources and write a report on their understanding of the assigned article in an appropriate format. Presidency University Library Link**Presentation:** There will be a review presentation (Phase 1, Phase 2, and Phase 3) from among the groups, where the students will be given a project, and they have to demonstrate the working and discuss the applications for the same

References:

1. Design Thinking by Clarke, Rachel Ivy. Series: Library Futures, Vol. 4. Chicago: ALA Neal-Schuman. 2020. eBook., Database: eBook Collection (EBSCOhost)
2. Neerparaj Rai, "Arduino Projects for Engineers," BPB Publishers, first edition, 2016.
3. Stewart Watkiss "Learn Electronics with Raspberry Pi" Apress Berkeley, CA. Second edition, 2020. ISBN978-1-4842-6348-8
4. Jo Prusa, "Basics of 3D printing", Prusa Research, 3rd edition.
5. Volker Ziemann, "A Hands-On Course in Sensors Using the Arduino and Raspberry Pi (Series in Sensors)", CRC Press, 1st Edition. 2018.
6. Professional CUDA C Programming. Professional CUDA C Programming. Wrox Press, 2014.

Online Resources (e-books, notes, PPTs, video lectures, etc.):

1. Arduino trending Projects - https://www. https://projecthub.arduino.cc
2. Introduction to Arduino - https://onlinecourses.swayam2.ac.in/aic20_sp04/preview
3. Case studies on Wearable technology - https://www.hticiitm.org/wearables
4. Raspberry Pi Projects - https://magpi.raspberrypi.com/articles/category/tutorials/

5. Introduction to internet of thingshttps://nptel.ac.in/courses/106105166>
6. Getting Started with Jetson Nano - https://www.youtube.com/watch?v=DVfmw3Ftu3I

E-content:

1. Tim Brown, "Design Thinking," Harvard Business Review, vol. 86, no. 6, pp. 84-92, Jun. 2008.

2. Rolf Faste, "Ambidextrous Thinking," in Innovations in Mechanical Engineering Curricula for the 1990s, American Society of Mechanical Engineers, 1994, pp. 1-6.

3. M. H. Hemanth Kumar, Ravi Pratap Singh, Nishu Sharma, and Pragya Singh, "IoT Based Smart Security System Using Arduino," Journal of Emerging Technologies and Innovative Research (JETIR), vol. 8, no. 8, Aug. 2021.

4. Basil and Eliza Sawant, S. D., "IoT-Based Traffic Light Control System Using Raspberry Pi," in Proceedings of ICECDS 2017, 2017. DOI: 10.1109/ICECDS.2017.8389604.

5. Espressif Systems, "ESP32 Series Datasheet," 2024.

6. Josef Průša, "Open-Source 3D Printers and the RepRap Project,"RepRap Magazine, 2018.

Topics relevant to the development of "SKILL":

System design for achieving Sustainable Development Goals.

Catalogue prepared by	Dr. Divyarani M S, Dr. M Manikandan
Recommended by the Board of Studies on	BoS Number:
Date of Approval by the AcademicCouncil	Academic Council Meeting No. dated on

Course Code: MAT1212	**Course Title:Mathematical foundations for Machine Learning and Data Analysis** **Type of Course:1] School Core**	**L-T- P- C**	**3**	**1**	**0**	**4**
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-requisites	NIL					

Course Description	This course provides a comprehensive foundation in advanced mathematical concepts essential for engineering and scientific applications. It covers integral calculus and special functions, linear transformations and matrix decompositions, vector spaces and inner product spaces, and probability theory with random variables and standard probability distributions. The course emphasizes mathematical modeling, analytical problem-solving, data analysis techniques, and engineering applications, enabling students to apply mathematical tools to solve real-world engineering problems.	
Course Objective	The course aims to develop a strong conceptual understanding of vector spaces, linear algebra, and vector calculus. It introduces mathematical techniques used in engineering analysis and modelling, enabling students to apply these concepts effectively in solving engineering problems. The course also enables students to apply probability distributions and statistical methods in practical situations involving uncertainty and data analysis. Further, it strengthens analytical and problem-solving skills through applications of differential equations and statistical inference. In addition, the course familiarizes students with computational tools for solving engineering mathematics problems, thereby enhancing their ability to address real-world engineering challenges.	
Course Outcomes	On successful completion of the course the students shall be able to: **CO1:** Apply integral calculus and special functions to solve engineering problems. (L3) **CO2:** Solve problems involving linear transformations and matrix techniques. (L3) **CO3:** Apply concepts of vector spaces, basis, dimension, and orthogonality in mathematical applications. (L3) **CO4:** Apply probability concepts and random variable distributions to solve problems involving uncertainty and stochastic phenomena (L3) **CO5:** Examine discrete and continuous random variables using probability distributions such as Binomial, Poisson, Uniform, Exponential, and Normal distributions. (L3) **CO6:**Utilize computational tools such as MATLAB, Python, or similar software for mathematical computation, data analysis, and engineering applications. (L3)	
Course Content:		
Module 1	**Integral Calculus and Special Functions**	**(8L+3T=11 Sessions)**
Definite and indefinite integrals, Evaluation of definite and improper integrals; Beta and Gamma functions and their properties; Surface areas, volumes, Sterlin approximation theorem.		
Module 2	**Vector Spaces and Inner Product Spaces**	**(12L+4T=16 Sessions)**
Vector Space, Examples, Properties, Subspaces, Linear Combinations, Spanning Set of a Vector Space, Linear Dependence and Independence, Basis and Dimension, Change of Basis; Normed Linear Spaces, Inner Products Spaces, Orthogonality, Orthogonal and Orthonormal Bases, Gram-Schmidt Orthogonalization Process, Orthogonal Complements, Orthogonal Projections. Applications to Difference Equations, Markov Chains.		
Module 3	**Linear Transformations and Matrix Decompositions**	**(10L+3T=13 Sessions)**
Linear Transformations, Algebra of Transformations, Singular and Nonsingular Linear Mappings,		

Isomorphisms, Matrix Representation of Linear Transformations, The Range and Null Space, Rank-Nullity Theorem.

Least Squares Approximation, Singular Value Decomposition: Singular Values, Computing Singular Value Decomposition, Data Compression Using Singular Value Decomposition and Introduction to Principal Component Analysis.

Module 4	**Probability and Random Variables for machine learning**	**(15L+5T =20 Sessions)**

Introduction to Probability, Conditional Probability, Bayes' Theorem and Problems. Random Variables (discrete and continuous), Probability Mass/Density Functions, Mathematical Expectations, discrete probability distributions - Binomial distribution, Poisson distribution, Continuous uniform distribution - Exponential distribution, Normal distribution.

Machine learning: large language models

Targeted Applications & Tools that can be used:

- Evaluation of definite and improper integrals for computing areas, volumes, surface areas, engineering design calculations, and scientific modeling using Beta and Gamma functions.
- Representation and analysis of high-dimensional vector spaces, feature extraction, orthogonal projections, signal processing, computer graphics, pattern recognition, and machine learning.
- Least-squares regression, image compression, recommendation systems, dimensionality reduction, Principal Component Analysis (PCA), data mining, and machine learning applications.
- Probabilistic modeling, Bayesian inference, uncertainty quantification, predictive analytics, classification algorithms, statistical learning, and foundations of large language models (LLMs).

Software: MATLAB, Mathematica, Maple, Python, GeoGebra, Octave, R, Scilab.

Text Book

1. E. Kreyszig, *Advanced Engineering Mathematics*, 10th ed. Hoboken, NJ, USA: John Wiley & Sons, 2015.
2. B. S. Grewal, *Higher Engineering Mathematics*, 45th ed. New Delhi, India: Khanna Publishers, 2017.sley Publishing Company.

Assignment:

Assignments will involve theoretical problem-solving and computational implementation using suitable programming tools. The assignments may be undertaken individually or in groups, as decided by the instructor.

References:

1. G. Strang, *Introduction to Linear Algebra*, 6th ed. Wellesley, MA, USA: Wellesley-Cambridge Press, 2023.
2. D. C. Lay, S. R. Lay, and J. J. McDonald, *Linear Algebra and Its Applications*, 6th ed. Harlow, U.K.: Pearson, 2021.
3. E. Kreyszig, *Advanced Engineering Mathematics*, 10th ed. Hoboken, NJ, USA: John Wiley & Sons, 2015.

<table>
<tr><td colspan="2">4. B. S. Grewal, Higher Engineering Mathematics, 45th ed. New Delhi, India: Khanna Publishers, 2017.
5. C. D. Meyer, Matrix Analysis and Applied Linear Algebra, 2nd ed. Philadelphia, PA, USA: SIAM, 2023.
E-resources/ Web links:https://open.umn.edu/opentextbooks/textbooks/24
1. https://open.umn.edu/opentextbooks/textbooks/introduction-to-probability
2. https://open.umn.edu/opentextbooks/textbooks/linear-transformations-on-vector-spaces</td></tr>
<tr><td colspan="2">Topics relevant to SKILL DEVELOPMENT: The course focuses on the concepts of calculus and differential equation with reference to specific engineering problems. The course is of both conceptual and analytical type in nature through Problem solving. This is attained through the assessment component mentioned in course handout.</td></tr>
<tr><td>Catalogue prepared by</td><td>Dr. Pavithra Sivaprakasam</td></tr>
<tr><td>Recommended by the Board of Studies on</td><td></td></tr>
<tr><td>Date of Approval by the Academic Council</td><td></td></tr>
</table>

<table>
<tr><td>Course Code:
CIV1200</td><td>Course Name: Foundations of Integrated Engineering
Type of Course: Theory Course</td><td>L- T-P- C</td><td>2-0-0-2</td></tr>
<tr><td>Version No.</td><td colspan="3">1.0</td></tr>
<tr><td>NCrF Level-</td><td colspan="3">5.0</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="3">NIL</td></tr>
<tr><td>Co-requisites</td><td colspan="3"></td></tr>
<tr><td>Anti-requisites</td><td colspan="3">CSEXXXX</td></tr>
<tr><td>Course Description</td><td colspan="3">This interdisciplinary course introduces first-year engineering students to foundational principles and practices across key engineering domains, emphasizing real-world problem-solving, sustainability, and ethical innovation. Students explore how civil, mechanical, electrical, and IT systems intersect with emerging technologies like IoT, AI, and geomatics to address global challenges. Through case studies, learners gain deeper understanding of smart infrastructure, prototyping mechanical/electronic systems, and securing IT solutions. Topics include bioinformatics for environmental monitoring, GIS-enabled urban planning, renewable energy integration, and cybersecurity fundamentals. The course cultivates a holistic understanding of engineering's role in sustainable development, safety, and ethical decision-making, preparing students to contribute meaningfully to multidisciplinary projects in a</td></tr>
</table>

<table>
<tr><td></td><td colspan="4">technology-driven world.</td></tr>
<tr><td>Course Objective</td><td colspan="4">The objective of the course is skill development of student by using Participative Learning techniques.</td></tr>
<tr><td>Course Outcomes</td><td colspan="4">On successful completion of the course the students shall be able to:
<table><tr><td>CO1: Recall key principles of Agile, DevOps, and bioinformatics used in interdisciplinary engineering contexts.</td><td>L1– Remember</td></tr><tr><td>CO2: Explain the role of GIS, LiDAR, and sustainable materials in designing smart infrastructure and disaster management systems.</td><td>L2– Understand</td></tr><tr><td>CO3: Describe core components of mechanical systems and their real-world applications.</td><td>L3 – Apply</td></tr><tr><td>CO4: Describe the functionality of IoT-enabled wearable devices, embedded systems, and renewable energy integration in smart grids.</td><td>L4 – Analyze</td></tr><tr><td>CO5: List foundational IT concepts such as cloud computing architectures, cybersecurity threats, and blockchain applications.</td><td>L5 – Evaluate</td></tr></table></td></tr>
<tr><td colspan="5">Course Content:</td></tr>
<tr><td>Module 1</td><td>Foundations of Engineering Practice</td><td>Assignment</td><td>Case studies</td><td>6 Sessions</td></tr>
<tr><td colspan="5">Real-world problem-solving using data logic and practical applications, Collaboration and Innovation through multi-domain project, Engineering Ethics & Environmental Impact

Emerging Fields: Automation, and Introduction to bioinformatics and its application

Sustainability & Safety: Circular economy principles, carbon footprint analysis.</td></tr>
<tr><td>Module 2</td><td>Civil Engineering & Geomatics</td><td>Assignment</td><td>Article Review</td><td>6 Sessions</td></tr>
<tr><td colspan="5">Smart Infrastructure & Geomatics: GIS mapping, LiDAR, drone surveys for urban planning, Geospatial data analysis for disaster management.

Sustainable Construction: 3D-printed structures, self-healing concrete, Digital twins for infrastructure monitoring.

Green Innovations: Net-zero energy buildings, rainwater harvesting systems.</td></tr>
<tr><td>Module 3</td><td>Mechanical Engineering in Action</td><td>Assignment & Quiz</td><td>Data Collection</td><td>6 Sessions</td></tr>
<tr><td colspan="5">Advanced Manufacturing: Collaborative robots (cobots), additive manufacturing and 3D printing, Reverse</td></tr>
</table>

engineering and prototyping.

Energy Systems: Solar/wind energy harvesting, piezoelectric applications.

Biomechanics: Prosthetics design, ergonomic product lifecycle.

Module 4	Electrical & Electronics Engineering	Assignment & Quiz	Data Collection and visualization	6 Sessions

Smart Devices & Systems: Embedded systems, Wearable technology, Edge computing and hardware platforms

Energy Innovations: EV charging infrastructure, wireless power transfer, Smart grid integration with renewables.

Module 5	Fundamentals of IT	Assignment & Quiz	Case studies	6 Sessions

Core IT Topics: Networking basics, Cloud computing

Cybersecurity & Data: Encryption, phishing prevention, zero-trust models, Database management.

Emerging Tech: Blockchain for supply chains, AI/ML basics, IoT integration with cloud platforms

Targeted Application & Tools that can be used:

Application Areas include Interdisciplinary problem-solving, Smart city planning, disaster management, Robotics prototyping, renewable energy systems, Wearable health tech, smart grids, Secure cloud systems.

Tools: 3D Printers, Autocad, Tinkercad, ArcGIS / QGIS, Arduino/Raspberry Pi

Text Book:

1. William Oakes & Les Leone, "Engineering Your Future: An Introduction to Engineering", Oxford University Press, 9th Edition, 2021
2. Barry F. Kavanagh, “Introduction to Geomatics”, Pearson, 5th Edition, 2021
3. Ian Gibson, David Rosen, & Brent Stucker, “Additive Manufacturing Technologies”, Springer, 3rd Edition, 2021
4. Sudip Misra, "The Internet of Things: Enabling Technologies, Protocols, and Use Cases", Wiley, 2nd Edition, 2022
5. James Kurose & Keith Ross, “Computer Networking: A Top-Down Approach”, Pearson, 8th Edition, 2020

References

1. Supratim Choudhuri, “Bioinformatics for Beginners: Genes, Genomes, and Molecular Evolution", Academic Press, 1st Edition, 2023,
2. Robert McGinn, "The Ethical Engineer: Contemporary Concepts and Cases", Princeton University Press, 1st Edition, 2020
3. Charles J. Kibert, “Sustainable Construction: Green Building Design and Delivery”, Wiley, 5th Edition, 2022
4. Anthony M. Townsend, "Smart Cities: Big Data, Civic Hackers, and the Quest for a New Utopia", W.W. Norton & Company, 1st Edition, 2020
5. David Buchla, "Renewable Energy Systems: A Smart Energy Systems Approach", Pearson, 2nd Edition, 2023
6. Charles Platt, "Make: Electronics: Learning Through Discovery", Make Community, 3rd Edition, 2021
7. Charles J. Brooks, Christopher Grow, & Philip Craig, “Cybersecurity Essentials", Wiley, 2nd Edition,

2021

Web-resources:

1. Post-parametric Automation in Design and Construction

https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1155197&site=ehost-live

2. Smart Cities : Introducing Digital Innovation to Cities

https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1993146&site=ehost-live

3. Innovation Energy: Trends and Perspectives or Challenges of Energy Innovation

https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=2323766&site=ehost-live

4. Additive Manufacturing: Opportunities, Challenges, Implications

https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1134464&site=ehost-live

Course Code: CHE2501	Course Name: Chemistry of Smart Materials Type of Course: Theory Course	L- T-P- C	3-0-0-3
Version No.	1.0		
NCrF Level-	5.0		
Course Pre-requisites	NIL		
Co-requisites	CHE2501		
Anti-requisites	CSEXXXX		
Course Description	Theobjective of the course is to introduce the students to concepts and applications of chemistry of smart materials. The course also aims to enhance the knowledge of smart materials associated with memory system, display devices, , sensors, energy devices and environment. It will also cultivate an ability to identify chemistry in each of smart engineered materialsand interpret solutions for the challenges connected to memory, display, energy, smart, green and sustainable technologies. It targets to strengthen the fundamental concepts behind chemistry of smart materials and then builds an interface with their industrial applications. **This course is designed to cater to Environment and Sustainability**		
Course Objective	The objective of the course is 'SKILL DEVELOPMENT' of the student by using Participative learning techniques.		

Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:**Relate the knowledge of chemistry to computational approaches to identify materials functionalities and properties **CO2**:Recognizeand interpret solutions for the challenges connected to memory, display, smart, green and sustainable technologies. **CO3:**Explain the quality parameters of engineering materials associated with environment and sensors. **CO4:**Interpret the knowledge of sustainable chemistry for E- waste management. **CO5:**Analyse the importance of various electrochemical sources in energy systems.	**L1**– Remembe **L2**– Understand **L3** – Apply **L4** – Analyze **L5** – Evaluate

Course Content:

Module 1	Computational Chemistry	Assignment	Data Collection and analysis	**09 Sessions**

Topics: Fundamental particles of atom – their mass, charge and location – atomic number and mass number, Stabilizing interactions: Bonded and non-bonded interactions. Chemistry of weak interactions – van der Waals force and hydrogen bonding, Density functional theory. 3D co-ordinate generation for small molecules, geometry optimization by Molview. Chemical Databases: Chemoinformatics, MSDS

Self- learning topics: Scope, cost and efficiency of computational modeling.

Module 2	Materials for Memory and Display Systems	Assignment	Data Collection and analysis	**09 Sessions**

Topics: MemorySystems : Introduction, classification of electronic memory devices- Transistor, capacitor, charge -transfer and Resistor, types of materials - organic, polymeric and hybrid materials, and applications, manufacturing of semiconductor chips.

Display Systems: photo and electroactivematerials , materials for display -Principle, Properties and applications: Liquid crystals for LCD-Liquid crystals display, Basics of LED: OLED-organic light emitting diode and light emitting electrochemical cells.

Self- learning topics: Green computing: Biocomposite based memory devices

Module 3	Nanomaterials based Smart Sensors and Devices	Assignment	Data Collection and analysis	**09 Sessions**

Topics: Nanomaterials- Introduction, classification based on dimensionality,quantum confinement. Size dependent properties,Synthesis, Properties of CNT and Graphene and their application as **Materials for data analysis and packaging** -RFID and IONT.

Sensors: Introduction, types, Principle and applications- electrochemical sensor: nanomaterials for sensing applications - Glucose, VOC sensing.

Self-learning topics: Fullerene, biomolecules in sensing, Strain sensors

Module 4	Sustainable Materials and	Quiz/Seminar	Data Collection and analysis	**09 Session**

	Development			**s**

Topics: E waste: Introduction, E waste Hazards, E- waste management,Recovery of precious metal- Cu by Hydrometallurgy.

Green Chemistry: Fundamentals and 12 principles with examples, Carbon footprint and sequestration

Sustainable Chemistry: -Introduction to Biomaterials- PLA , polymers in bio-compatible andbio-degradable materials - Poly(3-hydroxybutyrate-co-3-hydroxyvalerate) PHBV, synthesis and applications in drug delivery.

Self-learning topics:circular economy- case studies.

Module 5	Energy Science	Quiz/Seminar	Data Collection and analysis	**09 Sessions**

Topics:Battery technology: Fundamentalsof electrochemistry, Introduction to electrochemical storage devices: battery (Lithium-ion battery- $LiMnO_2$, $LiCoO_2$, metal air batteries- LiO_2) and supercapacitors- Introduction, Principle, Types - EDLC, pseudo and asymmetric capacitor.

Photovoltaics: Solar cells - Construction and working principle; types- Inorganic, Organic and quantum dot sensitized (QDSSC's).

Self -learning topics: Battery technology for e-mobility, Green hydrogen

Targeted Application & Tools that can be used:

Application areas are Data storage and analysis, logistics, Biomedicine, Energy, Environment and sustainability

Tools: Molview, chemdraw, excel etc

Project work/Assignment:

Assessment Type

- Midterm exam
- Assignment (review of digital/ e-resource from PU link given in references section - mandatory to submit screenshot accessing the digital resource.)
- Quiz/Student Seminar
- End Term Exam
- Self-learning

Text Book

1. Wiley, "Engineering Chemistry", Wiley.
2. G.A. Ozin and A.C. Arsenault, Nanochemistry: A chemical approach to nanomaterials, Royal Society of Chemistry, 2009

Reference Books

1. Functional and smart materials, Chander Prakash, Sunpreet Singh, J. Paulo Davim, 2020, CRC Press,
2. ISBN: 978-036-727-510-5.
3. E-waste recycling and management: present scenarios and environmental issues, Khan, Anish, and

4. Abdullah M. Asiri. 2019, Springer, Vol. 33. ISBN: 978-3-030-14186-8.
5. Essentials of computational chemistry: theories and models, Christopher J Cramer, 2013, John Wiley &
6. Sons. ISBN: 978-0-470-09182-1.
7. Energy storage and conversion devices: Supercapacitors, batteries and hydroelectric cells, Anurag Gaur,
8. L. Sharma, Anil Arya. 2021, CRC press, 1st edition, ISBN: 978-1-003-14176-1.
9. Fundamentals of analytical chemistry: An introduction, Douglas A. Skoogetetal., 2004 Thomson Asia pte
10. Ltd., 8th, ISBN: 978-0-495-55828-6
11. Functional and smart materials, Chander Prakash, Sunpreet Singh, J. Paulo Davim, 2020, CRC Press,
12. ISBN: 978-036-727-510-5.
13. Electrical and electronic devices, circuits and materials: Technological challenges and solutions. Tripathi,
14. L., Alvi, P. A., & Subramaniam, U, 2021, John Wiley & Sons, ISBN: 978-0367564261.
15. F. Jensen, Introduction to Computational Chemistry, 3rd edition, Wiley, 2017.

E resources

1. https://presiuniv.knimbus.com/user#/searchresult?searchId=computational%20chemistry&_t=1738054970142
2. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_48504
3. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_147967
4. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_130301
5. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_87297
6. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_67006
7. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_137261
8. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_86712

Skill Sets

All topics in theory component are relevant to Environment and Sustainability.

Course Code: CHE2502	Course Name: Chemistry of Smart Materials Lab Type of Course: Laboratory Course	L- T-P- C	0-0-2-1
Version No.	1.0		
NCrF Level-	5.0		
Course Pre-requisites	NIL		
Co-requisites			

Anti-requisites	
Course Description	The laboratory course aims to develop experimental skills and apply fundamental chemical principles to address chemistry-related problems in engineering. The experiments are carefully designed to complement the theoretical concepts covered in lectures, providing hands-on experience to deepen understanding and reinforce learning. This course is designed to cater to Environment and Sustainability.
Course Objective	The objective of the course is to familiarize the learners with the concepts of "Chemistry of Smart Materials Lab" and attain SKILL DEVELOPMENT through EXPERIENTIAL LEARNING techniques.
Course Outcomes	On successful completion of the course, students shall be able to: CO1: recognize the basic techniques and instrumentation used in chemistry laboratories for quantitative analysis. CO2: estimate the presence of acids and metal ions in domestic and industrial waste using laboratory techniques. CO3: review the experimental results and demonstrate improved experimental skills through hands-on laboratory experience. CO4: classify laboratory techniques such as experimental setups for synthesis, purification, recovery and analysis.

Course Content:

Experiment 1	Experimental	Data Collection	Analysis and Interpretation
Determination of strength of strong acid in battery electrolyte using conductometric sensors.			
Experiment 2	Experimental	Data Collection	Analysis and Interpretation
Estimation of iron from e-waste using Electrochemical sensors.			
Experiment 3	Experimental	Data Collection	Analysis and Interpretation
Determination of pKa of organic acid of battery electrolyte using pH sensor.			
Experiment 4	Experimental	Data Collection	Analysis and Interpretation
Estimation of copper from PCBs by using colorimeter (Optical Sensor).			

<table>
<tr><td>Course Code:
CSS2200</td><td colspan="2">Course Title: Program Solving Using C
Type of Course: Theory Course</td><td>L- T-P-C</td><td>2-0-0-2</td></tr>
<tr><td>Version No.</td><td colspan="4">1.0</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="4">NIL</td></tr>
<tr><td>Anti-requisites</td><td colspan="4">NIL</td></tr>
<tr><td>Course Description</td><td colspan="4">The course is designed to provide a strong foundation in the C programming language and problem-solving techniques. It enables students to develop logical thinking and programming skills required to design and implement applications using C. By understanding fundamental programming concepts such as algorithms, control structures, arrays, functions, pointers, and data handling, students will be prepared to learn and adapt to other programming languages and advanced computing technologies in the future.</td></tr>
<tr><td>Course Objective</td><td colspan="4">The objective of the course is to familiarize the learners with the concepts of Problem Solving Using C and attain Employability through Problem Solving Methodologies.</td></tr>
<tr><td>Course Outcomes</td><td colspan="4">On successful completion of this course the students shall be able to:
CO1: Explain problem-solving techniques using algorithms, pseudo code, and flowcharts for developing programs [Level 2 – Understand]
CO2: Apply fundamental C programming constructs such as data types, operators, decision-making, and looping statements to develop simple programs. [Level 3 – Apply]
CO3: Develop programs using arrays and strings to solve computational problems effectively [Level 3 – Apply]
CO4: Implement modular and reusable programs using functions, recursion, and pointers. [Level 3 – Apply]
CO5: Design applications using structures, unions, file handling concepts, and demonstrate the role of C programming in emerging technologies. [Level 3 – Apply]</td></tr>
<tr><td>Course Content:</td><td colspan="4"></td></tr>
<tr><td>Module 1</td><td>Introduction to C Language</td><td>Assignment (Algorithms, Flowcharts, and Basic C Programs)</td><td colspan="2">6 Sessions</td></tr>
<tr><td colspan="5">Topics:
Introduction to Programming – Algorithms – Pseudo Code - Flow Chart – Compilation – Execution – Preprocessor Directives (#define, #include, #undef) - Overview of C – Constants, Variables and Data types – Operators and Expressions – Managing Input and Output Operations – Decision Making and Branching - Decision Making and Looping.</td></tr>
<tr><td>Module 2</td><td>Introduction to Arrays and Strings</td><td>Programming Assignment (Arrays, Sorting, Searching, and String Handling in C)</td><td colspan="2">6 Sessions</td></tr>
<tr><td colspan="5">Topics:
Arrays: Introduction – One Dimensional Array – Initialization of One Dimensional Arrays – Example Programs – Sorting (Bubble Sort, Selection Sort) – Searching (Linear Search) - Two Dimensional Arrays – Initialization of Two Dimensional Arrays. Example Programs – Matrix operations. Strings: Introduction – Declaring and Initializing String
Variables – Reading Strings from Terminal – Writing String to Screen – String Handling Functions.</td></tr>
</table>

Module 3	**Functions and Pointers**	**Programming Assignment (Functions, Recursion, and Pointer Implementation in C)**	**6 Sessions**
Topics: **Functions:** Introduction – Need for User-defined functions – Elements of User-Defined Functions: declaration, definition and function call–Categories of Functions – Recursion. **Pointers:** Introduction – Declaring Pointer Variables – Initialization of Variables – Pointer Operators – Pointer Arithmetic – Arrays and Pointers – Parameter Passing: Pass by Value, Pass by Reference.			
Module 4	**Structures and Union and File Handling**	**Assignment (Structures, Unions, and File Operations in C)**	**6 Sessions**
Topics: **Structures:** Introduction – Defining a Structure – Declaring Structure Variable – Accessing Structure Members – **Union:** Introduction – Defining and Declaring Union – Difference Between Union and Structure.- **File handling:** Defining and Opening a File – Closing a File – Input / Output Operations on File.			
Module 5	**Application of C Programming in Emerging CSE Specializations**	**Mini Project (C Programming Applications in Cyber Security, Blockchain, and AI)**	**6 Sessions**
Topics: Applications of C Programming in modern technologies – Cyber Security: Simple password checking programs and basic security concepts – Blockchain: Secure data storage and transaction concepts – Artificial Intelligence and Machine Learning: Basic data processing programs.			
Text Book(s): 1. E. Balaguruswamy, "Programming in ANSI C", 8th Edition, 2019, McGraw Hill Education, ISBN: 978-93-5316- 513-0.			
Reference Book(s): 1. Yashwant Kanetkar, Let us C, 17th Edition, BPB Publications, 2020. 2. Reema Thareja, "Programming in C", Oxford University Press, Second Edition, 2016. 3. Kernighan, B.W and Ritchie, D.M, "The C Programming language", Second Edition, Pearson Education, 2015 4. Schildt Herbert, "C: The Complete Reference", Tata McGraw Hill Education, 4th Edition, 2014. 5. Stephen G. Kochan, "Programming in C", Addison-Wesley Professional, 4th Edition, 2014.			
Web Links and Video Lectures: 1. NPTEL – Programming in C 2. NPTEL – Problem Solving Through Programming in C 3. GeeksforGeeks – C Programming Language			

Course Code: **ENG2030**	**Course Title:** **Communication in Practice** **Type of Course:** **Integrated**	**L-T-P-C**	1	0	2	2
Version No.	1.0					
Course Pre-requisites	ENG XXXX – Fundamentals of Communication (Semester I)					
Anti-requisites	NIL					
Course Description	*Communication in Practice* is the second-semester course in sequence designed for first-year undergraduate students in engineering and computer science programmes. This course puts those foundations under professional pressure. The question is no longer *what* communication is, but *how* to deploy it when it counts: in a job interview, a pitch to a client, a negotiation with a colleague, a professional report directed at a real reader, or a leadership communication challenge that affects a team. The course is organised around four modules, each of which explicitly upgrades the work done in semester one. Assessment is 100% Continuous Assessment. There is no end-semester examination. The centrepiece is a capstone Professional Communication Portfolio that the student could realistically share with an employer.						
Course Objective	The course aims to articulate the foundational tenets of professional communicative identity, discourse ethics, and analytical text evaluation and apply them to high-stakes institutional scenarios and complex professional ecosystems. It also focuses on cultivating advanced oral communication strategies and persuasive vocal mechanics, including structured rhetorical delivery and spontaneous adaptive responses, for collaborative, hierarchical, and leadership contexts. Furthermore, it aims to refine technical authorship capabilities and artificial intelligence-mediated workflows to produce precise, highly structured, and audience-centric professional artifacts across formal engineering, computational, and cross-functional corporate domains.						
Course Outcomes	**On successful completion of this course the students shall be able to:** 1. **Explain** the principles governing professional communication identity, communicator ethics, and context-sensitive self-presentation across workplace, hierarchical, and digital contexts. 2. **Demonstrate** advanced vocal, rhetorical, and adaptive strategies across high-stakes spoken communication genres, including interviews, pitches, negotiations, and leadership scenarios. 3. **Demonstrate** advanced vocal, rhetorical, and adaptive strategies across high-stakes spoken communication genres, including interviews, pitches, negotiations, and leadership scenarios. 4. **Execute** the production of professional-grade written artifacts across high-						

	demand genres—including formal reports, business proposals, data communications, and advocacy briefs—meeting real-world standards of clarity, structure, and audience appropriateness. 5. **Implement** AI-assisted communication workflows in professional contexts, including systematic prompt design for high-stakes outputs, validation protocols for synthetic materials, and the application of ethical frameworks at individual and organizational levels.

Course Content: Communication in Practice

Module 1	**Words that Work: Vocabulary, Register, and Precision**	**CA1**	**Register Switching Activity**	**09 Sessions**

The topics will be explored through vocabulary games, role-play activities, communication challenges, case-based tasks, and workplace simulations:

- Why Words Matter in Professional and Technical Communication
- Choosing the Right Words for Different Situations
- Understanding Meaning: Positive, Negative, and Neutral Word Choices
- Commonly Confused Words and Miscommunication in Professional Contexts
- Using Clear, Precise, and Concise Language
- Explaining Technical Terms to Non-Technical Audiences
- Building and Using Personal/Professional Vocabulary

Module 2	**Listening, Processing, and Structured Response Writing**	**CA2**	**Rapid Listening and Response**	**12 Sessions**

The topics will be explored through podcasts, interviews, workplace recordings, collaborative listening tasks, discussions, and real-time writing activities:

- Identifying Key Information from Spoken Inputs
- Organizing Information from Discussions and Conversations
- Converting Spoken Communication into Clear and Structured Written Responses
- Writing Summaries from Audio and Discussion Inputs
- Writing Follow-Up Responses Based on Spoken Communication
- Writing Action Notes, Instructions, and Task Updates from Listening Inputs
- Converting Meetings and Discussions into Structured Written Outputs

Module 3	**High-Stakes Spoken Communication**	**CA**	**Mock Professional Interview, Pitching an Idea Pitch**	**12 Sessions**

- Speaking as a professional: identity, presence, and the first impression
- Vocal authority, physical presence, and communication across hierarchies
- Interview communication: structure, composure, and the STAR framework

- Pitching ideas: the 3-minute and 10-minute pitch architecture
- Negotiation and principled disagreement
- Communicating in leadership positions: chairing, facilitating, and giving direction
- Managing communication under pressure: hostile questions, silence, and contradiction

Module 4	Adaptive Communication	CA	Communication Strategy Plan	12 Sessions

The topics will be explored through leadership simulations, AI-assisted scenario practice, ethical debate exercises, case analysis, peer feedback sessions, and reflective portfolio activities:

- Communication Strategy: Audiences, Channels, Timing, and Feedback Loops
- What Emotional Intelligence Actually Does in High-Stakes Conversations
- Reading the Room: Adapting Style Across Personalities, Contexts, and Power Gaps
- Giving and Receiving Developmental Feedback as a Professional Practice
- Navigating Difficult Conversations: Disagreement, Criticism, and Bad News
- Repairing Communication Breakdowns: Apology, Accountability, and Recovery
- AI as a Strategic Communication Tool: Capabilities, Limits, and What Professionals Must Decide

Targeted Application & Tools that can be used: Otter.ai, Microsoft Word/PowerPoint/Teams/Clipchamp/Whiteboard/OneNote/Forms/Copilot, Miro, Padlet, Canva, Mentimeter, AllSides Media Bias Ratings, Ad Fontes Media Bias Chart, Snopes/factcheck.rog, Zotero, NotebookLM, Canva, Adobe Express, CapCut, OBS Studio, Claude, ChatGPT, Gemini, DeepSeek, Google Docs

Project Work / Assignment:

1. Register Switching Activity (Individual): Students receive a single professional scenario and must communicate the same core message to three different audiences — a peer, a manager, and a non-specialist outsider — switching register, vocabulary, and tone appropriately for each. Delivered live in class. Graded on accuracy of register shift, vocabulary choices, and adaptability under in-the-moment feedback from the assessor.

2. Word in the Wild Log (Individual): Over four weeks, students maintain a running log of 25 words or phrases encountered in professional contexts — podcasts, workplace conversations, news, interviews — that they did not fully understand or could not have used confidently themselves. For each entry, students record the source, the meaning in context, an alternative they might have used, and one original sentence demonstrating precise use. Submitted as a structured document at the end of Module 1. Graded on range, contextual accuracy, and quality of reflection.

3. Rapid Listening and Response (Individual): Students listen to an unedited workplace audio recording — a meeting excerpt, a briefing, or a professional conversation — played once, without transcripts. Under timed conditions, they produce a structured written response that captures key information, identifies the communication purpose, and drafts an appropriate follow-up action note. Graded on accuracy, organisation, and quality of written response.

4. Mock Professional Interview (Individual): Each student undergoes a 10-minute mock professional interview conducted by faculty or a peer panel. No prepared scripts permitted. Graded on composure,

structure, vocal authority, and use of the STAR framework. Written self-evaluation of 200 words submitted within 48 hours.

5. Idea Pitch (Individual): Students deliver a 3-minute pitch of a solution to a real or simulated professional problem, to a panel that includes at least one hostile or sceptical questioner. Immediately after the pitch, the panel asks two minutes of unscripted questions. Graded on clarity of argument, persuasive structure, handling of pressure, and vocal presence.

6. Difficult Conversation Simulation (Pair): Pairs of students are assigned roles in a scripted conflict scenario — a performance issue, a missed deadline, a broken professional agreement. Each student must navigate the conversation toward a constructive resolution without a script. Immediately after, both students complete a structured individual debrief identifying what communication choices they made, what they would change, and what they observed in their partner. Graded individually on communication behaviour during the simulation and quality of the debrief.

7. Communication Strategy Plan — Presentation and Portfolio Submission (Group): Students select a real or realistic professional scenario — a team conflict, a stakeholder disagreement, a leadership announcement, or a crisis communication moment. They develop a communication strategy plan that maps the audience, chooses appropriate channels and timing, anticipates likely responses, and identifies where AI tools could assist and where they must not. The plan is delivered as a 5-minute structured presentation to a simulated panel, followed by two minutes of unscripted Q&A. The portfolio submission includes the written plan, a one-page reflection on their communication growth across all four modules, and one piece of evidence from each module that demonstrates that growth. Graded on strategic thinking, clarity of presentation, quality of reflection, and coherence of the portfolio as a whole.

Text Book(s):

T1. Treasure, J. (2017). *How to be Heard: Secrets for Powerful Speaking and Listening.* Mango Publishing.
T2. Beebe, S. A., Beebe, S. J., & Ivy, D. K. (2019). *Communication: Principles for a Lifetime* (7th ed.). Pearson.

Reference Book(s):

R1. Goleman, D. (1995). *Emotional Intelligence: Why It Can Matter More Than IQ.* Bantam Books.

R2. Patterson, K., Grenny, J., McMillan, R., & Switzler, A. (2011). *Crucial Conversations: Tools for Talking When Stakes Are High* (2nd ed.). McGraw-Hill.

R3. Cuddy, A. (2015). *Presence: Bringing Your Boldest Self to Your Biggest Challenges.* Little, Brown and Company.

R4. Fisher, R., Ury, W., & Patton, B. (2011). *Getting to Yes: Negotiating Agreement Without Giving In* (3rd ed.). Penguin Books.

R5. Stone, D., Patton, B., & Heen, S. (2010). *Difficult Conversations: How to Discuss What Matters Most* (2nd ed.). Penguin Books.

Web Sources:

- ted.com/topics/communication
- ted.com/topics/leadership
- hbr.org (Harvard Business Review — communication and leadership articles)
- psychologytoday.com/us/basics/emotional-intelligence
- learnprompting.org
- unesco.org/en/artificial-intelligence/ethics
- yoodli.ai (AI speech coaching)
- themuse.com (professional communication and career resources)
- mindtools.com (communication skills and workplace tools)

Topics relevant to "SKILL DEVELOPMENT": Vocabulary building and register awareness; precise and concise language use; explaining technical terms to non-specialist audiences; active listening and information processing; converting spoken inputs into structured written responses; writing action notes and follow-up communications; professional presence and vocal authority; interview communication and the STAR framework; idea pitching and persuasive speaking; negotiation and principled disagreement; chairing and facilitating discussions; managing communication under pressure; communication strategy and audience mapping; emotional intelligence in professional contexts; adapting communication style across personalities and power gaps; giving and receiving developmental feedback; navigating difficult conversations; repairing communication breakdowns; AI as a strategic communication tool; ethics of AI in professional communication.

Catalogue prepared by	Dr Ajith K and Dr ShahanasPunnilathShanavas
Recommended by the Board of Studies on	
Date of Approval by the Academic Council	

Course Code: EEE1200	Course Name: Basics of Electrical and Electronics Engineering. Type of Course: Theory Course	L- T-P- C	3-0-0-3
Version No.	2.0		
Course Pre-requisites	NIL		
Co-requisites	EEE1251		
Anti-requisites	NIL		
Course Description	This is a fundamental Course which is designed to know the use of basics of electrical and electronics engineering principles occurs in various fields of Engineering. The course emphasises on the characteristics and applications of electrical and electronic devices. The course also emphasizes on the working, analysis and design of electrical circuits using both active & passive components. Additionally, this course creates a foundation for the future courses such as Electrical machines, power system, power electronics Linear Integrated Circuits, Analog Communication and Digital Communication etc.		
Course Objective	The objective of the course is to familiarize the learners with the concepts of Basics of Electrical and Electronics Engineering and attain Skill Development through Participative Learning techniques		
Course Outcomes	**On successful completion of the course the students shall be able to:** CO1: Apply basic laws of Electrical Engineering to compute voltage, currents and other parameters in the circuits. [Level 3 - Apply] CO2: Discuss various fundamental parameters appearing in the characteristics of semiconductor devices and their applications. [Level2– Understand] CO3: Summarize the operations of different biasing configurations of BJTs and amplifiers. [Level2– Understand] CO4 :Discuss the performance characteristics and applications of various electrical Machines. [Level2– Understand]] CO5: Apply basic laws of Electrical Engineering to compute voltage, currents and other parameters in the circuits. [Level 3 – Apply]		
Course Content:			
Module 1	Introduction to Electrical Circuits	Assignment/ Quiz	10 Sessions
Topics: DC Circuits: Concept of Circuit and Network, Types of elements, Network Reduction Techniques- Series and parallel connections of resistive networks, Star–to-Delta Transformations, Mesh Analysis, Nodal Analysis, Numerical examples. AC Circuits: Fundamentals of single phase circuits - Series RL, RC and R-L-C Circuits, Concept of active power,			

reactive power and Power factor, Numerical examples.

Introduction to three phase system and relation between line and phase values in Star & Delta connection, Numerical examples.

Module 2	Semiconductor and Diode applications	Assessment : Quiz + Assignment	10 Sessions

Topics:

Mass Action Law, Charge densities in a semiconductor, Types of SC, Junction diodes -Ideal and practical behaviour, Modelling the Diode Forward Characteristic, and Diode applications like rectifiers, Clipping and clamping circuits. Zener diode, characteristics and its applications like voltage regulator.

Module 3	Transistors and its Applications	Assessment : Assignment/ Quiz	10 Sessions

Topics:

Transistor characteristics, Current components, BJT Configurations (CB, CC, CE configurations) and their current gains. Operating point, Biasing & stabilization techniques: Fixed Bias, Voltage divider bias and its stability factor and load line analysis. Single and multistage amplifier, Darlington pair.

JFET (Construction, principal of Operation and Volt –Ampere characteristics). Pinch- off voltage, Comparison of BJT and FET. MOSFET (Construction, principal of Operation and symbol), MOSFET characteristics in Enhancement and Depletion modes.

Module 4	Fundamentals of Electrical Machines	Assessment : Assignment/ Quiz	10 Sessions

Topics:

Electrical Machines: Single phase transformers: principle of operation and EMF equation, Numerical examples. DC Motor: principle of operation, Back EMF, torque equation, Numerical examples. AC Motor: Principle operation of Induction Motors and its Applications.

Special Machines: Introduction to special electrical machines and its applications.

Targeted Applications & Tools:

Multisim	An electronic circuit simulation software used for designing, testing, and analyzinganalog and digital circuits.
PSpice	A simulation tool used to analyze the behavior and performance of electronic circuits.
Multimeter	An electronic measuring instrument used to measure voltage, current, and resistance in circuits.
Function Generator	A device used to generate different electrical waveforms for circuit testing and analysis.
Power Supplies	Equipment used to provide regulated electrical power to electronic circuits and devices.

Skill-Building Activities through Experiential Learning:

- Hands-on experiments on electrical circuits and electronic components.
- Measurement and analysis of voltage, current, resistance, and waveform characteristics using electrical instruments.
- Simulation and testing of electronic circuits using software tools such as Multisim and PSpice.
- Practical study of semiconductor devices, BJTs, amplifiers, and electrical machines.
- Team-based troubleshooting and circuit debugging activities to enhance problem-solving skills.

Text Book

T1: Kothari D. P. & Nagrath I. J., "Basic Electrical and Electronics Engineering", Tata McGraw-Hill Education
T2: Theraja B.L. and Theraja A.K., "A Textbook of Electrical Technology: Basic Electrical Engineering" in S.I. System of Units, 23rd ed., New Delhi: S. Chand, 2002.
T3: J. Millman, C. C. Halkias and C. D. Parikh, "Millman's Integrated Electronics", McGraw Hill Education, 2nd Edition.
Basics of Electrical & Electronics Laboratory Manual.

R1: Re John Hiley, Keith Brown and Ian McKenzie Smith, "HUGHES Electrical and Electronic Technology", 10th Edition (Indian Edition published by Dorling Kindersley), Pearson,2011
R2: Samarajit Ghosh, "Fundamentals of Electrical and Electronics Engineering", 2nd Edition, Prentice Hall India, 2007.
R3: K Uma Rao, A Jaya Lakshmi, "Basic Electrical engineering" IK International publishing house Pvt. Ltd
R4:R. L. Boylestad and L. Nashelsky, "Electronic Devices and Circuit Theory", Pearson Education India 7th Edition.

E-Resources/Web Resources

1. https://www.digimat.in/nptel/courses/video/108105112/L01
2. https://nptel.ac.in/courses/108/105/108105153/
3. https://nptel.ac.in/courses/108/102/108102095

Course Code: CSS2201	**Course Title: Program Solving Using C Lab** **Type of Course: Laboratory Course**	**L- T-P-C**	**0-0-4-2**
Version No.	1.0		
Course Pre-requisites	**NIL**		
Anti-requisites	**NIL**		
Course Description	The course is designed to provide a strong foundation in the C programming language and problem-solving techniques. It enables students to develop logical thinking and programming skills required to design and implement applications using C. By understanding fundamental programming concepts such as algorithms, control structures, arrays, functions, pointers, and data handling, students will be prepared to learn and adapt to other programming languages and advanced computing technologies in the future.		
Course Objective	The objective of the course is to familiarize the learners with the concepts of Problem Solving Using C and attain Employability through Problem Solving Methodologies.		

Course Outcomes	**On successful completion of this course the students shall be able to:** **CO1:** Explain algorithms, flowcharts, and basic programming concepts used in C programming. [Level 2 – Understand] **CO2:** Implement C programs using decision-making, looping statements, arrays, and strings to solve computational problems. [Level 3 – Apply] **CO3:** Construct modular programs using functions, recursion, and pointers for effective problem solving.. [Level 3 – Apply] **CO4:** Develop applications using structures, unions, and file handling concepts in C programming. [Level 3 – Apply] **CO5:** Design and implement simple real-world applications in areas such as Cyber Security and Artificial Intelligence using C programming concepts. [Level 3 – Apply]
Course Content:	

List of Practicals: **60 Sessions**

Lab Sheet 1: 10 Sessions

Program 1: Sum of Two Numbers
Program 2: Find the Greatest of Three Numbers
Program 3: Check Even or Odd using Conditional Operator
Program 4: Print Multiplication Table using Loop
Program 5: Count Digits in a Number using While Loop
Program 6: Demonstration of Preprocessor Directives
Program 7: Simple Calculator using Switch Case

Lab Sheet 2: 10 Sessions
Program 1: Check Whether a Number is Positive, Negative or Zero
Program 2: Find the Sum of First N Natural Numbers
Program 3: Check Whether a Number is Prime or Not
Program 4: Find Factorial of a Number
Program 5: Reverse a Number
Program 6: Simple Number Guessing Game

Lab Sheet 3: 10 Sessions
Program 1: Linear Search in a One-Dimensional Array
Program 2: Bubble Sort on an Integer Array
Program 3: Matrix Addition (2D Arrays)
Program 4: Count Vowels in a String
Program 6: Selection Sort on an Array

Lab Sheet 4: 10 Sessions
Program 1: Sum of Two Numbers Using User-Defined Function
Program 2: Factorial Using Recursion
Program 3: Swap Two Numbers Using Call by Value (No Swap)
Program 4: Swap Two Numbers Using Call by Reference (With Swap)
Program 5: Pointer Basics - Access and Modify Variable via Pointer
Program 6: Accessing Array Elements Using Pointers

Lab Sheet 5: 10 Sessions

Program 1: Basic Structure Usage
Program 2: Structure with Nested Structures (Date of Birth)
Program 3: Union Example and Member Access
Program 4: Write to a File (Text Mode)
Program 5: Read from a File (Text Mode)
Program 6: Append Data to a File

Lab Sheet 6: 10 Sessions
Program 1:Write a C program to check whether a password is strong or weak based on its length. If the password contains more than 8 characters, display "Strong Password"; otherwise display "Weak Password".
Program 2: Develop a simple login system using C. Store a predefined username and password in the program and verify the user input. Display "Login Successful" or "Invalid Credentials".
Program 3: Write a C program to input marks of 5 students and calculate the average marks. Display whether the class performance is "Good" or "Needs Improvement".
Program 4: Develop a C program to find the largest number in an array. This simulates basic data analysis used in AI systems.

Text Book(s):

1. E. Balaguruswamy, "Programming in ANSI C", 8th Edition, 2019, McGraw Hill Education, ISBN: 978-93-5316- 513-0.

Reference Book(s):

1. Yashwant Kanetkar, Let us C, 17th Edition, BPB Publications, 2020.
2. ReemaThareja, "Programming in C", Oxford University Press, Second Edition, 2016.
3. Kernighan, B.W and Ritchie,D.M, "The C Programming language", Second Edition, Pearson Education, 2015
4. Schildt Herbert, "C: The Complete Reference", Tata McGraw Hill Education, 4th Edition, 2014.
5. Stephen G. Kochan, "Programming in C", Addison-Wesley Professional, 4th Edition, 2014.

Web Links and Video Lectures:

1. NPTEL – Programming in C
2. NPTEL – Problem Solving Through Programming in C
3. GeeksforGeeks – C Programming Language
4. Programiz – Learn C Programming

Course Code: PPS1026	Course Title: Industry Readiness Program – II Type of Course: Practical Only Course	L-T- P- C	0-0-2-0
Version No.	1.0		
Course Pre-requisites	Students are expected to understand basic English. Students should have desire and enthusiasm to get involved, participate, and learn.		
Anti-requisites	NIL		
Course Description	This course is designed to enable students to set SMART goals, form professional & personal ethics for the success and learn various email writing techniques. The course will benefit learners in presenting themselves effectively through various activities and learning methodologies.		
Course Objective	The objective of the course is to familiarize the learners with the concepts of "Employability for Young Professionals" and attain SKILL DEVELOPMENT through PARTICIPATIVE LEARNING techniques.		

Course Outcomes	On successful completion of the course the students shall be able to: **CO1:** Apply different communication skills for success in workplace [Level 3 - Apply] **CO2:** Practice team building skills for career success [Level 3 - Apply] **CO3:** Demonstrate ethical leadership skills in workplace [Level 3 - Apply]		
Course Content:			
Module 1	Effective Communication	Classroom activities	10 Sessions
Topics: Practice effective communication skills (Verbal, Non-verbal, Written and Visual) Activity: Use social media prompts to prepare self-introduction videos			
Module 2	Team Building	Group Activity	10 Sessions
Topics: Skills of an effective team player Activity: Student group activity to build class networking			
Module 3	Leadership	Case study	10 Sessions
Topics: Types of leadership, using empathy in leadership Activity: Individual presentation by students on corporate leaders.			
Targeted Applications & Tools: TED Talks; You Tube Links; Activities			
Assignment proposed for this course Assignment 1: One minute reel Assignment 2: Team building assignment			
Continuous Individual Assessment Module 1: L-S-R-W class assessment Module 2: Team Presentation Module 3: Individual Assessment			

Course Code: EEE1251	**Course Title: Basics of Electricaland Electronics Engineering Laboratory** **Type of Course: ESC - Laboratory**	**L-T-P- C**	0-0-2-1
Version No.	**1.0**		
Course Pre-requisites	**Nil**		
Anti-requisites	**Nil**		
Course Description	This fundamental laboratory provides an opportunity to validate the concepts taught in the basics of electrical and electronics engineering and enhances the ability to visualize real system performance, using both hardware and simulation tools.		
Course Objective	The objective of the course is to familiarize the learners with the concepts of Basics of Electrical and Electronics Engineering and attain Skill Development through Experiential Learning techniques.		
Basic skill sets required for the laboratory	**The students shall be able to develop:** 1. An attitude of enquiry. 2. Confidence and ability to tackle new problems. 3. Ability to interpret events and results.		

	4. Ability to work as a leader and as a member of team. 5. Assess errors and eliminate them. 6. Observe and measure physical phenomenon. 7. Write Reports. 8. Select suitable equipment, instruments and materials. 9. Locate faults in systems. 10. Skills for setting and handling equipment. 11. The ability to follow standard test procedures. 12. An awareness of the need to observe safety precautions. 13. To judge magnitudes without actual measurement.
Course Outcomes	On successful completion of the course, the students shall be able to: 1. **Demonstrate** the behavior of passive elements and fundamental electrical circuit laws 2. **Determine** the power factor, impedance, and phase relationship of R and RL loads under AC supply conditions. 3. **Develop** electrical circuits and three-phase supply systems using MATLAB/Simulink to study resistance, voltage, and current characteristics. 4. **Analyze** the performance characteristics of electrical machines, including single-phase transformers and three-phase induction motors, under different operating conditions. 5. **Examine** the output characteristics and performance parameters of half-wave and full-wave rectifiers. 6. **Illustrate** the performance characteristics of Zener diode and MOSFET under different operating conditions.
Course Content:	List of Laboratory Tasks: **Experiment No 1:**Familiarization of passive elements. Level 1: Identify and understand the function and ratings of basic passive elements. Level 2: Study the behavior of passive elements in simple electrical circuits by measuring voltage, current, and impedance. **Experiment No 2:** Experimental Verification of Kirchhoff's Voltage Law (KVL) and Kirchhoff's Current Law (KCL). Level 1: Study and Verify KVL and KCL for the given electrical Circuit. **Experiment No 3:** Determination of Power Factor, and Impedance for R and RL Loads. Level 1: Determine the power factor and impedance of R and RL loads under steady-state AC supply conditions. Level 2: Analyze the effect of inductance on power factor, impedance, and phase angle for RL loads under varying load conditions. **Experiment No 4:** Modeling and Simulation of effective resistances in different circuit connections Using Simulink. Level 1: Model and simulate series and parallel resistor circuits in Simulink and determine their effective resistance. Level 2: Analyze the effect of different resistor combinations on total resistance, current distribution, and voltage drop. **Experiment No 5:** Simulation and analysis of line and phase voltages in Star-connected and Delta- connected supply systems.

	Level 1: Model and simulate star-connected and delta-connected three-phase supply systems and observe line and phase voltages. Level 2: Analyze and compare the relationship between line and phase voltages in star and delta connections. **Experiment 6:** Determination of transformation ratio of a Single-Phase Transformer Under different tapping conditions. Level 1: Determine the transformation ratio of a single-phase transformer under no-load condition for various tap settings. Level 2: Analyze the effect of different tapping conditions on secondary voltage, regulation, and transformer performance. **Experiment No. 7:** Study of the speed-torque characteristics of a Three-Phase Induction Motor. Level 1: Study and analyze the speed–torque characteristics of a three-phase induction motor. Level 2: Evaluate the effect of supply voltage and current on motor performance parameters such as slip, and torque. **Experiment 8:** Performance Analysis of Half-Wave and Full-Wave Rectifier Circuits Using Simulink. Level 1: Model and simulate half-wave and full-wave rectifier circuits in Simulink and analyze their output waveforms. Level 2: Evaluate and compare rectifier performance parameters such as ripple factor, efficiency, and output voltage. **Experiment 9:** Study of the characteristics of a Zener Diode. Level 1: Study the forward and reverse V–I characteristics of a Zener diode using DC analysis. Level 2: Analyze the Zener breakdown characteristics and determine the Zener voltage and regulation behavior. **Experiment 10:** Study of the Output and Transfer Characteristics of a MOSFET. Level 1: Study, analyze and implement the output characteristics of a MOSFET under different gate-source voltage conditions and observe their results. Level 2: Analyze the transfer characteristics of a MOSFET and evaluate key parameters such as threshold voltage and transconductance.

Targeted Application & Tools that can be used:

Targeted Applications: Application area includes all electrical and electronic circuits (power supply unit, regulator unit, embedded devices, hardware electronics etc.). The students will be able to join a profession which involves basics to high level of electronic circuit design.

Professionally Used Software: MATLAB SIMULINK

Besides these software tools hardware equipment such as Multimeters, Function Generators, Power Supplies, Oscilloscopes etc., can be used to perform component/circuit testing and analysis.

Course Material

1. Basics of Electrical and Electronics Engineering Laboratory Manual, Presidency University, Bengaluru.

Text Book:

1. D. P. Kothari and I. J. Nagrath, Basic Electrical Engineering, 4th Edition, McGraw Hill Education, New Delhi, 2019.

2. B. L. Theraja and A. K. Theraja, A Textbook of Electrical Technology, Volume-I: Basic Electrical Engineering in SI Units, 24th Edition, S. Chand Publishing, New Delhi, 2024.

References

1. John Hiley, Keith Brown and Ian McKenzie-Smith, Hughes Electrical and Electronic Technology, 12th Edition, Pearson Education, 2016.
2. S. K. Bhattacharya and D. Devaraj, Basic Electrical and Electronics Engineering, 2nd Edition, Pearson Education India, 2017.
3. T. K. Nagsarkar and M. S. Sukhija, Principles of Basic Electrical Engineering, 1st Edition, Oxford University Press, 2018.
4. Robert L. Boylestad and Louis Nashelsky, Electronic Devices and Circuit Theory, 12th Edition, Pearson Education, 2019.
5. A. K. Maini and Varsha Agrawal, Electronic Devices and Circuits, 2nd Edition, Wiley India, 2019.
6. Adel S. Sedra and Kenneth C. Smith, Microelectronic Circuits, 8th Edition, Oxford University Press, 2019.

Online Resources:

1. https://presidencyuniversity.linways.com
2. https://www.digimat.in/nptel/courses/video/108105112/L01 "Fundamentals of Electrical Engineering - Basic Concepts, Examples"
3. Video lectures on "Electronic Devices" by Prof. Dr. A. N. Chandorkar, IIT Bombay http://www.satishkashyap.com/2013/03/video-lectures-on-electron-devices-by.html
4. NPTEL lectures on "Rectifiers", by Prof. G. Bhuvaneswari, IIT Delhi https://nptel.ac.in/courses/108102145
5. NPTEL Lectures on "Electrical Machines II", by Prof. G. Sridhara Rao, IIT Madras. https://nptel.ac.in/courses/108106072

Topics relevant to "EMPLOYABILITY SKILLS": All the experiments which are listed are for **Skill Development** through **Experiential Learning Techniques**. This is attained through the assessment component mentioned in course handout.

Topics relevant to "HUMAN VALUES & PROFESSIONAL ETHICS": Standard test methods, safety procedure.

Catalogue prepared by	
Recommended by the Board of Studies on	
Date of Approval by the Academic Council	

Course Code: MAT2311	**Course Title: Vector Calculus, Numerical Methods and Sampling Distributions** **[Type of Course:1] School Core**	**L-T- P- C**	**3**	**1**	**0**	**4**
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-	NIL					

requisites		
Course Description	This course introduces the fundamental concepts and methods of ordinary differential equations, numerical techniques, and statistical inference used in engineering and scientific applications. The course covers analytical methods for solving differential equations, numerical methods for solving linear and nonlinear equations, interpolation, numerical differentiation and integration, and numerical solutions of ordinary differential equations. It also introduces sampling distributions and hypothesis testing for statistical decision-making. The course enables students to apply mathematical and statistical techniques to solve computational and real-world engineering problems.	
Course Objective	The objectives of this course are to develop the ability to solve ordinary differential equations arising in scientific and engineering applications. Introduce numerical techniques for solving systems of linear equations and nonlinear algebraic equations. Apply interpolation, differentiation, and integration methods for approximating mathematical functions and evaluating definite integrals. Employ numerical methods for obtaining approximate solutions of ordinary differential equations. Apply concepts of sampling distributions and statistical inference to estimate population parameters and support decision-making. Apply hypothesis testing techniques to real-world problems involving means, proportions, variances, and associations among variables.	
Course Outcomes	After successful completion of the course, students will be able to: **CO1:** Employ vector differential operators to solve problems involving scalar and vector fields and classify them as solenoidal or irrotational (L3) **CO2:** Apply vector integration techniques and integral theorems to a solve engineering problems. (L3) **CO3:** Use numerical methods to solve systems of linear and nonlinear equations. (L3) **CO4:** Implement numerical techniques for interpolation, differentiation, integration, and ordinary differential equations. (L3) **CO5:** Utilize sampling distributions and statistical inference techniques to solve statistical problems. (L3) **CO6:** Perform hypothesis testing for statistical decision-making. (L3)	
Course Content:		
Module 1	**Vector Calculus**	**(14L+5T=19 Sessions)**
Vector & Scalar Functions and Fields, Derivatives, Curve, Arc length, Curvature & Torsion, Gradient of Scalar Field, Directional Derivative, Divergence of a Vector Field, Curl of a Vector Field, Physical interpretation, Solenoidal and Irrotational Vector Fields. Problems. Line Integrals, Path Independence of Line Integrals, Surface and Volume Integrals, Green's Theorem in the plane, Divergence Theorem of Gauss, Stokes's Theorem. Problems on Image Edge Detection, Robot Path Planning and 3D Surface Modelling		
Module 2	**Solution of Linear Systems and non-linear of Equation**	**(9L+3T=12 Sessions)**

Introduction, Jacobi iteration method, Gauss-Seidel iteration method. Solution of Algebraic and Transcendental Equations: Bisection method, Regula-Falsi method, Newton-Raphson method, secant method, fixed point iteration method.

Page Ranking, Network Analysis, Circuit Simulation, Root Finding, Load Balancing, and Computational Optimization Problems.

Module 3	**Numerical Interpolations and Differential Equations**	**(10L+3T=13 Sessions)**

Newton's method, Divided difference method, Lagrange's method, Numerical differentiation, Numerical integration: Trapezoidal rule, Simpson's one-third rule, Simpson's three-eighth rule, Gaussian quadrature rule. Solution of ordinary differential equations: Taylor series method, modified Euler's method, Runge-Kutta method for 4th order. Problems on Signal Reconstruction, Sensor Data Interpolation and Trajectory Prediction.

Module 4	**Sampling Distributions**	**(12L+4T=16 Sessions)**

Random sampling, sampling distributions, Standard Error, Type I & Type II errors, Testing of Hypothesis, Test of significance - Large sample test for single proportion, difference of proportions, single mean, difference of means, and difference of standard deviations, Test for single mean, difference of means, test for ratio of variances - Chi-square test for goodness of fit and independence of attributes. A/B Testing, Fraud Detection, Spam Classification Evaluation, and Network Traffic Analysis.

Targeted Applications & Mathematical Tools:

- Analysis of fluid flow in open channels and pipelines, groundwater movement, stress and strain fields, terrain modelling, and 3D surface representation in civil engineering projects.
- Structural analysis, load distribution in trusses and frames, hydraulic network analysis, foundation design computations, and optimization problems in transportation and water resource systems.
- Estimation of terrain elevations, survey data interpolation, rainfall-runoff modelling, settlement prediction, traffic flow analysis, and numerical solutions of engineering differential equations.
- Quality control of construction materials, reliability analysis, traffic studies, environmental data analysis, hydrological investigations, and hypothesis testing in engineering experiments.
- Analysis of vector fields in fluid flow, electromagnetics, and heat transfer; image edge detection, robot path planning, and 3D surface modelling using gradients, divergence, curl, and integral theorems.

Software: MATLAB, Python, Microsoft Excel Solver, LINDO/LINGO, AutoCAD, ETABS, Scilab.

Assignment:

Assignments will involve theoretical problem-solving and computational implementation using suitable programming tools. The assignments may be undertaken individually or in groups, as decided by the instructor.

Text Book

1. M. K. Jain, S. R. K. Iyengar, and R. K. Jain, *Numerical Methods for Scientific and Engineering Computation*, 7th ed. New Delhi, India: New Age International Publishers, 2019.

2. B. S. Grewal, *Higher Engineering Mathematics*, 45th ed. New Delhi, India: Khanna Publishers, 2022.

3. S. C. Gupta and V. K. Kapoor, *Fundamentals of Mathematical Statistics*, 12th ed. New Delhi, India: Sultan Chand & Sons, 2020.

References:

1. M. K. Jain, S. R. K. Iyengar, and R. K. Jain, *Numerical Methods for Scientific and Engineering Computation*, 7th ed. New Delhi, India: New Age International Publishers, 2019.

2. S. S. Sastry, *Introductory Methods of Numerical Analysis*, 5th ed. New Delhi, India: PHI Learning Pvt. Ltd., 2012.

3. J. Kiusalaas, *Numerical Methods in Engineering with Python 3*, 3rd ed. Cambridge, U.K.: Cambridge University Press, 2025.

4. D. C. Montgomery and G. C. Runger, *Applied Statistics and Probability for Engineers*, 8th ed. Hoboken, NJ, USA: John Wiley & Sons, 2023.

5. D. D. Wackerly, W. Mendenhall, and R. L. Scheaffer, *Mathematical Statistics with Applications*, 7th ed. Belmont, CA, USA: Cengage Learning, 2008.

6. E. Kreyszig, *Advanced Engineering Mathematics*, 10th ed. Hoboken, NJ, USA: John Wiley & Sons, 2011.

7. R. L. Burden, J. D. Faires, and A. M. Burden, *Numerical Analysis*, 11th ed. Boston, MA, USA: Cengage Learning, 2021.

8. S. C. Chapra, *Applied Numerical Methods with MATLAB for Engineers and Scientists*, 4th ed. New York, NY, USA: McGraw-Hill Education, 2018.

E-resources/ Web links:

1. https://presiuniv.knimbus.com/portal/v2/default/viewDetail?searchResultType=ECATALOGUE_BASED&uniqueId=CUSTOM_PACKAGE_EBSCO_29052023_53751
2. https://presiuniv.knimbus.com/portal/v2/default/viewDetail?searchResultType=ECATALOGUE_BASED&uniqueId=TEXTBOOK_LIBRARY01_06082022_117
3. https://presiuniv.knimbus.com/portal/v2/default/viewDetail?searchResultType=ECATALOGUE_BASED&uniqueId=DOAB_1_06082022_7313
4. https://presiuniv.knimbus.com/portal/v2/default/viewDetail?searchResultType=ECATALOGUE_BASED&uniqueId=CUSTOM_PACKAGE_EBSCO_29052023_228965
5. https://presiuniv.knimbus.com/portal/v2/default/viewDetail?searchResultType=ECATALOGUE_BASED&uniqueId=DOAB_1_06082022_8838
6. https://presiuniv.knimbus.com/portal/v2/default/viewDetail?searchResultType=ECATALOGUE_BASED&uniqueId=SPRINGER_OPEN_ACCESS_BOOKS_10022025_1861
7. https://presiuniv.knimbus.com/portal/v2/default/viewDetail?searchResultType=ECATALOGUE_BASED&uniqueId=DOAB_1_06082022_7312

Topics relevant to SKILL DEVELOPMENT: Identification and solution of differential equations, application of numerical methods for algebraic and differential equations, implementation of interpolation and numerical integration techniques, statistical data analysis and hypothesis testing, and use of Python/MATLAB for

engineering problem solving.	
Catalogue prepared by	**Dr. V Ramalatha**
Recommended by the Board of Studies on	
Date of Approval by the Academic Council	

Course Code: CSS2251	Course Name: Data Communications and Computer Networks Type of Course: Theory Course	L- T-P- C	3-0-0-3
Version No.	1.0		
Course Pre-requisites	NIL		
Co-requisites			
Anti-requisites	NIL		
Course Description	The objective of this course is to provide knowledge in data communications and computer networks, its organization and its implementation, and gain practical experience in the installation, monitoring, and troubleshooting of LAN systems. . The associated laboratory is designed to implement and simulate various networks using Cisco packet tracer, NS2. All the lab exercises will focus on the fundamentals of creating multiple networks, topologies and analyzing the network traffics.		
Course Objective	The objective of the course is to familiarize the learners with the concepts of Data Communications and Computer Networks and attain Employability through Problem Solving Methodologies.		
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Illustrate The Basic Concepts Of Data Communication And Computer Networks. [Level 3 - Apply] **CO2**: Analyze the functionalities of the Data Link Layer. [Level 4 – Analyze] **CO3:** Apply the Knowledge of IP Addressing and Routing Mechanisms in Computer Networks. [Level 3 – Apply] **CO4:** Demonstrate the working principles of the Transport layer and Application Layer. [Level 3 – Apply]		
Course Content:			
Module 1	Introduction and Physical layer	Assessment: Assignment (Problem Solving)	9 Sessions

Topics:

Introduction to Computer Networks and Data communications, Network Components – Topologies, Transmission Media –Reference Models - TCP/IP Suite, OSI Model .

Physical Layer -Analog and Digital Signals – Digital and Analog Signals – Transmission - Multiplexing and Spread Spectrum.

Module 2	Data Link Layer	Assessment: Assignment (Problem Solving)	12 Sessions

Topics:

Data Link Layer - Error Detection and Correction– Parity, CRC, Hamming Distance Flow Control and Error Control, Stop and Wait, Multiple Access Protocols, CSMA/CD,CSMA/CA

Module 3	Network Layer	Assessment: Assignment (Problem Solving)	13 Sessions

Topics:

Network Layer Services - Network Layer Services, Switching Techniques, IP Addressing methods- IPv4 IPV6 – Subnetting. Routing, - Distance Vector Routing, Link State Routing, RIP, OSPF, BGPV4.

Module 4	Transport and Application Layer	Assessment: Assignment (Problem Solving)	11 Sessions

Topics:

Transport Layers - Connection management – Flow control-Sliding Window, Go-Back N ARQ, Selective Repeat ARQ, UDP, TCP, congestion control, Congestion avoidance The Application Layer: Domain Name System (DNS), Domain Name Space, FTP, Electronic Mail (SMTP), HTTP.

Targeted Applications & Tools: Cisco Packet Tracer, Wireshark, and NS2 Simulator.

Skill-Building Activities through Experiential Learning:

Text Book

T1: T1. Behrouz A. Forouzan, “Data Communications and Networking with TCP/IP Protocol Suite”, 6th Edition, Tata McGraw-Hill, 2022.
T2: Andrew S Tanenbaum, Nick Feamster & David J Wetherall, “Computer Networks” Sixth Edition, Pearson Publication, 2022.

References

R1: “Computer Networking: A Top-Down Approach”, Eighth Edition, James F. Kurose, Keith W. Ross, Pearson publication, 2021.
R2: William Stallings, Data and Computer Communication, 8th Edition, Pearson Education, 2007.
R3: Behrouz A. Forouzan, “Data Communications and Networking 5E”, 5th Edition, Tata McGraw-Hill, 2012

E-Resources/Web Resources

1. https://www.geeksforgeeks.org/what-is-spread-spectrum/

2.	https://www.geeksforgeeks.org/difference-between-fdma-tdma-and-cdma/
3.	https://archive.nptel.ac.in/courses/106/105/106105183/
4.	http://www.nptelvideos.com/course.php?id=393
5.	https://www.digimat.in/keyword/106.htmlhttps://puniversity.informaticsglobal.com/login

Course Code: CSS2252	Course Name: Data Communications and Computer Networks Lab Type of Course: Laboratory Course	L- T-P- C	0-0-2-1
Version No.	1.0		
Course Pre-requisites	NIL		
Co-requisites			
Anti-requisites	NIL		
Course Description	This lab-based course provides hands-on experience in the principles and practices of data communications and computer networking. It is designed to complement theoretical concepts covered in the associated lecture course. Through a series of structured experiments and practical exercises, students will gain proficiency in configuring, analyzing, and troubleshooting computer networks. Key topics include network topology design, IP addressing and subnetting, Ethernet and LAN technologies, routing and switching, TCP/IP protocol suite, and basic network security measures. Students will work with industry-standard tools and equipment, including routers, switches, protocol analyzers, and network simulation software such as Cisco Packet Tracer or Wireshark.		
Course Objective	The objective of this lab course is to provide students with practical, hands-on experience in the configuration, operation, and troubleshooting of data communication systems and computer networks. Through guided experiments and real-world scenarios, students will reinforce theoretical knowledge, develop essential technical skills, and gain a deeper understanding of networking concepts, protocols, and devices used in modern communication systems.		
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Design and configure basic network topologies using routers, switches, and end devices to meet specified requirements [Level 3 – Apply] . **CO2:** Analyze and troubleshoot network connectivity and performance issues using tools such as Wireshark and network simulators. [Level 4 – Analyze] **CO3:** Demonstrate understanding of key networking protocols (e.g., TCP/IP, ARP, ICMP, DHCP) through [Level 3 – Apply]		

	practical implementation and observation. **CO4:** Apply IP addressing and subnetting techniques to efficiently allocate and manage network resources in various networking scenarios. [Level 3 – Apply]

Course Content:

List of Laboratory Tasks:

- **Lab sheet -1, M-1, 3 [2 Hours]**
- **Experiment No 1:**
  Level 1: Study of basic network commands and network configuration commands.

  Lab sheet -2, M-1[2 Hours]
- **Experiment No 1:**
  Level 1: Identify and explore Network devices, models and cables. Introduction to Cisco packet tracer.
- **Experiment No. 2:**
  Level 2 – Create various network topologies using a cisco packet tracer.

  Lab sheet -3, M-2,3 [2 Hours]
- **Experiment No. 1:**
  Level 2 - Basic Configuration of switch/router using Cisco packet tracer.
- **Experiment No. 2:**
  Level 2 -Configure the privilege level password and user authentication in the switch/router.

  Lab sheet – 4, M-3 [2 Hours]
- **Experiment No. 1:**
  Level 2 - Configure the DHCP server and wireless router and check the connectivity

  Lab sheet – 5, M-3 [2 Hours]
- **Experiment No. 1:**
  Level 2 - Configure the static routing in the Cisco packet tracer.
- **Experiment No. 2:**
  Level 2 - Configure the dynamic routing protocol in the Cisco packet tracer.

  Lab sheet – 6, M-4 [2 Hours]
- **Experiment No. 1:** Configuration of DNS Server with Recursive & Integrative approach in Cisco packet tracer.

  Lab sheet – 7, M-4 [2 Hours]
- **Experiment No. 1:**
  Configure the telnet protocol in the router using the Cisco packet tracer.

  Lab sheet – 8, M-4[2 Hours]
- **Experiment No. 1:**
  Level1- Introduction to NS2 and basic TCL program.

  Lab sheet – 9, M-4 [2 Hours]

- **Experiment No. 1:**
  Level 1: Simulate three node Point to point network using UDP in NS2.
- **Experiment No. 2:**
  Simulate transmission of Ping message using NS2.

Lab sheet – 10, M-4[2 Hours]

- **Experiment No. 1:**
  Simulate Ethernet LAN using N-node in NS2.
- **Experiment No. 2:**
  Simulate Ethernet LAN using N-node using multiple traffic in NS2

Lab sheet –11, M-3,4 [2 Hours]

- **Experiment No. 1:**
  Level 1- Introduction to Wire Shark.
- **Experiment No. 2:**
  Level 2- Demonstration of packet analysis using wire shark.

Lab sheet –12, M-1,2,3 [2 Hours]

- **Experiment No. 1:**
  Level 2- Demonstration of switch and router configuration using real devices

Targeted Applications & Tools:

Cisco Packet Tracer, Wireshark, and NS2.

Case Study/Assignment: Choose and analyze a network from any organization/Assignment proposed for this course in CO1-CO4

Problem Solving: Choose and appropriate devices and implement various network concepts.

Programming: Simulation of any network using NS2.

Skill-Building Activities through Experiential Learning:

Text Book

T1: Behrouz A. Forouzan, "Data Communications and Networking 5E", 5 th Edition, Tata McGraw-Hill, 2017.
T2: Andrew S Tanenbaum, Nick Feamster & David J Wetherall, "Computer Networks" Sixth Edition, Pearson Publication, 2022

References

R1: "Computer Networking: A Top-Down Approach", Eighth Edition, James F. Kurose, Keith W. Ross, Pearson publication, 2021.
R2: William Stallings, Data and Computer Communication, 8th Edition, Pearson Education, 2007.
R3: Larry L. Peterson and Bruce S. Davie: Computer Networks – A Systems Approach, 4th Edition, Elsevier, 2007.

E-Resources/Web Resources

1.**https://archive.nptel.ac.in/courses/106/105/106105183/** **2.http://www.nptelvideos.com/course.php?id=393** **3.https://www.youtube.com/watch?v=3DZLItfbqtQ** **4.https://www.youtube.com/watch?v=_fIdQ4yfsfM** **5.https://www.digimat.in/keyword/106.html** 6. **https://puniversity.informaticsglobal.com/login**

Course Code: CSS2253	Course Title: Data Structures Type of Course: Theory Course (PCC)	L-T- P- C	3-0-0-3
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course introduces the fundamental concepts of data structures and emphasizes the importance of selecting appropriate data structures and techniques for efficient program development. The course includes both theory and laboratory components focusing on the implementation and applications of data structures using the Java programming language. It also introduces emerging applications of data structures in cybersecurity, Artificial Intelligence, Machine Learning, blockchain, and intelligent computing systems. With strong knowledge of data structure fundamentals and practical implementation skills, students will be able to design and develop efficient, secure, and intelligent software applications.		
Course Objective	To develop a strong foundation in data structures by enabling students to understand fundamental data structures and their operations, analyze algorithm efficiency, implement linear and non-linear data structures, and apply appropriate techniques to solve real-world problems, including applications in cybersecurity, AI/ML, blockchain, and intelligent systems, while evaluating trade-offs between different data structures.		
Course Outcomes	On successful completion of this course, students will be able to: CO1: Explain the concepts of basic data structures such as arrays, stacks, and queues and their operations. — Level 2 – Understand CO2: Apply linked list data structures to solve real-world problems. — Level 3 – Apply CO3: Utilize appropriate non-linear data structures such as trees and graphs for given problem scenarios. — Level 3 – Apply CO4: Implement and demonstrate various searching and sorting algorithms and analyze the efficiency of data structures and algorithms in terms of time and space complexity. — Level 3 – Apply CO5: Apply data structure concepts in cybersecurity, AI/ML, blockchain, and intelligent computing applications. — Level 3 – Apply		

Course Content:		
Module 1	Introduction to Data Structure and Linear Data Structure – Stacks and Queues	9 Sessions
Introduction – Introduction to Data Structures, Types and concept of Arrays. Stack - Concepts and representation, Stack operations, stack implementation using array and Applications of Stack. Queues - Representation of queue, Queue Operations, Queue implementation using array, Types of Queue and Applications of Queue.		
Module 2	Linear Data Structure- Linked List	12 Sessions
Topics: Linked List - Singly Linked List, Operation on linear list using singly linked storage structures, Circular List, Applications of Linked list. Recursion - Recursive Definition and Processes.		
Module 3	Non-linear Data Structures - Trees	12 Sessions
Topics: Trees - Introduction to Trees, Binary tree: Terminology and Properties, Use of Doubly Linked List, Binary tree traversals: Pre-Order traversal, In-Order traversal, Post - Order traversal, Binary Serach Tree, AVL Trees - Red Black Tree, Expression Tree , Heaps.		
Module 4	Non-linear Data Structures – Graphs, Hashing, Searching and Sorting	6 Sessions
Topics: Graphs: Basic Concept of Graph Theory and its Properties, Representation of Graphs . ADT, Elementary graph operations, Minimum Cost spanning trees, Shortest path and Transitive closure. Hashing: Introduction, Static Hashing, Dynamic Hashing Searching & Sorting Sequential and Binary Search, Sorting – Selection and Insertion sort, Quick sort, Merge Sort, Bubble sort.		
Module 5	Security and AI/ML Applications of Data Structures	6 Sessions
Topic: Secure Data Structures, Hashing Techniques, Secure Searching and Sorting, Trees and Graphs in Cybersecurity, Data Structures for AI & ML, Graph-based Intelligent Systems, Heaps in Machine Learning, Applications in Fraud and Intrusion Detection, Blockchain Data Structures, Quantum Computing Data Representation.		
Targeted Application & Tools that can be used Use of PowerPoint software for lecture slides and use of Modern IDE like VS Code and Eclipse for lab programs to execute.		
Project work/Assignment:		
Assignment: Students should complete the lab programs by end of each practical session and module wise assignments before the deadline.		

Text Book T1 Ellis Horowitz and Sartaj Sahni, Fundamentals of Data Structures in C, 2nd Edition, Universities Press, reprint 2018. T2 Seymour Lipschutz, Data Structures Schaum's Outlines, Revised 1st Ed, McGraw Hill, 2014.
References R1 Data structures and program design in C by Robert Kruse, Tondo C L, Bruce Leung, Pearson education publishers, 2017. R2 Programming and Data Structure by Jackulin C Salini etal., Ane books publishers, 2019. Web resources: 1. For theory: https://onlinecourses.nptel.ac.in/noc20_cs85/preview 2. https://puniversity.informaticsglobal.com/login
Topics relevant to development of "Skill Development": Linked list and stacks Topics relevant to development of "Environment and sustainability: Queues

Course Code:: CSS2254	Course Title: Data Structures Lab Type of Course:Laboratory Course (PCC)	L-T- P- C	0-0-2-1
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course introduces the fundamental concepts of data structures and to emphasize the importance of choosing an appropriate data structure and technique for program development. This course has theory and lab component which emphasizes on understanding the implementation and applications of data structures using Java programming language. With a good knowledge in the fundamental concepts of data structures and practical experience in implementing them, the student can be an effective designer, developer for new software applications.		
Course Objective	To develop programming and problem-solving skills by enabling students to implement various data structures, analyze their performance through experimentation and apply structured approaches to solve computational problems effectively.		
Course Outcomes	On successful completion of this course, students will be able to: CO1: Explain the concepts of fundamental data structures such as arrays, stacks, and queues and their operations. Understand CO2: Implement linear data structures Apply		

	such as linked lists to solve real-world problems.	
	CO3: Utilize appropriate non-linear data structures such as trees and graphs for given problem scenarios.	Apply
	CO4: Implement and demonstrate various searching and sorting algorithms and analyze the efficiency of data structures and algorithms in terms of time and space complexity.	Apply
	CO5: Apply data structure concepts in cybersecurity, AI/ML, blockchain, and intelligent computing applications.	Apply

Course Content: 30 Sessions

List of Laboratory Tasks: 30 Sessions

WEEK 1: Introduction & Arrays
Lab Sheet 1: Arrays Basics
Level 1: Write a program to create an array and perform insertion and deletion operations.
Level 2: Write a program to find largest, smallest, and average elements in an array.
Lab Sheet 2: Array Applications
Level 1: Write a program to reverse an array.
Level 2: Write a program to merge two arrays and remove duplicates.

WEEK 2: Stack
Lab Sheet 3: Stack Implementation
Level 1: Implement stack using array (push and pop operations).
Level 2: Write a program to check overflow and underflow conditions.
Lab Sheet 4: Stack Applications
Level 1: Write a program to convert infix expression to postfix.
Level 2: Write a program to evaluate postfix expression.

WEEK 3: Queue
Lab Sheet 5: Queue Implementation
Level 1: Implement queue using array (enqueue and dequeue).
Level 2: Implement circular queue.
Lab Sheet 6: Queue Applications
Level 1: Write a program to simulate a queue in real-life scenario (ticket booking).
Level 2: Implement priority queue.

WEEK 4: Linked List Basics
Lab Sheet 7: Singly Linked List
Level 1: Create a singly linked list and perform insertion.
Level 2: Perform deletion and display operations.
Lab Sheet 8: Linked List Applications
Level 1: Count number of nodes in a linked list.
Level 2: Reverse a linked list.

WEEK 5: Advanced Linked List

Lab Sheet 9: Circular Linked List
Level 1: Create and traverse a circular linked list.
Level 2: Perform insertion and deletion in circular linked list.
Lab Sheet 10: Linked List Problems
Level 1: Merge two linked lists.
Level 2: Detect loop in linked list.

WEEK 6: Recursion
Lab Sheet 11: Recursion Basics
Level 1: Write a recursive program for factorial.
Level 2: Write a recursive program for Fibonacci series.
Lab Sheet 12: Recursion Applications
Level 1: Solve Tower of Hanoi using recursion.
Level 2: Compare recursive and iterative approaches.

WEEK 7: Trees Basics
Lab Sheet 13: Binary Tree
Level 1: Create a binary tree and perform traversal.
Level 2: Implement preorder, inorder, and postorder traversal.
Lab Sheet 14: Tree Properties
Level 1: Count nodes and height of tree.
Level 2: Check if tree is balanced.

WEEK 8: Binary Search Tree
Lab Sheet 15: BST Implementation
Level 1: Insert elements into BST.
Level 2: Delete elements from BST.
Lab Sheet 16: BST Applications
Level 1: Search an element in BST.
Level 2: Find minimum and maximum elements.

WEEK 9: Advanced Trees
Lab Sheet 17: AVL Trees
Level 1: Implement AVL tree insertion.
Level 2: Perform rotations (LL, RR, LR, RL).
Lab Sheet 18: Heaps
Level 1: Implement max heap.
Level 2: Perform heap sort.

WEEK 10: Expression Trees & Red-Black Trees
Lab Sheet 19: Expression Tree
Level 1: Construct expression tree from postfix.
Level 2: Evaluate expression tree.
Lab Sheet 20: Red-Black Tree
Level 1: Understand insertion rules.
Level 2: Implement basic operations.

WEEK 11: Graph Basics
Lab Sheet 21: Graph Representation

Level 1: Represent graph using adjacency matrix.
Level 2: Represent graph using adjacency list.
Lab Sheet 22: Graph Traversal
Level 1: Implement BFS.
Level 2: Implement DFS.

WEEK 12: Graph Algorithms
Lab Sheet 23: Shortest Path
Level 1: Implement Dijkstra's algorithm.
Level 2: Analyze shortest path in weighted graph.
Lab Sheet 24: Minimum Spanning Tree
Level 1: Implement Prim's algorithm.
Level 2: Implement Kruskal's algorithm.

WEEK 13: Hashing
Lab Sheet 25: Hash Tables
Level 1: Implement hashing using division method.
Level 2: Handle collisions using chaining.
Lab Sheet 26: Advanced Hashing
Level 1: Implement linear probing.
Level 2: Compare static vs dynamic hashing.

WEEK 14: Searching
Lab Sheet 27: Searching Techniques
Level 1: Implement sequential search.
Level 2: Implement binary search.
Lab Sheet 28: Searching Applications
Level 1: Compare searching techniques.
Level 2: Analyze time complexity.
Lab Sheet 29: Basic Sorting
Level 1: Implement bubble sort and selection sort.
Level 2: Implement insertion sort.
Lab Sheet 30: Advanced Sorting
Level 1: Implement quick sort.
Level 2: Implement merge sort and compare performance
WEEK 15: Sorting
Lab Sheet 31: Secure Data Structures and Hashing
Level 1: Implement hashing techniques for secure data storage and retrieval.
Level 2: Demonstrate collision handling methods and simple password verification using hashing.
Lab Sheet 32: AI/ML and Intelligent Data Structure Applications
Level 1: Implement graph or tree-based data representation for intelligent systems.
Level 2: Develop a simple application using data structures for recommendation, prediction, or intrusion detection systems.

Targeted Application & Tools that can be used
Use of PowerPoint software for lecture slides and use of Modern IDE like VS Code and Eclipse for lab programs to execute.

Project work/Assignment:

Assignment: Students should complete the lab programs by end of each practical session and module wise assignments before the deadline.
Text Book T1 Ellis Horowitz and Sartaj Sahni, Fundamentals of Data Structures in C, 2nd Edition, Universities Press, reprint 2018. T2 Seymour Lipschutz, Data Structures Schaum's Outlines, Revised 1st Ed, McGraw Hill, 2014.
References R1 Data structures and program design in C by Robert Kruse, Tondo C L, Bruce Leung, Pearson education publishers, 2017. R2 Programming and Data Structure by Jackulin C Salini etal., Ane books publishers, 2019. Web resources: 1. For theory: https://onlinecourses.nptel.ac.in/noc20_cs85/preview 2. https://puniversity.informaticsglobal.com/login
Topics relevant to development of "Skill Development": Linked list and stacks Topics relevant to development of "Environment and sustainability: Queues

Course Code: CSS2255	Course Title: Object Oriented Programming Using Java Type of Course: Theory	L-T- P- C	3-0-0-3
Version No.	2.0		
Course Pre-requisites	CSE1502		
Anti-requisites	Nil		
Course Description	This course introduces the core concepts of Object-Oriented Programming. The course includes both theory and laboratory components that emphasize the implementation and application of object-oriented programming paradigms using Java. It enables students to develop real-time, secure, and intelligent applications by applying object-oriented concepts for effective problem solving. The course also introduces modern concepts related to AI/ML-based application development, secure programming practices, and intelligent software systems. Students will understand the significance of object-oriented programming in designing scalable, reusable, secure, and intelligent applications.		
Course Objective	The objective of the course is to familiarize learners with problem-solving concepts using Java and to attain skill development through experiential learning techniques. The course also aims to develop the ability to design secure, scalable, and intelligent applications using object-oriented programming concepts with exposure to AI/ML-oriented programming practices and modern software development approaches.		
Course Out Comes	On successful completion of the course the students shall be able to: CO1: Describe the basic programming concepts. L2 – Understand CO2: Apply the concept of classes, objects, and methods to solve problems. L3 – Apply		

	CO3: Apply the concept of arrays and strings. L3 – Apply CO4: Implement inheritance and polymorphism for building secure applications. L3 – Apply CO5: Apply Java I/O concepts for intelligent data processing, secure file handling, and AI/ML-based application development. L3 – Apply			
Course Content:				
Module 1	Basic Concepts of Programming and Java	Assignment	Problem Solving	9 Sessions
Topics: Introduction to Principles of Programming: Process of Problem Solving, Java program structure, Download Eclipse IDE to run Java programs, Sample program, Data types, Identifiers, Variables, Constants in java, Operators, Assignments and Expression, Basic Input/ Output functions, Control Statements: Branching and Looping.				
Module 2	Classes, objects, methods and Constructors	Assignment	Problem Solving	10 Sessions
Topics: Classes, Objects and Methods: Introduction to object Oriented Principles, defining a class, adding data members and methods to the class, access specifiers, instantiating objects, reference variable, accessing class members and methods. Static Polymorphism: Method overloading, constructors, constructor overloading, this keyword, static keyword, Nested classes, Accessing members in nested classes.				
Module 3	Arrays, String and String buffer	Assignment	Problem Solving	8 Sessions
Topics: Arrays: Defining an Array, Initializing & Accessing Array, Multi –Dimensional Array, Array of objects. String: Creation & Operation. String builder class, methods in String Buffer.				
Module 4	Inheritance and Polymorphism	Assignment	Problem Solving	10 Sessions
Topics: Inheritance: Defining a subclass, Types of Inheritance, super keyword. Dynamic Polymorphism: Method overriding. Final keyword: with data members, with member functions and with class. Abstract keyword: with data members, with member functions and with class, Exception handling.				
Module 5	Advanced Java I/O, Security and AI/ML Applications	Assignment	Problem Solving	8 Sessions

Input/output Operation in Java (java.io Package), Streams and the New I/O Capabilities, Understanding Streams, Working with File Object, File I/O Basics, Reading and Writing to Files, Buffer and Buffer Management, Read/Write Operations with File Channel, Serializing Objects, Observer and Observable Interfaces, AI-based File Processing Concepts, Intelligent Data Stream Management, Machine Learning Applications in File and Data Handling.

Text Book

T1 Herbert Schildt, "The Complete Reference Java 2", Tata McGraw Hill Education, 11th Edition,2019.

References

R1. Cay S Horstmann and Cary Gornell, "CORE JAVA volume I-Fundamentals", Tenth Edition, Pearson 2015.

R2: James W. Cooper, "Java TM Design Patterns – A Tutorial", Addison-Wesley Publishers.4th Edition, 2000.

R3. E. Balagurusamy, "Programming with Java", Tata McGraw Hill Education, 6th Edition, 2019.

E book link R1: http://rmi.yaht.net/bookz/core.java/9780134177373-Vol-1.pdf

E book link R2: Java(tm) Design Patterns: A Tutorial([PDF] [7qmsenjl97t0] (vdoc.pub)

Web resources

https://youtube.com/playlist?list=PLu0W_9lII9agS67Uits0UnJyrYiXhDS6q

https://puniversity.informaticsglobal.com:2229/login.aspx

Topics relevant to development of "Skill Development":

Static Polymorphism

Method overloading, constructors

constructor overloading

this keyword

static keyword and Inner classes

Inheritance and Polymorphism.

for Skill Development through Experiential Learning techniques. This is attained through assessment component mentioned in course handout.

<table>
<tr><td>Course Code: CSS2256</td><td>Course Title: Object Oriented Programming Using Java Lab
Type of Course: Laboratory Course (PCC)</td><td>L-T- P- C</td><td>0</td><td>0</td><td>2</td><td>1</td></tr>
<tr><td>Version No.</td><td colspan="6">2.0</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="6">NIL</td></tr>
<tr><td>Anti-requisites</td><td colspan="6">NIL</td></tr>
<tr><td>Course Description</td><td colspan="6">This course introduces the core concepts of Object-Oriented Programming. The course includes both theory and laboratory components that emphasize the implementation and application of object-oriented programming paradigms using Java. It enables students to develop real-time, secure, and intelligent applications by applying object-oriented concepts for effective problem solving. The course also introduces modern concepts related to AI/ML-based application development, secure programming practices, and intelligent software systems. Students will understand the significance of object-oriented programming in designing scalable, reusable, secure, and intelligent applications.</td></tr>
<tr><td>Course Objective</td><td colspan="6">The objective of the course is to familiarize learners with problem-solving concepts using Java and to attain skill development through experiential learning techniques. The course also aims to develop the ability to design secure, scalable, and intelligent applications using object-oriented programming concepts with exposure to AI/ML-oriented programming practices and modern software development approaches.</td></tr>
<tr><td>Course Outcomes</td><td colspan="6">On successful completion of the course, students shall be able to:
CO1: Demonstrate fundamental programming concepts. — L3 – Apply
CO2: Apply the concepts of classes, objects, and methods to solve problems. — L3 – Apply
CO3: Apply the concepts of arrays and strings. — L3 – Apply
CO4: Implement inheritance and polymorphism to develop secure applications. — L3 – Apply
CO5: Apply Java I/O concepts for intelligent data processing, secure file handling, and AI/ML-based application development. — L3 – Apply</td></tr>
<tr><td colspan="3">Course Content:</td><td colspan="4">30 Sessions</td></tr>
<tr><td colspan="7">List of Laboratory Tasks:
Week 1:
Lab Sheet 1: Java Program Structure & Basic I/O
Level 1:
Write a Java program to display “Hello, Java”.
Demonstrate use of variables and data types.
Level 2:
Write a Java program to take user input (name, age) and display formatted output.</td></tr>
</table>

Week 2:

Lab Sheet 2: Operators and Expressions
Level 1:
Write a Java program to perform arithmetic operations (+, -, *, /, %).
Level 2:
Develop a program to evaluate a complex expression using relational and logical operators.

Week 3:

Lab Sheet 3: Control Statements – Branching
Level 1:
Write a Java program to check whether a number is even or odd using if-else.
Level 2:
Implement a menu-driven calculator using switch-case.

Week 4:

Lab Sheet 4: Control Statements – Looping
Level 1:
Write programs using loops:
1. Print numbers from 1 to N
2. Find factorial of a number

Level 2:
Generate Fibonacci series and check whether a number is prime.

Week 5:

Lab Sheet 5: Classes and Objects
Level 1:
Create a class Student with data members and display details using object.
Level 2:
Create multiple objects and demonstrate accessing members using reference variables.

Week 6:

Lab Sheet 6: Methods and Access Specifiers
Level 1:
Write a class with methods to perform addition and subtraction.
Level 2:
Demonstrate use of public, private, and protected access specifiers.

Week 7:

Lab Sheet 7: Constructors and Overloading
Level 1:
Implement a class with a default constructor and parameterized constructor.
Level 2:
Demonstrate constructor overloading and use of this keyword.

Week 8:

Lab Sheet 8: Static Keyword & Nested Classes
Level 1:
Write a program demonstrating static variables and static methods.
Level 2:
Create a nested class and access its members from the outer class.

Week 9:

Lab Sheet 9: Arrays (1D and 2D)
Level 1:
Write programs for:
1. Sum of array elements
2. Find largest element

Level 2:
Perform matrix addition and multiplication using 2D arrays.

Week 10:

Lab Sheet 10: Array of Objects
Level 1:
Create an array of objects (e.g., Employee class).
Level 2:
Perform operations like sorting objects based on a field.

Week 11:

Lab Sheet 11: Strings and StringBuffer
Level 1:
Perform basic string operations: Length, concatenation, comparison
Level 2:
Demonstrate mutable operations using StringBuffer and StringBuilder.

Week 12:

Lab Sheet 12: Inheritance
Level 1:
Create a base class and derived class using single inheritance.
Level 2:
Implement multilevel inheritance and demonstrate use of super keyword.

Week 13:

Lab Sheet 13: Polymorphism & Abstract Classes
Level 1:
Demonstrate method overriding (runtime polymorphism).
Level 2:
Create an abstract class and implement its methods in a subclass.

Week 14:

Lab Sheet 14: Exception Handling
Level 1:
Handle exceptions using try-catch-finally.
Level 2:
Create custom exceptions and demonstrate multiple catch blocks.

Week 15:

Lab Sheet 15: File Handling and Serialization
Level 1:
Write a Java program to Create a file &Write and read data using File I/O
Level 2:
Implement object serialization and deserialization.
Use File streams and buffered streams for efficient I/O.

Week 15.1: AI/ML and Secure File Processing
Lab Sheet 16: Intelligent File Handling and Data Processing
Level 1:
Implement a Java program for secure file handling using buffered streams and file channels.
Level 2:
Develop a simple application for intelligent data analysis or log monitoring using Java File I/O and serialization concepts.

Targeted Application & Tools that can be used :JDK /Eclipse IDE/Visual Studio Code / net Beans IDE.

Text Book

1. Herbert Schildt, "The Complete Reference Java 2", Tata McGraw Hill Education, 11th Edition,2019.

References

1. Cay S Horstmann and Cary Gornell, "CORE JAVA volume I-Fundamentals", Tenth Edition, Pearson 2015.

2. James W. Cooper, "Java TM Design Patterns – A Tutorial", Addison-Wesley Publishers.4th Edition, 2000.

3. E. Balagurusamy, "Programming with Java", Tata McGraw Hill Education, 6th Edition, 2019.

E book link R1: http://rmi.yaht.net/bookz/core.java/9780134177373-Vol-1.pdf

E book link R2: Java(tm) Design Patterns: A Tutorial([PDF] [7qmsenjl97t0] (vdoc.pub)

Web resources
https://youtube.com/playlist?list=PLu0W_9lII9agS67Uits0UnJyrYiXhDS6q
https://puniversity.informaticsglobal.com:2229/login.aspx

Topics relevant to development of "Skill Development":
1. Static Polymorphism
2. Method overloading, constructors
3. constructor overloading

4. this keyword 5. static keyword and Inner classes 6. Inheritance and Polymorphism. This is attained through assessment component mentioned in course handout.

Course Code: CCS2257	Course Title: Computer Organization and Architecture Type of Course: Theory Course (PCC)	L-T-P- C	3-0-0-3
Version No.	2.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course introduces the core principles of computer architecture and organization from basic to intermediate level. This theory-based course emphasizes understanding the interaction between computer hardware and software. It equips students with the intuition behind assembly-level instruction set architectures and helps them interpret the operational concepts of computer technology and performance enhancement. The course also introduces emerging concepts in computer security, AI-enabled computing, intelligent processor optimization, and modern secure computing architectures.		
Course Objective	To develop a comprehensive understanding of computer system organization by enabling students to analyze processor architecture and instruction execution, explore memory hierarchy and cache mechanisms, examine input/output organization and interfacing, evaluate system performance and optimization techniques, and understand emerging applications in secure computing, AI-enabled architectures, and intelligent system design.		
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Describe the fundamental components of a computer system and their interconnections. — Remember **CO2:** Explain the concepts of Instruction Set Architecture (ISA) and memory organization. — Understand **CO3:** Apply techniques to perform arithmetic and logical operations in computer systems. — Apply **CO4:** Explain the organization and functioning of processor and memory subsystems. — Understand		
Course Content:			
Module 1	Basic Structure of Computer		12 Sessions

Topics:

Computer Types, Functional Units, Basic Operational concepts, Bus Structures, Computer systems RISC & CISC, Performance – Processor Clock, Basic Performance Equation, Clock Rate, Performance Measurement. Arithmetic Operations on Signed numbers. Instructions and Instruction Sequencing, Instruction formats,

Memory Instructions.

Module 2	Instruction Set Architecture and Memory Unit	12 Sessions

Topics:

Instruction Set Architecture: Addressing Modes, Stacks and Subroutines.

Memory System: Memory Location and Addresses, Memory Operations, Semiconductor RAM Memories, Internal Organization of Memory chips, Cache memory mapping Techniques.

Module 3	Arithmetic And Input/output Design	10 Sessions

Topics:

Arithmetic: Carry lookahead Adder, Signed-Operand Multiplication, Integer Division, and Floating point operations.

Input/output Design: Accessing I/O Devices, I/O communication, Interrupt Hardware, Direct Memory Access, Buses, Interface Circuits

Module 4	BPU and Pipelining	05 Sessions

Topics:

Basic Processing Unit: Fundamental Concepts, Single Bus organization, Control sequence, Execution of a Complete Instruction, Multiple Bus Organization.

Pipelining: Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, Hazards.

Module 5	Security, AI and Intelligent Computing Architecture	06 Sessions

Computer Security Fundamentals, Secure Instruction Execution, Memory Security and Protection Techniques, Cache and Processor Security, Cryptographic Fundamentals in Computer Architecture, AI-assisted Processor Design, Machine Learning Concepts in Hardware Optimization, Intelligent Memory Management Techniques, AI-based Performance Optimization, Applications of AI and Security in Modern Computing Systems.

Targeted Application & Tools that can be used:

Targeted employment sector is processor manufacturing and memory chip fabrication vendors like Intel, AMD, Motorola, NVidia, Samsung, Micron Technology, western Digital etc. Targeted job profiles include Memory circuit design and verification engineers, Physical system design engineer, System programmer, Fabrication engineer etc.

Tools:

Virtual Lab, IIT KGP Tejas – Java Based Architectural Simulator, IIT Delhi
Project work/Assignment:
Each batch of students (self-selected batch mates – up to 4 in a batch) will be allocated case studies/assignments
Textbook(s): 1. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, "Computer Organization", Sixth Edition, McGraw-Hill Higher Education, 2023 reprint. 2. William Stallings, "Computer Organization & Architecture – Designing for Performance", 11th Edition, Pearson Education Inc., 2019.
References 1. David A. Patterson & John L. Hennessy, "Computer Organization and Design MIPS Edition- The Hardware/Software Interface", 6th Edition, Morgan Kaufmann, Elsevier Publications, November 2020.
2. https://nptel.ac.in/courses/106105163 3. https://nptel.ac.in/courses/106106092 4. https://puniversity.informaticsglobal.com:2229/login.aspx

Course Code: CSS2264	**Course Title: Essentials of AI** **Type of Course: -Theory Course (ESC)**	**L-T-P-C**	**3-0-0-3**
Version No.	1.0		
Course Pre-requisites	CSS1500-Computational Thinking Using Python		
Anti-requisites	NIL		
Course Description	This course introduces foundational and practical Artificial Intelligence using modern platforms, prompt engineering, and engineering application frameworks. It maps real-world routing problems into computational state-spaces to implement classical uninformed (BFS, DFS) and heuristic-guided informed (A*, GBFS) search algorithms. The curriculum explores data engineering pipelines for text tokenization, sentiment pattern recognition, and similarity-driven recommendation workflows. It details the social, ethical, and economic impacts of automation while establishing paradigms for the responsible deployment of technologies. Concluding with integrated mini-projects, the course addresses advanced systems including chatbots, voice assistants, image recognition, and intelligent automation.		

Course Objectives	This course introduces the fundamentals of Artificial Intelligence, foundational AI tools, prompt engineering, and engineering application frameworks, alongside state-space problem reduction, computational graph mapping, uninformed blind searching, and heuristic-guided informed search algorithms. It explores text dataset preprocessing pipelines, pattern recognition models, similarity metrics, and recommendation system workflows, while addressing the social, ethical, and economic impacts of AI and strategies for the responsible deployment of modern technologies. Additionally, the course covers advanced AI fields including chatbots, voice assistants, image recognition, and intelligent automation through practical case studies and hands-on activities.
Course Outcomes	On successful completion of this course, students will be able to: **CO1:**Explain the Fundamental Concepts and Evolution of AI [Level 2 – Understand] **CO2:** Formulate and Implement Search Strategies for Computational Problem Solving. [Level 3 – Apply] **CO3:** Implement data collection and pre-processing workflows for different data types, and apply performance metrics to systematically evaluate the resulting AI modules. [Level 3 – Apply] **CO4:**Understand the social, ethical, and economic impacts of AI technologies and their influence on society, industries, and employment. [Level 2 – Understand] **CO5:** Analyze advanced AI technologies and their real-world applications through case studies and practical scenarios. [Level 4 – Analyze]

Course Content:

Module 1	**Introduction to Artificial Intelligence**	**Assignment (Fundamentals and Applications of Artificial Intelligence)**	**9 Sessions**

Topics:

Introduction to Artificial Intelligence – Definition and History of AI; Evolution of AI; Goals and Scope of AI; Types of AI – Narrow AI, General AI, Super AI; Applications of AI in Daily Life; AI in Engineering and Industry; Opportunities and Challenges of AI

Module 2	**Problem Solving & Path Searching**	**Practical Assignment (Problem Solving and Path Searching)**	**11 Sessions**

Topics:

Finding Answers: Translating a real-world problem into a clean "State-Space" map (e.g., maze navigation, puzzle solving).

Blind Searching: Uninformed search strategies—how computers search blindly using Breadth-First Search (BFS) and Depth-First Search (DFS).

Smart Searching: Informed search strategies—introducing "Heuristics" (smart guessing) through A* Search and Greedy Best-First Search

Module 3	**Data Collection and Pre-Processing**	**Assignment (Model Evaluation)**	**7 Sessions**
Topics: Foundations and the Role of Data in AI Systems-Taxonomy of Engineering Data: Structured, Unstructured, and Semi-Structured Data-Data Collection Methodologies and Acquisition Pipelines-Data Preparation Foundations: Cleaning, Imputation, and Noise Reduction-Data Manipulation, Transformation, and Feature Scaling-Data Augmentation Techniques for Unstructured Data			
Module 4	**Social, Ethical, and Economic Impacts of AI**	**Assignment (Social, Ethical, and Economic Impact Analysis of AI)**	**9 Sessions**
Topics: Social, Ethical, and Economic Effects of AI – Ethical issues in AI, future challenges and opportunities, impact of AI on the economy and employment, and responsible use of AI technologies.			
Module 5	**Advanced AI Technologies case study**	**Mini Project / Case Study (AI Technologies and Real-World Applications)**	**9 Sessions**
AI Technologies – Voice assistants, image recognition applications, AI chatbots Bias and fairness AI. AI and ML in Healthcare – Disease Prediction, Medical Imaging, Clinical Decision Support Systems, Personalized Healthcare.			

Targeted Application & Tools that can be used:

1. **Platform:**Google AI Experiments (Used to explore interactive Narrow AI benchmarks and live neural demonstrations).
2. **Visualizer:**Pathfinding Visualizer Repository (Used to simulate, trace, and view running animations of BFS, DFS, and A^* search algorithms).
3. **Model Trainer:**Teachable Machine by Google (Used to run no-code data pipelines, capture custom audio/image training sets, and deploy pattern recognition blocks).
4. **Fairness Engine:**IBM AI Fairness 360 (Used to test for algorithmic bias and examine data-driven fairness metrics).
5. **Ecosystem:**Hugging Face Models Hub (Used for deploying transformers, text summarization, sentiment analysis, and testing pre-trained NLP chatbots).

Textbook(s):

1. Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, 4th Edition, Pearson Education, 2022.
2. Tom Taulli, *Artificial Intelligence Basics: A Non-Technical Introduction*, 1st Edition, Apress, 2019.
3. Elaine Rich and Kevin Knight, *Artificial Intelligence*, 3rd Edition, McGraw Hill Education, 2009.

References:

1. Melanie Mitchell, *Artificial Intelligence: A Guide for Thinking Humans*, 1st Edition, Farrar, Straus and Giroux, 2019.
2. Kai-Fu Lee, *AI Superpowers*, 1st Edition, Houghton Mifflin Harcourt, 2018.
3. Ethem Alpaydin, *Introduction to Machine Learning*, 4th Edition, MIT Press, 2020

Weblinks

- **NPTEL Link:** * NPTEL: An Introduction to Artificial Intelligence (IIT Delhi Course Vault) – This course explicitly maps to the historical transitions and core architectural definitions of AI.
    - NPTEL Video Lectures: History and Scope of AI – Direct access to recorded lectures covering AI history from the 1940s to the present day.
- Google AI Experiments Ecosystem – A sandbox to experience real-time examples of specialized (Narrow) AI.
- **NPTEL Link:** * NPTEL: Artificial Intelligence Search Methods for Problem Solving (IIT Madras) – The absolute best resource for this module. Look at the specific lectures on *Search: DFS and BFS*, *Heuristic Search*, and *The $A^$* Algorithm*.
- Pathfinding Visualizer GUI Terminal – An interactive application to execute, track, and witness BFS, DFS, and A^* algorithms expanding over obstacles.
- **NPTEL Link:**
    - NPTEL: Data Science and Machine Learning Pipelines – Navigate down to the *Machine Learning, Data Processing, and Feature Learning Rules* chapters to match your syllabus framework.
- Teachable Machine by Google Platform – A no-code machine learning tool perfect for showing students how changing training datasets alters real-time classification decisions.
- **NPTEL Link:**
    - NPTEL Plus: AI Essentials and Social Frameworks Hub – Explores systemic oversight, data governance, and training ethics when launching AI products.
- IBM AI Fairness 360 Open Source Toolkit – A web portal used to mathematically study algorithmic biases and fairness metrics inside data columns.
- **NPTEL Link:**
    - Swayam Plus / NPTEL: Applied AI & ML Using Python Frameworks – This specialized track hosts practical mini-project guidelines for integrating smart applications across different domains (Healthcare, Agriculture, Cyber).
- Hugging Face Central Models Repository – Used to pull pre-trained transformers to run sentiment analysis, text summarization, and direct prompt-engineered chatbot interactions.

Course Code: CSS2265	**Course Title: Essentials of AI Lab** **Type of Course: LabCourse (PCC)**	**L-T-P-C**	**0-0-2-1**
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course introduces foundational and practical Artificial Intelligence using Python, modern IDEs, and tools like ChatGPT, Gemini, and Hugging Face. Students leverage prompt engineering to benchmark Narrow versus General AI, map structural state-spaces using networkx, and implement classical uninformed (BFS, DFS) and heuristic-guided informed (A*, GBFS) search algorithms. The curriculum covers data engineering pipelines for text tokenization and preprocessing, utilizing pre-trained NLP transformers for sentiment analysis and matplotlib for feature visualization. Students construct intelligent, interactive systems, including rule-based chatbots, content recommendation engines using vector similarity, and microphone-simulated		

	voice assistants. Evaluated via a dual-session framework that pairs conceptual exploration with active problem-solving, the course concludes with an integrated AI mini-project deploying a functional prototype for real-world applications.
Course Objectives	To develop practical skills in Artificial Intelligence by enabling students to explore foundation AI tools, apply prompt engineering, implement classical uninformed and heuristic-guided search algorithms, design data pipelines for preprocessing text datasets, apply pre-trained NLP models for sentiment analysis, utilize graphical visualization tools for feature distributions, develop interactive chatbot, recommendation, and voice-simulated applications, and construct integrated AI-driven prototypes for real-world engineering problems.
Course Outcomes	On successful completion of this course, students will be able to: **CO1:**Apply public AI platforms and structure precise text prompts to generate contextual, domain-specific engineering outputs. [Level 3 – Apply] **CO2:**Implement state-space models, uninformed searches (BFS, DFS), and heuristic-guided searches (A*, GBFS) to solve graph routing and navigation problems. . [Level 3 – Apply] **CO3:** Implement interactive computational models for rule-based chatbots, content recommendations, and voice command simulators. [Level 3 – Apply] **CO4:**Analyze raw datasets and model behaviors using automated text preprocessing, feature distribution matrices, and visual graphing techniques. [Level 4 – Analyze] **CO5:**Analyze the behavioral patterns, benchmarking metrics, and structural scope limitations of specialized vs. generalized AI architectures. [Level 4 – Analyze]
Course Content:	

Lab Sheet 1: Introduction to AI Platforms

- Context: Understanding the types, scope, and daily life applications of AI.
- Problem Statement: Explore different AI platforms such as ChatGPT, Gemini, and Hugging Face. Submit identical query prompts regarding historical AI milestones and daily life use cases, then compare their text outputs based on relevance, speed, and facts.

Lab Sheet 2: AI Classification and Benchmark Exploration

- Context: Breaking down the concepts of Narrow AI vs. General AI.
- Problem Statement: Interact with zero-code Narrow AI environments (e.g., Quick Draw or Google AI Experiments). Benchmark how a specialized computer vision network operates and write a brief analysis detailing why it cannot perform general-purpose tasks.

Lab Sheet 3: State-Space Representation and Graph Mapping

- Context: Translating a real-world problem into a clean "State-Space" map.
- Problem Statement: Take a physical layout problem (such as a 2D grid maze or a simple puzzle routing map) and write a Python script using the networkx library to map it out computationally as a structural state-space network of nodes and edges.

Lab Sheet 4: Uninformed (Blind) Searching Implementation

- **Context:** (Using the state-space map designed in Lab 4) — Implementing a layer-by-layer blind search strategy.
- **Problem Statement:** Write a Python program to execute a **Breadth-First Search (BFS)** algorithm on your graph environment. Given a starting node and a target destination node, use a **Queue (FIFO)** data structure to explore the graph layer by layer. Display the exact step-by-step node traversal order and the final shortest path found.

Lab Sheet 5: Uninformed (Blind) Searching Implementation

- Context: (Using the state-space map designed in Lab 4) — Implementing a deep-branch blind search strategy.
- Problem Statement: Write a Python program to execute a Depth-First Search (DFS) algorithm on your graph environment. Given a starting node and a target destination node, use a Stack (LIFO) data structure to explore deep along each branch before backtracking. Display the exact step-by-step node traversal order and the final path found.

Lab Sheet 6: Informed (Smart) Searching with Heuristics

- Context: — Introducing Heuristics through A* Search and Greedy Best-First Search.
- Problem Statement: Implement the A* Search algorithm on your coordinate-based graph or grid map. Define a straight-line Euclidean distance heuristic to guide the pathfinder and compare its path efficiency and node-expansion count against uninformed blind searches.

Lab Sheet 7: Informed Search – Greedy Best-First Search (GBFS)

- **Objective:** To implement a heuristic-guided, smart-guessing optimization algorithm to rapidly navigate towards a target node.

- **Problem Statement:** Write a simple Python program to execute a **Greedy Best-First Search (GBFS)** pathfinding algorithm across a graph network environment. Given a starting node and a target destination, create a lookup table mapping heuristic values (estimated straight-line distance to the target) for each vertex. At every step, implement a greedy loop that evaluates neighbours and moves directly into the unvisited node that possesses the lowest (best) heuristic score.

Lab Sheet 8: Dataset Collection & Preprocessing

- **Context:** Understanding the role of data, types of data, and processing pipelines.
- **Problem Statement:** Collect raw text-based datasets (such as product reviews or short logs from Kaggle). Write a Python script to preprocess the data by performing basic tokenization, lowercase conversion, and removing trivial characters..

Lab Sheet 9: Tabular Data Engineering Pipeline: Cleaning, Imputation, Text Parsing, and Feature Scaling of Corrupted Cyber-Physical Streams.

Lab Sheet 10: Spatial Engineering Pipelines: Image Noise Mitigation, Gaussian Filtering, and Geometric Data Augmentation.

Lab Sheet 11: Foundational Data Pre-Processing: Data Taxonomy, Handing Outliers, Mean Imputation, and Min-Max Feature Scaling**. Lab Sheet 12: AI Chatbot Development**

- **Problem Statement:**
- Develop a rule-based chatbot using Python and integrate contextual AI-generated responses for interactive communication.

Lab Sheet 13:: Implementing a Hugging Face Pipeline for Automated Plant Disease Diagnostics using**Python**

Lab Sheet 14: Voice Assistant Simulation

- **Problem Statement:**
- Develop a basic voice assistant using Python speech libraries and implement voice commands for search, greetings, and simple automation tasks.

Lab Sheet 15: AI Mini Project

- **Problem Statement:**
- Develop and demonstrate an AI-based application integrating concepts such as NLP, machine learning, chatbot systems, or recommendation systems.

Targeted Application & Tools that can be used:

1. ChatGPT, Gemini, Hugging Face, Canva AI, Python
2. Python IDEs like PyCharm-Jupyer Notebook /Google Collab

Project work/Assignment: Mention the Type of Project /Assignment proposed for this course

The course is a lab-based course with all the assessments centrally evaluated. Every experiment consists of **two sessions**. The first session involves exploring a solution to the problem. The second session involves solving a particular problem.

Textbook(s):

1. Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, 4th Edition, Pearson Education, 2022.
2. Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, and Vipin Kumar, *Introduction to Data Mining*, 2nd Edition, Pearson Education, 2019.

3. Lewis Tunstall, Leandro von Werra, and Thomas Wolf, *Natural Language Processing with Transformers*, Revised Edition, O'Reilly Media, 2022.

References:

1. Elaine Rich, Kevin Knight, and Shivashankar B. Nair, *Artificial Intelligence*, 3rd Edition, McGraw Hill Education, 2009.
2. Jake VanderPlas, *Python Data Science Handbook: Essential Tools for Working with Data*, 2nd Edition, O'Reilly Media, 2022.

Course Code: **APT4002**	**Course Title:** **Introduction to Aptitude (Audited)**	**L- T-P- C**	**0-0-2-0**
Version No.	1.0		
Course Pre-requisites	Students should know the basic Mathematics & aptitude along with understanding of English		
Anti-requisites	Nil		
Course Description	The objective of this course is to prepare the trainees to tackle the questions on various topics and various difficulty levels based on Quantitative Ability, and Logical Reasoning asked during the placement drives. There will be sufficient focus on building the fundamentals of all the topics, as well as on solving the higher order thinking questions. The focus of this course is to teach the students to not only get to the correct answers, but to get there faster than ever before, which will improve their employability factor.		
Course Objective	The objective of the course is to familiarize the learners with the concepts of Aptitude and attain Skill Development through Problem Solving techniques.		
Course Outcomes	On successful completion of this course, students will be able to: CO1: Recall all the basic mathematical concepts learnt in high school. — Level 1 – Remember CO2: Identify the principal concept needed in a question. — Level 2 – Understand CO3: Solve quantitative and logical ability questions using appropriate concepts. — Level 3 – Apply CO4: Analyze the data given in complex problems. — Level 4 – Analyze CO5: Rearrange the information to simplify the question. — Level 4 – Analyze		

Course Content:				
Module 1	Quantitative Ability	Assignment	Bloom's Level : Application	**12 Hours**
Topics: Introduction to Aptitude, working of Tables, Squares, Cubes				

Module 2	Logical Reasoning	Assignment	Bloom's Level : Application	**18 Hours**
Topics: Linear & Circular Arrangement Puzzle, Coding & Decoding, Blood Relations, Directions, Ordering and Ranking, Clocks and Calendars, Number Series, Wrong number series, Visual Reasoning				
Targeted Application & Tools that can be used: Application area: Placement activities and Competitive examinations. Tools: LMS				
Text Book 1. Quantitative Aptitude by R S Aggarwal 2. Verbal & Non-Verbal Reasoning by R S Aggarwal				
References 1. www.indiabix.com 2. www.youtube.com/c/TheAptitudeGuy/videos				
Topics relevant to Skill development: Quantitative and reasoning aptitude for Skill Development through Problem solving Techniques. This is attained through the assessment component mentioned in course handout.				

Course Code: CIV7601	Course Title: Universal Human Values and Ethics Type of Course: MAC course	L-T-P-C	1-0-0-0
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	The purpose of the course is to develop a holistic perspective in students' life. The course adopts a self-reflective methodology of teaching and is designed to equip the students to explore their role in all aspects of living as a part of the society. It presents a universal approach to value education by developing the right understanding of reality through the process of self-exploration. This self-exploration develops more confidence and commitment in students enabling them to critically evaluate their pre-conditioning and present beliefs. As an outcome of the holistic approach, the students will be able to practice the ethical conduct in the social and professional life. The prime focus throughout the course is toward affecting a qualitative transformation in the life of the student rather than just a transfer of information. This course is designed to cater to Human Values and Professional Ethics.		
Course Objective	The objective of the course is 'SKILL DEVELOPMENT' of the student by using 'SELF LEARNING' techniques		

Course Outcomes	On successful completion of this course the students shall be able to: CO1: Understand the concepts of human values, self-exploration, and holistic development in personal and social contexts. [Level 2 – Understand] CO2: Apply principles of human values and ethical conduct in personal, social, and professional life situations. [Level 3 – Apply] CO3: Apply the concepts of harmony and ethical reasoning to address issues in family, society, and professional environments. [Level 3 – Apply] CO4: Analyze the relationship between self and body, and the role of values in maintaining harmony within individuals and society. [Level 4 – Analyze] CO5: Evaluate personal beliefs, behaviors, and professional practices based on universal human values and ethical principles. [Level 5 – Evaluate]			
Course Content:				
Module 1	Introduction to Value Education	Online Assessment	MCQ Quiz	5 Sessions
Topics: Right Understanding, Relationship and Physical Facility (Holistic Development and the Role of Education), Understanding Value Education, Self-exploration as the Process for Value Education, Continuous Happiness and Prosperity – the Basic Human Aspirations, Happiness and Prosperity – Current Scenario, Method to Fulfil the Basic Human Aspirations.				
Module 2	Harmony in the Human Being	Online Assessment	MCQ Quiz	5 Sessions
Topics: Understanding Human being as the Co-existence of the Self and the Body, Distinguishing between the Needs of the Self and the Body, The Body as an Instrument of the Self, Understanding Harmony in the Self, Harmony of the Self with the Body, Programme to ensure self-regulation and Health				
Module 3	Harmony in the Family and Society	Online Assessment	MCQ Quiz	5 Sessions
Topics: Harmony in the Family – the Basic Unit of Human Interaction, 'Trust' – the Foundational Value in Relationship, 'Respect' – as the Right Evaluation, Other Feelings, Justice in Human-to-Human Relationship, Understanding Harmony in the Society, Vision for the Universal Human Order.				
Module 4	Implications of the Holistic Understanding – A Look at Professional Ethics	Online Assessment	MCQ Quiz	5 Sessions
Topics: Natural Acceptance of Human Values, Definitiveness of (Ethical) Human Conduct, A Basis for Humanistic Education, Humanistic Constitution and Universal Human Order, Competence in Professional Ethics, Holistic Technologies, Strategies for Transition towards Value-based Life and Profession				
Targeted Application & Tools that can be used: Application areas are Personal life, Education and Career, Workplace , Society and Environmental Responsibility Tools: Online Tools – NPTEL and Swayam.				
Project work/Assignment:				

Assessment Type

- Online exams (MCQs) will be conducted by the Department of Civil Engineering through Linways.

Online Link*:

1) UHV II - https://www.youtube.com/watch?v=NhFBzn5qKIM&list=PLWDeKF97v9SO8vvjC1KyqteziTbTjN1So&pp=0gcJCWMEOCosWNin
2) Lecture by Dr. Kumar Sambhav, NPTEL course: Universal Human Values, https://onlinecourses.swayam2.ac.in/aic22_ge23/preview
3) Lecture by Dr. Padmavati, Dr Narendran Thiruthy, NPTEL Course: Biodiversity Protection, Farmers and Breeders Rights, https://nptel.ac.in/courses/129105008, 2024.

* Other source links are available in below Resources link.

Text Book

1. A Foundation Course in Human Values and Professional Ethics, R R Gaur, R Asthana, G P Bagaria, 2nd Revised Edition, Excel Books, New Delhi, 2019. ISBN 978-93-87034-47-1
2. Human Values, A.N. Tripathi, New Age Intl. Publishers, New Delhi, 2019.
3. Premvir Kapoor, Professional Ethics and Human Values, Khanna Book Publishing, New Delhi, 2022.

Reference Books

1. E.F. Schumacher, 1973, Small is Beautiful: a study of economics as if people mattered, Blond & Briggs, Britain.
2. Sussan George, 1976, How the Other Half Dies, Penguin Press. Reprinted 1986.
3. Donella H. Meadows, Dennis L. Meadows, Jorgen Randers, William W. Behrens III, 1972, Limits to Growth – Club of Rome's report, Universe Books.
4. A Nagraj, 1998, Jeevan Vidya Ek Parichay, Divya Path Sansthan, Amarkantak.
5. P L Dhar, RR Gaur, 1990, Science and Humanism, Commonwealth Publishers.
6. A N Tripathy, 2003, Human Values, New Age International Publishers.
7. E G Seebauer& Robert L. Berry, 2000, Fundamentals of Ethics for Scientists & Engineers, Oxford University Press
8. M Govindrajran, S Natrajan& V.S. Senthil Kumar, Engineering Ethics (including Human Values), Eastern Economy Edition, Prentice Hall of India Ltd.
9. B P Banerjee, 2005, Foundations of Ethics and Management, Excel Books.
10. William P. Cunningham and Mary Ann Cunningham (2020), Principles of Environmental Science: Inquiry & Applications, 9th Edition, McGraw-Hill Education, USA.

Resources:

1. https://onlinecourses.swayam2.ac.in/imb25_mg195/preview
2. https://onlinecourses.nptel.ac.in/noc25_mg141/preview
3. https://onlinecourses.swayam2.ac.in/ini25_hs52/preview
4. https://onlinecourses.nptel.ac.in/noc25_hs219/preview
5. https://onlinecourses.swayam2.ac.in/cec25_mg14/preview
6. https://onlinecourses.swayam2.ac.in/imb25_mg195/preview
7. https://onlinecourses.swayam2.ac.in/imb25_mg196/preview

Topics relevant to Skill Development:

1. An attitude of enquiry.
2. Write reports

The topics related to Human values and Professional ethics: All topics in are relevant to Human values and Professional ethics.

Course Code: **MAT2321**	**Course Title:Discrete Mathematical Structures** **Type of Course:1] School Core**	**L-T- P- C**	**3**	**1**	**0**	**4**
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-requisites	NIL					
Course Description	This course introduces the fundamental concepts of Discrete Mathematics, providing the mathematical foundations essential for computer science and related disciplines. It covers sets, relations, functions, closures, equivalence relations, and primitive recursive functions, followed by partially ordered sets, lattices, and Boolean algebras. The course also explores graph theory, including graph types, isomorphism, Euler and Hamiltonian paths, circuits, and graph coloring. In addition, students study algebraic structures such as semigroups, monoids, and groups, residue arithmetic, and group codes. Emphasis is placed on developing logical reasoning, mathematical modeling, and problem-solving skills applicable to computing, information systems, and network analysis.					
Course Objective	The primary objective of this course is to provide students with a strong foundation in discrete mathematical structures and logical reasoning that underpin computer science and related disciplines. It aims to develop the ability to construct mathematical arguments, analyze discrete problems, and apply formal methods to problem solving. The course introduces fundamental concepts of logic, sets, relations, functions, Boolean algebra, graph theory, and algebraic structures such as semigroups, monoids, and groups, while also exploring formal languages and grammar. Through these topics, students will acquire the analytical and mathematical skills necessary for applications in algorithm design, digital systems, coding theory, cryptography, and other areas of computing.					
Course Outcomes	On successful completion of the course the students shall be able to: **CO1:** Apply propositional and predicate logic, logical equivalences, normal forms, inference rules, and proof techniques to analyze arguments and solve problems in mathematical reasoning and computer science. (L3) **CO2:** Apply the concepts of sets, relations, functions, closures, equivalence relations, and primitive recursive functions to model and solve problems in discrete mathematics and computer science. (L3) **CO3:** Employ partially ordered sets, lattices, Boolean algebras, and Boolean functions, and apply their properties to solve problems in discrete mathematical structures and logical systems. (L3) **CO4:** Apply the concepts of lattices, Boolean algebras, and Boolean functions to solve logical and computational problems. (L3)					

	CO5: Apply graph theoretic concepts and techniques to solve problems in networks, connectivity, and optimization applications. (L3) **CO6:** Employ group theory and residue arithmetic techniques in coding, computation, and information processing. (L3)	
Course Content:		
Module 1	**Fundamentals of Logic**	**(11L+4T=14 Sessions)**
Propositional Logic, Propositional Logic, Equivalences, Normal forms, Inference rules, Introduction to Proofs, Conversion to clausal form, Predicate calculus, The Statement function, Inference theory of the Predicate Calculus. Compiler optimization and program correctness verification, Cybersecurity and cryptographic protocol verification.		
Module 2	**Relations and Functions**	**(10L+3T=13 Sessions)**
Sets and set-operations, Relations, Functions, Properties of Relations & Representations of Relation, closure of different types of relations, equivalence relations, primitive recursive function. Database Management Systems, Artificial Intelligence and Machine Learning.		
Module 3	**Boolean Algebra and Graph Theory**	**(13L+4T=17 Sessions)**
Partially Ordered Sets, Lattices, Boolean Algebras, Functions of Boolean Algebras, Boolean Functions as Boolean Polynomials. Computer Networks: Routing and switching logic. Robotics: Decision-making using Boolean logic. Graph Theory-Basic Concept of Graph Theory, Types of Graphs, Isomorphism, Euler Paths and Circuits, Hamiltonian Paths and Circuits, Colouring of Graph. Social Networks (Facebook, Instagram, LinkedIn).and GPS & Navigation: Shortest path and route optimization.		
Module 4	**Groups**	**(11L+4T=15 Sessions)**
Semi Groups, Monoids, Groups, The applications of Residue Arithmetic to Computers, Group codes. Functional programming and distributed systems. Compiler design and programming languages.		
Targeted Application & Tools that can be used: • Compiler optimization, program correctness verification, cybersecurity protocol verification, automated reasoning, formal software verification. • Database Management Systems (DBMS), Artificial Intelligence (AI), Machine Learning (ML), knowledge representation, data modeling and query processing. • Digital circuit design, computer networks (routing and switching), robotics decision-making systems, logic controller design, social network analysis, GPS and navigation systems, route optimization, communication networks, and recommendation systems. • Cryptography, error-correcting codes, compiler design, distributed systems, residue arithmetic in computer architecture.		

Software: MATLAB, Lean 4, Mathematica, Maple, Python, GeoGebra, Octave, R, Scilab.

Assignment:

Assignments will involve theoretical problem-solving and computational implementation using suitable programming tools. The assignments may be undertaken individually or in groups, as decided by the instructor. Additionally, experiential learning activities will be conducted using Lean 4, enabling students to develop formal proofs, verify logical propositions, and gain hands-on experience in theorem proving and formal verification.

Text Book

6. K. H. Rosen, *Discrete Mathematics and Its Applications*, 8th ed. New York, NY, USA: McGraw-Hill Education, 2019.
7. F. Harary, *Graph Theory*. Reading, MA, USA: Addison-Wesley Publishing Company, 1969.

References:

1. S. Epp, *Discrete Mathematics with Applications*, 5th ed. Boston, MA, USA: Cengage Learning, 2019.
2. K. A. Ross and C. R. B. Wright, *Discrete Mathematics*, 6th ed. New Delhi, India: Pearson Education, 2018.
3. L. Mott, A. Kandel, and T. P. Baker, *Discrete Mathematics for Computer Scientists and Mathematicians*, 3rd ed. New Delhi, India: PHI Learning, 2016.

E-resources/ Web links:

3. https://presiuniv.knimbus.com/portal/v2/default/viewDetail?searchResultType=ECATALOGUE_BASED&uniqueId=TEXTBOOK_LIBRARY01_06082022_289
4. https://presiuniv.knimbus.com/portal/v2/default/viewDetail?searchResultType=ECATALOGUE_BASED&uniqueId=DOAB_1_06082022_8919
5. https://nptel.ac.in/courses/106108227
6. https://nptel.ac.in/courses/111106102
7. https://nptel.ac.in/courses/111106113

Topics relevant to SKILL DEVELOPMENT: The course develops skills in logical reasoning, mathematical modeling, problem-solving, algorithmic thinking, and graph analysis through the study of logic, relations, Boolean algebra, and graph theory. It also enhances computational and analytical skills by applying algebraic structures, formal languages, recursion, and coding techniques to real-world computing problems.

Catalogue prepared by	**Dr. Obbu Ramesh**
Recommended by the Board of Studies on	
Date of Approval by the Academic Council	

Course Code: CSS2260	Course Name: Database Management Systems Type of Course: Theory Course	L- T-P- C	3-0-0-3
Version No.	1.0		
Course Pre-requisites	NIL		
Co-requisites			
Anti-requisites	NIL		
Course Description	This course introduces the foundational principles of database management systems, including data models, schemas, and architecture. This course provides a solid foundation on the relational model of data and the use of relational algebra. It develops skills in SQL for data definition, manipulation, and control, enabling students to construct and execute complex queries. The course also introduces the concept of object oriented and object relational databases and modern database technologies like NoSQL. The also course allows the students to gain insights into data storage structures and indexing strategies for optimizing query performance.		
Course Objective	The objective of the course is to familiarize the learners with the concepts of Database Management Systems and attain Employability through Problem Solving Methodologies.		
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Describe the fundamental elements of relational database management systems. [Level 2 - Understand] **CO2**: Examine databases using SQL query processing and Optimization. [Level 3 – Apply] **CO3:** Design simple database systems applying the normalization constraints and demonstrate the database transaction processing, recovery, and security. [Level 3 – Apply] **CO4:** Interpret the concept of advanced databases and its applications. [Level 3 – Apply]		
Course Content:			
Module 1	Introduction to Database Modelling and Relational Algebra	Assessment: Assignment (Problem Solving)	10 Sessions
Topics: Introduction to Database: Schema, Instance, 3-shema architecture, physical and logical data independence, Data isolation problem in traditional file system, advantages of database over traditional file systems. Entity Relationship (ER) Model, ER Model to Relational Model, Examples on ER model. Relational Algebra with selection, projection, rename, set operations, Cartesian product, joins (inner and outer joins), and division operator. Examples on Relational Algebra Operations.			

Module 2	Fundamentals of SQL and Query Optimization	Assessment : Assignment (Programming)	11 Sessions
Topics: SQL Database Querying, DDL, DML, Constraints, Operators, Set Operators, Aggregate Functions, Joins, Views, Procedures, Functions and Triggers. Database programming issues and techniques: Embedded SQL, Dynamic SQL; SQL / PSM and NoSQL. Query Optimization: Purpose, transformation of relational expressions, estimating cost and statistics of expression, choosing evaluation plans, linear and bushy plans, dynamic programming algorithms.			
Module 3	Relational Database Design & Transaction Management	Assessment: Assignment (Problem Solving)	12 Sessions
Topics: Relational database design: Problems in schema design, redundancy and anomalies, Normal Forms based on Primary Keys-(1NF,2NF, 3NF), Boyce-Codd Normal Form, Multi valued Dependency (Fourth Normal Form), Join Dependencies (Fifth Normal Form), lossy and lossless decompositions, Database De-normalization. Transaction Management: The ACID Properties; Transactions and Schedules; Concurrent Execution of Transactions; Lock- Based Concurrency Control; Performance of locking; Transaction support in SQL; Introduction to crash recovery; 2PL, Serializability and Recoverability; Lock Management; The write-ahead log protocol; Check pointing; Recovering from a System Crash; Media Recovery; Other approaches and interaction with concurrency control.			
Module 4	Advanced DBMS Topics	Assessment : Assignment (Case Study)	12 Sessions
Topics: Advanced topics: Object oriented database management systems, Deductive databasemanagement systems, Spatial database management systems, Temporal database management systems, Constraint database management systems. New database applications and architectures such as Data warehousing, Multimedia, Mobility, NoSQL, Native XML databases (NXD), Document-oriented databases, Statistical databases.			
Targeted Applications & Tools: Application Area: Relational database systems for Business, Scientific and Engineering Applications. Tools/Simulator used: MySQL DB for student practice. Also demonstration of ORACLE DB on object-relational database creation and JDBC connection.			
1. Problem Solving: Constructing ER-Diagrams for a given real time requirements, Normalizing the databases, querying the databases using relational algebra. 2. Programming: Implementation of any given scenario using MySQL.			
Skill-Building Activities through Experiential Learning:			

Text Book **T1:** Elmasri R and Navathe S B, "Fundamentals of Database System", Pearson Publication, 7th Edition, 2018. **T2:** RamaKrishna& Gehrke, "Database Management Systems" 3rd Edition, 2018, McGraw-Hill Education. **T3:** W. Lemahieu, S. vandenBroucke and B. Baesens, "Principles of Database Management: Practical Guide to Storing, Managing and Analyzing Big and Small Data", Cambridge University Press, 2018.
References **R1:** Avi Silberschatz, Henry F. Korth, S. Sudarshan, "Database System Concepts", McGraw-Hill ,7th Edition, 2019. **R2:** M. Kleppmann, "Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems", O'Reilly, 2017.
E-Resources/Web Resources

Course Code: CSS2261	Course Name: Database Management Systems Lab Type of Course: Laboratory Course	L- T-P- C	0-0-2-1
Version No.	1.0		
Course Pre-requisites	NIL		
Co-requisites			
Anti-requisites	NIL		
Course Description	The Database Management Systems (DBMS) Laboratory is designed to provide students with hands-on experience in database design, implementation, and management using SQL and database management tools such as MySQL. The lab complements theoretical concepts learned in database courses by allowing students to practice database creation, querying, and optimization techniques. The DBMS Lab enables students to develop industry-relevant skills in database management, preparing them for careers in software development, data engineering, and database administration.		
Course Objective	The objective of the course is to familiarize the learners with the concepts of Database Management Systems and attain Employability through Problem Solving Methodologies.		
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Demonstrate the database concepts, practice, and SQL queries. [Level 3 – Apply] **CO2:** Design and implement database schemas while applying normalization techniques to optimize structure. [Level 3 – Apply] **CO3:** Develop and implement stored procedures, triggers, and views for automation and efficiency. [Level 3 – Apply] **CO4:** Design and build database applications for real [Level 3 – Apply]		

	world problems.

Course Content:

List of Laboratory Tasks:

Create Employee, Student, Banking and Library databases and populate them with required data. Do the following experiments of different lab sheets on those databases.

Labsheet-1 [3 Practical Sessions]

Experiment No 1: [1 Session]

1. To study and implement the different language of Structured Query Language.

Level 1: Perform operations using Data Definition Language and Data Manipulation Language commands including different variants of SELECT on Student DB.

Level 2: Identify the given requirements; valid attributes and data types and Perform DDL and DML operations on a given scenario. [Banking Databases]

Experiment No. 2: [2 Sessions]

2. To study and implement the concept of integrity constraints in SQL.

Level 1: Create tables on Banking database using PRIMARY KEY, NOT NULL, UNIQUE, FOREIGN KEY and demonstrate the working of relational, logical, pattern matching, BETWEEN, IS NULL, IN and NOT IN Special Operators on Student Database.

Level 2: Enforce different types of data and referential integrity constraints. Then try queries with special operators based on the student database. [Banking Database].

Labsheet-2 [3 Practical Sessions]

Experiment No. 3: [1 Session]

3. mplement complex queries in SQL.

Level 1: Implement the conjugate of GROUP BY, ORDER BY and aggregate functions on Banking Database. Level 2: Implement MySQL DB queries on library database using appropriate clauses and aggregate functions. Also order the data either in ascending and descending order using corresponding clause. [Library databases].

Experiment No. 4: [2 Session]

4. To study and implement different types of Set and Join Operations [2 Slots]

Level 1: Demonstrate different types of Set Operations (UNION, UNION ALL, INTERSECT, MINUS) and Join Operations (INNER JOINs, OUTER JOINS, CROSS JOIN, NATURAL JOIN) on two or more tables of Airline Database. Level 2: Use Set and Join operations to retrieve the data from two or more relations(tables) as per the given

scenario. [Airline Database]

Labsheet-3 [2 Practical Sessions]

Experiment No. 5: [2 sessions]

5. To study and implement Views, and Procedures in MySQL DB.

Level 1: Implement MySQL Views, and Procedures in ORACLE DB on Employee database.

Level 2: Analyze the requirement and construct views, and Procedures on Mini Project Domain. [Banking Database]

Labsheet-4 [2 Practical Sessions]

Experiment No. 6: [2 Sessions]

6. To study and implement Functions, and Triggers in MySQL DB.

Level 1: Implement Oracle Functions and Triggers in Oracle on Employee database.

Level 2: Analyze the requirement and construct Functions and Triggers. [Supply chain Database]

Labsheet-5 [2 Practical Sessions]

Experiment No. 7: [2 Sessions]

7. To study and implement Functions, and Triggers in MySQL DB.

Level 1: Implement Oracle Functions and Triggers in Oracle on Employee database.

Level 2: Determine the requirement and construct Functions and Triggers. [Supply chain Database]

Labsheet-6 [4 Practical Sessions]

Experiment No. 8: [2 Sessions]

8. To implement the concept of forms and reports.

Level 1: Implement the concept of forms and reports.

Level 2: Examine the schema relationship.

Experiment No. 9: [2 Sessions]

9. Create the database using the given schema. (Flight Management)

Level 1: Implement a relational database based on the provided schema for the Flight Management system, including the creation of tables, relationships, and constraints.

Level 2: Demonstrate schema relationships by defining primary and foreign keys to ensure data integrity within the Flight Management database.

Labsheet-7 [4 Practical Sessions]

Experiment No. 10: [2 Sessions]

10. Create the database using the given schema. (Company database)

Level 1: Implement the database schema by defining tables, relationships, and constraints according to the given Company Database schema.

Level 2: Demonstrate the schema's relationships and data integrity by creating and linking tables as per the specified requirements.

Experiment No. 11: [2 Sessions]

11. Create the database using the given schema. (Student Library)

Level 1: Implement forms and reports based on the provided Student Library database schema, ensuring effective data entry and reporting mechanisms.

Level 2: Demonstrate the schema relationships within the Student Library database, demonstrating how these relationships influence the creation and functionality of forms and reports.

Labsheet-8 [1 Sessions]

12. Design a mini project based on the databases such as Inventory Management System, University Management System, Hospital Management System, etc.

Level 1: Implement the real time database.

Level 2: Analyze the working of database in real time.

Targeted Applications & Tools:

Skill-Building Activities through Experiential Learning:

Text Book

T1: Elmasri R and Navathe S B, "Fundamentals of Database System", Pearson Publication, 7th Edition, 2018.
T2: RamaKrishna& Gehrke, "Database Management Systems" 3rd Edition, 2018, McGraw-Hill Education.
T3: W. Lemahieu, S. vandenBroucke and B. Baesens, "Principles of Database Management: Practical Guide to Storing, Managing and Analyzing Big and Small Data", Cambridge University Press, 2018.

References

R1: Avi Silberschatz, Henry F. Korth, S. Sudarshan, "Database System Concepts", McGraw-Hill ,7th Edition, 2019.
R2: M. Kleppmann, "Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems", O'Reilly, 2017.

E-Resources/Web Resources

Course Code: CAI2500	Course Name: Machine Learning Type of Course: Theory Course	L- T-P- C	3-0-0-3
Version No.	1.0		
NCrF Level-	5.0		
Course Pre-requisites	NIL		
Co-requisites	CAI2501		
Anti-requisites	CSEXXXX		
Course Description	This course builds a practical and conceptual foundation in machine learning through supervised, unsupervised, and ensemble techniques. It explores neural networks and perceptron models while emphasizing hands-on implementation. Using tools like Keras and Scikit-learn, it focuses on model development, dataset analysis, and performance evaluation across diverse real-world applications.		
Course Objective	This course is designed to improve the learners 'EMPLOYABILITY SKILLS' by using EXPERIENTIAL LEARNING techniques.		
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Apply the fundamentals of machine learning and supervised learning algorithms. [Level 3 - Apply] **CO2**: Apply probabilistic models such as Naïve Bayes for supervised learning [Level 3 – Apply] **CO3:** Apply ensemble methods like bagging, boosting, and random forests to improve model accuracy. [Level 3 – Apply] **CO4:** Apply association rule learning algorithms to identify frequent patterns [Level 3 – Apply] **CO5:** Implement clustering techniques for unsupervised learning tasks. [Level 3 – Apply]		
Course Content:			
Module 1	Foundations of Machine Learning and Supervised Models	Assessment: Assignment (Programming using Keras/Sklearn)	9 Sessions
Topics: Introduction to Machine Learning: Definitions, Workflow, and Categories (Supervised, Unsupervised, Reinforcement) Data Preprocessing and Feature Engineering: Types of Features, Missing Data Handling, Encoding Methods Regression Techniques: Simple & Polynomial Regression, Loss Functions: MSE, MAE, RMSE Classification Algorithms: Logistic Regression, Softmax Regression with Cross-Entropy Loss Evaluation Techniques: Train/Test/Validation Split, Cross-Validation, Confusion Matrix, Precision, Recall, F1			

Score Hands-on: Build regression and classification models on real datasets			
Module 2	Probabilistic Models and Support Vector Machines	Assessment : Quiz + Assignment (Sklearn/Statsmodels)	9 Sessions
Topics: Bayesian Inference in ML: Bayes' Theorem, Conditional Probabilities Naïve Bayes Classifier: Gaussian, Multinomial, Bernoulli Bayesian Belief Networks Support Vector Machines (SVM): Soft Margin Classifiers, Kernel Methods: Polynomial, RBF, Sigmoid, Hyperparameter Tuning and Model Evaluation Hands-on: Naïve Bayes and SVM Implementation with sklearn			
Module 3	Ensemble Learning and Model Aggregation	Assessment : Assignment (Programming using Keras/Sklearn)	9 Sessions
Topics: Bias-Variance Tradeoff and the Need for Ensembles Bagging & Pasting: Random Forests and Bootstrapping Feature Subsets: Random Patches & Subspaces Voting Classifiers: Hard vs Soft Voting Boosting Techniques: AdaBoost , Gradient Boosting (GBM, XGBoost basics) , Extremely Randomized Trees , Stacking and Meta-Learning Hands-on: Implement ensemble models and compare performance			
Module 4	Neural Networks and Deep Learning Fundamentals	Assessment : Programming Assignment (Keras/TensorFlow)	9 Sessions
Topics: Biological Inspiration to Perceptrons Linear Threshold Units and Logic Gates Activation Functions: Sigmoid, Tanh, ReLU, Leaky ReLU, Softmax Loss Functions: Binary Cross-Entropy, Categorical Cross-Entropy, MSE Multi-layer Perceptrons (MLPs): Architecture and Use Cases			

Backpropagation and Gradient Descent (SGD, Adam)

Hands-on: Building and Training Simple MLP using Keras

Module 5	Unsupervised Learning and Anomaly Detection	Assessment : Mini Project (Programming using Sklearn/Keras)	9 Sessions

Topics:

Clustering: K-Means and Mini-Batch K-Means, Evaluation: Elbow Method, Silhouette Score, K-Means++, MST-based Clustering

Hierarchical Clustering: Agglomerative and Divisive

Competitive Learning: Kohonen's SOM

Density-Based Clustering: DBSCAN

Probabilistic Clustering: Gaussian Mixture Models (GMM) with EM

Outlier and Anomaly Detection: Isolation Forest, LOF.

Hands-on: Cluster Analysis and Outlier Detection on Public Datasets

Targeted Applications & Tools:

Python (NumPy, pandas)	Core programming language and libraries used for data manipulation, cleaning, and preprocessing throughout the course.
Scikit-learn (sklearn)	Primary library for implementing supervised learning (regression, classification, SVM), ensemble methods, clustering, and model evaluation metrics.
Keras (with TensorFlow backend)	Building and training neural networks such as perceptrons and multi-layer perceptrons (MLPs), and implementing backpropagation efficiently.
Matplotlib & Seaborn	Visualizing data distributions, decision boundaries, model performance (e.g., confusion matrix, ROC curves), and clustering outcomes.
Jupyter Notebook / Google Colab	Interactive environments for implementing, testing, and documenting machine learning experiments with built-in visualization and hardware acceleration.

Skill-Building Activities through Experiential Learning:

- Comparative Analysis of Ensemble Techniques
  Learners experiment with ensemble models like Random Forest, AdaBoost, and Gradient Boosting, conducting comparative studies on model variance, bias, and robustness. Through group assignments, students simulate real-world decision-making scenarios where ensemble strategies outperform individual models.

- Neural Network Construction Using Keras
  Using the Keras API, students build and train basic feedforward neural networks. They gain practical experience in selecting activation functions, configuring loss functions, tuning hyperparameters (e.g., learning rate, epochs), and applying backpropagation using optimizers like SGD and Adam.

- Unsupervised Learning for Pattern Discovery
  Through clustering algorithms such as K-Means, Hierarchical Clustering, DBSCAN, and Gaussian Mixture Models (GMM), learners segment real-world datasets and visualize clusters. They apply evaluation methods like the Elbow Method and Silhouette Score to validate clustering quality.

Text Book

T1: Aurélien Géron, "Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow", Oreilly, Second Edition, 2019.
T2: Andreas C Muller, Sarah Guido, "Introduction to Machine Learning with Python :A Guide for Data Scientists", Oreilly, First Edition, 2018.
T3: Giuseppe Bonaccorso, "Machine Learning Algorithms: A reference guide to popular algorithms from data science and machine learning", Packt Publishing, 2017.

References

R1: Tan P. N., Steinbach M & Kumar V. "Introduction to Data Mining", Pearson Education, 2016.

R2: Molnar, C. (2020). Interpretable Machine Learning: A Guide for Making Black Box Models Explainable. Germany: Leanpub.
R3: Wichert, A. M., Sa-couto, L. (2021). Machine Learning - A Journey To Deep Learning: With Exercises And Answers. Singapore: World Scientific Publishing Company.

E-Resources/Web Resources

1. https://openFullText.html?DP=https://directory.doabooks.org/handle/20.500.12854/52889
2. https://openFullText.html?DP=https://open.umn.edu/opentextbooks/textbooks/92
3. https://openFullText.html?DP=https://open.umn.edu/opentextbooks/textbooks/178

Course Code: CAI2501	Course Name: Machine Learning Lab Type of Course: Lab Course	L- T-P- C	0-0-2-1
Version No.	1.0		
NCrF Level-	5.0		
Course Pre-requisites	NIL		
Co-requisites	CAI2500		
Anti-requisites	CSEXXXX		
Course Description	This lab course builds hands-on expertise in implementing machine learning algorithms using Python libraries like Scikit-learn, Matplotlib, Seaborn, and Keras. It emphasizes data preprocessing, regression, classification, ensemble learning, clustering, and outlier detection. Focus areas include model functionality, data visualization, and performance evaluation using real-world datasets and structured experiments.		

Course Objective	Course Objectives should clearly state what the course intends to achieve. They must be broad, concise and student-centered, focusing on the knowledge, skills and attitudes students are expected to develop by the end of the course. Use action verbs such as *introduce, develop, equip, enable, provide, familiarize* and *prepare.* Keep it limited to 3–5 objectives.
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Understand key machine learning concepts, algorithms, and their real-world applications. [Level 2 – Understand] **CO2:** Explain the steps involved in data preprocessing, model training, and evaluation. [Level 2 – Understand] **CO3:** Apply supervised learning techniques such as linear regression, logistic regression, and SVM to structured datasets. [Level 3 – Apply] **CO4:** Apply unsupervised learning algorithms like k-means and DBSCAN to perform clustering and outlier detection. [Level 3 – Apply] **CO5:** Analyze model performance using metrics such as accuracy, precision, recall, and confusion matrix. [Level 4 – Analyze]

Course Content:

List of Laboratory Tasks:

- **Experiment 1: Handling Missing Values**
  Level 1: Use Scikit-learn to handle missing values in a UCI dataset.
  Level 2: Write custom Python functions to implement missing value handling methods.

- **Experiment 2: Data Visualization**
  Level 1: Perform exploratory data analysis using Scatter Plot, Pair Plot, and Count Plot with Matplotlib and Seaborn.
  Level 2: Create advanced visualizations using Heat Maps and WordClouds.

- **Experiment 3: Regression Learning**
  Level 1: Implement Simple Linear Regression and evaluate model parameters and performance. Plot learning curves.
  Level 2: Implement Polynomial Regression and compare learning curves with linear regression.

- **Experiment 4: Logistic Regression**
  Level 1: Generate logistic/sigmoid plots using custom code.
  Level 2: Implement logistic regression, estimate class probabilities, and visualize decision boundaries.

- **Experiment 5: Bayesian Learning**
  Level 1: Apply Naïve Bayes algorithm on a classification problem using a UCI dataset.

- **Experiment 6: Support Vector Machines (SVM)**
  Level 1: Implement linear and non-linear SVM models for classification using UCI datasets.

- **Experiment 7: Ensemble Learning – Part 1**
  Level 1: Implement Bagging, Pasting, and Out-of-Bag evaluation.
  Level 2: Use Random Patches and Random Subspace methods.

- **Experiment 8: Ensemble Learning – Part 2**
  Level 1: Implement AdaBoost, Gradient Boosting, and Stacking ensemble models.

- **Experiment 9: Perceptron Learning**
  Level 1: Implement a Perceptron Classifier from scratch.
  Level 2: Build an image classifier using Keras Sequential API.

- **Experiment 10: Unsupervised Learning – Clustering with K-means**
  Level 1: Implement simple and mini-batch K-means. Use Elbow Method and Silhouette Coefficient to find optimal clusters.
  Level 2: Apply clustering for image segmentation and preprocessing. Use K-means++ initialization.

- **Experiment 11: Density-Based Clustering**
  Level 1: Implement DBSCAN for clustering using local density estimation. Perform both hard and soft clustering.

- **Experiment 12: Outlier Detection**
  Level 1: Detect outliers using Isolation Forest and Local Outlier Factor algorithms.

Targeted Applications & Tools:	
Python (NumPy & pandas)	Core language and libraries for data ingestion, cleaning, transformation, and numerical computing across all lab experiments (missing-value handling, custom algorithms, feature prep).
Scikit-learn	Ready-made implementations of regression, classification, SVM, clustering (k-means, DBSCAN), ensemble methods (Bagging, AdaBoost, Gradient Boosting, Stacking), and evaluation metrics used in Experiments 1–12.

Matplotlib & Seaborn	Exploratory data analysis and result visualization: scatter/ pair / count plots, heat-maps, learning-curve graphs, decision-boundary plots to interpret model behaviour.
Keras (TensorFlow backend)	Building neural models in Experiment 9 (Level 2) and enabling quick prototyping of perceptrons / image classifiers with the Sequential API.
Jupyter Notebook / Google Colab	Interactive coding environment to run, document, and share all lab notebooks, integrate narrative text with code, and leverage GPU acceleration when required.

Skill-Building Activities through Experiential Learning:

- **Data Preprocessing with Real-World Datasets**
  Learners perform hands-on preprocessing of real-world datasets (e.g., UCI, Kaggle) by handling missing values, scaling features, and encoding categorical variables. This foundational step helps them understand the impact of clean data on model performance.

- **Exploratory Data Analysis and Visualization**
  Using libraries such as Matplotlib and Seaborn, students conduct exploratory data analysis through plots like scatter plots, heatmaps, and pair plots. Visualizations are used to identify patterns, correlations, and anomalies before model building.

- **Model Building with Supervised Learning Algorithms**
  Students implement regression and classification algorithms—Linear Regression, Logistic Regression, Naïve Bayes, and SVM—on structured datasets. They analyze model predictions using decision boundaries, ROC curves, and learning curves.

Text Book

T1: Aurélien Géron, "Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow", Oreilly, Second Edition, 2019.
T2: Andreas C Muller, Sarah Guido, "Introduction to Machine Learning with Python :A Guide for Data Scientists", Oreilly, First Edition, 2018.
T3: Giuseppe Bonaccorso, "Machine Learning Algorithms: A reference guide to popular algorithms from data science and machine learning", Packt Publishing, 2017.

References

R1: Tan P. N., Steinbach M & Kumar V. "Introduction to Data Mining", Pearson Education, 2016.

R2: Molnar, C. (2020). Interpretable Machine Learning: A Guide for Making Black Box Models Explainable. Germany: Leanpub.
R3: Wichert, A. M., Sa-couto, L. (2021). Machine Learning - A Journey To Deep Learning: With Exercises And Answers. Singapore: World Scientific Publishing Company.

E-Resources/Web Resources

1. https://towardsdatascience.com/machine-learning/home
2. MITopencourseware:https://ocw.mit.edu/courses/6-0002-introduction-to-computational-thinking-and-data-science-fall-2016/resources/lecture-11-introduction-to-machine-learning/
3. https://onlinecourses.nptel.ac.in/noc21_cs85/preview

Course Code: **CAI2511**	**Course Name: Ethics of AI** **Type of Course: Theory Course**	**L- T-P- C**	**2-0-0-2**
Version No.	1.0		
NCrF Level-	5.0		
Course Pre-requisites	NIL		
Co-requisites	NIL		
Anti-requisites	CSEXXXX		
Course Description	This course examines the ethical implications of Artificial Intelligence, concentrating on bias, fairness, accountability, privacy, and societal implications. It integrates ethical theories, global governance frameworks, and real-world case studies. Emphasis is placed on critically evaluating AI systems and proposing responsible, human-centered solutions for ethical design and deployment.		
Course Objective	This course is designed to understand and apply classical ethical frameworks to AI dilemmas.		
Course Outcomes	**On successful completion of the course the students shall be able to:** CO1: Understand key ethical theories and principles relevant to Artificial Intelligence. — Level 2 – Understand CO2: Describe common ethical challenges in AI systems, including bias, fairness, and accountability. — Level 2 – Understand CO3: Explain the impact of AI technologies on privacy, surveillance, and data governance. — Level 2 – Understand CO4: Analyze the societal and ethical implications of AI systems, including issues of autonomy, human dignity, and fairness. — Level 4 – Analyze CO5: Apply ethical reasoning to evaluate AI development practices and suggest appropriate improvements. — Level 3 – Apply		
Course Content:			
Module 1	Foundations of Ethics in Technology	Assignment	8 Sessions
Introduction to Ethics: Moral theories and frameworks, Relevance of ethics in Al systems, Overview of Al capabilities and impact, Stakeholders in Al development and deployment. Case studies: Early ethical dilemmas in Al applications			
Module 2	Bias, Fairness, and Accountability in AI	Assignment	7 Sessions
Understanding algorithmic bias and discrimination, Fairness in machine learning models, Accountability and transparency in Al systems, Auditing Al for ethical compliance. Case studies: Al in hiring, finance, and law enforcement			
Module 3	Privacy, Surveillance, and Data Ethics	Assignment	8 Sessions
Data privacy and protection principles, Surveillance technologies and ethical concerns, Informed consent and			

user autonomy, Data governance and ethical data usage, Case studies: Social media platforms, facial recognition

Module 4	AI in Society: Autonomy, Human Dignity & Labor	Assignment	7 Sessions

Topics: Explainability and interpretability in Al Regulation and legal frameworks (e.g., EU AI Act, IEEE standards), Human-in-the-loop and value-sensitive design, Future of ethical Al: Aligning Al with human values, Case studies: Autonomous systems, healthcare Al, Digital divide and access to Al technologies

Targeted Applications & Tools:

AI Fairness 360 (IBM)	To explore algorithmic bias and test fairness metrics on real-world datasets.
Google's What-If Tool (TensorBoard)	Interactive visual interface to analyze ML models for bias, fairness, and interpretability.
OpenAI GPT Playground / ChatGPT	Used for understanding and evaluating real-time ethical implications in large language models.
Google Colab / Jupyter Notebook	Collaborative environment for running code and simulating ethical case studies using Python libraries.
Python (pandas, numpy, matplotlib)	For handling datasets, visualizing bias and fairness metrics, and analyzing data privacy concerns.

Skill-Building Activities through Experiential Learning:

- **Ethical Case Study Debates**
  Learners engage in structured debates and group discussions based on real-world AI controversies, such as algorithmic bias in hiring, facial recognition in public surveillance, or predictive policing. These sessions develop critical thinking and ethical reasoning by encouraging students to analyze competing perspectives and articulate responsible positions grounded in moral frameworks.
- **Privacy Impact Simulation**
  Through role-play and scenario analysis, learners simulate data collection and consent processes in applications like health tracking or social media. They assess privacy risks, explore anonymization strategies, and reflect on the ethical implications of surveillance and behavioral targeting.
- **Policy Brief and Regulatory Simulation**
  Teams draft ethical AI policy briefs addressing a societal issue (e.g., use of AI in autonomous vehicles or healthcare diagnostics). They participate in a mock regulatory hearing where each team defends its policy

Text Book

T1: Virginia Dignum – Responsible Artificial Intelligence: How to Develop and Use AI in a Responsible Way, Springer, 2019.
T2: Mark Coeckelbergh – AI Ethics, MIT Press, 2020.

References

R1: Wendell Wallach and Colin Allen – Moral Machines: Teaching Robots Right From Wrong, Oxford University Press, 2008.

R2: Shannon Vallor – Technology and the Virtues: A Philosophical Guide to a Future Worth Wanting, Oxford University Press, 2016.3.

R3: Cathy O'Neil – Weapons of Math Destruction: How Big Data Increases Inequality and Threatens

Democracy, Crown Publishing, 2016. **R4:** Nick Bostrom – Superintelligence: Paths, Dangers, Strategies, Oxford University Press, 2014
E-Resources/Web Resources 1. Ethics of Artificial Intelligence | UNESCO 2. https://www.bing.com/search?pglt=297&q=Ethics+of+AI+web+resources&cvid=28c06308df1d47db9de8becce15d7da8&gs_lcrp=EgRlZGdlKgYIABBFGDkyBggAEEUYOdIBCDU4NTlqMGoxqAIIsAIB&FORM=ANNTA1&PC=HCTS

Course Code: **CSS2262**	**Course Title: Analysis of Algorithms** **Type of Course: Theory Course (PCC)**	**L- T-P- C**	**3-1-0-4**
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	Nil		
Course Description	This course introduces techniques for the design and analysis of efficient algorithms and methods of application. The course discusses classic algorithm design approaches such as Divide and Conquer, Dynamic Programming, and Greedy Methods. It also covers strategies for searching solution spaces using Backtracking and Branch-and-Bound techniques. The course emphasizes the analysis of algorithms and their classification into various complexity classes. In addition, it introduces emerging concepts in AI-based optimization, intelligent search techniques, and algorithmic approaches for secure and smart computing applications.		
Course Objective	To develop the ability to design and analyze efficient algorithms by enabling students to evaluate time and space complexity, apply various algorithmic paradigms such as divide-and-conquer, greedy, dynamic programming, backtracking, and branch-and-bound, compare different approaches for solving computational problems, and understand intelligent optimization techniques used in AI-enabled and secure computing systems.		
Course Outcomes	On successful completion of this course, students will be able to: CO1: Explain the efficiency of algorithms in terms of time and space complexity. — Understand CO2: Apply divide-and-conquer and decrease-and-conquer techniques to solve searching and sorting problems. — Apply CO3: Apply dynamic programming and greedy techniques to solve computational problems. — Apply CO4: Apply time–space trade-off techniques and analyze the limitations of algorithms for problem solving. — Apply CO5: Analyze problems using backtracking, branch-and-bound, — Analyze		

	and AI-based optimization techniques to develop optimal solutions.	

Course Content:

Module 1	**Introduction**	**14 Sessions**
Introduction: Notion of algorithm, Importance of algorithm, Important Problem Types, Fundamentals of analysis of algorithm efficiency: General equation, Order of growth, basic efficiency classes, asymptotic notations, mathematical analysis of recursive & non-recursive algorithms, Brute Force technique: Bubble sort, selection sort, linear search, brute force string matching, Exhaustive search: Travelling salesperson problem, knapsack problem.		
Module 2	**Divide-and-conquer**	**11 Sessions**
Divide & Conquer technique: General Method, General recursive equation, Masters Theorem, Binary search, Merge sort, quick sort. Decrease & Conquer technique: General method, insertion sort, topological sorting		
Module 3	**Dynamic programming**	**13 Sessions**
Dynamic Programming: General method, Warshall algorithm, Floyd algorithm, 0/1 knapsack, Greedy method: General method, prim's algorithm, Kruskal algorithm, Dijkstra's algorithm, Huffman coding		
Module 4	**Space and time tradeoff:**	**11 Sessions**
Space and time tradeoff: introduction, Input enhancement in string matching: Horspool algorithm and Boyer-Moore algorithm, Limitations of algorithm power: Introduction, lower bound arguments, decision trees, P, NP and NP-complete problems		
Module 5	**Coping with Limitations of Algorithm Power and Intelligent Optimization**	**11 Sessions**
Introduction to limitations of algorithm power, Backtracking: n-Queens Problem, Subset Sum Problem, Hamiltonian Circuit Problem, Branch and Bound: Travelling Salesperson Problem, Knapsack Problem, Job Assignment Problem, AI-based Search and Optimization Techniques, Machine Learning Approaches for Optimization Problems, Intelligent Decision-Making Algorithms, Applications in Secure and Smart Computing Systems.		

Text Book:

1. Anany Levitin, *"Introduction to the Design and Analysis of Algorithms"*, 3rd edition, Pearson Education, 2018.
2. Thomas H.Cormen, Charles E.Leiserson, Ronald L. Rivest and Clifford Stein, *"Introduction to Algorithms"*, 4th edition, MIT Press, 2022.

Reference Books:

R1. AV Aho, J Hopcroft, JD Ullman, *"The Design and Analysis of Algorithms"*, Addison-Wesley, 1974

R2. Tim Roughgarden, *"Algorithms Illuminated"* (books 1 through 3), "Operating Systems Design and Implementation", Soundlikeyourself Publishing, 2017-2019.

R3. J. Kleinberg and E. Tardos, "*Algorithm Design*", Addison-Wesley, 2005

Online Resources:

1. EBook: https://presiuniv.knimbus.com/user#/home 2. https://onlinecourses.nptel.ac.in/noc26_cs67/ 3. www.youtube.com/watch?v=so4zT-u09LM&list=PL8GCet93Cu1iRF611mrUu-Rhy12fMwZOf
Topics relevant to "SKILL DEVELOPMENT": knapsack, prim's, kruskal's algorithm, quick sort, binary search for **Skill Development** through **Problem Solving methodologies**. This is attained through assessment component mentioned in course handout.

Course Code: CSS2263	Course Title: Analysis of Algorithms Lab Type of Course: Laboratory Course (PCC)	L- T-P- C	0-0-2-1
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course introduces techniques for the design and analysis of efficient algorithms and methods of application. The course discusses classic algorithm design approaches such as Divide and Conquer, Dynamic Programming, and Greedy Methods. It also covers strategies for searching solution spaces using Backtracking and Branch-and-Bound techniques. The course emphasizes the analysis of algorithms and their classification into various complexity classes. In addition, it introduces emerging concepts in AI-based optimization, intelligent search techniques, and algorithmic approaches for secure and smart computing applications.		
Course Objective	To develop the ability to design and analyze efficient algorithms by enabling students to evaluate time and space complexity, apply various algorithmic paradigms such as divide-and-conquer, greedy, dynamic programming, backtracking, and branch-and-bound, compare different approaches for solving computational problems, and understand intelligent optimization techniques used in AI-enabled and secure computing systems.		
Course Outcomes	On successful completion of this course, students will be able to: CO1: Compute the efficiency of algorithms in terms of time and space complexity. — Level 3 – Apply CO2: Apply divide-and-conquer techniques to solve searching and sorting problems. — Level 3 – Apply CO3: Apply dynamic programming techniques to solve computational problems. — Level 3 – Apply CO4: Apply greedy techniques for solving optimization problems. — Level 3 – Apply CO5: Analyze problems using backtracking, branch-and-bound, and AI-based optimization techniques to develop optimal solutions. — Level 4 – Analyze		
Course Content			

List of Laboratory Tasks: 30 Sessions

WEEK 1: Algorithm Basics & Analysis

Lab Sheet 1: Algorithm Fundamentals

Level 1: Write an algorithm and program to find minimum and maximum elements in a list.

Level 2: Analyze the number of comparisons and derive time complexity.

Lab Sheet 2: Asymptotic Notations

Level 1: Write programs to demonstrate linear and quadratic time complexity.

Level 2: Compare growth rates of different functions using Big-O, Ω, and Θ.

WEEK 2: Efficiency Analysis

Lab Sheet 3: Recursive Algorithms

Level 1: Write a recursive program for factorial.

Level 2: Solve recurrence relations for recursive algorithms.

Lab Sheet 4: Non-Recursive Analysis

Level 1: Write programs with nested loops and count operations.

Level 2: Derive general equations and order of growth.

WEEK 3: Brute Force Techniques

Lab Sheet 5: Sorting Algorithms

Level 1: Implement Bubble Sort and Selection Sort.

Level 2: Compare their time complexity experimentally.

Lab Sheet 6: Searching & String Matching

Level 1: Implement Linear Search.

Level 2: Implement brute-force string matching algorithm.

WEEK 4: Exhaustive Search

Lab Sheet 7: Knapsack Problem

Level 1: Solve 0/1 Knapsack using brute force.

Level 2: Analyze solution space complexity.

Lab Sheet 8: Travelling Salesperson Problem

Level 1: Solve TSP using brute force approach.

Level 2: Compare performance for different inputs.

WEEK 5: Divide and Conquer

Lab Sheet 9: Binary Search

Level 1: Implement Binary Search.

Level 2: Analyze recurrence relation.

Lab Sheet 10: Merge Sort

Level 1: Implement Merge Sort.

Level 2: Analyze using Master's Theorem.

WEEK 6: Advanced Divide & Conquer

Lab Sheet 11: Quick Sort

Level 1: Implement Quick Sort.

Level 2: Compare best and worst cases.

Lab Sheet 12: Master's Theorem

Level 1: Solve recurrence equations.

Level 2: Apply Master's Theorem to different cases.

WEEK 7: Decrease and Conquer

Lab Sheet 13: Insertion Sort

Level 1: Implement Insertion Sort.

Level 2: Compare with Bubble Sort.

Lab Sheet 14: Topological Sorting

Level 1: Implement Topological Sort.

Level 2: Apply on real-world dependency graph.

WEEK 8: Dynamic Programming

Lab Sheet 15: Warshall Algorithm

Level 1: Implement Warshall algorithm.

Level 2: Compute transitive closure.

Lab Sheet 16: Floyd Algorithm

Level 1: Implement Floyd's algorithm.

Level 2: Compare with Dijkstra.

WEEK 9: DP Applications

Lab Sheet 17: Knapsack DP

Level 1: Solve 0/1 Knapsack using DP.

Level 2: Compare with brute force approach.

Lab Sheet 18: DP Optimization

Level 1: Identify overlapping subproblems.

Level 2: Convert recursive solution to DP.

WEEK 10: Greedy Algorithms

Lab Sheet 19: MST Algorithms

Level 1: Implement Prim's algorithm.

Level 2: Implement Kruskal's algorithm.

Lab Sheet 20: Shortest Path

Level 1: Implement Dijkstra's algorithm.

Level 2: Compare greedy vs DP approach.

WEEK 11: Huffman Coding

Lab Sheet 21: Huffman Tree

Level 1: Construct Huffman Tree.

Level 2: Analyze compression efficiency.

Lab Sheet 22: Greedy Analysis

Level 1: Demonstrate greedy choice property.

Level 2: Prove correctness of greedy algorithms.

WEEK 12: String Matching Optimization

Lab Sheet 23: Horspool Algorithm

Level 1: Implement Horspool algorithm.

Level 2: Compare with brute-force method.

Lab Sheet 24: Boyer-Moore Algorithm

Level 1: Implement Boyer-Moore algorithm.

Level 2: Analyze efficiency improvements.

WEEK 13: Complexity Theory

Lab Sheet 25: Lower Bounds

Level 1: Analyze decision tree for sorting.

Level 2: Derive lower bounds.

Lab Sheet 26: NP Problems

Level 1: Classify problems as P or NP.

Level 2: Analyze NP-complete problems.

WEEK 14: Backtracking

Lab Sheet 27: N-Queens

Level 1: Solve N-Queens problem.

Level 2: Analyze search tree.

Lab Sheet 28: Subset Sum

Level 1: Solve subset sum problem.

Level 2: Compare with DP solution.

WEEK 15: Branch & Bound

Lab Sheet 29: TSP

Level 1: Solve TSP using Branch & Bound.

Level 2: Compare with brute-force approach.

Lab Sheet 30: Job Assignment

Level 1: Solve assignment problem.

Level 2: Analyze pruning efficiency.

WEEK 15.1: AI-Based Optimization and Intelligent Search

Lab Sheet 31: Intelligent Optimization Techniques

Level 1: Implement AI-based heuristic search for optimization problems.
Level 2: Compare heuristic search with Branch & Bound approach for performance analysis.

Lab Sheet 32: Smart Decision-Making Applications

Level 1: Implement a simple recommendation or path optimization system using graph/search techniques.
Level 2: Analyze intelligent pruning and optimization efficiency in real-world problem solving.

Targeted Application & Tools that can be used :

PyTorch/Jupyter Notebook – For Python programming

Text Book :

T1 Anany Levitin, *"Introduction to the Design and Analysis of Algorithms*", 3rd edition, Pearson Education, 2018.

T2 Thomas H.Cormen, Charles E.Leiserson, Ronald L. Rivest and Clifford Stein, *"Introduction to Algorithms*", 4th edition, MIT Press, 2022.

References :

R1. J. Kleinberg and E. Tardos, "*Algorithm Design*", Addison-Wesley, 2005.

R2. Tim Roughgarden, *"Algorithms Illuminated"* (books 1 through 3), "Operating Systems Design and Implementation", Soundlikeyourself Publishing, 2017-2019.

R3. AV Aho, J Hopcroft, JD Ullman, *"The Design and Analysis of Algorithms"*, Addison-Wesley, 1974.

R4. Donald E. Knuth, "*The Art of Computer Programming",* Volumes 1and 3 Pearson.

Web Based Resources and E-books:

W1. NPTEL: https://onlinecourses.nptel.ac.in/noc19_cs47/preview

W2. Coursera: Analysis of Algorithms by Princeton University

W3. Algorithms Specialization in Coursera by Stanford University(Group of 4 courses).

W4. Algorithms Coding Contest Links maintained by Prof Gerth Stølting Brodal of Aarhus University

Topics relevant to "EMPLOYABILITY SKILLS": The lab experiments and assessments enable the student to acquire Skill Development through Experiential Learning techniques

Course Code: APT4004	Course Title: Aptitude Training-Intermediate Type of Course: Practical Only Course	L- T - P- C	0-0-2-0
Version No.	1.0		
Course Pre-requisites	Students should have the basic concepts of Quantitative aptitude along with its applications in real life problems.		
Anti-requisites	NIL		
Course Description	This is a skill-based training program for the students. This course is designed to enable the students to enhance their skills in Quantitative Aptitude.		
Course Objective	The objective of the course is to familiarize the learners with the concepts of Aptitude and attain Skill Development through Problem Solving techniques.		
Course Outcomes	On successful completion of this course the students shall be able to: CO1: Recall all the basic mathematical concepts. — Level 1 – Remember CO2: Identify the principal concept needed in a question. — Level 2 – Understand CO3: Solve quantitative and logical ability questions using the appropriate concept. — Level 3 – Apply CO4: Analyze the data given in complex problems. — Level 4 – Analyze		
Course Content:			
Module 1	Quantitative Ability 1	Assignment	16 Hours
Topics: Number System, Percentage, Ratio and Proportion, Average, Mixture and Allegation, Time and Work, Profit and Loss			
Module 2	Quantitative Ability 2	Assignment	14 Hours
Topics: Time Speed and Distance, Boats and Streams, Simple Interest, Compound Interest, Probability, Permutation and Combination			
Targeted Application & Tools that can be used: Application area: Placement activities and Competitive examinations. Tools: LMS			
Continuous Evaluation:			
CA1 – Online Test CA2 – Online Test CA3 – Online Test Assignment			

Text Book: Fast Track Objective by Rajesh Verma R S Aggarwal Rakesh Yadav References: www.indiabix.com www.testbook.com www.youtube.com/c/TheAptitudeGuy/videos
Topics relevant to Skill Development: Quantitative aptitude for Skill Development through Problem solving Techniques. This is attained through components mentioned in course handout.

Course Code: CHE7601	Course Name: Environmental Studies Type of Course:MOOC course	L- T-P- C	1-0-0-0
Version No.	1.0		
NCrF Level-	5.0		
Course Pre-requisites	NIL		
Co-requisites	NIL		
Anti-requisites	CSEXXXX		
Course Description	This course is designed to improve the learners' SKILL DEVELOPMENT by using PATICIPATIVE LEARNING techniques. This course aims to familiarize students with fundamental environmental concepts and their relevance to business operations, preparing them to address forthcoming sustainability challenges. It is designed to equip students with the knowledge and skills needed to make decisions that account for environmental consequences, fostering environmentally sensitive and responsible future managers. This course is designed to cater to Environment and Sustainability		
Course Objective	The objective of the course is 'SKILL DEVELOPMENT' of the student by using 'PARTICIPATIVE LEARNING' techniques		

Course Outcomes	On successful completion of the course the students shall be able to: CO1: Describe the issues related to natural resources, ecosystems and biodiversity — L1– Remember CO2: Identify environmental hazards affecting air, water and soil quality — L2– Understand CO3:Recognize the importance of healthy environment and finding the sustainable methods to protect the environment — L3 – Apply CO4: Convert skills to address immediate environmental concerns through changes in environmental processes, policies, and decisions — L4 – Analyze

Course Content:

Module 1	Understanding Environment, Natural Resources, and Sustainability		

Topics:

Classification of natural resources, issues related to Population growth and their overutilization, and strategies for their conservation. Water, air, soil, mineral, energy and food source. Effect of human activities on natural resources.

Concept of sustainability- Sustainable Development Goals (SDGs)- targets and indicators, challenges and strategies for SDGs; Sustainable practices in managing resources, including deforestation, water conservation, Desalination – types, energy security, and food security issues, Life Cycle thinking and Circular Economy.

Module 2	Ecosystems, Biodiversity, and Sustainable Practices		

Topics:

Ecosystems and ecosystem services: Various natural ecosystems, Major ecosystem types in India and their basic characteristics; forests, wetlands, grasslands, agriculture, coastal and marine; Ecosystem services- classification and their significance.

The importance of biodiversity, Types of biodiversity, Biodiversity and Climate Change, the threats it faces, hotspots, and the methods used for its conservation. Strategies for in situ and ex situ conservation, mega diverse nation.

Module 3	Environmental Pollution, Waste Management, and Sustainable Development		

Topics:

Types of pollution- Chemical, - Biological, Biomedical, noise, air, water, soil, thermal, radioactive and marine pollution, and their impacts on society. Urbanization and Urban environmental problems; effects, and mitigation.

Causes of pollution, such as global climate change, ozone layer depletion, the greenhouse effect, and acid rain, with a particular focus on pollution episodes in India. Importance of adopting cleaner technologies; Solid waste management;

Sustainable Materials and Technologies: Biodegradable and compostable materials, Recycled and reclaimed materials (E-waste management), Sustainable manufacturing processes.

Module 4	Social Issues, Legislation, and Practical Applications		

Topics:

Overview of key environmental legislation and the judiciary's role in environmental protection, including the Water (Prevention and Control of Pollution) Act of 1974, the Environment (Protection) Act of 1986, and the Air (Prevention and Control of Pollution) Act of 1981. Hazardous waste Rule 1989, Biomedical Waste handling 1998, Fly Ash Rule 1999, Municipal Solid Waste Rule 2000, Battery Rules 2001, E- Waste Rules 2011, Plastic waste management Rules 2016, Construction Demolition waste Rules 2016 National Biodiversity Action Plan (NBAP)

Major International Environmental Agreements: Convention on Biological Diversity (CBD), The Biological Diversity (Amendment) Act, 2023, United Nations Framework Convention on Climate Change (UNFCCC); Kyoto Protocol; Paris Agreement.

Major International organisations and initiatives: United Nations Environment Programme (UNEP), United Nations Educational, Scientific and Cultural Organization (UNESCO), Intergovernmental Panel on Climate Change (IPCC).

Targeted Application & Tools that can be used:

Application areas are Energy, Environment and sustainability

Tools: Online Tools – NPTEL and Swayam.

Project work/Assignment:

Assessment Type

- Online exams (MCQs) will be conducted by the department of Chemistry

Online Link*:

1) Lecure by Dr. Samik Chowdhury, Dr. Sudha Goel, NPTEL course: Environmental Science, https://nptel.ac.in/courses/109105203, 2024.
2) Lecture by Dr. Padmavati, Dr Narendran Thiruthy, NPTEL Course: Biodiversity Protection, Farmers and Breeders Rights, https://nptel.ac.in/courses/129105008, 2024.

* Other source links are available in below Resources link.

Text Book

1. G. Tyler Miller and Scott Spoolman (2020), Living in the Environment, 20th Edition, Cengage Learning, USA
2. Poonia, M.P. Environmental Studies (3rd ed.), Khanna Book Publishing Co.

3. Bharucha, E. Textbook of Environmental Studies (3rd ed.) Orient Blackswan Private Ltd.
4. Dave, D., &Katewa, S. S. Text Book of Environmental Studies. Cengage Learning India Pvt Ltd.
5. Rajagopalan, R. Environmental studies: from crisis to cure (4th ed.). Oxford University Press.
6. Basu, M., & Xavier Savarimuthu, S. J. Fundamentals of environmental studies. Cambridge University Press.
7. Roy, M. G. Sustainable Development: Environment, Energy and Water Resources. Ane Books.
8. Pritwani, K. Sustainability of business in the context of environmental management. CRC Press.
9. Wright, R.T. & Boorse, D.F. Environmental Science: Toward A Sustainable Future (13th ed,). Pearson.

Reference Books

1. Varghese, Anita, Oommen, Meera Anna, Paul, Mridula Mary, Nath, Snehlata (Editors) (2022), Conservation through Sustainable Use: Lessons from India. Routledge.
2. William P. Cunningham and Mary Ann Cunningham (2020), Principles of Environmental Science: Inquiry & Applications, 9th Edition, McGraw-Hill Education, USA.
3. Richard A. Marcantonio, Marc Lame (2022). Environmental Management: Concepts and Practical Skills. Cambridge University Press.
4. Manahan, S.E. (2022). Environmental Chemistry (11th ed.). CRC Press. https://doi.org/10.1201/9781003096238
5. Theodore, M. K. and Theodore, Louis (2021) Introduction to Environmental Management, 2nd Edition. CRC Press

Resources:

1. https://nptel.ac.in/courses/109105203
2. https://archive.nptel.ac.in/courses/120/108/120108004/
3. https://nptel.ac.in/courses/127105018
4. https://onlinecourses.nptel.ac.in/noc23_lw06/preview
5. https://onlinecourses.swayam2.ac.in/ini25_bt02/preview
6. https://archive.nptel.ac.in/courses/120/108/120108002/
7. https://onlinecourses.swayam2.ac.in/ini25_bt02/preview
8. https://nptel.ac.in/courses/102104088
9. https://nptel.ac.in/courses/124107165
10. https://nptel.ac.in/courses/109106200
11. https://archive.nptel.ac.in/content/storage2/courses/120108004/module1/lecture1.pdf
12. https://onlinecourses.swayam2.ac.in/nou25_ge19/preview
13. https://onlinecourses.swayam2.ac.in/ini25_hs01/preview
14. http://kcl.digimat.in/nptel/courses/video/105105184/L32.html
15. https://nptel.ac.in/courses/105105169

Topics relevant to Skill Development:

1. An attitude of enquiry.
2. Write reports

The topics related to Environment and Sustainability :

All topics in theory component are relevant to Environment and Sustainability.

Course Code: FIN1002	Course Title: Essentials of Finance Type of Course: HSMC		L-T-P-C	3-0-0-3
Version No.	1.0			
Course Pre-requisites	This course is designed to be accessible to all students, regardless of their prior financial knowledge.			
Anti-requisites				
Course Description	This course is designed to equip students with a foundational understanding of key financial concepts and principles. It will enable them to comprehend the core functions of finance, delve into the intricacies of financial management within organizations, and gain insights into the fundamental aspects of taxation. The course aims to develop students' abilities to interpret financial statements, evaluate investment opportunities, understand capital structure decisions, and navigate the basics of tax implications.			
Course Objective	Upon successful completion of this course, students will be able to: Understand the basic forms of business organization and their financial implications. Understand the fundamental principles and concepts that influence financial decision-making in various contexts. Analyse and interpret financial statements to assess the financial health and performance of an organization. Identify income under various heads of income as per Income Tax Act, 1961 and determine the tax liability.			
Course Outcomes	List the course outcomes On successful completion of this course the students shall be able to: Understand the basic concepts of finance and financial markets and organizations. Apply and interpret financial information for business decision making. Identify various heads of income and deduction under Income Tax Act, 1961.			
Course Content:				
Module 1	Introduction to Finance	Assignment/ Quiz	Numerical solving Task	10 Sessions
Definition and Scope of Finance, Areas of Finance: Corporate Finance, Investments, Financial Institutions, International Finance; Types of Financial Markets: Money Markets vs. Capital Markets, Primary vs. Secondary Markets; Forms of Business Organization and Financial Goals: Shareholder Wealth Maximization vs. Profit Maximization; Understanding Financial Statements: Balance Sheet and Income Statement- Simple Numerical.				
Module 2	Financial Management	Assignment/ Quiz	Numerical solving Task	18 Sessions
Capital Budgeting Decisions: Payback Period, Net Present Value (NPV), Profitability Index (PI), Internal Rate of Return (IRR); Leverage- Basic Numerical; Capital Structure Decisions: Optimal Capital Structure, Trade-off Theory of Capital Structure; Cost of Capital: Equity, Debt, WACC; Dividend Policy: Factors influencing Dividend Policy.				
Module 3	Taxation	Assignment/ Quiz	Numerical solving Task	17 Sessions
Principles of a Good Tax System: Equity, Certainty, Convenience, Economy; Direct vs. Indirect Taxes; Residential Status of an Individual- Basic Problems; Heads of Income; Salary, House Property- Basic Numerical; Deductions under Chapter VI-A; Computation of Taxable Income and Tax Liability; E-Filing				

procedure.
Targeted Application & Tools that can be used: Textbooks, PPT, Spreadsheet Software (e.g., Microsoft Excel), Official Website of Income Tax Department.
Project Work/ Assignment:
1. Presentation: There will be a group presentation, where the students will be given a topic. They will have to explain/demonstrate the working and discuss the applications for the same. 2. Case Study: - At the end of the course students will be given a 'real-world' cases like business models of successful companies or tax evasion by reputed companies on which they have to come up with detailed analysis and assessment.
Text Book(s): Dr. Vinod K. Singhania & Dr. Monica Singhania. (Latest Assessment Year Edition). Students' Guide to Income Tax including GST. Taxmann Publications. Pandey, I. M. (2025). Financial Management. Vikas Publishing House.
Reference Book (s): Bhole, L.M., & Mahakud, J. (Current Edition). Financial Institutions and Markets: Structure, Growth and Innovations. McGraw Hill Education India. Mehrotra, H.C., & Goyal, S.P. (Latest Assessment Year Edition). Income Tax Law & Practice. Sahitya Bhawan Publications. Gordon, E., & Natarajan, K. (Current Edition). Financial Markets and Services. Himalaya Publishing House. Online Resources (e-books, notes, ppts, video lectures etc.): https://presidencyuniversity.linways.com https://onlinecourses.nptel.ac.in/noc24_ec01/preview https://www.incometax.gov.in/iec/foportal/
Topics relevant to "SKILL DEVELOPMENT": This course is designed to provide practical financial skills through participative learning techniques. Students will engage in performing suitable calculations to determine financial parameters (e.g., time value of money, investment returns, tax liabilities) and analysing financial statements to assess organizational performance and make informed decisions.

Course Code: CSS2266	**Course Title: Theory of Computation Type of Course: Theory Course (PCC)**	**L- T- P- C**	**3-0-0-3**
Version No.	2.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course introduces the theoretical foundations of computation through the study of formal languages and automata. It covers the design and analysis of finite automata and regular languages, context-free grammar and pushdown automata, and Turing machines as models of computation. The course further explores key concepts such as ambiguity, normal forms, computability, decidability, and complexity theory including classes like P and NP. It also highlights the relevance of Theory of Computation in modern computing domains such as artificial intelligence, natural language processing, cybersecurity, compilers, and intelligent systems.		
Course Objective	To build a strong foundation in formal languages and automata by enabling students to model and analyze computational systems using finite automata, pushdown automata, and Turing machines; to evaluate computability and complexity of problems; and to apply theoretical concepts to modern domains such as AI, cybersecurity, NLP, and		

	intelligent systems.
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain fundamental concepts of automata, formal languages, and finite state machines. — Level 2-Understand **CO2:** Design and analyze deterministic and nondeterministic finite automata and regular expressions for language recognition. — Level 3-Apply **CO3:** Construct and evaluate context-free grammar and pushdown automata, including ambiguity removal and normal forms. — Level 3-Apply **CO4:** Design Turing machines and interpret their role in solving computational problems and language recognition. — Level 3-Apply **CO5:** Explain the applications of Theory of Computation concepts in real-world domains such as AI/ML, NLP, cybersecurity, compilers, and robotics. — Level 2-Understand

Course Content:

Module 1	**Introduction to automata theory**	**7 Sessions**

Topics:
Introduction to Automata Theory, Applications of Automata Theory, Alphabets, Strings, Languages & operations on languages, Representation of automata, Language recognizers, Finite State Machines (FSM): Deterministic FSM, Regular languages, Designing FSM, Nondeterministic FSMs, Basic concepts of Finite automata, DFA- definitions of DFA, Deterministic Accepters Transition Graphs and Languages, and DFA's

Module 2	**Finite Automata and Regular languages**	**12 Sessions**

Topics:
Regular Languages, NFA- Definition of a Nondeterministic Accepter, Languages and NFA's Why Non-determinism? Equivalence of Deterministic and Nondeterministic Finite Accepters, Reduction of the Number of States in Finite Automata. Formal Definition of a Regular Expression, Languages Associated with Regular Expressions, Languages, Regular Languages (RL) and Non-regular Languages: Closure properties of RLs, to show some languages are not RLs, Closure Properties of Regular

Module 3	**Context Free Grammar and Pushdown automata**	**12 Sessions**

Topics:
Context Free Grammars-Examples of Context-Free Languages, Leftmost and Rightmost Derivations, Derivation Trees, Relation Between Sentential Forms and Derivation Trees, Ambiguity in Grammars and Languages: Ambiguous Grammars, Removing Ambiguity, Chomsky Normal Form, Gribiche Normal Form. Definition of a Pushdown Automaton, Language Accepted by a Pushdown Automaton, Acceptance by Final State, Acceptance by Empty Stack, From Empty Stack to Final State, From Final State to Empty Stack Equivalence of PDA's and CFG's: From Grammars to Pushdown Automata.

Module 4	**Turing Machines**	**06 Sessions**

Topics:
Definition of a Turing Machine, Turing Machines as Language Accepters, Example Languages to construct Turing machine, Turing Machines as Transducers, Halting Programming Techniques for Turing Machines

Module 5	**Complexity and Modern Applications of ToC**	**08 Sessions**

Topics:
Complexity Theory - Time and Space Complexity, Complexity Classes: P, NP, NP-Complete (introductory), Intractability and Hard Problems-Applications of ToC in AI/ML/DL - Computability and Learnability (basic idea), Role of algorithms and complexity in ML, Expressive power of neural networks, NLP & Compilers -

Regular Languages in Tokenization, CFG in Syntax Parsing, Ambiguity in Natural Language Processing- Cybersecurity & Blockchain- Hardness assumptions in cryptography, Automata in intrusion detection systems, Smart contracts as state machines, Decidability limits in security
Robotics & Intelligent Systems - Finite State Machines in robot control, Planning as a computational problem, Real-time decision constraints

Targeted Application & Tools that can be used:

Targeted Application:

1. Text Processing
2. Compilers
3. Text Editors
4. Robotics Applications
5. Artificial Intelligence

Tools:

1. JFLAP (Java Formal Language and Automata Package) Software simulation tool. Its interactive educational software written in Java to experiment topics in automata theory.
2. Turing machine Online simulators.

Text Book(s):

1. Peter Linz, "An introduction to Formal Languages and Automata", Jones and Bartlett Publications 6th Ed, 2018.

2. Michael Sipser, "Introduction to the Theory of Computation", Cengage Learning, 3rd Ed,2012

Reference(s):

1. Aho, Ullman and Hopcroft, "Theory of Computation", Pearson India 3rd Edition 2008.
2. Michael Sipser, "Theory of Computation", Cengage India 3rd Ed, 2014.

E Resources

1. NPTEL
2. Opencourseware
3. TOC Notes

Course Code: CAI2502	**Course Name: Deep Learning** **Type of Course: Theory Course**	**L- T-P- C**	**3-0-0-3**
Version No.	1.0		
NCrF Level-	5		
Course Pre-requisites	CAI2500		
Co-requisites	CAI2503		
Anti-requisites	CSEXXXX		
Course Description	This course explores deep learning principles through neural network architectures, optimization strategies, and advanced model development. It covers CNNs, RNNs, GANs, and VAEs for vision, sequence modeling, and NLP. Emphasis lies on real-world applications, deployment tools, ethical practices, and capstone projects that deliver end-to-end deep learning solutions.		
Course Objective	The objective of the course is to familiarize the learners with the concepts of Deep Learning Techniques and attain Skill Development through Participative Learning techniques.		
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:**Explain key components of neural networks, including perceptrons, activation functions, and backpropagation algorithms. [Level 2 - Understand] **CO2:**Explain optimization strategies, regularization techniques, and model diagnostics used in deep learning. [Level 2 - Understand] **CO3:**Apply CNNs and RNNs for solving vision and sequential data problems using deep supervised learning techniques. [Level 3 - Apply] **CO4:**Analyze deep generative models such as Autoencoders, GANs, and VAEs for unsupervised learning tasks. [Level 4 - Analyze] **CO5:**Implement real-world deep learning solutions using advanced architectures and deployment tools while considering ethical implications. [Level 3 - Apply]		
Course Content:			
Module 1	Foundations of Deep Learning	Assessment: Assignment	12 Sessions
Topics: Fundamentals of Deep Learning and Artificial Neural Networks ,Perceptron and Multi-Layer Perceptrons (MLPs) , Feedforward Neural Networks: structure and forward pass , Activation Functions: Sigmoid, Tanh, ReLU, Leaky ReLU , Loss Functions: MSE, Cross-Entropy , Gradient Descent and Variants (SGD, Momentum) ,BackpropagationAlgorithm , Understanding Vanishing/Exploding Gradients, Building a neural network from			

scratch using NumPy			
Module 2	Optimization, Regularization & Model Improvement	Assessment: Assignment	11 Sessions
Topics: Weight Initialization Strategies: Xavier, He, Overfitting vs. Underfitting: concepts and mitigation ,Regularization Techniques: L1, L2, Dropout ,Batch Normalization: mechanism and impact ,Optimizers: Adam, RMSprop, AdaGrad ,Learning Rate Scheduling ,Early stopping and model checkpointing ,TensorBoard for model diagnostics			
Module 3	Deep Supervised Learning for Vision and Sequence	Assessment: Assignment	11 Sessions
Topics: Convolutional Neural Networks (CNNs): convolution, pooling, filters ,CNN Architectures: LeNet, AlexNet, VGG, ResNet ,Transfer Learning and Fine-Tuning with Pre-trained Models (e.g., VGG, ResNet) ,Recurrent Neural Networks (RNNs): Vanishing Gradient Issue ,Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs) ,Attention Mechanism Basics ,Use Cases in Image Classification and Time-Series Prediction			
Module 4	Deep Unsupervised and Generative Models	Assessment: Assignment	11 Sessions
Topics: Autoencoders: vanilla, denoising, and sparse ,Restricted Boltzmann Machines (RBMs) ,Deep Belief Networks (DBNs) ,Kohonen Self-Organizing Maps (SOMs) ,Hopfield Networks and associative memory ,Generative Adversarial Networks (GANs): architecture, loss, training challenges ,Variational Autoencoders (VAEs) ,Applications of generative models: synthetic data, deepfakes, style transfer			
Targeted Applications & Tools:			
Tools / Software / Platform	Application		
Google Colab	Cloud-based Python environment with free GPU/TPU support. Ideal for training CNNs, RNNs, GANs, and running large DL models without local resource constraints.		
Anaconda	Integrated platform to manage Python environments and deep learning packages (e.g.,TensorFlow, Keras, PyTorch). Used for offline practice and deployment.		
Spyder IDE	Professional IDE bundled in Anaconda, suitable for debugging and developing deep learning models using libraries likeNumPy,Scikit-learn, andTensorFlow.		
TensorFlow	Industry-standard deep learning framework for building CNNs, RNNs,Autoencoders, GANs, and deploying models viaTensorFlowLite or ONNX.		
Keras (within TensorFlow)	High-level API for quick prototyping and implementation of deep neural		

	networks like MLPs, LSTMs, CNNs, and transfer learning models.
PyTorch	Alternative DL framework used for research and dynamic model building. Especially useful for experiments in attention mechanisms, VAEs, and transformers.

Skill-Building Activities through Experiential Learning:

- Building Deep Neural Networks from Scratch
  Through hands-on coding assignments using NumPy and PyTorch/TensorFlow, learners construct feedforward neural networks, implement backpropagation, and experiment with activation and loss functions. This reinforces theoretical understanding while fostering naming conventions, modular programming, and documentation practices critical for real-world ML/DL project development.

- Model Optimization and Diagnostic Visualization
  Skill development is enhanced by tuning model performance using regularization techniques (Dropout, L1/L2), weight initialization, and optimizers (Adam, RMSprop). Learners use TensorBoard to visualize training logs and perform diagnostics—offering experiential exposure to debugging deep models in real-time environments.

- Real-Time Data Analysis with CNNs and RNNs
  Learners work on real-time applications such as image classification, sequence prediction, and time-series forecasting using CNNs, RNNs, and LSTMs. Tasks include importing datasets (e.g., CIFAR-10, IMDB), preprocessing pipelines, and live inference simulations—building industry-relevant skills in data pipeline design and model deployment readiness.

Text Book

T1: Goodfellow, I., Bengio, Y., Courville, A. (2023). Deep Learning. Germany: Alanna Maldonado.

References

R1: Duda, R.O., Hart, P.E., and Stork, D.G. Pattern Classification. Wiley-Inderscience, 2nd Edition. 2013

R2: Theodoridis, S. and Koutroumbas, K. Pattern Recognition. Edition 4, Academic Press, 2015

R3:Russell, S. and Norvig, N. Artificial Intelligence: A Modern Approach. Prentice Hall Series in Artificial Intelligence, 2013

R4: Bishop, C. M. Neural Networks for Pattern Recognition, Oxford University Press, 2008.

E-Resources/Web Resources

https://sm-nitk.vlabs.ac.in/

17 Best Deep Learning Courses Online in 2025 [Free + Paid]

Top Deep Learning Courses Online - Updated [June 2025]

Course Code: **CAI2503**	**Course Name: Deep Learning Lab** **Type of Course: Lab Course**	**L- T-P- C**	**0-0-4-2**
Version No.	1.0		
NCrF Level-	5		
Course Pre-requisites	CAI2500		
Co-requisites	CAI2502		
Anti-requisites	CSEXXXX		
Course Description	This lab-based course builds practical skills in designing, implementing, and evaluating deep learning models using TensorFlow and Keras. It covers CNNs, RNNs, GANs, and Transformer-based architectures, along with transfer learning, attention mechanisms, and deep reinforcement learning. Experiments emphasize applications across image, text, and time-series datasets in real-world contexts.		
Course Objective	The objective of the course is to familiarize the learners with the concepts of Deep Learning Techniques and attain Skill Development through Participative Learning techniques.		
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Identify key deep learning frameworks such as TensorFlow and Keras to build and visualize simple neural network models. [Level 2 – Understand] **CO2:** Describe the structure and functionality of various neural network architectures, including ANN, CNN, RNN, LSTM, and GRU. [Level 2 – Understand] **CO3:** Apply suitable model architectures and training techniques to solve supervised and unsupervised learning problems using real-world datasets. [Level 3 – Apply] **CO4:** Implement transfer learning, attention mechanisms, and sequence models for image and NLP-based tasks. [Level 3 – Apply] **CO5:**Build and test reinforcement learning agents and generative models such as Autoencoders and GANs to solve dynamic and synthetic data generation tasks. [Level 3 – Apply]		

Course Content:

Experiment No 1: Perceptron: Binary Classification from Scratch

Level 1: Implement a perceptron and train it on a toy dataset for binary classification

Experiment No 2: Multilayer Perceptron (MLP) for MNIST Classification

Level 1: Build and train a simple fully connected network on the MNIST dataset

Level 2: Design and fine-tune an MLP for MNIST dataset using suitable activation functions and optimizers.

Experiment No 3: Activation Functions & Their Effects on MLP Performance

Level 1: Compare Sigmoid, Tanh, ReLU, and LeakyReLU activations on MNIST classification

Level 2: Design and fine-tune an MLP for speech recognition using suitable activation functions and optimizers.

Experiment No 4: Loss Functions & Optimization Basics

Level 1: Compare Cross-Entropy vs MSE and SGD vs Adam on MNIST classification

Level 2: Design and implement different loss functions and optimization techniques

Experiment No 5: Backpropagation from Scratch (NumPy)

Level 1: Implement forward and backward propagation manually for a 1-hidden-layer NN

Level 2: Design forward and backward propagation manually

Experiment No 6: Hyper parameter Tuning in Neural Networks

Level 1: Compare the effect of learning rate, batch size, and hidden units for MNIST dataset

Level 2: Implement NN for hyper parameter comparison

Experiment No 7: Regularization in Neural Networks

Level 1: Compare L1, L2, and Dropout regularization in MLP on MNIST

Level 2: Design and implement L1, L2, and Dropout regularization in MLP on MNIST

Experiment No 8: Batch Normalization in Neural Networks

Level 1: Compare training of MLP with and without Batch Normalization

Level 2: implement and build MLP for batch normalization

Experiment No 9: Overfitting Demonstration & Early Stopping

Level 1: Show overfitting on small data and how EarlyStopping helps

Level 2: Build model to Show overfitting on small data and how EarlyStopping helps for model performance

Experiment No 10: Vanishing & Exploding Gradients Demonstration

Level 1: Show how deep networks suffer from vanishing/exploding gradients and how activation choice impacts this.

Level 2: Build MLP model for demonstration

Experiment No 11: Introduction to CNN for Image Classification

Level 1: Build a CNN architecture for image classification using CIFAR-10 dataset

Level 2: Perform hyperparameter tuning to enhance the CNN model's accuracy and generalization.

Experiment No 12: Transfer Learning with CNNs (VGG16)

Level 1: Use pre-trained VGG16 for CIFAR-10 classification via transfer learning

Level 2: Implement VGG16 for CIFAR-10 classification via transfer learning

Experiment No 13: DenseNet for Image Classification

Level 1: Use pre-trained DenseNet for CIFAR-10 classification via transfer learning

Level 2: Implement DenseNet for CIFAR-10 classification via transfer learning

Experiment No 14: PixelNet-like CNN for CIFAR-10 Classification

Level 1: Use PixelNet-like CNN for CIFAR-10 classification

Level 2: Implement PixelNet-like CNN for CIFAR-10 classification

Experiment No 15: ResNet from Scratch for CIFAR-10 Classification

Level 1: Use pre-trained ResNet for CIFAR-10 classification via transfer learning

Level 2: Implement ResNet for CIFAR-10 classification via transfer learning

Experiment No 16: RNN for Sentiment Analysis (IMDB Dataset)

Level 1: Build an RNN model for time-series prediction and sentiment analysis using the IMDB dataset.

Level 2: Implement RNN model for sentimental analysis

Experiment No 17: LSTM for Sentiment Analysis (IMDB Dataset)

Level 1: Build an LSTM model for time-series prediction and sentiment analysis using the IMDB dataset.

Experiment No 18: GRU for Sentiment Analysis (IMDB Dataset)

Level 1: Develop a GRU-based model for time-series sentiment prediction on the IMDB dataset and compare its performance with LSTM.

Experiment No 19: Generative Adversarial Network (GAN) for MNIST

Level 1: Design and train a Generative Adversarial Network (GAN) to generate small color images based on the MNIST dataset.

Experiment No 20: Conditional GAN (cGAN) for MNIST

Level 1: Design a GAN architecture for realistic image generation.

Level 2: Extend the GAN to build an age prediction model from facial images or MNIST Dataset

Experiment No 21: Kohonen Self-Organizing Map (SOM) for Unsupervised Clustering

Level 1: Implement a SOM to cluster the Iris dataset

Experiment No 22: Autoencoder for Dimensionality Reduction and Reconstruction

Level 1: Construct an encoder-decoder Recurrent Neural Network model for Neural Machine Translation and evaluate its decoding performance.

Experiment No 23: Convolutional Autoencoder for MNIST Image Reconstruction

Level 1: Implement a convolutional autoencoder for image reconstruction

Level 2: Build convolutional autoencoder model for MNIST dataset

Experiment No 24: Variational Autoencoder (VAE) for MNIST

Level 1: Implement a VAE for MNIST digit reconstruction and generation

Level 2: Design and compare convolutional autoencoder and Variational Autoencoder

Experiment No 25: Restricted Boltzmann Machine (RBM) with Logistic Regression for Classification

Level 1: Implement an RBM to learn features from MNIST and use them for classification

Level 2: Build RBM model for MNIST dataset

Experiment No 26: Deep Belief Network (DBN) for Digit Classification

Level 1: Train a DBN on MNIST for handwritten digit recognition

Level 2: Implement DBN for MNIST dataset

Experiment No 27: Hopfield Network for Pattern Storage and Retrieval

Level 1: Implement a Hopfield network to store and retrieve binary patterns

Experiment No 28: Probabilistic Neural Network (PNN) for Classification

Level 1: Implement a PNN to classify Iris flowers

Level 2: Build Probabilistic Neural Network

Experiment No 29: Deep Q-Network (DQN) for CartPole

Level 1: Implement a basic Deep Q-Network (DQN) for the CartPole environment using TensorFlow.

Level 2: Extend the DQN agent for more complex environments like Atari Breakout.

Experiment No 30: Attention Model for Image Captioning

Level 1: Implement an image captioning system with Bahdanau attention

Targeted Applications & Tools:

Tools/Software/Platform	Application
TensorFlow	Designing, training, and deploying deep learning models (ANN, CNN, RNN, LSTM, GAN, etc.)
Keras	High-level API for building and training deep learning models with minimal code
Google Colab / Jupyter Notebook	Cloud-based or local environments for executing deep learning experiments interactively
Matplotlib / Seaborn	Data visualization for analyzing model performance and

	interpreting training outcomes
Hugging Face Transformers	Fine-tuning and deploying transformer-based models like BERT for NLP and sequence-to-sequence tasks
OpenAI Gym / Gymnasium	Simulation environments for training and evaluating reinforcement learning agents

Skill-Building Activities through Experiential Learning:

- Generative Modeling with Autoencoders and GANs

  Hands-on implementation of GANs and Autoencoders introduces learners to generative modeling. They design models to create realistic images, synthesize features like age prediction from faces, and reconstruct inputs using encoder-decoder architectures.

- Deep Reinforcement Learning Simulations

  Using environments like OpenAI Gym, learners build reinforcement learning agents using DQNs to solve problems like CartPole balancing and Atari games. They experiment with reward functions, exploration strategies, and model stability, connecting deep learning to dynamic decision-making.

- Comparative Model Analysis and Optimization

  For each experiment, students are required to compare performance across architectures (e.g., CNN vs. ResNet, LSTM vs. GRU) using metrics like accuracy, loss curves, and confusion matrices. This reinforces critical thinking and encourages model interpretability and diagnostic analysis.

Text Book

T1: François Chollet, "Deep Learning with Python", 2nd Edition, Manning Publications, 2022

T2: Ian Goodfellow, Yoshua Bengio, Aaron Courville, "Deep Learning", MIT Press, 2017.

References

R1: Amlan Chakrabarti Amit Kumar Das, Saptarsi Goswami, Pabitra Mitra , "Deep Learning", Pearson Publication, 2021.
R2: David Foster, "Generative Deep Learning" O'Reilly Publishers, 2020.
R3:John D Kellehar, "Deep Learning", MIT Press, 2020.

E-Resources/Web Resources

IEEE Transactions on Neural Networks and Learning Systems

https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=5962385

IEEE Transactions on Pattern Analysis and Machine Intelligence

https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=34http://ijaerd.com/papers/special_papers/IT032.pdf

International Journal of Intelligent Systems https://onlinelibrary.wiley.com/journal/1098111x

Course Code: CAI2512	**Course Name: Neural Networks and Fuzzy Logic** **Type of Course: Theory Course**	**L- T-P- C**	**3-0-0-3**
Version No.	1.0		
NCrF Level-	5.0		
Course Pre-requisites	CAI2500		
Co-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course introduces foundational concepts in Neural Networks and Fuzzy Logic, emphasizing their roles in pattern recognition, learning, and decision-making. It explores how neural networks mimic brain-like processing and how fuzzy logic captures human reasoning to manage uncertainty and imprecision in intelligent systems and real-world applications.		
Course Objective	The objective of the course is tofamiliarize the learners with the concepts ofNeural Networks and Fuzzy Logicand is designed to improve the learners' EMPLOYABILITY SKILLS through PROBLEM SOLVING METHODOLOGIES.		
Course Outcomes	On successful completion of the course, learners will be able to: CO1:Demonstrate a clear understanding of the basic principles of Neural Networks and their biological inspiration. — Level 2 – Understand CO2:Apply common learning algorithms such as Perceptron and Backpropagation in Neural Networks. — Level 3 – Apply CO3:Explain the concepts of Fuzzy Sets, Membership Functions, and Fuzzy Relations. — Level 2 – Understand CO4:Apply Fuzzy Logic Controllers for decision-making in uncertain environments. — Level 3 – Apply CO5:Apply Neural Network and Fuzzy Logic techniques to design intelligent solutions for real-world problems. — Level 3 – Apply		
Course Content:			
Module 1	Introduction to Neural Networks	Assignment	12 Sessions
Topics: Single Layer Perceptron Fundamentals of Neural Networks: Overview of biological and artificial neural networks, connection with artificial intelligence Neuron Models: Structure and functioning of biological neurons, mathematical models of single artificial neurons, activation functions Single Layer Perceptron: Architecture, training with Least Mean Square (LMS) algorithm, role of learning rate Introduction to competitive learning and basic perceptron models			
Module 2	Multilayer Neural Networks	Assignment	11 Sessions
Topics: Multilayer Perceptron Multilayer Perceptron (MLP): Solving non-linearly separable problems (e.g., XOR/XNOR) using backpropagation Heuristics to improve learning performance Practical applications of MLP Radial Basis Function (RBF) Networks: Concepts of interpolation, regularization, and learning strategies Kohonen Self-Organizing Maps (SOM): Unsupervised learning principles, SOM algorithm, and Learning Vector Quantization (LVQ)			

Module 3	Fuzzy Sets, Operations, and Relations	Assignment	11 Sessions

Topics:
Fuzzy Operations
Fuzzy Set Theory: Differences between crisp and fuzzy sets, α-cuts and their properties, membership functions, extension principles
Operations on Fuzzy Sets: Fuzzy complements, intersections, unions, combinations, and aggregation techniques
Fuzzy Relations: Binary fuzzy relations, equivalence, and compatibility relations in fuzzy logic

Module 4	Fuzzy Logic Systems	Assignment	11 Sessions

Topics:
Fuzzy Logic and Control
Overview of Classical and Multivalued Logic: Fuzzy propositions, quantifiers, linguistic hedges, conditional and qualified propositions
Fuzzy Inference Mechanisms: Components of fuzzy control systems — fuzzification, rule base, inference engine, defuzzification methods
Fuzzy Controller Design: Stepwise design and implementation illustrated with real-world examples

Targeted Applications & Tools:

Tools/Software/Platform	Application
Python (NumPy, Pandas)	Data preprocessing, neural network computation, and fuzzy logic implementation
Scikit-learn	Implementing perceptron, multilayer perceptron, and model evaluation techniques
TensorFlow / Keras	Designing and training neural networks including MLP, RBF, and deep learning models
MATLAB / Octave	Simulating fuzzy inference systems and designing fuzzy logic controllers
SciKit-Fuzzy (skfuzzy)	Creating membership functions, fuzzy rules, and building fuzzy decision systems

Skill-Building Activities through Experiential Learning:

Scheduling Algorithm Visualization using C/Python

- Learners design and implement CPU scheduling algorithms (FCFS, SJF, Round Robin, Priority) to compute metrics like turnaround time and waiting time. Results are visualized using Gantt charts, reinforcing performance trade-offs in real-world scheduling decisions.

- **Concurrency and Synchronization with POSIX Threads**
  Through multi-threaded C programs, students solve classic synchronization problems like Producer-Consumer and Dining Philosophers using semaphores and mutexes. This enables them to observe and manage race conditions, deadlocks, and thread synchronization issues.

- **Deadlock Handling through Resource Allocation Simulation**
  A matrix-based simulation models deadlock detection using the Banker's Algorithm. Students create resource allocation graphs to identify unsafe states and implement recovery strategies, reinforcing concepts of deadlock prevention and avoidance.

Text Book

T1: Haykin, Simon. "*Neural networks and learning machines"*, 3/E. Pearson Education India, 2020.
https://www.pearson.com/en-us/subject-catalog/p/Haykin-Neural-Networks-and-Learning-Machines-3rd-Edition/P200000003278/9780133002553
T2: George J. Klir and Bo Yuan, "*Fuzzy Sets and Fuzzy Logic- Theory and Applications*", Prentice Hall of India, 2015.
https://www.worldcat.org/title/fuzzy-sets-and-fuzzy-logic-theory-and-applications/oclc/505215200

References

R1: Shivanandam, Deepa S, "Principles of Soft computing", N Wiley India, 3rd Edition, 2018.https://www.wileyindia.com/principles-of-soft-computing-3ed.html
R2: Timothy J. Ross, "Fuzzy Logic with Engineering Applications", fourth Edition, Wiley, 2016. https://www.wiley-vch.de/en/areas-interest/engineering/fuzzy-logic-with-engineering-applications-978-1-119-23586-6
R3: Kumar S., "Neural Networks - A Classroom Approach", Tata McGraw Hill, 2nd Edition 2017.https://www.worldcat.org/title/neural-networks-a-classroom-approach/oclc/56955342
R4: Fakhreddine O. Karray, and Clarence W. De Silva. "Soft computing and intelligent systems design: theory, tools, and applications". Pearson Education, 2009.

E-Resources/Web Resources

1. https://www.pearson.com/en-gb/search.html?q=Karray%20Soft-Computing-and-Intelligent-Systems-Design-Theory-Tools-and-Applications
2. Fuzzy Logic and Neural Networks - Course

Course Code: **Course Code: CAI2509**	**Course Title: Web Technologies using AI** **Type of Course : Theory Course (PCC)**	**L-T-P- C**	**3 – 0 – 0 - 3**
Version No.	3.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course introduces modern web technologies integrated with AI-assisted development practices for designing intelligent and responsive web applications. It covers XHTML, CSS, JavaScript, PHP, MySQL, web security, and intelligent web architectures, emphasizing AI-driven code generation, UI optimization, database integration, and responsible web engineering practices.		
Course Objective	On successful completion of the course, the students shall be able to develop employability skills in web application development through AI-assisted and experiential learning techniques using modern web technologies, scripting frameworks, database integration, and intelligent web services.		
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain the architecture of the web, XHTML structures, HTTP protocols, and AI-assisted web development concepts. — L2 – Understand **CO2:** Apply CSS styling, layout techniques, responsive design principles, and AI-assisted UI generation methods for modern web interfaces. — L3 – Apply		

	CO3: Develop interactive client-side web applications using JavaScript, DOM manipulation, event handling, and form validation techniques.	L3 – Apply
	CO4: Develop server-side web applications using PHP, MySQL, SQL queries, and database integration techniques.	L3 – Apply
	CO5: Analyze web application security, intelligent web architectures, ethical AI practices, and responsible deployment techniques in modern web systems.	L4 – Analyze
Course Content:		
Module 1	**AI-Scaffolded Web Architectures & XHTML Foundations**	**10 Sessions**
Core Topics: Infrastructure of the Web: WWW, web browsers, HTTP/HTTPS protocols, web servers, and the Internet. Origins and Evolution of HTML/XHTML: Document Object Model syntax, structural validation, standard document schemas, basic text components, hyperlinking structures, lists, data tables, structural frames, and semantic forms. **AI Integration:** Utilizing AI tools (like Cursor/VS Code) for rapid boilerplate generation; Prompt engineering techniques for layout structure and inline technical documentation.		
Module 2	**Enhanced Styling, Layouts & Generative Design Frameworks**	**10 Sessions**
Core Topics: Cascading Style Sheets (CSS): Formatting cascading hierarchies, type selectors, pseudo-classes, and structural pseudo-elements. The Box Model: Margins, padding, borders, dimensional control, and visibility. Positioning Paradigms: Normal flow, absolute, relative, floating layouts, and responsive flex grid architectures. Overview of markup structures (XML basics). **AI Integration:** Using AI for automated color palette and typography generation;Using generative AI engines to translate text wireframes into semantic CSS grid configurations; AI-assisted refactoring of monolithic styles into modular variables**.**		
Module 3	**Computational Logic, Event Pipelines & Smart Interaction**	**10 Sessions**
Core Topics: JavaScript Scripting: Core paradigms, runtime mechanics, functions, parameters, custom objects, conditional switching, and execution loops. DOM Interfaces: Navigating nodes, editing elements, structural mutations, event registers, and operational event listening. Client-side form handling and validation patterns. **AI Integration:** AI-guided optimization of asynchronous tasks; Code completion tools for predicting complex event handlers and structural DOM tracking.		
Module 4	**Intelligence-Driven Server Logic & Persistence Architecture**	**08 Sessions**
Core Topics: PHP Server Ecosystem: Operational processing lifecycle, state collection arrays ($GET, $POST), superglobals, and active file manipulation. Object-Oriented PHP: Classes, instances, data modeling, and inheritance. Relational Data Management: Structural Query Language (SQL) essentials, database API		

integrations, processing active connection states, and structural query pipelines using MySQL.

AI Integration: AI-powered schema optimization models; Utilizing large language models to format multi-table SQL queries and discover complex data relations.

Module 5	**Defenses, Cognitive Services & Responsible Engineering**	**07 Sessions**

Core Topics: Web Security Ecosystem: Cross-Site Scripting (XSS) defenses, SQL Injection preventions, safe input parsing, and state management via session authorization. Intelligent Architectures: Embedding semantic web rules, conversational chatbot architectures, context-aware interface models, and intelligent agent integrations.

AI Integration: AI-Ops: Automated code auditing tools for scanning structural security flaws; Analyzing bias, user privacy vectors, and responsible boundaries in public web applications.

Targeted Application & Tools that can be used:

Visual Studio Code / Cursor IDE (AI Extensions enabled).

XAMPP development stack (Apache Server & MySQL Engine).

Project work/Assignment:

Assignments are given after completion of each module which the student need to submit within the stipulated deadline.

Textbook(s):

T1: Robert. W. Sebesta, "Programming the World Wide Web", Pearson Education, 8th Edition,2015.

T2: Jon Duckett – *HTML & CSS: Design and Build Web Sites* and *JavaScript & jQuery*, Wiley Publications, **2014**.

T3: Marijn Haverbeke – *Eloquent JavaScript (4th Edition)*, No Starch Press, 2024.

T4: Deitel, Deitel, Goldberg,"Internet & World Wide Web How to Program", Fifth Edition, Pearson Education,2021.

References:

R1. Randy Connolly, Ricardo Hoar,"Fundamentals of Web Development", Pearson Education India,1st. Edition.2016.

R2. Jeffrey C. Jackson,"Web Technologies: A Computer Science Perspective", Pearson Education, 1st Edition,2016.

R3: David Flanagan – *JavaScript: The Definitive Guide*, **7th Edition**, O'Reilly, **2020**

Topics related to development of "FOUNDATION": Web, WWW, Web browsers, Web servers, Internet. CSS, PHP. Designing for healthcare. **For Skill Development through Experiential Learning techniques.** This is attained through assessment component mentioned in course handout. **E-References:** • https://developer.mozilla.org • https://www.w3schools.com • https://platform.openai.com • https://getbootstrap.com

Course Code: CAI2510	**Course Title: Web Technologies using AI Lab** **Type of Course: Laboratory Course (PCC)**	**L- T-P- C**	**0-0-2-1**
Version No.	3.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	This laboratory course provides hands-on experience in developing modern web applications using XHTML, CSS, JavaScript, PHP, MySQL, and AI-assisted development tools. The course emphasizes responsive web design, client-side scripting, server-side processing, database integration, intelligent automation, and responsible AI-enabled web application development through practical experimentation.		
Course Objective	On successful completion of the course, the students shall be able to develop practical skills in web application development through experiential learning and AI-assisted programming techniques using modern web technologies, scripting tools, database integration, and intelligent web services.		
Course Outcomes	On successful completion of this course, students will be able to: CO1: Explain the fundamentals of XHTML document structures, web protocols, and AI-assisted web development tools. — L2 – Understand CO2: Develop responsive web layouts using CSS styling, positioning techniques, and responsive design principles. — L3 – Apply CO3: Develop client-side web applications using JavaScript, DOM manipulation, event handling, and form validation techniques. — L3 – Apply CO4: Develop server-side applications using PHP and integrate relational databases using MySQL. — L3 – Apply CO5: Analyze web application security, debugging workflows, AI-assisted code generation, and database integration techniques. — L4 – Analyze		

COURSE CONTENT:

List of Laboratory Tasks: (30 hours)

WEEK 1: Web Ecosystems & Document Structures

Lab Sheet 1: Introduction to XHTML & Structural Mockups

Level 1 : Create a basic XHTML document utilizing paragraphs, headers, and standard text formatting tags.

Level 2 : Use an AI assistant to scaffold a clean multi-page document layout and auto-generate structural comments.

WEEK 2: Structural Layout Components

Lab Sheet 2: Lists, Hyperlinks, and Data Tables

Level 1 : Design an XHTML page containing nested lists, external hyperlinks, and a clean structured table with rows and borders.

Level 2: Use AI to generate a raw dataset in XHTML table format from a plain-text prompt description.

WEEK 3: User Interaction Interfaces

Lab Sheet 3: Interactive Web Forms

Level 1 : Create a standard registration form utilizing input fields, radio buttons, drop-down menus, and submission buttons.

Level 2 : Use AI to review the form layout and auto-suggest proper input names, field matching, and placeholder options.

WEEK 4: Core Style Paradigms

Lab Sheet 4: CSS Implementation & Basic Selectors

Level 1: Implement an external CSS stylesheet defining font properties, text colors, and background treatments using element, class, and ID selectors.

Level 2 : Use AI to generate an optimized color palette and consistent typography scale mapped directly to native CSS variables.

WEEK 5: Element Dimensionality & Space

Lab Sheet 5: The CSS Box Model Application

Level 1 : Build a content webpage to demonstrate the interactive behavior of margins, padding, content widths, and visible borders.

Level 2 : Use AI to troubleshoot and resolve a pre-built broken layout affected by padding over-allocations

and margin collapse defects.

WEEK 6: Layout Architectures

Lab Sheet 6: Positioning, Flows, and Floats

Level 1: Group content cards into a clear layout structure using CSS float properties and absolute/relative positioning parameters.

Level 2 : Provide a raw text description or wireframe layout to an AI tool to auto-generate structural layout properties.

WEEK 7: Modern Flexible Layouts

Lab Sheet 7: Responsive Designs via Grid Layouts

Level 1 : Implement media queries to construct a fluid grid layout that shifts responsively between desktop displays and mobile viewports.

Level 2 : Provide an interface mockup image to an AI Vision model to auto-generate the matching raw CSS structural code.

WEEK 8: Structured Text Transformations

Lab Sheet 8: XML Documents & Schema Trees

Level 1 : Build a well-formed XML document designed to house a clear structured student information index.

Level 2 : Use AI tools to validate the data structure tree and automatically convert the XML payload nodes into matching readable XHTML lists.

WEEK 9: Client-Side Runtime Implementations

Lab Sheet 9: JavaScript Logic & Execution Streams

Level 1 : Write basic script tracks executing numeric calculations, variable switches, and basic conditional loops within a browser view.

Level 2 : Paste raw algorithmic goals into an AI code generator to instantly produce clean, clean function routines.

WEEK 10: Dynamic Document Manipulations

Lab Sheet 10: DOM Node Tree Management

Level 1: Develop a script structure that actively selects, appends, and replaces text nodes inside live XHTML elements using standard DOM actions.

Level 2 : Use AI to auto-generate a lightweight component array that lets users append list entries via form triggers.

WEEK 11: Interface Event Listening

Lab Sheet 11: Browser Event Processing & Integrity Filters

Level 1 : Map interaction responses to standard click, focus, and hover actions across input containers.

Level 2 : Use AI-assisted debugging tools to step through script errors, handle exception loops, and test event-listener tracking.

WEEK 12: Client-Side Input Safety Checks

Lab Sheet 12: Complete Client-Side Form Validations

Level 1 : Write an active input evaluation routine using regular expressions to check email syntax formats, password scales, and mandatory field matches.

Level 2 : Prompt an AI assistant to generate regex filters and formulate interactive alert windows detailing missing parameters.

WEEK 13: Server Runtime Engines

Lab Sheet 13: Server Processing with PHP Basics

Level 1: Configure a basic server environment script that captures and processes user input data via structural global tracking arrays ($_GET / $_POST).

Level 2 : Use AI to refactor monolithic procedural code structures into modular object-oriented classes and instances.

WEEK 14: Server-Side Relational Systems

Lab Sheet 14: Relational Persistent Integrations

Level 1 : Implement connection pathways linking operational server environments with relational databases using active PHP MySQL extensions.

Level 2 : Use AI to build clean database schemas and formulate relational SQL database query structures from natural language inputs.

WEEK 15: AI Integrations & Full-Stack Deployments

Lab Sheet 15: Mini Capstone – Cognitive Services Integration & Code Audits

Level 1 : Build and deploy a unified full-stack page that captures form values, writes transactions to a data

store, and reads confirmation logs.

Level 2 : Integrate a third-party GenAI API (OpenAI/Gemini) to process live inputs and use AI automated security scanners to audit structural file protections.

Targeted Application & Tools that can be used:

Visual Studio Code / Cursor IDE (AI Extensions enabled).

XAMPP Development Suite (Apache HTTP Server, MySQL Engine, PHP Runtime Environment).

Postman (API Testing Endpoint Interface).

Project work/Assignment:

Assignments are given after completion of each module which the student need to submit within the stipulated deadline.

Textbook(s):

T1. Robert. W. Sebesta, "Programming the World Wide Web", Pearson Education, 8th Edition,2015.

T2. *Generative AI for Full-Stack Development* (1st Edition, 2026), Shantanu Baruah (Wiley Publications)

T3. Deitel, Deitel, Goldberg,"Internet & World Wide Web How to Program", Fifth Edition, Pearson Education,2021.

References:

R1. Randy Connolly, Ricardo Hoar,"Fundamentals of Web Development", Pearson Education India,1st. Edition.2016.

R2. Jeffrey C. Jackson,"Web Technologies: A Computer Science Perspective", Pearson Education, 1st Edition,2016.

R3: *Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5* (6th Edition, Robin Nixon (O'Reilly Media)

E-References:

1. W3schools.com
2. Developer.mozilla.org/en-US/docs/Learn
3. docs.microsoft.com
4. informit.com/articles/ The Relationship Between Web 2.0 and Social Networking
5. https://presiuniv.knimbus.com/user#/home

Topics related to development of "FOUNDATION": Web, WWW, Web browsers, Web servers, Internet.

CSS, PHP. Designing for healthcare For Skill Development through Experiential Learning techniques. This is attained through assessment component mentioned in course handout.

Course Code: CSS2269	Course Name: Operating Systems Type of Course: Theory Course	L- T-P- C	3-0-0-3
Version No.	1.0		
Course Pre-requisites	NIL		
Co-requisites	CAI2501		
Anti-requisites	NIL		
Course Description	This course introduces the concepts of Operating Systems operations, Operating System structure and its design and implementation. It covers the Operating Systems internal algorithms such as process scheduling, synchronization, deadlocks detection and recovery and memory management. The course also enhances the problem solving, systems programming ability and case studies.		
Course Objective	To develop a comprehensive understanding of operating system principles by enabling students to analyze process management and scheduling algorithms, explore memory and virtual memory management, examine file systems and storage organization and evaluate synchronization mechanisms and deadlock handling techniques.		
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:**Explain the fundamental concepts of operating systems and relevant case studies. [Level 2 - Understand] **CO2**: Apply various CPU scheduling algorithms for process management. [Level 3 – Apply] **CO3:**Apply synchronization techniques to handle concurrency problems. [Level 3 – Apply] **CO4:**Apply deadlock detection and recovery methods in operating systems. [Level 3 – Apply] **CO5:** Explain various memory management techniques used in operating systems. [Level 3 -Apply]		
Course Content:			
Module 1	Introduction to Operating Systems	Assessment: Assignment (Programming using Keras/Sklearn)	9 Sessions
Topics: Introduction to OS , Operating-System Operations, Operating Systems Services, , System Calls and its types, Operating Systems Structure, System Program and its types, Linkers and Loaders, Overview of OS design and implementation, Open-source Operating Systems			
Module 2	Process Management	Assessment : Quiz + Assignment (Sklearn/Statsmodels)	12 Sessions

Topics:

Process Concept, Operations on Processes, Inter Process Communication, Communication in client-server systems (sockets, RPC, Pipes), Introduction to threads - Multithreading Models, Thread Libraries, Threading Issues, Process Scheduling– Basic concepts, Scheduling Criteria, Scheduling Algorithms: FCFS, SJF, SRTF, RR and Priority.

Module 3	**Process Synchronization and Deadlocks**	Assessment : Assignment (Programming using Keras/Sklearn)	12 Sessions

The Critical-Section Problem- Peterson's Solution, Synchronization hardware, Semaphores, Classic Problems of Synchronization with Semaphore Solution- Producer-Consumer Problem, Reader-Writer problems, Dining Philosopher's Problem, . Introduction to Deadlocks, Necessary conditions for deadlock, Resource allocation Graph, Methods for handling deadlock: Deadlock Prevention and Implementation, Deadlock Avoidance and Implementation, Deadlock detection & Recovery from Deadlock.

Module 4	Memory Management	Assessment : Programming Assignment (Keras/TensorFlow)	12 Sessions

Topics:

Introduction to Memory Management, Basic hardware-Base and Limit Registers, Memory Management Unit(MMU), Dynamic loading and linking, Swapping, Contiguous and Non-Contiguous Memory Allocation, Segmentation, Paging - Structure of the Page Table – Virtual Memory and Demand Paging – Page Faults and Page Replacement Algorithms, Copy-on-write, Allocation of Frames, Thrashing

Introduction to File system management: File System Interface (access methods, directory structures), File system implementation.

Project work/Assignment :

1. Demonstrate process concepts in LINUX OS.
2. Simulation of CPU scheduling algorithms.
3. Develop program to demonstrate use of Semaphores in threads.
4. Develop program to demonstrate use of deadlock avoidance algorithms.
5. Develop program to demonstrate use of page replacement algorithms.
6. Simulation of memory allocation strategies [first fit, best fit and worst fit].

Targeted Applications & Tools:

Use cases	Application area is traffic management system, banking system, health care and many more systems where in there are resources and entities that use and manage the resources.
Oracle Virtual Box/VMWare Virtualization software [Virtual Machine Managers]	Used to install and work on multiple guest Operating Systems on top of a host OS.
Intel Processor identification utility	This software is used to explain about multi-core processors. It helps to identify the specifications of your Intel processor, like no of cores, Chipset information, technologies supported by the processor etc.

Skill-Building Activities through Experiential Learning:

- Comparative Analysis of Ensemble Techniques
  Learners experiment with ensemble models like Random Forest, AdaBoost, and Gradient Boosting, conducting comparative studies on model variance, bias, and robustness. Through group assignments, students simulate real-world decision-making scenarios where ensemble strategies outperform individual models.

- Neural Network Construction Using Keras
  Using the Keras API, students build and train basic feedforward neural networks. They gain practical experience in selecting activation functions, configuring loss functions, tuning hyperparameters (e.g., learning rate, epochs), and applying backpropagation using optimizers like SGD and Adam.

- Unsupervised Learning for Pattern Discovery
  Through clustering algorithms such as K-Means, Hierarchical Clustering, DBSCAN, and Gaussian Mixture Models (GMM), learners segment real-world datasets and visualize clusters. They apply evaluation methods like the Elbow Method and Silhouette Score to validate clustering quality.

Text Book

T1: Silberschatz A, Galvin P B and Gagne G , "Silberschatz's Operating System Concepts", Paperback, Global Edition Wiley, 2019

References

R1: Silberschatz A, Galvin P B and Gagne G, "Operating System Concepts", 10th edition Wiley, 2018.

R2: William Stallings, "Operating Systems",Ninth Edition, By Pearson Paperback ,1 March 2018.
R3: Sundaram RMD, Shriram K V, Abhishek S N, B Chella Prabha, " Cracking the Operating System skills", Dreamtech, paperback, 2020.
R4: Remzi H. Arpaci-Dusseau Andrea C. Arpaci-dusseau , "Operating Systems: Three Easy Pieces, Amazon digital Services", September 2018.

E-Resources/Web Resources

1. https://www.os-book.com/OS9/
2. https://pages.cs.wisc.edu/~remzi/OSTEP/
3. https://codex.cs.yale.edu/avi/os-book/OS10/index.html

Course Code: CSS2270	Course Name: **Operating Systems Lab** Type of Course: Laboratory Course	L- T-P- C	0-0-2-1
Version No.	1.0		
NCrF Level-	5.0		
Course Pre-requisites	NIL		

Co-requisites	CAI2500
Anti-requisites	NIL
Course Description	This laboratory course provides hands-on experience with the core concepts of operating systems through practical assignments, simulations, and case studies. It covers foundational aspects such as system calls, process and thread management, inter-process communication, synchronization, deadlocks, memory management, and file systems. Students will implement and simulate real-time OS components and scheduling algorithms, fostering deeper understanding of OS architecture and design. The lab also introduces modern OS tools, programming interfaces, and the basics of open-source OS environments.
Course Objective	To develop practical skills in operating systems by enabling students to implement scheduling and memory management algorithms, simulate process synchronization and deadlock handling, and build system-level programming capabilities.
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:**Demonstrate system-level programming using system calls and operating system structures. [Level 3 – Apply] **CO2:**Simulate process scheduling and multithreading techniques. [Level 3 – Apply] **CO3:**Apply synchronization techniques using semaphores and shared memory to handle concurrency problems. [Level 3 – Apply] **CO4:**Demonstrate memory management and file system concepts using simulation or scripting tools. [Level 3 – Apply] **CO5:** Analyze the performance of operating system algorithms related to scheduling, synchronization and memory management. . [Level 4 – Analyze]

Course Content:

List of Laboratory Tasks:

- **Experiment 1: Introduction to OS**

  OS Basics

  Level 1: Write a program to demonstrate basic system calls (file operations like open, read, write).

  Level 2: Analyze different OS services with real-time examples.

  OS Structure

  Level 1: Illustrate OS structures (monolithic, layered, microkernel).

  Level 2: Compare open-source OS (Linux) with proprietary OS.

- **Experiment 2: System Parameters**

  Linkers & Loaders

  Level 1: Demonstrate compilation process (preprocessing, compiling, linking).

  Level 2: Analyze object files and executable generation.

  System Utilities

  Level 1: Write simple shell commands using system calls.

Level 2: Implement a mini shell program.

- **Experiment 3: Process Concept**

  Process Creation
  Level 1: Write a program using fork() system call.

  Level 2: Demonstrate parent-child communication.

  Process Operations
  Level 1: Implement process states and transitions.

  Level 2: Simulate process control block (PCB).

- **Experiment 4: Inter Process Communication**
  Pipes:

  Level 1: Implement communication using pipes.

  Level 2: Implement bidirectional communication using pipes.

  Sockets/RPC

  Level 1: Write a client-server program using sockets.

  Level 2: Simulate RPC communication.

- **Experiment 5: Threads**
  Thread Creation

  Level 1: Create threads using pthread library.

  Level 2: Compare multi-threading vs multi-processing.

  Thread Issues

  Level 1: Demonstrate thread synchronization issues.

  Level 2: Solve race condition problem.

- **Experiment 6: CPU Scheduling**
  FCFS & SJF

  Level 1: Implement FCFS scheduling.

  Level 2: Implement SJF and compare results.

  Advanced Scheduling

  Level 1: Implement Round Robin scheduling.

  Level 2: Implement Priority scheduling and analyze performance.

- **Experiment 7: Synchronization**

Critical Section

Level 1: Implement Peterson's solution.

Level 2: Analyze critical section problem.

Semaphores

Level 1: Implement semaphore operations

Level 2: Solve producer-consumer problem using semaphores.

- **Experiment 8: Classical Problems**
  Reader-Writer

  Level 1: Implement Reader-Writer problem.

  Level 2: Analyze starvation issues.

  Dining Philosophers

  Level 1: Implement Dining Philosophers problem.

  Level 2: Avoid deadlock using strategies.

- **Experiment 9: Dead Locks**
  Deadlock Detection

  Level 1: Implement Resource Allocation Graph.

  Level 2: Detect deadlock using algorithm.

  Deadlock Avoidance

  Level 1: Implement Banker's algorithm.

  Level 2: Analyze safe and unsafe states.

- **Experiment 10:Memory Management Basics**

  Memory Allocation

  Level 1: Implement First Fit, Best Fit, Worst Fit.

  Level 2: Compare memory utilization.

  Paging

  Level 1: Simulate paging mechanism.

  Level 2: Analyze page table structure.

- **Experiment 11: Virtual Memory**

  Page Replacement

Level 1: Implement FIFO page replacement.

Level 2: Implement LRU and Optimal algorithms.

Page Faults

Level 1: Calculate page faults.

Level 2: Analyze thrashing condition.

- **Experiment 12: Advanced Memory Management**

  Segmentation

  Level 1: Implement segmentation.

  Level 2: Compare segmentation vs paging.

  Demand Paging

  Level 1: Simulate demand paging.

  Level 2: Analyze copy-on-write mechanism.

- **Experiment 13: File System**

  File Operations

  Level 1: Implement file operations (create, read, write).

  Level 2: Simulate directory structure.

  File Allocation

  Level 1: Implement file allocation methods.

  Level 2: Compare allocation techniques.

- **Experiment 14: File System Implementation**

  File System Design

  Level 1: Design simple file system.

  Level 2: Analyze file system performance.

  Access Methods

  Level 1: Implement sequential access.

  Level 2: Implement indexed access.

- **Experiment 15: Integration & Simulation**

  OS Simulation

  Level 1: Simulate OS components (process + memory).

Level 2: Integrate scheduling and memory management.

Mini Project

Level 1: Implement OS concept-based project.

Level 2: Analyze performance and optimization.

Project Work/Assignments

1. Demonstrate process concepts in LINUX OS.
2. Simulation of CPU scheduling algorithms.
3. Develop program to demonstrate use of Semaphores in threads.
4. Develop program to demonstrate use of deadlock avoidance algorithms.
5. Develop program to demonstrate use of page replacement algorithms.
6. Simulation of memory allocation strategies [first fit, best fit and worst fit]

Targeted Applications & Tools:

Use cases	Application area is traffic management system, banking system, health care and many more systems where in there are resources and entities that use and manage the resources.
Oracle Virtual Box/VMWare Virtualization software [Virtual Machine Managers]	Used to install and work on multiple guest Operating Systems on top of a host OS.
Intel Processor identification utility	This software is used to explain about multi-core processors. It helps to identify the specifications of your Intel processor, like no of cores, Chipset information, technologies supported by the processor etc.

Skill-Building Activities through Experiential Learning:

- **Data Preprocessing with Real-World Datasets**
  Learners perform hands-on preprocessing of real-world datasets (e.g., UCI, Kaggle) by handling missing values, scaling features, and encoding categorical variables. This foundational step helps them understand the impact of clean data on model performance.

- **Exploratory Data Analysis and Visualization**
  Using libraries such as Matplotlib and Seaborn, students conduct exploratory data analysis through plots like scatter plots, heatmaps, and pair plots. Visualizations are used to identify patterns, correlations, and anomalies before model building.

- **Model Building with Supervised Learning Algorithms**
  Students implement regression and classification algorithms—Linear Regression, Logistic Regression, Naïve Bayes, and SVM—on structured datasets. They analyze model predictions using decision boundaries, ROC curves, and learning curves.

Text Book

T1: Silberschatz A, Galvin P B and Gagne G , “Silberschatz's Operating System Concepts”, Paperback, Global Edition Wiley, 2019

References

R1: Silberschatz A, Galvin P B and Gagne G, “Operating System Concepts”, 10th edition Wiley, 2018.

R2: William Stallings, “Operating Systems”,Ninth Edition, By Pearson Paperback ,1 March 2018.
R3: Sundaram RMD, Shriram K V, Abhishek S N, B Chella Prabha, “ Cracking the Operating System skills”, Dreamtech, paperback, 2020.
R4: Remzi H. Arpaci-Dusseau Andrea C. Arpaci-dusseau , “Operating Systems: Three Easy Pieces, Amazon digital Services”, September 2018.

E-Resources/Web Resources

1. https://www.os-book.com/OS9/
2. https://pages.cs.wisc.edu/~remzi/OSTEP/
3. https://codex.cs.yale.edu/avi/os-book/OS10/index.html

Course Code: CSS7000	**Course Title: Internship** **Type of Course: PRW**	**L- T-P- C**	**0-0-0-2**
Version No.	1.0		
Course Pre-requisites	Knowledge and Skills related to all the courses studied in previous semesters.		
Anti-requisites	NIL		
Course Description	Students observe science and technology in action, develop an awareness of the methods of scientific experimentation, and often get opportunities to see, study, and operate sophisticated and costly equipment. They also get opportunities to work with advanced software tools and modern technologies used in industries and research environments. Students use tools and technologies from emerging fields such as Artificial Intelligence (AI), Machine Learning (ML), Cyber Security, Blockchain, and Data Science to develop solutions for real-world and industry-related problems. They also learn about the implementation of management principles taught in the classroom by observing multidisciplinary teams of experts from engineering, science, economics, operations research, and management dealing with techno-economic problems at both micro and macro levels. The course further enables students to develop and refine their language, communication, teamwork, and interpersonal skills through various activities such as seminars, group discussions, project presentations, and report preparation. The broad-based core education, supported by strong foundations in mathematics, science, and analytical tools, provides the necessary background for students to properly understand and solve practical real-life problems. Students have the flexibility to pursue this course either as Project Work and Dissertation within the university, as Project Work in an Industry, Company, or Research Laboratory, or through Internship Programs in industries and organizations.		

Course Objectives	The objective of the course is to familiarize the learners with the concepts of Professional Practice and attain Employability Skills through Experiential Learningtechniques.
Course Outcomes	On successful completion of this course, students shall be able to: **CO1:** Identify engineering problems related to local, regional, national, and global needs. [Level 2 - Understand] **CO2:** Apply appropriate techniques and modern tools to solve engineering problems. [Level 3 - Apply] **CO3:** Design experiments in accordance with standard procedures and specifications. [Level 4- Analyze] **CO4:** Conduct experiments and collect relevant data for analysis. [Level 3 - Apply] **CO5:** Interpret experimental results and draw meaningful conclusions. [Level 6 - Create]

Course Code: APT4006	**Course Title: Logical and Critical Thinking** **Type of Course:** Audited	**L- T- P- C**	0-0-2-0	
Version No.	1.0			
Course Pre-requisites	Students should have the basic concepts of Logical reasoning and Critical thinking, along with its applications in real life problems.			
Anti-requisites	Nil			
Course Description	This is a skill-based training program for the engineering students (Undergraduate). This course is designed to enable the students to enhance their skills in Logical reasoning and Critical thinking.			
Course Objective	The objective of the course is to familiarize the learners with concepts in in Logical reasoning and Critical thinking through problem solving techniques suitable for their career development.			
Course Outcomes	On successful completion of the course the students shall be able to: CO1: Understand all the concepts. Level 2 – Understand CO2: Apply the concepts in problem solving. Level 3 – Apply CO3: Analyze and structure reasoning techniques and spatial visualization skills. Level 4 – Analyze			
Course Content:				
Module 1	Logical Thinking	Assignment		**16 Hours**
Syllogisms, Cubes and Dices, Mirror and Water images, Paper cutting and Folding, Embedded figures & Completion of figures, Data Interpretation, Data sufficiency				
Module 2	Critical Thinking	Assignment		**14 Hours**
Analogy, Symbol and Notations, Statement and assumption, Cause of action, Statement and conclusion, Puzzles				

Targeted Application & Tools that can be used: Application area: Placement activities and Competitive examinations. Tools: LMS	
Evaluation	**Continuous Evaluation** · Topic wise evaluation · Internal Assessments
Text Book 1. A new approach to reasoning verbal, non-verbal & analytical by BS Sijwali 2. R S Aggarwal 3. Kiran publications	
References 1. www.indiabix.com 2. www.testbook.com 3. www.youtube.com/c/TheAptitudeGuy/videos	
Topics relevant to Skill Development Logical reasoning and Critical thinking for Skill Development through Problem solving Techniques. This is attained through assessment component mentioned in course handout.	

Course Code: **CAI2507**	**Course Name: Reinforcement Learning** **Type of Course: Theory Course**	**L- T-P- C**	**2-0-0-2**
Version No.	X.0		
NCrF Level-	5.0		
Course Pre-requisites	CAI2500		
Co-requisites	CAI2508		
Anti-requisites	CSEXXXX		
Course Description	This course introduces foundational and advanced reinforcement learning (RL) techniques for solving complex, stochastic decision-making problems. Designed for both engineers and researchers, it covers industry-standard RL methods to model real-world scenarios and develop innovative, efficient solutions.		
Course Objective	This course is designed to improve the learners 'SKILL DEVELOPMENT' by using EXPERIENTIAL LEARNING techniques		

Course Outcomes	On successful completion of the course, learners will be able to:	
	CO1: Explain the foundational principles of reinforcement learning, including agents, environments, rewards, and policies.	Level 2 – Understand
	CO2: Apply dynamic programming, Monte Carlo, and temporal difference methods to solve Markov Decision Processes (MDPs).	Level 3 – Apply
	CO3: Implement RL algorithms such as SARSA and Q-learning in simulated environments using platforms like OpenAI Gym.	Level 3 – Apply
	CO4: Apply various RL strategies, including on-policy, off-policy, and bandit algorithms, to evaluate their performance and limitations.	Level 3 – Apply
	CO5: Apply appropriate RL techniques to solve different types of stochastic, real-world problems.	Level 3 – Apply

Course Content:

Module 1	Introduction to Reinforcement Learning	Assignment	12 Sessions

Topics:

Elements of Reinforcement Learning: Agent, Environment Interface, Goals, and Rewards, RL Platforms and Applications, Markov Decision Process (MDP): RL Environment as an MDP, Mathematical Foundations of RL: Policy and its Types, Episodic vs. Continuous Tasks, Return and Discount Factor, Core Functions of RL: Value Function, Q-Function, Learning Approaches: Model-Based and Model-Free Learning, Types of RL Environments, Solving MDPs using Bellman Equations, Dynamic Programming Algorithms: Value Iteration and Policy Iteration , Example: Frozen Lake Problem, Limitations and Scope of Reinforcement Learning

Module 2	Monte Carlo (MC) Methods	Assignment	12 Sessions

Topics:

Introduction to Monte Carlo Methods, Prediction and Control Tasks in RL, Monte Carlo Prediction: Algorithms, Types, Examples, Incremental Mean Updates , Monte Carlo Control: On-policy MC with ε-Greedy Policy, Off-policy MC Control

Limitations of Monte Carlo Methods .

Module 3	Temporal Difference (TD) Learning	Assignment	12 Sessions

Topics:

TD Learning: TD Prediction, TD Control, On-policy TD Control: SARSA, Computing Optimal Policy using SARSA, Off-policy TD Control: Q-Learning, Computing Optimal Policy using Q-Learning, Examples and Implementation, Differences between SARSA and Q-Learning, Comparison of DP, MC, and TD Methods

Module 4	Multi-Armed Bandit (MAB) Problem	Assignment	12 Sessions

Topics:

Understanding the MAB Problem, Exploration Strategies: ε-Greedy, Softmax Exploration, Upper Confidence Bound (UCB), Thompson Sampling, Applications of MAB: E.g., Optimizing Website Advertisement

Banners, Introduction to Contextual Bandits, Overview of Deep Reinforcement Learning (DRL), Introduction to Deep Q-Network (DQN)

Targeted Applications & Tools:

Python, OpenAI Gym (CartPole, FrozenLake)	Simulating agent navigation using random policies in classic control tasks
Python, OpenAI Gym, Value Iteration, Policy Iteration	Computing optimal navigation policies through Dynamic Programming techniques in discrete MDPs
Python, Gym (Blackjack), Monte Carlo Algorithms	Performing policy evaluation in card-based environments using every-visit and first-visit Monte Carlo
Python, Gym, Epsilon-Greedy MC Control	Controlling agents with on-policy Monte Carlo and ε-Greedy strategies for decision-making under uncertainty
Python, Gym (FrozenLake), TD(0), SARSA	Implementing TD prediction and SARSA for grid-based environments with real-time learning

Skill-Building Activities through Experiential Learning:

- **RL Agent Hackathon: "Learn to Play"**
  Student teams design and develop reinforcement learning agents to solve dynamic challenges in Gym environments like CartPole, Blackjack, and FrozenLake. They prototype agents using different RL algorithms (e.g., MC, TD, Q-learning) and compete on metrics such as performance, stability, and convergence in a time-bound hackathon setup.
- **Environment Analysis and Agent Modeling**
  Students explore Gym-based environments (e.g., FrozenLake, Blackjack) to identify state-action-reward structures, transition probabilities, and optimal strategies. They then model and document these environments as formal MDPs, linking theoretical constructs to empirical agent behavior using value functions and policy representations.
- **Peer Policy Simulation Challenge**
  In collaborative groups, learners implement and test different agent policies (e.g., SARSA vs. Q-learning or ε-greedy vs. softmax) in the same environments. Peers analyze each other's agents for reward efficiency, convergence rate, and policy robustness, followed by a reflective improvement phase based on collective feedback and leaderboard comparisons.

Text Book

T1: Richard S. Sutton and Andrew G. Barto, "Reinforcement Learning: An Introduction", MIT press, Second Edition, 2018.
T2: Sudharshan Ravichandiran, "Deep Reinforcement Learning with Python", Packt Publishers, Second Edition, 2020

References

R1: Laurra Graesser and Wan Loon Keng, "Foundations of Deep Reinforcement Learning", Pearson, 2022

R2: D., P. W. P. (2020). Reinforcement Learning. United States: O'Reilly Media.

R3: Lapan, M. (2024). Deep Reinforcement Learning Hands-On. United Kingdom: Packt Publishing Ltd
E-Resources/Web Resources https://www.udemy.com/course/artificial-intelligence-reinforcement-learning-in-python/ web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf

Course Code: **CAI2508**	**Course Name: Reinforcement Learning Lab** **Type of Course: Lab Course**	**L- T-P- C**	**0-0-2-1**
Version No.	X.0		
NCrF Level-	5.0		
Course Pre-requisites	CAI2500		
Co-requisites	CAI2508		
Anti-requisites			
Course Description	This lab develops practical skills in building intelligent agents for complex environments using Python and OpenAI Gym. It covers dynamic programming, Monte Carlo methods, temporal-difference learning, and deep reinforcement learning. Focus areas include modeling, training, and optimizing RL systems for real-world applications in robotics, finance, healthcare, and autonomous decision-making.		
Course Objective	This course is designed to improve the learners 'SKILL DEVELOPMENT' by using EXPERIENTIAL LEARNING techniques		
Course Outcomes	On successful completion of the course, learners will be able to: CO1: Explain the foundational concepts of reinforcement learning, including Markov Decision Processes (MDPs), policies, and reward structures. — Level 2 – Understand CO2: Implement standard RL algorithms such as Dynamic Programming, Monte Carlo, and Temporal Difference methods using Python and OpenAI Gym. — Level 3 – Apply CO3: Apply different learning strategies such as on-policy and off-policy learning to solve problems in varying environments. — Level 3 – Apply CO4: Apply agent–environment interaction techniques using exploration strategies such as ε-greedy and softmax in stochastic settings. — Level 3 – Apply		

	CO5: Apply reinforcement learning models and evaluation methods to improve performance in real-world applications like robotics or recommender systems.	Level 3 – Apply

Course Content:

Lab 1: Software Setup
Installing Anaconda, OpenAI Gym and Universe.

- 1.1: Basic simulations of some gaming environments in Gym

Lab 2: Working with Gym Environments to Create Agents with Random Policy

- 2.1: Create the Frozen Lake GYM environment and explore the states, action, transition probability, reward functions, and generating episodes
- 2.2: Create an agent for the Cart-Pole environment using a random policy and record the game

Lab **3:** Finding the Optimal Policy for the Agent Using Dynamic **Programming**

- 3.1: Compute the optimal policy for the Frozen Lake Environment using Value Iteration method
- 3.2: Compute the optimal policy for the Frozen Lake Environment using Policy Iteration method

Lab 4: Implementing Monte Carlo Prediction Method Using Blackjack Game

- 4.1: Every-Visit MC Prediction
- 4.2: First-Visit MC Prediction

Lab 5: Implementing On-Policy MC Control Method Using the Epsilon-Greedy Policy for the Blackjack Game

Lab 6: Implementing Temporal Difference Prediction for the Frozen Lake Environment for a Random Policy

Lab 7: Computing the Optimal Policy Using On-Policy TD Control – SARSA

Lab 8: Computing the Optimal Policy Using Off-Policy TD Control – Q-Learning

Lab 9: Multi-Armed Bandit (MAB) Problem

- 9.1: Creating a MAB in Gym
- 9.2: Compute the best arm using various exploration strategies such as epsilon-greedy

and softmax exploration method

Lab 10: Application of MAB
Finding the best advertisement banner for a website using MAB

Targeted Applications & Tools:

Python, OpenAI Gym (CartPole, FrozenLake)	Simulating agent navigation using random policies in classic control tasks
Python, OpenAI Gym, Value Iteration, Policy Iteration	Computing optimal navigation policies through Dynamic Programming techniques in discrete MDPs
Python, Gym (Blackjack), Monte Carlo Algorithms	Performing policy evaluation in card-based environments using every-visit and first-visit Monte Carlo
Python, Gym, Epsilon-Greedy MC Control	Controlling agents with on-policy Monte Carlo and ε-Greedy strategies for decision-making under uncertainty
Python, Gym (FrozenLake), TD(0), SARSA	Implementing TD prediction and SARSA for grid-based environments with real-time learning

Skill-Building Activities through Experiential Learning:

- **RL Agent Hackathon: "Learn to Play"**
  Student teams design and develop reinforcement learning agents to solve dynamic challenges in Gym environments like CartPole, Blackjack, and FrozenLake. They prototype agents using different RL algorithms (e.g., MC, TD, Q-learning) and compete on metrics such as performance, stability, and convergence in a time-bound hackathon setup.
- **Environment Analysis and Agent Modeling**
  Students explore Gym-based environments (e.g., FrozenLake, Blackjack) to identify state-action-reward structures, transition probabilities, and optimal strategies. They then model and document these environments as formal MDPs, linking theoretical constructs to empirical agent behavior using value functions and policy representations.
- **Peer Policy Simulation Challenge**
  In collaborative groups, learners implement and test different agent policies (e.g., SARSA vs. Q-learning or ε-greedy vs. softmax) in the same environments. Peers analyze each other's agents for reward efficiency, convergence rate, and policy robustness, followed by a reflective improvement phase based on collective feedback and leaderboard comparisons.

Text Book

T1: Richard S. Sutton and Andrew G. Barto, "Reinforcement Learning: An Introduction", MIT press, Second Edition, 2018.
T2: Sudharshan Ravichandiran, "Deep Reinforcement Learning with Python", Packt Publishers, Second Edition, 2020

References

R1: Laurra Graesser and Wan Loon Keng, "Foundations of Deep Reinforcement Learning", Pearson, 2022

R2: D., P. W. P. (2020). Reinforcement Learning. United States: O'Reilly Media.

R3: Lapan, M. (2024). Deep Reinforcement Learning Hands-On. United Kingdom: Packt Publishing Ltd
E-Resources/Web Resources 1. https://www.udemy.com/course/artificial-intelligence-reinforcement-learning-in-python/ 2. web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf

Course Code: **CAI2504**	**Course Name: Natural Language Processing** **Type of Course: Theory Course**	**L- T-P- C**	**3-0-0-3**
Version No.	1.0		
NCrF Level-	5.0		
Course Pre-requisites	NIL		
Co-requisites	CAI2505		
Anti-requisites	NIL		
Course Description	This course examines how machines interpret and generate human language through core NLP tasks like tokenization, parsing, sentiment analysis, NER, and translation. It integrates traditional and modern text representations, including TF-IDF, Word2Vec, and BERT. Using tools like spaCy, NLTK, and Hugging Face, it builds end-to-end NLP applications.		
Course Objective	This course introduces the basics of Natural Language Processing methods with specific emphasis on modern applications. The course will teach students different concepts of natural language processing, such as word representations, text representations, part-of-speech tagging, word sense disambiguation, parsing, etc. Topics: Word representations, Part-of-Speech tagging, chunking, parsing, text classification, sentiment analysis, named entity recognition, and machine translation.		
Course Outcomes	**On successful completion of the course the students shall be able to:** CO1: Explain fundamental NLP concepts, tasks, and applications such as tokenization, PoS tagging, sentiment analysis, and machine translation. [Level 2 – Understand] CO2: Explain the workings of text representation techniques and embedding models like Word2Vec, GloVe, and BERT. [Level 2 – Understand] CO3: Apply rule-based, statistical, and neural models to perform NLP tasks like PoS tagging, NER, parsing, and translation. [Level 3 – Apply]		

	CO4: Apply sequence labeling techniques and evaluation metrics such as BLEU score and inter-annotator agreement to assess NLP system performance. CO5: Implement and adapt pretrained transformer models using modern NLP libraries for solving real-world language processing problems.	[Level 3 – Apply] [Level 3 – Apply]

Course Content:

Module 1	**Introduction to Natural Language Processing**	Assessment: Assignment	12 Sessions

Topics:

Overview and scope of Natural Language Processing , Key NLP tasks and pipeline overview , Sentence segmentation and word tokenization , Basics of Part-of-Speech (PoS) tagging and syntactic parsing , Introduction to text classification , Applications of NLP:

Sentiment Analysis , Named Entity Recognition (NER) , Machine Translation

Module 2	**Word and Text Representation Techniques**	Assessment: Assignment	10 Sessions

Topics:

Word embeddings: fundamentals and motivation , Skip-gram and CBOW models using Word2Vec ,Pretrained embeddings: GloVe, fastText , Cross-lingual word embeddings (e.g., MUSE) , Bag-of-Words, TF-IDF, and kernel-based representations , Embedding-based sentence and document representations , Introduction to contextual embeddings using BERT and multilingual models

Module 3	**PoS Tagging, Chunking, and Parsing**	Assessment: Assignment	12 Sessions

Topics:

Sequence labeling tasks and Hidden Markov Models (HMMs) , Viterbi algorithm for decoding sequences ,PoS tagging using NLTK and spaCy , Building a rule-based or statistical PoS tagger , Chunking: shallow parsing of phrases , Constituency parsing and parse tree generation , Dependency parsing basics

Module 4	**Core NLP Applications**	Assessment: Assignment	11 Sessions

Topics:

Lexical resource creation and evaluation , Inter-annotator agreement metrics (Cohen's Kappa, etc.) ,**Sentiment Analysis** , Lexicon-based and ML-based approaches , Challenges: sarcasm, thwarting, domain adaptation ,**Named Entity Recognition (NER)** , NER vs. PoS tagging , Rule-based, CRF-based, and neural models ,**Machine Translation (MT)** , Rule-based, Statistical (SMT), and Neural (NMT) approaches , BLEU score and other evaluation metrics

Targeted Applications & Tools:

Google Colab	Cloud-based platform to run Python code for NLP projects using CPUs, GPUs, or TPUs; ideal for large model experiments and collaborations.

NLTK (Natural Language Toolkit)	Foundational toolkit for NLP tasks such as tokenization, PoS tagging, chunking, classification, and parsing. Includes corpora and lexical resources like WordNet.
Hugging Face Transformers	Industry-standard library offering pre-trained models like BERT, RoBERTa, GPT, and multilingual transformers for tasks including sentiment analysis, NER, and machine translation.
spaCy	High-performance NLP library for PoS tagging, dependency parsing, NER, and pipeline deployment. Suitable for building production-level applications.
Gensim	Used for word embedding models like Word2Vec, GloVe, and topic modeling; helpful in building custom vector space representations.

Skill-Building Activities through Experiential Learning:

- **Tool Familiarization and Word Embedding Exploration**
  The process begins with the installation and configuration of industry-standard NLP libraries such as NLTK, spaCy, and HuggingFace Transformers. Static embeddings like Word2Vec and GloVe are compared with contextual models like BERT and RoBERTa, allowing for hands-on visualization of vector space semantics, analogy resolution, and word similarity metrics.

- **Part-of-Speech Tagging and Syntax-Aware Feature Engineering**
  Natural language structures are explored by extracting grammatical patterns using PoS taggers and dependency parsers. Custom rule-based or statistical taggers are developed to understand sequence labeling. Features derived from syntactic parsing are used in downstream tasks such as classification and information extraction.

- **Sentiment and Named Entity Recognition from Real-World Text**
- Pre-trained sentiment analysis and NER models are applied to unstructured text sources such as product reviews, news articles, and social media posts. The outputs—polarity scores, emotion labels, and named entities—demonstrate practical applications in areas like customer feedback analysis and entity-based indexing.

Text Book

1. Daniel Jurafsky, James H. Martin. “Speech and Language Processing: An Introduction to Natural Language Processing”, Computational Linguistics and Speech, Pearson Publication, 2023 (3rd Edition Draft).

2. Aditya Joshi, Pushpak Bhattacharyya. “Natural Language Processing”, Wiley Publication, 2023 (1st Edition).

References

1. Chris Manning and Hinrich Schütze. “Foundations of Statistical Natural Language Processing”, MIT Press. Cambridge, MA. 1999.

2. Zhang, Y., Teng, Z. (2021). Natural Language Processing: A Machine Learning Perspective. India: Cambridge University Press.

3. Hapke, H., Howard, C., Lane, H. (2019). Natural Language Processing in Action:

Understanding, Analyzing, and Generating Text with Python. United States: Manning.
E-Resources/Web Resources NPTEL online course: https://nptel.ac.in/courses/106106211 Latest edition of Text Book: https://web.stanford.edu/~jurafsky/slp3/

Course Code: **CAI2505**	**Course Name: Natural Language Processing Lab** **Type of Course: Lab Course**	**L- T-P- C**	**0-0-2-1**
Version No.	1.0		
NCrF Level-	5.0		
Course Pre-requisites	NIL		
Co-requisites	CAI2504		
Anti-requisites	NIL		
Course Description	This lab course develops hands-on expertise in modern NLP tools and techniques. It covers embeddings, sentiment analysis, NER, PoS tagging, lexical simplification, and translation with transformer models. Emphasis is placed on traditional and neural approaches, including cross-lingual NLP, contextual word substitution, and building practical, evaluable models for real-world applications.		
Course Objective	This course introduces the basics of Natural Language Processing methods with specific emphasis on modern applications. The course will teach students different concepts of natural language processing, such as word representations, text representations, part-of-speech tagging, word sense disambiguation, parsing, etc. Topics: Word representations, Part-of-Speech tagging, chunking, parsing, text classification, sentiment analysis, named entity recognition, and machine translation.		
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Explain key NLP concepts and tools such as word embeddings, tokenization, and language models. [Level 2 – Understand]		

	CO2: Explain the principles behind PoS tagging, NER, sentiment analysis, and lexical simplification techniques.	[Level 2 – Understand]
	CO3: Apply NLP libraries and pre-trained models to perform tasks like translation, PoS tagging, and entity extraction.	[Level 3 – Apply]
	CO4: Apply model evaluation and linguistic feature techniques to improve accuracy in sequence labeling and classification tasks.	[Level 3 – Apply]
	CO5: Apply techniques for lexical simplification, candidate ranking, and multilingual NLP based on contextual relevance.	[Level 3 – Apply]

Course Content:

List of Laboratory Tasks:

- **Introduction to Using Word Representations and NLP Tools**

Familiarize with fundamental NLP tools and libraries such as NLTK, spaCy, and HuggingFace. Explore word representation techniques including word embeddings (Word2Vec, GloVe) and contextual embeddings (BERT, RoBERTa).

- **Complex Word Identification**

Implement techniques to detect complex or uncommon words in a given sentence using frequency-based and machine learning approaches. Use corpora to train models that identify words difficult for target audiences.

- **Sentiment Analysis and Named Entity Recognition**

Perform sentiment classification on text using pre-trained models. Build models or use tools (e.g., spaCy or Flair) for Named Entity Recognition (NER) to extract entities like people, organizations, and locations from text.

- **Lexical Simplification**

Develop a pipeline to simplify complex sentences or words while preserving meaning. Apply synonym replacement using WordNet, embedding similarity, or transformer-based ranking to generate simpler alternatives.

- **Cross-Lingual NLP**

Explore multilingual models like XLM-R or mBERT. Perform translation, classification, or sentiment analysis across languages. Understand tokenization, alignment, and transfer learning for low-resource languages.

- **Extracting PoS Features**

Use NLP libraries to extract Part-of-Speech (PoS) tags from text. Analyze the distribution of PoS tags and utilize them as features in downstream NLP tasks like text classification or chunking.

- **Building PoS Tagger**

Train a custom PoS tagger using rule-based, statistical (e.g., HMM), or machine learning methods (e.g., CRF, BiLSTM). Evaluate performance using metrics like accuracy and confusion matrix.

- **Building HMM for Title Case Conversion**

Use Hidden Markov Models to convert noisy or inconsistent text (e.g., all lowercase) into proper title case. Model the problem as a sequence labeling task where each word's correct casing is predicted.

- **Machine Translation Using Transformers**

Implement or fine-tune a transformer-based machine translation model using libraries like Fairseq or HuggingFace. Translate text between language pairs and evaluate with BLEU scores.

- **Lexical Candidate Generation and Ranking Using Pre-trained Language Models**

Generate lexical substitution candidates for a target word in context and rank them using language models like BERT. Measure contextual appropriateness using masked language modeling and semantic similarity.

Targeted Applications & Tools:

Google Colab	Cloud-based Python environment for running NLP experiments with GPU/TPU support. Ideal for working with large models.
NLTK (Natural Language Toolkit)	Core library for basic NLP tasks such as tokenization, PoS tagging, stemming, and lexical simplification.
Hugging Face Transformers	Provides access to state-of-the-art pre-trained transformer models like BERT, RoBERTa, GPT, mBERT, XLM-R for tasks such as sentiment analysis, NER, translation, and masked word prediction.
spaCy	Industrial-strength NLP toolkit for fast PoS tagging, NER, dependency parsing, and text processing.
Flair	NLP framework built on PyTorch, used for sequence labeling tasks like NER with contextual embeddings.

Skill-Building Activities through Experiential Learning:

- **Tool Familiarization and Word Embedding Exploration**

Participants begin by installing and configuring industry-standard NLP libraries such as NLTK, spaCy, and HuggingFace Transformers. They visualize and compare static (Word2Vec, GloVe) vs. contextual (BERT, RoBERTa) embeddings, gaining foundational understanding of vector space semantics and word similarity.

- **Complex Word Identification through Corpus-Based Modeling**

Experiments include training simple frequency-based and machine learning classifiers to flag difficult vocabulary based on user profiles. Real datasets from news articles and academic texts are used to simulate adaptive content simplification, enhancing awareness of accessibility in NLP.

- **Sentiment and Named Entity Analysis from Real Text**

By applying pre-trained sentiment analysis and NER pipelines on social media posts, movie reviews, or business reports, learners understand how NLP aids in extracting emotion and structured information from unstructured data, bridging theory with real-world business intelligence tasks.

Text Book

1. Daniel Jurafsky, James H. Martin. "Speech and Language Processing: An Introduction to Natural Language Processing", Computational Linguistics and Speech, Pearson Publication, 2023 (3rd Edition Draft).
2. Aditya Joshi, Pushpak Bhattacharyya. "Natural Language Processing", Wiley Publication, 2023 (1st Edition).

References

1. Chris Manning and Hinrich Schütze. "Foundations of Statistical Natural Language Processing", MIT Press. Cambridge, MA. 1999.
2. Zhang, Y., Teng, Z. (2021). Natural Language Processing: A Machine Learning Perspective. India: Cambridge University Press.
3. Hapke, H., Howard, C., Lane, H. (2019). Natural Language Processing in Action: Understanding, Analyzing, and Generating Text with Python. United States: Manning.

E-Resources/Web Resources

NPTEL online course: https://nptel.ac.in/courses/106106211

Latest edition of Text Book: https://web.stanford.edu/~jurafsky/slp3/

Course Code: CSS2271	Course Title: Software Design and Development Type of Course: Theory Course (PCC)	L- T-P- C	3-0-0-3

Version No.	1.0
Course Pre-requisites	NIL
Anti-requisites	NIL
Course Description	This course introduces foundational concepts of software engineering and emphasizes a practical, structured approach to software system development. It spans software requirement analysis, process modeling, system design, Agile principles, DevOps culture, testing strategies, quality assurance, and software maintenance. Hands-on assignments and problem-solving activities foster understanding of contemporary software practices and industry tools such as CASE tools, GitHub, and Selenium. The course also introduces secure software engineering practices, DevSecOps concepts, and AI-enhanced tools for documentation, automation, intelligent testing, and software quality prediction workflows.
Course Objectives	To develop a comprehensive understanding of software engineering by enabling students to analyze software development life cycle models, apply design principles and patterns, develop modular and maintainable software systems, perform testing and quality assurance, manage software projects effectively, and understand secure and AI-driven software development practices.
Course Outcomes	On successful completion of this course, students will be able to: CO1: Explain software engineering principles, ethics, and various software development life cycle models. — Level 2 – Understand CO2: Identify software requirements and develop design models, including Software Requirement Specification (SRS) documents. — Level 3 – Apply CO3: Apply Agile development practices and DevOps principles in software development. — Level 3 – Apply CO4: Apply software testing strategies, configuration management, and maintenance techniques. — Level 3 – Apply CO5: Evaluate secure and AI-driven software engineering practices to ensure software quality, reliability, scalability, and intelligent automation. — Level 5 – Evaluate

Course Content:

Module 1	Introduction to Software Engineering and Process Models	10 Sessions

Introduction: Need for Software Engineering, Professional Software Development, Software Engineering Ethics, Software Engineering Practice-Essence of Practice, General Principles Software Development Life Cycle. Models: Waterfall Model – Classical Waterfall Model, Iterative Waterfall Model, Evolutionary model-Spiral, Prototype.

Module 2	Software Requirements, Analysis and Design	12 Sessions
Requirements Engineering: Eliciting requirements, Functional and non- Functional requirements, Software Requirements Specification (SRS), Requirement Analysis and validation. Requirements modelling- Introduction to Use Cases, Activity diagram and Swim lane diagram. CASE support in Software Life Cycle, Characteristics of CASE Tools, Architecture of a CASE Environment. Design: Design concepts, Architectural design, Component based design, User interface design.		
Module 3	Agile Principles & Devops	10 Sessions
Agile: Scrum Roles and activities, Sprint Agile software development methods - Scaling, User Stories, Agile estimation techniques, Product backlogs, Stake holder roles, Dynamic System Development Method. DevOps: Introduction, definition, history, tools.		
Module 4	Software Testing and Maintenance	08 Sessions
Software Testing-verification and validation, Test Strategies - White Box Testing, Black box Testing. Automation Tools for Testing. Software Quality Assurance-Elements of software quality assurance, SQA Tasks, Goals and Metrics, Software configuration management- SCM process, SCM Tools (GitHub). Maintenance- Characteristics of Software Maintenance, Software Reverse Engineering, Software Maintenance Process Models.		
Module 5	Secure, Intelligent and AI-Driven Software Engineering	05 Sessions
Secure Software Development Practices, Software Security Testing, AI in Software Engineering, Intelligent Requirement Analysis, AI-based Testing and Debugging, DevSecOps Concepts, Machine Learning for Software Quality Prediction, Smart Project Management Tools, Applications of AI in Modern Software Development.		
Targeted Application & Tools that can be used: Git hub/Lambda/Taskade/Bit.ai/Testsigma/TestRogor		
Text Books: T1. Roger S. Pressman, "Software Engineering – A Practitioner's Approach", VII Edition, McGraw-Hill, 2017. T2. Bob Hughes, Mike Cotterell, Rajib Mall, "Software Project Management", VI Edition, McGraw-Hill, 2018.		
References: R1. Rajib Mall, "Fundamentals of Software Engineering", VI Edition, PHI learning private limited, 2015. R2. Ian Sommerville, "Software Engineering", IX Edition, Pearson Education Asia, 2011. R3. Agile Software Development Principles, Patterns and Practices.1st Edition, Wiley, 2002 Journals / Magazines • IEEE Software • ACM Transactions on Software Engineering and Methodology • Journal of Systems and Software SWAYAM / NPTEL / MOOCs • NPTEL – Software Engineering (Prof. Rajib Mall, IIT KGP) • Coursera – Agile Development (University of Virginia)		

• Udemy – DevOps Masterclass • EdX – Software Testing Fundamentals (TUM)
Topics Relevant to "Skill Development": Black box Testing, White box Testing, AI tools for design, modelling, testing and document Automated Testing for Skill development through Participative Learning Techniques. This is attained through assessment mentioned in the course handout

Course Code: CSS2274	Course Title: Competitive Programming and Problem Solving Type Of Course: Laboratory Course (ESC)	L-T-P-C	0-0-4-2
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course equips students with efficient problem-solving skills for competitive programming and real-world computational challenges. Starting with brute-force approaches, students learn to optimize time and space complexity using advanced techniques such as greedy algorithms, dynamic programming, backtracking, graph algorithms, and heuristic optimization methods. Hands-on practice using platforms such as CodeChef and Codeforces helps students solve problems related to number theory, data structures, and algorithmic paradigms. The course also introduces intelligent problem-solving strategies, AI-assisted coding concepts, and optimization techniques for competitive programming. By understanding competitive programming constraints and developing a strategic problem-solving mindset, students gain confidence for coding competitions, technical interviews, and real-world applications.		
Course Objective	The objective of the course is to familiarize learners with the concepts of Competitive Programming and Problem Solving, enabling them to develop efficient coding, optimization, and analytical skills through experiential learning techniques and intelligent problem-solving approaches.		
Course Outcomes	On successful completion of the course the students shall be able to: **CO1:** Understand the challenges of online competitive programming platforms, interpret problem constraints, and understand brute-force solution approaches used in competitive programming. — **Level 2 – Understand** **CO2:** Analyze the time and space complexity of number-theoretic solutions to determine the need for optimization under competitive programming constraints. — **Level 4 – Analyze** **CO3:** Apply appropriate algorithmic paradigms and data structures to solve problems within defined time and memory constraints. — **Level 3 – Apply** **CO4:** Analyze execution failures and identify algorithmic inefficiencies in solutions involving advanced data structure concepts to redesign optimized solutions within given time and memory constraints. — **Level 4 – Analyze** **CO5:** Analyze advanced problem-solving and intelligent — **Level 4 – Analyze**		

	optimization techniques using dynamic programming, greedy methods, backtracking, and heuristic approaches for large-scale competitive programming problems.

Course Content:

Module 1	Introduction to Competitive Programming	NA

Topics:
Overview of Efficient Coding for Problem Solving and CP: Introduction to competitive programming (CP); revisit of complexity analysis; introduction to online platforms such as codechef, codeforcesetc and online submission; constraints during CP, online testing process and common errors such as TLE; use of STL.
CP Problem Solving using Advanced Topics: concept of disjoint sets and their efficient representation, algorithmic approaches such as Greedy, Backtracking, Dynamic Programming and applying them for CP problems using bottom-up dynamic programming.

Module 2:	Number Theory for Problem-Solving	NA

Topics:
Use of Number Theory for problem-solving: reducing time/space complexity of brute force coding solution of Sieve Method, Inverse Module, Euclidian Method of factorization; efficient coding for Permutation Combination; XORing based and pattern-based solutions.

Module 3:	Optimizing Time & Space Using Sequential Storage	NA

Topics:
Coding for Optimizing time and Space using Sequential Storage: two pointer approach; problem-solving using arrays and strings such as rotation on sorted arrays, duplicate removal, string matching algorithms; Kadane's algo, stacks, priority-queues and hashing based efficient coding; median based problems and alternate solutions.

Module 4:	Non-Linear Data Structures & Problem Solving using Advanced Topics	NA

Topics:
Applying Non-Linear Data Structures for real-life problems: design of efficient solutions for problems such as finding loops in a linked list, memory efficient DLL, block reversal in LL; problem solving using trees and binary trees, Catalan numbers, applications of graphs, spanning tree and path algos for CP problems with reduced time/space complexity. CP Problem Solving using Advanced Topics: concept of disjoint sets and their efficient representation, algorithmic approaches such as Greedy, Backtracking, Dynamic Programming and applying them for CP problems using bottom-up dynamic programming.

Module 5:	Advanced Competitive Programming and Intelligent Problem Solving	NA

Topics:
Disjoint Sets and Efficient Representation, Greedy Algorithms, Backtracking Techniques, Dynamic Programming using Bottom-up Approach, Advanced Graph Algorithms for Competitive Programming, AI-assisted Problem Solving Concepts, Heuristic and Optimization Techniques, Intelligent Coding Strategies for Time and Space Optimization.

List of Practical Tasks 60 Sessions

WEEK 1

Lab Sheet 1: Sorting & Ranking

Level 1: Write a program to find the runner who finished in the third position from the given finishing times of N runners.

Level 2: Write a program to find the top 10 runners along with their bib numbers efficiently for large datasets.

Lab Sheet 2: Searching Techniques

Level 1: Write a program to check whether a given book ID exists in a library database.

Level 2: Design and implement a system to search books efficiently by title using an appropriate data structure.

WEEK 2

Lab Sheet 3: Cost Calculation

Level 1: Write a program to calculate the total cost of products in a shopping cart.

Level 2: Design an algorithm to apply multiple discount rules and compute the final bill.

Lab Sheet 4: Modular Arithmetic

Level 1: Write a program to compute (a^m \mod p).

Level 2: Implement binary exponentiation for efficient computation of large powers modulo p.

WEEK 3

Lab Sheet 5: Combinatorics

Level 1: Write a program to calculate the number of ways to choose K cards from N cards.

Level 2: Design an algorithm to calculate probabilities of drawing specific card combinations.

Lab Sheet 6: Graph Connectivity

Level 1: Write a program to determine if a path exists between two nodes in a network.

Level 2: Design an algorithm to detect anomalies in a network using XOR-based comparison techniques.

WEEK 4

Lab Sheet 7: Array Processing

Level 1: Write a program to find the minimum time required for cars to pass a point based on their speeds.

Level 2: Implement a two-pointer approach to simulate movement of cars with different lengths.

Lab Sheet 8: String Matching

Level 1: Write a program to count occurrences of a substring in a given string.

Level 2: Implement the KMP algorithm to efficiently find all occurrences of a pattern in a text.

WEEK 5

Lab Sheet 9: Priority Queue

Level 1: Write a program to track the highest bid in an auction system.

Level 2: Design a system to handle concurrent bids efficiently and update the highest bid.

Lab Sheet 10: Shortest Path

Level 1: Write a program to check if two users are connected in a social network.

Level 2: Implement Dijkstra's algorithm to find the shortest path between users.

WEEK 6

Lab Sheet 11: Tree Traversal

Level 1: Write a program to traverse a file system and display all files and directories.

Level 2: Design a file system supporting operations like create, delete, and search.

Lab Sheet 12: Tree Applications

Level 1: Write a program to calculate the total price of items in a shopping cart represented as a tree.

Level 2: Design a system to apply discounts efficiently in a hierarchical shopping cart.

WEEK 7

Lab Sheet 13: Disjoint Set

Level 1: Write a program to check whether all users in a group are connected.

Level 2: Find the minimum number of connections required to connect all users.

Lab Sheet 14: Greedy Algorithms

Level 1: Write a program to find the cheapest path to reach a destination.

Level 2: Design an algorithm to find the fastest path considering time constraints.

WEEK 8

Lab Sheet 15: BFS Applications

Level 1: Write a program to find the minimum number of moves for a rook to reach a target square.

Level 2: Implement BFS to find shortest path in a grid.

Lab Sheet 16: Game Algorithms

Level 1: Represent a game state using a tree structure.

Level 2: Implement minimax algorithm with alpha-beta pruning.

WEEK 9

Lab Sheet 17: Sorting Algorithms

Level 1: Write a program to implement insertion sort.

Level 2: Write a program to implement merge sort or quick sort.

Lab Sheet 18: Searching Algorithms

Level 1: Write a program to perform linear search.

Level 2: Write a program to perform binary search efficiently.

WEEK 10

Lab Sheet 19: Algorithm Analysis

Level 1: Write a program to compare execution time of two algorithms.

Level 2: Optimize a given algorithm to improve performance.

Lab Sheet 20: Recursion

Level 1: Write a program to solve Tower of Hanoi.

Level 2: Compare recursive and iterative solutions for a problem.

WEEK 11

Lab Sheet 21: Graph Representation

Level 1: Represent a graph using adjacency list.

Level 2: Represent a graph using adjacency matrix.

Lab Sheet 22: Graph Traversal

Level 1: Implement Depth First Search (DFS).

Level 2: Implement Breadth First Search (BFS).

WEEK 12

Lab Sheet 23: Data Structure Integration

Level 1: Combine arrays and lists to solve a problem.

Level 2: Use multiple data structures (tree + hash table) in one solution.

Lab Sheet 24: Real-world Application

Level 1: Design a simple recommendation system.

Level 2: Optimize it for large-scale data.

WEEK 13

Lab Sheet 25: String Algorithms

Level 1: Write a program for basic string matching.

Level 2: Implement efficient pattern matching algorithm.

Lab Sheet 26: Network Algorithms

Level 1: Check connectivity in a network.

Level 2: Detect anomalies using data comparison techniques.

WEEK 14

Lab Sheet 27: Algorithm Design

Level 1: Solve a problem using brute-force approach.

Level 2: Improve the solution using greedy or dynamic approach.

Lab Sheet 28: Performance Evaluation

Level 1: Analyze runtime of a program.

Level 2: Compare different algorithmic approaches.

WEEK 15: Intelligent Optimization and Competitive Programming

Lab Sheet 29: Advanced Problem Solving

Level 1: Solve optimization problems using greedy or dynamic programming techniques.
Level 2: Implement backtracking or graph-based solutions for competitive programming problems.

Lab Sheet 30: Intelligent Algorithm Analysis

Level 1: Analyze time and space complexity of advanced algorithms.
Level 2: Compare heuristic, dynamic programming, and brute-force approaches for performance optimization.

Targeted Application & Tools that can be used:

1. C or C++ Compiler (g++): The standard compiler for CP. Familiarize students with compilation flags (e.g., -O2 for optimization).
2. IDE (Integrated Development Environment): Code:: Blocks, Visual Studio, CLion, or similar IDEs. These provide debugging capabilities, code completion, and other helpful features.
3. Online Judges (CodeChef, Codeforces, LeetCode, HackerRank): Essential for practicing and submitting solutions.
4. Debugger (gdb): Crucial for understanding code execution and finding bugs. Origin, excel and Mat lab soft wares for programming and data analysis.
5. Number Theory Libraries: Some libraries provide pre-built functions for number theory operations (though often it's better to implement them yourself for learning).
6. Wolfram Alpha: A useful tool for verifying number theory calculations and exploring concepts.
7. String Libraries: Familiarize students with the string manipulation functions available in C++.
8. Graph Visualization Tools: Tools like Graphviz can be helpful for visualizing graphs and understanding graph algorithms.
9. DP Debugging Techniques: Practice debugging DP solutions, as they can be complex. Visualizing the DP table can be helpful

Text Books:

1 Guide to Competitive Programming: Learning and Improving Algorithms Through Contests" (3rd Edition), *Antti Laaksonen, springer, 2024*

2 "Data Structures and Algorithms in Java: A Project-Based Approach" – *Dan S. Myers, Cambridge University Press*

Reference Books:

1. Data Structures and Algorithmic Thinking with Python/C++/Java", Narasimha Karumanchi, 5th Edition, Career Monk, 2017.
2. Introduction to Algorithms, Thomas H. Cormen (Author), Charles E. Leiserson (Author), RonaldL. Rivest , fourth edition April 2022 .

Web Resources

1. https://nptel.ac.in/courses/106106231

Project work/Assignment: Mention the Type of Project /Assignment proposed for this course

Assessment Type

1. Midterm exam
2. Assignment (review of digital/ e-resource from PU link given in references section - mandatory to submit screen shot accessing digital resource.)
3. Quiz
4. End Term Exam
5. Self-Learning

Course Code: **LAW7601**	Course Name: **Indian Constitution**	L- T-P- C	1-0-0-0
Version No.	1.0		

Course Pre-requisites	NIL
Co-requisites	NIL
Anti-requisites	NIL
Course Description	This course is designed to improve the learners' SKILL DEVELOPMENT by using PATICIPATIVE LEARNING techniques. This course aims to familiarize students with fundamentals of Indian Constitution concepts and their relevance to 75+ Years of Republic of India (https://constitution75.com/) as well as **#AzaadiKaAmrutMahotsav / Azadi Ka Amrit Mahotsav** (https://amritmahotsav.nic.in). It is designed to equip students with the knowledge about the Constitution of India. This course aims to introduce the constitutional law of India to students from all walks of life and help them understand the constitutional principles as applied and understood in everyday life. The objective of making the Constitution of India, familiar to all students, and not only to law students, this course aims and objectifies legal understanding in the simplest of forms. **This course is designed to cater to Constitutional Studies.**
Course Objective	The objective of the course is **'SKILL DEVELOPMENT'** of the student by using **'PARTICIPATIVE LEARNING'** techniques
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Describe the basic understanding of the Indian Constitution and the concepts and issues relevant to day-to-day life of the nation and equip citizens with the zeal for capacity building. — Level 2 – Understand **CO2:** Recognize and identify the values of the Constitution of India. — Level 1 – Remember **CO3:** Explain citizen-centric awareness of the rights and responsibilities of the State. — Level 2 – Understand **CO4:** Explain the role of State actors in building India. — Level 2 – Understand
Course Content:	
Module 1	**Understanding the Making of the Constitution: The Constituent Assembly & The Constitution of India**
Topics: Historical Context of Constituent Assembly - Compositions & Functions of Constituent Assembly What is a Constitution? – Why have a Constitution? – Constitutional Change - Features of Indian Constitution – Preamble of Indian Constitution	
Module 2	**Citizen's Fundamental Rights and State's Responsibilities (Directive Principles)**
Topics: Introduction to Fundamental Rights - Right to Equality – Facets of Right to Equality - Right to Freedom - Constitutional Position of Some Democratic Rights - Right Against Exploitation - Right to Freedom of Religion - Right to Constitutional Remedies	

Directive Principles of the State Policy	
Module 3	**Organs Of the Government**
Topics: **Executive**: The President of India - Powers and Functions of President of India - Emergency Powers and the Position of the President **Legislature**: Union Council of Ministers - Prime Minister - The Rajya Sabha - The Lok Sabha - Relation between the Lok Sabha & Rajya Sabha - Office of the Speaker – Important Parliamentary Committees **Judiciary**: The Structure and Organization of the Judiciary & the High Court - The Supreme Court - Role of The Supreme Court - Judicial Activism in India - Basic Structure Doctrine & PIL	
Module 4	**Federalism & Decentralization**
Topics: What is Federalism? - Centre-State Legislative Relations - Centre-State Administrative Relations - Centre-State Financial Relations The 5th & 6th Schedules - Municipality- (History of Indian Municipality, Organization & Functions) – Panchayat 1 (Idea of Panchayat, Organization and Powers of Panchayats in India)	
Targeted Application & Tools that can be used: Application areas to familiarize students with fundamentals of Indian Constitutional concepts. **Tools:** Online Tools – NPTEL and Swayam.	
Project work/Assignment:	
Assessment Type Online end term exam will be conducted as notified by the Presidency University.	
Online Link*: 1) Prof. Amitabha Ray, SWAYAM Course: "Constitutional Government & Democracy in India" https://onlinecourses.swayam2.ac.in/cec19_hs13/preview * Other source links are available in below Resources link. **Text Book** 1. Durga Das Basu --- Introduction to the Constitution of India, 23rd Edition (Gurgaon; LexisNexis, 2018). 2. MP Jain's Constitutional Law of India, Lexis Nexis 3. V.N Shukla's Indian Constitutional Law, M.P Singh 13th Edition 4. MV Pylee's Constitution of India 5. J.C.Johari -- The Constitution of India: A Politico-Legal Study (Greater Noida: Sterling Publishers Pvt. Ltd. 2013). 6. Himangshu Roy and M.P.Singh – Indian Political System, 4th Edition (Bengaluru; Pearson Education, 2018) 7. Vidya Bhushan &Vishnoo Bhagwan--- Indian Administration (S. Chand, 2011) 8. S.R.Maheswari --- Indian Administration (Orient Blackswan, 2001) 9. Dr. A.Avasthi& A.P. Avasthi --- Indian Administration (L.N. Agarwal Educational Publishing, 2017). 10. B. L. Fadia --- Indian Government and Politics (Sahitya a. Bhawan, 13th Revised Edition, 2017).	

11. P.M.Bakshi – The Constitution of India (Prayagraj, UP; a. Universal Law Publishing, January, 2018)

Reference Books

1. HM Seervai, Constitutional Law of India, 4th Ed. Vol I, II, & III
2. Uday Raj Rai, Constitutional Law-I
3. Democracy and Constitutionalism in India, Oxford University Press 2009

Resources:

1. https://onlinecourses.nptel.ac.in/noc20_lw03/course?&force_user=true
2. https://onlinecourses.swayam2.ac.in/cec19_hs13/course?&force_user=true
3. https://nptel.ac.in/courses/129106003
4. https://nptel.ac.in/courses/129106411
5. https://nptel.ac.in/courses/129105608
6. https://nptel.ac.in/courses/129106002

Topics relevant to Skill Development:

1. An attitude of inquiry.
2. Write reports

The topics related to Constitutional Studies and its application :

All topics in theory component are relevant to Indian Constitution.

Course Code: APT4005	Course Title: Aptitude For Employability Type of Course: Practical Only	L- T-P- C	0-0-2-1
Version No.	1.0		
Course Pre-requisites	Students should have the basic concepts of Quantitative aptitude, Verbal ability along with its applications in real life problems.		
Anti-requisites	Nil		
Course Description	This course is designed to enable the students to enhance their skills in quantitative aptitude and verbal ability skills.		
Course Objective	The objective of the course is to familiarize the learners with concepts in Quantitative Aptitude and Verbal ability through problem solving techniques suitable for their career development.		
Course Outcomes	On successful completion of the course the students shall be able to: CO1: Recall all the basic mathematical concepts. — Level 1 – Remember CO2: Identify the principal concept needed in a question. — Level 2 – Understand CO3: Solve quantitative and logical ability questions using the appropriate concept. — Level 3 – Apply		

Course Content:

Module 1	Quantitative Ability	Lab-10hrs		Platform Assessment-10hrs	20 Hours

Topics: Number System, Percentage, Ratio and Proportion, Average, Mixture and Allegation, Time and Work, Profit and Loss, Time Speed and Distance, Simple Interest and Compound Interest, Probability, Permutation and Combination.					
Module 2	Verbal Ability	Lab-5hrs		Platform Assessment-5hrs	10 Hours
Topics: - Parts of Speech, Subject Verb Agreement, Spotting Error, Cloze Test, Verbal Analogies, Reading Comprehension, Idioms & Phrases, Para Jumbles					
Targeted Application & Tools that can be used: Application area: Placement activities and Competitive examinations. Tools: LMS					
Evaluation	Continuous Evaluation Topic wise evaluation				
Text Book Fast track objective by Rajesh Verma R S Aggarwal S.P Bakshi					
References www.indiabix.com www.testbook.com www.youtube.com/c/TheAptitudeGuy/videos					
Topics relevant to Skill development:Quantitative and reasoning aptitude for Skill Development through Problem solving Techniques.This is attained through assessment component mentioned in course handout					

Course code: PPS3018	**Course Title: Preparedness for Interview Type of Course: Practical Only Course**	**L- T- P- C**	**1-0-0-0**
Version No.	1.0		
Course Pre-requisites	Students are expected to understand Basic English. Students should have desire and enthusiasm to involve, participate, and learn.		
Co-requisites	NIL		
Anti-requisites	NIL		

<table>
<tr><td>Course Description</td><td colspan="3">This course is designed to enable students to understand soft skills concepts to be corporate-ready. The modules are set to improve self-confidence, communicate effectively and Prepare for the Interview to assist in employability. It helps the students to get a glimpse of the acceptable corporate readiness and equip them with the fundamental necessities of being able to confidently deal with the highly competitive corporate environment and helps in crafting different types of resumes. The pedagogy used will be group discussions, flipped classrooms, continuous feedback, role-play and mentoring.</td></tr>
<tr><td>Course Objective</td><td colspan="3">The objective of the course is to familiarize the learners with the concepts of “Preparing for Interview” and attain SKILL DEVELOPMENT through PARTICIPATIVE LEARNING techniques.</td></tr>
<tr><td>Course Outcomes</td><td colspan="3">On successful completion of the course the students shall be able to:

CO1: Develop professional resumes that effectively showcase qualifications, skills, achievements, and career objectives. — L3 – Apply
CO2: Illustrate and organize resume content effectively using appropriate formats, design principles, and professional standards. — L2 – Understand
CO3: Apply communication skills, subject knowledge, and professional etiquette to participate actively and effectively in group discussions and interviews. — L3 – Apply</td></tr>
<tr><td colspan="4">Course Content:</td></tr>
<tr><td>Module 1</td><td>Resume Building</td><td>Classroom activity</td><td>10 Sessions</td></tr>
<tr><td colspan="4">Topics: Resume structure, use of templates, Do’s and Don'ts, ATS methods, Cover Letter and Video Resume
Activity: Real world scenarios</td></tr>
<tr><td>Module 2</td><td>Group Discussion</td><td>Mock G D</td><td>9 Sessions</td></tr>
<tr><td colspan="4">Topics: -Group discussion as a placement process, GD techniques like Keyword. SPELT & POV of affected parties. Do & Don’t of GD, Case-lets and topics for GD, practice session and evaluation
Activity:- Real world scenarios</td></tr>
<tr><td>Module 3</td><td>Personal Interview</td><td>Grooming checks + Evaluation + Mock Interview+ Role Play</td><td>9 Sessions</td></tr>
<tr><td colspan="4">Topics: Placement process, Different interview rounds, HR interviews, Interview questions and desired answers, Different types of interviews, Do’s and Don'ts.
Activity: - Role Play & Real-world scenario</td></tr>
<tr><td>Module 4</td><td>Recap/Revision /Feedback Session</td><td>Practice sessions</td><td>2 Sessions</td></tr>
<tr><td colspan="4">Targeted Application & Tools that can be used:
1. TED Talks
2. You Tube Links
3. Role Play activities</td></tr>
<tr><td colspan="4">Project work/Assignment: Mention the Type of Project /Assignment proposed for this course</td></tr>
<tr><td colspan="4">Continuous Individual Assessment</td></tr>
<tr><td colspan="4">The Topics related to Skill Development:Art Of Presentation and Group Discussion for Skill Development through Participative Learning Tech- niques. This is attained through assessment Component mentioned in course handout.</td></tr>
</table>

Course Code: CSS7100	**Course Title: Mini Project** **Type of Course: PRW**	**L- T-P- C**	**0-0-0-4**
Version No.	1.0		
Course Pre-requisites	Knowledge and Skills related to all the courses studied in previous semesters.		
Anti-requisites	NIL		
Course Description	Students observe science and technology in action, develop an awareness of the methods of scientific experimentation, and often get opportunities to see, study, and operate sophisticated and costly equipment. They also get opportunities to work with advanced software tools and modern technologies used in industries and research environments. Students use tools and technologies from emerging fields such as Artificial Intelligence (AI), Machine Learning (ML), Cyber Security, Blockchain, and Data Science to develop solutions for real-world and industry-related problems. They also learn about the implementation of management principles taught in the classroom by observing multidisciplinary teams of experts from engineering, science, economics, operations research, and management dealing with techno-economic problems at both micro and macro levels. The course further enables students to develop and refine their language, communication, teamwork, and interpersonal skills through various activities such as seminars, group discussions, project presentations, and report preparation. The broad-based core education, supported by strong foundations in mathematics, science, and analytical tools, provides the necessary background for students to properly understand and solve practical real-life problems. Students have the flexibility to pursue this course either as Project Work and Dissertation within the university, as Project Work in an Industry, Company, or Research Laboratory, or through Internship Programs in industries and organizations.		
Course Objectives	The objective of the course is to familiarize the learners with the concepts of Professional Practice and attain Employability Skills through Experiential Learningtechniques.		
Course Outcomes	On successful completion of this course, students shall be able to: **CO1:** Identify engineering problems related to local, regional, national, and global needs. [Level 2 - Understand] **CO2:** Apply appropriate techniques and modern tools to solve the intended problems. [Level 3 - Apply] **CO3:** Design experiments in accordance with standards and specifications. [Level 4- Analyze] **CO4:** Interpret events and results to derive meaningful conclusions. [Level 5- Evaluate] **CO5:** Appraise project findings and communicate effectively through scholarly publications. [Level 6 - Create]		

Course Code: CSS7300	**Course Title: Capstone Project** **Type of Course: PRW**	**L- T-P- C**	**0-0-0-10**
Version No.	1.0		
Course Pre-requisites	Knowledge and Skills related to all the courses studied in previous semesters.		
Anti-requisites	NIL		
Course Description	Students learn by observing science and technology in real-life applications and gain knowledge through practical experimentation and hands-on activities. They get opportunities to study and work with advanced equipment, software tools, and modern technologies. Students use tools and technologies from emerging fields such as Artificial Intelligence (AI), Machine Learning (ML), Cyber Security, Blockchain, and Data Science to develop solutions for real-world and industry-related problems. The course helps students understand how engineering, science, economics, operations research, and management work together to solve technical and business-related challenges. By working on projects, seminars, group discussions, and report preparation, students improve their communication, teamwork, presentation, and interpersonal skills. The strong foundation in mathematics, science, and analytical thinking helps students understand and solve practical problems effectively. Students can complete the course through Project Work and Dissertation within the university, by doing projects in industries or research laboratories, or through internship programs in companies and organizations.		
Course Objectives	The objective of the course is to familiarize the learners with the concepts of Professional Practice and attain Employability Skills through Experiential Learningtechniques.		
Course Outcomes	On successful completion of this course, students shall be able to: **CO1:** Identify engineering problems at various societal levels [Level 2 - Understand] **CO2:** Apply tools and techniques for problem solving. [Level 3 - Apply] **CO3:** Design and plan experiments per industry standards. [Level 3- Apply] **CO4:** Interpret the events and results for meaningful conclusions. [Level 5- Evaluate] **CO5:** Appraise project findings and communicate effectively through scholarly publications. [Level 6- Create]		

Course Code: CAI3400	Course Title: Image Processing and Analysis Type of Course: Theory Integrated Lab	L- T-P- C	2-0-2-3
Version No.	1.0		

Course Pre-requisites	ISE2504 /CAI2502		
Anti-requisites	NIL		
Course Description	This course explores the theory and practical application of Digital Image Processing and Analysis, integrated with modern AI-based techniques for visual computing. It covers classical methods for image enhancement, restoration, segmentation, and feature extraction, while introducing deep learning-based approaches such as Convolutional Neural Networks (CNNs), semantic segmentation, object detection, image super-resolution, and generative models. The curriculum includes hands-on experience with OpenCV, PyTorch/TensorFlow, and pretrained models to address real-world image processing problems across various domains.		
Course Objective	The course aims to equip students with the knowledge and skills to acquire and process multidimensional and multispectral images, enhance and restore image quality, apply efficient image representation and compression techniques, and utilize methods such as registration, fusion, and super-resolution in real-world applications.		
Course Outcomes	On successful completion of this course, the students shall be able to: **CO1:** Describe the key concepts of digital image formation, acquisition, and basic transformations. — Level 2 – Understand **CO2:** Apply enhancement techniques in spatial and frequency domains to improve image quality. — Level 3 – Apply **CO3:** Use segmentation and morphological operations to separate and analyze image regions. — Level 3 – Apply **CO4:** Interpret image features for classification and object identification tasks. — Level 3 – Apply **CO5:** Implement basic image processing functions using Python and OpenCV for practical applications. — Level 3 – Apply		
Course Content:			
Module 1	**Introduction to Image Processing**	Assignment 1	6 Sessions
Topics: Basics of digital images: pixels, resolution, and color models Image acquisition, sampling, and quantization Intensity transformation functions and histogram operations Basic geometric transformations: scaling, rotation, and translation **AI Perspective:** Introduction to CNN-based representations vs. classical transforms			
Module 2	**Image Enhancement and Restoration**	Assignment 1	6 Sessions
Topics: Enhancement in spatial domain: contrast stretching, histogram equalization Filtering techniques: smoothing and sharpening filters Frequency domain enhancement using Fourier Transform Noise models and image degradation Restoration methods including inverse filtering and Wiener filtering **AI Perspective:**			

AI-based denoising (DnCNN, Autoencoders)

AI-based super-resolution (ESRGAN)

Module 3	Image Compression and Representation for Analysis	Assignment 2	6 Sessions

Topics:

Edge detection methods: Sobel, Prewitt, Canny
Thresholding techniques: global, adaptive, and Otsu's method
Region-based segmentation: region growing, splitting and merging
Morphological operations: erosion, dilation, opening, closing
Shape analysis and processing of binary images
AI Perspective:
Semantic segmentation using pretrained **U-Net**
Comparative study: AI vs classical segmentation

Module 4	**Image Analysis and Applications**	Assignment	6 Sessions

Topics:

Feature extraction: color, texture, and shape descriptors
Texture analysis using statistical and structural methods
Object recognition using descriptors and matching techniques
Applications in biomedical imaging, satellite image analysis, and industrial automation
Introduction to OpenCV-based projects and datasets for image analysis
AI Perspective:
Deep feature extraction (ResNet, VGG)

Object detection with YOLO/SSD

AI-based anomaly detection

Module 5	**AI-based Image Processing Techniques**	Assessment: Mini-project	6 Sessions

Topics:

Introduction to Deep Learning for Image Processing
CNN Architectures overview: VGG, ResNet, MobileNet
Transfer Learning for feature extraction
Image classification using pretrained networks
Object detection with YOLO/SSD
Semantic segmentation with U-Net
Image generation using GANs (Generative Adversarial Networks)
AI-based super-resolution (ESRGAN, RealSR)
Hands-on labs using PyTorch/TensorFlow + OpenCV

List of laboratory tasks:

1. Perform basic image operations like read, write, and display using OpenCV.
2. Implement image enhancement using histogram equalization.
3. Apply various smoothing and sharpening filters to an image.
4. Convert color images to grayscale and binary formats.
5. Perform geometric transformations: rotation, scaling, and translation.
6. Add Gaussian, salt-and-pepper noise and apply noise removal filters.
7. Implement edge detection using Sobel, Prewitt, and Canny methods.
8. Segment an image using thresholding and region-based techniques.
9. Perform morphological operations: erosion, dilation, opening, and closing.
10. Extract features using contour detection and bounding boxes.
11. Analyze texture patterns using GLCM (Gray Level Co-occurrence Matrix).
12. Object detection using color segmentation and contour properties.
13. Image classification using histogram features and a simple classifier.
14. Real-time video processing using webcam and OpenCV.
15. Mini project: Build an end-to-end image processing pipeline for a selected application.

Project work/Assignment:

Assignment 1 on (Module 1 and Module 2)

Assignment 2 on (Module 3)

Mini project

Setup and Tools

Tool / Library	Purpose	Resource Link
Python 3.8+	Base language	https://www.python.org
OpenCV (cv2)	Classical image processing	https://opencv.org
NumPy + Matplotlib	Data handling + visualization	https://numpy.org + https://matplotlib.org
PyTorch	DL framework (preferred for vision)	https://pytorch.org
TensorFlow + Keras	Alternate DL framework	https://tensorflow.org
Google Colab	Free cloud notebooks with GPU	https://colab.research.google.com

Textbooks

- Gonzalez, R. C., Woods, R. E. (2018). Digital Image Processing. India: Pearson.
- Jain, A. K. (2005). Fundamentals of Digital Image Processing. India: Pearson Education.
- Annadurai, S. (n.d.). Fundamentals of Digital Image Processing. India: Pearson.
- Nixon & Aguado. Feature Extraction and Image Processing for Computer Vision, Elsevier, 2025.

Reference Books

- Jähne, B. (2014). Digital Image Processing. United States: Springer Berlin Heidelberg.
- Sonka, M., Hlavac, V., Boyle, R. (2015). Image Processing, Analysis, and Machine Vision. United States: Cengage Learning.
- Nixon, M., Aguado, A. (2025). Feature Extraction and Image Processing for Computer Vision. United Kingdom: Elsevier Science.

J OURNALS/MAGAZINES

1. IECE Journal of Image Analysis and Processing (JIAP):

2. Medical Image Analysis:

3. IPOL Journal (Image Processing On Line):

SWAYAM/NPTEL/MOOCs:

1. Digital Image Processing

2. Computer Vision and Image Processing – Fundamentals and Applications

3 Medical Image Analysis

Course Code: CAI3401	Course Name: Big Data Analytics for AI Type of Course: Theory Integrated Lab	L- T-P- C	2-0-2-3
Version No.	1		
NCrF Level-	5.5		
Course Pre-requisites	Nil		
Co-requisites	Nil		
Anti-requisites	Nil		
Course Description	This course introduces the integration of big data technologies with artificial intelligence, focusing on scalable data processing, distributed storage systems, and basic machine learning techniques. Students gain hands-on experience with tools such as Hadoop, Spark, and NoSQL databases to process large datasets and develop simple data-driven AI applications.		

<table>
<tr><td>Course Objective</td><td colspan="3">The objective of this course is to enable learners to understand and apply big data technologies for processing large datasets and building basic AI applications through hands-on and experiential learning.</td></tr>
<tr><td>Course Outcomes</td><td colspan="3">On successful completion of the course, the students shall be able to:
CO1: Explain the fundamentals of big data, its characteristics, and its role in AI applications. — Level 2 – Understand
CO2: Describe the architecture and components of distributed storage and computing frameworks such as Hadoop and Spark. — Level 2 – Understand
CO3: Implement data ingestion, preprocessing, and transformation workflows using industry-standard big data tools. — Level 3 – Apply
CO4: Apply Apache Spark for distributed data processing and basic machine learning tasks. — Level 3 – Apply
CO5: Analyze and interpret results of big data analytics and AI applications using visualization and case studies. — Level 4 – Analyze</td></tr>
<tr><td colspan="4">Course Content:I</td></tr>
<tr><td>Module 1</td><td>Foundations of Big Data and AI</td><td>Assessment: Quiz / Conceptual Case Study etc</td><td>6L + 6P = 12</td></tr>
<tr><td colspan="4">Topics:
Big Data fundamentals: Characteristics, challenges, and opportunities (5Vs), Overview of AI systems powered by big data, Big data lifecycle and its role in AI pipelines, Batch vs. stream processing in AI tasks, Architectures: Lambda, Kappa, and Delta</td></tr>
<tr><td>Module 2</td><td>Data Storage and Retrieval Systems</td><td>Assessment: Quiz / Conceptual Case Study etc</td><td>7L + 7P = 14</td></tr>
<tr><td colspan="4">Topics:
Hadoop Distributed File System (HDFS), Introduction to cloud storage (Amazon S3, Google Cloud Storage), NoSQL databases: MongoDB (document-based), Cassandra (columnar), Redis (key-value)
Introduction to Graph Databases (Neo4j) for AI reasoning, Data warehousing, lakes, and data lakehouses</td></tr>
<tr><td>Module 3</td><td>Data Ingestion, Integration, and Preprocessing</td><td>Assessment: Quiz / Conceptual Case Study etc</td><td>6L + 6P = 12</td></tr>
<tr><td colspan="4">Topics:
Data ingestion tools: Apache Kafka, Flume, NiFi, and Sqoop, ETL vs ELT: design and implementation,
Data cleaning, normalization, and transformation, Stream vs. batch ingestion design for AI readiness
Introduction to schema-on-read and schema-on-write</td></tr>
<tr><td>Module 4</td><td>Distributed Computing and Scalable Analytics</td><td>Assessment: Quiz / Conceptual Case</td><td>7L + 7P =</td></tr>
</table>

	with Spark with Machine Learning use cases	Study etc	14
Topics: Apache Spark architecture and execution model, RDDs, Data Frames, and Datasets, Spark SQL and performance optimization, Stream processing with Spark Streaming, Integration with Hive, HBase, and cloud data lakes, MLflow for experiment tracking and deployment, Use cases: recommender systems, fraud detection, predictive maintenance			
Module 5	**Interpretation, Visualization, and Applications of Big Data in AI**	Assessment: Quiz / Conceptual Case Study etc	4L + 4P = 8
Topics: Understanding outputs of big data processing and AI models, Basics of data visualization for large datasets (charts, dashboards using notebooks), Interpreting model results: accuracy, errors, and limitations (conceptual level), Communicating insights from big data (reports and storytelling), Introduction to responsible data usage (privacy, bias – basic concepts only), Real-world AI applications using big data: Recommendation systems, Fraud detection, Social media analytics			

List of Laboratory Tasks:

Lab 1: Hadoop Ecosystem and HDFS Operations

Objective: Explore the Hadoop ecosystem, configure HDFS, and manage distributed storage.

Tasks:

Set up a single-node or pseudo-distributed Hadoop cluster using Docker or VM.
Perform file operations: upload, read, delete, and list directories in HDFS.
Analyze HDFS block placement and replication factor through the NameNode web UI.
Compare HDFS with Amazon S3 or Google Cloud Storage in terms of performance and accessibility.

Lab 2: Working with NoSQL Databases

Objective: Operate and query NoSQL databases for big data storage and analytics.

Tasks:

Set up MongoDB and Cassandra; explore GUI tools like MongoDB Compass or Cassandra Studio.
Perform CRUD operations and create index structures.
Design a schema for storing semi-structured AI application data (e.g., chat logs, image tags).
Implement aggregation pipelines in MongoDB and execute time-series queries in Cassandra.

Lab 3: Data Ingestion and ETL Pipeline

Objective: Implement ingestion and preprocessing.

Tasks:

Simulate data ingestion (CSV/JSON)
Perform data cleaning and normalization
Apply ETL process and store processed data

Lab 4: Batch and Streaming Data Processing

Objective: Understand ingestion strategies.

Tasks:

Perform batch processing using Spark

Implement basic streaming using Spark Structured Streaming

Compare batch vs streaming outputs

Lab 5: Writing and Tuning MapReduce Jobs

Objective: Develop and optimize MapReduce programs for large-scale processing.
Tasks:

Implement a word count program and test with real-world datasets (e.g., Wikipedia dump).

Modify the job to calculate word frequency and filter stop words.

Analyze execution logs to understand job stages and bottlenecks.

Perform basic tuning using parameters like the number of mappers, block size, and combiner

Lab 6: Spark Programming and Data Processing

Objective: Apply Apache Spark for high-performance distributed analytics.
Tasks:

Install Spark locally or use a cloud-based environment (Databricks/Google Colab with PySpark).

Perform ETL operations using RDDs, DataFrames, and Spark SQL.

Implement window functions and joins on streaming or batch data.

Benchmark performance using Spark UI and understand memory usage.

Lab 7: Scalable Machine Learning with Spark MLlib

Objective: Build and evaluate machine learning models at scale using Spark.
Tasks:

Load real-world datasets (e.g., telecom churn, credit scoring) and perform preprocessing using Pipelines.

Implement classification (Logistic Regression), regression (Linear Regression), and clustering (KMeans).

Tune models using ParamGridBuilder and CrossValidator.

Track and compare experiment results using MLflow.

Lab 8: Result Analysis and Visualization of Big Data Applications

Objective: Focus on interpreting and presenting results

Tasks:

Use output from previous labs (Spark/MLlib models)

Visualize results using charts (matplotlib / notebook tools)

Evaluate model outputs (accuracy, confusion matrix)

Identify errors or limitations in predictions

Targeted Applications & Tools:	
Hadoop Distributed File System (HDFS)	Distributed, fault-tolerant storage system for managing large datasets.
Apache Spark (Core, SQL, Streaming, MLlib)	Scalable, in-memory distributed computing for data processing and machine learning tasks.
MongoDB and Cassandra	NoSQL databases for flexible, scalable storage of semi-structured and time-series data.
Neo4j (Graph Database)	Graph-based storage and querying for AI reasoning, recommendation, and relationship modeling.
Apache Kafka, NiFi, Flume, Sqoop	Tools for real-time and batch data ingestion into big data processing pipelines.
Amazon S3 / Google Cloud Storage	Cloud-based object storage for scalable AI and big data applications.
MLflow	Platform for tracking experiments, managing models, and simplifying deployment.
Databricks / Google Colab with PySpark	Cloud-based collaborative development environment for distributed AI pipelines.
Jupyter Notebooks and Streamlit	Tools for prototyping, interactive visualizations, and AI application interfaces.

Skill-Building Activities through Experiential Learning:

AI Ethics Case Simulation and Role Play

Participants analyze case studies (e.g., biased hiring algorithms, facial recognition misuse). Each group assumes the role of stakeholders—AI developer, policymaker, affected citizen, or ethicist—and debates decisions in a mock ethics review board.

Responsible Data Use Audit on a Public Dataset

Participants select an open dataset (e.g., health, social media, education) and audit it for potential privacy risks, representational bias, or consent issues. They document findings and propose mitigation strategies.

Design Challenge – Fair AI Model Prototyping

Using a Jupyter Notebook or a simple Streamlit app, groups build a basic classifier (e.g., for credit approval or school admissions) and apply fairness-aware preprocessing or post-processing methods. They visualize fairness metrics (e.g., demographic parity) and report on ethical trade-offs.

Text Book

T1: "Big Data: Principles and Best Practices of Scalable Real-Time Data Systems" Author: Nathan Marz, James Warren Publisher: Manning Publications

T2:WHITE, T. (2015). HADOOP: The Definitive Guide. India: SHROFF PUBLISHERS & DISTR.

T3: Damji, J. S., Wenig, B., Damji, J. S., Das, T., Lee, D. (2020). Learning Spark: Lightning-fast Data Analytics. United States: O'Reilly Media.

References

R1:Chambers, B., Zaharia, M. (2018). Spark: The Definitive Guide: Big Data Processing Made Simple. United States: O'Reilly Media.
R2:Kleppmann, M. (2017). Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. United States: O'Reilly Media.
R3:Leskovec, J., Rajaraman, A., Ullman, J. D. (2020). Mining of Massive Datasets. United Kingdom: Cambridge University Press.
R4:Koul, A., Ganju, S., Kasam, M. (2019). Practical Deep Learning for Cloud, Mobile, and Edge: Real-World AI & Computer-Vision Projects Using Python, Keras& TensorFlow. United States: O'Reilly Media.

E-Resources/Web Resources

1)https://www.coursera.org/in/articles/big-data-analytics
2)https://www.ibm.com/think/topics/big-data-analytics
3)https://aws.amazon.com/what-is/big-data/

Course Code: CAI3402	Course Title: Optimization Techniques for Machine Learning Type of Course: Theory Integrated Lab	L- T-P- C	2-0-2-3
Version No.	1.0		
Course Pre-requisites	CSS2264		
Anti-requisites	NIL		
Course Description	This course introduces a range of machine learning models and optimization tools that are used to apply these models in practice. Course will introduce what lies behind the optimization tools often used as a black box as well as an understanding of the trade-offs of numerical accuracy and theoretical and empirical complexity. For the students with some optimization background this course will introduce a variety of applications arising in machine learning and statistics as well as novel optimization methods targeting these applications.		

Course Objective	The objective of the course is to familiarize the learners with the concepts of Optimization Techniques for Machine Learning and attain Skill Development through Participative Learning techniques.			
Course Outcomes	On successful completion of this course the students shall be able to: CO1: Explain fundamental concepts of optimization techniques used in machine learning. — Level 2 – Understand CO2: Apply optimization models such as logistic regression, support vector machines, and coordinate descent to solve classification problems. — Level 3 – Apply CO3: Apply gradient-based and advanced optimization techniques, including SGD, momentum, RMSProp, and Newton-based methods, for machine learning tasks. — Level 2 – Understand CO4: Apply constrained optimization techniques, including Lagrangian methods and dual formulations, to solve machine learning problems such as SVM. — Level 3 – Apply CO5: Apply optimization techniques in computational graph frameworks, including DAG-based modelling and backpropagation, for real-world machine learning applications. — Level 3 – Apply			
Course Content:				
Module 1:	Optimization Fundamentals	Quiz	Knowledge based Quiz	10 [5L + 5P]
Introduction to Optimization in Machine Learning Bivariate and Multivariate Optimization Convex vs Non-convex Functions Properties of Convex Optimization Optimization Landscapes and Critical Points Least Squares Classification				
Module 2:	Optimization Models	Quiz	Comprehension based Quiz	12 [6L + 6P]
Logistic Regression Optimization Support Vector Machines (SVM) – Primal Form Binary Classification Optimization Models Multiclass Optimization Models Coordinate Descent Methods Model Evaluation and Trade-offs				
Module 3:	Optimization Techniques	Quiz	Comprehension based Quiz	16 [8L + 8P]
Gradient Descent and Stochastic Gradient Descent (SGD) Challenges in Gradient-Based Optimization Momentum-Based Optimization RMSProp and Adaptive Methods Newton Methods in Machine Learning Computationally Efficient Variations of Newton Method The Subgradient Method Proximal Gradient Method				

<table>
<tr><td colspan="5">Non-differentiable Optimization Functions: Designing Surrogate Lose Functions.</td></tr>
<tr><td>Module 4</td><td>Constrained Optimization</td><td>Assignment</td><td>Batch-wise Assignments</td><td>10 [5L + 5P]</td></tr>
<tr><td colspan="5">Constrained Optimization Basics
Primal Gradient Descent Methods
Lagrangian Relaxation and Duality
Fundamentals of SVM Dual
Optimization Algorithms for SVM Dual</td></tr>
<tr><td>Module 5:</td><td>Optimization in Computational Graphs</td><td>Assignment and Presentation</td><td>Batch-wise Assignment and Presentations</td><td>12 [6L + 6P]</td></tr>
<tr><td colspan="5">Optimization in Directed Acyclic Graphs (DAGs)
Broad Framework for Computational Graph Optimization
Node-to-Node Derivations using Brute Force
Concept of Backpropagation in Graphs (implicit connection)</td></tr>
<tr><td colspan="5">Targeted Application & Tools that can be used: Use of Matlab tool</td></tr>
<tr><td colspan="5">Project work/Assignment:
Survey on Methods for convex optimization
Survey on Machine learning models related to optimization</td></tr>
<tr><td colspan="5">Introduction to Optimization Problems using Python/Matlab.
Implement Bivariate and Multivariate Optimization.
Solve Least-Square Classification Problem.
Implement Support Vector Machine (SVM) Optimization.
Logistic Regression Model Optimization.
Coordinate Descent Algorithm Implementation.
Gradient Descent and Stochastic Gradient Descent Techniques.
Implement Momentum-based Gradient Descent.
RMSProp Optimization Method Application.
Newton Method Implementation for Machine Learning.
Subgradient Method for Non-differentiable Functions.
Proximal Gradient Method Implementation.
Solve Constrained Optimization Problems with Lagrangian Methods.
Optimization in Directed Acyclic Graphs.
Survey and Comparative Analysis of Optimization Algorithms.</td></tr>
</table>

Text Book T1. Charu C. Aggarwal, " Linear Algebra and Optimization for Machine Learning", Springer, 2020. T2. Sra Suvrit, Nowozin Sebastian, and Wright Stephen J, "Optimization for Machine Learning", The MIT Press,2012.
References R1.Guanghui Lan, "First-order and Stochastic Optimization Methods for Machine Learning", Springer Cham, 2020. Web References W1. https://sm-nitk.vlabs.ac.in/ W2. https://nptel.ac.in/courses/

Course Code: CAI3403	Course Title: Deep Reinforcement Learning Type of Course: Theory Integrated Lab	L-T-P-C	2-0-2-3
Version No.	1.0		
Course Pre-requisites	CAI2507		
Anti-requisites	NIL		
Course Description	This course focuses on advanced concepts in Deep Reinforcement Learning (DRL), emphasizing the integration of reinforcement learning with deep neural networks for solving complex decision-making problems. It covers function approximation techniques, value-based methods such as Deep Q-Networks (DQN), policy gradient methods, and actor-critic architectures including Proximal Policy Optimization (PPO). The course also addresses continuous control problems and modern DRL algorithms such as DDPG and Soft Actor-Critic (SAC). Emphasis is placed on both theoretical understanding and practical implementation using Python, PyTorch, and Gymnasium environments.		
Course Objectives	To understand function approximation techniques in reinforcement learning using neural networks To design and implement value-based deep reinforcement learning algorithms such as DQN and its variants To develop and analyze policy gradient and actor-critic methods for complex environments To apply deep reinforcement learning techniques to discrete and continuous control		

	problems To build and evaluate DRL models for real-world decision-making applications	
Course Outcomes	On successful completion of the course the students shall be able to: CO1: Explain function approximation and the role of neural networks in reinforcement learning. — L2 – Understand CO2: Design and implement value-based deep reinforcement learning algorithms such as DQN and its variants. — L3 – Apply CO3: Apply policy gradient methods to learn optimal policies in complex environments. — L3 – Apply CO4: Develop and evaluate actor-critic models such as A2C and PPO for reinforcement learning tasks. — L3 – Apply CO5: Apply advanced deep reinforcement learning algorithms to solve discrete and continuous decision-making problems. — L3 – Apply	
Course Content:		
Module 1	Function Approximation in Reinforcement Learning	L-5, P-6
Function approximation in RL: Motivation and challenges Supervised vs reinforcement learning with function approximation Linear function approximation Non-linear approximation using neural networks Feature representation and embeddings for RL Stability and divergence issues in RL with function approximation Batch vs online learning in RL Introduction to deep neural networks for RL		
Module 2	Deep Q-Networks and Value-Based Deep RL	L-5 P-6
Q-learning with neural networks Deep Q-Network (DQN): Architecture and algorithm Experience Replay: mechanism and benefits Target Networks for stable learning Training pipeline for DQN agents Exploration strategies in deep RL Extensions of DQN: Double DQN, Dueling DQN, Prioritized Experience Replay Performance evaluation and limitations		
Module 3	Policy Gradient Methods	L-6 P -7
Policy-based reinforcement learning Policy Gradient Theorem REINFORCE algorithm Gradient estimation and optimization challenges Stochastic vs deterministic policy gradients Variance reduction techniques (baseline, advantage functions) Implementation of policy gradient methods in continuous and discrete environments		

Module 4	Actor-Critic and Modern DRL Algorithms	L-4 P -5
Actor-Critic framework: architecture and intuition Advantage Actor-Critic (A2C) Asynchronous Advantage Actor-Critic (A3C) – overview Proximal Policy Optimization (PPO): Clipped surrogate objective, Stability and efficiency Trust Region Policy Optimization (TRPO) – overview Comparison of value-based vs policy-based vs actor-critic methods		
Module 5	Continuous Control and Advanced Deep RL	L-8 P -6
Challenges in continuous action spaces Deep Deterministic Policy Gradient (DDPG) Twin Delayed DDPG (TD3) – overview Soft Actor-Critic (SAC) – overview Exploration in continuous domains Multi-agent reinforcement learning (introductory concepts) Transfer learning and generalization in RL Real-world applications: robotics, autonomous driving, recommendation systems		

List of Lab Tasks:

Lab 1: Function Approximation in Reinforcement Learning

Objective: Implement value function approximation
Tasks:

- Implement linear function approximation for value prediction
- Extend to non-linear approximation using neural networks
- Compare tabular vs function approximation performance
- Analyze convergence behavior

Lab 2: Implementing Deep Q-Network (DQN)

Objective: Train a DQN agent
Tasks:

- Build a DQN agent for CartPole environment
- Implement experience replay buffer
- Train and evaluate performance
- Plot rewards vs episodes

Lab 3: Improving DQN Performance

Objective: Explore stability improvements
Tasks:

- Implement Target Network in DQN
- Extend to Double DQN
- Compare DQN vs Double DQN performance
- Analyze overestimation bias

Lab 4: Dueling DQN and Advanced Exploration

Objective: Enhance value-based methods
Tasks:

- Implement Dueling DQN architecture
- Experiment with epsilon-greedy vs softmax exploration
- Compare learning efficiency
- Evaluate policy robustness

Lab 5: Policy Gradient – REINFORCE Algorithm

Objective: Implement policy-based learning
Tasks:

- Implement REINFORCE algorithm
- Train agent on CartPole or similar environment
- Analyze variance in training
- Plot policy performance

Lab 6: Actor-Critic Methods (A2C)

Objective: Implement hybrid RL architecture
Tasks:

- Implement Advantage Actor-Critic (A2C)
- Train on a Gym environment
- Compare with REINFORCE performance
- Study bias-variance trade-off

Lab 7: Proximal Policy Optimization (PPO)

Objective: Train a stable and efficient RL agent
Tasks:

- Implement PPO (or use a standard library like Stable-Baselines3)
- Train on CartPole / LunarLander
- Analyze clipped objective behavior
- Compare PPO vs A2C

Lab 8: Continuous Control using DDPG

Objective: Work with continuous action spaces
Tasks:

- Implement DDPG algorithm
- Train on a continuous control environment (Pendulum)
- Understand actor-critic interaction
- Evaluate performance challenges

Lab 9: Advanced DRL Algorithms (TD3 / SAC – Overview Implementation)

Objective: Explore modern DRL techniques
Tasks:

- Implement or use TD3 / SAC from libraries

- Compare with DDPG performance
- Analyze stability and exploration improvements

Lab 10: Mini Project – DRL Agent Design

Objective: Apply DRL to a custom problem

Targeted Application & Tools that can be used:

Execution of Deep Reinforcement Learning algorithms will be carried out using environments provided by Gymnasium (Farama Foundation) in Google Colab (https://colab.research.google.com/) or Jupyter Notebook. Lab tasks will be implemented using Python-based libraries such as PyTorch and supporting tools for building and training DRL models.

Text Book

T1: Richard S. Sutton and Andrew G. Barto, "Reinforcement Learning: An Introduction", MIT press, Second Edition, 2018.

T2: Sudharshan Ravichandiran, "Deep Reinforcement Learning with Python", Packt Publishers, Second Edition, 2020

References

R1: Laura Graesser and Wah Loon Keng, *Foundations of Deep Reinforcement Learning*, Pearson, 2022.
R2: Maxim Lapan, *Deep Reinforcement Learning Hands-On*, Packt Publishing, 2018 (or latest edition).
R3: OpenAI Spinning Up, *Deep Reinforcement Learning Documentation*, 2018.

Web resources

https://www.udemy.com/course/artificial-intelligence-reinforcement-learning-in-python/

https://spinningup.openai.com/en/latest/

https://gymnasium.farama.org/

https://pytorch.org/tutorials/

https://www.udemy.com/course/artificial-intelligence-reinforcement-learning-in-python/

Course Code: CAI3404	Course Name: **AI in Cybersecurity** Type of Course: Theory Integrated Lab	L- T-P- C	2-0-2-3
Version No.	1.0		
NCrF Level-	5.5		
Course Pre-requisites	CSS2264		
Co-requisites	Nil		
Anti-requisites	Nil		

Course Description	This course explores the integration of Artificial Intelligence in cybersecurity, focusing on threat detection, anomaly analysis, malware classification, and threat intelligence. It covers machine learning, deep learning, and natural language processing techniques applied to real-time security challenges, enhancing proactive defense mechanisms and automated incident response across digital infrastructure.		
Course Objective	This course is designed to improve the learners 'SKILL DEVELOPMENT' by using EXPERIENTIAL LEARNING techniques		
Course Outcomes	On successful completion of the course, the students shall be able to: CO1: Explain fundamental cybersecurity concepts, AI techniques, and different types of threats. — Level 2 – Understand CO2: Explain how AI methods enhance anomaly detection, intrusion prevention, and cyber defense. — Level 2 – Understand CO3: Apply machine learning and deep learning techniques for malware detection and classification. — Level 3 – Apply CO4: Analyze cyber threat intelligence using NLP and behavioral analytics for automated response. — Level 4 – Analyze CO5: Explain the limitations, risks, and ethical considerations of AI-based cybersecurity systems. — Level 2 – Understand		
Course Content:			
Module 1	**Introduction to AI in Cyber Security**	Assessment: Assignment	12 (6L + 6P)
Overview of cybersecurity: CIA triad, threats, vulnerabilities, Role of AI in cyber defense, cyber kill chain, and defense strategies, ML/DL basics for cyber applications, Data sources: system logs, NetFlow, packet capture, Threat Intelligence Platforms (TIPs), AI use cases in industry; deployment challenges (e.g., adversarial ML, data imbalance)			
Module 2	**Anomaly and Intrusion Detection Systems**	Assessment: Assignment	14 (7L + 7P)
Intrusion Detection Systems: signature-based vs. anomaly-based, Cyber security datasets: KDDCup, NSL-KDD, CICIDS2017, Data preprocessing: encoding, normalization, feature engineering, ML for intrusion detection: SVM, Random Forest, Decision Trees, Unsupervised learning: K-Means, Isolation Forests, Autoencoders, Evaluation metrics: precision, recall, ROC, Real-time detection: Kafka, PySpark, Case Study: ML-based IDS implementation			
Module 3	**Malware Detection and Classification**	Assessment: Assignment	12 (6L + 6P)
Malware types: virus, ransomware, spyware, trojans, Static vs. dynamic analysis, Feature extraction: opcode, API sequences, binary patterns, DL techniques: CNN for image-based, LSTM for sequence learning, Training with Malimg and Microsoft Malware datasets, Optimization: dropout, tuning, early stopping, Malware embeddings and adversarial evasion techniques			
Module 4	**AI for Cyber Threat Intelligence and Response**	Assessment: Assignment	12 (6L + 6P)
Threat intelligence sources: OSINT, commercial feeds, NLP for cyber threat intelligence (CTI), Entity			

recognition from security reports, ML/NLP for phishing URL and email detection, Behavioral biometrics: keystroke and mouse dynamics, DL for SIEM systems, AI-powered incident response and SOAR tools, Case Study: Detecting phishing websites with NLP and ensemble models

Module 5	**Security and Limitations of AI in Cybersecurity**	Assessment: Assignment / Case Study	10 (5L + 5P)

Introduction to security risks in AI-based systems, Adversarial attacks on AI models, Data poisoning and evasion attacks, Limitations of AI in cybersecurity systems, Issues of false positives and false negatives in detection systems, Basic concepts of explainability in AI models, Ethical and privacy concerns in AI-based cybersecurity systems

Lab experiments:

Lab 1: Cybersecurity Dataset Exploration

- Load datasets such as NSL-KDD or CICIDS2017
- Analyze features, attack types, and class distribution
- Idenify imbalance in data

Lab 2: Exploring Cyber Data Sources

- Work with sample system logs / network traffic data
- Understand structure of NetFlow / log data
- Perform basic visualization of events

Lab 3: Data Preprocessing for Intrusion Detection

- Perform encoding, normalization, and feature engineering
- Prepare dataset for ML models

Lab 4: Intrusion Detection using Supervised Learning

- Train models such as Decision Tree / Random Forest
- Evaluate using accuracy, precision, recall

Lab 5: Unsupervised Anomaly Detection

- Apply K-Means and Isolation Forest
- Detect anomalies and compare results

Lab 6: Model Evaluation and Performance Analysis

- Generate confusion matrix
- Plot ROC curve
- Compare multiple IDS models

Lab 7: Malware Dataset Analysis

- Load malware dataset (Malimg / similar)

- Visualize different malware classes
- Explore feature patterns

Lab 8: Malware Classification using Deep Learning

- Implement a simple CNN or LSTM model
- Train and evaluate classification performance

Lab 9: Phishing Detection using NLP

- Preprocess email or URL dataset
- Apply basic NLP techniques
- Train a classifier to detect phishing

Lab 10: Threat Intelligence Extraction

- Extract entities from threat reports
- Identify patterns using NLP techniques
- Visualize extracted information

Targeted Applications & Tools:

Tools/Software/Platform	**Application**
Scikit-learn, TensorFlow, Keras, Pandas	Develop and evaluate AI models for threat detection, malware classification, and anomaly analysis.
Wireshark, Tcpdump	Capture and analyze network traffic to generate datasets for intrusion detection experiments.
NSL-KDD, CICIDS2017, Malimg, Microsoft Malware Datasets	Real-world benchmark datasets for building and testing intrusion and malware detection models.
PySpark, Apache Kafka	Stream processing for real-time anomaly detection in large-scale network environments.
ELK Stack (Elasticsearch, Logstash, Kibana)	Collect, analyze, and visualize security logs and threat patterns.
Snort, Suricata	Open-source Intrusion Detection Systems to generate labeled security event data.
Jupyter Notebook, Google Colab	Develop, document, and share AI-driven cybersecurity experiments and workflows.
NLTK, SpaCy	Perform NLP tasks for extracting threat intelligence from textual data like emails or reports.

TensorBoard, Seaborn, Matplotlib	Visualize model performance, threat patterns, and feature distributions for interpretation.
Streamlit, Flask	Build interactive dashboards for real-time visualization of AI-based threat detection and alerts.

Skill-Building Activities through Experiential Learning:

- **AI-Powered Intrusion Detection System Development.**
  Design and deploy a machine learning-based intrusion detection system using real-world datasets like NSL-KDD. Participants pre-process data, build supervised learning models (SVM, Random Forest), and evaluate detection performance. This hands-on activity provides practical exposure to applying AI for real-time threat detection in network environments.
- **Malware Classification with Deep Learning.**
  Build and train a deep learning model using CNN or LSTM to classify malware based on extracted features like opcode patterns or API calls. Learners gain experience in feature engineering, model design, optimization, and evaluation for malware detection using real-world datasets such as Mali mg.
- **Cyber Threat Intelligence with NLP and Dashboards.**
  Apply NLP techniques to extract key entities from threat reports and detect phishing URLs or emails. Develop an AI-driven threat intelligence dashboard integrating real-time detection, visualization, and alerts. This activity enhances practical skills in NLP, security analytics, and dashboard development for proactive cyber defences.

Text Book

T1. Stamp, M. Introduction to Machine Learning with Applications in Information Security. United States: CRC Press. 2022.
T2. Chio, C., Freeman, D. Machine Learning and Security: Protecting Systems with Data and Algorithms. United States: O'Reilly Media. 2018.

References

R1: Xiaofeng Chen, Cyber Security: AI and Big Data Perspective, Springer, 2021.

R2: Sumeet Dua, Xian Du, Data Mining and Machine Learning in Cybersecurity, CRC Press, 2011.

R3: Géron, A., *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, O'Reilly Media, 2019.

R4: Abou El Kalam, A., et al., *Deep Learning for Cybersecurity*, Springer, 2021.

E-Resources/Web Resources:

- **IEEE Transactions on Information Forensics and Security**
  https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=8858
  Covers the theory and practice of information forensics, cyber defense, and security.
- **ACM Transactions on Privacy and Security (TOPS)**
  https://dl.acm.org/journal/tops
  Peer-reviewed research on cyber security systems, privacy-preserving AI, and secure protocols.
- **Computers& Security (Elsevier)**
  https://www.sciencedirect.com/journal/computers-and-security
  Practical and academic articles on cyber threats, security analytics, and AI applications in security.
- **Journal of Cybersecurity (Oxford Academic)**
  https://academic.oup.com/cybersecurity

Multidisciplinary research on digital threats and AI-driven defensive mechanisms.

- **IEEE Security& Privacy Magazine**
  https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=8013
  Blends academic depth with industry relevance; covers AI, forensics, and emerging cyber threats.
- **Cybersecurity Magazine**
  https://cybersecurity-magazine.com
  Regularly features expert opinion, trends, and technologies including AI in cyber security.

NPTEL:

- NPTEL – Introduction to Machine Learning (IIT Kharagpur)
  https://onlinecourses.nptel.ac.in/noc22-cs58
- AI for Cybersecurity Specialization – IBM

Coursera: https://www.coursera.org/specializations/ai-cybersecurity
Covers: AI and ML techniques to detect malware, phishing, and threats in network traffic. Hands-on labs included.

Course Code: CAI3405	Course Name: Explainable AI Type of Course: Theory Integrated Lab	L- T-P- C	2-0-2-3
Version No.	1.0		
NCrF Level-	5.5		
Course Pre-requisites	CAI2502		
Co-requisites	Nil		
Anti-requisites	Nil		
Course Description	This course explores techniques and tools that enhance the transparency and interpretability of AI models. It emphasizes ethical considerations, model accountability, and post-hoc explanation methods. Through hands-on work with real-world datasets, the course develops the ability to design AI systems that are explainable, trustworthy, and aligned with regulatory standards.		
Course Objective	The course aims to help students understand the need for and principles of explainable AI, explore techniques for interpreting model predictions, gain hands-on experience with modern XAI tools and libraries, and develop interpretable models that meet ethical and regulatory standards.		
Course Outcomes	**On successful completion of the course, the students shall be able to:** **CO1:** Describe the importance of explainability in AI, including ethical and regulatory considerations. — Level 2 – Understand **CO2:** Differentiate between interpretable and black-box models based on performance and transparency. — Level 4 – Analyze **CO3:** Use post-hoc techniques such as LIME, SHAP, and feature importance methods to explain AI model outputs. — Level 3 – Apply		

	CO4: Demonstrate visual and textual explanation methods for image, text, and tabular data. **CO5:** Apply explainability tools in developing AI-based applications to enhance understanding and decision-making.	Level 3 – Apply Level 3 – Apply

Course Content: **I**

Module 1	Foundations of Explainable A	Assessment : Quiz / Conceptual Case Study etc	06 Sessions

Why Explainability? Historical context, challenges, Ethical AI, fairness, and trust ,Human-AI collaboration and regulatory frameworks (GDPR, EU AI Act) , Introduction to Explainability metrics (fidelity, completeness, consistency

Module 2	Interpretable vs. Black-box Models	Assessment : Quiz / Conceptual Case Study etc	06 Sessions

Glass-box models: Decision trees, linear models, symbolic AI ,Complex models: Deep neural networks, ensembles ,Trade-offs between accuracy, interpretability, and scalability,Regularization and sparsity for interpretability,
Lab Activities: Comparative model training, metric analysis, trust assessment

Module 3	Post-hoc Explanation Techniques and Toolkits	Assessment : Quiz / Conceptual Case Study etc	06 Sessions

Local and global explanations ,LIME, SHAP, Anchors, Counterfactual explanations ,Integrated Gradients, Grad-CAM, DeepLIFT

Introduction to tools: L2X, What-If Tool, Captum, IBM AI360, Lab Activities: Implementation of SHAP & LIME, visualization and storytelling.

Module 4	Explainability in Domain-Specific Models	Assessment : Quiz / Conceptual Case Study etc	06 Sessions

Explainable NLP: Attention mechanisms, transformers, prompt tracing,Vision models: Saliency maps, Grad-CAM for CNNs ,

Time series and anomaly detection models ,Case studies: Healthcare, finance, hiring, credit scoring ,Lab Activities: NLP/XAI integration, Vision Explainability project.

Module 5	Generative AI, Fairness & Future of XAI	Assessment : Quiz / Conceptual Case Study etc	06 Sessions

Explainability in Generative AI (e.g., GANs, LLMs) ,Challenges in interpreting transformers and diffusion models ,XAI for fairness auditing and de-biasing ,Future directions: Human-centric XAI, Responsible AI

pipelines ,Lab Activities: Prompt explanations in LLMs, fairness toolkits (FairLearn, AIF360)

Lab Experiments

1. Comparative Analysis of Glass-box and Complex Models.
2. Design Deep Learning and Ensemble Models using Hybrid Techniques.
3. Comparative Study of Local and Global Explanation Methods.
4. Visual Explanations for Deep Neural Networks.
5. Understanding Model Decisions through Counterfactual and What-If Analysis.
6. Interpreting Transformer-based NLP Models using Attention and Prompt Tracing.
7. Visual Interpretation of CNN Models using Saliency and Activation Mapping.
8. Interpreting Anomalies in Time-Series Data with Real-world Applications.
9. Understanding and Explaining Prompt Behavior in Large Language Models.
10. Interpreting and Auditing Bias in Generative Models.
11. Designing a Human-Centric Explainable and Fair AI Pipeline

Targeted Applications & Tools:

Tools/Software/Platform	Application
SHAP, IBM AI Explainability 360 Toolkit	Apache Pig, HDFS
Hadoop MapReduce, Eclipse or IntelliJ IDEA	Interpret CNN-based image classification decisions in a medical imaging use case.
Apache Hive, HDFS, Beeline or Hive CLI	Evaluate fairness and transparency in an AI hiring tool
Apache Pig, HDFS	Build an explainable model for predicting student dropout risk
SHAP + Dash/Streamlit/Flask Web App	Develop a dashboard for explainable fraud detection in banking
Alibi (for drift + explanation), SHAP	Analyze model behavior changes due to data drift in loan approval models
LIME, SHAP, Counterfactual Explanations	Demonstrate transparency in an AI-based health insurance claim approval system
Decision Tree vs. XGBoost + SHAP	Compare interpretable vs. black-box models on same dataset (e.g., loan approval)
HuggingFace Transformers, Attention Heatmaps	Visualize attention weights in NLP sentiment analysis system
DiCE (Diverse Counterfactual Explanations), Alibi	Generate counterfactual explanations for denied loan cases

Skill-Building Activities through Experiential Learning:

Explainability Hackathon
Participants collaboratively build AI models using real-world datasets and integrate explainability

tools like SHAP or LIME. The activity emphasizes generating transparent model insights and communicating them effectively through dashboards. This hands-on challenge promotes technical skills, teamwork, and the ability to translate complex AI outputs for end-users.

Peer Review of Model Explanations

Learners apply XAI techniques to their models and engage in structured peer reviews. They critique each other's explanations for clarity, correctness, and potential bias. This activity fosters critical thinking, enhances understanding of model interpretability tools, and builds the ability to communicate AI outputs responsibly to diverse audiences.

Design Thinking for Ethical AI

Interdisciplinary teams apply design thinking principles to develop AI solutions for societal problems with built-in fairness and explainability. Using tools like SHAP, LIME, and user interface prototypes, teams explore ethical challenges and propose transparent AI applications suitable for real-world deployment in sensitive domains like healthcare or finance.

Text Book

1. Molnar, C. (2022). Interpretable Machine Learning: A Guide for Making Black Box Models Explainable. Germany: INDEPENDENTLY PUBLISHED.
2. Sameer Singh et al. – Explainable AI: A Guide for Practitioners
3. Gunning &Aha – DARPA's XAI Program Publications

References

R1. IEEE XAI publications

R2. Research papers from NeurIPS, ICML, and ACL on XAI

R3. XAI Fairness & Bias Toolkits by Google, IBM, and Microsoft

E-Resources/Web Resources

NPTEL: Responsible AI by IIT Madras
Coursera: Explainable AI with Google Cloud
FastAI: Modules on Model Interpretation
IBM AI Explainability 360 Toolkit

Course Code:	Course Name: Responsible AI	L- T-P- C	2-0-2-3

CAI3406	Type of Course: Theory Integrated Lab		
Version No.	1.0		
NCrF Level-	5.5		
Course Pre-requisites	CAI2502		
Co-requisites	Nil		
Anti-requisites	Nil		
Course Description	This course explores techniques and tools that enhance the transparency and interpretability of AI models. It emphasizes ethical considerations, model accountability, and post-hoc explanation methods. Through hands-on work with real-world datasets, the course develops the ability to design AI systems that are explainable, trustworthy, and aligned with regulatory standards.		
Course Objective	The objective of Responsible AI is to develop and deploy artificial intelligence in a way that is ethical, fair, transparent, and aligned with human values.		
Course Outcomes	On successful completion of this course, the students shall be able to: CO1: Explain core principles, ethical foundations, and regulatory frameworks of responsible AI. — Level 2 – Understand CO2: Explain sources of bias and methods to assess fairness in AI models. — Level 2 – Understand CO3: Apply interpretability and explainability techniques across different AI systems. — Level 3 – Apply CO4: Apply bias mitigation techniques during data preprocessing and model training. — Level 3 – Apply CO5: Implement privacy-preserving approaches such as differential privacy and federated learning in real-world scenarios. — Level 3 – Apply		
Course Content:			
Module 1	Foundations of Responsible AI	Assessment : Quiz / Conceptual Case Study etc	8 (5L + 3P)
What is Responsible AI? Core values: fairness, transparency, accountability, privacy ,Legal and policy frameworks (e.g., GDPR, EU AI Act, India's DPDP Bill) ,Introduction to AI alignment and value-sensitive design ,Risks and harms of misaligned or opaque AI systems Lifecycle of Responsible AI: Design, Deployment, Monitoring mpleteness, consistency			
Module 2	Fairness, Bias, and Equity	Assessment : Quiz / Conceptual Case Study etc	14 (6L + 8P)
Sources of bias: historical, representational, measurement, aggregation , Fairness definitions: statistical parity, equalized odds, counterfactual fairness ,Bias mitigation methods: Pre-processing (reweighing, sampling), In-processing (adversarial debiasing), Post-processing ,Fairness across demographics,			

<table>
<tr><td colspan="4">intersectionality, and contexts (e.g., hiring, lending)

Auditing tools: AIF360, Fairlearn, What-If Tool.</td></tr>
<tr><td>Module 3</td><td>Interpretability, Explainability, and Accountability</td><td>Assessment : Quiz / Conceptual Case Study etc</td><td>12 (5L + 7P)</td></tr>
<tr><td colspan="4">Transparent vs. opaque models, Intrinsic vs. post-hoc interpretability , Visual, textual, and feature attribution methods (LIME, SHAP, Grad-CAM, Counterfactuals) , Model documentation: datasheets for datasets, model cards , AI accountability, algorithmic auditing, and stakeholder reporting.</td></tr>
<tr><td>Module 4</td><td>Privacy, Security & Sustainability in AI</td><td>Assessment : Quiz / Conceptual Case Study etc</td><td>12 (6L + 6P)</td></tr>
<tr><td colspan="4">Threats: model inversion, membership inference, data leakage,Differential privacy, federated learning, secure multi-party computation ,

Green AI: energy efficiency, carbon impact of training LLMs ,Sustainable AI lifecycle practices ,Real-world case studies in healthcare, finance, and education</td></tr>
<tr><td>Module 5</td><td>Responsible Generative AI & Societal Impact</td><td>Assessment : Quiz / Conceptual Case Study etc</td><td>14 (8L + 6P)</td></tr>
<tr><td colspan="4">Ethical issues in generative models (LLMs, GANs, diffusion models) ,Prompt injection, hallucination, misuse prevention ,Explainability in LLMs and safety alignment ,Human-in-the-loop decision-making ,Future directions: Responsible AGI, alignment research, cultural bias</td></tr>
<tr><td colspan="4">Lab experiments:

Lab 1: Exploring Ethical Issues in AI Systems

Objective: Understand real-world ethical challenges.

Tasks:
• Analyze a sample AI application (e.g., hiring or recommendation system)
• Identify issues related to fairness, bias, and transparency
• Document risks and propose basic mitigation strategies

Lab 2: Bias Detection in Dataset

Objective: Identify bias in data.

Tasks:
• Load a dataset (e.g., loan approval / hiring dataset)
• Analyze class distribution across demographics
• Identify imbalance and bias</td></tr>
</table>

Lab 3: Fairness Evaluation using Tools

Objective: Evaluate fairness metrics.

Tasks:

- Use Fairlearn or AIF360
- Compute fairness metrics (statistical parity, equal opportunity)
- Visualize disparities

Lab 4: Bias Mitigation Techniques

Objective: Reduce bias in models.

Tasks:

- Apply preprocessing techniques (reweighing / sampling)
- Train model before and after mitigation
- Compare fairness and accuracy

Lab 5: Model Explanation using LIME/SHAP

Objective: Interpret model predictions.

Tasks:

- Train a simple ML model
- Apply LIME or SHAP
- Explain individual predictions

Lab 6: Feature Importance and Visualization

Objective: Understand feature contributions.

Tasks:

- Compute feature importance
- Visualize using plots
- Interpret model behavior

Lab 7: Model Documentation and Reporting

Objective: Practice accountability.

Tasks:

- Create a simple model card
- Document dataset, model performance, risks
- Prepare a short report

Lab 8: Differential Privacy Basics

Objective: Understand privacy-preserving techniques.

Tasks:

- Add noise to dataset or outputs
- Observe impact on model performance
- Compare privacy vs accuracy

Lab 9: Responsible Prompt Design

Objective: Improve AI outputs responsibly.

Tasks:

- Design prompts to reduce bias and hallucination
- Compare outputs before and after refinement
- Document improvements

Lab 10: End-to-End Responsible AI Audit and Evaluation

Objective: Perform a comprehensive audit of an AI model covering fairness, explainability, and risk assessment.

Tasks:

- Select or train a machine learning model on a real-world dataset (e.g., loan approval, hiring)
- Evaluate model performance (accuracy, precision, recall)
- Analyze fairness using metrics (e.g., statistical parity or equal opportunity)
- Apply explainability techniques (SHAP or LIME) to interpret predictions
- Identify bias, limitations, and potential risks

- Propose mitigation strategies and improvements
- Prepare a structured audit report summarizing findings

Targeted Applications & Tools:

Tools/Software/Platform	Application
Tweepy (Twitter API), scikit-learn, Fairlearn, Matplotlib	Real-time sentiment monitoring on trending topics with bias detection
Dialogflow, Rasa, spaCy, ExplainNLP	Build a transparent AI chatbot that justifies its responses
Surprise library, TensorFlow Recommenders, Fairlearn, Pandas	Adaptive recommendation engine with demographic fairness constraints
River (online ML), scikit-multiflow, SHAP	Online fraud classifier with concept drift detection
OpenCV, DeepFace, FER2013 Dataset, Differential Privacy Libraries	Emotion detection from video feed with ethical safeguards
TensorFlow Lite, Android Studio /Raspberry Pi, Quantization Toolkit	Deploy a low-latency, privacy-focused ML model to edge devices
PyCaret, SHAP, LIME, Kafka Streams, Flask	Explainable real-time fraud detection in banking data
Streamlit, SHAP, Fairlearn, TensorFlow, and GDPR audit toolkit	Streamlit, SHAP, Fairlearn, TensorFlow, and GDPR audit toolkit

Skill-Building Activities through Experiential Learning:

Responsible AI Audit Challenge

Participants analyze a pre-trained AI model (e.g., resume shortlisting or loan approval) to uncover ethical flaws, bias, or lack of transparency. They present an audit report with recommendations.

AI Bias Bounty Simulation

Participants act as "bias hunters" and compete to find and fix fairness issues in a dataset or model (e.g., facial recognition, sentiment analysis). Each group defends their solution based on evidence and social impact.

Privacy-Aware AI Design Sprint

Small groups prototype an AI-based solution for a real-world problem (e.g., mental health chatbot, educational recommender) using privacy-preserving techniques like differential privacy or federated learning.

Text Book

1. Dignum, V. (2020). Responsible Artificial Intelligence: How to Develop and Use AI in a Responsible Way. Switzerland: Springer International Publishing.
2. Molnar, C. (2022). Interpretable Machine Learning: A Guide for Making Black Box Models

Explainable. Germany: Christoph Molnar.
References The Cambridge Handbook of Responsible Artificial Intelligence: Interdisciplinary Perspectives. (2022). United States: Cambridge University Press.
E-Resources/Web Resources Responsible AI for generative models: Designing for responsibility Responsible AI Microsoft Responsible AI - Fairness https://aws.amazon.com/ai/responsible-ai/

Course Code: CAI3407	Course Name: **Agentic AI** Type of Course:Theory Integrated Lab	L- T-P- C	2-0-2-3
Version No.	1.0		
NCrF Level-	5.5		
Course Pre-requisites	CSS2264		
Co-requisites	Nil		
Anti-requisites	Nil		
Course Description	This course provides a comprehensive introduction to Agentic AI, focusing on building autonomous, tool-using systems powered by Large Language Models (LLMs). Students will explore the foundations of multi-step LLM workflows, prompt engineering, and asynchronous execution while gaining hands-on experience with leading frameworks like OpenAI API, CrewAI, LangGraph, and AutoGen. Through modules on multi-agent collaboration, graph-based agent design, and model context protocols, the course emphasizes practical skills in designing, orchestrating, and deploying intelligent agents capable of communication, memory management, and decision-making. Real-world case studies across domains such as robotics, finance, and smart cities will illustrate the transformative potential of agentic systems.		
Course Objective	This course aims to introduce the principles of Agentic AI by enabling students to design, develop, and evaluate intelligent autonomous agents that can perceive their environment, make decisions, and act effectively using AI techniques such as		

	planning, reasoning, and learning.		
Course Outcomes	On successful completion of this course the students shall be able to: **CO1:** Explain the fundamentals of Agentic AI, including multi-step LLM workflows, prompt engineering, asynchronous execution, and tool-based function calling. — Level 2 – Understand **CO2:** Design and implement multi-agent systems using CrewAI, incorporating task coordination, memory, communication, and collaboration strategies. — Level 3 – Apply **CO3:** Construct graph-based agent workflows using LangGraph, leveraging state machines, reducers, and checkpointing. — Level 3 – Apply **CO4:** Manage agent messaging protocols using AutoGen and Model Context Protocol (MCP). — Level 4 – Analyze **CO5:** Apply agentic AI systems to real-world domains such as robotics, finance, and smart infrastructure through case studies. — Level 3 – Apply		
Course Content:			
Module 1	Introduction to Agentic AI and Foundations of LLM Workflows	Assessment: Quiz / Conceptual Case Study etc	15[8L+7P] Sessions
Concept of Agentic AI & Multi-step LLM workflows, Building your first agentic system using OpenAI API, Setting up environments (Windows/Mac, Git, Cursor IDE, APIs), Prompt engineering for tool-based autonomy, Agent vs workflow patterns, Async Python and LLM function calling, Tool use and function execution in LLMs, Sequential and parallel processing in agentic workflows			
Module 2	Multi-Agent Workflows and LLM Orchestration	Assessment: Quiz / Conceptual Case Study etc	14[7L+7P] Sessions
Designing with CrewAI: Agents, tasks, tools, processes, crews, memory, building a web search tool; Multi-agent: Architectures, designing a Multi Agent Structure with Message Passing, Modes of Collaboration Among AI Agents, communication and collaboration, Case Studies: Robotics, Finance, Smart Cities and Energy Grids, Pydantic AI: Introduction, Creating Simple Agent, Building agent with personas, goals, and dynamic memory.			
Module 3	Graph-Based Agent Design with LangGraph	Assessment: Quiz / Conceptual Case Study etc	14[7L+7P] Sessions
LangGraph architecture and components, State machines: Nodes, edges, reducers, Workflow graphs and supersteps, Checkpointing and persistent memory, Web automation with Playwright + LangGraph, Structured outputs and feedback loops, LangChain tool integration			
Module 4	AutoGen and Model Context Protocol (MCP)	Assessment: Quiz / Conceptual Case Study etc	10[5L+5P] Sessions
AutoGen : agents, messaging, single agent, multiple agent. MCP: Introduction, Architecture, context			

fragmentation, clients, servers and protocol.			
Module 5	Real-Time Applications	Assessment: Quiz / Conceptual Case Study etc	7[4L+3P] Sessions
Real-time agent communication, integration of multiple agents, deployment of agentic systems, scalability and performance considerations, real-world applications: robotics, finance, smart infrastructure.			

List of Lab Tasks:

Lab 1: Setting Up the Agentic AI Development Environment

Objective: Install and configure required tools for agentic system development.

Task: Set up Python, Git, OpenAI API key, Cursor IDE, and environment variables.

Activity: Perform a guided installation and validate API access with a basic GPT query.

Lab 2: First Agentic System using OpenAI API

Objective: Understand LLM workflows through simple agent design.

Task: Build a basic Python agent that takes user input and responds using GPT-4.

Activity: Implement asynchronous function calling and test multiple prompts.

Lab 3: Prompt Engineering for Tool Use

Objective: Design effective prompts for LLMs to interact with tools.

Task: Create prompt templates for a calculator, summarizer, and code explainer tools.

Activity: Compare output quality with different prompting strategies.

Lab 4: Agent vs Workflow Pattern

Objective: Distinguish between agent-based and workflow-based models.

Task: Implement both an agent and a predefined workflow for a document summarizer.

Activity: Analyze performance and flexibility trade-offs.

Lab 5: Building a Multi-Agent CrewAI System

Objective: Learn to coordinate multiple agents in a single task.

Task: Design a system with researcher, summarizer, and presenter agents.

Activity: Use CrewAI to assign roles, tools, and memory, then simulate a research task.

Lab 6: Case Study – Multi-Agent System for Web Search

Objective: Apply multi-agent collaboration to a real-world problem.

Task: Implement a system that searches the web and compiles a report.

Activity: Assign different sub-tasks to agents and use memory for tracking.

Lab 7: Building an Agent with Persona and Goals using Pydantic AI

Objective: Customize agent behavior using metadata.

Task: Create an agent with a defined personality, goals, and memory.

Activity: Use Pydantic schemas to enforce structure and simulate goal-seeking behavior.

Lab 8: LangGraph Basics – Nodes and Edges

Objective: Design workflows as state machines.

Task: Build a LangGraph with nodes for input, processing, and output.

Activity: Visualize the graph and simulate input-output flows.

Lab 9: Checkpointing and Feedback Loops in LangGraph

Objective: Implement persistent memory and iterative feedback.

Task: Add checkpoint nodes and feedback validation in LangGraph.

Activity: Re-run tasks based on user feedback or system failure.

Lab 10: Web Automation using LangGraph + Playwright

Objective: Enable agents to perform automated web tasks.

Task: Build an agent that logs into a site and extracts content.

Activity: Integrate Playwright actions into LangGraph nodes.

Lab 11: Building AutoGen Agents with Messaging

Objective: Explore agent communication and message passing.

Task: Create single-agent and multi-agent setups using AutoGen.

Activity: Simulate collaborative planning via structured messaging.

Lab 12: Model Context Protocol (MCP) – Context Fragmentation

Objective: Handle large context and multi-client coordination.

Task: Build an agent system using MCP to divide and manage context fragments.

Activity: Simulate a client-server model and trace message flows.

Targeted Applications & Tools:

Tools/Software/Platform	**Application**
Python, NetworkX, A* algorithm, PyGame/Matplotlib	Create a Smart Navigation Agent for Indoor Mapping
OpenAI Gym, NumPy, Python, Value Iteration, Policy Iteration	Simulate Decision-Making in a Stochastic Grid Environment (Warehouse Bot)
NetLogo, Python (Mesa), JSON messaging	Design a Multi-Agent Foraging Simulation with Local Coordination
Python, Game Theory libraries (e.g., Axelrod), Flask, or CLI interfaces	Build a Negotiation Framework Between Two Resource-Sharing Agents
Raspberry Pi / Arduino, MQTT protocol, Python, Sensors (DHT11, PIR)	Develop a Reflex Agent to Control an IoT-based Thermostat System
LangChain, OpenAI API, Python, FastAPI/Flask	Implement an LLM-Powered Virtual Assistant with Tool Usage Capability
AirSim (by Microsoft), Unreal Engine, Python API, ROS (Robot Operating System)	Create an Autonomous Drone Simulation Using Agentic Principles
Mosquitto MQTT Broker, Python (Paho MQTT client), Docker (for scaling multiple agents)	Real-Time Agent Communication over MQTT for Fleet Management
Gymnasium (formerly OpenAI Gym), Stable Baselines3, Python	Build a Reinforcement Learning Agent for Elevator Control Simulation
NetLogo, Python (Swarm intelligence libraries), Jupyter Notebooks	Simulate Agent Consensus in a Disaster Response Coordination Task (Swarm AI)

Skill-Building Activities through Experiential Learning:

- Real-Time Log Analysis Using HDFS and MapReduce

This activity simulates a real-world scenario where large volumes of log files from web servers or applications need to be stored, processed and analyzed for insights. Logs will be collected continuously using Apache Flume, stored in HDFS and processed with MapReduce to identify access trends, error patterns and peak usage times.

- Big Data Pipeline Simulation with Hive, Sqoop, and MySQL
  This activity focuses on integrating traditional relational databases with Hadoop for scalable data processing. Participants simulate a typical enterprise workflow by extracting structured data from MySQL using Sqoop, storing it in HDFS, and running queries using Hive for reporting and analysis.
- Social Media Sentiment Analysis with Hadoop Ecosystem
  Unstructured text data from platforms like Twitter or Reddit provides rich insights when processed at scale. This activity involves collecting social media data using APIs, storing it on HDFS and analyzing sentiment patterns using Hive and Pig scripting.

Text Book

T1: Building Agentic AI Systems, Anjanava Biswas Wrick Talukdar, 2025 Packt Publishing

References

R1: Michael Wooldridge, An Introduction to MultiAgent Systems, 2nd Ed

R2: AI research papers from IJCAI, AAAI, AAMAS

R3: OpenAI research on agentic models and autonomous systems

R4: Case studies on autonomous robotics and virtual agent behavior

E-Resources/Web Resources

JOURNALS/MAGAZINES

Autonomous Agents and Multi-Agent Systems (Springer)

Journal of Artificial Intelligence Research (JAIR)

Artificial Intelligence Journal (Elsevier)

IEEE Transactions on Cognitive and Developmental Systems

SWAYAM/NPTEL/MOOCs:

NPTEL: Artificial Intelligence – Search Methods for Problem Solving

Coursera: Autonomous Agents – University of Alberta

edX: Multi-Agent Systems and Distributed AI

OpenAI Blog: Research articles on emerging agentic models (e.g., AutoGPT)

Course Code: CAI3408	Course Name: Deep Neural Networks Type of Course: Theory Integrated Lab	L- T-P- C	2-0-2-3
Version No.	1.0		
NCrF Level-	5.5		
Course Pre-requisites	CAI2502		
Co-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course explores foundational and advanced concepts in deep neural networks, emphasizing modern architectures and applications. It covers feedforward networks, CNNs, RNNs, generative models, and transfer learning. Through hands-on implementation and responsible AI practices, the course enables the design, training, and deployment of scalable, ethical deep learning models across diverse domains.		
Course Objective	This course is designed to improve the learners EMPLOYABILITY SKILLS by using EXPERIENTIAL LEARNING techniques.		
Course Outcomes	**On successful completion of the course the students shall be able to:** CO1: Identify the basic structure and functioning of neural networks, including types of neurons and activation functions — Level 2 – Understand CO2: Describe the role of unsupervised learning methods such as autoencoders and basic generative models — Level 2 – Understand CO3: Apply available frameworks to implement simple convolutional neural networks for image classification. — Level 3 – Apply CO4: Apply basic recurrent models such as RNN and LSTM to solve sequential data tasks. — Level 3 – Apply CO5: Use pre-trained models through transfer learning for small-scale classification or prediction problems — Level 3 – Apply		
Course Content:			
Module 1	**Foundations of Neural Networks**	Assessment : Quiz / Conceptual Case Study etc	6L+6P Sessions
Topics: Artificial vs. Biological Neuron Models, Activation Functions (ReLU, Leaky ReLU, Softmax, Swish), Feedforward Neural Networks – Single and Multi-Layer, Loss Functions: Cross-Entropy, MSE, Backpropagation and Gradient Descent, Optimizers: SGD, Adam, RMSProp, Regularization: L1, L2, Dropout, Batch Normalization, Overview of Keras and TensorFlow APIs.			

Module 2	**Convolutional Neural Networks (CNNs)**	Assessment : Quiz / Conceptual Case Study etc	6L+6P Sessions
Topics: Convolution Operation – Filters, Padding, Strides, Pooling Layers – Max, Average, CNN Architectures – VGG, ResNet, MobileNet, Data Augmentation Techniques, Transfer Learning – Feature Extraction, Fine-Tuning, Use Cases – Object Recognition, Image Classification, Visualization – Feature Maps and Filters			
Module 3	**Recurrent Neural Networks and Sequential Modeling**	Assessment : Quiz / Conceptual Case Study etc	6L+6P Sessions
Topics: Introduction to RNNs and Sequence Modeling, Limitations of Vanilla RNNs, Long Short-Term Memory (LSTM), Gated Recurrent Units (GRUs), Bidirectional RNNs, Sequence-to-Sequence (Seq2Seq) Models, Use Cases: Time Series Forecasting, Sentiment Analysis, Intro to Attention Mechanisms and Transformer Overview			
Module 4	**Generative Deep Learning Models**	Assessment : Quiz / Conceptual Case Study etc	6L+6P Sessions
Topics: Autoencoders – Basic and Variational, Denoising Autoencoders, Generative Adversarial Networks (GANs), Deep Convolutional GAN (DCGAN), Conditional GANs, Applications – Image Generation, Anomaly Detection, Style Transfer, Metrics – Inception Score, FID.			
Module 5	**Advanced Topics and Responsible AI**	Assessment : Quiz / Conceptual Case Study etc	6L+6P Sessions
Topics: Transfer Learning – BERT, ViT, and Hugging Face Models, Fine-tuning Pretrained Models, Explainability – SHAP, Grad-CAM, Fairness, Ethics, and Bias in Deep Learning, Model Deployment – TensorFlow Lite, ONNX, Flask APIs, Introduction to Large Language Models (LLMs), Use Cases – Text Generation, Visual Question Answering.			

List of Laboratory tasks:

Lab 1: Working with Deep Learning Frameworks

- **Objective:** Explore and compare popular deep learning frameworks.
- **Tasks:** Install and configure TensorFlow, Keras, and PyTorch. Analyse API differences and use cases.
- **Activity:** Implement a basic MNIST classifier in two frameworks and compare performance metrics.

Lab 2: Building a Basic Artificial Neural Network

- **Objective:** Construct and evaluate a simple ANN model.
- **Tasks:** Design input, hidden, and output layers using Keras. Apply suitable activation functions and optimizers.
- **Activity:** Train a binary classification model on the Pima Indians Diabetes dataset and evaluate with confusion matrix and ROC curve.

Lab 3 & Lab 4: Multi-Layer Feedforward Network for Regression

- **Objective:** Develop a regression model using a deep feedforward network.
- **Tasks:** Pre-process the Boston Housing or California Housing dataset, apply regularization, and tune learning rate.
- **Activity:** Implement, train, and validate the model. Visualize predicted vs actual values and analyse residual errors.

Lab 5: Building Auto encoder Models

- **Objective:** Create an unsupervised learning model for dimensionality reduction and denoising.
- **Tasks:** Design symmetric encoder-decoder architecture with bottleneck layers.
- **Activity:** Train on the Fashion-MNIST dataset, visualize reconstructed images, and analyse learned features.

Lab 6: Generative Adversarial Networks (GANs)

- **Objective:** Implement a GAN to generate synthetic image data.
- **Tasks:** Construct generator and discriminator networks. Handle mode collapse and overfitting issues.
- **Activity:** Train the GAN on CelebA or MNIST and generate new images. Evaluate training progression using loss curves and sample outputs.

Lab 7: Convolutional Neural Network (CNN)

- **Objective:** Design and deploy a CNN for image classification.
- **Tasks:** Build layered CNN architecture with dropout, pooling, and data augmentation.
- **Activity:** Train on Dog vs Cat dataset. Plot training/validation accuracy and confusion matrix.

Lab 8 & Lab 9: Recurrent Neural Networks for Time-Series and Text

- **Objective:** Build RNN and LSTM models for sequence learning.
- **Tasks:** Pre-process sequential data (IMDB or stock prices), tokenize inputs, and structure batches.
- **Activity:** Implement sentiment analysis using LSTM/GRU. Analyse accuracy and visualize attention weights (if applicable).

Lab 10: Gated Recurrent Unit (GRU) Model

- **Objective:** Compare GRU with LSTM for sequence modelling.
- **Tasks:** Train a GRU model on language translation or text generation task.
- **Activity:** Evaluate BLEU scores or prediction accuracy. Compare training time and performance with LSTM.

Lab 11 & Lab 12: Transfer Learning and Sequence-to-Sequence Models

- **Objective:** Use pre-trained models for classification and language translation.
- **Tasks:** Leverage Hugging Face Transformers or Tensor Flow Hub models for downstream tasks.
- **Activity:** Fine-tune BERT or DistilBERT for sentiment classification and implement a basic Seq2Seq model for translation using encoder-decoder architecture.

Targeted Applications & Tools:

Tools/Software/Platform	Application
TensorFlow, Keras, PyTorch, OpenCV, Scikit-learn	Image Classification, Object Detection
TensorFlow, Keras, Hugging Face Transformers, NLTK, SpaCy, BERT/GPT models	Text Classification, Sentiment Analysis, Translation
TensorFlow, Keras, Pandas, Matplotlib, LSTM/GRU layers	Time-Series Forecasting and Sequence Modeling
TensorFlow, PyTorch, GANs, Autoencoders, Hugging Face, OpenAI API	Generative Modeling (Image/Text Generation)
Scikit-learn, TensorFlow, Keras, Pandas, Seaborn	Regression and Prediction Tasks
Autoencoders, t-SNE, PCA using Scikit-learn, TensorFlow	Feature Extraction and Dimensionality Reduction
TensorBoard, Flask, Streamlit, ONNX, OpenCV, Matplotlib, Plotly	Model Deployment and Visualization
Hugging Face Transformers, TensorFlow Hub, PyTorch Hub, Keras Applications	Transfer Learning and Fine-Tuning Pre-trained Models

Skill-Building Activities through Experiential Learning:

- **Mini Project: End-to-End Deep Learning Application**
  Design a simple deep learning application, such as handwritten digit recognition (MNIST), emotion detection from text (IMDB), or object classification (CIFAR-10). Perform data preprocessing, model design using CNN or RNN, training, and basic evaluation. Present the findings, model performance metrics, and reflections on challenges faced. Tools: TensorFlow/Keras, Matplotlib.
- **Model Performance Analysis Activity**
  Implement two different deep learning models (e.g., ANN vs CNN or CNN vs Transfer Learning) on the same dataset. Analyze differences in training time, accuracy, loss curves, and generalization. A short comparative report with graphs and interpretations will be submitted. Tools: TensorFlow, Pandas, Seaborn, Scikit-learn.
- **Interactive Deployment and Explainability Task**
  Deploy a trained image or text classification model using Streamlit or Flask. Integrate basic explainability techniques such as Grad-CAM (for CNNs) or attention visualization (for RNNs) to interpret model predictions. The deployed model will be presented in a live demo session with peer feedback. Tools: TensorFlow, Streamlit, OpenCV, LIME or SHAP (basic use).

Text Book

T1: Goodfellow, I., Bengio, Y., Courville, A. (2023). Deep Learning. Germany: Alanna Maldonado.

T2: Chollet, F. (2018). Deep Learning with Python. United Kingdom: Manning

References

R1: Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. Japan: O'Reilly Media.
R2: Patterson, J., Gibson, A. (2017). Deep Learning: A Practitioner's Approach. Taiwan: O'Reilly Media.
R3: Aggarwal, C. C. (2023). Neural Networks and Deep Learning: A Textbook. Switzerland: Springer International Publishing
R4: Moolayil, J. (2019). Learn Keras for Deep Neural Networks: A Fast-track Approach to Modern Deep Learning with Python. United States: Distributed to the Book trade worldwide by Springer Science+Business Media New York
R5: Silaparasetty, V. (2020). Deep Learning Projects Using TensorFlow 2: Neural Network Development with Python and Keras. United States: Apress
R6: Chollet, F. (2018). Deep Learning with Python. United Kingdom: Manning
R7: S Rajasekaran, G A Vijayalakshmi Pai, "Neural Networks, FuzzyLogic and Genetic Algorithm, Synthesis and Applications", PHI Learning, 2017
R8: Pattanayak, S. (2017). Pro Deep Learning with TensorFlow: A Mathematical Approach to Advanced Artificial Intelligence in Python. United States: Apress
R9: Freeman, J. A., Skapura, D. M. (2003). Neural Networks: Algorithms, Applications, and Programming Techniques. India: Pearson Education

E-Resources/Web Resources

1. IEEE Transactions on Neural Networks and Learning Systems
    a. https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=5962385
2. IEEE Transactions on Pattern Analysis and Machine Intelligence
    a. https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=34http://ijaerd.com/papers/special_papers/IT032.pdf
3. International Journal of Intelligent Systems https://onlinelibrary.wiley.com/journal/1098111x
4. Swayam Nptel – Deep Learning – IIT Ropar https://onlinecourses.nptel.ac.in/noc21_cs35/preview
5. Coursera – Neural Networks and Deep Learning Andrew Ng
6. Coursera - Neural Networks for Machine Learning by Geoffrey Hinton in Coursera

Course Code: CAI3409	Course Title: Speech Recognition and Synthesis Type of Course: Theory Integrated Lab	L- T-P- C	2-0-2-3
Version No.	1.0		
Course Pre-requisites	CSS2264		
Anti-requisites	NIL		
Course Description	This course introduces fundamental principles and practical aspects of speech signal processing. It covers techniques in speech analysis, recognition, and synthesis, enabling students to build interactive voice-based AI systems. Emphasis is placed on acoustic modeling, feature extraction, and the use of machine learning and deep learning models for speech applications.		

Course Objective	To understand the basic concepts and characteristics of speech signals. To explore feature extraction and pattern matching techniques used in ASR. To study the principles of speech synthesis and TTS systems. To provide hands-on experience with speech processing tools and APIs.		
Course Outcomes	On successful completion of this course, the students shall be able to: CO1: Analyze speech signals and extract relevant features such as pitch, formants, and MFCC. (L4 – Analyze) CO2: Apply feature extraction and pattern matching techniques for speech processing tasks. (L3 – Apply) CO3: Apply machine learning and deep learning methods for automatic speech recognition. (L3 – Apply) CO4: Design speech synthesis systems using classical and neural network-based approaches. (L4 – Analyze) CO5: Develop real-time speech-based applications using APIs and open-source tools. (L3 – Apply)		
Course Content:			
Module 1	Speech Signal Fundamentals	Assessment: Assignment	16[8L+8P] Sessions
Topics: Human speech production and auditory perception, Speech signal representation: time and frequency domains, Preprocessing: sampling, quantization, windowing, pre-emphasis, Speech signal features: pitch, formants, energy, ZCR, Spectrogram and Short-Time Fourier Transform (STFT).			
Module 2	Feature Extraction &Modeling Techniques	Assessment: Assignment	12[6L+6P] Sessions
Topics: MFCC, PLP, LPC, Delta features, DTW (Dynamic Time Warping), GMM-HMM based acoustic modelling, Viterbi decoding and alignment, Basics of phonetics and phoneme modelling			
Module 3	Automatic Speech Recognition (ASR)	Assessment: Assignment	12[6L+6P] Sessions
Topics: Architecture of ASR systems, Language modeling: N-grams, smoothing, Deep learning for speech: DNN, CNN, LSTM, End-to-end models: CTC, Attention, Transformers, Tools: Kaldi, CMU Sphinx, DeepSpeech.			
Module 4	Speech Synthesis and TTS	Assessment: Assignment	12[6L+6P]

			Sessions
Topics: Concatenative, Parametric, and Neural synthesis, TTS pipeline: text normalization, phoneme mapping, HMM-based synthesis, Tacotron and WaveNet architectures, APIs: Google TTS, Amazon Polly, Festival, eSpeak.			
Module 5	Applications & Real-Time Systems	Assessment: Assignment	8[4L+4P] Sessions
Real-time speech processing systems, streaming vs batch processing, latency and noise challenges, integration of ASR and TTS systems, end-to-end speech pipeline, speech recognition APIs (Google Speech-to-Text, Amazon Transcribe, Azure Speech).			
List of Lab Tasks: Lab 1: Introduction to Speech Signal Processing Objective: Understand the nature of speech signals. Task: Record and visualize a speech waveform using a microphone and Python. Activity: Capture audio input, analyze waveform, and visualize with matplotlib. Lab 2: Time Domain Analysis of Speech Signals Objective: Analyze speech features in the time domain. Task: Compute Zero Crossing Rate (ZCR) and short-time energy of speech samples. Activity: Implement ZCR and energy plots for different speech segments using Python. Lab 3: Frequency Domain Analysis Using Spectrogram Objective: Understand frequency components of speech. Task: Apply Short-Time Fourier Transform (STFT) to speech signals. Activity: Generate and interpret spectrograms using Librosa and Matplotlib. Lab 4: Extraction of MFCC Features Objective: Extract key speech features using MFCC. Task: Extract and visualize MFCC features from recorded speech.			

Activity: Use Librosa or python_speech_features to extract MFCCs.

Lab 5: Linear Predictive Coding (LPC)

Objective: Implement LPC-based feature extraction.

Task: Apply LPC to model the vocal tract.

Activity: Compute LPC coefficients and analyze spectral envelope.

Lab 6: Implement Dynamic Time Warping (DTW)

Objective: Understand time alignment of speech.

Task: Compare two speech signals using DTW.

Activity: Use fastdtw to align utterances of the same word.

Lab 7: Acoustic Modeling using GMM-HMM

Objective: Implement basic acoustic modeling.

Task: Train a GMM-HMM model for a simple digit recognition task.

Activity: Use hmmlearn or HTK for acoustic modeling.

Lab 8: Viterbi Algorithm for Speech Alignment

Objective: Apply sequence decoding in speech.

Task: Use Viterbi decoding to find the best state path in an HMM.

Activity: Implement Viterbi algorithm in Python and visualize state paths.

Lab 9: Build a Simple ASR System using CMU Sphinx

Objective: Use open-source ASR toolkit.

Task: Set up CMU Sphinx and perform isolated word recognition.

Activity: Create language and acoustic models and test recognition.

Lab 10: Real-time Speech Recognition using DeepSpeech

Objective: Apply deep learning in ASR.

Task: Use Mozilla DeepSpeech to transcribe audio files.

Activity: Install DeepSpeech, load pre-trained model, and test transcription.

Lab 11: Evaluate ASR using Google Speech-to-Text API

Objective: Explore cloud-based speech recognition.

Task: Use Google Cloud API to transcribe different accents.

Activity: Send audio to API and analyze word error rate.

Lab 12: Introduction to Text-to-Speech (TTS) using Festival

Objective: Create basic speech synthesis output.

Task: Convert written text to speech using Festival.

Activity: Install Festival, input text, and generate synthesized speech.

Lab 13: Implement a Neural TTS system using Tacotron 2

Objective: Use deep learning for speech synthesis.

Task: Synthesize speech from text using Tacotron 2.

Activity: Use pre-trained Tacotron 2 model and generate speech samples.

Lab 14: Speech Synthesis using Google TTS and Amazon Polly

Objective: Use cloud-based TTS systems.

Task: Convert sample sentences into speech using APIs.

Activity: Use Python SDKs to access TTS services and compare outputs.

Lab 15: Mini Project – Build a Voice Assistant

Objective: Integrate ASR and TTS.

Task: Build a simple voice-controlled assistant.

Activity: Combine speech recognition, intent detection, and speech synthesis using tools like SpeechRecognition + gTTS + Pyttsx3.

TEXTBOOKS

Lawrence Rabiner and Ronald Schafer, Theory and Applications of Digital Speech Processing, Pearson, 2011.

Daniel Jurafsky and James H. Martin, Speech and Language Processing, Pearson, 3rd Edition (Draft).

https://web.stanford.edu/~jurafsky/slp3/

REFERENCE BOOKS

Ben Gold and Nelson Morgan, Speech and Audio Signal Processing, Wiley, 2nd Edition, 2011.

Thomas Dutoit, An Introduction to Text-to-Speech Synthesis, Springer, 1997.

Xuedong Huang, Alex Acero, Hsiao-Wuen Hon, Spoken Language Processing, Prentice Hall, 2001.

Tokuda et al., Speech Synthesis: Algorithms for Text-to-Speech and Speech Modification, Springer, 2021.

JOURNALS / MAGAZINES

IEEE Transactions on Audio, Speech and Language Processing
https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=10376

Computer Speech & Language (Elsevier)
https://www.sciencedirect.com/journal/computer-speech-and-language

Speech Communication (Elsevier)
https://www.journals.elsevier.com/speech-communication

SWAYAM / NPTEL / MOOCs

NPTEL – Spoken Language Processing (IIT Madras)
https://nptel.ac.in/courses/106106168

Coursera – Audio Signal Processing for Music Applications (UPF)
https://www.coursera.org/learn/audio-signal-processing

edX – Speech Processing (KTH Royal Institute of Technology)
https://www.edx.org/course/speech-processing

Fast.ai – Practical Deep Learning (TTS + ASR projects)
https://course.fast.ai/

Course Code: CAI3410	Course Name: AI Chatbots without Programming Type of Course:Theory Integrated Lab	L- T-P- C	2-0-2-3
Version No.	1.0		
NCrF Level-	5.5		
Course Pre-requisites	CAI3410		

Co-requisites	Nil		
Anti-requisites	Nil		
Course Description	This course introduces the fundamentals of conversational AI and empowers individuals to create intelligent chatbots using no-code platforms. It covers chatbot design, natural language understanding, and deployment techniques, enabling the development of domain-specific conversational agents for various applications, including customer support, education, and e-commerce, without the need for programming.		
Course Objective	This course aims to democratize chatbot development by empowering students to create intelligent conversational agents without programming knowledge, using experiential learning on visual platforms.		
Course Outcomes	On successful completion of this course, the students shall be able to: CO1: Explain the components and architecture of AI-powered chatbots. (L2 – Understand) CO2: Demonstrate the use of visual tools to design effective conversational flows. (L3 – Apply) CO3: Identify user intents and entities using no-code NLP platforms. (L2 – Understand) CO4: Develop functional chatbots tailored to specific domains and tasks. (L3 – Apply) CO5: Assess chatbot performance using user interactions and feedback data. (L3 – Apply)		
Course Content:			
Module 1	Introduction to Conversational AI and Chatbots	Assessment : Quiz / Conceptual Case Study etc	10Sessions
Evolution and history of chatbots , Types of chatbots: rule-based vs. AI-driven ,Components of chatbot architecture: NLP engine, dialogue manager, integrations ,Role of conversational AI in customer service, education, and e-commerce ,Overview of chatbot ecosystems and visual development tools			
Module 2	Designing Conversational Experiences	Assessment : Quiz / Conceptual Case Study etc	10 Sessions
Fundamentals of conversation design ,Mapping intents, user personas, and user journeys ,Structuring flows using decision trees and flowcharts Designing for multilingual and accessible interactions ,Prototyping and			

user testing of chatbot conversations			
Module 3	Building AI Chatbots with No-Code Platforms	Assessment : Quiz / Conceptual Case Study etc	10 Sessions
Overview of no-code tools: Dialogflow, Chatfuel, Microsoft PVA, Landbot ,NLP essentials: intents, entities, training phrases , Managing contexts and session handling ,Integration with messaging platforms: WhatsApp, Telegram, and websites , Customization and feature extensions with built-in plugins.			
Module 4	Testing, Integration & Debugging	Assessment : Quiz / Conceptual Case Study etc	8 Sessions
Strategies for testing chatbots, debugging techniques and error handling, validation of chatbot responses, connecting chatbots to APIs and backend databases, integration with external services, ensuring reliability and performance.			
Module 5	Deployment, Analytics & Optimization	Assessment : Quiz / Conceptual Case Study etc	7 Sessions
Deployment of chatbots on websites and social media platforms, hosting and scaling considerations, gathering user interaction data and feedback, analyzing performance using usage metrics, improving chatbot conversation flow, continuous monitoring and optimization.			
Targeted Applications & Tools:			

Tools/Software/Platform	Application
Chatfuel, Landbot, Microsoft PVA	Rapid prototyping and deployment of rule-based and AI chatbots using no-code visual platforms
Dialogflow (Google)	Intent recognition, entity extraction, context management, and AI-driven conversational flows.
Twilio API	WhatsApp and SMS integration for deploying conversational AI on messaging platforms.
Google Sheets, Airtable	Backend data storage, capturing chatbot interactions and user inputs in real time.
Microsoft Teams, Web Integration (iFrame/Script)	Deployment of chatbots on websites and collaboration platforms
Google Translate API, Built-in Language Support in Dialogflow	Designing and delivering multilingual chatbot experiences.
Built-in Analytics (Dialogflow, Chatfuel), Google Analytics Integration	User interaction tracking, performance evaluation, and chatbot improvement.

Skill-Building Activities through Experiential Learning:

Conversational Flow Design Sprint

Participants work in small groups to map user journeys and design chatbot conversational flows using visual tools like Landbot or Microsoft PVA. The focus is on structuring engaging, accessible, and multilingual interactions. Groups present their chatbot prototypes for peer feedback and iteration.

Intent and Entity Identification Challenge

Participants explore Dialogflow's no-code interface to define user intents and extract relevant entities for a chosen domain, such as e-commerce or education. The activity emphasizes understanding user queries, managing context, and improving chatbot accuracy through practical experimentation and testing.

Live Chatbot Deployment and Feedback Loop

Learners deploy their chatbot on a website or messaging platform using iFrame or Twilio integration. Real-time user testing is conducted, and participants collect interaction logs to evaluate chatbot performance, identify improvement areas, and iterate conversational designs based on user feedback.

Text Book

1. Jason D. Brown, Designing Bots: Creating Conversational Experiences, O'Reilly Media, 2017.
2. Akshay Kulkarni and Adarsha Shivananda, Building Chatbots with Google Dialogflow, Apress, 2019.

References

1.Rashid Khan, Build Better Chatbots: A Complete Guide to Getting Started with Chatbots, Apress, 2017.

2.Michael McTear, Conversational AI: Dialogue Systems, Conversational Agents, and Chatbots, Springer, 2020.

3.Navin Sabharwal et al., Designing Human-Centric AI Experiences, Apress, 2021.

E-Resources/Web Resources

Coursera – Building AI Powered Chatbots Without Programming (IBM)

edX – Introduction to Chatbots (Microsoft)

Udemy – Chatbot Development Without Coding

Course Code: CAI3411	Course Name: Generative AI Type of Course: Theory Integrated Lab	L- T-P- C	2-0-2-3
Version No.	1.0		
NCrF Level-	5.0		
Course Pre-requisites	CSS2264		
Co-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course builds the foundational insight of understanding generative AI models and to explore various architectures, algorithms and practices of Gen AI skills to accelerate strategic decision making with data and deliver cutting-edge products faster with GenAI-augmented software development and leverage Gen AI tools to optimize workflows.		
Course Objective	The objective of the course is to familiarize the learners to explore the competence in benchmarking and comprehend the potential generative AI models and techniques to revolutionize industries and create prominent Gen AI tools to attain Employability Skills through Experiential Learning techniques.		
Course Outcomes	**On successful completion of the course the students shall be able to:** CO1: Explain the fundamentals of Generative AI models, LLMs, and prompt engineering techniques for customized output generation. [Level 2 – Understand] CO2: Apply transformer architectures, attention mechanisms, embeddings, and text-based generative models for NLP tasks. [Level 3 – Apply] CO3: Practice advanced generative AI techniques using the LangChain Python framework. [Level 3 – Apply] CO4: Solve real-time applications using multi-modal generative AI models. [Level 3 – Apply] CO5: Analyze the performance, limitations, and ethical considerations of generative AI models in real-world applications. [Level 4 – Analyze]		
Course Content:			
Module 1	Introduction to Generative AI	Assessment: Module Test	L-6 P-4 10 sessions
Introduction to Generative models: Historical perspective and evolution, Applications, Types of Generative models for different data modalities, Large Language Models (LLMs) – Introduction, evolution, Generative pre-trained transformers (GPT) and its variants, Google DeepMind's, PaLM2, LLaMa and its series of models by Meta AI, Claude and its variants by Anthropic, Prompt Engineering-basic prompting.			
Module 2	Text based generative models	Assessment: Module Test	L-6 P-6 12 Sessions
Text-based Generative models: State-of-the Art models, RNN, LSTM, Transformer Architecture, Transformer based Generative models: BERT, GPT, Training and Fine tuning LLMs for Generative task, Open AI's Pre-trained transformers for Text Generation: ChatGPT, Limitations of LLMs: Lack of context and Hallucination risks, Techniques to mitigate these limitations: chaining and retrieval augmentation, Workflow of an LLM application.			
Module 3	Introduction to LangChain framework	Assessment: Quiz	L-6 P-8 14 Sessions

Introduction to Lang chain: Types, Components, Information retrieval using agents and tools in Lang chain, Retrieval Augmented Language Models (RaLM): Understanding Retrieval and vectors: Embeddings, Vector storage, Vector indexing, Vector Libraries, Vector Databases, Chatbot using memory and conversation buffer.			
Module 4	Generative Models for other data modalities	Assessment: Module Test	L-8 P-8 16 Sessions
Generative Adversarial Networks (GAN): GAN Architecture, GAN variants, Neural Style transfer with GAN, Training GANs and common challenges, GAN applications in image and text generation, Variational Auto Encoders (VAEs) and its variants, Image generation models: Dall-E, MidJourney and stable diffusion: Architecture and components of stable diffusion, Text-to-image Generation, Parameter tuning, Image-to-image generation, Training custom models, In-Painting: Exchanging classes, Multi-modal generative models using Whisper for Audio: Speech-to-Text generation.			
Module 5	Ethical AI, Evaluation, and Real-World Applications	Assessment: Module Test	L-4 P-4 8 Sessions
Performance evaluation metrics for Generative AI, Bias and fairness in AI systems, Ethical considerations in Generative AI, Privacy and security issues in LLMs, Responsible AI practices, Explainability and transparency in AI, Real-world case studies of Generative AI, Future trends in Generative AI			
List of laboratory tasks:			
Experiment No.1: Setting up Python IDE(Spyder) and OpenAI API key. Introduction to OpenAI playground and prompting Level 1: Document the installation and the process for generating models in OpenAI Level 2: Solve various GenAI models of OpenAI from Playground using prompts Experiment No.2: Text classification, summarization, sentiment analysis, chatbot application, code explanation with generating single and multiple response(S). Level 1: Practice the text generation model of OpenAI and Spyder IDE to implement various applications. Experiment No.3: Embeddings – for words, similarity between words, text embeddings, plagiarism check of documents Level 1: Use generating embeddings for words, text and documents Level 2: Apply the embeddings API to develop applications for plagiarism check Experiment No.4: Image generation using Dall E. Using GPT-Vision model for text to image generation and image-to-text. Level 1: Apply GPT-vision model for text-to-image generation and image-to-image Experiment No.5: Transformer based text and email classification Level 1: Develop transformer-based AI models for classifying text/email Experiment No.6: BERT for masked token generation Level 1: Develop BERT based model for generating masked tokens Experiment No.7: Creating applications using different types of LangChains – Simple Sequential, Sequential and map reduce Level 1: List the various types of chains in Langchain Level 2: Practice different types of chains using Spyder IDE and OpenAI			

Experiment No.8: Information retrieval using agents and tools in Langchain.
Level 1: Use agents and tools with Langchain for information retrieval

Experiment No.9: RAG with Custom Document loading and retrieval in LangChain using ChromaDB
Level 1: Understand ChromaDb
Level 2: Apply RAG with Langchain to generate information retrieval model from custom document

Experiment No.10: Using action agents, human as a tool and plan and execute agents for information retrieval.
Level 1: Understand action agents and plan and execute agents
Level 2: Use agents and tools for information retrieval

Experiment No.11: Text to Image generation using Dall-e/stable diffusion using prompts
Level 1: List various image generation models
Level 2: Use an image generation model to generate image from prompts

Experiment No.12: Speech to text and multi-modal generative models using Whisper for Audio
Level 1: Identify the generative model for text, image and audio data
Level 2: Use Langchain to create models for generating different data modalities. Ex: Audio-to-text

Experiment No.13: Fairness evaluation and bias mitigation in AI models using fairness metrics

Targeted Applications & Tools:

Tools/Software/Platform	Application
Python (Anaconda / Jupyter Notebook)	For developing the GenAI application
GPT 3.5 Turbo, Gemini 3.5 – Flash	For text generation based exercises
Dall-E-3, Stability AI	For Image generation based exercises
Whisper	For audio-to-text generation exercises
LangChain framework	For implementing simpleChain and Sequential Chain workflows

Skill-Building Activities through Experiential Learning:

- **Plagcheck using Embedding models of OpenAI/Google**
  Learners will develop a comprehensive application for identifying the source of a plagiarised text. This task uses the embedding model of OpenAI/Google.

- **Router Chain based application**
  Learners will develop an application to play the role of a math/science/history tutor based on the user choice. This task uses the router chain of Langchain and the text generation models of OpenAI/Google.

Text Book
T1:: Ben Auffarth, Generative AI with LangChain, Ist Edition , Packt. Inc. ISBN: 978-1-83508-346-8, Decemeber 2023.

T2: David Foster, Generative Deep Learning, 2nd Edition by O'Reilly Media, Inc. ISBN: 9781098134181, May 2023.
T3: James Phoenix, Mike Taylor, Prompt Engineering for Generative AI, O'Reilly Media, Inc., ISBN:9781098153373, July 2024.
T4: AI Fairness
Mahoney, T., Varshney, K. R., & Hind, M. (2020). AI Fairness: How to Measure and Reduce Unwanted Bias in Machine Learning. O'Reilly Media.

References

R1: Bandi, A., Adapa, P. V. S. R., & Kuchi, Y. E. V. P. K. (2023). The power of Generative AI: a review of requirements, models, Input–Output formats, evaluation metrics, and challenges. Future Internet, 15(8), 260. https://doi.org/10.3390/fi15080260
R2. Barachini, F., & Stary, C. (2022). From digital twins to digital selves and beyond. In Springer eBooks. https://doi.org/10.1007/978-3-030-96412-2
R3. Hadi, M. U., Tashi, Q. A., Qureshi, R., Shah, A., Muneer, A., Irfan, M., Zafar, A., Shaikh, M. B., Akhtar, N., Wu, J., & Mirjalili, R4. S. (2023). Large Language Models: A Comprehensive Survey of its Applications, Challenges, Limitations, and Future Prospects. https://doi.org/10.36227/techrxiv.23589741.v4
R4. Hai-Jew, S. (n.d.). Generative AI in Teaching and Learning. IGI Global

E-Resources/Web Resources

1. MOOC's/Swayam Courses/Online Courses: https://onlinecourses.swayam2.ac.in/imb24_mg116/preview
2. Certification Course by Google : https://www.cloudskillsboost.google
3. https://www.credly.com/badges/90e3eae0-87f3-44e3-af82-658e837aad3d/public_url
4. https://www.coursera.org/learn/generative-ai-with-llms
5. https://www.coursera.org/specializations/prompt-engineering
6. https://openai.com
7. https://python.langchain.com/v0.2/docs/introduction/
8. https://www.udemy.com/course/master-ai-image-generation-using-stable-diffusion/?kw=Image+generation+using&src=sac&couponCode=LETSLEARNNOWPP
9. https://huggingface.co/google-t5/t5-base
10. https://dominguezdaniel.medium.com/exploring-image-generative-ai-models-9359705b15d3
11. https://cloud.google.com/use-cases/retrieval-augmented-generation?hl=en#
12. https://ig.ft.com/generative-ai/
13. https://medium.com/@samia.khalid/bert-explained-a-complete-guide-with-theory-and-tutorial-3ac9ebc8fa7c

Course Code: CAI3412	Course Name: Machine Learning for Finance Type of Course: Theory Integrated Lab	L- T-P- C	2-0-2-3
Version No.	1.0		
NCrF Level-	5.5		
Course Pre-requisites	CAI2500		
Co-requisites	Nil		

Anti-requisites	Nil		
Course Description	This course explores the application of machine learning in financial analysis and decision-making. It covers financial data modeling, time-series forecasting, portfolio optimization, and algorithmic trading. Emphasis is placed on hands-on implementation using Python, enabling the development of data-driven solutions for risk assessment, fraud detection, and financial strategy design.		
Course Objective	To enable students to apply machine learning methods to financial data analysis, portfolio optimization, and risk modeling. Students will gain practical knowledge to build ML-based financial systems using Python and related libraries.		
Course Outcomes	On successful completion of this course the students will be able to: CO1: Identify different types of financial data and their relevance in machine learning applications. **(**Level 2 - Understand) CO2: Apply machine learning models to forecast market trends and evaluate financial risks. (Level 3 - Apply) CO3: Examine the impact of model performance on credit scoring and risk prediction. (Level 4 - Analyze) CO4: Discuss basic portfolio optimization strategies using financial indicators and machine learning techniques. (Level 2 - Understand) CO5: Apply algorithmic trading rules to test strategies using historical market data. (Level 3 - Apply)		
Course Content: **I**			
Module 1	Introduction to Financial Data and ML	Assessment : Assignment 1	18 hours (8L + 10P)
Basics of financial systems and instruments, Types of financial data: stock prices, returns, volumes, Introduction to time-series characteristics, Role of machine learning in finance, Overview of supervised learning models (linear regression, decision trees, SVM), Introduction to financial dataset sources (Yahoo Finance, Quandl, Kaggle)			
Module 2	Financial Forecasting and Risk Modeling	Assessment : Assignment 1	14 hours (7L + 7P)
Time-series forecasting: ARIMA, exponential smoothing, Deep learning for forecasting: LSTM basics, Risk quantification: Value at Risk (VaR), Conditional VaR, Credit scoring using classification models (Logistic Regression, Random Forest), Feature engineering for financial risk prediction			
Module 3	Portfolio Optimization and Strategy Design	Assessment : Assignment 2	14 hours (6L + 8P)
Modern Portfolio Theory and asset correlation, Efficient frontier and Sharpe Ratio, ML models for asset allocation, Basics of Reinforcement Learning in Portfolio Rebalancing, Backtesting strategies and performance metrics			
Module 4	Algorithmic Trading and Fraud Detection	Assessment :	14 hours (6L + 8P)
Introduction to algorithmic trading workflows, Strategy development using ML: momentum vs. mean reversion, Real-time data streaming and analysis with Python, Fraud detection using anomaly detection			

(Isolation Forest, Autoencoders), Ethical considerations in financial ML systems			
Module 5	Reinforcement Learning for Investment Management	Assessment :	10 hours (6L + 4P)

Reinforcement Learning for Investment Management - Merton's optimal consumption, Optimal hedging strategies, Robo-advisors for optimal allocation, Learning an investor's preferences.

List of laboratory tasks:

- Load and analyze historical stock prices using pandas and yfinance.
- Visualize financial time-series data and calculate returns.
- Implement linear regression for stock price forecasting.
- Build logistic regression model to predict credit default.
- Use Random Forests to classify high-risk vs low-risk customers.
- Implement ARIMA model for time-series forecasting.
- Use LSTM model to predict stock prices with Keras/TensorFlow.
- Apply K-Means clustering on customer transaction data.
- Calculate and visualize Value at Risk (VaR) for a portfolio.
- Optimize asset allocation using ML-based mean-variance optimization.
- Backtest a trading strategy using historical stock data.
- Implement reinforcement learning for portfolio management.
- Detect financial fraud using anomaly detection techniques.
- Build a real-time financial dashboard using Streamlit.
- Mini-project: Develop and evaluate an ML model for a selected finance use case.

Targeted Applications & Tools:

Tools/Software/Platform	Application
Python, pandas, yfinance	Load, preprocess, and visualize historical stock price and volume data.
statsmodels, ARIMA	Build time-series forecasting models for financial market trends.
scikit-learn, Random Forest, Logistic Regression	Develop credit scoring and risk classification models.
TensorFlow/Keras, LSTM	Implement deep learning models for stock price prediction.
matplotlib, seaborn	Visualize financial indicators, portfolio performance, and risk metrics.
numpy, scipy	Perform portfolio optimization calculations and asset correlation analysis.
reinforcement learning libraries (e.g., Stable-Baselines3)	Develop and evaluate portfolio rebalancing strategies.
sklearn, Isolation Forest, Autoencoders	Detect financial fraud using anomaly detection techniques.
Streamlit, Plotly	Build interactive dashboards for financial data monitoring and reporting.

Skill-Building Activities through Experiential Learning:

Stock Price Forecasting with ARIMA

Learners will apply ARIMA models on historical stock price data to forecast short-term price movements. The activity develops skills in time-series preprocessing, model selection, and performance evaluation. Visualizations of predicted vs. actual trends will help interpret model effectiveness for real-world financial forecasting applications.

Credit Risk Classification using Machine Learning

Participants will build and evaluate classification models, such as Logistic Regression and Random Forest, to predict loan default risk. The activity focuses on feature engineering, model evaluation using ROC-AUC, and practical insights for financial risk assessment, preparing learners for real-world credit scoring tasks.

Portfolio Optimization with Machine Learning

Learners will implement mean-variance optimization using Python and machine learning techniques to construct an efficient investment portfolio. They will analyze asset correlations, calculate performance metrics like Sharpe Ratio, and visualize optimal asset allocations to understand how AI-driven portfolio design improves financial decision-making.

Text Book

1. Hilpisch, Y. J. (2018). Python for Finance. United States: O'Reilly Media.
2. de Prado, M. L. (2018). Advances in Financial Machine Learning. United Kingdom: Wiley.
3. Tatsat, H., Puri, S., Lookabaugh, B. (2020). Machine Learning and Data Science Blueprints for Finance. China: O'Reilly Media.

References

- Tucker Balch, "Machine Learning for Trading", Udacity course materials.
- Aronson, D. (2011). Evidence-Based Technical Analysis: Applying the Scientific Method and Statistical Inference to Trading Signals. Germany: Wiley.
- E. Tsang, "Foundations of Computational Finance with Machine Learning", Springer, 2021

E-Resources/Web Resources

SWAYAM/NPTEL/MOOCs:

1. Machine Learning for Engineering and Science Applications

2. Artificial Intelligence: Search Methods for Problem Solving

3 Deep Learning

Course Code: CAI3431	Course Name: Fundamentals of Quantum Computing and Machine Learning Type of Course: Theory Laboratory Integrated	L- T-P- C	2-0-2-3
Version No.	1.0		
NCrF Level-			
Course Pre-	MAT2311		

<table>
<tr><td>requisites</td><td colspan="4"></td></tr>
<tr><td>Co-requisites</td><td colspan="4">NIL</td></tr>
<tr><td>Anti-requisites</td><td colspan="4">-</td></tr>
<tr><td>Course Description</td><td colspan="4">This course introduces the fundamental principles of quantum computing required for computation and information processing. It covers qubits, superposition, quantum measurement, entanglement, quantum logic gates, and quantum circuit design. The course further explores multi-qubit systems, quantum communication protocols, and foundational quantum algorithms such as Deutsch–Jozsa, Grover's, and Bernstein–Vazirani. Emphasis is placed on hands-on learning using quantum computing frameworks such as Qiskit or Cirq to design, simulate, and analyze quantum circuits.</td></tr>
<tr><td>Course Objective</td><td colspan="4">The objective of this course is to enable learners to understand the fundamental principles of quantum computing, including qubit representation, quantum states, and quantum operations. The course aims to develop the ability to construct and simulate quantum circuits using single- and multi-qubit gates, analyze multi-qubit systems and quantum entanglement, and implement quantum communication protocols. It further focuses on applying and analyzing fundamental quantum algorithms such as Deutsch–Jozsa, Grover's, and Bernstein–Vazirani through hands-on experience using quantum computing frameworks.</td></tr>
<tr><td rowspan="6">Course Outcomes</td><td colspan="4">On successful completion of this course, the students shall be able to:</td></tr>
<tr><td colspan="3">CO1: Explain the fundamental principles of quantum computing, including qubits, superposition, measurement, and representation using mathematical formalism.)</td><td>(L2 – Understand</td></tr>
<tr><td colspan="3">CO2: Apply single- and multi-qubit quantum gates to construct and simulate basic quantum circuits.</td><td>(L3 – Apply)</td></tr>
<tr><td colspan="3">CO3: Analyze multi-qubit systems and demonstrate quantum entanglement using controlled operations and measurement outcomes.</td><td>(L4 – Analyze)</td></tr>
<tr><td colspan="3">CO4: Implement quantum circuits and communication protocols such as teleportation, and evaluate circuit properties like reversibility and depth.</td><td>(L3 – Apply)</td></tr>
<tr><td colspan="3">CO5: Implement and analyze fundamental quantum algorithms such as Deutsch–Jozsa, Grover's, and Bernstein–Vazirani using simulation tools.</td><td>(L4 – Analyze)</td></tr>
<tr><td colspan="5">Course Content:</td></tr>
<tr><td>Module 1</td><td>Foundations of Quantum Computing</td><td colspan="2">Assessment: Module Test</td><td>L-6 P-6
12 sessions</td></tr>
<tr><td colspan="5">Topics:
Classical vs Quantum Computing ,Qubits, Superposition, and Measurement,Bloch Sphere Representation Quantum States and Dirac Notation ,Introduction to Quantum Gates (X, Y, Z, H)</td></tr>
</table>

Module 2	Quantum Circuits and Operations	Assessment: Module Test	L-8 P-6 14 Sessions
Topics: Single-Qubit and Multi-Qubit Gates ,Controlled Gates (CNOT, CZ, SWAP) ,Quantum Circuit Design Principles Quantum Circuit Simulation ,Reversibility in Quantum Computing			
Module 3	Quantum Entanglement and Information Theory	Assessment: Quiz	L-6 P-6 12 Sessions
Topics: Multi-Qubit Systems ,Entanglement and Bell States ,Quantum Measurement and Correlations ,Basics of Quantum Information Theory ,Applications of Quantum Information			
Module 4	Quantum Algorithms	Assessment: Module Test	L-6 P-6 12 Sessions
Deutsch and Deutsch–Jozsa Algorithm,Bernstein–Vazirani Algorithm ,Grover's Search Algorithm ,Quantum Teleportation Protocol ,Algorithm Performance and Complexity			
Module 5	Quantum Machine Learning & Applications	Assessment: Module Test	L-6 P-6 12 Sessions
Introduction to Quantum Machine Learning (QML),Quantum Regression and Classification ,Hybrid Quantum-Classical Models ,Applications:QuantumSearch ,Quantum Cryptography (BB84) ,Quantum Communication Tools & Platforms: Qiskit, Cirq, IBM Quantum			
List of laboratory tasks:			
Experiment 1: Introduction to Qiskit/Cirq and Quantum State Preparation • Level 1: Install Python/Qiskit, document setup, generate simple qubit states • Level 2: Visualize qubits on the Bloch sphere **Experiment 2: Single-Qubit Gates** • Level 1: Implement X, Y, Z, H, S, T gates on a qubit • Level 2: Measure output probabilities and visualize results **Experiment 3: Multi-Qubit States and Entanglement** • Level 1: Create Bell states • Level 2: Demonstrate entanglement through measurement correlations **Experiment 4: Controlled Gates and Multi-Qubit Circuits** • Level 1: Implement CNOT, CZ, SWAP • Level 2: Build a small circuit involving multiple operations **Experiment 5: Quantum Circuit Simulation and Reversibility** • Level 1: Construct circuits showing reversibility • Level 2: Measure circuit depth and analyzebehavior			

Experiment 6: Quantum Teleportation Protocol

- Level 1: Implement teleportation in Qiskit
- Level 2: Validate transferred qubit state

Experiment 7: Deutsch–Jozsa Algorithm

- Level 1: Implement Deutsch's algorithm
- Level 2: Extend to 2 qubits for DJ algorithm

Experiment 8: Grover's Algorithm

- Level 1: Implement Grover's search for 1 solution among 4 items
- Level 2: Visualize amplitude amplification

Experiment 9: Simulation of Vazirani Algorithm in Qiskit

- Level 1: QASM Simulator, AER
- Level 2: BV Circuit

Targeted Applications & Tools:

Tools/Software/Platform	Application
Python (Anaconda / Jupyter Notebook)	Development of quantum programs
Qiskit / Cirq	Quantum circuit design & execution
IBM Quantum Platform	Running circuits on real quantum hardware
Bloch Sphere Simulator Tools	Visualization of qubits
QuTiP	Quantum system simulation (optional)

Skill-Building Activities through Experiential Learning:

- Quantum Teleportation Demo Application
  Learners build a full teleportation protocol demonstrating quantum communication.
- Grover Search Application
  Develop an application using Grover's algorithm for small database search.
- Quantum Secure Communication Mini Project
  Implement a simple BB84 quantum cryptography simulation demonstrating secure key generation.

Text Book

T1: M. A. Nielsen & I. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press.
T2: Chris Bernhardt, *Quantum Computing for Everyone*, MIT Press.
T3: Phillip Kaye, Raymond Laflamme, Michele Mosca, *An Introduction to Quantum Computing*, Oxford University Press.

References

R1: Qiskit Textbook (Open-source)
R2: Michael Nielsen's Quantum Computing Lectures (Online)

R3: Preskill, J. *Quantum Computing in the NISQ era and beyond* **R4: IBM Quantum Documentation**
E-Resources/Web Resources 1. https://qiskit.org/textbook 2. https://quantum-computing.ibm.com 3. https://quantum.country/qcvc 4. https://learn.qiskit.org 5. Quantum MOOC – Coursera: *Quantum Mechanics for Scientists and Engineers* 6. MIT OpenCourseWare – Quantum Computation 7. https://cirq.readthedocs.io 8. https://qutip.org 9. https://arxiv.org/archive/quant-ph

Course Code: CAI3413	Course Title: Industrial IoT Type of Course: Theory Integrated Lab	L- T-P- C	2-0-2-3
Version No.	1.0		
Course Pre-requisites	CSS2264		
Anti-requisites	NIL		
Course Description	This course introduces the foundational concepts and applications of the Industrial Internet of Things (IIoT), emphasizing industrial automation, smart manufacturing, and predictive analytics. Students will gain hands-on experience with sensor integration, cloud platforms, and industrial communication protocols necessary for developing and managing IIoT systems.		
Course Objective	To provide practical skills and theoretical knowledge for designing, implementing, and managing Industrial IoT applications in real-world industrial and manufacturing environments.		
Course Outcomes	On successful completion of this course, the students shall be able to: CO1: Explain the architecture, components, and ecosystem of Industrial IoT systems. (L2 – Understand) CO2: Integrate sensors, actuators, and edge devices for real-time industrial data acquisition and monitoring. (L3 – Apply) CO3: Apply communication protocols and networking techniques for reliable data transmission in IIoT environments. (L3 – Apply)		

	CO4: Utilize cloud platforms and analytics techniques for processing and analyzing industrial data.	(L3 – Apply)
	CO5: Analyze security challenges and apply appropriate security mechanisms in Industrial IoT systems.	(L4 – Analyze)

Course Content:

Module 1	Introduction to Industrial IoT	Assessment: Assignment	18[8L+10P] Sessions

Topics:

Fundamentals of IoT and IIoT: Definitions, history, evolution, IIoT vs. Consumer IoT, IIoT Architecture Layers: Perception, Network, Middleware, Application Industrial applications: Smart factories, Energy grids, Process automation Introduction to Industry 4.0 and its pillars (Cyber-Physical Systems, Big Data, Cloud Computing), Business benefits and ROI of IIoT deployment

Challenges in IIoT: Scalability, legacy systems, interoperability, security, Standards and regulatory compliance in IIoT (ISA-95, ISO/IEC 30141).

Module 2	Devices and Communication Protocols	Assessment: Assignment	14[7L+7P] Sessions

Topics:

Types of Sensors and Actuators: Environmental, Proximity, Motion, Flow, Vibration, Microcontrollers and Development Boards: Arduino, Raspberry Pi, ESP32, IIoT Gateways and Edge Devices: Intel NUC, Jetson Nano, Communication Technologies: Wired: Modbus RTU, CAN, Ethernet/IP,Wireless: Zigbee, LoRaWAN, Wi-Fi, NB-IoT, Bluetooth LE, Protocols for IIoT:MQTT: Publish-Subscribe Model,OPC-UA: Industrial interoperability and data exchange, CoAP, HTTP/HTTPS,Integration with SCADA and PLC Systems,Device provisioning and configuration management, Network Topologies and Addressing in IIoT environments.

Module 3	IIoT Data Processing and Analytics	Assessment: Assignment	14[6L+8P] Sessions

Topics:

Data Acquisition: Sampling rate, ADC/DAC, Signal conditioning, Local Storage: SQLite, Edge databases, Stream Processing: Apache Kafka, Apache Flink, Data Preprocessing: Noise reduction, outlier detection, filtering techniques, Introduction to Edge Computing and Fog Computing, Real-time vs. Batch Analytics:Predictive Analytics in Manufacturing:Predictive maintenance models,Failure prediction using historical data,Machine Learning for Sensor Data: Regression, Clustering, Classification,Anomaly Detection in Industrial Systems,Data Encryption and Access Control at Edge and Cloud,Security Threats: DDoS, firmware tampering, man-in-the-middle attacks, Best practices for data governance and GDPR compliance

Module 4	IIoT Cloud Integration and Applications	Assessment: Assignment	12[6L+6P] Sessions

Overview of Cloud Computing Models (IaaS, PaaS, SaaS) in IIot, IIoT Cloud Platforms:

AWS IoT Core: Device Shadow, Rule Engine, Azure IoT Hub and IoT Central, Google Cloud IoT Core, Device-to-Cloud Communication Models, Storage Solutions: Amazon S3, Azure Blob Storage, InfluxDB, Visualization Tools: Grafana, Power BI, Google Data Studio.			
Module 5	Digital Twin of a robotic arm	Assessment: Assignment	12[6L+6P] Sessions
Digital Twin of a robotic arm, Smart HVAC system for energy optimization,Real-time supply chain visibility, Building Custom Dashboards and Alerts,Integration with ERP and MES Systems,Final Capstone Project:Design, simulate, and evaluate an end-to-end IIoT system, Include edge device, communication, cloud, analytics, and visualization.			
Project work/Assignment:			
Assignment 1: Sensor Data Acquisition and Visualization Assignment 2: Edge-Based Analytics for Predictive Maintenance Mini Project (Team-based): End-to-End IIoT System for a Smart Industry Scenario			
Setup Raspberry Pi/Arduino for IIoT applications. Interface DHT11, vibration, and ultrasonic sensors. Send sensor data via MQTT to a cloud broker. Create Node-RED dashboards for IIoT data. Build OPC-UA server and client communication. Log and visualize sensor data using Python. Apply edge analytics using Raspberry Pi and filtering techniques. Stream real-time data using Kafka. Integrate with AWS IoT Core for data monitoring. Forecast sensor values using LSTM (predictive maintenance). Detect anomalies in sensor data using Scikit-learn. Connect multiple IIoT devices into a secure network. Simulate a digital twin for a production line. Develop a simple IIoT-based security alert system. Capstone Project: Deploy an end-to-end IIoT prototype for a smart manufacturing scenario.			
REFERENCE MATERIALS: TEXTBOOKS Alasdair Gilchrist, Industry 4.0: The Industrial Internet of Things, Apress, 2016.			

Sabina Jeschke, Christian Brecher, Houbing Song, Danda B. Rawat, Industrial Internet of Things, Springer, 2017.

REFERENCES

Arshdeep Bahga, Vijay Madisetti, Internet of Things: A Hands-On Approach, Universities Press, 2015.

Hakima Chaouchi, The Internet of Things: Connecting Objects, Wiley, 2010.

Rajkumar Buyya, Amir Vahid Dastjerdi, Internet of Things: Principles and Paradigms, Morgan Kaufmann, 2016.

JOURNALS/MAGAZINES

IEEE Internet of Things Journal
https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=6488907
Publishes articles on the design, development, and deployment of IoT applications in industrial settings.

ACM Transactions on Internet of Things (TIOT)
https://dl.acm.org/journal/tiot
Covers architectures, algorithms, and applications related to IoT including edge computing and real-time analytics.

Sensors (MDPI - Special Issue on Industrial IoT)
https://www.mdpi.com/journal/sensors
Features studies on sensor development, wireless communication, and industrial sensor networks.

Journal of Industrial Information Integration (Elsevier)
https://www.sciencedirect.com/journal/journal-of-industrial-information-integration
Focuses on integrating data from industrial systems for smart manufacturing and digital twins.

IEEE Spectrum – IoT and Industry 4.0
https://spectrum.ieee.org/
Publishes accessible articles on cutting-edge tech including IIoT, smart robotics, and industrial AI.

Industrial IoT World – Insights & Reports
https://www.iiot-world.com/
Offers case studies, whitepapers, and expert commentary on real-world IIoT deployments.

Automation World
https://www.automationworld.com/
Covers automation systems, IIoT trends, cloud integration, and edge devices.

Industry 4.0 Magazine (Industry40.today)
https://industry40.today/
Industry-focused magazine offering updates on cyber-physical systems, smart factories, and digital transformation.

Regularly features expert opinion, trends, and technologies including AI in cyber security.

SWAYAM/NPTEL/MOOCs:

NPTEL – Industrial Internet of Things (IIT Kharagpur)

Instructor: Prof. Sudip Misra https://onlinecourses.nptel.ac.in/noc23_cs69/preview NPTEL – Introduction to Industry 4.0 and Industrial Internet of Things (IIT Roorkee) Instructor: Prof. Sudeb Dasgupta https://onlinecourses.nptel.ac.in/noc21_me88/preview

Course Code: CAI3414	Course Title: Smart Farming Type of Course: Theory Integrated Lab	L- T-P- C	2-0-2-3
Version No.	1.0		
Course Pre-requisites	CSS2264		
Anti-requisites	NIL		
Course Description	This course introduces the interdisciplinary domain of Smart Farming, integrating IoT, AI, cloud computing, and sensor networks to enable precision agriculture. Students will learn how to design, develop, and deploy data-driven agricultural systems for real-time monitoring, crop management, and decision-making.		
Course Objective	To provide practical and theoretical insights into the application of smart technologies in agriculture, with emphasis on sustainable practices, yield optimization, and automation.		
Course Outcomes	On successful completion of this course, the students shall be able to: CO1: Explain the fundamentals and ecosystem of smart agriculture systems. (L2 – Understand) CO2: Apply IoT technologies and sensor networks for environmental and crop monitoring. (L3 – Apply) CO3: Analyze agricultural data using AI and machine learning techniques for predictive insights. (L4 – Analyze) CO4: Design and deploy smart farming solutions using cloud and mobile platforms. (L3 – Apply) CO5: Apply automation techniques for efficient irrigation, crop management, and resource optimization. (L3 – Apply)		
Course Content:			
Module 1	Introduction to Smart Farming and Precision Agriculture	Assessment: Assignment	18[8L+10P] Sessions

Topics: Evolution from traditional to precision agriculture, Components and architecture of Smart Farming systems, Applications: crop monitoring, irrigation control, pest detection, Overview of remote sensing and satellite-based agriculture, Soil health and weather data integration.			
Module 2	IoT and Sensor Systems in Agriculture	Assessment: Assignment	14[7L+7P] Sessions
Topics: Sensor types: soil moisture, temperature, humidity, pH, NDVI, Microcontrollers: Arduino, ESP32, Raspberry Pi,Wireless communication protocols: LoRa, Zigbee, Wi-Fi, GSM, Energy harvesting and power management for field devices, Edge computing in agriculture.			
Module 3	Data Analytics and AI in Agriculture	Assessment: Assignment	14[6L+8P] Sessions
Topics: Introduction to agri-data lifecycle, Data preprocessing and anomaly detection, AI/ML techniques: regression, classification, clustering for yield prediction and disease detection,Time series forecasting for rainfall, irrigation, and climate impact, Decision support systems and dashboards.			
Module 4	Cloud, Mobile, and Drone Integration	Assessment: Assignment	12[6L+6P] Sessions
Cloud platforms for agriculture: ThingSpeak, AWS IoT, Blynk, Azure IoT, Mobile app integration for farmer advisory systems, Role of drones in smart agriculture: aerial imaging, spraying, crop health assessment.			
Module 5	Case studies	Assessment: Assignment	12[6L+6P] Sessions
Case studies: Smart greenhouses, hydroponics, aquaponics, Capstone: End-to-end smart farming system design.			
Project work/Assignment:			
Assignment 1: Real-Time Crop and Soil Monitoring System Assignment 2: Crop Yield Prediction using Machine Learning Mini Project (Team-based): Smart Farm Automation System			
List of Lab Tasks : Setup Arduino/ESP32 for collecting soil and climate data Interface with soil moisture, DHT11, and pH sensors Transmit data wirelessly using LoRa or Wi-Fi Real-time dashboard for field data (using Blynk/ThingSpeak)			

Predict crop yield using linear regression Train an image classifier for leaf disease detection Setup automated irrigation control system Use GPS for geotagging sensor data Drone-based simulation for crop monitoring Preprocess and visualize multivariate agri-data using Python Connect field devices to cloud platform (AWS IoT or Firebase) Alert system for low soil moisture via SMS/email Forecast rainfall using time series techniques Build a mobile-based decision support system (low-code platform) Capstone: Build and present a complete Smart Farm prototype
TEXTBOOKS Rajesh Singh, Anita Gehlot, Bhupendra Singh, Internet of Things and Wireless Sensor Networks in Smart Agriculture, CRC Press, 2021. Subhas Chandra Mukhopadhyay, Internet of Things in Smart Agriculture, Springer, 2020.
REFERENCES Himanshu Patel, Smart Farming Technologies for Sustainable Agricultural Development, IGI Global, 2020. G. R. Kanagachidambaresan, Internet of Things for Sustainable Community Development, Springer, 2021. IEEE Papers and Reports on Smart Agriculture, Remote Sensing, and Precision Farming
JOURNALS/MAGAZINES EEE Access – Special Section on Smart Agriculture Computers and Electronics in Agriculture (Elsevier) Agricultural Systems Journal Smart Farming Magazine IoT for Agri-Tech (IoT World Today) SWAYAM/NPTEL/MOOCs: NPTEL – Introduction to Smart Agriculture (IIT Kanpur) https://onlinecourses.nptel.ac.in/noc22_ge15/preview NPTEL – Applications of IoT in Agriculture (IIT Kharagpur)

https://onlinecourses.nptel.ac.in/noc23_cs91/preview Coursera – Smart Agriculture with IoT https://www.coursera.org/learn/smart-agriculture-iot

Course Code: CAI3415	Course Name: AI for Autonomous Systems Type of Course: Theory Integrated Lab	L- T-P- C	2-0-2-3
Version No.	1.0		
NCrF Level-	5.5		
Course Pre-requisites	CAI2500		
Co-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course explores the design and development of intelligent autonomous systems using artificial intelligence. It covers perception, sensor fusion, navigation, learning-based decision-making, and system integration. Emphasis is placed on real-time deployment, simulation environments, and edge AI, enabling the creation of adaptive, safe, and efficient autonomous agents across various domains.		
Course Objective	To equip students with knowledge and practical skills in using AI techniques to develop safe, adaptive, and intelligent autonomous systems that can sense, plan, and act in dynamic environments.		
Course Outcomes	On successful completion of this course, the students shall be able to:		
	CO1: Explain the architectural components and working principles of autonomous systems.		(L2 – Understand)
	CO2: Apply computer vision techniques and sensor fusion methods for environmental perception.		(L3 – Apply)
	CO3: Apply path planning and control algorithms for autonomous navigation.		(L3 – Apply)
	CO4: Implement learning-based models for real-time decision-making in autonomous systems.		(L3 – Apply)
	CO5: Analyze the performance and behaviour of autonomous systems in dynamic environments.		(L4 – Analyze)
Course Content:			
Module 1	Foundations of Autonomous Systems	Assessment :Assignment	12 hours (8L + 4P)
Topics: Types of autonomous systems: ground, aerial, marine, hybrid; Core components: sensing, planning, control, communication; System architecture: Perception–Planning–Action loop; Overview of ROS and ROS2			

middleware architecture; Hardware integration: sensors, actuators, computing platforms; Case studies: Autonomous vehicle stack, drone delivery systems; Ethical and legal considerations: functional safety (ISO 26262), explainability			
Module 2	Perception, Mapping and Sensor Fusion	Assessment : Assignment	12 hours (8L + 4P)
Topics: Sensor technologies: LiDAR, Radar, RGB-D, GPS, IMU, Ultrasonic; Visual perception: object detection (YOLOv8), segmentation, lane tracking; SLAM: ORB-SLAM2, Cartographer, Visual-Inertial Odometry (VIO); Environment representation: occupancy grid, voxel map, OctoMaps; Sensor fusion techniques: Kalman Filter, Extended Kalman Filter, Particle Filter; Practical: Fuse IMU and GPS data; lane detection using OpenCV			
Module 3	Localization, Planning and Navigation	Assessment : Assignment	12 hours (8L + 4P)
Topics: Localization methods: AMCL, GPS-based tracking, sensor fusion; Path planning: Dijkstra, A*, RRT, D* Lite for dynamic environments; Reactive navigation: Potential Fields, Dynamic Window Approach; Trajectory generation: B-splines, velocity profiling, cost maps; Behavioral planning: Finite State Machines, Behavior Trees; Case study: Indoor robot vs. outdoor drone navigation challenges			
Module 4	Learning-Based Control and Decision-Making	Assessment : Assignment	12 hours (8L + 4P)
Topics: Reinforcement learning: Q-Learning, Deep Q Networks, PPO; Decision-making under uncertainty: MDPs, POMDPs; Imitation learning: behavioral cloning and inverse RL; Edge deployment: TensorFlow Lite, NVIDIA Jetson for inference; Simulation environments: OpenAI Gym, CARLA, Webots, Gazebo; Case study: Train agent for lane follows in CARLA using DQN			
Module 5	System Integration, Simulation and Testing	Assessment : Mini Project	12 hours (8L + 4P)
Topics: Autonomous system stack integration in ROS2; Sensor pipeline design and ROS launch files; Simulator-based validation using Gazebo/CARLA/Webots; Autonomous testing: scenario generation, metrics, KPIs; Real-world deployment: latency, energy, fail-safes; Capstone: Full-stack testing on a simulated robot or drone; Mini Project: Build and validate an autonomous prototype (individual/team-based)			

Targeted Applications & Tools:

Tools/Software/Platform	Application
ROS, Gazebo, RViz, Python	Autonomous Navigation in a Grid World
OpenCV, YOLOv5, Python, Jupyter Notebook	Real-time Object Detection in Road Scenes
Kalman Filter, Python, ROS, simulated sensor data	Sensor Fusion for Robot Localization
CARLA Simulator, Python, Deep Reinforcement Learning Libraries	Simulated Driving Agent in CARLA
GMapping/Cartographer, ROS, TurtleBot (or simulator)	SLAM-based Environment Mapping and Navigation

TensorFlow Lite, Raspberry Pi/Jetson Nano, Python	Edge AI Object Classification on Live Camera Feed

Skill-Building Activities through Experiential Learning:

- Autonomous Parking Assistant Challenge
  Design and simulate an autonomous parking assistant that detects parking spaces and performs parallel parking. The activity enhances understanding of perception, decision-making, and motion planning using ROS, OpenCV, and Gazebo in Python.

- Sim-to-Real Deployment Drill

  Develop a robot navigation pipeline in simulation using platforms like CARLA or Gazebo and deploy the same system on a physical robot. This activity builds practical knowledge on transferring AI models from virtual simulations to real-world environments.

- Sensor Fusion Escape Maze
  Guide a robot through a simulated maze using GPS and IMU data. Compare Kalman and particle filter techniques to improve localization. The activity develops intuition for handling uncertainty and integrating multiple sensors in autonomous systems.

List of laboratory tasks:

1. **Simulate a Differential Drive Robot in ROS2**
   Configure URDF and launch files in ROS2 to simulate motion dynamics and sensor feedback for a two-wheel robot using RViz.
2. **Interface Camera and Distance Sensors with Raspberry Pi**
   Capture real-time video feed and distance measurements; stream data to a local server for monitoring and preliminary processing.
3. **Object Detection using Pre-trained YOLOv5 in Real-time Video Streams**
   Integrate YOLOv5 with OpenCV to detect pedestrians, vehicles, and obstacles from a moving platform or live webcam.
4. **Lane and Road Detection using HSV and Canny-based Segmentation**
   Perform robust road lane detection using color and edge-based techniques; apply perspective transformation for bird's-eye view.
5. **Sensor Fusion: Integrate GPS and IMU Data Using Extended Kalman Filter**
   Localize a mobile robot on a map by combining noisy GPS and IMU readings to enhance positional accuracy.
6. **Implement Visual SLAM using RTAB-Map or ORB-SLAM2**
   Perform real-time 3D mapping and localization in indoor environments using RGB-D camera data.
7. **Obstacle Detection and Avoidance using LiDAR and Costmap Layers**
   Use raw LiDAR data to detect dynamic/static obstacles and plan safe navigation paths in ROS Navigation Stack.
8. **Path Planning with A and D Algorithms on Dynamic Grid Maps****
   Compare static vs. dynamic re-planning using simulated agents navigating through obstacle-laden environments.
9. **Simulate Urban Driving Scenarios using CARLA with Sensor Suite**
   Build an end-to-end simulation in CARLA with camera, LiDAR, and radar data streams to test autonomous driving pipelines.
10. **Implement PID and Pure Pursuit Controllers for Path Tracking**
    Tune PID parameters and compare with geometric tracking controllers for smooth trajectory following in differential robots.

11. **Reinforcement Learning Agent for Goal-based Navigation in Grid World**
    Train an agent using Q-learning to reach a specified goal state while avoiding obstacles and penalties.
12. **Edge Deployment: Run a TensorFlow Lite Model for Real-time Object Detection on Jetson Nano**
    Quantize and deploy lightweight models for object recognition on resource-constrained hardware platforms.
13. **Compare Decision Trees, SVM, and k-NN for Task Classification in Robotics**
    Use a dataset (e.g., sensor-based actions or object types) to train and evaluate classification models for task decision making.
14. **Traffic Sign and Signal Recognition using Haar Cascades and CNNs**
    Detect and classify road signs and traffic lights in video sequences; integrate results into behavior planning logic.
15. **Capstone Lab: Design, Integrate, and Test a Miniature Autonomous Robot Prototype**
    Combine perception, planning, control, and learning modules into a physical or simulated mobile platform to demonstrate autonomous navigation in a controlled environment.

Text Book

- **T1**: Raj Madhavan, Autonomous Systems: Issues and Challenges, Springer, 2021
- **T2**: Corke, P. (2021) Robotic Vision: Fundamental Algorithms in MATLAB®. Switzerland: Springer International Publishing.

References

R1: Thrun, S., Burgard, W., Fox, D. (2005). Probabilistic robotics. Cambridge: MIT Press.
R2: Khemani, D. (2013). A First Course in Artificial Intelligence. India: McGraw Hill Education (India).
R3: Benjamin Kuipers, Principles of Intelligent Autonomous Systems, MIT Press, 2016
R4: Rehtanz, C. (2010). Autonomous Systems and Intelligent Agents in Power System Control and Operation. Germany: Springer.

E-Resources/Web Resources

- JOURNALS/MAGAZINES
  - IEEE Transactions on Robotics
  - Journal of Field Robotics (Wiley)
  - Autonomous Robots (Springer)
  - IEEE Transactions on Intelligent Vehicles

SWAYAM/NPTEL/MOOCs:

- NPTEL – Robotics: Perception, Planning and Control (IIT Kanpur)
  https://onlinecourses.nptel.ac.in/noc21_ee68/preview
- NPTEL – Artificial Intelligence: Search Methods for Problem Solving (IIT Madras)
  https://onlinecourses.nptel.ac.in/noc21_cs53/preview
- Self-Driving Cars Specialization (University of Toronto)
  https://www.coursera.org/specializations/self-driving-cars

Course Code: CAI3416	Course Title: Edge Computing Type of Course: Theory Integrated Lab	L- T-P- C	2-0-2-3
Version No.	1.0		
Course Pre-requisites	CSS2264		
Anti-requisites	NIL		
Course Description	This course introduces the fundamentals of edge computing, an emerging paradigm that enables data processing and analytics at the network's edge. It covers system architecture, communication protocols, edge AI, and real-time applications with a hands-on approach using edge devices.		
Course Objective	To equip students with knowledge and skills in designing and deploying edge computing systems that offer low-latency, energy-efficient, and scalable solutions for real-time and distributed applications.		
Course Outcomes	On successful completion of this course, the students shall be able to: CO1: Explain the principles and architecture of edge computing systems. (L2 – Understand) CO2: Develop and deploy applications on edge devices such as Raspberry Pi and Jetson Nano. (L3 – Apply) CO3: Integrate edge computing with IoT, cloud, and AI systems for real-time applications. (L3 – Apply) CO4: Apply edge AI techniques for data processing and analytics on resource-constrained devices. (L3 – Apply) CO5: Analyze and optimize the performance of edge-based applications in distributed environments . (L4 – Analyze)		
Course Content:			
Module 1	Introduction to Edge Computing	Assessment: Assignment	
Topics: Evolution from cloud to fog to edge, Edge computing architecture: core, edge, and device layers, Use cases: smart cities, industrial automation, healthcare, autonomous systems, Edge vs. Cloud: latency, bandwidth, privacy, scalability, Edge device overview: Raspberry Pi, Nvidia Jetson, Coral Dev Board			
Module 2	Edge Devices and Communication	Assessment: Assignment	

Topics: Edge device setup and configuration, Operating systems: Raspbian, Ubuntu Core, Yocto, Edge and fog networking protocols: MQTT, CoAP, 6LoWPAN, Containerization with Docker at the edge, Device orchestration and provisioning, Energy and resource management in constrained devices.			
Module 3	Edge AI and Analytics	Assessment: Assignment	
Topics: Edge AI: Concepts and advantages, Lightweight ML frameworks: TensorFlow Lite, ONNX, OpenVINO, Model quantization and compression for deployment, Real-time data processing and event-based analytics, Case studies: object detection, audio classification, sensor fusion, Ethical considerations and security at the edge			
Module 4	Edge Computing Platforms & Communication	Assessment: Assignment	
Edge-to-cloud communication pipelines, edge data flow architectures, AWS Greengrass, Azure IoT Edge, Google Edge TPU, deployment of AI models at the edge, microservices architecture on the edge, serverless computing at the edge.			
Module 5	Industrial Edge Systems & Applications – Topics	Assessment: Assignment	
Industrial edge computing concepts, communication protocols: OPC-UA and Modbus, real-time data acquisition and processing, building dashboards for edge data visualization and insights, monitoring and analytics, capstone project: end-to-end edge system for real-world applications.			
Project work/Assignment:			
Assignment 1: IoT Data Stream Processing at the Edge Assignment 2: Edge AI Inference Deployment Mini Project (Team-based): Real-World Edge Solution			
List of Lab Tasks : Setting up Raspberry Pi or Jetson Nano for edge deployment Installing and configuring Docker on an edge device Connecting sensors (camera, DHT11, ultrasonic) to edge devices Building and running MQTT-based data pipeline Collecting and visualizing data using Node-RED or Grafana Edge inference using TensorFlow Lite object detection model Deploying ONNX model for sensor-based classification Streaming video analytics at the edge			

Building a containerized AI service on the edge Edge-to-cloud integration using AWS Greengrass or Azure IoT Edge Monitoring resource usage on constrained devices Detecting anomalies in time-series data at the edge Designing a simple mobile dashboard to view edge insights Security setup: HTTPS, authentication, and encryption Capstone: Build a complete edge-to-cloud application prototype.
TEXTBOOKS Perry Lea, Edge Computing: From Hype to Reality, Packt Publishing, 2020 Mahmoud Tawfik, Edge Computing and Internet of Things for Smart Agriculture, Springer, 2021
REFERENCES Satyanarayanan M., The Emergence of Edge Computing, IEEE Computer, 2017 Flavio Bonomi et al., Fog Computing and Its Role in the Internet of Things, MCC Workshop, ACM 2012 OpenFog Consortium Architecture White Paper, 2017
JOURNALS/MAGAZINES IEEE Internet of Things Journal ACM Transactions on Internet Technology (TOIT) IEEE Edge Computing Magazine Elsevier Future Generation Computer Systems – Special Issues on Edge Computing SWAYAM/NPTEL/MOOCs: NPTEL – Cloud Computing and Distributed Systems (IIT Kharagpur) https://onlinecourses.nptel.ac.in/noc22_cs62/preview NPTEL – Internet of Things (IIT Kharagpur) https://onlinecourses.nptel.ac.in/noc21_cs60/preview Coursera – Edge AI and Computer Vision (Intel + OpenVINO) https://www.coursera.org/learn/introduction-to-edge-ai

Course Code: CAI3417	Course Name: Cognitive Computing Type of Course: Theory Integrated Lab	L- T-P- C	2-0-2-3
Version No.	1.0		
NCrF Level-	5.5		

Course Pre-requisites	CAI2500		
Co-requisites	Nil		
Anti-requisites	Nil		
Course Description	This course delves into cognitive computing systems that mimic human reasoning, perception, and interaction. It integrates machine learning, natural language understanding, vision, and speech processing using cognitive APIs and transformer-based models. Emphasis is placed on building ethical, multimodal, and intelligent applications using real-world frameworks and platforms.		
Course Objective	To enable students to design and implement cognitive computing systems that emulate human intelligence using advanced AI techniques. To develop proficiency in applying machine learning, NLP, vision, and speech technologies ethically through real-world cognitive frameworks.		
Course Outcomes	On completion of this course, students will be able to: **CO1:** Describe the architecture and core components of cognitive computing systems. (Level 2 - Understand) **CO2:** Identify appropriate machine learning and natural language processing techniques for cognitive tasks. (Level 2 - Understand) **CO3:** Demonstrate the use of cognitive APIs for text, speech, and image processing. (Level 3 - Apply) **CO4:** Develop basic cognitive applications using tools such as IBM Watson and transformer models. (Level 6 - Create) **CO5:** Evaluate ethical considerations and societal impacts related to cognitive computing solutions. (Level 5 - Evaluate)		
Course Content:			
Module 1	**Foundations of Cognitive Computing**	Assessment : Assignment	12 Sessions (6 L + 6 P)
Definition and evolution of cognitive computing, Cognitive architecture Cognitive vs. traditional AI systems Cognitive loop: perception, interpretation, reasoning, learning Human-computer interaction and multimodal cognition Architecture: symbolic reasoning vs. neural inference Applications: decision support, assistive tech, personalization			
Module 2	**Machine Learning & Deep Learning for Cognition**	Assessment : Assignment	12 Sessions (6 L + 6 P)
Overview of ML, DL, and neural networks for cognition Feature representation and embeddings CNNs for vision and LSTMs/Transformers for sequence modeling Cognitive APIs overview (IBM Watson, Azure, HuggingFace, Google AI , OpenAI) Building custom pipelines using pretrained models			
Module 3	**Natural Language**	Assessment : Assignment	12 Sessions

	Understanding and Dialogue Systems		(6 L + 6 P)
Language modeling, parsing, and embeddings Transformer architectures: BERT, RoBERTa, GPT Intent recognition, dialogue flow, Q&A systems Sentiment, summarization, translation, named entity recognition Chatbots and virtual agents using Dialogflow, Rasa, and LLM APIs			
Module 4	**Multimodal and Cognitive APIs in Practice**	Assessment : Assignment	12 Sessions (6 L + 6 P)
Visual recognition and image captioning (Vision APIs) Speech processing: STT and TTS (Google, IBM, Amazon) Multimodal cognition: integrating text, speech, vision Deploying cognitive services as microservices Real-world case studies in healthcare, education, and assistive tech			
Module 5	Ethics, Explainability and Deployment of Cognitive Systems	Assessment : Assignment	12 Sessions (6 L + 6 P)
Ethical issues in cognitive computing: bias, fairness, transparency. Explainable AI (XAI): interpretability of ML and deep learning models. Bias detection and mitigation techniques (e.g., fairness toolkits) . Privacy, security, and responsible AI practices. Model evaluation metrics for cognitive systems (accuracy, F1, BLEU, etc.) . Deployment of cognitive applications using APIs and cloud platforms .Monitoring, feedback loops, and continuous learning systems . Case studies on responsible AI in healthcare, finance, and social systems			

List of laboratory tasks:

Lab 1: Exploring Cognitive Computing Platforms

- **Objective:** Understand the landscape of cognitive platforms and tools.
- **Task:** Set up IBM Watson, Microsoft Azure Cognitive Services, and Hugging Face accounts.
- **Activity:** Compare feature sets and APIs. Build a basic "Hello Cognitive World" application demonstrating a simple NLP or visual recognition task across two platforms.

Lab 2: Sentiment Analysis with Multilingual Support

- **Objective:** Analyze sentiment in user-generated content across multiple languages.
- **Task:** Use IBM Watson NLP and Hugging Face Transformers for multilingual sentiment analysis.
- **Activity:** Visualize sentiment trends in tweets, product reviews, or news headlines across English, Hindi, and Spanish.

Lab 3: Named Entity Recognition and Contextual Mapping

- **Objective:** Extract and relate entities from text using contextual understanding.
- **Task:** Use spaCy and BERT-based NER models.
- **Activity:** Highlight and cluster related entities from articles (e.g., COVID-19 news), then map them

to categories like people, organizations, locations.

Lab 4: Real-time Speech-to-Text Transcription and Analysis

- **Objective:** Convert live or recorded speech to text and perform keyword detection.
- **Task:** Integrate Google Cloud API or Whisper with Python.
- **Activity:** Record speech samples and extract named entities or keywords in real-time.

Lab 5: Interactive Text-to-Speech Web Assistant

- **Objective:** Build an interactive TTS interface.
- **Task:** Use IBM Watson or Google Text-to-Speech with user-defined parameters.
- **Activity:** Deploy a web application that accepts user input and reads it aloud with tone/emotion settings.

Lab 6: Visual Recognition for Real-Time Scene Detection

- **Objective:** Identify objects and scenes from webcam or image feed.
- **Task:** Use Watson Visual Recognition and YOLO/ViT models.
- **Activity:** Tag scenes in real-time (e.g., indoor/outdoor, traffic signs, furniture) and display confidence scores.

Lab 7: Building a Conversational AI with Personality

- **Objective:** Create a context-aware, goal-driven chatbot.
- **Task:** Design intents, entities, and small talk personality traits using Dialogflow or Rasa.
- **Activity:** Deploy on a website that responds differently to formal/informal queries and handles multi-turn conversations.

Lab 8: Transformer-based Question Answering System

- **Objective:** Use transformers for context-aware Q&A.
- **Task:** Integrate BERT or DistilBERT with custom document datasets.
- **Activity:** Upload a knowledge base (PDF/text), ask factual or inferential questions, and visualize responses with highlighted context.

Lab 9: Dynamic Knowledge Graph Construction from Text

- **Objective:** Create and query semantic knowledge representations.
- **Task:** Use Neo4j to construct graphs from extracted text data (NER + relations).
- **Activity:** Feed articles or transcripts to auto-generate entity-relationship graphs and query insights (e.g., "Who worked with whom on Project X?").

Lab 10: LSTM-based Sentiment Classification with User Feedback Loop

- **Objective:** Train, test, and fine-tune a sentiment classifier.
- **Task:** Build an LSTM model and incorporate real-time user feedback to improve predictions.
- **Activity:** Evaluate model using updated feedback and retrain iteratively.

Lab 11: Neural Image Captioning with Transfer Learning

- **Objective:** Generate contextual image descriptions using CNN-RNN.
- **Task:** Use pre-trained CNN (e.g., ResNet) with LSTM decoder.
- **Activity:** Upload photos and display human-like captions, evaluate with BLEU/METEOR scores.

Lab 12: Emotion Detection from Facial Expressions in Real Time

- **Objective:** Detect facial expressions using webcam and ML models.
- **Task:** Integrate OpenCV with pre-trained emotion classifiers.
- **Activity:** Build a mood-tracking interface that logs emotion states over time.

Lab 13: Custom Voice Assistant with Intent Prediction

- **Objective:** Build a voice-controlled assistant capable of handling varied queries.
- **Task:** Use speech recognition + TTS + intent classifier (e.g., logistic regression or BERT).
- **Activity:** Execute basic tasks like opening apps, reading calendar, or answering questions via voice.

Lab 14: Capstone Project – Build a Multi-Modal Intelligent System

- **Objective:** Integrate all modalities: vision, speech, and text.
- **Task:** Develop an assistive application (e.g., virtual classroom helper, smart assistant for visually impaired).
- **Activity:** Demonstrate end-to-end system, document architecture, and evaluate usability.

Targeted Applications & Tools:

Tools/Software/Platform	Application
IBM Watson NLP API, Hugging Face Transformers, Python, Jupyter Notebook	Analyze customer sentiment in multilingual product reviews
Dialogflow, Python Flask, Google Cloud Functions, HTML/CSS	Build a chatbot that answers student FAQs for a university website
IBM Watson Text-to-Speech, Sentiment Analysis API, JavaScript, Bootstrap	Create an app that reads out news articles with emotional tone
spaCy, Neo4j, NetworkX, Pandas	Extract key entities and build a knowledge graph from news headlines
SpeechRecognition (Python), pyttsx3, NLTK, Google TTS/STT APIs	Deploy a voice assistant for basic system control (weather, calendar, tasks)
ResNet, CNN-RNN models, TensorFlow, Keras, OpenCV	Identify objects in uploaded images and provide auto-generated captions
OpenCV, Pre-trained Emotion Detection Models, Dlib	Perform real-time facial expression detection from webcam feed
Hugging Face Transformers (BERT), Streamlit, Python	Build a QA system using BERT for answering questions from uploaded documents
LSTM (Keras/TensorFlow), Python, Streamlit	Train a sentiment classifier with user feedback loop for movie reviews
IBM Fairness 360 Toolkit, OpenCV, Scikit-learn	Detect model bias in gender classification from facial data

Skill-Building Activities through Experiential Learning:

- **Cognitive AI for Campus Feedback Analysis**

Participants gather feedback from social media, Google reviews, and surveys regarding campus events and facilities. Using IBM Watson NLP and spaCy, they perform sentiment analysis, entity extraction, and summarization to identify key concerns. This real-world exercise enhances skills in cognitive text analytics and social data interpretatio

- **Build a Multimodal Personal Assistant**

Learners design a simple personal assistant combining voice and visual capabilities. Using Google APIs, OpenCV, and Watson Text-to-Speech, the assistant can process spoken commands, identify objects through a webcam, and schedule reminders. The project introduces multimodal interaction concepts and practical AI integration for everyday tasks.

- **Bias Detection and Ethical Audit of a Cognitive Model**

Participants explore the ethical risks of cognitive AI systems through a bias detection simulation. Using the Fairness 360 Toolkit, they analyze pre-trained models for bias based on gender, race, or age. The activity includes reviewing real-world case studies and proposing strategies to mitigate discriminatory AI behavior.

Text Book

1. Hurwitz, J., Kaufman, M., Bowles, A. (2015). Cognitive Computing and Big Data Analytics. United States: John Wiley & Sons, Incorporated.
2. Rajiv Mathur, Cognitive Computing: Theory and Applications, CRC Press, 2022.
3. Cognitive Engineering for Next Generation Computing: A Practical Analytical Approach. (2021). United States: Wiley.

References

1. High, R., Bakshi, T. (2019). Cognitive Computing with IBM Watson: Build Smart Applications Using Artificial Intelligence as a Service. Germany: Packt Publishing.
2. Masood, A., Hashmi, A. (2019). Cognitive Computing Recipes: Artificial Intelligence Solutions Using Microsoft Cognitive Services and TensorFlow. Germany: Apress.
3. Sebastian Raschka, Natural Language Processing with Transformers, O'Reilly, 2021

E-Resources/Web Resources

JOURNALS/MAGAZINES

- IEEE Intelligent Systems
- ACM Transactions on Interactive Intelligent Systems (TIIS)
- Cognitive Computation (Springer)
- Journal of Artificial Intelligence Research (JAIR)

SWAYAM/NPTEL/MOOCs:

• NPTEL – Deep Learning for Computer Vision (IIT Hyderabad) https://nptel.ac.in/courses/106106231 • Coursera – Introduction to IBM Watson (IBM) https://www.coursera.org/learn/ai-watson edX – IBM Applied AI: Cognitive Services https://www.edx.org/professional-certificate/ibm-applied-artificial-intelligence

Course Code: CAI3418	Course Name: Geospatial Data Analytics Type of Course: Theory Integrated Lab	L- T-P- C	2-0-2-3
Version No.	1.0		
NCrF Level-	5.5		
Course Pre-requisites	CSS2264		
Co-requisites	Nil		
Anti-requisites	Nil		
Course Description	This course introduces the fundamentals of geospatial data and spatial analysis using modern analytical tools and technologies. It focuses on acquiring, visualizing, processing, and analyzing geographic data to extract meaningful patterns. Applications include urban planning, environmental monitoring, remote sensing, and location-based services.		
Course Objective	To understand the fundamentals of geospatial data and coordinate systems. To perform spatial data processing and analysis using GIS tools. To visualize geospatial data using mapping libraries and dashboards. To apply machine learning and statistical models to spatial datasets.		
Course Outcomes	On completion of this course, students will be able to: CO1: Describe the fundamental concepts of geospatial data, coordinate systems, and GIS architecture. (Level 2 - Understand) CO2: Apply geospatial data handling techniques using formats, tools, and libraries such as GeoPandas. (Level 3 - Apply) CO3: Perform spatial analysis and geoprocessing using vector and raster data. (Level 4 - Analyze) CO4: Develop geospatial visualizations and interactive dashboards using GIS platforms and Python-based tools. (Level 6 - Create) CO5: Evaluate geospatial solutions for real-world applications considering data quality, ethics, and societal impact. (Level 5 - Evaluate)		
Course Content: **I**			
Module 1	Introduction to Geospatial Data and GIS	**Assessment:** Assignment	12 Sessions (6 L + 6 P)
Basics of geospatial data: raster vs vector, Coordinate reference systems (CRS) and map projections, GIS architecture and components, Geospatial data sources: satellite, drone, GPS, open data.			
Module 2	Geospatial Data Handling and	**Assessment:** Assignment	12 Sessions (6 L + 6 P)

	Visualization		
Shapefiles, GeoJSON, and TIFF formats, Loading and manipulating geospatial data with GeoPandas, Map visualizations with Folium, Kepler.gl, and Plotly, Spatial joins, clipping, filtering, and reprojection			
Module 3	Spatial Analysis and Remote Sensing	**Assessment:** Assignment	12 Sessions (6 L + 6 P)
Overlay operations, buffering, and geoprocessing, Raster analysis using rasterio, Remote sensing concepts and multispectral image analysis, NDVI, LST, and change detection from satellite images.			
Module 4	Applications and Machine Learning in Geospatial Analytics	**Assessment:** Assignment	12 Sessions (6 L + 6 P)
Spatial clustering and hotspot analysis, Geospatial ML: land cover classification, object detection, Time series and spatiotemporal data analysis, Case studies in disaster management, urban planning, and climate monitoring.			
Module 5	Geospatial Systems, Ethics and Real-World Deployment	**Assessment:** Assignment	12 Sessions (6 L + 6 P)
Web GIS architecture and APIs. Interactive dashboards using Leaflet, Mapbox, Streamlit. Cloud integration and real-time geospatial pipelines. Data quality, uncertainty, and error analysis. Ethics in geospatial analytics: privacy and surveillance. Location privacy and anonymization. Deployment of geospatial decision support systems. Case studies: smart cities, transport, environment			

List of Lab Tasks

Lab 1: Introduction to QGIS and Map Projections
Objective: Understand GIS interface and coordinate systems
Tasks: Load vector data and explore CRS
Activity: Visualize administrative boundaries and reproject layers

Lab 2: Handle Vector Data using GeoPandas
Objective: Load and manipulate shapefiles in Python
Tasks: Read, filter, and plot shapefile data
Activity: Analyze India's state-level boundaries using GeoPandas

Lab 3: Perform Spatial Joins and Buffering
Objective: Learn spatial joins and geoprocessing
Tasks: Combine population and boundary datasets
Activity: Create buffer zones around city centers

Lab 4: Visualize Maps using Folium and Kepler.gl
Objective: Create interactive maps
Tasks: Generate heatmaps and choropleth maps
Activity: Visualize crime or COVID-19 data on an interactive map

Lab 5: Work with Raster Data using Rasterio
Objective: Read and manipulate satellite imagery
Tasks: Load and clip raster images
Activity: Display elevation or vegetation maps

Lab 6: Calculate NDVI from Satellite Images
Objective: Perform vegetation index analysis

Tasks: Use Red and NIR bands to compute NDVI
Activity: Visualize vegetation cover changes

Lab 7: Geocoding and Reverse Geocoding
Objective: Convert addresses to coordinates
Tasks: Use Geopy and Google Maps API
Activity: Geocode location list and map them

Lab 8: Perform Land Use Classification using K-Means
Objective: Apply clustering to satellite images
Tasks: Use unsupervised learning on raster data
Activity: Classify urban vs green areas

Lab 9: Detect Change over Time in Remote Sensing Data
Objective: Analyze temporal changes
Tasks: Compare satellite images from different years
Activity: Detect urban expansion or deforestation

Lab 10: Build Dashboard using Plotly and Dash
Objective: Create an interactive geospatial dashboard
Tasks: Combine charts and maps
Activity: Display flood zones with demographic data

Lab 11: Perform Hotspot and Cluster Analysis
Objective: Identify spatial patterns
Tasks: Use DBSCAN or Getis-Ord Gi* statistic
Activity: Detect crime or disease hotspots

Lab 12: Integrate GPS Data for Route Mapping
Objective: Process and visualize GPS tracks
Tasks: Load GPX/CSV files and plot paths
Activity: Analyze cycling or delivery routes

Lab 13: Apply Spatial Regression Models
Objective: Model spatial relationships
Tasks: Fit spatial autoregression (SAR) or GWR
Activity: Predict housing prices based on location

Lab 14: Use Google Earth Engine for Remote Sensing Analysis
Objective: Access cloud-based satellite processing
Tasks: Load and analyze Sentinel/Landsat data
Activity: Monitor water bodies or land surface temperature

Lab 15: Capstone – Geospatial Data Analytics Project
Objective: Apply geospatial techniques to a real dataset
Tasks: Perform end-to-end analysis
Activity: Present findings via dashboard or report

Text Book

1. Paul Longley et al., Geographic Information Systems and Science, Wiley, 4th Edition, 2015
2. Bolstad, Paul, GIS Fundamentals: A First Text on Geographic Information Systems, Eider Press, 6th

Edition, 2019
References Michael Dorman, Spatial Data Analysis in Python, Manning Publications, 2023 Andrew Cutts, Geospatial Analysis: A Comprehensive Guide, Winchelsea Press Bonny P. McClain, Mastering Geospatial Analysis with Python, Packt Publishing, 2022
E-Resources/Web Resources International Journal of Geographical Information Science Remote Sensing of Environment Journal of Spatial Information Science GIScience& Remote Sensing NPTEL – Introduction to GIS (IIT Roorkee) https://nptel.ac.in/courses/105107120 Coursera – Geospatial and Environmental Analysis (UC Davis) https://www.coursera.org/learn/environmental-analysis edX – Geospatial Data Science and Applications (Tsinghua University) https://www.edx.org/course/geospatial-data-science Google Earth Engine Tutorials https://developers.google.com/earth-engine/tutorials

Course Code: CAI3419	Course Name: **AI for energy consumption optimization** Type of Course: Theory Integrated Lab	L- T-P- C	2-0-2-3
Version No.	1.0		
NCrF Level-	5.5		
Course Pre-requisites	CAI2500		
Co-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course focuses on applying artificial intelligence techniques to optimize energy consumption in smart buildings, industrial systems, and renewable energy grids. It introduces energy systems, data acquisition, and preprocessing methods for real-world energy datasets. The course covers forecasting, anomaly detection, and intelligent control using machine learning, deep learning, and reinforcement learning techniques. Emphasis is placed on real-		

	time data analysis, simulation, and sustainable energy management through practical, hands-on learning and real-world applications.		
Course Objective	This course aims to impart practical knowledge in applying artificial intelligence techniques for optimizing energy consumption in smart systems. It focuses on developing skills in data acquisition and preprocessing, energy forecasting, anomaly detection, and intelligent control using machine learning, deep learning, and reinforcement learning methods. The course also emphasizes improving energy efficiency, reducing waste, and enabling sustainable energy management through experiential and hands-on learning.		
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain the structure and functioning of energy systems, including generation, consumption patterns, and optimization challenges. (Level 2 – Understand) **CO2:** Apply data preprocessing techniques to clean, transform, and prepare energy datasets from smart meters and IoT sources for analysis. (Level 3 – Apply) **CO3:** Apply machine learning and time-series models to forecast energy consumption and detect anomalies in energy usage data. (Level 3 – Apply) **CO4:** Analyze and implement AI-based control strategies, including deep learning and reinforcement learning, for optimizing energy usage in smart systems. (Level 4 – Analyze) **CO5:** Apply AI techniques to real-world energy applications, including smart buildings, renewable energy integration, and sustainable energy management systems. (Level 3 – Apply)		
Course Content:			
Module 1	**Introduction to Energy Systems and Optimization**	Assessment : Assignment	6 Sessions
Topics: Overview of energy generation, transmission, and consumption Challenges in energy optimization: cost, load, and efficiency Smart energy systems and the role of IoT sensors Introduction to energy usage, scheduling, and optimization techniques Energy management frameworks and demand-side strategies			
Module 2	**Data Handling and Preprocessing for Energy Systems**	Assessment : Assignment	6 Sessions
Topics: Sources of energy data: smart meters, IoT, BAS ,Data cleaning: missing values, noise handling Normalization and feature engineering ,Time-series alignment and preprocessing pipelines			
Module 3	Machine Learning for Energy Forecasting	Assessment : Assignment	6 Sessions
Regression models: Linear Regression, Decision Trees ,Time-series forecasting: ARIMA, Prophet Deep learning intro for forecasting (LSTM basics) ,Model evaluation: MAE, RMSE, MAPE			
Module 4	AI for Energy Control and Optimization	Assessment : Assignment	6 Sessions
Topics: Deep learning for consumption patterns , Anomaly detection (autoencoders, Isolation Forest) Reinforcement learning for energy control ,Case study: HVAC optimization ,IoT feedback and control systems			
Module 5	**Applications and Emerging**	Assessment : Assignment	6 Sessions

	Trends in Smart Energy		

AI in smart homes, buildings, cities ,Renewable energy integration (solar, wind forecasting)
Edge AI and federated learning ,Industrial and microgrid case studies ,Green AI and sustainable computing trend**s**

Project work/Assignment:
Develop a forecasting model for household energy optimization, implement an RL-based agent to control energy consumption in a simulated environment, and design a smart energy dashboard with real-time analytics.

List of laboratory tasks:

- **Lab 1:** Load, explore, and visualize energy consumption datasets from sources such as UCI and OpenEI; identify trends, seasonal patterns, and missing data.
- **Lab 2:** Clean and preprocess raw energy data using techniques like imputation, normalization, feature extraction, and time-index alignment for model readiness.
- **Lab 3:** Build and evaluate a linear regression model to forecast residential or industrial energy consumption using historical datasets.
- **Lab 4:** Apply decision tree and random forest classifiers to categorize energy usage profiles by consumer type or appliance behavior.
- **Lab 5:** Implement an ARIMA model for time-series forecasting, and evaluate its performance using metrics like RMSE and AIC on seasonal energy data.
- **Lab 6:** Develop an LSTM-based deep learning model to capture long-term dependencies in smart meter readings for high-accuracy energy prediction.
- **Lab 7:** Perform anomaly detection using statistical and machine learning methods (e.g., Isolation Forest) to identify irregularities in smart grid data.

- **Lab 8:** Develop and interpret a clustering model (e.g., K-Means, DBSCAN) to segment usage patterns across consumer groups or device classes.
- **Lab 9:** Design and deploy an interactive dashboard using tools like Plotly Dash or Power BI for real-time and historical energy data visualization.
- **Lab 10:** Build a rule-based control logic system using Python or Node-RED to automate energy-efficient operation of smart home appliances.
- **Lab 11:** Implement Q-learning to dynamically control a simulated smart thermostat and optimize energy usage based on occupancy and weather.
- **Lab 12:** Use reinforcement learning (e.g., Deep Q-Network) to optimize lighting control systems in smart buildings for comfort and energy efficiency.
- **Lab 13:** Forecast solar energy generation using hybrid models that integrate historical weather data, panel orientation, and local consumption patterns.
- **Lab 14:** Integrate live IoT sensor data with AI models for real-time intelligent decision-making in energy usage and predictive maintenance.
- **Lab 15:Capstone Project:** Design and prototype a smart energy optimization system combining forecasting, control, and visualization for an IoT-based microgrid

Targeted Applications & Tools:

Tools/Software/Platform	**Application**
Python, Pandas, Scikit-learn, ARIMA, Prophet, Matplotlib	Build a predictive model to forecast daily household energy usage patterns.
Python, Scikit-learn, Autoencoders, TensorFlow, Smart Meter Data	Detect abnormal spikes or drops in energy usage using AI-based anomaly detection.
OpenAI Gym, Python, TensorFlow, Simulated HVAC Environment	Implement a reinforcement learning agent to control HVAC for energy efficiency.

Python, LSTM, Pandas, OpenWeather API, TensorFlow/Keras	Predict solar energy generation based on weather data and historical trends.

Skill-Building Activities through Experiential Learning:

- **Smart Home Energy Forecasting Challenge**
  Learners build a time-series forecasting model using household energy data to predict daily consumption. They compare ARIMA, Prophet, and LSTM models and visualize the forecasts to support decision-making for smart home energy management.

- **Reinforcement Learning for HVAC Optimization**
  Participants implement a reinforcement learning agent to control a simulated HVAC system. The activity focuses on reducing energy usage while maintaining comfort, introducing concepts like state-action space, reward functions, and real-world challenges in energy-efficient control.

- **Anomaly Detection in Smart Grid Data**
  Learners apply unsupervised learning techniques to detect anomalies in smart meter datasets. Using autoencoders and clustering methods, they identify unusual consumption patterns that could indicate faults, inefficiencies, or energy theft, building practical skills in AI-based monitoring.

Text Book

T1: Abhishek Kumar, Machine Learning and Data Science in the Energy Sector, Wiley, 2022.

T2: Peter Palensky et al., Energy Informatics: Fundamentals and Applications, Springer, 2021.

References

R1: Soteris Kalogirou, Artificial Intelligence in Energy and Renewable Energy Systems, Nova Science Publishers, 2013.
R2:Klaus-Dieter Thoben et al., AI Methods for Smart Energy Systems and Industry 4.0, Springer, 2020.
R3:SubramanianVadari, Smart Grid Redefined: Transformation of the Electric Utility, CRC Press, 2020.

E-Resources/Web Resources

1. Soteris Kalogirou, Artificial Intelligence in Energy and Renewable Energy Systems, Nova Science Publishers, 2013.
2. Klaus-Dieter Thoben et al., AI Methods for Smart Energy Systems and Industry 4.0, Springer, 2020.
3. Subramanian Vadari, Smart Grid Redefined: Transformation of the Electric Utility, CRC Press, 2020.
4. NPTEL – Smart Grid Technology by IIT Kharagpur
5. Coursera – AI for Energy
6. edX – Data Science and Machine Learning for Energy Systems

Course Code: CAI3420	Course Title: Bio Medical Informatics Type of Course: Theory Integrated Lab	L- P- T-C	2-0-2-3
Version No.	1.0		
Course Pre-requisites	CSS2264		

Anti-requisites	NIL
Course Description	This course provides a comprehensive introduction to Bio Medical Informatics, focusing on the acquisition, management, and analysis of healthcare data using modern computational and artificial intelligence techniques. It enables students to understand domain-specific biomedical data, healthcare information systems, and standards used in clinical environments. The course covers the complete data lifecycle, including data collection, preprocessing, feature engineering, and the development of machine learning models for healthcare applications such as disease prediction and clinical decision support. Emphasis is placed on practical implementation through hands-on exercises, as well as on ethical, legal, and societal considerations in the use of AI in healthcare. By integrating theoretical knowledge with experiential learning, the course prepares students to design and deploy intelligent solutions for real-world challenges in the health sector.
Course Objective	The objective of this course is to promote **skill development** in Bio Medical Informatics through **experiential learning**, by: Building a strong foundation in biomedical informatics concepts and healthcare information systems Developing competencies in biomedical data collection, management, and preprocessing techniques Enabling students to apply machine learning and AI methods for healthcare data analysis and decision-making Enhancing practical skills in model evaluation, interpretation, and deployment in real-world healthcare scenarios Fostering awareness of ethical, legal, and societal aspects in the design and use of AI-based healthcare solutions
Course Out Comes	On successful completion of this course, the students shall be able to: CO1: Explain the scope, concepts, and significance of Bio Medical Informatics, including healthcare data types, systems, and standards. (L2 – Understand) CO2: Apply appropriate methods for biomedical data collection, integration, and management while ensuring data privacy and security. (L3 – Apply) CO3: Perform data preprocessing and feature engineering techniques on healthcare datasets to prepare them for analysis. (L3 – Apply) CO4: Develop and evaluate machine learning models for healthcare applications using suitable performance metrics and interpretation techniques. (L4 – Analyze) CO5: Analyze and evaluate AI-based healthcare solutions with respect to design effectiveness, deployment strategies, and ethical, legal, and societal considerations. (L4 – Analyze)
Course Content:	

Module 1	Introduction to Bio Medical Informatics	Assignment : Program activity	6 Hours
Fundamentals, scope, and importance ,Types of biomedical data (structured, semi-structured, unstructured) ,Healthcare information systems (EHR) ,Standards: ICD, SNOMED CT, HL7 ,Role of AI in healthcare (diagnostics, workflows)			
Module 2	**Biomedical Data Collection & Management**	Assignment: Program activity	6 Hours s
Data sources: clinical trials, IoT sensors, surveys ,Data acquisition techniques ,Data storage and integration ,Handling heterogeneous healthcare data ,Privacy, security, and de-identification			
Module 3	**Data Preprocessing & Feature Engineering**	Assignment : Program activity	6 Hours
Data cleaning and transformation ,Handling missing values ,Feature selection and extraction ,Preprocessing for signals, images, and text ,Tools and techniques for data preparation			
Module 4	**Machine Learning in Healthcare**	Assignment: Program activity	6 Hours
Supervised learning: Logistic Regression, Decision Trees, SVM ,Unsupervised learning: ClusteringModel training and validation ,Performance metrics: Accuracy, Sensitivity, Specificity, AUC ,Model interpretability: SHAP, LIME			
Module 5	**Deployment, ethics in AI for healthcare, and case studies**	Assignment: Program activity	6 Hours
Model deployment: Flask, Streamlit ,Clinical Decision Support Systems (CDSS) ,Case studies in healthcare AI ,Personalized medicine, genomics, epidemiology ,Ethical issues: bias, fairness, accountability			
List of Lab Tasks: Lab Sheet 1: Introduction to healthcare datasets using Python; Data visualization using matplotlib and seaborn Lab Sheet 2: Data cleaning techniques; Feature selection methods for health data Lab Sheet 3: Logistic regression for disease prediction; Evaluate with confusion matrix Lab Sheet 4: Decision trees and random forests; Cross-validation comparisons			

Lab Sheet 5: Clustering with K-means; Hierarchical clustering Lab Sheet 6: Time series forecasting with ARIMA; Anomaly detection Lab Sheet 7: Genomic sequence preprocessing; Feature extraction Lab Sheet 8: Medical image preprocessing; CNN classification (e.g., chest X-rays) Lab Sheet 9: NLP on clinical notes; Named Entity Recognition Lab Sheet 10: SHAP/LIME interpretability; Deployment with Flask/Streamlit Lab Sheet 11: Dashboard design; Integration with cloud or mobile apps Lab Sheet 12: Bias detection and mitigation Lab Sheet 13: COVID-19 case study; Real-time data visualization
Targeted Application & Tools that can be used Python, scikit-learn, pandas, matplotlib, seaborn, Jupyter, TensorFlow/PyTorch, Streamlit
Project work/Assignment:
Assignment: Assignments include module-wise exercises and real-world project implementation.
Text Book T1: Adam Bohr & Kaveh Memarzadeh – Artificial Intelligence in Healthcare, Academic Press, 2020 T2: Kevin Franks – Machine Learning for Healthcare, Apress, 2022
References R1: Recent journal articles from IEEE, Nature, and PubMed R2: Online materials from NPTEL, CourseraWeb resources:
Topics relevant to development of “Skill Development”:

Health informatics, AI modeling, data analytics Topics relevant to development of "Environment and sustainability: Public health data analysis, epidemiology

Course Code: CAI3421	Course Name: Intelligent System for Disease Prediction and Drug Discovery Type of Course: Theory Integrated Lab	L- T-P- C	2-0-2-3
Version No.	1.0		
NCrF Level-	5.5		
Course Pre-requisites	CAI2500		
Co-requisites	Nil		
Anti-requisites	Nil		
Course Description	This course explores the application of intelligent systems for disease prediction and drug discovery. It covers knowledge-based systems, healthcare data acquisition and preprocessing, machine learning models for disease prediction, and computational techniques for drug discovery. The course also addresses clinical integration, ethical considerations, and real-world applications of AI in healthcare. Emphasis is placed on hands-on learning using modern tools to develop, evaluate, and deploy intelligent healthcare solutions.		
Course Objective	This course aims to equip students with the knowledge and practical skills to apply intelligent systems in disease prediction and drug discovery. It focuses on developing competencies in healthcare data processing, machine learning and deep learning model development, and computational drug discovery techniques. The course also emphasizes ethical practices, clinical integration, and the use of AI-driven tools to enhance healthcare decision-making and accelerate drug development through experiential learning.		
Course Outcomes	On completion of this course, students will be able to: **CO1:** Explain the fundamental concepts of intelligent systems and their applications in healthcare and disease prediction. (Level 2 – Understand) **CO2:** Apply data acquisition, preprocessing, and feature engineering techniques to clinical and molecular datasets. (Level 3 – Apply) **CO3:** Develop and evaluate machine learning and deep learning models for disease prediction and risk analysis. (Level 3 – Apply) **CO4:** Apply computational techniques such as QSAR modeling, molecular docking, and AI-based methods for drug discovery. (Level 3 – Apply) **CO5:** Analyze and implement intelligent healthcare solutions by integrating AI models into clinical workflows while considering ethical and regulatory aspects. (Level 4 – Analyze)		
Course Content: **I**			
Module 1	**Fundamentals of Intelligent Systems in Healthcare**	Assessment : Assignment	6 Sessions
Intelligent systems in healthcare, Knowledge-based & expert systems ,Clinical decision support systems ,Rule-based diagnosis ,Introduction to Explainable AI (XAI)			

Module 2	**Healthcare Data Acquisition and Preprocessing**	Assessment : Assignment	6 Sessions
Clinical, genomic, and molecular data sources ,Data cleaning and preprocessing Feature engineering and selection ,Handling bias and fairness in healthcare datasets			
Module 3	**Machine Learning for Disease Prediction**	Assessment : Assignment	6 Sessions
Classification models (LR, DT, RF) , Risk prediction models , Model evaluation (ROC, CV, etc.) Deep learning basics for disease prediction			
Module 4	**AI in Computational Drug Discovery**	Assessment : Assignment	6 Sessions
QSAR modeling , Molecular docking , Ligand & structure-based approaches , Neural networks for drug activity , Generative AI (GANs, VAEs) , Ethical issues in drug discovery			
Module 5	**Clinical Integration and Advanced Applications**	Assessment : Assignment	6 Sessions
AI in precision medicine , Clinical workflow integration ,Validation& regulatory frameworks Real-time decision support ,Case studies in healthcare & drug discovery			

List of laboratory tasks:

Lab Sheet 1: Introduction to Healthcare Data & Molecular Datasets

- Level 1: Introduction to healthcare, clinical, genomic, and molecular datasets.
- Level 2: Data loading, exploration, and basic statistics using Python (pandas, NumPy).

Lab Sheet 2: Data Preprocessing & Feature Engineering

- Level 1: Handling missing values, data normalization, and encoding categorical variables.
- Level 2: Feature selection techniques for clinical, genomic, and drug-related datasets (e.g., correlation-based, mutual information).

Lab Sheet 3: Classical Machine Learning for Disease Prediction

- Level 1: Implementation of classification models: Logistic Regression, Decision Trees, Random Forest.
- Level 2: Model evaluation using confusion matrix, ROC-AUC, and cross-validation.

Lab Sheet 4: Deep Learning for Disease Prediction

- Level 1: Development of neural network models using TensorFlow/PyTorch for disease classification.
- Level 2: Hyperparameter tuning, performance comparison, and introduction to transfer learning for medical image or sequence data.

Lab Sheet 5: Molecular Representations & Similarity Analysis

- Level 1: SMILES notation, molecular fingerprints, and graph-based representations.
- Level 2: Compound similarity calculation, clustering, and visualization of chemical space.

Lab Sheet 6: QSAR Modeling& Activity Prediction

- Level 1: Building QSAR models using linear regression and tree-based methods.
- Level 2: Interpretation of chemical descriptors, model performance evaluation, and applicability domain assessment.

Lab Sheet 7: Structure-Based Drug Discovery & Molecular Docking

- Level 1: Introduction to protein-ligand interactions, structure retrieval, and molecular docking principles.
- Level 2: Running docking simulations with open-source tools (AutoDock, PyRx) and interpreting docking scores.

Lab Sheet 8: Expert Systems for Disease Diagnosis

- Level 1: Development of a basic rule-based expert system for symptom-based disease prediction.
- Level 2: Simulation of knowledge base, inference engine, and decision-making process.

Lab Sheet 9: Data Visualization & Interpretability in AI Models

- Level 1: Visualization using seaborn, matplotlib, and plotly for clinical and drug data.
- Level 2: Building interactive dashboards with Streamlit to visualize AI model predictions and explanations.

Lab Sheet 10: Ethical AI & Fairness in Healthcare Applications

- Level 1: Case study analysis on ethical challenges in AI for healthcare.
- Level 2: Design and evaluation of fairness-aware AI models to address bias in disease prediction or drug discovery.

Lab Sheet 11: Natural Language Processing for Drug Discovery (Contemporary Addition)

- Level 1: Introduction to text mining from biomedical literature using spaCy or NLTK.
- Level 2: Extraction of drug-disease relationships from PubMed abstracts.

Lab Sheet 12–15: Capstone Mini-Project (Real-World Problem Solving)
Includes:

- Problem definition based on healthcare or drug discovery challenge.
- Data acquisition, cleaning, and preprocessing.
- Model development (classical ML, deep learning, or expert system).
- Evaluation, result interpretation, and technical report writing.

Targeted Applications & Tools:

Tools/Software/Platform	Application
Python, scikit-learn, pandas, NumPy	Data preprocessing, machine learning model development, clinical and molecular data analysis
TensorFlow, PyTorch	Deep learning for disease prediction, neural network implementation for healthcare applications
Jupyter Notebook	Interactive development environment for AI model building and experimentation
RDKit, Open Babel	Molecular structure representation, compound similarity calculation, cheminformatics tasks
AutoDock, PyRx	Molecular docking simulations for structure-based drug discovery
Streamlit, seaborn, matplotlib, plotly	Data visualization, creation of interactive dashboards for healthcare model interpretation
spaCy, NLTK	Text mining and relationship extraction from biomedical literature (Contemporary Addition)

Skill-Building Activities through Experiential Learning:

- **Disease Prediction Model Development**
  Participants will collect clinical or molecular datasets, preprocess the data using Python libraries, and implement machine learning models like Decision Trees or Logistic Regression to predict disease outcomes. The exercise emphasizes model evaluation using confusion matrices and ROC-AUC, fostering understanding of real-world AI applications in healthcare.

- **Molecular Docking for Drug Discovery**
  This task involves selecting target proteins and candidate drug molecules, preparing molecular structures using RDKit, and performing docking simulations with AutoDock. Participants will analyze docking results to assess binding affinity, providing practical exposure to computational drug discovery workflows.

- **AI Dashboard for Healthcare Insights**
  Participants will use Streamlit or matplotlib to design an interactive dashboard that visualizes disease prediction results or drug discovery outputs. The task develops skills in data interpretation, communication of AI-driven insights, and building user-friendly healthcare analytics interfaces.

Text Book

1. Adam Bohr & Kaveh Memarzadeh – Artificial Intelligence in Healthcare, Academic Press, 2020
2. Kevin Franks – Machine Learning for Healthcare, Apress, 2022

References

- Krittanawong, C., Johnson, K.W., Rosenson, R.S., et al. Deep learning for cardiovascular medicine: A practical primer. European Heart Journal, 2020.
- Ekins, S., Puhl, A.C., Zorn, K.M., et al. Exploiting machine learning for end-to-end drug discovery and development. Nature Materials, 2019.
- Topol, E. Deep Medicine: How Artificial Intelligence Can Make Healthcare Human Again. Basic Books, 2019.

E-Resources/Web Resources

- NPTEL Course – https://nptel.ac.in/courses/103106681

- Coursera Specialization – AI in Healthcare Specialization, Stanford University, https://www.coursera.org/specializations/ai-healthcare
- https://www.leewayhertz.com/generative-ai-in-drug-discovery/

Course Code: CAI3422	Course Name: **AI for Medical Imaging** Type of Course: Theory Integrated Lab	L- T-P- C	2-0-2-3
Version No.	1.0		
NCrF Level-	**5.5**		
Course Pre-requisites	ISE2504		
Co-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course explores the application of artificial intelligence in medical imaging, focusing on preprocessing, deep learning models, multi-modal analysis, explainability, and deployment. It emphasizes practical skills through real datasets and tools, enabling the development of diagnostic AI systems for disease detection, segmentation, and clinical integration in healthcare environments.		
Course Objective	The objective of the course is SKILL DEVELOPMENT of students using EXPERIENTIAL LEARNING techniques		
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Explain medical imaging modalities, image acquisition principles, file formats, and pre-processing techniques used in AI-based medical image analysis. (Level 2 – Understand) **CO2:** Apply deep learning models such as CNNs, transfer learning architectures, and segmentation techniques for classification, detection, and segmentation tasks in medical imaging. (Level 3 - Apply) **CO3:** Apply advanced techniques including multi-modal fusion, 3D volumetric analysis, and handling imbalanced datasets for improved medical image modelling. (Level 3 - Apply) **CO4:** Develop and deploy AI-based diagnostic systems using web frameworks, cloud/edge inference, and containerized pipelines for healthcare applications. (Level 4 - Analyse) **CO5:** Analyze model performance, interpretability, robustness, and ethical implications of AI systems using evaluation metrics, explainable AI techniques, and validation strategies. (Level 4 - Analyse)		
Course Content:			
Module 1	**Foundations of Medical Imaging and Preprocessing**	Assessment : Assignment	10 hours (5L + 5P)

Topics:

- Imaging modalities: X-ray, CT, MRI, Ultrasound, PET
- DICOM, NIfTI file formats and metadata
- Physics of image acquisition
- Preprocessing: denoising, normalization, histogram equalization
- Data augmentation and quality improvement

Module 2	**Deep Learning Architectures for Medical Imaging**	Assessment : Assignment	15 hours (7L + 8P)

Topics:

- CNNs, transfer learning with VGG, ResNet, EfficientNet
- Semantic segmentation (U-Net, Mask R-CNN)
- Object detection (YOLOv11, Faster R-CNN)
- Multi-class disease classification (e.g., chest X-rays)
- Fine-tuning models on medical datasets

Module 3	**Multi-modal and 3D Imaging Techniques**	Assessment : Practical Assignment	14 hours (7L + 7P)

Topics:

- Fusion of multiple imaging types (e.g., CT + PET)
- Working with 3D volumetric data using SimpleITK, nibabel
- Slice-wise segmentation and volume reconstruction
- Transformers in medical imaging (e.g., Vision Transformers)
- Handling imbalance in medical datasets

Module 4	**Deployment, Optimization, and Clinical Integration**	Assessment : Capstone Project	10 hours (5L + 5P)

Topics:

- Web-based diagnostic tools using Streamlit or Flask
- Cloud-based inference using TensorFlow Serving or TorchServe
- Edge AI for mobile health diagnostics
- Deployment pipelines with Docker and APIs
- Evaluating clinical utility and post-deployment monitoring

Module 5	**Model Validation, Ethics & Reliability in Medical AI**	Assessment : Capstone Project	11 hours (6L + 5P)

Topics:

- Bias and fairness in medical datasets
- Cross-validation and external validation in healthcare
- Model robustness and generalization
- Uncertainty estimation (Monte Carlo dropout, Bayesian approaches)
- Regulatory and ethical considerations (clinical safety, transparency)
- Data privacy and security in medical AI systems

List of laboratory tasks:

Lab Sheet 1: Medical Image Loading and Preprocessing

- **Level 1**: Load and visualize various medical image formats (DICOM, NIfTI, PNG, JPEG) using Python libraries such as pydicom, nibabel, and OpenCV.
- **Level 2**: Apply preprocessing techniques like histogram equalization, denoising (Gaussian/median

filtering), CLAHE, and contrast stretching. Compare effects visually and statistically.

Lab Sheet 2: Annotation and Segmentation Preparation

- **Level 1**: Annotate X-ray/CT/MRI images using LabelImg, CVAT, or VIA; export annotations in YOLO/COCO format.
- **Level 2**: Extract Regions of Interest (ROI), generate binary masks, and perform morphological operations. Analyze annotation quality using overlap metrics.

Lab Sheet 3: CNN-Based Classification

- **Level 1**: Build and train a basic CNN for binary classification (e.g., COVID-19 vs. Normal) using Keras or PyTorch.
- **Level 2**: Fine-tune a pre-trained model (VGG16, ResNet50, DenseNet121) on a curated medical dataset. Implement learning rate scheduling and early stopping.

Lab Sheet 4: Semantic Segmentation and Evaluation

- **Level 1**: Implement U-Net and perform semantic segmentation on medical images (e.g., lung masks, tumors).
- **Level 2**: Evaluate predictions using Dice coefficient, Jaccard Index (IoU), and Hausdorff distance. Visualize overlap between prediction and ground truth.

Lab Sheet 5: Multi-Class Classification and Performance Metrics

- **Level 1**: Train a multi-class classifier on annotated chest X-rays (Normal, Pneumonia, COVID-19).
- **Level 2**: Evaluate using confusion matrix, sensitivity, specificity, precision, recall, F1-score, and ROC-AUC curves. Plot class-wise performance.

Lab Sheet 6: Explainable AI in Medical Diagnosis

- **Level 1**: Generate Grad-CAM, LIME, and SHAP heatmaps for classification models.
- **Level 2**: Interpret visualizations to explain AI decisions and detect failure cases. Relate visual explanations to clinical signs and symptoms.

Lab Sheet 7: 3D Imaging and Volumetric Analysis

- **Level 1**: Visualize 3D CT/MRI data using SimpleITK, VTK, or Plotly.
- **Level 2**: Extract slices, resample volumes, and perform slice-wise segmentation. Convert formats (NIfTI ↔ PNG) and display MPR (Multi-Planar Reconstruction) views.

Lab Sheet 8: Deployment and User Interface Development

- **Level 1**: Build a basic Streamlit or Flask interface for uploading images and displaying classification/segmentation results.
- **Level 2**: Integrate model predictions and Grad-CAM explanations into the interface. Simulate a radiologist-friendly diagnostic tool.

Lab Sheet 9: Model Robustness and Clinical Use Case Study

- **Level 1**: Perform a comparative study of models with vs. without data augmentation (e.g., rotation, zoom, brightness).
- **Level 2**: Implement an end-to-end workflow for a clinical use case like breast cancer detection, diabetic retinopathy grading, or tuberculosis classification.

Lab Sheet 10: Capstone Project – AI-Driven Imaging Pipeline

- **Objective**: Develop an end-to-end application incorporating loading, preprocessing, training, explainability, and deployment.
- **Examples**:
  - Brain tumor segmentation and 3D visualization
  - COVID-19 detection from chest X-rays with Grad-CAM
  - Retina disease classification using fundus images
- **Output**: Working prototype + deployment + short technical report

Lab Sheet 11: Model Validation, Robustness and Ethical Analysis

- **Level 1:**
  - Perform k-fold cross-validation on a medical imaging dataset
  - Evaluate model robustness under noise, blur, and intensity variations
  - Compare performance across different patient subgroups (if available)
- **Level 2**:
  - Implement uncertainty estimation (e.g., Monte Carlo dropout)
  - Analyze bias and fairness in predictions
  - Conduct a mini case study on ethical risks and model limitations
  - Prepare a short report on reliability and clinical readiness

Targeted Applications & Tools:

Tools / Software/ Platform	**Application**
pydicom, OpenCV, Matplotlib	Load and Visualize DICOM Chest X-rays
OpenCV, PIL, SimpleITK, Seaborn	Enhance CT Scan Images
LabelImg, CVAT, VIA	Annotate Lung Regions for Segmentation
Keras, TensorFlow, PyTorch, NIH Chest X-ray Dataset	Build a CNN for COVID-19 Classification
PyTorch, TensorFlow, BraTS, OpenCV, nibabel	Fine-tune Pretrained Model on BraTS Dataset
Keras, TensorFlow, U-Net, SimpleITK, BraTS	Perform Semantic Segmentation with U-Net
Scikit-learn, Seaborn, Matplotlib	Evaluate Model Performance on Multi-class Chest X-ray Dataset
TensorFlow, PyTorch, Grad-CAM, OpenCV	Explain Model Decisions Using Grad-CAM
SimpleITK, nibabel, VTK, Matplotlib	3D Visualization of MRI Scans
Streamlit, Flask, OpenCV, Grad-CAM, Pydicom	Deploy AI Model for Real-time Prediction in Web Interface

Skill-Building Activities through Experiential Learning:

- **Diagnostic Pipeline for Pneumonia Detection**
  Develop a complete AI pipeline for pneumonia detection from chest X-rays, including preprocessing, model training, evaluation, and explainability using Grad-CAM. The system is deployed via a Streamlit interface for real-time inference.
- **Brain Tumor Segmentation and Performance Analysis**
  Implement a segmentation pipeline for brain tumor detection using MRI data. The activity includes annotation, U-Net model training, and performance evaluation using Dice score and IoU, with comparison across different data augmentation setups.
- **AI-Enabled Breast Cancer Detection Dashboard**

Build a deployable AI dashboard for breast cancer detection using mammography images. The system integrates preprocessing, model training, explainability with Grad-CAM, and ethical analysis, culminating in a functional web interface for result display

Text Book

Adam Bohr & Kaveh Memarzadeh, Artificial Intelligence in Healthcare, Academic Press, 2020.

Deep Learning for Medical Image Analysis. (2023). United Kingdom: Elsevier Science.

References

M. A. Haidekker, Medical Imaging Technology, Springer, 2013.

Geert Litjens et al., A survey on deep learning in medical image analysis, Medical Image Analysis, Elsevier, 2017.

E-Resources/Web Resources

- https://www.coursera.org/learn/ai-for-medical-diagnosis – AI for Medical Diagnosis by DeepLearning.AI
- https://cs231n.stanford.edu/ – CS231n: Convolutional Neural Networks for Visual Recognition, Stanford University
- https://onlinecourses.nptel.ac.in/noc22_bt34/preview
- https://www.kaggle.com/datasets – Public datasets for medical image classification and segmentation
- https://grand-challenge.org/ – AI challenges and annotated datasets for medical imaging research

Course Code: CAI3423	Course Name: **Genomic Data Science** Type of Course: Theory Integrated Lab	L- T-P- C	2-0-2-3
Version No.	1.0		
NCrF Level-	**5.5**		
Course Pre-requisites	MAT1212		
Co-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course explores modern techniques in genomic data analysis, including sequencing, variant annotation, gene expression profiling, and machine learning. It emphasizes practical skills through real-world datasets, enabling the development of end-to-end AI-driven genomic pipelines for personalized medicine, diagnostics, and predictive modeling in healthcare, agriculture, and environmental sustainability		

<table>
<tr><td>Course Objective</td><td colspan="3">The objective of the course is SKILL DEVELOPMENT of students using EXPERIENTIAL LEARNING techniques..</td></tr>
<tr><td rowspan="6">Course Outcomes</td><td colspan="3">On successful completion of the course the students shall be able to:</td></tr>
<tr><td colspan="2">CO1: Explain genome structure, sequencing technologies, genomic data formats, and biological databases used in bioinformatics.</td><td>(Level 2 - Understand)</td></tr>
<tr><td colspan="2">CO2: Apply genomic data preprocessing techniques including quality control, alignment, variant calling, and annotation using standard bioinformatics tools.</td><td>(Level 3 - Apply)</td></tr>
<tr><td colspan="2">CO3: Apply machine learning techniques for classification, clustering, and feature selection in genomic and transcriptomic datasets.</td><td>(Level 3 - Apply)</td></tr>
<tr><td colspan="2">CO4: Apply genomic analysis methods to real-world applications such as cancer genomics, pharmacogenomics, and precision medicine.</td><td>(Level 3 - Apply)</td></tr>
<tr><td colspan="2">CO5: Analyze multi-omics data and evaluate deep learning models for biological interpretation, integrating ethical considerations and data privacy aspects.</td><td>(Level 4 - Analyse)</td></tr>
<tr><td colspan="4">Course Content:</td></tr>
<tr><td>Module 1</td><td>Foundations of Genomics and Data Ecosystems</td><td>Assessment : Assignment</td><td>12 Hours(6 L+6P)</td></tr>
<tr><td colspan="4">Topics:
• Genome structure, epigenome, transcriptome, and microbiome overview
• Genome sequencing: Sanger, NGS, and long-read technologies (PacBio, Oxford Nanopore)
• Introduction to genomic file formats (FASTA, FASTQ, VCF, GTF)
• Major repositories: NCBI, Ensembl, 1000 Genomes, TCGA, GEO</td></tr>
<tr><td>Module 2</td><td>Genomic Data Preprocessing & Functional Annotation</td><td>Assessment : Assignment</td><td>12 Hours(6 L+6P)</td></tr>
<tr><td colspan="4">Topics:
• Quality control (FASTQC), trimming, alignment (BWA/Bowtie2), SAM/BAM conversion
• Variant calling (bcftools, GATK), annotation (SnpEff, VEP), and filtering
• Transcript quantification and normalization (TPM, FPKM, RPKM)
• Functional genomics: gene ontology (GO), KEGG, Reactome pathways</td></tr>
<tr><td>Module 3</td><td>Machine Learning in Genomics Analysis</td><td>Assessment : Assignment</td><td>12 Hours(6 L+6P)</td></tr>
<tr><td colspan="4">Topics:
Machine learning approaches are applied to tasks such as disease susceptibility prediction, biomarker discovery, and patient stratification. Techniques include classification, clustering, and feature selection using scikit-learn and bioinformatics tools like BioPython and Bioconductor.</td></tr>
<tr><td>Module 4</td><td>Applications in Personalized & Predictive Genomics</td><td>Assessment : Case Study</td><td>12 Hours(6 L+6P)</td></tr>
<tr><td colspan="4">Topics:
• Genomics in cancer (somatic mutation profiling), pharmacogenomics, population genetics
• Precision medicine: AI-based diagnosis, treatment planning, polygenic risk scores</td></tr>
</table>

- Emerging domains: nutrigenomics, microbial genomics, sustainable agriculture
- Data privacy and ethical compliance (HIPAA, GDPR, consent protocols)

Module 5	**Multi-Omics Integration & Deep Learning in Genomics**	Assessment : Case Study	12 Hours(6 L+6P)

Topics:

- Multi-omics data integration (genomics, transcriptomics, proteomics, metabolomics)
- Network biology and gene regulatory networks
- Deep learning for genomics (CNNs/RNNs for sequence analysis)
- Graph-based learning in biological networks
- Single-cell genomics and clustering (scRNA-seq basics)
- AI for drug discovery and target identification

List of laboratory tasks:

Lab Sheet 1: Genomic Data Retrieval and Visualization

- Level 1: Retrieve DNA, RNA, and protein sequences using NCBI Entrez Direct, Ensemble REST API, and UCSC Table Browser.
- Level 2: Visualize genes, SNPs, and structural variants using the UCSC Genome Browser and IGV (Integrative Genomics Viewer); generate genome tracks.

Lab Sheet 2: Sequence Alignment and Format Conversion

- Level 1: Perform pairwise and multiple sequence alignment using BLAST and Clustal Omega.
- Level 2: Conduct high-throughput short-read alignment using BWA; convert outputs using SAMtools; inspect reads with IGV.

Lab Sheet 3: Variant Calling and Annotation

- Level 1: Perform variant calling from aligned reads using bcftoolsmpileup and GATK HaplotypeCaller.
- Level 2: Annotate variants using VEP (Variant Effect Predictor) and SnpEff; classify variants based on functional impact.

Lab Sheet 4: Gene Expression Analysis

- Level 1: Preprocess microarray datasets (e.g., background correction, normalization) using R (limma).
- Level 2: Execute full RNA-Seq analysis: quality control with FASTQC, alignment with STAR/HISAT2, quantification with HTSeq/featureCounts.

Lab Sheet 5: Feature Engineering and Dimensionality Reduction

- Level 1: Encode categorical and continuous genomic features using One-Hot Encoding, Z-score normalization.
- Level 2: Apply PCA and t-SNE to visualize expression patterns; analyze gene clusters using elbow method and silhouette scores.

Lab Sheet 6: Supervised Learning on Genomic Data

- Level 1: Build classification models (Logistic Regression, Decision Tree, SVM) to predict disease risk

using scikit-learn.

- Level 2: Evaluate model performance using confusion matrix, ROC-AUC, precision-recall curves; perform 5-fold cross-validation.

Lab Sheet 7: Clustering and Unsupervised Learning

- Level 1: Apply K-means and hierarchical clustering on expression profiles; determine optimal clusters using elbow and dendrogram methods.
- Level 2: Visualize gene clusters using seaborn heatmaps; interpret biological meaning through GO term enrichment (DAVID/Enrichr).

Lab Sheet 8: Multi-Omics Analysis and Deep Learning in Genomics

- Level 1: Integrate gene expression and mutation datasets, Construct gene co-expression networks using correlation matrices, Visualize networks using Cytoscape / Python (NetworkX)
- Level 2: Implement a simple deep learning model (CNN) for DNA sequence classification (e.g., promoter prediction), Perform clustering on single-cell RNA-seq data (Scanpy/Seurat basics), Analyze feature importance and biological relevance

Targeted Applications & Tools:

Tools/Software/Platform	Application
NCBI Entrez, Ensembl, UCSC Genome Browser	Extract and visualize the BRCA1 gene sequence from the human genome
BLAST, Clustal Omega, FASTQC	Perform multiple sequence alignment to identify conserved regions in homologous genes
BWA, SAMtools, FASTQC	Align short DNA reads from a cancer sample to the reference genome
bcftools, GATK, VEP, SnpEff	Identify and annotate SNPs from aligned sequencing data
DESeq2, edgeR, limma (via R/Bioconductor)	Analyze gene expression changes in cancer vs. normal tissues using RNA-Seq data
BioPython, pyVCF, pandas	Convert annotated genomic variants into a machine learning-ready dataset
scikit-learn, TensorFlow, XGBoost	Build a disease risk prediction model using expression data
seaborn, pheatmap, matplotlib	Visualize gene expression clusters and correlation heatmaps
Galaxy platform, Streamlit, Flask	Create a full workflow for genomic analysis via an intuitive web interface
Jupyter Notebook, RStudio, Google Colab	Perform data wrangling and model training in a reproducible notebook environment
NCBI dbSNP, Ensembl, custom scripts via Biopython	Compare variant frequencies across populations using 1000 Genomes dataset
EMBL-EBI Tools, DAVID, Enrichr	Enrich gene lists with functional annotations and pathways

Skill-Building Activities through Experiential Learning:

- **Genome Browsing and Data Retrieval**
  Learners retrieve specific gene sequences using NCBI and Ensembl tools and visualize their locations and regulatory elements via UCSC Genome Browser. They explore gene structure variations across species using EnsemblCompara, building familiarity with genomic databases essential for research and clinical interpretation.

- **Machine Learning Applications in Genomics**
  Students apply machine learning techniques such as Decision Trees and SVM to classify disease types using gene expression data. They use clustering methods to identify molecular subtypes and perform feature selection for biomarker discovery, gaining hands-on exposure to AI applications in genomics research.

- **Development of Predictive Models Using Omics Data** Participants build and validate predictive models using omics datasets like gene expression or mutation profiles. They apply dimensionality reduction techniques such as PCA and t-SNE for visualizing complex patterns. Model evaluation using ROC-AUC and precision-recall metrics enhances practical skills for genomic AI development

Text Book

- Michael C. Schatz et al., Genomic Data Science, Cold Spring Harbor Lab Press, 2022
- Jason H. Moore & Scott M. Williams, Bioinformatics for Geneticists, Wiley, 2020
- Sofi, M. Y., Shafi, A., Masoodi, K. Z. (2021). Bioinformatics for Everyone. Netherlands: Academic Press.
- Kim, J. H. (2019). Genome Data Analysis. Germany: Springer Nature Singapore.

References

R. Durbin et al., Biological Sequence Analysis, Cambridge University Press, 1998

Online resources from Coursera (e.g., Genomic Data Science Specialization by Johns Hopkins), NPTEL, and EMBL-EBI

E-Resources/Web Resources

- https://www.ncbi.nlm.nih.gov/ – NCBI Genomics Portal
- https://www.ensembl.org/ – Ensembl Genome Browser
- https://galaxyproject.org/ – Web-based bioinformatics analysis platform
- **https**://bioconductor.org/ – Open software for genomic data analysis

Course Code: CAI3424	Course Name: **Clinical Data Science** Type of Course: Theory Integrated Lab	L- T-P- C	3-0-2-3
Version No.	1.0		
NCrF Level-	**5.5**		
Course Pre-requisites	CAI3439		
Co-requisites	NILL		
Anti-requisites	NILL		

Course Description	This course explores the acquisition, processing, and analysis of clinical data to support healthcare decision-making. It covers electronic health records, coding standards, data integration, predictive modeling, text analytics, and dashboard development, equipping individuals with the skills to build data-driven solutions for clinical risk prediction, reporting, and real-time monitoring		
Course Objective	The objective of the course is SKILL DEVELOPMENT of students using EXPERIENTIAL LEARNING techniques		
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Identify various clinical data sources, formats, and coding systems used in healthcare informatics. (Level 2 - Understand) **CO2:** Apply preprocessing, integration, and cohort selection techniques to prepare structured and unstructured clinical datasets for analysis. (Level 3 - Apply) **CO3:** Apply machine learning and survival analysis techniques to develop predictive models for clinical decision support. (Level 3 - Apply) **CO4:** Develop visualization dashboards and deploy clinical analytics solutions using APIs and healthcare data standards. (Level 3 - Apply) **CO5:** Analyze model performance, fairness, interpretability, and reliability in clinical analytics using appropriate evaluation and validation techniques. (Level 4 - Analyse)		
Course Content:			
Module 1	**Clinical Data Foundations & Standards**	Assessment : Assignment	10 hours (5L + 5P)
Topics: • Introduction to clinical data types: EHR, PACS, LIS, and claims data • Clinical data models: OMOP, HL7, and FHIR • Coding systems and ontologies: ICD-10, SNOMED CT, LOINC, CPT • Formats and access mechanisms: CSV, JSON, XML, HL7 messages • Ethical and legal considerations: HIPAA, GDPR basics			
Module 2	**Clinical Data Wrangling & Integration**	Assessment : Assignment	13 hours (6L + 7P)
Topics: • Structured vs. unstructured data: cleaning and standardization • SQL and pandas for querying hospital datasets • Temporal data alignment and sequencing (e.g., lab tests, vitals) • Merging datasets across systems (EHR + diagnostics + medications) • Cohort creation and patient filtering using clinical criteria			
Module 3	**Predictive Modeling & Risk Stratification**	Assessment : Quiz + Assignment	15 hours (7L + 8P)
Topics: • Supervised ML for patient risk scoring: logistic regression, decision trees • Evaluation metrics: confusion matrix, ROC-AUC, precision-recall • Disease progression and readmission prediction • Introduction to survival analysis: Kaplan-Meier estimator, Cox model • Calibration and interpretation in clinical settings			
Module 4	**Clinical Dashboards & Deployment**	Assessment : Quiz + Assignment	10 hours (5L

			+ 5P)

Topics:

- Visualization of patient outcomes and hospital KPIs
- Building dashboards with Streamlit / Power BI
- Integration with EHR systems using APIs (FHIR, HL7)
- HIPAA/GDPR considerations for clinical deployment
- Case study: real-time clinical monitoring system

Module 5	**Advanced Clinical Analytics, Fairness & Model Reliability**	Assessment : Mini Project	12 hours (7L + 5P)

Topics:

- Bias and fairness in clinical datasets (demographic bias, sampling bias)
- Model explainability in healthcare (SHAP, LIME for tabular data)
- Model calibration and uncertainty estimation
- Handling imbalanced clinical datasets (SMOTE, class weighting)
- Drift detection and continuous monitoring in clinical systems
- Clinical validation and audit trails for decision support systems

List of laboratory tasks:

Lab Sheet 1: Clinical Data Exploration

- **Level 1**: Explore and interpret the structure of synthetic electronic health records (EHRs) using CSV and JSON formats.
- **Level 2**: Extract, filter, and visualize demographic and clinical metadata using SQL queries and pandas DataFrames. Create summary statistics and bar charts for patient distribution by age, gender, and diagnosis.

Lab Sheet 2: Preprocessing Structured and Unstructured Data

- **Level 1**: Handle missing values, detect and treat outliers, and normalize data fields using pandas and NumPy.
- **Level 2**: Process clinical text using NLP (tokenization, stemming, stop word removal) and convert it into a structured format for integration into downstream tasks.

Lab Sheet 3: Data Integration and Cohort Generation

- **Level 1**: Join datasets from laboratory reports, diagnosis codes, and medication records using keys like patient ID and admission ID.
- **Level 2**: Apply temporal logic to extract patient cohorts (e.g., those readmitted within 30 days) using SQL and Python. Create a mini-cohort dashboard for visualization.

Lab Sheet 4: Predictive Modeling for Clinical Decision Support

- **Level 1**: Build baseline models (Logistic Regression, Decision Trees) for binary outcomes such as hospital readmission.
- **Level 2**: Evaluate model performance using metrics like confusion matrix, ROC-AUC, precision-recall curves, and classification reports. Compare models using cross-validation.

Lab Sheet 5: Visualization and Dashboard Development

- **Level 1**: Create interactive visualizations of patient trends (e.g., time-series vitals, discharge outcomes) using matplotlib and seaborn.
- **Level 2**: Develop a real-time dashboard using Streamlit or Power BI to monitor hospital metrics (e.g., occupancy rate, infection rate) with live filters and KPIs.

Lab Sheets 6–8: Capstone Project (End-to-End Pipeline)

- Design and implement a complete clinical analytics pipeline using real or simulated datasets:
  - Data Acquisition & Cleaning
  - Feature Engineering & NLP for Notes
  - Model Development & Evaluation
  - Survival Analysis (optional)
  - Interactive Dashboard for Stakeholder Presentation
- Integrate clinical reasoning with data insights to propose actionable interventions and strategies for care improvement.

Lab Sheet 9-11: Model Explainability, Fairness & Reliability Analysis

- Level 1: Apply class imbalance techniques (SMOTE / resampling) on clinical datasets, Generate feature importance using SHAP or LIME for trained models, Visualize model predictions and feature contributions

- Level 2: Evaluate fairness across subgroups (e.g., gender/age bias analysis), Perform calibration analysis (reliability curves, Brier score), Simulate data drift and analyze model performance degradation, Prepare a short report on clinical reliability and ethical implications

Targeted Applications & Tools:

Tools/Software/Platform	Application
MIMIC-III, eICU, SQL, pandas	Extract ICU patient records for sepsis cases using real-world datasets
HL7, FHIR parsers, Python, csv module	Convert HL7 messages into structured CSV for downstream analysis
pandas, NumPy, Jupyter Notebook	Clean and preprocess clinical data with missing lab values and outliers
matplotlib, seaborn	Create histograms and heatmaps for lab test values across age groups
Streamlit, Plotly	Build a dashboard to monitor patient vitals in real-time
scikit-learn, XGBoost	Train a model to predict in-hospital mortality using clinical risk factors
Lifelines (Python), Kaplan-Meier, CoxPH model	Compare survival time between patients receiving different treatments
spaCy, SciSpacy, NLTK	Tokenize and analyze unstructured clinical notes to extract symptom mentions
SQL, pandas	Join data from demographics, labs, and diagnoses to create patient cohorts
Streamlit, Power BI, Tableau (optional)	Build and deploy a clinical data dashboard for hospital administration
R Studio, survival package	Run a survival analysis report using R for cancer treatment outcome data
Google Colab, matplotlib, seaborn, scikit-learn	Use Colab notebooks for real-time model training and visualizations on large datasets

Skill-Building Activities through Experiential Learning:

- **Clinical Data Handling and Preprocessing**
  Learners explore large-scale clinical datasets such as MIMIC-III using Python and SQL. They clean lab

results and vitals data by addressing missing values and outliers, and structure unstructured clinical notes using regular expressions and basic NLP techniques to prepare datasets for downstream analysis and machine learning applications.

- **SQL and Python for Healthcare Data Integration**
  Participants retrieve and analyze patient cohorts by writing SQL queries based on diagnosis and lab results. They integrate datasets such as medications and demographics using SQL joins and automate data extraction workflows with Python, simulating real-world healthcare data management for clinical analytics and outcome evaluation.

- **Predictive Modeling for Clinical Risk Scoring**
  Students develop logistic regression or decision tree models to predict patient outcomes such as readmission. They assess model performance using confusion matrices and ROC-AUC scores and interpret key features, helping to translate machine learning outputs into actionable insights for clinical decision support and risk stratification.

Text Book

- Mark L. Braunstein, Practitioner's Guide to Health Informatics, Springer, 2015.
- Pradeep Menon, Applied Clinical Informatics: A Practical Guide for Healthcare Professionals, CRC Press, 2021.
- Braunstein, M. L. (2014). Contemporary Health Informatics. United States: AHIMA Press.
- Leveraging Data Science for Global Health. (2020). Germany: Springer International Publishing

References

- J. D. Dalianis, Clinical Text Mining: Secondary Use of Electronic Patient Records, Springer, 2018.
- Steinar Carlsen et al., Health Informatics: An Interprofessional Approach, Elsevier, 2020.
- Online resources including MIMIC-III tutorials (MIT-LCP), NPTEL Health Analytics courses, and Coursera's Data Science in Stratified Healthcare and Precision Medicine

E-Resources/Web Resources

- https://registry.opendata.aws/mimiciii/ – MIMIC-III: Medical Information Mart for Intensive Care
- https://www.hl7.org/ – Health Level Seven International (HL7 standards)
- https://nptel.ac.in/courses/106/106/106106213 – NPTEL: Health Informatics and Analytics
- https://www.coursera.org/learn/clinical-data-science – Coursera: Clinical Data Science Specialization
- https://streamlit.io/ – Streamlit: Rapid development of clinical data dashboards

Course Code: CAI3425	Course Name: AI in Epidemiology and Public Health Analytic Type of Course: Theory Integrated Lab	L- T-P- C	2-0-2-3
Version No.	1.0		
NCrF Level-	5.5		
Course Pre-requisites	MAT1212		
Co-requisites	Nil		

Anti-requisites	Nil		
Course Description	This course examines how AI techniques can be applied to epidemiology and public health for disease modeling, outbreak prediction, and population health analysis. It covers public health data acquisition, geospatial modeling, ethical AI use, and AI-driven surveillance systems to enable informed policymaking and sustainable health interventions.		
Course Objective	The objective of the course is SKILL DEVELOPMENT of students using EXPERIENTIAL LEARNING techniques.		
Course Outcomes	On completion of this course, students will be able to: CO1: Explain epidemiological concepts, public health datasets, and surveillance systems used in population health analysis. (Level 2 - Understand) CO2: Apply data preprocessing, exploratory analysis, and cohort-based techniques to prepare epidemiological datasets for modeling. (Level 3 - Apply) CO3: Apply machine learning, time series forecasting, and compartmental models to analyze disease patterns and predict public health risks. (Level 3 - Apply) CO4: Develop geospatial visualizations and interactive dashboards for monitoring and decision support in public health systems. (Level 3 - Apply) CO5: Analyze public health interventions, model outcomes, and policy impacts using causal inference, explainable AI, and evaluation techniques. (Level 4 - Analyse)		
Course Content: **I**			
Module 1	Principles of Epidemiology and Health Intelligence	Assessment : Assignment	10 hours (5L + 5P)
Epidemiological parameters: incidence, prevalence, morbidity, mortality, Understanding health systems and surveillance networks (e.g., CDC, WHO),Public health datasets: NHANES, NFHS, DHS, GHO, Basics of data quality and health indicators			
Module 2	Public Health Data Wrangling and Exploration	Assessment : Assignment	13 hours (6L + 7P)
Data formats in epidemiology (CSV, JSON, HL7), Data preprocessing: missing values, harmonization, bias detection, Exploratory Data Analysis (EDA) using pandas, seaborn, and statsmodels, Introduction to population stratification and risk profiling			
Module 3	AI for Outbreak Forecasting and Risk Prediction	Assessment : Assignment	16 hours (8L + 8P)
Machine Learning models: Decision Trees, Logistic Regression, XGBoost, Time series forecasting with Prophet and ARIMA, SIR/SEIR disease spread modeling, Vaccination uptake prediction models and population-level simulations			
Module 4	Geospatial Analysis, Public Health Mapping and Dashboards	Assessment : Mini Project	11 hours (5L + 6P)
Introduction to GIS in health analytics, Mapping disease incidence with geopandas, folium, plotly, Hotspot detection and clustering (e.g., DBSCAN), Environmental and socio-demographic overlays on health data, Dashboard development: Streamlit, Tableau, Scenario-based decision support systems, Case studies: COVID-19 surveillance, vaccination equity dashboards			
Module 5	Public Health Decision Analytics & Policy Evaluation	Assessment : Project	10 hours(6L + 4P)

Causal inference in public health (correlation vs causation), Impact evaluation methods (A/B testing, quasi-experiments, difference-in-differences), Health policy modeling and intervention analysis, Cost-effectiveness and resource allocation strategies, Explainable AI (XAI) in public health decision-making, Ethical considerations and bias in population-level AI systems

List of laboratory tasks:

Lab Sheet 1: Public Health Data Exploration

- Level 1: Load large-scale epidemiological datasets such as NHANES, DHS, or NFHS using pandas or APIs (e.g., WHO GHO). Parse metadata, check data integrity, and understand variable descriptions.
- Level 2: Conduct descriptive analysis—compute prevalence, incidence rates, and central tendencies (mean, median, mode). Visualize trends by demographics using box plots and histograms.

Lab Output: Clean dataset summary, disease-wise distribution by region and age.

Lab Sheet 2: Data Cleaning and Demographic Cohort Analysis

- Level 1: Use pandas and missingno for identifying and handling missing data. Normalize and recode categorical variables (e.g., income levels, region codes).
- Level 2: Perform group-wise aggregations based on age, gender, income, and urban-rural location. Calculate cohort-based disease burden and visualize with stacked bar graphs.

Lab Output: Cleaned cohort datasets with summary tables and insights for segmentation.

Lab Sheet 3: Geospatial Disease Visualization

- Level 1: Use geopandas, folium, or plotly to visualize disease distribution geographically. Overlay disease incidence on base maps.
- Level 2: Create choropleth maps for regional health indicators (e.g., maternal mortality, malaria) and embed filters for year, region, and age group using plotly.express or Dash.

Lab Output: Interactive disease maps embedded with filters and legends.

Lab Sheet 4: Predictive Modeling for Health Risk Classification

- Level 1: Train a classification model (Decision Tree, Logistic Regression, or Random Forest) to identify high-risk individuals (e.g., diabetes, hypertension).
- Level 2: Use scikit-learn to evaluate model performance via ROC-AUC, confusion matrix, precision-recall curves, and feature importance visualization.

Lab Output: Model report with evaluation plots and health risk insights.

Lab Sheet 5: Disease Forecasting with Time Series Analysis

- Level 1: Apply ARIMA and/or Prophet models to forecast weekly or monthly case trends (e.g., influenza, dengue).
- Level 2: Compare predicted vs. actual using RMSE, MAE. Visualize forecast intervals, anomaly windows, and trend seasonality decomposition.

Lab Output: Forecast report with visualized trends and model diagnostics.

Lab Sheet 6: Epidemic Simulation with Compartmental Models

- Level 1: Implement a basic SIR or SEIR model in Python using differential equations to simulate spread over time.
- Level 2: Fit model to historical outbreak data (e.g., COVID-19), calibrate parameters (beta, gamma) using curve_fit or optimization, and validate outcomes.

Lab Output: Simulation plots and parameter tuning dashboard.

Lab Sheet 7: Public Health Dashboard Development

- Level 1: Use Streamlit to build a basic interactive dashboard showing disease statistics, cohort summaries, or trends.
- Level 2: Integrate multiple modules: maps, charts (via plotly or matplotlib), filters (age, gender, region), and real-time updates using file watchers or APIs.

Lab Output: Deployable Streamlit dashboard with user-friendly UI and embedded analytics.

Lab Sheets 8–10: Capstone Project – AI-Powered Public Health System

- Objective: Design a multi-component system addressing real-world public health needs (e.g., outbreak detection, vaccination coverage, environmental health surveillance).
- Requirements:
  - Use historical datasets (e.g., COVID-19, polio, dengue).
  - Incorporate forecasting, risk classification, and geospatial mapping.
  - Build a decision-support dashboard for public health policymakers or field officers.
  - Include explainable AI (XAI) elements to justify predictions and alerts.

Lab Sheet 11: Policy Impact Analysis and Decision Support

- Level 1:Analyze pre- and post-intervention datasets (e.g., vaccination campaigns), Perform statistical comparison (t-test / trend comparison), Visualize intervention impact over time
- Level 2:Implement a simple causal analysis (difference-in-differences), Evaluate cost-effectiveness of interventions using simulated data, Integrate insights into a mini decision-support dashboard (Streamlit), Interpret results for policymaker recommendations
- Output:
  Policy evaluation report + interactive dashboard with actionable insights

Targeted Applications & Tools:	
Tools/Software/Platform	Application
Python, pandas, matplotlib	Extract and visualize trends in mortality and morbidity from NHANES/WHO datasets
pandas, seaborn, statsmodels	Compare health indicators (e.g., stunting, child mortality) across the NFHS and DHS datasets
geopandas, plotly, folium	Perform GIS-based mapping of disease prevalence using public datasets
pandas, seaborn, statsmodels	Segment public health records by socioeconomic or demographic factors for cohort analysis
scikit-learn, XGBoost, pandas	Build a machine learning model to forecast diabetes or obesity risk using DHS data
Prophet, statsmodels	Predict immunization coverage trends using time series modeling
Streamlit, pandas, matplotlib	Create a Streamlit-based dashboard summarizing maternal and child health indicators for a selected region
scikit-learn (fairness metrics), audit datasets	Conduct an ethical audit of a trained AI model to identify potential biases in public health predictions
Skill-Building Activities through Experiential Learning:	
Epidemiological Data Cleaning & Summarization	

Clean and preprocess public health datasets like NHANES or NFHS by handling missing values and formatting variables. Perform statistical summarization and visualizations to understand patterns in health indicators such as morbidity or mortality, enabling data-driven insights for population health trends.

Machine Learning for Disease Prediction & Forecasting
Apply machine learning models like decision trees and logistic regression for disease risk prediction. Use time-series forecasting tools like ARIMA or Prophet to model outbreak trends. Evaluate performance using accuracy metrics to guide timely intervention and preparedness in healthcare systems.

Geospatial Mapping & Hotspot Analysis
Use GeoPandas and Folium to create choropleth maps and visualize health trends spatially. Perform hotspot analysis to identify clusters of disease burden or health risks. Correlate environmental and demographic factors with epidemiological data for targeted health policy development.

Text Book

1. Michael J. Paul & Mark Dredze, Social Monitoring for Public Health, Morgan & Claypool, 2017.
2. R. Bonita, R. Beaglehole & T. Kjellström, Basic Epidemiology, 2nd Edition, WHO Press, 2006.
3. Bovbjerg, M. L. (2020). Foundations of Epidemiology. United States: Oregon State Open Educational Resources.

References

- David L. Streiner& Geoffrey R. Norman, Health Measurement Scales: A Practical Guide to Their Development and Use, Oxford University Press, 2015.
- Online resources including CDC WONDER database, WHO Health Data Platform, and training modules from Johns Hopkins' Coursera series on Public Health Data Science
- Tutorials and documentation for SIR/SEIR modeling, GIS-based health data visualization, and epidemiological surveillance dashboards

E-Resources/Web Resources

- https://www.who.int/data – WHO Global Health Observatory (GHO)
- https://www.cdc.gov/datastatistics – CDC Data & Statistics
- https://www.coursera.org/specializations/public-health-data-science – Coursera: Public Health Data Science Specialization
- https://ourworldindata.org/coronavirus – COVID-19 data and public health analysis tools
- https://www.healthdata.org/ – Institute for Health Metrics and Evaluation (IHME)

Course Code: CAI3425	Course Name: AI in Epidemiology and Public Health Analytic Type of Course: Integrated	L- T-P- C	2-0-2-3
Version No.	1.0		
NCrF Level-	5.5		
Course Pre-requisites	MAT1212		
Co-requisites	Nil		

Anti-requisites	Nil		
Course Description	This course examines how AI techniques can be applied to epidemiology and public health for disease modeling, outbreak prediction, and population health analysis. It covers public health data acquisition, geospatial modeling, ethical AI use, and AI-driven surveillance systems to enable informed policymaking and sustainable health interventions.		
Course Objective	The objective of the course is SKILL DEVELOPMENT of students using EXPERIENTIAL LEARNING techniques.		
Course Outcomes	On completion of this course, students will be able to: CO1: Explain epidemiological concepts, public health datasets, and surveillance systems used in population health analysis. (Level 2 - Understand) CO2: Apply data preprocessing, exploratory analysis, and cohort-based techniques to prepare epidemiological datasets for modeling. (Level 3 - Apply) CO3: Apply machine learning, time series forecasting, and compartmental models to analyze disease patterns and predict public health risks. (Level 3 - Apply) CO4: Develop geospatial visualizations and interactive dashboards for monitoring and decision support in public health systems. (Level 3 - Apply) CO5: Analyze public health interventions, model outcomes, and policy impacts using causal inference, explainable AI, and evaluation techniques. (Level 4 - Analyse)		
Course Content: **I**			
Module 1	Principles of Epidemiology and Health Intelligence	Assessment : Assignment	10 hours (5L + 5P)
Epidemiological parameters: incidence, prevalence, morbidity, mortality, Understanding health systems and surveillance networks (e.g., CDC, WHO),Public health datasets: NHANES, NFHS, DHS, GHO, Basics of data quality and health indicators			
Module 2	Public Health Data Wrangling and Exploration	Assessment : Assignment	13 hours (6L + 7P)
Data formats in epidemiology (CSV, JSON, HL7), Data preprocessing: missing values, harmonization, bias detection, Exploratory Data Analysis (EDA) using pandas, seaborn, and statsmodels, Introduction to population stratification and risk profiling			
Module 3	AI for Outbreak Forecasting and Risk Prediction	Assessment : Assignment	16 hours (8L + 8P)
Machine Learning models: Decision Trees, Logistic Regression, XGBoost, Time series forecasting with Prophet and ARIMA, SIR/SEIR disease spread modeling, Vaccination uptake prediction models and population-level simulations			
Module 4	Geospatial Analysis, Public Health Mapping and Dashboards	Assessment : Mini Project	11 hours (5L + 6P)
Introduction to GIS in health analytics, Mapping disease incidence with geopandas, folium, plotly, Hotspot detection and clustering (e.g., DBSCAN), Environmental and socio-demographic overlays on health data, Dashboard development: Streamlit, Tableau, Scenario-based decision support systems, Case studies: COVID-19 surveillance, vaccination equity dashboards			
Module 5	Public Health Decision Analytics & Policy Evaluation	Assessment : Project	10 hours(6L + 4P)

Causal inference in public health (correlation vs causation), Impact evaluation methods (A/B testing, quasi-experiments, difference-in-differences), Health policy modeling and intervention analysis, Cost-effectiveness and resource allocation strategies, Explainable AI (XAI) in public health decision-making, Ethical considerations and bias in population-level AI systems
List of laboratory tasks: Lab Sheet 1: Public Health Data Exploration • Level 1: Load large-scale epidemiological datasets such as NHANES, DHS, or NFHS using pandas or APIs (e.g., WHO GHO). Parse metadata, check data integrity, and understand variable descriptions. • Level 2: Conduct descriptive analysis—compute prevalence, incidence rates, and central tendencies (mean, median, mode). Visualize trends by demographics using box plots and histograms. Lab Output: Clean dataset summary, disease-wise distribution by region and age. Lab Sheet 2: Data Cleaning and Demographic Cohort Analysis • Level 1: Use pandas and missingno for identifying and handling missing data. Normalize and recode categorical variables (e.g., income levels, region codes). • Level 2: Perform group-wise aggregations based on age, gender, income, and urban-rural location. Calculate cohort-based disease burden and visualize with stacked bar graphs. Lab Output: Cleaned cohort datasets with summary tables and insights for segmentation. Lab Sheet 3: Geospatial Disease Visualization • Level 1: Use geopandas, folium, or plotly to visualize disease distribution geographically. Overlay disease incidence on base maps. • Level 2: Create choropleth maps for regional health indicators (e.g., maternal mortality, malaria) and embed filters for year, region, and age group using plotly.express or Dash. Lab Output: Interactive disease maps embedded with filters and legends. Lab Sheet 4: Predictive Modeling for Health Risk Classification • Level 1: Train a classification model (Decision Tree, Logistic Regression, or Random Forest) to identify high-risk individuals (e.g., diabetes, hypertension). • Level 2: Use scikit-learn to evaluate model performance via ROC-AUC, confusion matrix, precision-recall curves, and feature importance visualization. Lab Output: Model report with evaluation plots and health risk insights. Lab Sheet 5: Disease Forecasting with Time Series Analysis • Level 1: Apply ARIMA and/or Prophet models to forecast weekly or monthly case trends (e.g., influenza, dengue). • Level 2: Compare predicted vs. actual using RMSE, MAE. Visualize forecast intervals, anomaly windows, and trend seasonality decomposition. Lab Output: Forecast report with visualized trends and model diagnostics. Lab Sheet 6: Epidemic Simulation with Compartmental Models • Level 1: Implement a basic SIR or SEIR model in Python using differential equations to simulate spread over time. • Level 2: Fit model to historical outbreak data (e.g., COVID-19), calibrate parameters (beta, gamma) using curve_fit or optimization, and validate outcomes. Lab Output: Simulation plots and parameter tuning dashboard.

Lab Sheet 7: Public Health Dashboard Development

- Level 1: Use Streamlit to build a basic interactive dashboard showing disease statistics, cohort summaries, or trends.
- Level 2: Integrate multiple modules: maps, charts (via plotly or matplotlib), filters (age, gender, region), and real-time updates using file watchers or APIs.

Lab Output: Deployable Streamlit dashboard with user-friendly UI and embedded analytics.

Lab Sheets 8–10: Capstone Project – AI-Powered Public Health System

- Objective: Design a multi-component system addressing real-world public health needs (e.g., outbreak detection, vaccination coverage, environmental health surveillance).
- Requirements:
  - Use historical datasets (e.g., COVID-19, polio, dengue).
  - Incorporate forecasting, risk classification, and geospatial mapping.
  - Build a decision-support dashboard for public health policymakers or field officers.
  - Include explainable AI (XAI) elements to justify predictions and alerts.

Lab Sheet 11: Policy Impact Analysis and Decision Support

- Level 1:Analyze pre- and post-intervention datasets (e.g., vaccination campaigns), Perform statistical comparison (t-test / trend comparison), Visualize intervention impact over time
- Level 2:Implement a simple causal analysis (difference-in-differences), Evaluate cost-effectiveness of interventions using simulated data, Integrate insights into a mini decision-support dashboard (Streamlit), Interpret results for policymaker recommendations
- Output:
  Policy evaluation report + interactive dashboard with actionable insights

Targeted Applications & Tools:	
Tools/Software/Platform	Application
Python, pandas, matplotlib	Extract and visualize trends in mortality and morbidity from NHANES/WHO datasets
pandas, seaborn, statsmodels	Compare health indicators (e.g., stunting, child mortality) across the NFHS and DHS datasets
geopandas, plotly, folium	Perform GIS-based mapping of disease prevalence using public datasets
pandas, seaborn, statsmodels	Segment public health records by socioeconomic or demographic factors for cohort analysis
scikit-learn, XGBoost, pandas	Build a machine learning model to forecast diabetes or obesity risk using DHS data
Prophet, statsmodels	Predict immunization coverage trends using time series modeling
Streamlit, pandas, matplotlib	Create a Streamlit-based dashboard summarizing maternal and child health indicators for a selected region
scikit-learn (fairness metrics), audit datasets	Conduct an ethical audit of a trained AI model to identify potential biases in public health predictions
Skill-Building Activities through Experiential Learning:	
Epidemiological Data Cleaning & Summarization	

Clean and preprocess public health datasets like NHANES or NFHS by handling missing values and formatting variables. Perform statistical summarization and visualizations to understand patterns in health indicators such as morbidity or mortality, enabling data-driven insights for population health trends.

Machine Learning for Disease Prediction & Forecasting
Apply machine learning models like decision trees and logistic regression for disease risk prediction. Use time-series forecasting tools like ARIMA or Prophet to model outbreak trends. Evaluate performance using accuracy metrics to guide timely intervention and preparedness in healthcare systems.

Geospatial Mapping & Hotspot Analysis
Use GeoPandas and Folium to create choropleth maps and visualize health trends spatially. Perform hotspot analysis to identify clusters of disease burden or health risks. Correlate environmental and demographic factors with epidemiological data for targeted health policy development.

Text Book

1. Michael J. Paul & Mark Dredze, Social Monitoring for Public Health, Morgan & Claypool, 2017.
2. R. Bonita, R. Beaglehole & T. Kjellström, Basic Epidemiology, 2nd Edition, WHO Press, 2006.
3. Bovbjerg, M. L. (2020). Foundations of Epidemiology. United States: Oregon State Open Educational Resources.

References

- David L. Streiner& Geoffrey R. Norman, Health Measurement Scales: A Practical Guide to Their Development and Use, Oxford University Press, 2015.
- Online resources including CDC WONDER database, WHO Health Data Platform, and training modules from Johns Hopkins' Coursera series on Public Health Data Science
- Tutorials and documentation for SIR/SEIR modeling, GIS-based health data visualization, and epidemiological surveillance dashboards

E-Resources/Web Resources

- https://www.who.int/data – WHO Global Health Observatory (GHO)
- https://www.cdc.gov/datastatistics – CDC Data & Statistics
- https://www.coursera.org/specializations/public-health-data-science – Coursera: Public Health Data Science Specialization
- https://ourworldindata.org/coronavirus – COVID-19 data and public health analysis tools
- https://www.healthdata.org/ – Institute for Health Metrics and Evaluation (IHME)

Course Code: CAI3426	Course Name: Time Series Analysis for Patient Monitoring Type of Course: Theory Integrated Lab	L- T-P- C	2-0-2-3
Version No.	1.0		
NCrF Level-	5.5		
Course Pre-requisites	MAT1212		
Co-requisites	Nil		

Anti-requisites	Nil		
Course Description	This course explores advanced methods for analyzing time series data from patient monitoring systems, including ICU sensors and wearable devices. It covers data preprocessing, forecasting, anomaly detection, and real-time health analytics using AI models, equipping professionals to build intelligent, ethical, and deployable clinical decision support and monitoring solutions.		
Course Objective	To provide students with fundamental and advanced techniques for analyzing healthcare time series data and enable them to develop forecasting, anomaly detection, and real-time monitoring solutions for patient care.		
Course Outcomes	On completion of this course, students will be able to: CO1: Describe the structure and clinical significance of time series data in patient monitoring. (Level 2 - Understand) CO2: Describe the components of healthcare time series and summarize trends using visualization techniques. (Level 2 - Understand) CO3: Apply preprocessing methods to handle noise, missing data, and irregular sampling in time series. (Level 3 - Apply) CO4: Analyze forecasting and anomaly detection models for predicting patient conditions. (Level 4 - Analyze) CO5: Analyze monitoring interfaces for real-time alerts using time-series insights. (Level 4 - Analyze)		
Course Content: **I**			
Module 1	Fundamentals of Patient Time Series and Monitoring Systems	Assessment : Assignment	6 + 6 sessions
Nature and structure of time series in healthcare: ECG, BP, SpO_2, respiration, etc., Components of time series: trend, seasonality, cyclicity, irregularity, Introduction to ICU monitoring systems and wearable biosensors, Visualization and interpretation using matplotlib, seaborn, and plotly			
Module 2	Data Preprocessing and Feature Engineering	Assessment : Assignment	6 + 6 sessions
Handling missing data, noise reduction, signal smoothing, resampling, Temporal alignment and dealing with irregular time intervals, Feature extraction: rolling statistics, HRV, signal windowing, Exploratory analysis using pandas, NumPy, and statsmodels			
Module 3	Forecasting and Anomaly Detection for Vital Signs	Assessment : Mini Project	8 + 6 sessions
Forecasting models: ARIMA, SARIMA, Holt-Winters, Prophet, Deep learning for prediction: LSTM, GRU, Autoencoders, Anomaly detection techniques: Z-score, Isolation Forest, One-Class SVM, Model evaluation: MAE, RMSE, precision-recall in clinical settings			
Module 4	Real-Time Systems, Edge Monitoring and Visualization dashboards	Assessment : Project	4 + 6 sessions
Real-time simulation from stored sensor data using MQTT/Flask, Building lightweight applications for ICU monitoring using Streamlit, Alert triggers and clinical threshold detection, Integration with wearable/IoT devices for remote care, Visualization dashboards for clinicians using Streamlit/Dash.			
Module 5	Advanced Healthcare Analytics and Ethics	Assessment: Assignment / Mini Project	6 + 6 Sessions
Integration of AI models in clinical decision support systems , Data privacy, security, and ethical considerations in healthcare analytics , Case studies in healthcare time series (ICU monitoring, wearable devices) ,Regulatory and deployment challenges in healthcare AI			

List of laboratory tasks:

Lab Sheet 1: Introduction to Healthcare Time Series
Level 1: Load real-world patient time series datasets (e.g., ECG, heart rate, respiratory rate from PhysioNet) and explore structure, frequency, and units.
Level 2: Visualize time series using line plots, moving averages, and dual-axis charts; decompose into trend, seasonality, and residuals using additive/multiplicative models.

Lab Sheet 2: Preprocessing and Signal Cleaning
Level 1: Handle missing data using linear, spline, and time-based interpolation; compare methods for different signals.
Level 2: Normalize and denoise physiological data using rolling mean, Gaussian smoothing, and median filters; resample irregular data to uniform intervals.

Lab Sheet 3: Feature Engineering for Patient Monitoring
Level 1: Extract time-domain features (mean, standard deviation, HRV) and compare across patient conditions.
Level 2: Segment signals using sliding and expanding windows; prepare structured inputs for ML models with lagged features and rolling statistics.

Lab Sheet 4: Classical Forecasting Techniques
Level 1: Fit ARIMA, SARIMA, and Holt-Winters models to predict short-term trends in vitals (e.g., temperature, BP).
Level 2: Evaluate model performance using MAE, RMSE, AIC, and residual analysis; visualize forecast confidence intervals.

Lab Sheet 5: Deep Learning for Signal Forecasting
Level 1: Build and train LSTM/GRU models for multi-step prediction of vital signs; preprocess input sequences and design network layers.
Level 2: Apply hyperparameter tuning (batch size, epochs, learning rate); visualize loss curves and forecast output.

Lab Sheet 6: Anomaly Detection in Critical Care Data
Level 1: Detect outliers using Z-score, Hampel filters, and Isolation Forest; apply to signals like SpO_2 and blood pressure.
Level 2: Compare anomaly detection accuracy and precision; annotate and visualize anomalies on plots for interpretability.

Lab Sheet 7: Real-Time Data Simulation and Alerting
Level 1: Simulate a streaming data pipeline using stored ICU data in real time (e.g., simulate per-second input).
Level 2: Configure alerting mechanisms using threshold breaches (e.g., HR > 120 bpm) with Streamlit or Flask dashboard.

Lab Sheets 8–10: Project
Project Scope: Design and implement a complete patient monitoring pipeline:
Data ingestion from real-world sources

Signal preprocessing and feature extraction
Forecasting and anomaly detection
Real-time alerting and dashboard reporting

Targeted Applications & Tools:

Tools/Software/Platform	Application
Python, pandas, matplotlib, seaborn, MIMIC-III, PhysioNet, Jupyter Notebook	Load and visualize real-time ICU sensor data (e.g., ECG, heart rate)
NumPy, pandas, statsmodels, pmdarima, openICPSR data, Google Colab	Preprocess and clean noisy physiological time series with missing values
statsmodels, pmdarima, matplotlib, MIMIC-III	Develop ARIMA/SARIMA models to forecast patient blood pressure trends
Prophet, pandas, seaborn, MAE/RMSE evaluation metrics	Predict respiratory rate using Prophet and evaluate model accuracy
TensorFlow, Keras, NumPy, pandas, VS Code	Train LSTM/GRU models on multivariate vitals for future signal prediction
scikit-learn, NumPy, statsmodels, matplotlib	Detect anomalies in patient vitals using Isolation Forest and Z-score analysis
MQTT, Flask, pandas, Google Colab (simulated streaming), threshold-based alerts	Create a real-time streaming simulator for ICU vitals with alert mechanism
Streamlit, Plotly, Dash, matplotlib, seaborn	Build a clinical dashboard for visualizing patient trends and anomalies
pandas, geopandas, seaborn, MIMIC-III, external JSON/CSV-based patient meta-data	Integrate and overlay vital signal data with demographic profiles for analysis
statsmodels, TensorFlow, Keras, matplotlib, RMSE/MAE plots	Compare forecasting accuracy of ARIMA vs. LSTM on glucose level predictions

Skill-Building Activities through Experiential Learning:

Preprocessing and Analyzing Time Series Data from Patient Monitoring Systems

Learners will work with real-world ICU datasets such as MIMIC-III to clean, resample, and preprocess vital signs data like heart rate and ECG signals. Techniques include handling missing values, filtering noise, and decomposing time series into trends and seasonal components to derive meaningful clinical insights.

Building Forecasting Models Using ARIMA, LSTM, and Hybrid Techniques

Participants will apply ARIMA, SARIMA, and LSTM models for forecasting critical health parameters like blood pressure or respiratory rate. They will also explore hybrid models such as Prophet-ARIMA, evaluate forecasting accuracy using metrics like MAE and RMSE, and compare model performance for short-term patient condition prediction.

Developing Interactive Dashboards and Alerting Systems for Clinical Decision Support

Participants will design interactive dashboards using tools like Streamlit and Plotly to visualize patient vital signs, trends, and forecasted values. They will incorporate real-time anomaly detection and user-friendly visual alerts to assist healthcare professionals in timely clinical decision-making based on continuous time series data.

Text Book 1. Aileen Nielsen, *Practical Time Series Analysis*, O'Reilly, 2019 2. Paolo Emiliozzi, *Time Series Forecasting in Python*, Leanpub, 2021 3. Bob Mather, *Time Series with Python*, 2020
References Hyndman & Athanasopoulos, *Forecasting: Principles and Practice*, 3rd Edition, OTexts PhysioNet Datasets: https://physionet.org/ NPTEL and Coursera courses on Time Series and Healthcare Analytics
E-Resources/Web Resources https://physionet.org/ https://otexts.com/fpp3/ https://www.coursera.org/learn/time-series https://github.com/awslabs/gluon-ts https://tensorflow.org/tutorials/structured_data/time_series

Course Code: CAI3427	Course Name: Language Models for Text Mining Type of Course:Theory Integrated Lab	L- T-P- C	2-0-2-3
Version No.	1.0		
NCrF Level-	5.5		
Course Pre-requisites	CAI2504		
Co-requisites	NIL		
Anti-requisites	CSEXXXX		
Course Description	This course introduces text mining and natural language processing using statistical and neural techniques. It covers preprocessing, text representation, language modeling, and deep learning methods. Emphasis is placed on practical applications such as classification, topic modeling, and sequence labeling using modern tools like Keras for hands-on NLP system development.		
Course Objective	To equip students with fundamental and advanced concepts of text mining and natural language processing, and enable them to apply statistical and deep learning techniques to analyze textual data through experiential learning.		
Course Outcomes	**On successful completion of the course the students shall be able to:** CO1: Explain the fundamental concepts of text mining and natural language processing. [Level 2 – Understand] CO2: Illustrate the techniques of text preprocessing, including tokenization, stemming, and encoding. [Level 2 – Understand] CO3: Apply text representation models such as Bag-of-Words, TF-IDF, and N-Grams for analytical tasks. [Level 3 – Apply] CO4: Analyze machine learning-based classification and topic modeling techniques for text datasets. [Level 4 – Analyze]		

	CO5: Analyze the performance of advanced NLP models, including transformer-based approaches, using appropriate evaluation metrics [Level 4 – Analyze]		
Course Content:			
Module 1	**Foundations of Text Mining and NLP**	Assessment: Adversarial Quiz, Module Test	7 + 6 sessions
Topics: Overview of text mining and its relationship with NLP Text mining pipeline: extraction, preprocessing, analysis, evaluation Corpus creation and lexical resources Basics of string manipulation and cleaning techniques Introduction to sequential data and sequence labeling Viterbi algorithm and Hidden Markov Models (HMMs) Handling unknown words in sequence models			
Module 2	**Text Preprocessing Techniques**	Assessment: Adversarial Quiz, Module Test	6 + 6 Sessions
Topics: Tokenization, stop-word removal, lemmatization, and stemming Part-of-Speech (PoS) tagging Integer encoding, padding, and one-hot encoding Preprocessing workflows for structured text input			
Module 3	**Text Representation and Modeling**	Assessment: Adversarial Quiz, Module Test	7 + 6 Sessions
Topics: Statistical language modeling: N-Grams, Bag-of-Words Term-Document Matrix, TF, IDF, and TF-IDF Cosine similarity for document comparison Naïve Bayes classifier for text classification Introduction to topic modeling: LSA, SVD, Truncated SVD Latent Dirichlet Allocation (LDA) for topic discovery			
Module 4	**Neural NLP with Keras**	Assessment: Adversarial Quiz, Module Test	4 + 6 Sessions
Topics: Word embeddings: CBOW, Skip-gram Comparison of one-hot vs. dense embeddings Deep learning models for document classification Using Keras for building and training NLP models			
Module 5	**Advanced NLP Techniques and Applications**	Adversarial Quiz, Module Test	6 + 6 Sessions
Introduction to Transformer-based models (BERT, GPT – overview) Attention mechanisms in NLP Fine-tuning pre-trained language models Text generation and summarization			
List of laboratory tasks:			
• **Experiment No. 1: File Handling and Text Extraction**			

Level 1: Read and extract meaningful content from raw text, CSV, and JSON files using Python.
Level 2: Parse structured and unstructured files, clean data using regular expressions, and prepare the text for NLP tasks like classification or translation.

- **Experiment No. 2: NLP Toolkit Familiarization**
  Level 1: Install and explore basic text processing with NLTK—tokenization, frequency analysis, and stop-word removal.
  Level 2: Perform advanced tasks using SpaCy—PoS tagging, Named Entity Recognition (NER), and syntactic dependency visualization on real-world text.

- **Experiment No. 3: Corpus Cleaning and Preprocessing**
  Level 1: Implement end-to-end cleaning pipelines—tokenization, lemmatization, and noise removal with NLTK/SpaCy.
  Level 2: Prepare cleaned datasets suitable for downstream machine learning tasks and visualize token distributions and word clouds.

- **Experiment No. 4: Word Embeddings and Semantic Similarity**
  Level 1: Download and explore pre-trained word vectors (Word2Vec, GloVe, FastText) for word similarity tasks.
  Level 2: Visualize word embeddings using dimensionality reduction (e.g., t-SNE) and perform analogy tasks like king - man + woman = queen.

- **Experiment No. 5 & 6: Language Identification and Multilingual Analysis**
  Level 1: Build a basic language detector using Bag-of-Words features and Scikit-learn.
  Level 2: Extend the language identifier to support multiple languages using character-level features and evaluate performance on noisy text.

- **Experiment No. 7 & 8: Lexical Simplification for Readability Enhancement**
  Level 1: Implement a rule-based lexical simplifier replacing complex terms with simpler synonyms.
  Level 2: Use word embeddings or transformer models (e.g., BERT) for context-aware lexical simplification preserving semantic meaning.

- **Experiment No. 9 & 10: Sentiment Analysis and Comparison of Approaches**
  Level 1: Implement sentiment classifiers using rule-based approaches (VADER, TextBlob) on social media or review data.
  Level 2: Build and compare ML-based or transformer-based sentiment models, evaluating with accuracy, F1-score, and confusion matrix.

- **Experiment No. 11: Named Entity Recognition (NER) with Classical and Neural Approaches**
  Level 1: Perform basic NER using NLTK for person, location, and organization detection.
  Level 2: Use SpaCy or transformer-based models for robust NER on large text datasets and compare extraction quality.

- **Experiment No. 12 & 13: Hidden Markov Models (HMM) for Sequence Modeling**
  Level 1: Implement a generic HMM for basic sequence prediction tasks.
  Level 2: Apply HMM for PoS tagging or NER and evaluate model performance using real-world tagged corpora.

- **Experiment No. 14: Linguistic HMM for Structured Prediction**
  Level 1: Design an HMM specifically for PoS tagging using annotated corpora.
  Level 2: Evaluate tagging accuracy, compare with baseline models, and discuss limitations of HMMs versus modern sequence models (e.g., Bi-LSTM).

- **Experiment No. 15: Machine Translation with Modern Pre-trained Models**
  Level 1: Use Hugging Face Transformers to implement Machine Translation on English-Indian language pairs.
  Level 2: Evaluate translation quality using BLEU scores and Round-Trip Translation. Explore translation errors and suggest improvements.

Targeted Applications & Tools:

Tools/Software/Platform	Application
Python (Anaconda / Jupyter Notebook)	General NLP development, text processing, and experimentation
Google Colab	Cloud-based environment for running NLP experiments and deep learning models
NLTK (Natural Language Toolkit)	Text preprocessing, tokenization, stop-word removal, PoS tagging, and Named Entity Recognition
SpaCy	Advanced NLP tasks such as tokenization, PoS tagging, NER, and dependency parsing
Scikit-learn	Machine learning models for text classification and topic modeling
Gensim	Implementation of topic modeling techniques like LDA and text similarity analysis
Keras with TensorFlow backend	Development and training of deep learning models for document classification and embeddings
Hugging Face Transformers	Access to pre-trained models for advanced NLP tasks like machine translation
Matplotlib / Seaborn	Visualization of text data, model outputs, and analysis results

Skill-Building Activities through Experiential Learning:

- **Text Preprocessing with Real Datasets**
  Learners will perform comprehensive text cleaning on real-world datasets such as tweets or product reviews. Activities include tokenization, stop-word removal, stemming, and lemmatization using NLTK and SpaCy. The focus is on preparing high-quality input for downstream NLP tasks, ensuring noise reduction and consistency in text data for effective analysis.

- **Topic Modeling using Gensim LDA**
  Participants will apply Latent Dirichlet Allocation (LDA) with Gensim to discover hidden topics in a large corpus. Through preprocessing, dictionary creation, and model training, learners will explore dominant themes in news articles or research papers. The activity enhances understanding of unsupervised learning and interprets real-world text datasets effectively.

- **Document Classification with Deep Learning**
  Students will build document classification models using Keras with pre-trained word embeddings. The activity involves preparing labeled text data, constructing deep learning models, and evaluating performance. Learners gain practical experience in applying neural approaches to classify reviews, news articles, or sentiment data, emphasizing end-to-end NLP system development.

Text Book

T1: Daniel Jurafsky, James H. Martin. "Speech and Language Processing: An Introduction to Natural Language Processing", Computational Linguistics and Speech, Pearson Publication, 2025 (3rd Edition Draft).
T2: Aditya Joshi, Pushpak Bhattacharyya. "Natural Language Processing", Wiley Publication, 2023 (1st Edition).
T3: Vajjala, S., Majumder, B., Gupta, A., Surana, H. (2020). Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems. United States: O'Reilly Media.

References

R1: D. Manning, C., Schütze, H. (2016). Foundations of Statistical Natural Language Processing. (n.p.): CreateSpace Independent Publishing Platform.
R2: Pawan Goyal. "Natural Language Processing". 1st Edition, 2016

E-Resources/Web Resources

1. E-Book: https://drive.google.com/file/d/10nbwAJd-dv6htOOZVBgAvLd1WscI0RqC/view
2. Speech and Language Processing (Jurafsky& Martin): https://web.stanford.edu/~jurafsky/slp3/
3. NPTEL Courses:
   https://nptel.ac.in/courses/106106211
   https://nptel.ac.in/courses/106105158
   https://nptel.ac.in/courses/106101007
   https://nptel.ac.in/courses/106105572

Course Code: CAI3428	Course Name: **Practical Deep Learning with TensorFlow** Type of Course: Theory Integrated Lab	L- T-P- C	2-0-2-3
Version No.	1.0		
NCrF Level-	5.5		
Course Pre-requisites	CSS2264		

Co-requisites	Nil		
Anti-requisites	Nil		
Course Description	This course provides hands-on experience in building deep learning models using TensorFlow and Keras. It covers foundational neural network concepts, backpropagation, and model optimization. Focus areas include image classification, sequence modelling, and transfer learning, emphasizing practical implementation of deep learning solutions in real-world contexts through structured modules and tool-based exploration		
Course Objective	To provide students with fundamental and practical knowledge of deep learning and enable them to design, implement, and evaluate deep learning models using TensorFlow and Keras.		
Course Outcomes	On successful completion of this course, the students shall be able to: **CO1:** Explain the basic principles of neural networks, including perceptron, backpropagation, and gradient descent. (Level 2 - Understand) **CO2:** Describe the architecture, components, and functionalities of TensorFlow and Keras for deep learning applications. (Level 2 - Understand) **CO3:** Apply deep learning techniques to build and train models for tasks such as image classification and sequence modeling. (Level 3 - Apply) **CO4:** Analyze convolutional and recurrent neural network models implemented using TensorFlow and Keras for real-world datasets. (Level 4 - Analyze) **CO5:** Analyze transfer learning approaches to fine-tune pre-trained models for domain-specific deep learning tasks. (Level 4 - Analyze)		
Course Content:			
Module 1	**Foundations of Neural Networks**	Assessment: Quiz / Conceptual Case Study etc	6 + 6 Sessions
Understanding Perceptron with Excel, Understanding Multilayer Perceptron with Excel, From Multilayer Perceptron to Deep Learning, Error Backpropagation and Gradient Descent to reduce errors, Activation Functions, Deep Learning, Problems with Deep Learning with solutions.			
Module 2	**Getting Started with TensorFlow**	Assessment: Quiz / Conceptual Case Study etc	6 + 6 Sessions
TensorFlow fundamentals and architecture, Working with TensorFlow datasets and pipelines, Defining, compiling, and training basic models, Tensor operations and data flow graphs, Visualization with TensorBoard			
Module 3	**Deep Learning Architectures using TensorFlow and Keras**	Assessment: Quiz / Conceptual Case Study etc	6 + 6 Sessions
Overview of Keras and its integration with TensorFlow, designing sequential and functional API models, Building models for image classification (CNNs), Recurrent neural networks (RNNs) for sequence modelling, Transfer learning and model fine-tuning.			
Module 4	Advanced Deep Learning Architectures	Assessment: Quiz / Conceptual Case Study etc	6 + 6 Sessions

Convolutional Neural Networks (CNNs) , Recurrent Neural Networks (RNNs, LSTM, GRU)

Model evaluation techniques ,Hyperparameter tuning

Module 5	Transfer Learning and Applications	Assessment: Quiz / Conceptual Case Study etc	6 + 6 Sessions

Transfer learning concepts, Fine-tuning pre-trained models, Real-world applications (image classification, NLP tasks), Model optimization and deployment basics

List of laboratory tasks:

Lab 1: Exploring Deep Learning Frameworks
Objective: Familiarize with popular deep learning frameworks and their capabilities.
Tasks: Compare features of Keras, TensorFlow, and PyTorch for model development.
Activity: Install frameworks, explore official documentation, run simple demo models, and summarize differences in approach, syntax, and ecosystem.

Lab 2: Building a Basic Artificial Neural Network (ANN)
Objective: Design and train a simple ANN for binary classification.
Tasks: Use Keras with TensorFlow to create an ANN for predicting diabetes using the Pima Indians dataset.
Activity: Implement data preprocessing, model building, training, and basic evaluation using accuracy and a confusion matrix.

Lab 3: Multilayer Perceptron (MLP) for Regression
Objective: Develop an MLP for predicting continuous values.
Tasks: Build and fine-tune an MLP for predicting house prices using a real-world dataset (e.g., the Boston Housing dataset).
Activity: Train the model, visualize training curves, and compare performance with/without feature scaling.

Lab 4: Tensor Operations in TensorFlow
Objective: Understand tensor creation and manipulation using lists and NumPy arrays.
Tasks: Create tensors from different sources, apply reshaping and slicing operations.
Activity: Demonstrate creation, reshaping, and slicing of tensors using TensorFlow functions with practical examples.

Lab 5: Mathematical Operations on Tensors
Objective: Perform arithmetic and advanced mathematical operations on tensors.
Tasks: Apply element-wise operations like addition, multiplication, square, square root, and exponentials.
Activity: Visualize operations and explore broadcasting in TensorFlow for operations on mismatched tensor shapes.

Lab 6: Combining and Manipulating Tensors
Objective: Explore tensor concatenation, stacking, and splitting.
Tasks: Combine multiple tensors along different axes and split tensors for parallel processing.
Activity: Implement concatenation, stacking, and splitting operations to prepare data for neural network inputs.

Lab 7: Loading and Processing Datasets from Local Files
Objective: Build a dataset pipeline from CSV files using TensorFlow.
Tasks: Load real-world datasets (e.g., Titanic dataset), preprocess them, and convert to TensorFlow Dataset objects.
Activity: Perform feature scaling, batching, and shuffling to prepare data for model training.

Lab 8: Working with TensorFlow Datasets (TFDS)
Objective: Utilize pre-loaded datasets from TFDS for experimentation.
Tasks: Load datasets like MNIST or CIFAR-10, split into training/testing sets, and preprocess them.
Activity: Visualize samples, normalize images, and prepare batches for model training.

Lab 9: Image Classification with Convolutional Neural Networks (CNN)
Objective: Develop and evaluate a CNN for image classification tasks.
Tasks: Build a CNN to classify images from the Dogs vs. Cats dataset or Fashion-MNIST.
Activity: Train the model, plot learning curves, evaluate performance, and visualize feature maps.

Lab 10: Sequence Modeling with RNN and LSTM
Objective: Build models for sequential data using RNNs and LSTMs.
Tasks: Develop a sentiment analysis model for movie reviews using the IMDB dataset with RNN/LSTM layers.
Activity: Preprocess text data, implement the model, evaluate with accuracy and loss curves, and experiment with GRU layers.

Lab 11: Transfer Learning for Image Classification
Objective: Apply transfer learning to fine-tune pre-trained models.
Tasks: Use a pre-trained model like MobileNet or VGG16 for classifying custom image datasets.
Activity: Freeze base layers, add new classification head, fine-tune the model, and compare results with baseline models.

Targeted Applications & Tools:

Tools/Software/Platform	**Application**
Python (Anaconda / Jupyter Notebook)	General deep learning development, model building, experimentation
TensorFlow	Building and training deep learning models for real-world tasks
Keras	High-level API for designing neural network architectures
Google Colab	Cloud-based platform for running TensorFlow and Keras experiments without local setup
Matplotlib / Seaborn	Visualization of model performance, loss curves, and data distributions
TensorBoard	Visualizing model architecture, training metrics, and debugging deep learning workflows
Pandas and NumPy	Data manipulation, preprocessing, and tensor creation
Scikit-learn	Preprocessing pipelines and basic machine learning comparison with deep learning models
Hugging Face Datasets (Optional)	Access to curated datasets for experimentation and model evaluation

Skill-Building Activities through Experiential Learning:

- **Visualizing Neural Networks with TensorBoard**
  Learners will build simple neural network models using Keras and visualize their architecture, loss curves, and training metrics using TensorBoard. This activity helps develop intuition about model behavior during training, identifying issues like overfitting or vanishing gradients through real-time interactive visualizations.
- **Image Classification with Transfer Learning**
  Participants will fine-tune a pre-trained Convolutional Neural Network (e.g., MobileNet or VGG16)

using a small, custom dataset for image classification tasks. This activity provides practical exposure to leveraging transfer learning for improving model performance with minimal data, a crucial real-world deep learning application.

- **Sentiment Analysis using RNNs**
  Learners will implement a Recurrent Neural Network with LSTM layers to perform sentiment analysis on movie reviews from the IMDB dataset. This task introduces sequence modeling for text data, enabling participants to understand how RNNs capture temporal dependencies for tasks like language understanding.

Text Book
T1: François Chollet, *Deep Learning with Python*, 2nd Edition, Manning, 2022
T2: Goodfellow, Bengio, Courville, *Deep Learning*, MIT Press, 2016
T3: Anjanava Biswas, Wrick Talukdar, *Building Agentic AI Systems*, Packt, 2025

References
R1: Amlan Chakrabarti Amit Kumar Das, Saptarsi Goswami, Pabitra Mitra , "Deep Learning", Pearson Publication, 2021.
R2: David Foster, "Generative Deep Learning" O'Reilly Publishers, 2020.
R3: John D Kellehar, "Deep Learning", MIT Press, 2020.

E-Resources/Web Resources

IEEE Transactions on Neural Networks:
https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=5962385
IEEE Transactions on Pattern Analysis:
https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=34
International Journal of Intelligent Systems: https://onlinelibrary.wiley.com/journal/1098111x
NPTEL – Deep Learning (IIT Ropar): https://onlinecourses.nptel.ac.in/noc21_cs35/preview
Coursera – Neural Networks and Deep Learning (Andrew Ng)
Coursera – Neural Networks for Machine Learning (Geoffrey Hinton)

Course Code: CAI3429	Course Name: Deep Learning for Computer Vision Type of Course: Theory Integrated Lab	L- T-P- C	2-0-2-3
Version No.	1.0		
NCrF Level-	5.5		
Course Pre-requisites	CAI2500		
Co-requisites	Nil		
Anti-requisites	Nil		
Course Description	This course explores deep learning techniques for solving computer vision problems, including image classification, object detection, segmentation, and generative modeling. Emphasis is placed on practical implementation using TensorFlow and PyTorch, with applications in real-world scenarios and deployment on edge devices, highlighting model robustness, explainability, and contemporary vision architectures.		
Course Objective	To provide students with fundamental and advanced concepts of deep learning for computer vision and enable them to design, implement, and evaluate vision-based		

	models using modern frameworks.		
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Explain key components of deep learning models used in computer vision. (Level 2 – Understand) **CO2:** Describe the working of CNNs, object detection, and segmentation techniques. (Level 2 – Understand) **CO3:** Apply Vision Transformers and GANs for basic image processing tasks. (Level 3 – Apply) **CO4:** Analyze the use of pre-trained models for deployment on mobile or embedded devices. (Level 4– Analyze) **CO5:** Analyze model behavior, robustness, and explainability in computer vision applications. (Level 4 – Analyze)		
Course Content: **I**			
Module 1	**Foundations of Deep Learning for Vision**	**Assessment:** Assignment	6 + 6 Sessions
Deep learning vs traditional image processing, Architecture of CNNs: convolution, pooling, fully connected layers, Backpropagation, activation functions, and optimizers, Transfer learning with pretrained models (ResNet, VGG, EfficientNet)			
Module 2	**Object Detection and Image Segmentation**	**Assessment:** Assignment	8 + 6 Sessions
Object Detection: R-CNN, SSD, YOLOv5, YOLOv8, and Faster R-CNN, Region Proposal Networks (RPNs), Semantic and instance segmentation: U-Net, Mask R-CNN, Datasets: COCO, Pascal VOC, custom datasets, Use cases: face detection, autonomous vehicles			
Module 3	**Advanced Techniques for Vision**	**Assessment:** Assignment	6 + 6 Sessions
Vision Transformers (ViTs), Attention Mechanisms, Generative Adversarial Networks (GANs): DCGAN, StyleGAN, Self-supervised Learning: SimCLR, BYOL, Multi-modal Learning: CLIP, DALL·E, Prompt engineering basics for vision tasks			
Module 4	**Deployment and Edge Optimization**	**Assessment:** Assignment	4 + 6 Sessions
Deploying models using TensorFlow Lite, ONNX, and OpenVINO, Adversarial attacks and model robustness, Explainability: Grad-CAM, SHAP, Edge/embedded vision applications: mobile apps, Raspberry Pi, Jetson Nano, Industrial case studies: surveillance, retail, healthcare			
Module 5	**Advanced Vision Applications and Case Studies**	**Assessment:** Assignment	6 + 6 Sessions
Real-time computer vision applications (surveillance, healthcare, retail) , OCR and document analysis using deep learning ,Gesture and face recognition systems , Case studies of deployed vision systems			
List of Laboratory Tasks:			
Lab Sheet 1: Neural Network Models with Keras APIs **Objective:** Build and compare models using Sequential and Functional APIs. • Load and explore a dataset (e.g., Fashion-MNIST). • Build a basic MLP model using Keras Sequential API. • Configure and apply optimizers, loss functions, and callbacks. • Build a custom model with skip connections using Functional API. • Train both models and evaluate their performance metrics.			

- Compare the learning curves and confusion matrices.

Lab Sheet 2: Softmax Regression and Model Evaluation

Objective: Apply softmax regression for multi-class classification.

- Load and preprocess multi-class image data.
- Design and train a softmax regression model using Keras.
- Plot accuracy, precision, and recall per class.
- Use confusion matrix and classification report for evaluation.
- Apply dropout and observe impact on generalization.

Lab Sheet 3: CNNs for Grayscale Image Classification

Objective: Develop a CNN for grayscale image datasets (e.g., MNIST).

- Preprocess images (reshape, normalize).
- Define a CNN with multiple convolution and pooling layers.
- Apply data augmentation using KerasImageDataGenerator.
- Train and evaluate performance using test dataset.
- Visualize learned filters and activation maps.

Lab Sheet 4: CNNs for Color Image Classification

Objective: Train CNNs on RGB images (e.g., CIFAR-10).

- Load and explore dataset.
- Build deeper CNNs and compare training with and without batch normalization.
- Visualize feature maps from intermediate layers.
- Perform transfer learning using a pre-trained model (e.g., VGG16).

Lab Sheet 5: Time Series Forecasting with RNNs

Objective: Forecast future values using RNNs.

- Use a real-world dataset (e.g., temperature or stock prices).
- Preprocess and reshape time series data.
- Train Exponential Smoothing model and compare with a simple RNN.
- Predict future values and visualize performance using MAE/RMSE.

Lab Sheet 6: Document Classification with LSTM

Objective: Classify news/text using an LSTM-based classifier.

- Load and clean a text dataset (e.g., IMDB, 20Newsgroups).
- Tokenize, pad sequences, and convert text to embeddings.
- Build and train an LSTM classifier.
- Evaluate performance and visualize ROC curve.

Lab Sheet 7: Binary Document Classification with LSTM

Objective: Build and evaluate a binary classifier for sentiment analysis.

- Repeat preprocessing steps for binary labeled dataset.
- Train LSTM model and test it on unseen reviews.
- Add dropout and L2 regularization to prevent overfitting.

- Use Grad-CAM or LIME for interpretability.

Lab Sheet 8: Hybrid LSTM + CNN Text Classifier

Objective: Integrate CNNs with LSTM for improved feature extraction.

- Design a hybrid architecture combining CNN (feature extraction) + LSTM (temporal pattern).
- Train on labeled dataset.
- Compare performance with standalone CNN or LSTM.
- Visualize embedding vectors with t-SNE.

Lab Sheet 9: Handwritten Digit Recognition

Objective: Implement Softmax and MLP for MNIST classification.

- Use Softmax regression to classify digits.
- Build and evaluate a multi-layer perceptron (MLP).
- Compare confusion matrices and visualize misclassifications.
- Fine-tune hyperparameters (activation, optimizer).

Lab Sheet 10: Object Detection using YOLOv5

Objective: Detect objects in real-time using YOLOv5.

- Install YOLOv5 and configure environment.
- Run detection on images and video streams.
- Train YOLOv5 on a small custom dataset (if feasible).
- Evaluate precision, recall, mAP.

Lab Sheet 11: Image Segmentation with U-Net

Objective: Segment medical or satellite images using U-Net.

- Load and preprocess image-mask pairs.
- Train a U-Net model for pixel-wise segmentation.
- Visualize prediction masks.
- Compute IoU, Dice coefficient for evaluation.

Lab Sheet 12: Vision Transformers and Generative Models

Objective: Classify images using Vision Transformers and generate images with GANs.

- Implement Vision Transformer using pre-trained weights (e.g., ViT from HuggingFace).
- Train and test DCGAN or StyleGAN on a small image dataset.
- Visualize generated images and compare with training set.
- Perform latent space interpolation.

Group Project Options

- **Gesture Recognition System**
  - Detect and classify hand gestures using CNNs or MediaPipe.
  - Build dataset with webcam feed.
  - Apply preprocessing and classification.
- **OCR Pipeline**
  - Build end-to-end OCR system using OpenCV + Tesseract/CNN.
  - Detect text regions, perform segmentation, and recognize characters.

- **Custom Object Detection**
  - Train a Faster R-CNN model using custom annotated data (via LabelImg).
  - Evaluate detection using IoU and real-time testing with a webcam.

Targeted Applications & Tools:

Tools/Software/ Platform	**Application**
OpenCV 4, Python 3.7	Real-time face detection using webcam
Python 3.7 (Keras, TensorFlow), MATLAB	Image segmentation for medical images
OpenCV 4, Python 3.7 (Tesseract), MATLAB	Optical Character Recognition (OCR) on scanned documents
OpenCV 4, Python 3.7	Gesture recognition using hand tracking
OpenCV 4, Python 3.7	Vehicle number plate detection
Python 3.7 (Keras/TensorFlow)	Image classification using CNNs
Python 3.7, OpenCV 4	Custom object detection with YOLOv5
Python 3.7 (TensorFlow), MATLAB	Time series forecasting of sensor data
OpenCV 4, Python 3.7	Face recognition using eigenfaces or deep embeddings
MATLAB	Simulating convolutional operations on images

Skill-Building Activities through Experiential Learning:

- **Real-Time Sign Language Translator**
  Develop a deep learning model to detect hand gestures and translate them into text in real time. The activity builds skills in data annotation, video processing, and deep learning for gesture recognition using Python, OpenCV, and TensorFlow, fostering practical understanding of real-world computer vision applications.

- **AI-Based Art Generator Using GANs**
  Use Generative Adversarial Networks (GANs) to generate artistic images by applying style transfer to real-world photos. Participants gain hands-on experience with adversarial learning, dataset preparation, and image generation techniques, using Python and deep learning frameworks to explore creativity in AI applications.

- **Explainable Image Classifier for Medical Diagnosis**
  Build a deep learning image classifier for medical images and integrate explainability tools like Grad-CAM or LIME. This activity enhances understanding of model interpretability, trustworthiness, and critical analysis, providing exposure to AI applications in healthcare using Python, Keras, and open-source medical datasets.

Text Book

1. "Deep Learning for Computer Vision Image Classification, Object Detection and Face Recognition in Python" Jason Brownlee (2019)
2. "Deep Learning for Computer Vision with python" Adrian Rosebrock (2017)
3. Goodfellow, I., Bengio, Y., Courville, A., *Deep Learning*, MIT Press, **2016**.
4. Adrian Rosebrock, *Deep Learning for Computer Vision with Python*, PyImageSearch, **2019** (latest available edition).

References

R1. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
A foundational book covering deep learning principles, including CNNs, optimization, and generative models.

R2. **Raschka, S., &Mirjalili, V. (2022).** Machine Learning with PyTorch and Scikit-Learn. Packt Publishing.
Covers practical deep learning techniques using PyTorch, including CNNs and transfer learning.

R3. **Geron, A. (2022).** Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (3rd Edition). O'Reilly Media.
Provides hands-on implementations of deep learning for computer vision using TensorFlow and Keras.

R4. Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). Dive into Deep Learning. Available online (https://d2l.ai).
Open-access book covering CNNs, object detection, and advanced vision techniques with PyTorch and TensorFlow.

R5. Chollet, F. (2021). Deep Learning with Python (2nd Edition). Manning Publications.
Explains deep learning fundamentals and applications with Keras, including image classification and segmentation.

R6. **Ballé, J., Laparra, V., & Simoncelli, E. P. (2017).** Deep Learning for Computer Vision: A Brief Introduction.
A concise introduction to CNNs, object detection, and generative models.

E-Resources/Web Resources

- https://www.geeksforgeeks.org/computer-vision/deep-learning-for-computer-vision/
- https://www.coursera.org/learn/advanced-deep-learning-techniques-computer-vision
- https://onlinecourses.nptel.ac.in/noc20_cs88/preview

Course Code: CSS3426	Course Title: Front-end Full Stack Development Course Type :Theory Integrated Lab	L- T-P- C	2-0-2-3
Version No.	1.0		
Course Pre-requisites	CSS2269		
Anti-requisites	NIL		
Course Description	This intermediate course enables students to perform front-end full stack development, with emphasis on employability skills. The course covers key technologies and architectures that enables the student to design and implement front-end. On successful completion of this course, the student shall be able to pursue a career in full-stack development. The students shall develop strong problem-solving skills as part of this course.		

Course Objectives	The objective of the course is to familiarize the learners with the concepts Front end Full Stack Development and attain Employability through experiential Learning techniques.
Course Outcomes	On successful completion of the course the students shall be able to: CO1: Describe the fundamentals of DevOps and front-end full-stack development. — L2 – Understand CO2: Illustrate a basic web design using HTML, CSS, and JavaScript. — L3 – Apply CO3: Develop responsive web pages using appropriate web technologies and design principles. — L3 – Apply CO4: Apply Angular.js concepts to develop interactive web front-end applications. — L3 – Apply

Course Content:

Module 1	Fundamentals of DevOps	Project		Programming	04 Sessions

Topics:

Introduction to Agile Methodology; Scrum Fundamentals; Scrum Roles, Artifacts and Rituals; DevOps – Architecture, Lifecycle, Workflow & Principles; DevOps Tools Overview – Jenkins, Docker, Kubernetes.

Review of GIT source control.

Module 2	Web Design & Development	Project		Programming	03 Sessions

Topics:

HTML5 – Syntax, Attributes, Events, Web Forms 2.0, Web Storage, Canvas, Web Sockets; CSS3 – Colors, Gradients, Text, Transform;

Assignment: Develop a website for managing HR policies of a department.

Module 3	Responsive web design	Project		Programming	08 Sessions

Topics:

BootStrap for Responsive Web Design; JavaScript – Core syntax, HTML DOM, objects, classes, Async; Ajax and jQuery Introduction

Assignment: Design and develop a website that can actively keep track of entry-exit information of a housing society.

Module 4	Fundamentals of Angular.js	Project		Programming	15 Sessions

Topics:

Setting up Development & Build Environment: Node.js and NPM; Introduction to TypeScript; Working with OOP concepts with TypeScript; Angular Fundamentals; Angular CLI; Introduction to TypeScript; Debugging

Angular applications; Components & Databinding in Depth; Angular Directives; Using Services & Dependency Injection; Angular Routing; Observables; Handling Forms in Angular Apps; Output transformation using Pipes; Making Http Requests; Authentication & Route Protection; Dynamic Components; Angular Modules &Optimizing Angular Apps; Deploying an Angular App; Angular Animations; Adding Offline Capabilities with Service Workers; Unit Testing in Angular Apps (Jasmine, Karma). Overview of React.js Assignment: Develop a software tool to do inventory management in a warehouse.
Targeted Application & Tools that can be used: Application Area is to Design and Analyzing the efficiency of Algorithms. This fundamental course is used by all application developers. Professionally Used Software: GCC compiler.
Text Book: T1. Fender, Young, "Front-end Fundamentals", Leanpub, 2015 T2. Northwood, Chris, "The Full Stack Developer: Your Essential Guide to the Everyday Skills Expected of a Modern Full Stack Web Developer", APress, 2018
References: R1. Flanagan D S, "Javascript : The Definitive Guide" 7th Edition. 7th ed. O'Reilly Media; 2020. R2. Alex Libby, Gaurav Gupta, and AsojTalesra. "Responsive Web Design with HTML5 and CSS3 Essentials", Packt Publishing, 2016 R3. Duckett J Ruppert G Moore J. "Javascript&Jquery : Interactive Front-End Web Development."; Wiley; 2014. R4. Web Reference: https://www.youtube.com/watch?v=JGNTYXkVCVY&list=PLd3UqWTnYXOkTSBCBNyyhxo_jxlY_uTWA&index=2 R5. Web Reference: https://www.freecodecamp.org/news/frontend-web-developer-bootcamp/ https://puniversity.informaticsglobal.com:2229/login.aspx?direct=true&db=nlebk&AN=2233842&site=ehost-live https://nptel.ac.in/courses/106102064
Topics relevant to development of "Employability": DevOps Tools Overview – Jenkins, Docker, Kubernetes for developing Employability Skills through Experiential Learning techniques. This is attained through assessment component mentioned in course handout.

Course Code:	Course Title: Java Full Stack Development	L- T-P- C	2-0-2-3

CSS3427	Course Type: Theory Integrated Lab		
Version No.	1.0		
Course Pre-requisites	CAI2509		
Anti-requisites	Nil		
Course Description	This advanced level course enables students to perform full stack development using Java, with emphasis on employability skills. The key technologies used for Full Stack development is based on either Java technology or .NET technology. In this course, the focus is on using Java, and the related technologies/tools like Java EE, Java Persistence, Hibernate, Maven, Spring Core, etc. On successful completion of this course, the student shall be able to pursue a career in full-stack development. The students shall develop strong problem-solving skills as part of this course.		
Course Objectives	This course is designed to improve the learners' EMPLOYABILITY SKILLS by using PROBLEM SOLVING Methodologies.		
Course Outcomes	On successful completion of the course the students shall be able to: CO1: Practice the use of Java for full-stack application development. L3 – Apply CO2: Develop web applications using Java EE technologies. L3 – Apply CO3: Solve application development problems using Java Persistence API (JPA) and Hibernate. L3 – Apply CO4: Apply Spring framework concepts to develop full-stack applications. L3 – Apply CO5: Employ automation tools such as Maven and Selenium for full-stack development and testing. L3 – Apply		

Course Content:

Module 1	Introduction	Project	Programming	03 Sessions

Topics:

Review of Java; Advanced concepts of Java; Java generics; Java IO; New Features of Java. Unit Testing tools.

Module 2	Java EE Web Applications	Project	Programming	05 Sessions

Topics:

Introduction to Eclipse & Tomcat; JSP Fundamentals; Reading HTML form Data with JSP; State Management with JSP; JSP Standard Tag Library - Core & Function Tags; Servlet API Fundamentals; ServletContext, Session, Cookies; Request Redirection Techniques; Building MVC App with Servlets & JSP; Complete App - Integrating JDBC with MVC App

Assignment: Develop an application for managing HR policies of a department.

Module 3	Java Persistence using JPA and	Project	Programming	06 Sessions

	Hibernate			

Topics:

Fundamentals of Java Persistence with Hibernate; JPA for Object/Relational Mapping, Querying, Caching, Performance and Concurrency; First & Second Level Caching, Batch Fetching, Optimistic Locking & Versioning; Entity Relationships, Inheritance Mapping & Polymorphic Queries; Querying database using JPQL and Criteria API (JPA)

Assignment: Design and develop a website that can actively keep track of entry-exit information of a housing society.

Module 4	Spring Core	Project	Programming	10 Sessions

Topics:

Spring Core, Spring MVC, Spring Boot REST API; Understanding Spring Framework; Using Spring MVC; Building a Database Web App with Spring and Hibernate o Spring AOP (Aspect Oriented Programming); Implementing Spring Security; Developing Spring REST API; Using Spring Boot for Rapid Development

Assignment: Develop a software tool to do inventory management in a warehouse.

Module 5	Automation tools	Project		Programming	06 Sessions

Topics:

Introduction to Automation Tools; Apache Maven: Maven Fundamentals, Software Setup - Commandline and Eclipse, pom.xml and Directory Structure, Multi-Module Project Creation, Scopes, Dependency Management, Profiles; Functional/BDD Testing using Selenium, Selenium Fundamentals and IDE, Selenium WebDriver, Installation and Configuration, Locating WebElements, Driver Commands, WebElement Commands

Assignment: Illustrate the use of automation tools in the development of a small software project.

Targeted Application & Tools that can be used:

Application Area is to Design and Analyzing the efficiency of Algorithms. This fundamental course is used by all application developers.

Professionally Used Software: Eclipse, NetBeans, Hibernate, Selenium, Maven, GIT.

Project work/Assignment:

Problem Solving: Design of Algorithms and implementation of programs.

Programming: Implementation of given scenario using Java.

Text Book:

T1. Fender, Young, “Front-end Fundamentals”, Leanpub, 2015

References

R1. Soni, Ravi Kant. “Full Stack AngularJS for Java Developers: Build a Full-Featured Web Application from Scratch Using AngularJS with Spring RESTful.” , Apress, 2017.

R2. Mardan, Azat. “Full Stack JavaScript: Learn Backbone.js, Node.js and MongoDB.”, Apress, 2015

Course Code: CSS3428	Course Title: .NET Full Stack Development Course Type: Theory Integrated Lab	L- T-P- C	2-0-2-3
Version No.	1.0		
Course Pre-requisites	CAI2509		
Anti-requisites	Nil		
Course Description	This advanced level course enables students to perform full stack development using .NET, with emphasis on employability skills. The key technologies used for Full Stack development is based on either Java technology or .NET technology. In this course, the focus is on using .NET and the related technologies/tools like C#, ASP.NET, Entity Framework Core, etc. On successful completion of this course, the student shall be able to pursue a career in full-stack development. The students shall develop strong problem-solving skills as part of this course.		
Course Objectives	This course is designed to improve the learners' EMPLOYABILITY SKILLS by using PROBLEM SOLVING Methodologies.		
Course Outcomes	On successful completion of the course the students shall be able to: CO1: Practice the use of C# for developing small-scale applications. L3 – Apply CO2: Develop web applications using Entity Framework. L3 – Apply CO3: Solve web application development problems using SQL and ASP.NET technologies. L3 – Apply CO4: Apply ASP.NET concepts to develop full-stack web applications. L3 – Apply		

Course Content:

Module 1	C# Programming for Full Stack Development	Project	Programming	10 Sessions

Topics:

.NET Framework Fundamentals, Visual Studio IDE Fundamentals, C# Language Features, Working with arrays and collections, Working with variables, operators, and expressions, Decision and iteration statements, Managing program flow and events, Working with classes and methods, OOP concepts, Properties, Auto Implemented, Delegates, Anonymous Methods and Anonymous Types, Extension methods, Sealed Classes/Methods, Partial Classes/Methods, Asynchronous programming and threading, Data validation and working with data collections including LINQ, Handling errors and exceptions, Working with Files, Unit Testing – Nunit framework

Assignment: Develop a small application for managing library using C#.

Module 2	Entity Framework Core 2.0	Project	Programming	06 Sessions
Topics: Entity Framework Core 2.0 Code First Approach; Introduction To Entity Framework and EDM; Querying the EDM; Working With Stored Procedures; Advanced Entity Framework - DbContext [EF6]; Advanced Operations; Performance Optimization; Data Access with ADO.NET Assignment: Develop an application for managing HR policies of a department.				
Module 3	ASP.NET	Project	Programming	06 Sessions
Topics: ASP.NET Core, ASP.Net Core 3.1 MVC, ASP.NET Core Middleware and Request pipeline, Review of SQL using MS SQL, Working With Data In Asp.Net, Razor View Engine, State Management In Asp. Net MVC & Layouts; Assignment: Develop a web application to mark entry/exit of guests in a building.				
Module 4	ASP.NET	Project	Programming	08 Sessions
Topics: Introduction To Models, Validations In Asp.Net MVC, Authentication and Authorization In Asp.Net MVC, Advanced Asp. Net MVC - Ajax Action Link In MVC, Advanced Asp.Net MVC - Ajax Forms In MVC, Microsoft Testing Framework – Unit Testing the .NET Application Assignment: Develop a software tool to do inventory management in a warehouse.				
Targeted Application & Tools that can be used: Application Area is to Design and Analyzing the efficiency of Algorithms. This fundamental course is used by all application developers. Professionally Used Software: Visual Studio				
Project work/Assignment:				
Problem Solving: Design of Algorithms and implementation of programs. Programming: Implementation of given scenario using .NET.				
Text Book: T1. Fender, Young, “Front-end Fundamentals”, Leanpub, 2015 T2. Valerio De Sanctis, “ASP.NET Core 5 and Angular: Full-stack web development with .NET 5 and Angular 11”, 4th Edition, Packt, 2021.				
References R1. Benjamin Perkins, Jon D. Reid, “Beginning C# and .NET”, Wiley, 2021 Reid, 2021. R2. Piotr Gankiewicz, “Full Stack .NET Web Development”, Packt Publishing, 2017.				

R3. Tamir Dresher, Amir Zuker, Shay Friedman, “Hands-On Full-Stack Web Development with ASP.NET Core”, Packt Publishing, 2018.

R4. Dustin Metzgar, “Exploring .NET core with microservices, ASP.NET core, and Entity Framework Core”, Manning, 2017.

Course Code: CSS3502	Course Title: Autonomous Learning Systems and Model Optimization Type of Course: PCC Theory Integrated Lab	L-T-P-C	2-0-2-3
Version No.	1.0		
Course Prerequisites	CSEXXXX		
Antirequisites	NIL		
Course Description	This advanced course focuses on the design, development, and optimization of autonomous learning systems capable of improving performance through experience without explicit programming. It emphasizes scalable learning frameworks, dynamic feedback systems, and model self-adjustment mechanisms essential in AI-driven decision environments. Learners will explore high-performance training loops, self-optimization algorithms, model compression strategies, reinforcement learning policies, and real-world deployment use cases in FinTech, robotics, personalization, and operations. The curriculum integrates structured experimentation, hardware-aware model tuning, and production-ready feedback loops across distributed and online learning systems.		
Course Objective	To equip students with the technical ability to build self-improving learning architectures, optimize model performance across environments, and apply autonomous intelligence in production-grade systems. The course emphasizes automation in hyperparameter tuning, architecture search, real-time adaptation, and continuous performance monitoring.		
Course Outcomes	On successful completion of this course the students shall be able to: 1. Design learning architectures capable of autonomous adaptation and feedback integration. 2. Implement optimization workflows for neural network efficiency, latency, and performance. 3. Apply reinforcement learning techniques for decision-making in real-time systems. 4. Optimize AI systems using search-based, gradient-based, and population-based methods. 5. Deploy scalable and resource-aware learning pipelines for domain-specific applications.		

Course Content:					
Module 1:	Autonomous Model Selection and Self-Tuning Systems	Quiz		Knowledge based Quiz	9 Sessions
Topics: Automated hyperparameter tuning (Bayesian, grid, random search), AutoML pipelines, architecture search techniques (NAS), learning rate scheduling, dynamic validation strategies Tools: Optuna, Ray Tune, Auto-sklearn, Google Vizier Use Cases: • Personalized credit limit recommendation engine using self-tuned gradient boosting • 0041d-spend model optimization using NAS					
Module 2:	Reinforcement Learning for Decision IntelligenceX	Quiz		Knowledge based Quiz	9 Sessions
Topics: Policy gradient methods, Q-learning variants, actor-critic frameworks, reward shaping, explorationexploitation balancing, RL environments and benchmarking Tools: OpenAI Gym, Stable-Baselines3, RLlib, TensorBoard Use Cases: • Transaction fraud detection engine with feedback-loop learning • robo-advisory investment rebalancing using reinforcement-based decisions					
Module 3:	Model Compression and Efficiency Engineering	Quiz		Knowledge based Quiz	9 Sessions
Topics: Pruning and quantization techniques, knowledge distillation, sparsity-driven modeling, latencyaware optimization, compression-aware retraining. Tools: TensorFlow Model Optimization Toolkit, ONNX Runtime, PyTorch Quantization, DeepSpeed Use Cases: • Real-time credit scoring on low-powered devices • Optimizing latency of image-based KYC models for mobile onboarding					
Module 4:	Continual and Online Learning Architectures	Quiz		Knowledge based Quiz	9 Sessions

Topics: Data drift detection, incremental learning, lifelong learning strategies, catastrophic forgetting mitigation, adaptive weight freezing, real-time data assimilation
Tools: River (online ML), Avalanche, TensorFlow Data Service, ELT + stream pipelines Use Cases:

- Portfolio optimization engine adapting to live market data
- UPI transaction classification with daily update loops

Module 5:	Adaptive Deployment and Model Governance	Quiz		Knowledge based Quiz	9 Sessions

Topics: Model versioning and rollback systems, performance-aware routing (A/B/n), shadow deployments, adaptive retraining triggers, model governance and policy controls
Tools: MLflow, Kubernetes, Seldon Core, Prometheus Use Cases:

- Dynamic pricing engine with automated rollout strategy
- Recommender systems updated via feedback threshold detection

Project work/Assignment:

Students will work on building an end-to-end autonomous learning system tailored to a real-world use case. This includes developing a credit risk scoring engine that continuously updates based on live borrower behavior or constructing a reinforcement learning-based decision agent for optimizing loan approval thresholds. The project must demonstrate automated model selection, self-tuning capabilities, real-time feedback integration, and performance governance using tools like Optuna, Ray Tune, or MLflow. Deliverables will include a modular codebase, deployment plan, and performance monitoring dashboard.

Text Book

T1: Co-branded Zell and Paytm Textbook for Autonomous Learning Systems and Model Optimization

References

- Automated Machine Learning – Frank Hutter, Lars Kotthoff, Joaquin Vanschoren
- Reinforcement Learning: An Introduction – Richard S. Sutton, Andrew G. Barto
- Efficient Processing of Deep Neural Networks – Vivienne Sze et al. (MIT Press)
- Hands-On AutoML with Python – Sibanjan Das, Umit Mert Cakmak
- Goodfellow, Ian, Bengio, Yoshua, and Courville, Aaron. Deep Learning, MIT Press, 2016.

Web References

- https://optuna.org/
- https://www.rllib.io/
- https://pytorch.org/tutorials/intermediate/pruning_tutorial.html
- https://mlflow.org/
- https://www.deepspeed.ai/

Topics relevant to SKILL DEVELOPMENT: This course develops critical expertise in building autonomous AI systems with minimal manual intervention. Students will work hands-on with hyperparameter tuning tools like Optuna and Ray Tune, construct reinforcement learning workflows with Stable-Baselines3, and apply model compression techniques for latency-sensitive environments. The curriculum enhances proficiency in real-time adaptation using online learning frameworks, model versioning via MLflow, and feedback-driven governance strategies for production systems. Learners will develop a strong command over distributed deployment logic, dynamic feedback loops, and modularized pipelines—essential for roles in applied machine learning, AI engineering, and algorithmic systems development across FinTech, robotics, and personalization domains.

Course Code: CSS3502	Course Title: Architectures for Deep Representation and Adaptive Learning Type of Course: PCC – Theory Integrated Lab	L-T-P- C	2	0	2	3
Version No.	1.0					
Course Prerequisites	CSEXXXX					
Antirequisites	NIL					
Course Description	This advanced course explores cutting-edge neural network architectures that enable deep representation learning and dynamic adaptability across tasks and domains. It focuses on learning robust, transferable features from high-dimensional data using advanced models like Transformers, Variational Autoencoders, Siamese Networks, and Meta-Learning frameworks. Students will engage with domain-specific challenges in FinTech, healthcare, computer vision, and NLP where architectures must evolve, generalize, and adapt. The course emphasizes building scalable systems with embedded representation learning capabilities for tasks such as fraud detection, user intent recognition, and automated credit assessment.					
Course Objective	To train students in designing deep learning architectures that can extract abstract representations and dynamically adapt to data variability, multi-task environments, and cross-domain shifts. The course equips learners to experiment with deep feature engineering, multi-head attention, meta-learners, and hybrid neural systems for highperformance intelligent applications.					

<table>
<tr><td>Course Outcomes</td><td colspan="5">On successful completion of this course the students shall be able to:
1. Design and implement deep learning architectures for robust feature representation across complex datasets.
2. Apply attention mechanisms and transformer-based models to sequence and document data.
3. Leverage meta-learning techniques to build fast-adapting learners across FinTech and health domains.
4. Optimize models using contrastive, self-supervised, and hybrid learning approaches.
5. Evaluate generalization performance across dynamic, multi-task, and real-world datasets.</td></tr>
<tr><td>Course Content:</td><td colspan="5"></td></tr>
<tr><td>Module 1:</td><td>Deep Feature Learning with Hybrid Architectures</td><td>Quiz</td><td></td><td>Knowledge based Quiz</td><td>9 Sessions</td></tr>
<tr><td colspan="6">Topics: Deep embedding architectures, contrastive loss, Siamese and triplet networks, self-supervised encoders, hybrid feature stacks
Tools: PyTorch Lightning, Keras Functional API
Use Cases: Similarity-based fraud detection, user profile clustering in FinTech</td></tr>
<tr><td>Module 2:</td><td>Transformer Architectures and Attention Models</td><td>Quiz</td><td></td><td>Knowledge based Quiz</td><td>9 Sessions</td></tr>
<tr><td colspan="6">Topics: Scaled dot-product attention, multi-head attention, encoder-decoder transformers, fine-tuning strategies (BERT, RoBERTa, GPT-like models), transformer scaling and compression
Tools: Hugging Face Transformers, TensorFlow Hub
Use Cases: Intelligent KYC document understanding, transaction categorization using NLP</td></tr>
<tr><td>Module 3:</td><td>Meta-Learning and Few-Shot Adaptation</td><td>Quiz</td><td></td><td>Knowledge based Quiz</td><td>9 Sessions</td></tr>
<tr><td colspan="6">Topics: Model-agnostic meta-learning (MAML), Prototypical Networks, Reptile algorithm, gradientbased task generalization, episodic learning
Tools: Higher (PyTorch), Meta-World Gym, Scikit-learn for meta-prediction
Use Cases: Loan approval for sparse-profile users, dynamic pricing prediction for new product SKUs</td></tr>
<tr><td>Module 4:</td><td>Variational and Disentangled Representation Models</td><td>Quiz</td><td></td><td>Knowledge based Quiz</td><td>9 Sessions</td></tr>
</table>

Topics: VAEs, latent variable modeling, disentanglement metrics, sampling and interpolation, regularization techniques (β-VAE, InfoGAN) Tools: Pyro, Edward2, Keras-VAE libraries Use Cases: Credit score simulation engines, generative stress-testing of digital lending portfolios					
Module 5:	Multi-Task Learning and Cross-Domain Generalization	Quiz		Knowledge based Quiz	9 Sessions
Topics: Multi-task loss balancing, shared-bottom and cross-stitch architectures, adversarial domain adaptation, domain-invariant representation learning Tools: TensorFlow Extended, PyTorch Ignite, DomainBed Use Cases: Unified user scoring across credit, insurance, and wealth; domain-adapted recommendation engines					
Project work/Assignment: Students will develop an adaptive deep representation system for a FinTech or e-commerce use case, such as a hybrid fraud detection model using Siamese networks and transaction embeddings, or a document intelligence system powered by transformers for classifying multi-language KYC documents. The final deliverables include a codebase, trained model artifacts, model adaptation report, and an interactive demo interface showcasing representation quality and transferability across data domains.					
Text Book T1: Co-branded Zell and Paytm Textbook for Architectures for Deep Representation and Adaptive Learning					
References • Representation Learning: A Review and New Perspectives – Yoshua Bengio et al. • Deep Learning for Natural Language Processing – Palash Goyal et al. • Meta-Learning – Andrei A. Rusu et al. • Generative Deep Learning – David Foster (O'Reilly)					
Web References • https://huggingface.co/ • https://pytorch-lightning.ai/ • https://keras.io/guides/functional_api/ • https://meta-world.github.io/ • https://domainbed.github.io/					

Topics relevant to SKILL DEVELOPMENT: This course fosters expertise in constructing deep representation systems that can learn from limited data, adapt to new tasks, and generalize across domains. Students develop capabilities in working with stateof-the-art libraries like Hugging Face, PyTorch Lightning, and Meta-World to implement hybrid, transformer, and meta-learning models. Practical skills include contrastive feature engineering, transformer fine-tuning, task adaptation workflows, latent space visualization, and domain-invariant training pipelines. These skills are critical for careers in AI product development, research engineering, and real-time intelligent system deployment, especially within FinTech, NLP, and autonomous decision systems.

Course Code: CSS3502	Course Title: Cross-Modal AI: Vision, Speech, and Contextual Fusion Models Type of Course: Theory Integrated Lab	L-T-P- C	2	0	2	3
Version No.	1.0					
Course Prerequisites	CSEXXXX					
Antirequisites	NIL					
Course Description	This course focuses on designing and deploying cross-modal AI systems that integrate visual, auditory, and textual data to enable contextual intelligence. Students will explore transformer-based and contrastive architectures for multimodal understanding, fusion techniques such as attention-based integration and cross-encoders, and learn how models like CLIP, Flamingo, and Whisper power modern applications. Emphasis is placed on real-world FinTech, accessibility, e-commerce, and surveillance applications where vision, speech, and language must interact seamlessly for downstream prediction, annotation, or automation tasks.					
Course Objective	To develop student expertise in building advanced multimodal models that combine vision, speech, and text inputs for predictive, generative, and interpretive AI use cases across industries.					
Course Outcomes	On successful completion of this course the students shall be able to: 1. Design cross-modal pipelines that integrate vision, speech, and text inputs. 2. Apply and adapt multimodal transformer architectures for retrieval, captioning, classification, and intent detection. 3. Implement contrastive learning techniques for alignment and fusion of multisource data. 4. Develop deployable multimodal agents with real-world tasks such as KYC document verification, voice-command transaction systems, and fraud detection from behavioral cues.					

Course Content:					
Module 1:	Multimodal Representations and Embedding Alignment	Quiz		Knowledge based Quiz	9 Sessions
Topics: Shared representation learning, visual and audio embedding spaces, contrastive pretraining (CLIP, ALIGN), modality alignment loss functions Tools: PyTorch, OpenCLIP, Scikit-learn for t-SNE Use Cases: Image-text retrieval for financial dashboards, audio-visual emotion tagging in fraud calls					
Module 2:	Vision-Language Transformers and Cross Attention Models	Quiz		Knowledge based Quiz	9 Sessions
Topics: Cross-attention encoders, dual encoders vs. fusion models, ViLBERT and FLAVA architecture analysis Tools: Hugging Face Transformers, MMF (Facebook), PyTorch Lightning Use Cases: Automatic policy document reading from scanned images, visual report explanation bots					
Module 3:	Speech-to-Text Integration and Whisper-based Systems	Quiz		Knowledge based Quiz	9 Sessions
Topics: Whisper, DeepSpeech, end-to-end speech-to-text, voice activity detection, speaker embeddings Tools: Whisper API, Mozilla DeepSpeech, Wave2Vec 2.0 Use Cases: Audio onboarding in FinTech apps, voice-activated investment advisory, voice-to-summary transcription					
Module 4:	Cross-Modal Fusion and Reasoning	Quiz		Knowledge based Quiz	9 Sessions
Topics: Late vs. early fusion, attention gating, co-attentional reasoning, multimodal inference pipelines, zero-shot capabilities Tools: Multimodal Transformers, LangChain, OpenAI API, TorchMultimodal Use Cases: Cross-modal Q&A systems, multi-input fraud alert systems (screen + voice + click behavior)					
Module 5:	Multimodal Agent Deployment and Latency Optimization	Quiz		Knowledge based Quiz	9 Sessions
Topics: Lightweight models for edge deployment, quantization for multimodal inputs, real-time inference optimization, fallback strategies Tools: ONNX, TensorRT, Nvidia Triton, OpenVINO Use Cases: Real-time fraud signal detection, agent-assisted compliance checks, in-branch biometric authentication kiosks					

Project work/Assignment: Students will build a functional cross-modal AI system that fuses at least two modalities—such as speech and image or text and video—targeting a FinTech or e-commerce problem. Suggested projects include visual fraud claim validation using document OCR and audio explanation, or voice-to-contract summarization with legal-risk flagging. The project must demonstrate alignment of input modalities, latency-aware architecture, and robust output interpretation for end-users.
Text Book T1: Co-branded Zell and Paytm Textbook for Cross-Modal AI: Vision, Speech, and Contextual Fusion Models
References • Multimodal Machine Learning – Baltrusaitis, Ahuja & Morency (ACM Computing Surveys) • Deep Learning for Vision Systems – Mohamed Elgendy • Speech and Language Processing (3rd Edition Draft) – Jurafsky & Martin • Foundations of Multimodal Learning – Dieterich Lawson Web References https://github.com/openai/whisperhttps://huggingface.co/openai/clip
https://facebookresearch.github.io/mmf/https://pytorch.org/audio/https://paperswithcode.com/task/multimodal-learning
Topics relevant to SKILL DEVELOPMENT: This course builds proficiency in developing and deploying multimodal AI solutions that can seamlessly combine visual, auditory, and textual data streams. Students gain hands-on experience in aligning diverse data representations using contrastive loss, designing attention-based fusion architectures, and evaluating semantic accuracy across modalities. Skills include integrating pre-trained speech-to-text models, working with OCR for documents, and designing agents capable of handling ambiguous or conflicting input signals. These capabilities are crucial for AI product development in fields like accessibility tech, legaltech automation, financial surveillance, and hybrid chat interfaces.

Course Code: CSS3502	Course Title: Decentralized Data Platforms and Digital Finance Applications Type of Course: Theory Integrated Lab	L-T-P- C	2	0	2	3
Version No.	1.0					
Course Prerequisites	CSEXXXX					

Antirequisites	NIL				
Course Description	This course explores the engineering of decentralized data systems that support applications in smart finance, digital assets, and programmable money. Students will work with hybrid on-chain/off-chain architectures, oracles, event-driven analytics, and audit-focused data storage for financial automation systems. Emphasis is placed on transaction transparency, state integrity, time consistency, and compliance-aware ledger analytics.				
Course Objective	To equip students with the skills to design, analyze, and operate decentralized data infrastructure for digital financial ecosystems involving smart contracts, token economics, and programmable settlement.				
Course Outcomes	On successful completion of this course the students shall be able to: 1. Design event-driven data architectures for decentralized financial platforms 2. Integrate off-chain computation and smart contract state into analytics pipelines 3. Model and store transactional data for auditability and real-time finance dashboards 4. Handle validator node data, oracle feeds, and digital asset transfer logs				
Course Content:					
Module 1:	Data Architecture for Digital Finance Ecosystems	Quiz		Knowledge based Quiz	9 Sessions
Topics: On-chain vs o -chain data separation, blockchain ledger structures, DeFi primitives (AMMs, staking, lending), decentralized identity (DID) data models Tools: Ethereum JSON-RPC, Polygon SDK, Web3.py, Alchemy, Infura Use Cases: • Extract token swaps and liquidity movements from Uniswap • Track staking activity across validator nodes					
Module 2:	Oracles, Smart Contracts, and Off-chain Aggregation	Quiz		Knowledge based Quiz	9 Sessions
Topics: Time-series data feeds, oracle design patterns, event aggregation, layer-2 settlements, bridge mechanisms Tools: Chainlink, Band Protocol, Subgraphs (The Graph), Gelato Network Use Cases: • Build oracle flow for asset price ingestion into lending protocol • Create a subgraph to index on-chain governance proposals					

Module 3:	Storage Systems and State Snapshots	Quiz		Knowledge based Quiz	9 Sessions
Topics: Immutable storage, off-chain state compression, Merkle tree snapshots, IPFS-based backups, validator checkpoints Tools: IPFS, Arweave, LevelDB, Filecoin, Chainbase Use Cases: • Store weekly snapshots of DeFi TVL (total value locked) • Design a validator checkpointing system with traceable logs					
Module 4:	Auditability, Compliance, and Analytics Pipelines	Quiz		Knowledge based Quiz	9 Sessions
Topics: Proof-of-reserve systems, compliance-led event tracing, DeFi risk metrics (impermanent loss, liquidation tracking), structured logging Tools: Dune Analytics, Flipside Crypto, Tenderly, BigQuery for Ethereum, ELT via Airbyte Use Cases: • Build real-time liquidation monitor for collateralized lending • Create an audit dashboard for NFT treasury activity					
Module 5:	Capstone – Applied DeFi Data Stack Design	Quiz		Knowledge based Quiz	9 Sessions
Topics: Complete decentralized finance analytics stack: ingestion, normalization, dashboarding, and lifecycle management for token and protocol metrics Tools: Apache Airflow, Superset, Subgraphs, Google Cloud Functions					
Use Cases: • Capstone: Build a full-stack data platform to track the health of a synthetic asset protocol, combining on-chain event data with off-chain analytics and live dashboards					
Project work/Assignment: Students will create a DeFi analytics platform that includes data ingestion from validator nodes or contract events, normalized off-chain processing, and a secure dashboard with financial KPIs. Deliverables include a functional stack, schema documentation, and a compliance-ready audit narrative..					

Text Book T1. T1. Co-branded Zell and Paytm Textbook for Decentralized Data Platforms and Digital Finance Applications
References T2. Elad Elrom – The Blockchain Developer T3. Narayanan et al. – Bitcoin and Cryptocurrency Technologies T4. The Graph Foundation – Subgraph Design Docs Web References W1. https://defillama.com W2. https://dune.com
Topics relevant to SKILL DEVELOPMENT: This course equips students to engineer data systems for modern decentralized finance environments, combining protocol-level thinking with real-time analytics and compliance constraints. Learners gain hands-on

Course Code: CSS3502	Course Title: Transformer-based Language Systems and Semantic Intelligence Type of CourseTheory Integrated Lab	L-T-P- C	2	0	2	3
Version No.	1.0					
Course Prerequisites	CSEXXXX					
Antirequisites	NIL					
Course Description	This course delves into advanced language representation models and semantic understanding frameworks driven by transformers. Students will explore the architectural principles and practical implementations of state-of-the-art models such as BERT, RoBERTa, GPT, T5, and language-adaptive pretraining variants. Emphasis is placed on fine-tuning these systems for real-world applications like question answering, entity linking, semantic search, legal document classification, and contextual language generation. The course promotes deep engagement with pre-trained models, parameter-efficient tuning techniques, and embedding-based knowledge distillation within enterprise NLP workflows.					

Course Objective	To equip students with deep expertise in developing, adapting, and deploying transformerbased language models that capture semantic meaning and contextual understanding for domain-specific NLP applications.				
Course Outcomes	On successful completion of this course the students shall be able to: 1. Analyze transformer language model architecture and layer-wise representation behavior. 2. Fine-tune large language models using task-specific datasets and parameter-efficient strategies. 3. Implement semantic similarity, topic modeling, and entity extraction using contextual embeddings. 4. Build scalable, domain-specific NLP systems for legal tech, FinTech, and health tech applications. 5. Optimize inference, latency, and interpretability of transformer-based pipelines in production.				
Course Content:					
Module 1:	Semantic Representations with Transformers	Quiz		Knowledge based Quiz	9 Sessions
Topics: Self-attention and positional encoding behavior, contextual token embeddings, masked language modeling, layer-wise semantic evolution Tools: TensorBoard, Captum, Hugging Face tokenizers Use Cases: Word-sense disambiguation, semantic error detection in financial contracts					
Module 2:	Transformer Fine-Tuning and Optimization	Quiz		Knowledge based Quiz	9 Sessions
Topics: Task-specific heads, parameter-efficient tuning (LoRA, adapters, prompt-tuning), learning rate schedulers, mixed-precision training Tools: Transformers library, DeepSpeed, Optuna Use Cases: Contract clause classification, chatbot response ranking, regulatory text flagging					
Module 3:	Knowledge-Aware and Cross-Lingual Transformers	Quiz		Knowledge based Quiz	9 Sessions
Topics: Knowledge embedding (K-BERT, LUKE), multilingual transformers (mBERT, XLM-R), entitylevel attention, alignment-based pretraining Tools: spaCy, DGL-KE, fastText Use Cases: Legal document summarization across jurisdictions, entity linking in insurance KYC documents					

Module 4:	Retrieval-Augmented and Memory-Enhanced Models	Quiz		Knowledge based Quiz	9 Sessions
Topics: Dense passage retrieval, dual encoders, RAG architecture, embedding index construction, context window expansion techniques Tools: Haystack, FAISS, SentenceTransformers Use Cases: Customer support retrieval systems, semantic search for investment reports, knowledgeintensive					
Module 5:	Deployable Transformer Architectures	Quiz		Knowledge based Quiz	9 Sessions
Topics: Quantization and distillation (TinyBERT, DistilBERT), serverless inference, endpoint security, prompt chaining Tools: ONNX Runtime, Triton Inference Server, LangChain Use Cases: Chatbot-as-a-service for wealth advisors, intelligent front-ends for legal discovery platforms					
Project work/Assignment: Students will build a domain-adaptive language model pipeline for a real use case, such as semantic contract validation using BERT-based clause classification, or a multilingual transformer system for extracting insights from investment whitepapers. The project must demonstrate fine-tuning strategy, semantic evaluation metrics (STS, NDCG), and production-readiness via scalable inference and embedding indexing.					
Text Book T1: Co-branded Zell and Paytm Textbook for Transformer-based Language Systems and Semantic Intelligence					
References • Transformers for Natural Language Processing – Denis Rothman • Natural Language Processing with Transformers – Lewis Tunstall, Leandro von Werra, Thomas Wolf • Deep Learning for Search – Tommaso Teofili • Semantic Web for the Working Ontologist – Dean Allemang, James Hendler Web References • https://huggingface.co/ • https://haystack.deepset.ai/ • https://github.com/huggingface/transformers • https://langchain.com/					
https://github.com/facebookresearch/faiss					

Topics relevant to SKILL DEVELOPMENT: Students will build hands-on expertise in transformer architecture engineering, semantic embedding generation, and domain-specific fine-tuning for NLP tasks. The course provides depth in using low-code tools and scalable frameworks for retrieval-based question answering, semantic similarity engines, and enterprise text classification systems. Learners gain proficiency in deploying models via ONNX and Triton, designing hybrid search stacks, and implementing prompt engineering workflows for production use. These skills are foundational for careers in applied NLP, AI product development, legal/FinTech automation, and multilingual AI systems.						
Course Code: CSS3502	Course Title: Generative Intelligence and Structured Uncertainty Modeling Type of Course: Theory Integrated Lab	L-T-P- C	2	0	2	3
Version No.	1.0					
Course Prerequisites	CSEXXXX					
Antirequisites	NIL					
Course Description	This course explores advanced generative modeling techniques and their interplay with structured uncertainty frameworks in machine learning. Students will study probabilistic generative models, variational and diffusion-based approaches, and learn to quantify model and data uncertainty using Bayesian and ensemble methods. Special emphasis is placed on applying these methods to financial risk modeling, synthetic data generation, and explainable AI, with a focus on trustworthy decision systems in volatile environments like FinTech, insurance, and fraud analytics.					
Course Objective	To enable students to design, train, and evaluate generative systems that incorporate probabilistic reasoning and uncertainty modeling in real-world scenarios involving ambiguous, sparse, or high-risk data environments.					
Course Outcomes	On successful completion of this course the students shall be able to: 1. Apply variational and diffusion-based generative models to complex data generation problems. 2. Model aleatoric and epistemic uncertainty in machine learning pipelines using Bayesian networks, MC Dropout, and ensemble techniques. 3. Evaluate uncertainty-aware generative systems for high-stakes applications in FinTech, insurance, and healthcare. 4. Deploy hybrid models that combine deterministic prediction with stochastic generation or sampling under constraints.					
Course Content:						

Module 1:	Probabilistic Generative Models and Variational Techniques	Quiz		Knowledge based Quiz	9 Sessions
Topics: Gaussian mixture models, latent variable models, Variational Autoencoders (VAEs), KL divergence, ELBO Tools: Pyro (Probabilistic Programming), TensorFlow Probability, PyTorch Use Cases: Synthetic transaction log generation for anomaly training, credit pattern modeling with missing features					
Module 2:	Diffusion Models and Stochastic Sampling	Quiz		Knowledge based Quiz	9 Sessions
Topics: Score-based generative modeling, denoising diffusion probabilistic models (DDPM), guided generation Tools: Hugging Face Diffusers, Stable Diffusion, OpenAI sample notebooks Use Cases: Synthetic KYC image generation for model robustness, document watermark recovery and completion					
Module 3:	Uncertainty Types and Quantification Methods	Quiz		Knowledge based Quiz	9 Sessions
Topics: Aleatoric vs epistemic uncertainty, MC Dropout, deep ensembles, calibration curves, predictive entropy Tools: Scikit-learn, Pyro, Deep Ensembles Toolkit (Detkit) Use Cases: Dynamic interest rate prediction under uncertainty, fraud alert confidence scoring for realtime triage					
Module 4:	Bayesian Deep Learning and Trustworthy Modeling	Quiz		Knowledge based Quiz	9 Sessions
Topics: Bayes by Backprop, variational inference in neural nets, hierarchical modeling, posterior sampling Tools: TensorFlow Probability, PyMC3, Edward2 Use Cases: Loan approval models with Bayesian posterior scoring, confidence-aware AI explanations for compliance					
Module 5: Generative AI under Constraints and Evaluation		Quiz		Knowledge based Quiz	9 Sessions

Topics: Constraint-aware generation, rejection sampling, coverage metrics (FID, IS), distribution shift handling, adversarial robustness
Tools: Alibi Detect, WhyLogs, Captum
Use Cases: Compliance-constrained synthetic data generation, safe generative augmentation pipelines for regulated domains

Project work/Assignment:

Students will build a hybrid model that generates structured financial or legal data with built-in uncertainty metrics. Example projects include: a VAE that simulates future transaction logs with confidence scores, or a diffusion-based document completion pipeline with regulatory risk flags. The project must include model calibration, uncertainty visualization, and application of at least one evaluation metric related to structured uncertainty.

Text Book

T1: Co-branded Zell and Paytm Textbook for Generative Intelligence and Structured Uncertainty Modeling

References

- Probabilistic Machine Learning – Kevin Murphy
- Deep Generative Models – Jakub Tomczak & Max Welling
- Bayesian Reasoning and Machine Learning – David Barber
- Deep Learning with PyTorch – Eli Stevens, Luca Antiga, Thomas Viehmann (Ch. on VAEs, uncertainty)

Web References

- https://huggingface.co/docs/diffusers
- https://pyro.ai/
- https://www.tensorflow.org/probabilityhttps://pymc.io/
- https://arxiv.org/abs/2107.00630 (Diffusion Models Survey)

Topics relevant to SKILL DEVELOPMENT:

This course equips students with the ability to generate data and insights from models that respect uncertainty, risk, and regulatory boundaries. Students gain exposure to probabilistic programming, advanced deep learning architectures, and stochastic inference pipelines. They will be able to critically evaluate model confidence, interpret outputs in volatile or sparse data environments, and design ethical synthetic data systems. These capabilities are critical for roles in risk analytics, data science in regulated industries, and GenAI product development.

Course Code: CAI3450	**Course Name: IoT in Agriculture** **Type of Course:** Theory Integrated Lab	L- T-P- C	2-0-2-3
Version No.	1.0		
Course Pre-requisites	CSS2264- Essential of AI		
Co-requisites	NIL		
Anti-requisites	CAI3414		

Course Description	This course introduces the fundamentals of the Internet of Things (IoT) and its applications in smart agriculture. It covers IoT architecture, embedded systems, agricultural sensors, wireless communication, cloud-based IoT platforms, and data monitoring. The course also provides an overview of AI applications in precision agriculture, including crop disease detection and smart irrigation, while emphasizing IoT security and sustainable farming practices through hands-on laboratory experiments.		
Course Objective	The objective of this course is to familiarize learners with the principles of IoT and smart agriculture by developing practical skills in sensor interfacing, cloud-based monitoring, and AI-enabled agricultural applications. The course aims to enable students to design and implement simple IoT solutions for precision farming through experiential learning.		
Course Outcomes	**CO1:** Explain the fundamentals of IoT architecture, embedded systems, sensors, actuators, and their applications in smart agriculture. — Level 2 – Understand **CO2:** Apply agricultural sensors, wireless communication technologies, and microcontrollers to acquire and monitor real-time agricultural data. — Level 3 – Apply **CO3:** Develop IoT-based smart agriculture applications using cloud platforms for data collection, visualization, and remote monitoring. — Level 3 – Apply **CO4:** Apply basic AI and machine learning techniques for crop disease detection and other precision agriculture applications. — Level 3 – Apply **CO5:** Analyze IoT security issues, sustainability aspects, and emerging trends in smart agriculture for developing secure and efficient IoT solutions. — Level 4 – Analyze		
Course Content:			
Module 1	Introduction to IoT and Smart Agriculture	**Assessment:** Module Test	L-6 P-4 10 sessions
Introduction to IoT, IoT architecture, enabling technologies, sensors and actuators, embedded systems, microcontrollers, applications of IoT			
Module 2	Smart Agriculture and IoT Devices	**Assessment:** Module Test	L-6 P-6 12 Sessions
Introduction to smart agriculture, Smart agriculture ecosystem, precision agriculture, challenges in conventional farming, digital farming, and agriculture automation. Agricultural sensors, wireless communication technologies, sensor interfacing, data acquisition, IoT gateway, power management.			
Module 3	Cloud IoT and Data Monitoring	**Assessment:** Module Test	L-6 P-8 14 Sessions
Introduction to cloud computing for IoT, ThingSpeak, Blynk and Node-RED, cloud data collection and storage, dashboards, real-time monitoring, REST APIs, overview of edge computing.			
Module 4	AI-enabled Precision Agriculture		L-8 P-8 16 Sessions
Introduction to AI and Machine Learning in agriculture, crop health monitoring, disease detection, irrigation prediction, fertilizer recommendation, yield prediction, drone-based monitoring, satellite imagery, computer vision applications, digital twins for agriculture, autonomous farming systems.			
Module 5	IoT Security and Future Trends		L-4 P-4 8 Sessions
IoT security basics, authentication, encryption, privacy issues, sustainable agriculture, AIoT, smart farming,			

future trends in agriculture.	
List of laboratory tasks:	
Experiment 1 Setting up ESP32 Development Environment Objective: Install and configure Arduino IDE for ESP32 programming. **Experiment 2** Reading Temperature and Humidity Data Objective: Interface DHT11/DHT22 sensor with ESP32. **Experiment 3** Soil Moisture Monitoring System Objective: Measure soil moisture using a soil moisture sensor. **Experiment 4** Smart Irrigation System Objective: Automate irrigation based on soil moisture. **Experiment 5** Light Intensity Monitoring Objective: Interface LDR sensor for ambient light detection. **Experiment 6** IoT Cloud Integration using ThingSpeak Objective: Upload agricultural sensor data to the cloud. **Experiment 7** Mobile Monitoring using Blynk IoT Objective: Monitor agricultural sensors remotely. **Experiment 8** Agricultural Data Visualization Objective: Analyze sensor data collected over time. **Experiment 9** Farm Monitoring Dashboard Objective: Develop a real-time agricultural monitoring dashboard. **Experiment 10** Crop Disease Detection using AI Objective: Detect crop diseases using image classification. **Mini Project – Smart Agriculture System** Objective: Develop a complete IoT-enabled smart farming prototype.	
Targeted Applications & Tools:	
Tools/Software/Platform	Application
Arduino IDE	Programming and configuring ESP32 development boards
ESP32 Development Board	Development of IoT-based smart agriculture applications
DHT11/DHT22 Sensor	Temperature and humidity monitoring
Soil Moisture Sensor	Soil moisture measurement for irrigation management
LDR Sensor	Light intensity monitoring for crop environment analysis
Relay Module	Automatic control of irrigation pumps and actuators
ThingSpeak	Cloud-based data acquisition, storage, and visualization
Blynk IoT Platform	Mobile-based remote monitoring and control of IoT devices
MQTT (HiveMQ/Mosquitto)	Publish-subscribe communication between IoT devices
Python (Jupyter Notebook / Google Colab)	Agricultural data analysis, visualization, and machine learning

TensorFlow / Keras	AI-based crop disease detection using image classification
OpenWeather API	Integration of weather forecast data with IoT applications
Node-RED	Development of real-time farm monitoring dashboards

Skill-Building Activities through Experiential Learning:

Smart Irrigation System: Students will develop an automatic irrigation system that monitors soil moisture and controls water pumps using IoT.
Crop Disease Detection: Students will develop an AI-enabled crop disease detection system using image classification.
Weather-based Irrigation Prediction: Students will integrate weather APIs with IoT sensor data to optimize irrigation schedules.
Precision Farming Dashboard: Students will build a real-time dashboard displaying agricultural sensor data, alerts, and recommendations.

Text Book

T1: Arshdeep Bahga & Vijay Madisetti, *Internet of Things – A Hands-on Approach*, Universities Press, 2021.

T2: Rajkumar Buyya & Amir Vahid Dastjerdi, *Internet of Things: Principles and Paradigms*, Elsevier.

T3: Olivier Hersent, David Boswarthick & Omar Elloumi, *The Internet of Things: Key Applications and Protocols*, Wiley.
T4: Cuno Pfister, *Getting Started with the Internet of Things*, O'Reilly.

References

R1: Daniel Minoli, *Building the Internet of Things with IPv6 and MIPv6*, Wiley.
R2: Vijay Madisetti, *Internet of Things: Smart Environments and Smart Cities*, Springer.
R3: Research articles on Smart Agriculture, Precision Farming, AI-enabled IoT, and Wireless Sensor Networks from IEEE Xplore, Springer, and Elsevier.

E-Resources/Web Resources

1. **Arduino Official Documentation**
   https://docs.arduino.cc/
2. **Espressif ESP32 Documentation**
   https://docs.espressif.com/projects/esp-idf/en/latest/esp32/
3. **ThingSpeak IoT Platform**
   https://thingspeak.com/
4. **Blynk IoT Documentation**
   https://docs.blynk.io/
5. **HiveMQ MQTT Essentials (MQTT Tutorials)**
   https://www.hivemq.com/mqtt-essentials/
6. **Node-RED Official Documentation**
   https://nodered.org/docs/
7. **TensorFlow Tutorials**
   https://www.tensorflow.org/tutorials
8. **OpenCV Documentation**
   https://docs.opencv.org/
9. **OpenWeather API Documentation**
   https://openweathermap.org/api
10. **Kaggle – Agriculture and Crop Datasets**
    https://www.kaggle.com/datasets
11. **IEEE Internet of Things Journal**

https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=6488907

12. **IoT For All – Industrial and Agricultural IoT Resources**
    https://www.iotforall.com/
13. **Cisco Networking Academy – Introduction to IoT**
    https://www.netacad.com/courses/iot

Course Code: CAI 3451	Course Name: Artificial Intelligence in Agriculture Type of Course: Theory Integrated Lab	L- T-P- C	2-0-2-3
Version No.	1.0		
Course Pre-requisites	CSS2264- Essential of AI		
Co-requisites	NIL		
Anti-requisites			
Course Description	This course introduces students to the application of Artificial Intelligence (AI) technologies in modern agriculture. It explores how AI enhances agricultural productivity through precision farming, intelligent crop monitoring, soil analysis, agricultural robotics, and data-driven decision-making. Students will gain practical exposure to AI tools used for sustainable farming, resource optimization, and automation of agricultural operations.		
Course Objective	• The objective of this course is to familiarize learners with AI technologies that transform agricultural practices by improving productivity, sustainability, and resource management through intelligent systems and participative learning techniques. • Explore the foundational concepts of Artificial Intelligence as they pertain to agriculture and farming, understanding the transformative impact of AI on modern agricultural practices. • Learn how AI technologies enable precision agriculture, allowing farmers to optimize resource use and enhance crop yield through data-driven decision-making. • Discover the applications of AI in monitoring soil health, assessing crop conditions, and implementing effective management strategies to improve productivity and sustainability. • Examine the role of AI in agricultural robotics, including the automation of tasks such as planting, harvesting, and monitoring, and understand how these technologies are revolutionizing modern farming.		
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Explain the fundamental concepts of Artificial Intelligence and its applications in modern agriculture. — [Level 2-Understand] **CO2**: Apply AI techniques for precision agriculture to optimize crop production, — [Level 3-Apply]		

	irrigation, and resource utilization.	
	CO3: Analyze agricultural data for crop monitoring, soil health assessment, disease detection, and yield prediction using AI techniques.	[Level4-Analyze]
	CO4: Apply AI-based robotic and automation technologies for agricultural operations such as planting, harvesting, and field monitoring.	[Level 3-Apply]
	CO5: Analyze AI-enabled agricultural systems to evaluate their effectiveness in improving productivity, sustainability, and resource management.	[Level4-Analyze]

Course Content:

Module 1	Introduction to AI in Agriculture and Farming	Quiz	Knowledge based Quiz	12 [7L + 5P]
Topics: Introduction to Artificial Intelligence, Evolution of Smart Agriculture, Digital Agriculture and Agriculture 4.0, AI Technologies in Farming, Benefits and Challenges of AI Adoption, Future Trends in AI-driven Agriculture				
Module 2	AI for Precision Agriculture	Quiz	Comprehension based quiz	18 [10L + 8P]
Topics: Precision Agriculture Concepts, Remote Sensing and GIS, IoT Sensors in Agriculture, AI-based Crop Monitoring, Smart Irrigation Systems, Yield Prediction Models				
Module 3	AI in Crop and Soil Management	Quiz	Comprehension based quiz	18 [10L + 8P]
Topics: Soil Health Monitoring, Crop Disease Detection using AI, Nutrient Recommendation Systems, Pest Detection and Management, Weather Forecast-based Crop Planning, Sustainable Crop Management				
Module 4	AI for Agricultural Robotics and Automation	Assignment	Batch-wise Assignments	12 [7L + 5P]
Topics: Agricultural Robotics, Autonomous Tractors, Harvesting Robots, AI-powered Weed Detection, Drone-based Field Monitoring, Future of Agricultural Automation				

List of Laboratory Experiments

Lab 1: Python Environment Setup

Objective: Familiarize students with the programming environment.

Tasks:

- Install Python and Jupyter Notebook/Google Colab.
- Install NumPy, Pandas, Matplotlib, and Scikit-learn.
- Write a simple Python program to display agricultural information.

Lab 2: Exploring Agricultural Datasets

Objective: Understand agricultural datasets.

Tasks:

- Load a CSV dataset.
- Display rows and columns.

- Find summary statistics.
- Identify missing values.

Lab 3: Data Preprocessing
Objective: Clean agricultural datasets.
Tasks:

- Remove duplicate records.
- Handle missing values.
- Normalize numerical features.
- Save the cleaned dataset.

Lab 4: Data Visualization

Objective: Visualize agricultural data.
Tasks:

- Plot bar charts.
- Plot line graphs.
- Create pie charts.
- Draw scatter plots for rainfall vs crop yield.

Lab 5: Weather Data Analysis
Objective: Analyze weather conditions affecting agriculture.
Tasks:

- Import rainfall and temperature data.
- Calculate averages.
- Plot monthly rainfall.
- Identify seasonal trends.

Lab 6: Crop Yield Prediction using Linear Regression
Objective: Build a simple prediction model.
Tasks:

- Load crop dataset.
- Train a Linear Regression model.
- Predict crop yield.
- Calculate prediction accuracy.

Lab 7: Crop Recommendation System
Objective: Recommend suitable crops.
Tasks:

- Load soil and weather dataset.
- Train a Decision Tree classifier.
- Predict the best crop.
- Test with sample inputs.

Lab 8: Soil Nutrient Analysis
Objective: Analyze soil characteristics.
Tasks:

- Read soil nutrient dataset.
- Identify nutrient deficiencies.
- Classify soil quality.
- Recommend fertilizers.

Lab 9: Smart Irrigation Recommendation
Objective: Automate irrigation decisions.
Tasks:

- Load soil moisture dataset.
- Define irrigation threshold.
- Recommend irrigation schedule.
- Display recommendations.

Lab 10: Fertilizer Recommendation System
Objective: Recommend fertilizers using AI.
Tasks:

- Analyze soil nutrients.
- Train a simple classifier.
- Predict fertilizer type.
- Evaluate prediction results.

Lab 11: Plant Disease Detection using Image Processing
Objective: Perform basic image classification.
Tasks:

- Load plant leaf images.
- Resize images.
- Display healthy and diseased leaves.
- Classify images using a pre-trained CNN model.

Lab 12: Image Processing using OpenCV
Objective: Learn basic image operations.
Tasks:

- Read images.
- Convert to grayscale.
- Detect image edges.
- **Display processed images.**

Lab 13: Pest Detection
Objective: Detect pests from crop images.
Tasks:

- Load pest image dataset.
- Detect objects using OpenCV.
- Count detected pests.
- Display results.

Lab 14: Crop Health Monitoring
Objective: Monitor crop conditions.
Tasks:
Analyze crop images.
Calculate vegetation indices.
Detect unhealthy crops.
Visualize affected regions.
Lab 15: Drone Image Analysis
Objective: Analyze aerial crop images.
Tasks:

- Load drone images.
- Perform image enhancement.
- Identify crop regions.
- Compare healthy and unhealthy areas.

Lab 16: IoT Sensor Data Analysis
Objective: Analyze agricultural sensor data.

Tasks:

- Load sensor readings.
- Plot sensor values.
- Detect abnormal readings.
- Generate summary report.

Lab 17: Agricultural Chatbot (Simple Rule-Based)
Objective: Develop a simple chatbot.
Tasks:

- Create question-answer pairs.
- Accept farmer queries.
- Display appropriate responses.
- Test chatbot performance.

Lab 18: Agricultural Data Dashboard
Objective: Visualize agricultural information.
Tasks:

- Import crop dataset.
- Create interactive charts.
- Display production statistics.
- Generate dashboard report.

Lab 19: Mini Project Development
Objective: Develop an AI application for agriculture.
Tasks:

- Select a project topic.
- Prepare dataset.
- Train AI model.
- Test application.

Lab 20: Project Demonstration and Result Analysis
Objective: Evaluate and present project outcomes.
Tasks:

- Demonstrate the developed system.
- Measure model performance.
- Visualize results.
- Present findings and future scope.

Targeted Applications & Tools: Python, Jupyter Notebook, Google Colab, TensorFlow, Scikit-learn, OpenCV, Roboflow, Kaggle , Agricultural Datasets

Project work/Assignment:

- Crop Yield Prediction
- Crop Recommendation System
- Pest Detection
- Weather-Based Crop Advisory
- Agricultural Chatbot
- Smart Farming Dashboard

Text Book

T1: U. Kose, V. B. Surya Prasath, M. Rubaiyat Hossain Mondal, P. Podder, and S. Bharati, Eds., *Artificial Intelligence and Smart Agriculture Applications.* Boca Raton, FL, USA: Auerbach Publications, CRC Press, 2023
T2: R. K. Naresh, S. S. Walia, N. Ravisankar, R. K. Gupta, et al., *Artificial Intelligence in Agriculture: Revolutionising Crop Management.* New Delhi, India: New India Publishing Agency (NIPA), 2025.

References
R1: Q. Zhang, Ed., Precision Agriculture Technology for Crop Farming. Boca Raton, FL, USA: CRC Press, 2015. R2: D. P. Mishra, Artificial Intelligence and IoT in Agriculture, Springer.
E-Resources/Web Resources
1. https://www.naarm.org.in/wp-content/uploads/2024/02/MOOC-on-Artificial-Intelligence-in-Agriculture.pdf 2. https://nanoschool.in/aipg/ai-in-agriculture/

Course Code: CAI3452	**Course Name: Crop Monitoring and Disease Detection using AI** **Type of Course:** Theory Integrated Lab	**L- T-P-C**	2-0-2-3
Version No.	1.0		
Course Pre-requisites	CSS2264- Essential of AI		
Co-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course introduces students to the practical application of Artificial Intelligence and Computer Vision in modern agriculture. Designed as an easy-to-follow, hands-on framework, it covers how images are collected from farms (using cameras and drones), how to clean and prepare those images using Python, and how to build simple machine learning models to automatically classify plant diseases. Students will get direct experience working with simple tools like OpenCV and PyTorch to create real-world agritech apps.		
Course Objective	To introduce undergraduate students to fundamental image processing, data preparation, and introductory deep learning techniques used to build automated farming and plant health tracking systems.		
Course Outcomes	On successful completion of the course the students shall be able to: **CO1:** Describe how digital cameras, drones, and simple spectral sensors capture crop data. [Level 2- Understand] **CO2:** Write basic Python scripts using OpenCV to remove backgrounds and highlight leaf spots. [Level 3- Apply] **CO3:** Crop, organize, label, and balance raw field images to create clean datasets.. [Level 3- Apply] CO4: Understand and use a Convolutional Neural Network (CNN) to sort healthy crops from diseased crops [Level 3- Apply] **CO5**: Analyze a basic web/script-based [Level 4- Analyze]		

	prototype that takes an image and outputs a disease diagnosis.

Course Content:

Module 1	**Introduction to Smart Farming and Farm Data**	**Assessment: Continuous Evaluation / Laboratory Lab Assignment**	**6 Sessions**
Introduction to Smart Farming and Precision Agriculture; Concepts of Crop Monitoring Systems; Image Data Acquisition Platforms; Cameras, Smartphones, and Drones in farming; Physics of Vegetation Light Reflection; Spectral Color Bands; Healthy vs. stressed plant tissue reflectance; Benchmarking Agritech Data Repositories; The *PlantmVillage* Dataset infrastructure.			
Module 2	**Image Cleaning and Leaf Spot Highlighting**	**Practical Assessment & Quiz**	**6Sessions**
Environmental Noise in Field Imagery; Shadows, Varying Sunlight, and Soil Clutter; Digital Image Restoration Techniques; Brightness Adjustment; Image Cropping; Spatial Sharpening; Analytical Color-Space Conversions; RGB to HSV and $L^*a^*b^*$ Mapping; Plant Pathology Visual Extraction; Principles of Binary Masking; Foliage Background Separation.			
Module 3	**Organizing and Preparing Your Datasets**	**Portfolio Submission / Data Pipeline Review**	**6 Sessions**
Foundations of Agritech Data Engineering; Data Quality Bottlenecks; Automated Structural Image Cropping; File Sorting Pipelines; Bounding Box Annotation Mechanics; Labeling Tool Integration; Bounding Coordinate Verification; Rectification of Class Imbalances; Synthetic Data Augmentation; Multi-Axis Rotations and Affine Warps.			
Module 4	**Building Your First AI Crop Diagnoser**	**Assessment: Programming Challenge & Performance Matrix Audit**	**6 Sessions**
Transition from Classical Computer Vision to Autonomous Artificial Intelligence; Feature Learning Maps in Deep Networks; Edge, Texture, and Spatial Color Extractors; Structural Components of Convolutional Neural Networks (CNNs); Convolution Layers; Pooling Layers; Dense Layers; Statistical Model Performance Metrics; Accuracy; Precision; Recall; Confusion Matrices.			
Module 5	**Using Smart Models and Creating Your App Prototype**	**Assessment: Integrated Mini-Project Defense & Live Demonstration**	**6 Sessions**
Concepts of Deep Transfer Learning; Feature Reusability; Fine-Tuning Lightweight Convolutional Topologies; MobileNet Architectures for Targeted Vegetation Taxonomy; Resource-Constrained Microcontroller Constraints; Fundamentals of Real-Time Object Detection Frameworks; YOLO and SSD frameworks for Localized Anomaly Mapping; Architectural Layouts for Applied Graphical User Interfaces; Front-End Logic Pipelines.			

Lab Sheets

Lab 1: Environment Setup and Library Verification Configure Python vision environments and verify installation of core agritech frameworks (OpenCV, Pillow, Matplotlib).

Lab 2. Dataset Loading and Visual Exploration Build scripts to programmatically access, crop, resize, and display multi-band image samples from plant disease repositories.

Lab 3. Crop Color Index Computation Design a custom processing pipeline to calculate image greenness indicators for regional plant health assessment.
Lab 4. Soil Background Extraction Implement automated pixel-masking scripts to cleanly isolate live green leaf tissue from ambient background dirt.
Lab 5. Spatial Illumination Equalization Apply spatial image filters to normalize field lightning imbalances and mitigate sharp environmental shadow noise.
Lab 6. Foliage Disease Spot Quantification Develop an algorithmic tool to calculate the exact structural area ratio of active leaf lesions compared to healthy green tissue.
Lab 7. Interactive Bounding Annotation Utilize open-source data annotation tools to generate coordinate targets and bounding box overlays over crop defects.
Lab 8. Annotation Alignment Auditing Write validation scripts to parse annotation files for labeling placement errors and structural misalignments.
Lab 9. Multi-Axis Data Augmentation Construct geometric and spatial transformation sequences to mathematically balance mismatched data folders.
Lab 10. Multi-Class Crop Classification Setup Compile and initialize a 3-layer Convolutional Neural Network (CNN) framework to differentiate healthy and infected samples.
Lab 11. Diagnostic Matrix Evaluation Generate localized confusion matrices and track validation accuracy profiles across various vegetation categories.
Lab 12. Pre-Trained Model Adaption Execute a deep transfer learning sequence by adapting a pre-trained, lightweight MobileNet architecture onto local crop files.
Lab 13. Deep Architecture Parameter Tuning Fine-tune model layers and adjust learning configurations to optimize classification boundaries.
Lab 14. Target Localization Optimization Configure lightweight object detection layers to isolate specific pest zones or early-stage anomalies on live feeds.
Lab 15. Mini Project: Automated Crop Assessment Interface End-to-end integration of image cleaning, deep classification inference, mapping vectors, and structural health metric readouts**.**

Targeted Applications & Tools:

Python (NumPy, pandas)	Data preprocessing, analysis, and numerical computation.
TensorFlow and PyTorch	Building and training deep learning models such as CNNs and neural networks. Flexible deep learning framework for research and experimentation.
OpenAI Gym / Gymnasium	Simulation environments for training RL agents
Stable-Baselines3	Pre-built implementations of RL algorithms
Matplotlib / Seaborn	Visualization of learning performance
Jupyter Notebook / Google Colab	Interactive environments for implementing, testing, and documenting machine learning experiments with built-in visualization and hardware acceleration.

Skill-Building Activities through Experiential Learning:

- **Agritech Data Cleaning & Balancing: Students write specialized image normalization matrices to resolve data imbalances across rare disease sets.**
- **Spectral Spatial Segmentation Engineering: Learners isolate green foliage signatures from high-dimensional matrices, stripping out background soil artifacts.**
- **Convolutional Pathology Network Design: Students implement custom multi-layer CNNs to auto-detect subtle structural changes in leaves.**
- **Multimodal Sensor Stream Synthesis: Learners merge visual machine learning model inferences with raw IoT environmental numbers to build unified predictive health readouts.**

Textbooks

T1: Ian Goodfellow, Yoshua Bengio, Aaron Courville – *Deep Learning*, MIT Press.

T2: Marcus Borengasser, Edward F. Hungate, Russell Watkins – *Hyperspectral Remote Sensing: Principles and Applications*, CRC Press.

References

R1: François Chollet – *Deep Learning with Python*, Manning Publications.

R2: Aurélien Géron – *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, O'Reilly.
R3:*Kanish Shah et al. – "Crop protection and disease detection using artificial intelligence and computer vision: a comprehensive review", Springer Nature.*

E-Resources/Web Resources

1. https://onlinecourses.nptel.ac.in/noc22_ag05/preview
2. OpenCV Official Interactive Documentation
3. PyTorch Vision Models Library Platforms
4. TensorFlow Core Agritech Classification Guides

Course Code: CAI3453	**Course Name: Edge AI for Crop Monitoring** **Type of Course:** Theory Integrated Lab	**L- T-P- C**	2-0-2-3
Version No.	1.0		
Course pre-requisites	CSS2264- Essential of AI		
Co-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course introduces the programming and deployment of lightweight artificial intelligence models for automated crop monitoring. Students will learn how to write Python routines to process crop imagery, simulate localized data matrices, detect plant leaf diseases, and optimize model performance for edge-device deployment. The curriculum emphasizes hands-on programming in cloud-based sandboxes to build efficient, resource-optimized agricultural inference scripts without complex mathematical or hardware constraints..		
Course Objective	To enable students to program and optimize edge-ready machine learning components that analyze visual and tabular crop data to automate plant health surveillance.		

Course Outcomes	**On successful completion of the course, the students shall be able to:** **CO1:** Explain Edge AI concepts and local inference workflows for specialized crop monitoring. (Understand) **CO2:** Write Python code to handle, preprocess, and analyze tabular crop health data arrays. (Understand) **CO3:** Implement computer vision scripts to classify plant leaf diseases using pre-trained model APIs. (Apply) **CO4:** Apply model compression and quantization programs to reduce agricultural model storage and execution sizes. (Apply) **CO5:** Analyze compressed AI pipeline scripts for runtime efficiency, error tracking, and data safety compliance. (Analyze)

Course Content:

Module 1	**Python Data Pipelines for Crop Metrics**	**6 Sessions**
Edge AI architectures for localized crop health tracking, Edge hardware memory constraints and array storage, Python script structures for loading and cleaning tabular data, Parsing crop metrics: hydration, soil nutrients, and micro-climate values, Data-driven conditional statements for automated farm alerts.		
Module 2	**Programming Visual Crop Disease Classifiers**	**6 Sessions**
Digital image formats and representations in agriculture, Python computer vision loops for leaf image resizing and normalization, Integrating pre-trained image classification blocks via APIs, Evaluating model confidence scores for automated diagnosis, Algorithmic identification: leaf-spots, pest damages, and leaf deficiencies		
Module 3	**Code Scaffolding for Automated Yield Analytics**	**6 Sessions**
Rule-based programming logic for agricultural automation, Structuring Python data loops to receive live multi-channel inputs, AI-assisted code generation for script scaffolding and boilerplate logic, Automated verification pipelines for user-entered farm observations,Writing validation functions for predictable model text outputs		
Module 4	**Crop Model Optimization and Quantization**	**6 Sessions**
Processing bottlenecks on edge devices: RAM, CPU, and battery caps, Introduction to model compression: weights, pruning, and tuning, Post-training quantization programming steps using Python, Exporting neural network models into .tfliteand .onnx formats		
Module 5	**Script Validation, Security, and Project Integration**	**6 Sessions**
Debugging exception codes in optimized edge deployment scripts, Data security protocols: Regex patterns for masking GPS farm coordinates, Managing training dataset bias and handling input data anomalies Mini-Project: Integration of local tabular alerts and visual crop classification scripts		

List of Laboratory Tasks: (30 hours)

Lab 1: Memory Footprint Mapping – Write a basic Python script to initialize a local crop data list and print its memory size.

Lab 2: Crop Table Feature Selection – Load a simple 20-row mock dataset of farm values and write code to filter and print specific columns (e.g., Hydration vs. Soil Nutrients).

Lab 3: Condition-Based Farm Alerts – Code a Python script using basic if-else blocks to print automated alert strings when a crop metric falls below a critical threshold.

Lab 4: Data Clean-up Filters – Write a script to scan a list of raw sensor readings and automatically replace missing or invalid values (zeros) with a baseline average.

Lab 5: Automated Image Resizing – Write 5 lines of Python code to load a crop leaf image, display its resolution, and downsize it for resource-constrained setups.

Lab 6: Running a Disease Classifier API – Write a script to send an image of a leaf to a pre-trained image identification API and print out the predicted disease class.

Lab 7: Extracting Model Confidence Scores – Program a loop to look at an image classifier's confidence output array and flag labels that fall below a 70% threshold.

Lab 8: AI-Assisted Script Scaffolding – Use an AI code generator to scaffold a clean Python script template for validating user-logged farm observations.

Lab 9: Text-to-Rule Logic Synthesis – Use an AI engine to convert an English operational rule (e.g., *"If soil moisture is under 40%, flag critical status"*) into functional Python code.

Lab 10: Multi-Channel Input Aggregation – Write a simple Python function that combines tabular text flags and numeric moisture arrays into a single readable data structure.

Lab 11: Post-Training Model Quantization – Take a standard pre-built agricultural neural network and execute a 3-line compression script to reduce its file size.

Lab 12: Performance & Execution Benchmarking – Code a simple execution timer using the time library to measure and print the millisecond difference between the original and compressed models.

Lab 13: Privacy Masking for Farm Logs – Develop a basic script using text replacement to locate GPS farm coordinates in a log file and replace them with masking characters.

Lab 14: Exception Handling in Crop Pipelines – Insert a try-except block into an inference script to prevent the program from crashing if a corrupted data value is passed.

Lab 15: Integrated Mini Project – Combine your optimized image classification function (Module 2/4) and tabular alert script (Module 1/3) into a single, clean command-line script execution.

Targeted Applications & Tools:	
Environments & Runtimes:	Google Colab, Python 3.x, VS Code

AI Libraries	Lightweight TensorFlow Lite Runtime, ONNX Runtime Basics, OpenCV/PIL for image manipulation.
AI Code Assistants	GitHub Copilot, Gemini Web Interfaces, Cursor

Skill-Building Activities through Experiential Learning:

- Writing Python code to handle and parse tabular crop metrics.
- Implementing automated, data-driven crop health alert systems.
- Building visual image preprocessing and crop disease classification scripts.
- Using AI tools for model quantization, debugging, and code optimization.
- Integrated mini-project development for local edge monitoring.

Textbooks

T1: Dhaygude, Suman Kumar Swarnkar, Priya Chugh, & Yogesh Kumar Rathore, *Smart Agriculture: Harnessing Machine Learning for Crop Management*, 1st Edition, CRC Press, 2024.
T2: Jen-Tsung Chen, *Machine Learning for Plant Biology*, 1st Edition, John Wiley & Sons, 2026.
T3: Pete Warden & Daniel Situnayake, *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*, 1st Edition, O'Reilly Media, 2020.

References

R1: Veronica Saiz-Rubio, Francisco Rovira-Mas, and Chittaranjan Kole, *Smart Agriculture: Concepts, Strategies, and Case Studies*, 1st Edition, CRC Press, 2026.
R2: Pete Warden and Daniel Situnayake, *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*, 1st Edition, O'Reilly Media, 2020

E-Resources / Web Resources

https://colab.research.google.com/
https://www.tensorflow.org/lite
https://onnxruntime.ai/

Course Code: CAI3454	Course Name: Smart Agriculture and Precision Farming Type of Course: Theory Integrated Lab	L- T-P- C	2-0-2-3
Version No.	1.0		
Course Pre-requisites	CSS2264- Essential of AI		
Co-requisites	NIL		
Anti-requisites	-		
Course Description	Smart Agriculture and Precision Farming is an interdisciplinary course that explores the application of advanced digital technologies to improve agricultural productivity, sustainability, and resource efficiency. The course covers the fundamentals of precision agriculture, including Global Positioning Systems (GPS), Geographic Information Systems (GIS), remote sensing, Internet of Things (IoT), wireless sensor networks, drones, artificial		

<table>
<tr><td></td><td colspan="3">intelligence (AI), machine learning, and data analytics. Students will gain an understanding of how these technologies enable data-driven decision-making for crop monitoring, irrigation management, soil health assessment, pest and disease detection, yield prediction, and farm automation. Through case studies and practical applications, the course prepares learners to design and implement smart farming solutions that address modern agricultural challenges while promoting sustainable and climate-resilient farming practices.</td></tr>
<tr><td>Course Objective</td><td colspan="3">The objective of this course is to provide students with a comprehensive understanding of smart agriculture and precision farming by introducing modern digital technologies such as IoT, GPS, GIS, remote sensing, drones, artificial intelligence, machine learning, and data analytics for agricultural applications. The course aims to develop the ability to apply these technologies for precision crop management, efficient utilization of resources, real-time monitoring, and data-driven decision-making while promoting sustainable, climate-resilient, and technology-enabled agricultural practices to address the challenges of modern farming.</td></tr>
<tr><td>Course Outcomes</td><td colspan="3">On successful completion of this course, the students shall be able to:
CO1: Explain the fundamental concepts, principles, and technologies of smart agriculture and precision farming, including IoT, GPS, GIS, and remote sensing. (L2 – Understand)
CO2: Apply geospatial technologies, sensor networks, and IoT-based systems to monitor agricultural resources and support precision farming practices. (L3 – Apply)
CO3: Develop basic machine learning and data analytics models for crop yield prediction, disease detection, and smart irrigation using Python. (L3 – Apply)
CO4: Implement data-driven decision support applications using agricultural datasets, visualization tools, and AI techniques for precision farm management. (L3 – Apply)
CO5: Analyze the effectiveness of smart farming technologies and evaluate sustainable precision agriculture solutions for improving productivity, resource utilization, and environmental conservation. (L4 – Analyze)</td></tr>
<tr><td colspan="4">Course Content:</td></tr>
<tr><td>Module 1</td><td>Fundamentals of Smart Agriculture</td><td>Assessment: Module Test</td><td>L-6 P-6
12 sessions</td></tr>
<tr><td colspan="4">Topics:
Evolution of agriculture: Traditional, Precision, and Smart Agriculture
Concepts and principles of Precision Farming
Agriculture 4.0 and Digital Agriculture
Challenges in modern agriculture
Components of smart farming systems
Benefits, limitations, and applications of precision agriculture</td></tr>
<tr><td>Module 2</td><td>Geospatial and Sensing Technologies</td><td>Assessment: Module Test</td><td>L-6 P-6
12 Sessions</td></tr>
<tr><td colspan="4">Topics:
Global Positioning System (GPS)
Geographic Information System (GIS)
Remote sensing and satellite imagery
Soil sampling and nutrient mapping
Variable Rate Technology (VRT)
Yield monitoring and field mapping
Spatial variability in crop production</td></tr>
<tr><td>Module 3</td><td>IoT and Automation in</td><td>Assessment: Quiz</td><td>L-6 P-6</td></tr>
</table>

	Agriculture		12 Sessions

Topics:

- Internet of Things (IoT) architecture for agriculture
- Agricultural sensors: soil moisture, temperature, humidity, pH, and weather
- Wireless Sensor Networks (WSN)
- Smart irrigation and fertigation systems
- Cloud and edge computing for agriculture
- Farm automation and smart monitoring systems

Module 4	Artificial Intelligence and Data Analytics	Assessment: Module Test	L-6 P-6 12 Sessions

- Introduction to Artificial Intelligence (AI) in agriculture
- Machine learning for crop prediction and yield estimation
- Computer vision for crop health and disease detection
- Drone and UAV-based crop monitoring
- Big data analytics and decision support systems
- Predictive analytics for precision farming

Module 5	Smart Farming Applications and Future Trends	Assessment: Module Test	L-6 P-6 12 Sessions

- Precision irrigation and nutrient management
- Smart greenhouse and protected cultivation
- Livestock monitoring and management
- Agricultural robotics and autonomous farm machinery
- Climate-smart agriculture and sustainable farming
- Emerging technologies: Blockchain, Digital Twins, and Autonomous Farming Systems

List of laboratory tasks:

Experiment 1: Analysis of Weather Data

- Read agricultural weather data (temperature, humidity, rainfall) using Pandas.
- Calculate mean, maximum, minimum, and standard deviation.
- Visualize trends using Matplotlib.

Experiment 2: Soil Moisture Data Visualization

- Import soil moisture sensor data from a CSV file.
- Plot moisture levels over time.
- Identify dry and wet periods using threshold values.

Experiment 3: Crop Yield Prediction Using Linear Regression

- Load a dataset containing rainfall, fertilizer usage, and crop yield.
- Train a simple Linear Regression model.
- Predict crop yield for new input values.

Experiment 4: Crop Disease Image Classification

- Load sample crop leaf images.
- Perform basic image preprocessing.
- Classify healthy and diseased leaves using a pre-trained CNN model.

Experiment 5: Rainfall Trend Analysis

- Analyze historical rainfall data.
- Generate monthly and yearly rainfall graphs.
- Identify rainfall patterns affecting agriculture.

Experiment 6: Smart Irrigation Decision System

- Develop a Python program that recommends irrigation based on:
  - Soil moisture
  - Temperature
  - Humidity
- Display irrigation recommendations.

Experiment 7: Fertilizer Recommendation System

- Create a rule-based Python program.
- Recommend fertilizer based on:
  - Soil pH
  - Nitrogen level
  - Phosphorus level
  - Potassium level

Experiment 8 : IoT Sensor Data Monitoring

- Read simulated sensor values from a CSV file.
- Monitor temperature, humidity, and soil moisture.
- Generate alerts when values exceed predefined limits.

Experiment 9: Crop Production Data Analysis

- Analyze district-wise or state-wise crop production data.
- Identify the highest-yielding crops using Python.
- Create bar charts and pie charts.

Targeted Applications & Tools:

Tools/Software/Platform	Application
Python (Anaconda / Jupyter Notebook / Google Colab)	Data analysis, visualization, and development of smart agriculture applications
Pandas & NumPy	Processing and analyzing agricultural datasets
Matplotlib & Plotly	Visualization of weather, soil, and crop data
Scikit-learn	Machine learning for crop yield prediction, disease classification, and decision support
OpenCV	Image processing for crop monitoring and disease detection
TensorFlow / Keras	Deep learning models for plant disease identification and crop classification
Streamlit / Dash	Development of interactive smart agriculture dashboards
Google Earth Engine / QGIS	Remote sensing, GIS analysis, and agricultural land mapping
ThingSpeak / Blynk (Optional)	IoT sensor data visualization and cloud-based monitoring
DroneDeploy / Pix4D (Optional)	Drone image processing and precision field analysis

Skill-Building Activities through Experiential Learning:

- **Smart Irrigation Decision Support System**
  Develop a Python application that recommends irrigation schedules based on soil moisture, temperature, humidity, and rainfall data.
- **Crop Yield Prediction Model**
  Build a machine learning model using historical agricultural data to predict crop yield and evaluate prediction accuracy.
- **Plant Disease Detection System**
  Design an image-based application using OpenCV and TensorFlow/Keras to identify common

crop diseases from leaf images.

- **Agricultural Data Analytics Dashboard**
  Create an interactive dashboard using Streamlit or Dash to visualize weather conditions, soil parameters, and crop performance.
- **Precision Farming Data Analysis Project**
  Analyze GPS/GIS and remote sensing datasets to identify field variability and recommend precision farming practices.
- **IoT-Based Farm Monitoring Simulation**
  Simulate real-time monitoring of environmental parameters using sensor datasets and generate alerts for abnormal conditions.
- **Weather-Based Crop Advisory System**
  Develop a Python application that provides crop management recommendations based on weather forecasts and historical climate data.
- **Fertilizer Recommendation System**
  Implement a rule-based or machine learning-based application that recommends fertilizers using soil nutrient and pH data.
- **Remote Sensing Image Classification**
  Process satellite or drone imagery to classify vegetation and assess crop health using Python image processing libraries.
- ·**Mini Project: Smart Agriculture Decision Support System**
  Integrate weather, soil, and crop datasets into a comprehensive decision support application for precision farming using Python and machine learning

Text Book

T1 -Precision Agriculture Technology for Crop Farming, Qin Zhang, **CRC Press**, **2016**.
T2 - Precision Agriculture Basics, **PrecisionAg Institute**, **2019**.
T3 - Internet of Things and Data Analytics Handbook, edited by Hwaiyu Geng, **Wiley**, **2017**.
T4 - Artificial Intelligence in Agriculture, **Academic Press**, **2022**.

References

R1 - GIS Fundamentals: A First Text on Geographic Information Systems, Paul Bolstad, **Eider Press**, **6th Edition, 2019**.
R2 - Remote Sensing of the Environment: An Earth Resource Perspective, John R. Jensen, **Pearson**, **2nd Edition, 2007**.
R3 - Introduction to Precision Agriculture, edited by Raj Khosla, **ASA, CSSA, and SSSA**, **2021**.
R4 - Artificial Intelligence for Agriculture Innovation, edited by David Riaño, **Burleigh Dodds Science Publishing**, **2022**.

E-Resources/Web Resources

1. https://www.fao.org/digital-agriculture
2. https://swayam.gov.in
3. https://nptel.ac.in
4. https://earthengine.google.com
5. https://docs.qgis.org
6. https://www.kaggle.com/datasets
7. https://www.tensorflow.org/tutorials
8. https://opencv.org
9. https://scikit-learn.org/stable/
10. https://thingspeak.mathworks.com

<table>
<tr><td>Course Code:
CAI3455</td><td colspan="2">Course Name: Digital Agriculture with Blockchain
Type of Course:Theory Integrated Lab</td><td>L- T-P- C</td><td>2-0-2-3</td></tr>
<tr><td>Version No.</td><td colspan="4">X.0</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="4">CSS2264- Essential of AI</td></tr>
<tr><td>Co-requisites</td><td colspan="4">· Basic knowledge of Computer Networks
· Fundamentals of Data Structures / Programming (C/Python)
· Basic understanding of Databases
· Introductory knowledge of IoT or Embedded Systems</td></tr>
<tr><td>Anti-requisites</td><td colspan="4">NIL</td></tr>
<tr><td>Course Description</td><td colspan="4">This course introduces the concepts of Digital Agriculture (Agriculture 4.0) integrated with Blockchain Technology to address challenges in modern farming systems. It covers the fundamentals of blockchain, smart contracts, and decentralized systems, followed by their applications in agricultural supply chains, farm management, and agri-finance. The course also explores the integration of IoT, AI, and blockchain for building transparent, efficient, and secure agricultural ecosystems. Students will gain both theoretical understanding and practical exposure through case studies and lab-based implementations.</td></tr>
<tr><td>Course Objective</td><td colspan="4">This course aims to provide students with a comprehensive understanding of digital agriculture systems integrated with blockchain technology. It focuses on enabling learners to explore modern agricultural practices driven by IoT, AI, and data analytics, while leveraging blockchain for transparency, traceability, and secure transactions. The course equips students with the knowledge and practical skills required to analyze blockchain architectures, develop basic smart contracts, and apply these technologies to real-world agricultural challenges such as supply chain management, land records, and agri-finance systems.</td></tr>
<tr><td>Course Outcomes</td><td colspan="4">On successful completion of the course the students shall be able to:

<table><tr><td>CO1:Explain the fundamentals of digital agriculture and blockchain technology.</td><td>[Level 2 - Understand]</td></tr><tr><td>CO2:Analyze blockchain architecture, consensus mechanisms, and cryptographic principles.</td><td>[Level 4 – Analyze]</td></tr><tr><td>CO3:Develop smart contracts and decentralized applications using blockchain platforms.</td><td>[Level 3 – Apply]</td></tr><tr><td>CO4: Apply blockchain concepts to agricultural applications such as supply chain and land management.</td><td>[Level 3 – Apply]</td></tr><tr><td>CO5: Apply integration of IoT, AI, and blockchain for smart agriculture systems.</td><td>[Level 3 – Apply]</td></tr></table></td></tr>
<tr><td colspan="5">Course Content:</td></tr>
<tr><td>Module 1</td><td>Digital Agriculture and Smart</td><td>Assignment</td><td colspan="2">9 Sessions</td></tr>
</table>

	Farming		
Topics: Introduction to Agriculture 4.0; Challenges in traditional agriculture; Digital agriculture: concepts and benefits; Role of IoT in agriculture (sensors, drones, smart irrigation); Data-driven agriculture and precision farming; Overview of AI applications in agriculture			
Module 2	**Fundamentals of Blockchain Technology**	Assignment	9 Sessions
Topics: Introduction to Blockchain and Distributed Ledger Technology; Structure of blockchain (blocks, hashing, Merkle tree); Cryptographic principles: hashing, digital signatures; Consensus mechanisms: PoW, PoS, PBFT; Bitcoin architecture and transaction lifecycle			
Module 3	**Blockchain Platforms and Smart Contracts**	Assignment	9 Sessions
Topics: Ethereum architecture and working; Smart contracts: concepts and lifecycle; Solidity programming basics; Hyperledger Fabric overview; Decentralized applications (DApps); Security issues in blockchain			
Module 4	**Applications of Blockchain in Agriculture**	Assignment	9 Sessions
Topics: Blockchain in agricultural supply chain management; Farm-to-fork traceability; Smart contracts in agri-finance and crop insurance; Land record management using blockchain; Agri marketplaces and digital transactions; Case studies in India and global context			
Module 5	**Integration of AI, IoT and Blockchain**	Assignment	9 Sessions
Topics: Smart agriculture systems architecture; Integration of IoT and blockchain; AI-based decision-making in agriculture; Digital twin and predictive agriculture; Government initiatives (AgriStack, eNAM); Challenges, scalability, and future trends			
List of Laboratory Tasks: 1. Study blockchain environment setup using tools such as Ganache, MetaMask, and Remix IDE and explore basic transaction workflows. 2. Develop a simple blockchain model using a programming language to understand block structure, hashing, and chain linking. 3. Create and test a basic smart contract using Solidity for storing and retrieving agricultural data. 4. Deploy and interact with smart contracts on a local blockchain network using Remix or similar tools. 5. Perform cryptocurrency transactions and analyze transaction details such as gas fees and confirmations. 6. Design and implement a blockchain-based agricultural supply chain system to track product movement from farmer to consumer. 7. Develop a blockchain-based land registry system to simulate ownership recording and transfer of agricultural land. 8. Implement a decentralized farmer record management system for storing crop, yield, and transaction data.			

9. Simulate integration of IoT sensor data (e.g., soil moisture, temperature) with blockchain for secure data storage.
10. Build a mini project based on a real-world use case such as crop traceability, agri marketplace, or smart irrigation system using blockchain concepts.

Text Book

T1: Rekh Ram Janghel et al., Sustainable Agriculture Production Using Blockchain Technology, Wiley, 2026.

T2: Imran Bashir, Mastering Blockchain, 3rd Edition, Packt Publishing.

References

R1: Arvind Narayanan et al., Bitcoin and Cryptocurrency Technologies, Princeton University Press.

R2: Andreas M. Antonopoulos and Gavin Wood, Mastering Ethereum, O'Reilly Media.

R3: Daniel Drescher, Blockchain Basics, Apress.

R4: Andreas Kamilaris et al., Blockchain in Agriculture and Food Supply Chain, Elsevier.

R5: Ganesh and Kalita, Harvesting Data: Blockchain, AI and Advanced Innovations in Agriculture, Wiley.

E-Resources/Web Resources

1. https://www.sciencedirect.com/browse/journals-and-books
2. https://ieeexplore.ieee.org/Xplore/home.jsp
3. https://www.coursera.org/specializations/blockchain

Course Code: CAI3456	Course Name: Food Traceability in Blockchain Type of Course:Theory Integrated Lab	L- T-P- C	2-0-2-3
Version No.	1.0		
Course Pre-requisites	CSS2264- Essential of AI		
Co-requisites	NIL		
Anti-requisites	CAIXXXX		
Course Description	This course introduces the fundamentals of blockchain technology and its role in enhancing food traceability across modern supply chains. It covers blockchain architecture, smart contracts, digital product identification, and IoT-enabled data capture for secure and transparent food tracking. The course explores blockchain-based traceability systems, food safety, product authenticity, regulatory compliance, and recall management through real-world applications. Students gain practical knowledge of enterprise blockchain platforms and emerging technologies used to improve transparency, trust, and efficiency in farm-to-fork food supply chains.		
Course Objective	This course is designed to improve the learners' EMPLOYABILITY SKILLS through EXPERIENTIAL LEARNING by providing a comprehensive understanding of blockchain technology and its applications in food traceability systems.		

Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Explain the fundamental concepts of blockchain technology, food supply chains, and traceability mechanisms for ensuring transparency and product authenticity. [Level 2 - Understand] **CO2**: Apply blockchain architecture, smart contracts, digital identification technologies, and IoT-enabled data collection techniques to implement food traceability solutions. [Level 3 – Apply] **CO3:** Analyze blockchain-enabled food supply chain architectures by evaluating data integrity, security, transparency, and regulatory compliance requirements. [Level 4 – Analyze] **CO4:** Evaluate blockchain-based traceability platforms and real-world food supply chain case studies to assess their effectiveness in improving food safety and recall management. [Level 5 – Evaluate] **CO5:** Design a secure blockchain-based food traceability framework by integrating blockchain, IoT, and emerging digital technologies to enhance end-to-end supply chain transparency. [Level 6 – Create]		
Course Content:			
Module 1	Foundations of Food Traceability and Blockchain	Assessment: Assignment	6 Sessions
Topics: Introduction to Food Supply Chains and Traceability, Food Safety Challenges and Food Fraud, Farm-to-Fork Traceability Concept, Need for Digital Traceability Systems, Fundamentals of Blockchain Technology, Types of Blockchain (Public, Private, Consortium), Distributed Ledger and Decentralization, Consensus Mechanisms (PoW, PoS, PBFT), Benefits and Limitations of Blockchain in Food Traceability			
Module 2	Blockchain Architecture for Food Traceability	Assessment: Quiz + Assignment	6 Sessions
Topics: Blockchain Network Architecture, Blocks, Transactions and Hash Functions, Nodes and Peer-to-Peer Networks, Smart Contracts Fundamentals, Digital Identity of Food Products, QR Codes, RFID, NFC and Barcode Technologies, IoT Sensors for Food Monitoring, Data Collection and Validation, Blockchain Data Flow from Farm to Consumer			
Module 3	Blockchain-enabled Food Supply Chain Systems	Assessment: Assignment (Programming/Case Study)	6 Sessions
Topics: Farm-to-Fork Blockchain Workflow, Hyperledger Fabric Architecture, Ethereum for Food Traceability, IBM Food Trust Platform, Oracle Integration, Supply Chain Stakeholders, Food Recall Management, Product Authentication and Anti-Counterfeiting, Case Studies (Walmart, IBM Food Trust, Carrefour)			
Module 4	Security, Transparency and Regulatory Compliance	Assessment: Assignment (Case Study Analysis)	6 Sessions
Topics: Data Integrity and Immutability, Privacy and Access Control, Cybersecurity in Blockchain Systems, Traceability Standards (GS1, EPCIS), HACCP and Food Safety Regulations, Regulatory Compliance, Blockchain			

Performance and Scalability, Challenges and Limitations, Sustainability and Ethical Food Supply Chains			
Module 5	Emerging Technologies and Blockchain Applications	Assessment: Mini Project	6 Sessions

Topics:
Artificial Intelligence with Blockchain, IoT-enabled Smart Food Supply Chains, Cloud-based Traceability Systems, IPFS and Distributed Storage, Digital Twins in Agriculture, Food Fraud Detection, Blockchain Analytics and Dashboards, Future Trends in Food Traceability
Mini Project: Design a Blockchain-based Food Traceability System

List of Laboratory Experiments:

Lab 1: Blockchain Development Environment Setup

Objective: Familiarize students with the blockchain development environment and essential software tools.

Tasks:

- Install MetaMask, Ganache, and Remix IDE.
- Configure a local Ethereum blockchain network.
- Create and verify an Ethereum wallet.
- Explore the basic blockchain transaction workflow.

Lab 2: Digital Food Product Identification

Objective: Understand digital identification techniques used in blockchain-enabled food traceability.

Tasks:

- Generate QR codes for food products.
- Encode product information into QR codes.
- Verify product details using QR code scanners.
- Compare QR codes with RFID-based identification systems.

Lab 3: Blockchain Transaction Simulation

Objective: Learn how blockchain records and validates food supply chain transactions.

Tasks:

- Simulate blockchain transactions.
- Create and validate new blocks.
- Observe hash generation and block linking.
- Verify transaction immutability.

Lab 4: Smart Contract Development for Food Registration

Objective: Develop smart contracts for recording food product information on the blockchain.

Tasks:

- Write a simple Solidity smart contract.
- Register food product details.
- Deploy the contract using Remix IDE.
- Test product registration functions.

Lab 5: Blockchain-based Product Traceability

Objective: Implement blockchain-based tracking of food products across the supply chain.

Tasks:

- Record product movement through different stakeholders.
- Update ownership information.
- Verify product history.
- Retrieve traceability records.

Lab 6: QR Code and RFID-based Food Tracking

Objective: Implement product tracking using digital identification technologies.

Tasks:

- Generate QR codes for multiple food products.

- Simulate RFID-based tracking.
- Update blockchain records after each transaction.
- Compare QR and RFID tracking methods.

Lab 7: IoT-enabled Cold Chain Monitoring

Objective: Monitor environmental conditions using IoT devices for food transportation.

Tasks:

- Simulate temperature and humidity sensors.
- Record sensor data.
- Store monitoring information on blockchain.
- Generate alerts for abnormal conditions.

Lab 8: Food Recall and Product Verification

Objective: Develop a blockchain-based food recall mechanism.

Tasks:

- Identify contaminated food batches.
- Trace affected products.
- Generate recall reports.
- Verify product authenticity using blockchain records.

Lab 9: Blockchain Data Visualization

Objective: Visualize food supply chain transactions using blockchain data.

Tasks:

- Import blockchain transaction records.
- Generate product movement reports.
- Display supply chain flow.
- Analyze transaction history.

Lab 10: Comparative Analysis of Blockchain Platforms

Objective: Compare enterprise blockchain platforms used for food traceability.

Tasks:

- Study Hyperledger Fabric architecture.
- Study Ethereum blockchain architecture.
- Compare permissioned and public blockchains.
- Document advantages and limitations.

Lab 11: Blockchain Security and Privacy Analysis

Objective: Analyze security mechanisms used in blockchain-enabled food traceability systems.

Tasks:

- Study cryptographic hashing.
- Analyze digital signatures.
- Explore access control mechanisms.
- Evaluate blockchain security features.

Lab 12: Mini Project – Blockchain-enabled Food Traceability System

Objective: Design and develop a prototype blockchain-based food traceability system.

Tasks:

- Register food products.
- Record supply chain transactions.
- Integrate QR code-based verification.
- Demonstrate complete farm-to-fork traceability.

Targeted Applications & Tools:

Hyperledger Fabric	Enterprise blockchain platform for developing secure and permissioned food traceability systems.

IBM Food Trust	Industry-leading blockchain platform for end-to-end food supply chain transparency and traceability.
Ethereum & Solidity	Development of smart contracts for automating food traceability transactions and product verification.
QR Code / RFID Technologies	Digital identification and tracking of food products throughout the supply chain.
IoT Sensors	Real-time monitoring of environmental conditions (temperature, humidity, location) for blockchain-enabled traceability.

Skill-Building Activities through Experiential Learning:

- **Blockchain-based Food Traceability System Design**
  Students design a farm-to-fork blockchain-enabled food traceability framework by identifying supply chain stakeholders, recording product movement, and mapping transactions to ensure transparency and authenticity.
- **Smart Contract Development for Food Supply Chains**
  Students develop and test basic smart contracts using Ethereum and Solidity to automate food product registration, ownership transfer, and traceability verification.
- **Case Study Analysis of Industrial Food Traceability Platforms**
  Students analyze real-world blockchain implementations such as IBM Food Trust, evaluating their impact on food safety, recall management, supply chain transparency, and consumer trust.

Text Book

T1: Imran Bashir, *Mastering Blockchain*, 4th Edition, Packt Publishing, 2023.

T2: Daniel Drescher, *Blockchain Basics: A Non-Technical Introduction in 25 Steps*, 2nd Edition, Apress, 2020.

T3: Nir Kshetri, *Blockchain and Supply Chain Management*, Elsevier, 2021.

References

R1: Ellahi, R. M., Westerlund, L. C., & Ben Dhaou, A. E., *Blockchain-Based Frameworks for Food Traceability: A Systematic Review*, *Foods*, MDPI, 2023.

R2: Mirabelli, G., Solina, V., et al., *Blockchain and Agricultural Supply Chains Traceability: Research Trends and Future Challenges*, *Procedia Manufacturing*, Elsevier, 2020.

R3: Bhat, S. A., Huang, N. F., et al., *Agriculture Supply Chain Management Based on Blockchain and IoT: A Narrative on Enterprise Blockchain Interoperability*, *Agriculture*, 2021.

E-Resources/Web Resources

- TRUST-FOOD Learning Platform – https://trust-food.ubitech.eu/course/blockchain-and-traceability/
- IBM Food Trust – https://www.ibm.com/products/food-trust
- MDPI Foods: *Blockchain-Based Frameworks for Food Traceability* – https://www.mdpi.com/2304-8158/12/16/3026
- Skills for Africa – Food Traceability Blockchain Systems – https://skillsforafrica.org/in/course/food-traceability-blockchain-systems-enhancing-transparency-and-trust-in-supply-chains-training-course