GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

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 GROUP 50 YEARS OF ACADEMIC WISDOM

PRESIDENCY SCHOOL OF INFORMATION SCIENCE

Program Regulations and Curriculum
2025-2027

MASTER OF COMPUTER APPLICATIONS (MCA)

based on Choice Based Credit System (CBCS) and Outcome Based Education (OBE)

Regulations No.: PU/ AC-26.10/SOIS07/MCA/2025-2027

Resolution No. 10 of the 26th Meeting of the Academic Council held on 25th July 2025, and ratified by the Board of Management in its 27th Meeting held on 28th July 2025.

JULY-2025

Table of Contents

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 Information Science

To be a global centre of excellence in information science and research, fostering innovation and producing professionals with integrity and ethical responsibility.

1.4 Mission of Presidency School of Information Science

- To provide high-quality education in information science, equipping students with strong technical expertise and problem-solving skills.
- To promote research and innovation in information science and technology, addressing real-world challenges through industry collaboration.
- To nurture graduates with strong ethical values and a commitment for lifelong learning for sustained professional growth in the IT sector and allied fields.

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 MCA degree.

The Curriculum is designed to take into the factors listed in the Choice Based Credit System (CBCS) with focus on Social Project Based Learning, Industrial Training, and Internship to enable the students to become eligible and fully equipped for employment in industries, choose higher studies or entrepreneurship.

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 Master of Computer Applications (MCA) Degree Program Regulations and Curriculum 2025-2027

b. These Regulations are subject to, and pursuant to the Academic Regulations.

c. These Regulations shall be applicable to the MCA Degree Programs of the 2025-2027 batch, and to all other MCA Degree Programs which may be introduced in future.

d. These Regulations shall supersede all the earlier MCA Degree Program Regulations and Curriculum, along with all the amendments thereto.

e. These Regulations shall come into force from the Academic Year 2025-2026.

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 MCA 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 MCA Degree Program Regulations and Curriculum, 2025-2027;

ff. “Program” means the Master of Computer Applications (MCA) Degree Program;

gg. "PSIS" means the Presidency School of Information Science;

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 Programme shall be called Master of Computer Applications, abbreviated as MCA. The MCA Degree Program Regulations and Curriculum 2025-2027 are subject to, and, pursuant to the Academic Regulations. These Program Regulations shall be applicable to the ongoing MCA Degree Program of 2025-2027 offered by the Presidency School of Information Science.

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 MCA Degree Program is a Two-Year, Full-Time Semester based program. The minimum duration of the MCA Program is Two (02) years and each year comprises of two academic Semesters (Odd and Even Semesters) and hence the duration of the MCA program is four (04) 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 16.1 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 19.0 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 2-3 years of successful completion of the program, the graduates will:

PEO1: Establish themselves as competent professionals or pursue higher education by applying technical knowledge, analytical thinking, and problem-solving skills in the field of computer applications.

PEO2: Demonstrate ethical responsibility, strong communication skills, and a commitment to lifelong learning by adapting to emerging technologies and evolving industry practices.

PEO3: Lead and contribute to project teams in designing and implementing innovative software solutions that support organizational and societal needs.

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:

PO 1: Foundation Knowledge: Apply knowledge of mathematics, programming logic and coding fundamentals for solution architecture and problem solving.

PO 2: Problem Analysis: Identify, review, formulate and analyse problems for primarily focussing on customer requirements using critical thinking frameworks.

PO 3: Development of Solutions: Design, develop and investigate problems with as an innovative approach for solutions incorporating ESG/SDG goals.

PO 4: Modern Tool Usage: Select, adapt and apply modern computational tools such as development of algorithms with an understanding of the limitations including human biases.

PO 5: Individual and Teamwork: Function and communicate effectively as an individual or a team leader in diverse and multidisciplinary groups. Use methodologies such as agile.

PO 6: Project Management and Finance: Use the principles of project management such as scheduling, work breakdown structure and be conversant with the principles of Finance for profitable project management.

PO 7: Ethics: Commit to professional ethics in managing software projects with financial aspects. Learn to use new technologies for cyber security and insulate customers from malware.

PO 8: Life-long learning: Change management skills and the ability to learn, keep up with contemporary technologies and ways of working.

8.2 Program Specific Outcomes (PSOs):

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

PSO1: Design and develop software solutions for complex real-world problems in specialized domains such as Full-Stack Development, Artificial Intelligence, and

Cybersecurity by applying software engineering principles and advanced programming skills.

PSO2: Identify, analyse and implement use cases across diverse sectors including Healthcare, Agriculture, Banking, and Insurance by utilizing modern tools, effective teamwork, and project management practices.

PSO3: Demonstrate professional ethics, embrace lifelong learning of emerging technologies, and actively contribute to the field of computer applications.

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 MCA Program are listed in the following Sub-Clauses:

9.1 An applicant who has successfully completed BCA/ Bachelor's Degree in Computer Science Engineering or equivalent Degree. OR Passed B.Sc./ B.Com./ B.A. with Mathematics at 10+2 Level or at Graduation Level (with additional bridge Courses as per the norms of the concerned University).

9.2 The applicant must have appeared Karnataka PG-CET or any other State-level Entrance Examinations.

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

9.4 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.

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

9.6 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.

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

10 Transfer Students requirements

10.1 Transfer of student(s) from another recognized University to the 2nd year (3rd Semester) of the MCA 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 MCA, two-Year Degree Program from another recognized University, may be permitted to transfer to the 2nd Year (3rd Semester) of the MCA Program of the University as per the rules and guidelines prescribed in the following Sub-Clauses:

10.1.1 Candidates seeking transfer may be required to complete specified bridge Courses, if any, as prescribed by the University. Such bridge Courses shall not be included in the CGPA computations.

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

10.1.3 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 Presidency University no later than the last date of registration of the concerned year/semester for admission to the 2nd Year (3rd Semester) MCA Program commencing on August 1 on the year concerned.

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

10.1.5 The transfer may be provided on the condition that the Courses and Credits completed by the concerned student in the 1st Year of the MCA 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 MCA Program of the University.

11. Mandatary Bridge Course for Non-Computer Science Discipline Students

Students who have studied Mathematics at the 10+2 level or at the graduation level, but have not undergone Computer Science related courses, are required to undergo the following Bridge Courses before starting of the regular first semester classes to facilitate them with foundational knowledge in computer science.

Sl. No.	Course Code	Course Name	Contact Hours (Modular Basis)
1	CSA9001	C PROGRAMMING AND DATA STRUCTURES	15
2	CSA9002	Fundamentals of Information Technology	15
		TOTAL	**30**

Requirement: Students must successfully complete the Bridge Course and obtain a minimum of 50% marks in the qualifying examination to proceed with their regular academic 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 8.8 of Academic Regulations) shall be clearly defined in the Course Plan for every Course, and 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 Grade (Clause 8.10 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

Table 1: Assessment Components and Weightage for different category of Courses

Nature of Course and Structure	Evaluation Component	Weightage
Lecture-based Course L component in the L-T-P Structure is predominant (more than 1) (Examples: 3-0-0; 3-0-2; 2-1-0; 2-0-2, 2-0-4 etc.)	Continuous Assessments	50%
	End Term Examination	50%
Lab/Practice-based Course P component in the L-T-P Structure is predominant (Examples: 0-0-4; 1-0-4; 1-0-2; etc.)	Continuous Assessments	75%
	End Term Examination	25%
Skill based Courses like Industry Internship, Major project, Research Dissertation, Integrative Studio, Interdisciplinary Project, Summer / Short Internship, Social Engagement / Field Projects, Portfolio, and such similar Non-Teaching Credit Courses, where the pedagogy does not lend itself to a typical L-T-P structure	Guidelines for the assessment components for the various types of Courses, with recommended weightages, shall be specified in the concerned Program Regulations and Curriculum / Course Plans, as applicable.	

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 5.2 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 the sub-clauses 8.9.1 and 8.9.2 of Academic Regulations) 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 ANNEXURE B of Academic Regulations) 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 Clause 13.3 (as per Academic Regulations) and transfer equivalent credits to partially or fully complete the mandatory credit requirements of Discipline 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 Discipline 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 be 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 8.11 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 for the award of the concerned Degree.

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), 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.

PART B: PROGRAM STRUCTURE

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

The MCA Program Structure (2025-2027) totalling 85 credits. Table 3 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: MCA 2025-2027: Summary of Mandatory Courses and Minimum Credit Contribution from various Baskets

Sl. No.	Baskets	Credit Contribution
1	Program Core (PC)	40
2	Elective Course (EC)	15
3	Project Course (Proj)	15
4	Foundation Course (FC)	12
5	Open Elective (OE)	3
	Total Credits	**85**

In the entire Program, the practical and skill-based course component contribute to an extent of approximately 82% out of the total credits of 85 for MCA Computer Applications program of Two-year duration.

15. Minimum Total Credit Requirements of Award of Degree

A minimum of 85 credits is required for the award of an MCA 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 5.0 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 19.2.1 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

Table 3.1 : List of Program Core

Sl. No.	Course Code	Course Name	L	T	P	C
Program Core						
		Theory Course				
1	CSA4201	Data Structures and Algorithms	3	0	0	3
2	CSA4202	Database Systems	3	0	0	3
3	CSA4203	Computer Networks and Security	3	0	0	3
4	CSA4501	Cloud Computing	2	0	0	2
5	CSA4204	Object Oriented Programming using Java	2	0	0	2
6	CSA4205	Adaptive Software Engineering	3	0	0	3
7	CSA4502	Machine Learning	2	0	0	2
8	CSA4503	Data Analytics and Visualization	2	0	0	2
9	CSA4504	MERN Full Stack Development	2	0	0	2
		Practical Course				22
10	CSA4303	Web Technology	1	0	4	3
11	CSA4305	Advanced Python Programming	1	0	4	3
12	CSA4301	Data Structures and Algorithms Lab	0	0	2	1
13	CSA4302	Database Systems Lab	0	0	2	1
14	CSA4601	Cloud Computing Lab	0	0	2	1
15	CSA4304	Object Oriented Programming using Java Lab	0	0	4	2
16	CSA4602	Machine Learning Lab	0	0	2	1
17	CSA4603	Data Analytics and Visualization Lab	0	0	2	1
18	CSA4604	MERN Full Stack Development Lab	0	0	4	2
19	CSA4605	Mobile Application Development using Flutter	1	0	4	3

		Total				18 (40)

Table 3.2: List of Project Courses

Sl. No.	Course Code	Course Name	L	T	P	C
Foundation Courses						
1	CSA8100	Mini Project	0	0	0	3
2	CSA8300	Major Project	0	0	0	12
						15

Table 3.3: List of Foundation Courses

Sl. No.	Course Code	Course Name	L	T	P	C
Foundation Courses						
1	ENG5001	English for Employability	2	1	0	3
2	MAT4001	Probability and Statistics	3	0	0	3
3	PPS4008	Quantitative skills and logical reasoning	1	0	2	2
4	PPS3001	Problem Solving through Aptitude	0	0	2	1
5	PPS3019	Corporate Communications	0	0	2	1
6	MAT4002	Introduction to Operation Research	2	0	0	2
						12

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

Practical / Skill based Courses like internship, project work, major 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, 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 MCA 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). The same shall be prescribed in the Course Handout.

18.1 Mini Project

A student may opt to do a Project Work for a period of 6-8 weeks in an Industry / Company or academic / research institution or the University Department(s) as an equivalence of Internship during the 3rd Semester as applicable, subject to the following conditions:

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

18.1.2 The student may do the project work in an Industry / Company or academic / research institution of her / his choice subject to the above-mentioned condition (Sub-Clause 18.1.1). Provided further, that the Industry / Company or academic / research institution offering such project work confirms to the University that the project work will be conducted in accordance with the Program Regulations and requirements of the University.

18.2 Major Project

A student may undergo a Project for a period of 12-14 weeks in an industry / company or academic / research institution in the 4th Semester as applicable, subject to the following conditions:

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

18.2.2 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 Major Project to a student;

18.2.3 The number of Project available for the concerned Academic Term. Further, the available number of Major 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 Major Project, as stated in Sub-Clause 18.2.2 above.

18.2.4 A student may opt for 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 I Project on her / his own. Provided further, that the Industry / Company or academic / research institution offering such Major Project confirms to the University that the Major Project shall be conducted in accordance with the Program Regulations and Internship Policy of the University.

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

18.3 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 Major Project, subject to the following conditions:

18.3.1 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 18.3.1). 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

Table 3.4: List of Discipline Electives

Sl. No.	Course Code	Course Name	L	T	P	C
Full Stack Stream						
1	CSA4701	Agile Methodology and Devops	3	0	0	3
2	CSA4702	Full Stack Development	1	0	4	3
3	CSA4703	Responsive Web Designing	3	0	0	3
4	CSA4704	Data Modelling with NoSQL Databases	2	0	2	3
5	CSA4705	Backend Development with Node.js	2	0	2	3
Artificial Intelligence Stream						
1	CSA4706	Computer Vision	3	0	0	3
2	CSA4707	Natural Language Processing	1	0	4	3
3	CSA4708	Reinforcement Learning	3	0	0	3
4	CSA4709	Deep Learning	2	0	2	3
5	CSA4710	Generative AI	2	0	2	3
CYBER SECURITY STREAM						
1	CSA4711	Cyber Security and Ethical Hacking	3	0	0	3
2	CSA4712	Web Application Security	1	0	4	3
3	CSA4713	Cybersecurity Testing	3	0	0	3
4	CSA4714	Cloud Security	2	0	2	3
5	CSA4715	AI in Cyber Security	2	0	2	3
FINTECH STREAM						
1	CSA4716	Applied AI in Financial Risk & Credit Modelling	2	0	2	3
2	CSA4717	Smart Contract Engineering & Decentralized Finance (DeFi)	2	0	2	3
3	CSA4718	Advanced FinTech Cybersecurity & Compliance Architecture	2	0	2	3
4	CSA4719	Portfolio Optimization & Robo-Advisory Algorithms	2	0	2	3
5	CSA4720	RegTech Systems, Fraud Analytics & Compliance Automation	2	0	2	3

20. List of Open Electives to be offered by the School / Department:

Table 3.5: List of Open Electives

Sl. No.	Course Code	Course Name	L	T	P	C
		Civil Engineering Basket				
1.	CIV5001	Sustainable Smart Cities	3	0	0	**3**
2.	CIV5002	Systems Design for Sustainability	3	0	0	**3**
3.	CIV5003	Self-Sustainable Buildings	3	0	0	**3**
4.	CIV5004	Energy and Buildings	3	0	0	**3**
		Law Basket				
1.	LAW5001	International Trade Law	3	0	0	**3**
2.	LAW5002	Law relating to Business Establishment	3	0	0	**3**
3.	LAW5003	Data Protection Law	3	0	0	**3**
4.	LAW5004	Law Relating to Consumer Protection	3	0	0	**3**
5.	LAW5005	Law Relating to Infrastructure Projects	3	0	0	**3**
		Electronics and Communication Engineering Basket				
1.	ECE5001	Wearable Computing	3	0	0	**3**
2.	ECE5002	MEMS and Nanotechnology	3	0	0	**3**
3.	ECE5003	Advanced Computer Networks	3	0	0	**3**
4	ECE5004	Pervasive Computing	3	0	0	**3**
		Mechanical Engineering Basket				
1	MEC5001	Optimization Techniques	3	0	0	**3**
2	MEC5002	Industry 4.0	3	0	0	**3**
3	MEC5003	Six Sigma for Engineers	3	0	0	**3**
4	MEC5004	Design for Internet of Things	3	0	0	**3**
5	MEC2003	Supply Chain Management	3	0	0	**3**
		Management Basket				
1.	MBA3026	Essentials of Leadership	3	0	0	**3**
2.	MBA3037	Fundamentals of Accounting	3	0	0	**3**
3.	MBA3038	Sales Techniques	3	0	0	**3**
4.	MBA3039	Principles of Management	3	0	0	**3**

		Media Studies Basket				
1	BAJ5001	Media and Entertainment Business	3	0	0	**3**
2	BAJ5002	TV Journalism and News Management	2	0	2	**3**
		Research Basket				
1.	RES5001	Research Methodology	3	0	0	**3**
2.	URE7001	University Research Experience	-	-	-	3
3.	URE7002	University Research Experience	-	-	-	0

21. List of MOOC (NPTEL) 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 MOOC Courses for MCA Program

Table 3.9: MOOC Discipline Elective Courses
Discipline Elective Courses duration is 4 weeks (01 credit)/ 8 weeks (02 credits)/ 12 weeks (03 credits)

Sl.No	NPTEL Course Code	Course Code	Course Name	Credits
1	NOC25_GE70	CSA4721	Learning Analytics Tools	3
2	NOC25-CS203	CSA4722	Introduction to Industry 4.0 and Industrial Internet of Things	3

Table 3.10: MOOC Open Elective Courses
Discipline Elective Courses duration is 4 weeks (01 credit)/ 8 weeks (02 credits)/ 12 weeks (03 credits)

1	BBB2015	Artificial Intelligence (AI) for Investments	3	0	0	3
2	BBB2016	Business Analytics for Management Decision	3	0	0	3
3	BBA2021	E-Business	3	0	0	3
4	BBA2022	Supply Chain Digitization	3	0	0	3

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

Semester 1:

Sl. No.	Course Code	Course Name	Credit Structure				Contact Hours	Type of Course
			L	T	P	C		
1	ENG5001	English for Employability	2	1	0	3	3	FC
2	MAT4001	Probability and Statistics	3	0	0	3	3	FC
3	CSA4203	Computer Networks and Security	3	0	0	3	3	PC
4	CSA4201	Data Structures and Algorithms	3	0	0	3	3	PC
5	CSA4202	Database Systems	3	0	0	3	3	PC
6	CSA4303	Web Technology	1	0	4	3	5	PC
7	CSA4305	Advanced Python Programming	1	0	4	3	5	PC
8	CSA4301	Data Structures and Algorithms Lab	0	0	2	1	2	PC
9	CSA4302	Database Systems Lab	0	0	2	1	2	PC
		TOTAL	**16**	**1**	12	23	**29**	

Semester 2:

Sl. No.	Course Code	Course Name	Credit Structure				Contact Hours	Type of Course
			L	T	P	C		
1	CSA4501	Cloud Computing	2	0	0	2	2	PC
2	CSA4204	Object Oriented Programming using Java	2	0	0	2	2	PC
3	CSA4205	Adaptive Software Engineering	3	0	0	3	3	PC
4	CSA4502	Machine Learning	2	0	0	2	2	PC
5		Elective 1	3	0	0	3	3	EC
6	PPS4008	Quantitative skills and logical reasoning	1	0	2	2	3	FC
7	PPS3001	Problem Solving through Aptitude	0	0	2	1	2	FC

8		Elective 2	1	0	4	3	5	EC
9	CSA4601	Cloud Computing Lab	0	0	2	1	2	PC
10	CSA4304	Object Oriented Programming using Java Lab	0	0	4	2	4	PC
11	CSA4602	Machine Learning Lab	0	0	2	1	2	PC
		TOTAL	**14**	**0**	**16**	**22**	**30**	

Semester 3:

Sl. No.	**Course Code**	**Course Name**	**Credit Structure**				**Contact Hours**	**Type of Course**
			L	**T**	**P**	**C**		
1	CSA4503	Data Analytics and Visualization	2	0	0	2	2	PC
2	CSA4504	MERN Full Stack Development	2	0	0	2	2	PC
3		Elective 3	3	0	0	3	3	EC
4		Elective 4	2	0	2	3	4	EC
5		Elective 5	2	0	2	3	4	EC
6	MAT4002	Introduction to Operation Research	2	0	0	2	2	
7	PPS3019	Corporate Communications	0	0	2	1	2	FC
8	CSA4603	Data Analytics and Visualization Lab	0	0	2	1	2	PC
9	CSA4604	MERN Full Stack Development	0	0	4	2	4	PC
10	CSA4605	Mobile Application Development using Flutter	1	0	4	3	5	PC
11	CSA8100	Mini Project	0	0	0	3		PC
		TOTAL	**14**	**0**	**16**	**25**	**30**	

Semester 4:

Sl. No.	**Course Code**	**Course Name**	**Credit Structure**				**Contact Hours**	**Type of Course**
			L	**T**	**P**	**C**		
1	xxxxx	Open Elective	3	0	0	3	0	OE
2	CSA8300	Major Project	0	0	0	12	0	PC
		TOTAL	**3**	**0**	**0**	**15**		

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.

Semester 1:

Course Code: ENG5001	**Course Title: English for Employability**		**L- T-P- C**	**2**	**1**	**0**	**3**
Version No.	3.0						
Course Pre-requisites	Graduate Level English Language Proficiency						
Anti-requisites	NIL						
Course Description	The purpose of this course is to enable students enhance their vocabulary, pronunciation and accent and thus ensuring employability. The course is designed in a structured format so as to help students internalize the content. The modules provide adequate scope for internalization through meaningful and relevant activities. Assessments are built at regular intervals to facilitate learning. They also acquire research writing skills which enables them in academic writing.						
Course Outcomes	On successful completion of the course the students shall be able to: 1. Interpret main ideas and supporting details while listening attentively 2. Develop speaking ability in English both in terms of fluency and comprehensibility 3. Discover reading skills, reading speed and read to analyze and interpret information 4. Adapt the knowledge of mechanics of research writing and write a research article.						
Module 1	**Active Listening**	**Listening to audio clips and answering the questions**	**Listening skills and Vocabulary Building**				**10**

1. Listening to Speeches for Vocabulary and Intonation- TED TALKS and Podcasts. 2. Barriers to Effective Listening 3. Types of Listening – Informational, Discriminative, Critical, Empathetic, Appreciative 4. Listening and Note Taking- Activity 1.				
Module 2	**Effective Speaking**	**Presentation**	**Speaking Skills**	**12**
1. Workplace Communication and Communication Etiquette 2. Practical frameworks to improve speaking 3. Attending Interviews 4. Asking and responding to questions, Formal and Informal Communication 5. Expressing views, opinions and preferences 6. Presentation Skills 7. Short speeches				
Module 3	**Reading Strategies**	**Reading Research Articles**	**Reading Skills**	**12**
1. Components of reading 2. Improving thinking skills, analytical abilities, and decision making through Reading 3. Reading Strategies 4. **Reading and Note Making- Activity**				
Module 4	**Scientific Writing/Writing dissertation**	**Writing Reports**	**Writing Skills**	**10**
1. Report Writing- Types of reports, Components of a Report, Structuring a Technical Report. 2. Referencing Skills for Academic Report Writing 3. Writing a Research Article. 4. Writing bibliography				
Texts: 1. Redman, Stuart. English Vocabulary in Use. Cambridge University Press, 1997. 2. MacCarthy, Michael, and Felicity O'Dell. English Vocabulary in Use, Cambridge University Press				

3. Turton, Nigel D. ABC of Common Grammatical Errors. Macmillan India, 1995 https://1filedownload.com/wp-content/uploads/2020/12/Abc-Of-Common- Grammatical-Errors-.pdf	
References: 1. Hart, Steve, Aravind R Nair, and Veena Bhambhani. Embark: English for Undergraduates. Cambridge University Press 2. Hari Prasad, M., John Verghese, R.Kishore Kumar, Komali Prakash, and U. Saraswati Rao. Strengthen Your Steps: A Multimodal Course in Communication Skills. Maruti Publications.	3

Course Code: **MAT4001**	**Course Title: Probability and Statistics** **Type of Course:1] School Core**		**L-T- P- C**	**3**	**0**	**0**	**3**
Version No.		1.0					
Course Pre-requisites		Knowledge of Central Tendency and Measure of Dispersion					
Anti Pre-requisites		**NIL**					
Course Description		The course introduces the concepts of probability theory and statistical analysis, covering how to collect, organize, interpret, and draw inferences from data using mathematical models to understand randomness and uncertainty, with applications across various fields like science, engineering, economics, and social sciences.					
Course Objective		The objective of the course is to equip students with the foundational knowledge of probability theory and statistical methods, enabling them to collect, analyze, and interpret data, and make informed decisions based on the likelihood of events occurring in various situations, often applied across different fields like science, engineering, and business.					
Course Out Comes		On successful completion of the course the students shall be able to: CO1 - be able to compute conditional probabilities directly and using Bayes' theorem, and check for independence of events. CO2 - be able to set up and work with discrete & continuous random variables; in particular, to understand the Bernoulli, binomial, geometric, Poisson distributions, uniform, normal, and exponential distributions. CO3 - Identifying different types of data relationships (linear, polynomial, exponential, logarithmic). CO4 - be able to use specific significance tests, including z-test, t-test (one- and two-sample), and chi-squared test					
Course Content:							

Module 1	**Basic Probability**		**(4 Classes)**
Probability of an Event, multiplication rule, permutations, combinations, Addition Law, Multiplication Law, Conditional Probability, Bayes's Theorem and Problems.			
Module 2	**Random Variables and Bivariate Distributions**	**Assignment**	**(18 Classes)**
Random Variables (discrete and continuous), Probability Mass/Density Functions, Mathematical Expectations, discrete probability distributions - Binomial distribution, Poisson distribution, geometric distribution, Continuous uniform distribution - exponential distribution, normal distribution, gamma distribution. Bivariate distributions and their properties, distribution of sums and quotients, conditional densities, Bayes' rule. Joint Probability distribution for two discrete random variables, expectation and covariance.			
Module 3	**Statistical Methods**		**(11 Classes)**
Descriptive Statistics - Moments, skewness and Kurtosis, Correlation - Karl Pearson's coefficient of correlation and rank correlation (with & Without repetition, Multiple Correlation - Problems. Regression analysis - lines of regression, Multiple regression - Problems. Curve Fitting (Straight Line ($y = a + bx$), Parabola ($y = a + bx + cx^2$), Exponential Curves ($y = ae^{bx}$, $y = ab^x$ and $y = ax^b$)			
Module 4	**Sampling Theory**	**Assignment**	**(12 Classes)**
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 and correlation coefficients, test for ratio of variances - Chi-square test for goodness of fit and independence of attributes.			
Targeted Application & Tools that can be used: The contents of this course has direct applications in most of the core engineering courses for problem formulations, Problem Solution and system Design. Tools Used: R software (Open Source)			
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. Ronald .E. Walpole, Raymond. H. Myers, Sharon. L Myers, and Keying E. Ye, "Probability and Statistics for Engineers and Scientists", Pearson Education, Delhi-9th edition, 2012. 2. B. S. Grewal (2017), Higher Engineering Mathematics by, 44th Edition, Khanna Publishers.			
References: 1. Miller and Freund, Probability and Statistics for Engineers, Pearson Education Ltd. 2. Erwin Kreyzig, Advanced Engineering Mathematics, John Wiley and sons, Inc.10th Edition. 3. Douglas C. Montgomery & George Runger, Applied Statistics and Probability for Engineers, , Wiley Publications **E-resources/ Web links:**			

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_10427

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_100198

https://nptel.ac.in/courses/109104124

https://nptel.ac.in/courses/111106051

https://nptel.ac.in/courses/111102137

https://www.math.hkust.edu.hk/~maqian/ma006_0607F.html

https://www.scu.edu.au/study-at-scu/units/math1005/2022/

Presidency University's Knimbus library URL is: presiuniv.knimbus.com

Topics relevant to SKILL DEVELOPMENT: The course focuses on the concepts of Vector calculus and Linear Algebra 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 the course handout.

Catalogue prepared by		**Dr. Naveen Kumar S H**
Recommended by the Board of Studies on		**14th BOS – Friday, 6th June 2025**
Date of Approval by the Academic Council		

Course Code: CSA4203	**Course Title: Computer Networks and Security** **Type of Course: Program Core**	**L-T- P- C**	3	0	0	3
Version No.	1.0					
Course Pre-requisites	**NIL**					
Anti-requisites	**NIL**					
Course Description	This course provides a comprehensive foundation in computer networking and security, focusing on both theoretical principles and practical applications. It begins with core networking concepts, including network models, signal transmission, and error detection. The course then explores essential communication protocols and techniques used in network routing and data transport. Building on this, students will gain a solid understanding of network security concepts, cryptographic foundations, and modern encryption techniques. Emphasis is also placed on secure communication protocols used in real-world systems. Through quizzes and assignments, students will reinforce their learning and develop skills applicable to both academic and professional contexts.					
Course Objective	**The objective of the course is EMPLOYBILITY of student by using PARTICIPATIVE LEARNING techniques.**					
Course Outcomes	**On successful completion of this course, the students shall be able to:** **CO1**: Apply core networking concepts, including network models (OSI, TCP/IP), signal types, transmission impairments, and error detection/correction techniques. [**Apply**] **CO2:** Apply knowledge of network layer protocols, including IPv4, IPv6, subnetting, routing techniques, and flow control mechanisms in transport protocols. [**Apply**] **CO3**: Analyze the fundamentals of network security, including security principles, attacks, cryptographic techniques, and error handling methods in data communication. [**Analyze**] **CO4:** Analyze the symmetric and asymmetric cryptographic techniques, along with secure communication protocols (SSL/TLS), to ensure secure data transmission. [**Analyze**]					
Course Content:						
Module 1	**Fundamentals of Data Communication and Error Control - CO1**	**Assignment**	**Problem Solving**	**11 Sessions**		

Physical Layer: Introduction - Network Models: OSI Model - TCP/ IP Protocol Suite. Data and Signals: Basics of Analog and Digital Signals - Transmission Impairment. Guided and Unguided Media - Circuit Switched Networks - Datagram Networks. Data Link Layer: Error Detection and Correction: Types of Errors –Parity Check, Two-Dimensional Parity Check, Checksum and CRC, Hamming Distance.				
Module 2	**Concepts of Network Communication and Protocols – CO2**	**Assignment**	**Problem Solving**	**11 Sessions**
Network Layer: IPv4 – Subnetting, Routing - Distance Vector Routing – Link State Routing, IPv6. Transport and Application Layer: Flow control - Sliding Window, Go-Back N ARQ, Selective Repeat ARQ. UDP, TCP, Congestion Control.				
Module 3	**Foundations of Network Security and Cryptographic Techniques – CO3**	**Quiz**	**Theory Heading**	**11 Sessions**
Security Concepts: Introduction, The need for security, Security approaches, Principles of security, Types of Security attacks, Security services, Security Mechanisms, A model for Network Security. Cryptography Concepts and Techniques: Introduction, plain text and cipher text, substitution techniques, transposition techniques, encryption and decryption, symmetric and asymmetric key cryptography, steganography, possible types of attacks.				
Module 4	**Cryptographic Techniques and Secure Communication Protocols – CO4**	**Quiz**	**Theory Heading**	**12 Sessions**
Symmetric key Ciphers: Block Cipher principles, DES, AES, Blowfish, IDEA. Asymmetric key Ciphers: RSA algorithm, Diffie-Hellman Key Exchange. Transport-level Security: Web security considerations, Secure Socket Layer and Transport Layer Security.				
Project work/Assignment: **1. Solve numerical problems on Parity Check, CRC, and Hamming Code error detection.** **2. Explain Distance Vector Routing and Link State Routing with examples.** **3. Manually encrypt and decrypt text using a substitution cipher and a transposition cipher.** **4. Demonstrate RSA encryption and decryption with a small numerical example.** **5. Project: Subnetting, Routing Algorithms, and Transport Protocols** **6. Project: Symmetric, Asymmetric Encryption and Web Security Protocols**				
Topics related to **1. Problem Solving:** Apply network design techniques (like subnetting, routing algorithm selection, and flow control methods) and cryptographic strategies (encryption/decryption) to solve real-world communication, data security, and transmission reliability challenges.				

2. Employability: Hands-on experience with configuring IPv4/IPv6 networks, analyzing TCP/UDP traffic, simulating routing protocols, implementing cryptographic algorithms, and securing data using SSL/TLS protocols to enhance practical skills for careers in networking, cybersecurity, and system administration.

Textbook(s):

T1. Forouzan Behrouz A., “Data Communication and Networking”, 5th Edition, Tata McGraw Hill Publishing Company, New Delhi, 2018.

T2. William Stalllings, “Cryptography and Network Security: Principles and Practice”, 7thEdition, Pearson India Education Services Pvt., Ltd., 2017.

T3. Cryptography and Network Security: Forouzan Mukhopadhyay, Mc Graw Hill, 3rd Edition, 2015.

References

R1. James F. Kurose, Keith W. Ross, “Computer Networking a Top-Down Approach”, 8th Edition, Pearson, 2023.

R2. Computer Networks, Tanenbaum, 5th Edition, Pearson Education Media, 2023.

R3. Behrouz A. Forouzan and Debdeep Mukhopadhyay, “Cryptography and Network Security”, 3rd Edition, Tata McGraw-Hill Education Pvt. Ltd, 2015.

R4. AtulKahate, “Cryptography and Network Security”, 3rdEdition, Tata McGraw Hill Education, 2013.

E-Resources:

1. https://inl.info.ucl.ac.be/cnp3.html
2. https://www.netacad.com/catalogs/learn/networking
3. https://github.com/ssllabs/research/wiki/SSL-and-TLS-Deployment-Best-Practices

Course Code: CSA4201	**Course Title:** **Data Structure and Algorithms** **Type of Course:** **Program Core & Theory**	**L-T-P-C**	**3**	0	0	3
Version No.	1.0					
Course Pre-requisites	Nil					
Anti-requisites	NIL					
Course Description	This course will provide exposure to understand the ADT/libraries, the necessary mathematical abstraction and choose appropriate data structures. It familiarizes students with advanced data structures and paradigms. Course includes theory as well as practical components. Topics to Include: Review of traditional data structures, Dictionaries, Implementation of Dictionaries. Hashing, Skip Lists, Trees, Text Processing and introduction to Computational Geometry					
Course Objective	The objective of the course is to familiarize the learners with the concepts of **Advanced Data Strucuture and Algorithms** and attain **Skill Development** through **Experiential Learning** techniques.					
Course Out Comes	On successful completion of the course the students shall be able to: [1] Review the fundamental concepts and implementation of basic data structures. [Remember] [2] Practice a variety of advanced abstract data type (ADT) and Data structures using various searching and hashing techniques. [Apply] [3] Write algorithms for some of the trees, graphs. [Remember] [4] Apply the basic principles of different string-matching algorithms [Apply]					
Course Content:						
Module 1	**Review of traditional Data Structures**	Quiz	Data Collection	**11 Sessions**		
Based on the background of students, revise programming in C/C++, Stacks, Queues, Lists and Graphs (Dijekstra's algorithm, Spanning tree algorithms).						
Module 2	**Dictionaries and Hash Tables**	Assign ment	Programming Task	**11 Sessions**		

Definition, Dictionary Abstract Data Type, Implementation of Dictionaries. Hashing: Review of Hashing, Hash Function, Collision Resolution Techniques in Hashing, Separate Chaining, Open Addressing, Linear Probing, Quadratic Probing.				
Module 3	**Skip Lists AND Trees**	Quiz	Programming Task	**11 Sessions**
Need for Randomizing Data Structures and Algorithms , Binary Search Trees, AVL Trees, Splay Trees, Heap				
Module 4	**Text Processing**	Assignment	Programming Task	**12 Sessions**
String Operations, Brute-Force Pattern Matching, The Boyer - Moore Algorithm, The Knuth-Morris-Pratt Algorithm, Standard Tries, Compressed Tries, Suffix Tries, The Huffman Coding Algorithm, The Longest Common Subsequence Problem (LCS).				
Project Work / Assignment / Case Study				
1. Hashing 2. Binary Search Tree, AVL Tree 3. String Matching				
Text Books T1. Mark Allen Weiss, "*Data Structures and Algorithm Analysis in C++*", 2nd Edition, Pearson, 2004. http://182.72.188.195/cgi-bin/koha/opac-detail.pl?biblionumber=5882&query_desc=kw%2Cwrdl%3A%20Data%20Structures%20and%20Algorithm%20Analysis%20in%20C%2B%2B T2. M T Goodrich, Roberto Tamassia, "Algorithm Design: foundations, analysis and Internet examples", John Wiley, 2014. http://182.72.188.195/cgi-bin/koha/opac-detail.pl?biblionumber=13008&query_desc=kw%2Cwrdl%3A				
References R1. Thomas Coremen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, "Introduction to Algorithms", Third edition, MIT Press, 2009. https://sd.blackball.lv/library/Introduction_to_Algorithms_Third_Edition_(2009).pdf R2. Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, "Data Structures and Algorithms", Pearson Education, 2013. http://182.72.188.195/cgi-bin/koha/opac-detail.pl?biblionumber=5884&query_desc=kw%2Cwrdl%3A				

R3. Adam Drozdek, “Data Structures and Algorithms in C++”, Fourth Edition, Cengage Learning, 2013. https://itlectures.ro/wp-content/uploads/2016/04/AdamDrozdek__DataStructures_and_Algorithms_in_C__4Ed.pdf

E-Resources

W1. https://sites.cs.ucsb.edu/~suri/cs130a/cs130a

W2. https://www.seas.upenn.edu/~swati/ee22003.html

Course Code: CSA4202	**Course Title: DataBase Systems** **Type of Course: Program Core**	**L-T-P-C**	3	0	0	3
Version No.	**1.0**					
Course Pre-requisites	**Nil**					
Anti-requisites	**NIL**					
Course Description	This course is designed to provide a foundation in data management concepts and database systems. It includes representing information with the relational database model, manipulating data with an interactive query language (SQL) and database programming, database development including database security, integrity and privacy issues. Mastering the core concepts and techniques of Relational and Non Relational database, will help the students to apply their knowledge to a wide range of database solution for a business or organization.					
Course Objective	The objective of the course is to familiarize the learners with the concepts of **Advanced DBMS** and attain **Employability Skills** through **Experiential Learning** technique					
Course Outcomes	**On successful completion of this course the students shall be able to:** 1.Understand the basic concepts of database[Understand] 2.Understand the techniques and tools to design, build and extract information from a database [Understand] 3.Apply MySQL to find solutions to a broad range of queries [Application] 4.Apply the concepts of NoSQL Database. [Application]					

Course Content:				
Module 1	**Introduction to Database, Quiz,**	Quiz	Coding Assignment **12 Sessions**	**12 Sessions**
Topics: Introduction-Database System Applications-Database-System Applications -Purpose of Database Systems-View of Data-Database Languages-Relational Databases-Database Design-Database Model-Data Storage and Querying-Transaction Management-Database Architecture-Normalization				
Module 2	**Relational Databases**	Coding Assignment,	Case Study	**11 Sessions**
Topics: Introduction to the Relational Model-Structure of Relational Databases-Structure of Relational Databases-Database Schema-Keys-Schema Diagrams-Relational Query Languages-Relational Operations-ER Diagrams-Mapping to ER Model to Relational model -Introduction to SQL -Data Definition-Data Types-Basic Structure of SQL Queries-Operations.				
Module 3	**MySQL and NoSQL Database**	Coding Assignment	Project	**11 Sessions**
Topics: MySQL Introduction-MySQL Features- Data types- Variables-MySQL Database Creation-Table-Queries-MySQL Clauses-Introduction to NoSQL : MongoDB CRUD Operation-Insert- Update-Delete-Query-Indexing-Replication-Using MongoDB with Python-Advanced MongoDB Features – Cassandra: Data Model-Table Operations, CRUD Operations.				
Module 4	**Advanced Database Systems**	Quiz	Project	**11 Sessions**
Topics: Object Oriented Databases-Need for Complex Data Types - The Object Oriented Data Model-Object-Oriented Languages-Spatial Databases:Spatial Data Types-Spatial Relationships-Spatial Data Structures–Mobile Databases-Multimedia Databases-Overview of PostgreSQL database				
Targeted Application & Tools that can be used: **Tools: MYSQL, MongoDB, pgAdmin** **Tools/Software Required:** Apache Cassandra (latest stable version like 4.0+) CQLSH for command-line queries				

Mini Project - Applying database skills to a real-world application 1. Create a system that helps streamline tasks, improve efficiency, and provide real-time reports on university operations.

Project work/Test:

Mini Project - During the course, students would need to do coding assignments to learn to train and use different database models. Sample coding assignments include:

1. Healthcare Management System
2. Financial Data Analysis
3. Online Learning Platform
4. Employee Management System etc

Textbook(s):

1. **Sadalage, P. & Fowler, NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, 1st Edition, 2019(Wiley Publications).**
2. **Stefano Ceri, Giuseppe Pelagatti , Distributed Databases: Principles and Systems,, 2017(McGraw Hill Education).**

References

1. **Elmasri R and Navathe S B, “Fundamentals of Database System”,7th Edition, 2017(Pearson Publication).**
2. **Pivert. *NoSQL Data Models: Trends and Challenges*, 1st edition(Wiley).**

Weblink(s):

1. https://www.tpointtech.com/dbms-tutorial
2. https://dev.mysql.com/doc/
3. https://onlinecourses.nptel.ac.in/noc19_cs46/preview

Course Code: **CSA4303**	**Course Title: Web Technology** **Type of Course: Program core** **Theory & Integrated Laboratory**	**L-T- P- C**	1	0	4	3
Version No.	1.0					
Course Pre-requisites	**Basic Programming and Database Concepts**					
Anti-requisites	**NIL**					
Course Description	The purpose of this Course is to introduce the basic concepts and architecture of the World Wide Web. The course contains concepts that enable students to build web pages using various web technologies such as Hypertext Markup Language and Cascading Style Sheets. Students will be trained to plan and design effective web					

	pages by writing codes using current leading trends in the web domain, and enhancing web pages with the use of page layout techniques, text formatting, graphics, images, and multimedia. The focus is on niche technologies that will help students to build Internet- and web-based applications that interact with other applications and with databases.			
Course Objective	The objective of the course is to familiarize the learners with the concepts of **Web Technology** and attain **Skill Development** through **Experiential Learning** techniques.			
Course Outcomes	On successful completion of this course, the students shall be able to: CO1: Make use of Internet concepts and develop static web pages using HTML tags. [Apply] CO2: Apply CSS features and Bootstrap elements to develop a website. [Apply] CO3: Build web-based applications using client-side scripting languages. [Apply] CO4: Develop database driven applications with server-side scripting language using PHP [Apply]			
Course Content:				
Module 1	**Introduction to Internet Standards and HTML**	Quiz	Internet Standards	**15Sessions (L-3, P-12)**
Topics: Basics Of Internet Client/Server Computing: Introduction to WWW, WWW Architecture, Web Browsers, Web servers, SMTP, POP3, MIME, File Transfer Protocol, Overview of HTTP, HTTP request-response, Types of Web servers, Error Response Codes. Markup Language (HTML): Introduction to HTML and HTML5, Basic Structure of HTML Page, Formatting, Commenting, Anchors, Images, Hyperlinks, Lists, Tables, HTML Forms.				
Module 2	UI Design	Assignment	CSS	**18Sessions (L-4, P-14)**
Cascading Style Sheet (CSS): The need for CSS, Introduction to CSS, Basic syntax and structure, Inline Styles, Embedding Style Sheets, Linking External Style Sheets, Levels of CSS, Selectors, Font, color and Text Properties, BOX Model Backgrounds, Manipulating text, Margins, and Padding - Positioning using CSS. Responsive Design, CSS frameworks. **Introduction to Bootstrap**: Containers, Bootstrap elements: Colors, tables, images, buttons, button groups, progress bars, Forms, utilities, Sessions, alerts, custom forms, Grid System.				
Module 3	**Introduction to JavaScript**	Assignment	DOM Model	**20Sessions (L-4, P-16)**

Topics : Introduction to Client Side Scripting, JavaScript Features, Programming Constructs, Arrays and Functions, Document Object Model, Event Handling, Browser functions, Form handling and Validation.

Introduction to JQuery, Syntax, JQuery Fundamentals, Event handling, JQuery Event Model

Module 4	Server-Side Development	Mini Project	Web Application	**22Sessions (L-4, P-18)**

Topics: Introduction to server-side Development with PHP, PHP structure, Data Types, Arrays, $GET and $ POST, Reading/Writing Files, PHP Sessions and Objects, Object Oriented Design, Working with Databases, SQL, Database APIs, Managing MySQL Database. Accessing MySQL in PHP.

List of Laboratory Tasks:

Experiment No. 1: Demonstration of HTML document and formatting tags

Level 1: Design a paragraph about MCA course. Bold, italicize this text and set font color and size also.

Level 2: Design a page with a background image and demonstrate all attributes of a background image. Set the image properties.

Experiment No. 2: Demonstration of HTML List

Level 1: Design an unordered list and an ordered list of different items.

Level 2: Develop a web page with a Menu and a nested menu with ordered and un-ordered lists.

Experiment No. 3: Demonstration of HTML Hyper Link

Level 1: Design a web page with different schools of the Presidency University. When user clicks on the school, the next page should display the details about the school.

Level 2: Design and develop static web pages for an online bookstore and link all pages and show the link on same page also.

Experiment No. 4: Demonstration of HTML table tag

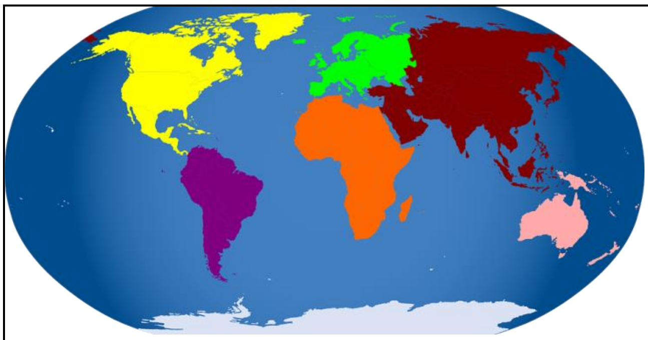
Level 1: Demonstrate the various courses of the university and link those courses using a table and a link tag.

Level 2: Design a Resume page using table and image tags. The page should have all necessary information, photo and signature also.

Experiment No. 5: Demonstration of HTML frame,iframe and IMagemap

Level 1: Design a page using frame and iframe.

Level2 : Demonstrate image mapping for any four countries.



Experiment No. 6: Demonstration of HTML form

Level 1: Design a Login form with a submit and reset button, When the user clicks on the Submit button, it displays the message “Login successful.”, reset button clears the form.

Level 2: The University is organizing a cultural festival, and the organizing team wants to collect registrations for various events with the help of a web page. Design a registration form for collecting the participant details.

Experiment No. 7: Demonstration of different types of CSS

Level 1: Design the webpage by applying the different styles using inline stylesheets

Level 2: Design the webpage by applying the different styles using external & internal style sheets.

Experiment No. 8: Demonstration of CSS image styles

Level 1: Create a web page to change the background color of elements.

Level 2: Create a web page to set the background image, and repeat the image horizontally and fixed background image.

Experiment No. 9: Application of CSS in web designing

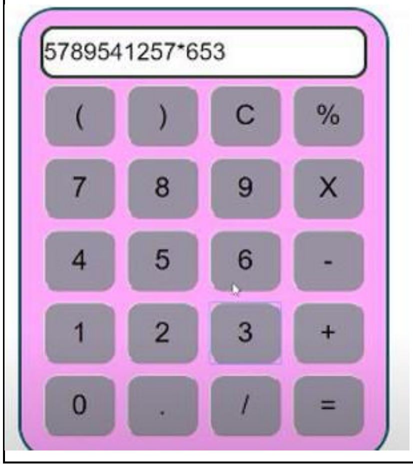
Level 1: Design a document using HTML and CSS to create a catalog of items for online shopping.

Level 2: Create an HTML document for employees’ information in a table and display the same using a cascaded style sheet.

Experiment No. 10: Application of CSS in web designing

Level1 : Design a Web page to show various CSS text properties

Level 2: Create a following calculator interface with HTML and CSS



Experiment No. 11: Application of Bootstrap

Level 1: Design a basic Bootstrap page with a responsive fixed width container.

Level 2: **:** Design a basic Bootstrap page with Bootstrap Grid Structure.

Experiment No. 12: Application of Bootstrap

Level 1: Design a Image gallery page using Bootstrap Image**.**

Level 2: Create a panel with Hello World using Bootstrap**.**

Experiment No. 13: Demonstration of JavaScript

Level 1: Write a Java Script program that on clicking a button, displays scrolling text which moves from left to right with a small delay

Level 2: Write a JavaScript code to change the background color at frequent intervals

Experiment No. 14: Demonstration of JavaScript

Level 1: To write a JavaScript program to define a user defined function for sorting the values in an array. Use HTML5 for the user interface.

Level 2: Develop and demonstrate JavaScript with POP-UP boxes and functions for the following problems:

a) Input: Click on the Display Date button using onclick( )
function Output: Display date in the textbox

b) Input: A number n obtained using prompt
Output: Factorial of n number using alert

c) Input: A number n obtained using prompt and add
another number using confirm
Output: Sum of the entire n numbers using alert

Experiment No. 15: Demonstration of JavaScript

Level 1: Write a JavaScript to change the size of the image when the user clicks on the button.

Level 2: Write a JavaScript program to open a web page after confirming from the user. Otherwise the window should be closed. Use confirm, open, and close methods

Experiment No. 16: Demonstration of JavaScript Validation

Level 1: Write a JavaScript program to give access to some web pages only by presidency University students.

Level 2: Write JavaScript to validate the following fields of the above registration page.

1. Name (Name should contains alphabets and the length should not be less than 6 characters).
2. Password (Password should not be less than 6 characters length).
3. E-mail id (should not contain any invalid characters and must follow the standard pattern (name@domain.com)
4. Phone number (Phone number should contain 10 digits only).

Experiment No. 17: Demonstration of jQuery

Level 1: Develop a page to show Blink text using jQuery**.**

Level 2: Write a code to Disable right click menu in html page using jquery.

Experiment No. 18: Demonstration of jQuery

Level 1: Develop the page to detect when a textbox's content has changed using jQuery.

Level 2: Develop a page to set the background-image using the jQuery CSS property.

Experiment No. 19: Demonstration of jQuery

Level 1: Develop the page to detect when a textbox's content has changed using jQuery.

Level 2: Develop a page to set the background-image using the jQuery CSS property.

Experiment No. 20: Demonstration of jQuery

Level 1: Write a code to access HTML form data using jQuery.

Level 2: Design a page to animate an element, by changing its height and width using jQuery

Experiment No. 21: : **Web design using PHP**

Level 1: Write a PhP program to Get name of the user from a form and show greeting text.

Level 2: Write a PHP Script to find out the Sum of the Individual Digits.

Experiment No. 22: Web design using PHP

Level 1: Write a PHP Program to display current Date, Time and Day.

Level 2: Write a php program to find largest values of two numbers using nesting of function.

Experiment No. 23: **Web design using PHP**

Level 1: Write a php program to show Array manipulation.

Level 2: A web application that takes a name as input and on submit it shows a
hello <name> page where name is taken from the request. It shows the start
time at the right top corner of the page and provides a logout button. On
clicking this button, it should show a logout page with Thank You <name >
message with the duration of usage (hint:Use session to store name and
time).

Experiment No 24: Web design using PHP

Level 1: Write a PHP program to read the personal information of a person such as first name, last name, age, permanent address, and pin code entered by the user into a table created in MySQL. Read the same information from the database and display it on a web page

Level 2: Using PHP develop a web page that accepts book information such as ISBN number, title, authors, edition, and publisher and store information submitted through the web page in MySQL database.

Experiment No 25: Web design using PHP

Level 1: Write a php program to Read from existing file

Level 2: Create a web page to advertise a product of the company using images and audio.

Experiment No 26: PHP Cookies and Filters

Level 1: Write a PHP program to show Cookie concepts

Level 2: Write a PHP program to show different filters.

Experiment No 27: PHP session

Level 1: Write a PHP program to show session concepts

Level 2: Write a PHP program to store page views count in SESSION, to increment the count on each refresh, and to show the count on web page.

Experiment No. 28: Building a website.

Level 1: Build a website for organizing an International Conference. The conference website must be able to collect the author's details and upload a file.

Level 2: Develop the PHP code for partial web pages for ordering vegetables from Bigbasket

Targeted Application & Tools that can be used:

Xampp web server to be used to demonstrate PHP.

Project work/Assignment:

Mini Web application development in a group

Textbook(s):

1. Paul Deitel, Harvey Deitel, Abbey Deitel,"*Internet & World Wide Web How to Program*", Fifth Edition, Pearson Education, 2021.
2. Robin Nixon, " Learning PHP, MySQL & JavaScript: A Step-by-Step Guide to Creating Dynamic Websites, Sixth Edition" Shroff/O'Reilly; Sixth Edition ,(6 August 2021)

References

1. Ivan Bay Ross, "HTML, DHTML, JavaScript, Perl CGI", BPB Publication, 5th Revised Edition, 2022

2.John Pollock , "JavaScript: A Beginner's Guide", TMH, 5th Edition, 2020

3. Ben Frain,"Responsive Web Design with HTML5 and CSS", Packt Publishing,4th Edition 2022

Web references :

1. W3Schools Online Web Tutorials
2. **https://www.tutorialspoint.com/internet_technologies**

Course Code: CSA4305	**Course Title:** Advanced Python Programming **Type of Course: Program Core – Lab Integrated**	**L-T- P- C**	1	0	4	3
Version No.	1.0					
Course Pre-requisites	**Python Basic Programming**					
Anti-requisites	Nil					
Course Description	The advanced Python course covers a wide range of topics and skills to enhance your proficiency in Python programming. Throughout the course, you will delve into advanced concepts such as Object Oriented Programming,Web Scraping,Tkinter,Data analysis and Data visualization. By completing this course, student will have a solid understanding of advanced Python techniques and be well-equipped to tackle complex programming tasks, analyze data, build applications, and work on projects in various domains of AI, machine learning, deep learning and reinforcement learning.					
Course Objectives	The objective of the course is to familiarize the learners with the concepts of Advance Python and attain **Skill Development** through **Experiential Learning** techniques.					
Course Outcomes	**On successful completion of this course, the students shall be able to:** **CO1:**Apply the OOP concepts of python programming for advanced applications. (Apply) **CO2:**Apply advanced python concepts for creating custom modules. (Apply) **CO3:**Apply optimization and parameter tuning techniques for improved Machine Learning algorithms. (Apply) **CO4:** Apply python to GUI based development. (Apply)					
Course Content:						
Module 1	**Object Oriented Programming Concepts**	Assignment	Coding Assignment/Quiz	**21 Sessions (L5 + P16)**		

Overview of Python basics and syntax , Variables, data types ,conditional statements, Lists, Tuples, Sets, Dictionary, Functions, Overview of object-oriented programming (OOP) concepts and principles.				
Module 2	**Advanced Python Concepts**	Assignment	Coding Assignment	**20 Sessions (L4 + P16)**
Regular Expressions and Pattern Matching,Python Modules, Creating and importing own modules, Multithreading, Multiprocessing, Sharing Data between processes, python Testing Frameworks, Lambda functions,Map functions.				
Module 3	**Python Essentials for Data Analysis and Machine Learning**	Experimental Learning	Brainstorming session/Quiz	**20 Sessions (L4 + P16)**
Introduction to Numpy:Vectors,Matrix,Matrix manipulation,Array operations,Slicing,Basing data analysis using Pandas,Pandas data structures,Machine Learning,Regression Analysis,Data Cleaning,Data Visualization using Matplotlib, Seaborn, Plotly,Mysql Connectivity				
Module 4	**GUI Development,Case Study and Project**	Project based Learning	Project	**14 Sessions (L2 + P12)**
Building interfaces with Tkinter or PyQt,GUI Building Libraries,Digital Image Processing,Web Scraping and Data Analysis-Case Study:Python for Data Analysis,Python for Data Visualization and Interactive Dashboards.				
Basics: Use UCI repository and Kaggle dataset for each experiments. Introduction to Python Stack for Data Science, Core Python Libraries for data analysis, Anaconda platform and its installation, Executing programs on Jupiter IDE. **List of Laboratory Tasks:** **Experiment 1** **Exploring Data Structures in Python** **LO1**: Demonstrate creation and manipulation of Lists, Tuples, Sets, and Dictionaries. **LO2**: Perform operations like sorting a list of dictionaries, filtering sets, and aggregating data (e.g., summing values).				

Experiment 2

Object Oriented Programming

LO1: Write and execute Python programs using OOP concepts.

LO2: Run basic Python programs demonstrating Classes and objects, Polymorphism

Experiment 3

Class and Inheritance

LO1: Create a class Employee with attributes name, id, and salary. Add methods to compute bonus and display details.

LO2: Implement multilevel inheritance involving classes Person, Employee, and Manager.

Experiment 4

Regular expressions for pattern matching

LO1: Write a Python program that matches a word at the beginning and ending of a string.

LO2: Validate Email Address with Regular Expressions (RegEx)

Experiment 5

Regex Functions

LO1: Write a Python program that Validate a String Using Regular Expressions

LO2: Extract All Dates in DD-MM-YYYY Format

Experiment 6

Modules and Packages

LO1: Create a module math_utils.py with functions for factorial, prime checking, and GCD.

LO2: Use datetime, os, and sys modules in a script to show file stats and runtime info.

Experiment 7

Building a Custom Python Module

LO1: Develop a Python module with utility functions for mathematical operations (e.g., prime checking, matrix addition).

LO2: Import and use this module in another Colab notebook. Add proper docstrings and comments.

⍰ ⍰

Experiment 8

Multithreading and Multiprocessing

LO1: Implement multithreading to perform tasks like downloading multiple files simultaneously.

LO2: Use multiprocessing to perform heavy computations (e.g., matrix multiplication) on a dataset.

Experiment 9

Functional Programming

LO1:Create a Lambda function that adds 15 to a given number.

LO2:Create a Lambda function that multiplies two arguments and prints the result and use a lambda function inside another function.

Experiment 10

Arrays in Numpy

LO1: Create a NumPy array from a list and print its shape, size, and datatype.

LO2: Perform Array operations like Slice Arrays in Multiple Dimensions,Reshaping and Sorting Arrays.

Experiment 11

Data Analysis on a Real world dataset.

LO1: Load a **Weather Data Analysis Using NumPy.**

Dataset: Weather data (temperature, humidity, rainfall)

LO2: Perform basic statistical and mathematical analysis on temperature and humidity, identify trends over time.

Experiment 12

Advanced Pandas Operations

LO1: Demonstrate data cleaning by handling missing values, duplicates, and outliers in a large dataset.

LO2: Analyze temporal trends in stock price data using Pandas time series methods.

Experiment 13

Data Exploration and Cleaning

LO1: Load and analyze a weather dataset using Pandas

LO2: Explore the ways to detect NaN values in Python, using NumPy and Pandas.

Experiment 14

Regression Analysis in Machine Learning

LO1: Given a data set from UCI repository, implement the simple linear regression algorithm

LO2: Plot the learning curves using Matplotlib and seaborn

Experiment 15

Predictions using Machine Learning

LO1: Train a simple Linear regression model using Scikit-learn

LO2: To predict house prices and visualize the line of best fit.

Experiment 16

Sports performance using Matplotlib

LO1: Create interactive plots to track the performance of athletes over time

LO2: Use maps to visualize geospatial data.

Experiment 17

Data Visualization using Seaborn library

LO1:Load iris dataset using the Seaborn library in Python.

LO2:Create effective visualizations using the Seaborn library in Python.Add titles, color palettes, style, or size arguments.

Experiment 18

Data Visualization with Interactive Dashboards

LO1: Create interactive dashboards using Plotly

LO2: Use maps (via Plotly) to visualize geospatial data.

Experiment 19

MySQL Database Connectivity

LO1: Connect to a MySQL/SQLite database

LO2: Perform CRUD operations and display database records.

Experiment 20

Student Database using MySQL with Python

LO1: Connect MySQL database and create table for Student.

LO2: Perform Student record management system and manage student profiles and grades.

Experiment 21

GUI Development using Tkinter

LO1:Create a GUI to accept and display user input.

LO2: Add a button labeled “Submit” and while clicking the button display the entered name.

Experiment 22

GUI Development using Tkinter

LO1:Create a window with a title, fixed size, and a simple label.

LO2: Accept username and password and display login status (e.g., “Login successful” or “Invalid credentials”).Add a button to close the window.

Experiment 23

GUI Development using Tkinter

LO1: Create a student database using MySQL and perform CRUD operations

LO2: Build a GUI-based calculator using Tkinter

Experiment 24

Web Scraping:BeautifulSoup

LO1: Fetch the HTML content of a webpage and parse it using BeautifulSoup.

LO2:Extract Links and images from a Website

Experiment 25

Web Scraping and Data Analysis

LO1: Scrape data from a live website (e.g., weather data, product prices) using BeautifulSoup or Scrapy. Save the data as a CSV.

LO2: Use Pandas to clean, manipulate, and analyze the scraped data. Visualize findings with Matplotlib and Seaborn.

Experiment 26

Testing and Debugging

LO1: Writing unit tests with unittest or pytest

LO2:Debugging techniques and tools

Experiment 27

End-to-End Data Analysis Project

LO1: Identify a dataset from Kaggle (e.g., Global Warming Data). Scrape, clean, and preprocess the data.

LO2: Visualize insights using an interactive dashboard or multi-chart report. Include predictive analysis using a simple ML model.

Experiment 28

Micro Level Project:Movie Rating Analysis

Targeted Application & Tools that can be used:

Jupyter Notebooks/Google Colab, IDEs, Anaconda / Jupyter Lab, SpeechRecognition (Python Library).

Project work/Assignment: Choose and analyse a network from any organization/Assignment proposed for this course in CO1-CO4

Topics related to

1. Problem Solving: Give any problem to solve using Python.

2. Employability: Doing mini project in Machine Learning using Python.

Text Books

1. Learning Python,Mark Lutz,5th Edition,O'Reilly Media,April 2025.
2. Advanced Python Programming,Quan Nguyen,2nd Edition,Packt Publishing,2021.
3. Think Python: How to Think like a Computer Scientist,Allen B. Downey,2nd Edition, O’Reilly Publishers, 2016.
4. Python Data Analytics with Pandas, NumPy and Matplotlib,Fabio Nelli , Second edition, Apress,2021
5. Core Python Programming, Dr R Nageswara Rao , Second edition ,Dreamtech press, 2018.
6. Test Driven Development with python,Harry J.W Percival,2017, first edition , O'Reilly Media
7. Python Machine Learning Cookbook,by Prateek Joshi,2016, Packt Publishing

Reference Books:

1. Paul Deitel and Harvey Deitel, “Python for Programmers”, Pearson Education, 1st Edition, 2021.
2. G Venkatesh and Madhavan Mukund, “Computational Thinking: A Primer for Programmers and Data Scientists”, 1st Edition, Notion Press, 2021.
3. John V Guttag, "Introduction to Computation and Programming Using Python: With Applications to Computational Modeling and Understanding Data”, Third Edition, MIT Press, 2021
4. Eric Matthes, “Python Crash Course, A Hands - on Project Based Introduction to Programming”, 2nd Edition, No Starch Press, 2019.
5. https://www.python.org/ 6. Martin C. Brown, “Python: The Complete Reference”, 4th Edition, Mc-Graw Hill, 2018.

Web References

1. **https://indico.cern.ch/event/884989/contributions/3732167/attachments/1991200/3331806/Advanced_Programming_with_Python.pdf**
2. **https://nptel.ac.in/courses/**
3. **https://www.udemy.com/course/**
4. **https://www.coursera.org/learn/**

Certificate Course Registration Link:

1.https://www.coursera.org/specializations/python?utm_medium=sem&utm_source=gg&utm_campaign=b2c_india_python_umich_ftcof_specializations_cx_dr_bau_gg_sem_pr_in_all_m_hyb_24-04_x&campaignid=21151281836&adgroupid=164206015567&device=c&keyword=good%20python%20courses&matchtype=p&network=g&devicemodel=&creativeid=695485485980&assetgroupid=&targetid=kwd443443384513&extensionid=&placement=&gad_source=1&gclid=Cj0KCQjwh_i_BhCzARIsANimeoH2zx0_IV2APPaKKmoTWbf4bui3yjB5XIvnTFJw8jFztpgCce7FAhkaAquDEALw_wcB

2.https://www.tpointtech.com/python-oops-concepts

3.https://www.coursera.org/courses?query=python&productDifficultyLevel=Advanced

4.https://www.udemy.com/course/learn-pro-advanced-python-programming/?couponCode=ST14MT150425G3

5.https://www.studocu.com/in/document/anna-university/python-programming/advanced-programming-with-python/29355483

Course Code: CSA4301	**Course Title: Data structures and Algorithms Lab** **Type of Course: Lab**	**L-T- P- C**	0	0	2	1
Version No.	1.0					
Course Pre-requisites						
Anti-requisites	**NIL**					
Course Description	This course will provide exposure to understand the ADT/libraries, the necessary mathematical abstraction and choose appropriate data structures. It familiarizes students with advanced data structures and paradigms. Course includes theory as well as practical components. Topics to Include: Review of traditional data structures, Dictionaries, Implementation of Dictionaries. Hashing, Skip Lists, Trees, Text Processing and introduction to Computational Geometry					
Course Objective	The objective of the course is to familiarize the learners with the concepts of Advanced Data Structure and Algorithms and attain Skill Development through Experiential Learning techniques.					
Course Outcomes	On successful completion of the course the students shall be able to: Co1: Apply Advanced Abstract Data Type (ADT) and using various searching and hashing techniques. [2] Apply various DSA algorithms to implement the trees, graphs [3] Apply advanced data structure concepts of LCS and Red-Black Trees on sequence data					
List of Experiments / Exercises						
List of Laboratory Tasks: Experiment No. 1: Implementing Lists, stacks in C++ Level 1: Understanding the concepts of lists and stacks Level 2: Implementation of a significant subset of STL vector and list classes Experiment No. 2: Implementing Queues and Graphs Level 1: Prioritize the significance of Queue structure						

Level 2: Implementation of a significant subset of STL vector and list classes

Experiment No. 3: Implementations of dictionaries & Hash Tables
Level 1: Building the classic algorithms quadratic probing and separate chaining
Level 2: Analyzing the cuckoo hashing concepts and hopscotch hashing
Experiment No. 4: Implementation Tree algorithms and Heap
Level 1: Importance of general-purpose Tree algorithms
Level 2: Implementation of binary heaps and concepts of pairing heap
Experiment No. 5: Implementation of sorting algorithms
Level 1: Understanding of sorting conditions with comparison-based sorting.
Level 2: Implementation of methods to solve O(N^2) and O(N log N) sorting algorithms
Experiment No. 6: Implementation Minimum spanning Tree
Level 1: Analyze an MST with minimum possible total edge weight
Level 2: Implement MST for Prim's algorithm and Kruskal's algorithm
Experiment No. 7: Implementation of Shortest-Path Algorithms
Level 1: Analyze the representation of Graphs, Acyclic graphs and negative edge costs
Level 2: Implementation of unweighted shortest path and Dijkstra's Algorithm

Experiment No. 8: Implementation of Fibonacci heaps
Level 1: Analyze the concepts of lazy merging for Binomial Queues and Implement the Fibonacci heap operations
Level 2: Implement proof of the Time bound in Fibanacci Heaps

Experiment No. 9: Implementation of Binominal Queues
Level 1: Understanding of Binomial Queue structure and Binomial Queues operations
Level 2- Amortized Analysis of Data structures is carried on queues

Experiment No. 10: Implementation of Huffman coding Algorithm
Level 1- Apply simple scheduling problem (Greedy algorithms)
Level 2- Implement Huffman codes and with approximate Bin packing

Experiment No. 11: Implementation of Knuth-Morris-Pratt (KMP) Algorithm
Level 1- Analyze of worst-case complexity of KMP algorithm o(n+m) and comparision with Naïve algorithm worst case complexity of O(m(n-m+1)).
Level 2- Implement the pattern searching using KMP algorithm

Experiment No. 12: Implementation coding Tries, Suffix Tries
Level 1- Applying tree-like data structure that stores and retrieves strings based on shared prefixes

Level 2- Implement Suffix Tries for searching large sequences like genomes.

Experiment No. 13: Implementation of Longest common Subsequence Problem (LCS)

Level 1- Calculate the common subsequence for the given set of strings (Computational problems)

Level 2- Implement the problems of DNA sequence analysis using LCS algorithm

Experiment No. 14: Implementation of Red-Black Trees

Level 2- Analyze the Advanced Data Structures concepts of Red-Black trees with Bottom-up Insertion, Top-Down Red-Black trees and Top-Down Deletion

References/Manual/Software:

1. Open-source implementation of algorithms in multiple languages (Python, C++, Java)
2. Visualgo – Visualizing DSA concepts such as trees, graphs, sorting etc.,
3. Offline/IDE tools: VS Code, code runner, C++ Intellisense

Course Code: CSA4302	**Course Title:** **DATABASE SYSTEMS LAB** **Type of Course:**	**L-T- P- C**	0	0	2	1
Version No.	1.0					
Course Pre-requisites	**Should be selected only from the subjects studied by the students in the previous semesters.**					
Anti-requisites	**NIL**					
Course Description	This course is designed to introduce the fundamental concepts of database design, implementation, and utilization, covering topics like data models, MySQL, database design principles, and transaction management.It presents the fundamental concepts of database design and use. It provides a study of data models, data description languages, and query facilities including relational algebra and SQL, data normalization, transactions and their properties, physical data organization and indexing, security issues and object databases. It also looks at the new trends in databases.					
Course Objective						
Course Outcomes	**On successful completion of this course, the students shall be able to:** **CO1**: To understand the concepts of Open Source DBMS. Apply					

	CO2: To Understand Data Definition Language. Apply **CO3**: To Apply MySQL to find solutions to a broad range of queries. Apply **CO4: To** Apply the concepts of NoSQL Database. Apply

List of Experiments / Exercises

Lab Experiments:

EXPERIMENT 1:MYSQL INSTALLATION AND DATABASE CREATION

LO1:Installation of MySql and database creation

LO2:Create a database School and table Students with fields: ID, Name, Age, Class, Marks.Insert at least 10 records and display all.

EXPERIMENT 2:FILTER AND SORT DATA

LO1:Creating student tables and using clauses for filter and sort data.Display students with marks > 80 and Sort students by Age in descending order.

LO2:Find students who scored more than the average marks and List courses not enrolled by any student.

EXPERIMENT 3: AGGREGATE FUNCTIONS & GROUP BY

LO1:Create table structures using MySQL data types and apply constraints to MySQL tables .

LO1:Summarize data:Count total students,Calculate average marks and Show total students per class

EXPERIMENT:4 DDL COMMANDS

LO1:Practicing DDL commands in MySQL.

LO2: Insert, Select update and delete queries(DML) from within a MySQL table.

EXPERIMENT 5:MYSQL CLAUSES

LO1:MySQL Querying Using aggregate functions COUNT and SUM.

LO2:Filter records based on conditions using WHERE,DISTINCT, GROUPBY and HAVING Clause

EXPERIMENT 6: DATA MODIFICATION

LO1:To update and delete records.

LO2: Increase marks by 5 for all students in class '8A' and Delete students with marks < 40

EXPERIMENT 7: DATA FORMATTING

LO1: To use functions for formatting dates in MySQL.

LO2: Build formatting dates, string functions using UPPER(), LOWER(), CONCAT(), SUBSTRING(), LENGTH()

EXPERIMENT 8: MONGODB INSTALLATION and CONFIGURATION

LO1: Installation and Configuration of MongoDB in windows.

LO2: Create a database and collection in MongoDB.

EXPERIMENT 9: QUERYING DOCUMENTS IN MONGODB

LO1: Query documents using different criteria and operators using find() and findOne()

LO2:Updating and Deleting Documents

EXPERIMENT 10: PROJECTION AND LIMITING

LO1:Use projection to limit fields and limit() to restrict documents in mongoDB

LO2:Sort documents in ascending or descending order and perform group-based operations using $group.

EXPERIMENT 11: MONGODB OPERATORS

LO1:Practice MongoDB query operators like $in, $and, $or, $exists.

LO2:Create and view an indexes

EXPERIMENT 12: CRUD OPERATIONS IN MONGODB

LO1:To insert and multiple documents into a MongoDB collection.

LO2:Implement CRUD on documents with embedded/nested objects and arrays.

EXPERIMENT 13: CASSANDRA INSTALLATION

LO1:Installing Cassandra on Local Machine (Windows)

LO2:Creating Tables in Cassandra

EXPERIMENT 14: CASSANDRA CRUD OPERTIONS

LO1:Create CRUD operations in Cassandra

LO2:Querying data using SELECT statement

List of Laboratory Tasks:

References/Manual/Software:

1.Abraham Silberschatz, Henry F.Korth,“Database Ssystem Concepts”, McGraw HIll,7th Edition, 2020.

2.Elvis C. Foster, Shripad V. Godbole, “Database Systems”,Apress,2014

3.Daniel Nichter,Efficient MySQL Performance: Best Practices and Techniques,O'Reilly Media,First Edition,January 2022

4. Michael E Kirshteyn,Mastering NoSQL Database Design: A Comprehensive Guide to Building Scalable, High-Performance, and Flexible Data Systems,April 2024.

5.Vinicius M. Grippa,Learning MySQL: Get a Handle on Your Data 2nd Edition,O'Reilly Media,October 2021.

Course Code: **CSA9001**	**Course Title: C Programming And Data Structures** **Type of Course: Bridge cours** **Theory & Integrated Laboratory**	LTPC	1	0	0	0
Version No.	1.0					
Course Pre-requisites	**Basic Knowledge of Computers**					
Anti-requisites	**NIL**					
Course Description	The course offers a comprehensive introduction to C programming and data structures, focusing on foundational and advanced concepts crucial for software development. It begins with C basics—data types, variables, operations, control flow, functions, and arrays—before advancing to complex features like pointers, file handling, and pre-processor directives. Students will explore linear data structures (lists, stacks, queues) and non-linear structures (binary trees, hashing), gaining hands-on experience in implementation and applications. The course also covers key sorting and searching algorithms, emphasizing efficient problem-solving. By course end, students will have solid skills in designing, implementing, and optimizing algorithms and data structures in C.					
Course Objective	This course aims to introduce the fundamentals of C programming, explore its advanced features, understand ADTs and linear data structures, learn non-linear data structures and hashing, and become familiar with sorting and searching techniques.					
Course Outcomes	On successful completion of this course, the students shall be able to: CO1: Understand fundamental programming concepts, including data types, variables, control statements, functions, and array structures, to solve basic computational problems. CO2: Apply advanced features of C programming such as structures, unions, pointers, and file handling to manage and manipulate complex data.					

	CO3: Implement and analyse linear data structures like linked lists, stacks, and queues, along with their applications, to efficiently handle and organize data. CO4: Demonstrate proficiency in non-linear data structures and algorithms, including trees, hashing, and sorting techniques, to enable efficient searching and data retrieval operations.			
Course Content:				
Module 1	**Programming Fundamentals**	Assignment		**4 Sessions**
Topics: Data Types – Variables – Operations – Expressions and Statements – Conditional Statements – Functions – Recursive Functions – Arrays – Single and Multi-Dimensional Arrays.				
Module 2	**C Programming – Advanced Features**	Quiz		**3 Sessions**
Topics: Structures – Union –Pointers: Pointers to Variables, Arrays and Functions – File Handling				
Module 3	**Linear Data Structures**	Assignment		**4 Sessions**
Topics : Linked List – Doubly- Linked Lists – Circular Linked List – Stack ADT – Implementation of Stack – Applications – Queue ADT – Priority Queues – Queue Implementation – Applications.				
Module 4	**Non-Linear Data Structures And Searching and Sorting Techniques**	Assignment		**4 Sessions**
Topics: Trees – Binary Trees – Tree Traversals – Binary Search Tree – Hashing - Hash Functions - Bubble Sort – Linear Search – Binary Search				
List of Laboratory Tasks: Experiment 1: Write a program to read two numbers and find the largest of the two. Experiment 2: Write a program to check if a given number is even or odd. Experiment 3: Write a program to check whether a given number is negative, positive, or zero. Experiment 4: Write a program to read three numbers and find the largest among them.				

Experiment 5: Write a program to read a number between 1 and 7 and display the corresponding day of the week (1 for Monday, 2 for Tuesday, etc.).

Experiment 6: Write a program to read a number and find the sum of its digits.

Experiment 7: Write a program to calculate the factorial of a number using a non-recursive function.

Experiment 8: Write a program to calculate the factorial of a number using a recursive function.

Experiment 9: Write a program to read an array of integers and print the elements in reverse order.

Experiment 10: Write a program to read a matrix of size mXn and display the sum of principal diagonal elements.

Experiment 11: Write a program to swap two variables using pointers.

Experiment 12: Implement a stack using an array with basic push and pop operations.

Experiment 13: Implement a queue using an array with enqueue and dequeue operations.

Experiment 14: Write a program to perform linear search on an array of integers, where the user provides the array and the target element.

Experiment 15: Write a program to perform bubble sort on an array of integers and display the sorted array.

Targeted Application & Tools that can be used:

System software and Application software

Programming Professionally Used Software : MinGW / C/C++ IDE

Textbook(s):

1) Mark Allen Weiss, “Data Structures and Algorithm Analysis in C”, Second Edition, Pearson Education, 1997.
2) Reema Thareja, “Programming in C”, Second Edition, Oxford University Press, 2016.

References

1. Brian W. Kernighan, Rob Pike, “The Practice of Programming”, Pearson Education, 1999.
2. Paul J. Deitel, Harvey Deitel, “C How to Program”, Seventh Edition, Pearson Education, 2013.
3. Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, “Data Structures and Algorithms”,Pearson Education,1983.
4. Ellis Horowitz, SartajSahni and Susan Anderson, “Fundamentals of Data Structures”,Galgotia, 2008.

Web references :

1. https://www.coursera.org/specializations/data-structures-algorithms

2. https://nptel.ac.in/courses/112107243 3. https://nptel.ac.in/courses/112105598
Topics relevant to development of **“Foundation Skills”:** Fundamentals of Data structure**,** **“Skill Development”:** Implementation Linear and non-linear data structure**,** **“Employability”:** Linear & Non-linear Data Structure

Course Code: **CSA9002**	**Course Title: Fundamentals of Information Technology** **Type of Course: Bridge Course** **Theory**	**L- T- P- C**	1	0	0	0
Version No.	1.0					
Course Pre-requisites	**Nil**					
Anti-requisites	**NIL**					
Course Description	This course provides an introduction to the foundational concepts of information technology. Key topics include computer arithmetic and number systems, primary and secondary storage systems, and data storage/retrieval methods. The course explores object-oriented programming concepts such as Sessions, objects. It also covers relational database management systems, SQL, database design. Additionally, students will learn about operating systems, and various processing methods. Networking fundamentals, data communication, and key networking devices and protocols are also discussed. This course aims to build a comprehensive understanding of IT principles and their practical applications in modern computing.					
Course Objective	The objective of this course is to provide students with a fundamental understanding of key information technology concepts, including computer arithmetic, number systems, and data storage methods. It aims to introduce essential principles of object-oriented programming, relational databases, and software engineering, while also covering the basics of operating systems, system performance, and data communication. Through these topics, students will develop a solid foundation in IT fundamentals and gain practical insights into networking and database management systems.					

Course Outcomes	On successful completion of this course, the students shall be able to: CO1: Define fundamental concepts of computer arithmetic, number systems, data storage methods, and basic networking principles. [Remember] CO2: Apply object-oriented programming concepts, relational database management principles, and SQL to solve real-world problems. [Apply] CO3: Develop an understanding of operating systems, system performance, and data communication protocols. [Apply] CO4: Implement and manage basic networking devices, communication modes, and protocols for efficient data transmission. [Apply]			
Course Content:				
Module 1	**Computer arithmetic & storage fundamentals**	Quiz		**4 Sessions**
Topics : Binary, Binary Arithmetic, Number System: Positional & Non Positional, Binary. Primary Vs Secondary Storage, Data storage & retrieval methods. Primary Storage:. Secondary Storage				
Module 2	**Object oriented concepts**	Assignment		**4 Sessions**
Topics : Concepts of object oriented programming, Managing software complexity abstraction, class, object, member data, member methods, encapsulation, data hiding, inheritance, polymorphism, binding.				
Module 3	**Relational Database management**	Assignment		**4 Sessions**
Topics : Basic RDBMS concepts, database design, SQL comments, embedded SQL concepts, OLTP concepts.				
Module 4	**Operating system and Data communication**	Mini Project		**3 Sessions**
Topics: Functions, features, Single Processor & Multiprocessor systems, Real-Time System, Data, Communication, Basic Networking Devices, Communication Process, Types of Networks, LAN Topologies, Concepts relating to networking.				
Targeted Application & Tools that can be used: **Vs code can be used to demonstrate Html/css**				
Project work/Assignment:				

Mini Web application development in a group

Textbook(s):

1) Andrew. S. Tanenbanum, “Structured Computer Organization”, Fourth Edition, PHI, 2018.
2) Abraham Silberschatz, Henry F. Korth, S. Sudharshan, “Database System Concepts”, Fourth Edition, Tata McGraw, 2014.
3) Roger S. Pressman, “Software Engineering – A practitioner’s approach”, Sixth Edition, McGraw Hill Publishers, 2004.

References

1) Andrew S. Tanenbaum, "Structured Computer Organization", PHI, 4th Edition, 2018.
2) Abraham Silberschatz, Henry F. Korth, S. Sudarshan, "Database System Concepts", Tata McGraw-Hill, 4th Edition, 2014.
3) Roger S. Pressman, "Software Engineering: A Practitioner’s Approach", McGraw-Hill, 6th Edition, 2004.
4) Abraham Silberschatz, Peter B. Galvin, Greg Gagne, "Operating System Concepts", Wiley, 10th Edition, 2018.
5) Jon Duckett, "HTML and CSS: Design and Build Websites", Wiley, 1st Edition, 2011.

Web references :

1. W3Schools Online Web Tutorials
2. **https://www.tutorialspoint.com/internet_technologies**

Topics relevant to “SKILL DEVELOPMENT”:

Operating system concepts and Networking for skill development through Experiential learning techniques. This is attained through the assessment component mentioned in the course handout.

Semester 2

Course Code: **CSA4501**	**Course Title: Cloud Computing** **Type of Course: Program Core – Lab Integrated**	**L-T-P-C**	2	0	2	3
Version No.	1.0					
Course Pre-requisites	Computer Networks					
Anti-requisites	NIL					

Course Description	This course provides a hands-on comprehensive study of Cloud concepts and capabilities across the various Cloud service models including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). It dives into all of the details that a student needs to know in order to plan for developing applications on the cloud and what to look for when using applications or services hosted on a cloud.			
Course Objective	The objective of the course is to familiarize the learners with the concepts of Cloud Computing and attain Skill Development through Experiential Learning techniques.			
Course Outcomes	Upon successful completion of the course the students shall be able to: 1) Understand the significance of Cloud computing technologies 2) Identify appropriate Virtualization techniques to virtualize infrastructures 3) Discuss Cloud mechanisms to optimize the QoS parameters 4) Develop applications using Cloud services and VM instances			
Course Content:				
Module 1	**Introduction to Cloud services**	Assignment	Theory	**No. of Sessions:7**
Topics: Evolution of cloud computing, Computing Platforms and Technologies, Cloud Computing Architecture, IaaS, PaaS, SaaS, Types of Clouds, Cloud Computing Environments.				
Module 2	**Virtualization Techniques**	Assignment	Theory	**No. of Sessions:7**
Topics: Basics of Virtualization - Types of Virtualizations, Taxonomy of Virtualization Techniques, Implementation Levels of Virtualization.				
Module 3	**Cloud QoS and Management**	Assignment	Theory	**No. of Sessions:8**
Topics: Cloud Infrastructure Mechanisms, SLAs, Specialized Cloud Mechanisms, Cloud Management Mechanisms, Cloud Security Mechanisms				
Module 4	**Application development in Cloud**	Assignment	Case Study	**No. of Sessions:8**
Topics: Programming Models for Cloud Computing - Software Development in Cloud - Service creation environments to develop cloud-based applications. Development				

environments for service development (Demonstration using AWS Cloud); Dockers and Containers.

Targeted Application & Tools that can be used:

Targeted Applications:

Developing applications on Cloud Platforms via Virtual machines

Cloud Tools:

- CloudSim
- VMWare
- Amazon EC2
- Google Compute Engine
- Microsoft Azure

Project work/Assignment:

1. Automation of performance analysis of students through the Cloud
2. Chatbots development using Cloud resources
3. Blog creation using Cloud computing

Analysis of Case Studies: When deciding to adopt cloud computing architecture, decide if the cloud is right for your requirements (for the application identified).

Text Book(s)

1. Daniel Vaughan, "Cloud Native Development with Google Cloud". O'Reilly Media Publishers. 1st Edition 2023.
2. Rajkumar Buyya, Christian Vecchiola, and Thamarai Selvi, "*Mastering Cloud Computing*", McGraw Hill Education, 2017edition.
3. John Rittinghouse and James Ransome, "*Cloud Computing, Implementation, Management and Security*", CRC Press, 2010 edition**.**

References

1. Thomas Erl, Zaigham Mahmood, and Ricardo Puttini, "*Cloud Computing Concepts, Technology & Architecture*", PHI publisher 2013 edition.

2. Anthony T Velte, Toby J Velte, Robert Elsenpeter, "*Cloud Computing: A Practical Approach*", Tata McGraw-Hill, 2010 edition.

3. David E.Y. Sarna, "*Implementing and Developing Cloud Applications*", CRC Press, 2018 edition.

4. Manvi, Sunilkumar, and Gopal K. Shyam. *“Cloud Computing: Concepts and Technologies”*. CRC Press, 2021.

Web Resources and Research Articles links:

1. IEEE Transactions on Cloud Computing- https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=6245519
2. International Journal of Cloud Computing- https://www.inderscience.com/jhome.php?jcode=ijcc
3. CloudSim Resources- https://javadoc.io/doc/org.cloudsimplus/cloudsim-plus/latest/org/cloudbus/cloudsim/resources/class-use/Resource.html
4. Journal of Network and Computer Networking- https://www.journals.elsevier.com/journal-of-network-and-computer-applications

Topics relevant to “SKILL DEVELOPMENT”: Taxonomy of Virtualization Techniques,**Specialized** Cloud Mechanisms, Cloud Management Mechanisms, Cloud Security Mechanisms for **Skill Development** through **Experiential Learning Techniques.** This is attained through assessment component mentioned in course handout.

Course Code: CSA4204	**Course Title: Object Oriented Programming using Java** **Type of Course: Program Core**	**L-T-P-C**	**2**	**0**	**0**	**2**
Version No.	1.0					
Course Pre-requisites	**NIL**					
Anti-requisites	**NIL**					
Course Description	The main objective is to learn the basic concept and techniques, which form the object-oriented programming paradigm. Object-oriented programming is a new way of thinking about problem using models organized around real world concept. It investigates the software engineering principles of encapsulation, information hiding and code reuse, and discusses how these concepts are used to build abstract data types. The object oriented programming features of Sessions, inheritance, polymorphism and composition are studied, along with constructors and method overloading. Students implement Java programs incorporating features from the Java programming language.					
Course Objectives	The objective of the course is to familiarize learners with Object-Oriented Programming concepts using Java, while developing general-purpose applications with database connectivity through experiential learning and object-oriented design principles.					
Course Out Comes	On successful completion of this course the students shall be able to: 1. Apply Object-Oriented Programming principles in Java to design modula and reusable code that efficiently solve the real-world problems. (Apply) 2. Utilize the concepts of Inheritance, Multithreading, and Exception Handling to develop efficient and robust code. (Apply) 3. Develop Serverside java applications using Servlet and JSP concepts. (Apply) 4. Construct basic applications that demonstrate efficient interaction with relational database systems through JDBC and Hibernate frameworks. (Apply)					

<table>
<tr><td><b>Module 1</b></td><td>Introduction to OOPs</td><td>Assignment</td><td></td><td>Programming activity</td><td><b>6 Sessions</b></td></tr>
<tr><td colspan="6">Topics:
The Java Buzzwords – Overview of Java – Data Types, Variables and Arrays – Operators – Control Statements – Introducing Sessions – Methods and Sessions: Overloading Methods – Passing and returning Objects – Recursion – Access control – static – final – Nested and Inner classes
String: Creation & Operation. String builder and String Buffer class, methods in String Buffer</td></tr>
<tr><td><b>Module 2</b></td><td>Inheritance, Exception Handling, and Multithreading</td><td>Assignment</td><td></td><td>Problem Solving</td><td><b>10 Sessions</b></td></tr>
<tr><td colspan="6">Topics:

Inheritance: Basics – Using super – Method Overriding – Dynamic Method dispatch – Abstract Class – final with Inheritance.
Packages and Interfaces : Packages – Packages and Member Access – importing Packages – Interfaces – Default Interface Methods – static Methods in Interface - Private Interface method
Fundamentals – Types – Uncaught Exceptions – try and catch – Multiple catch – Nested try – throw – throws – finally – Built-in Exceptions
Multithreading : Java Thread Model – Main Thread – Creating a Thread and Multiple threads – Thread methods, Synchronization</td></tr>
<tr><td><b>Module 3</b></td><td>Collections, Servlets and Java server Pages</td><td>Quiz</td><td></td><td>Programming activity</td><td><b>8 Sessions</b></td></tr>
<tr><td colspan="6">Topics:
Collection Frameworks: Collection Interfaces - Collection Sessions: ArrayList - LinkedList - HashSet - TreeSet - Priority Queue - Iterator - Map: Map Interfaces - Map Sessions: HashMap – TreeMap – Comparators, Lambda Expressions
Working with Servlets: Features – Servlet API – Servlet Life Cycle – Creating a Sample Servlet, Session and Cookies
Java Server Pages: Architecture of JSP pages – Life Cycle of JSP – Working with JSP Basic Tags and implicit objects – Exploring Action Tags.</td></tr>
</table>

Module 4	IO operation,JDBC, Hibernate	Assignment		Programming activity	**6 Sessions**
Topics: 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 Working with JDBC: Introduction - JDBC Drivers – Features of JDBC – JDBC API – Major Sessions and Interfaces – Process with java.sql package Working with Hibernate: Architecture – Exploring HQL – Hibernate O/R mapping – Working with Hibernate.					
Targeted Application & Tools that can be used: Notepad++, Eclipse IDE, NetBeans IDE					
Project work/Assignment: Mention the Type of Project /Assignment proposed for this course **Certification and Mini Project**					
Text Book 1. Herbert Schildt, ”Java: The Complete Reference”, 12th Edition, McGraw Hill, 2021.					
References 1. E. Balagurusamy, “Programming with Java”, Tata McGraw Hill Education, 6th Edition, 2019. 2. Cay Horstmann,” Core Java -Volume 1: Fundamentals”, 12th Edition, Oracle Press, 2021 **Web References** W1. NPTEL Course on “Java Programming”, Prof. Debasis Samanta, https://archive.nptel.ac.in/courses/106/105/106105191/ W2. https://docs.oracle.com/javase/tutorial/ W3. https://dev.java/learn/					
Topics relevant to “SKILL DEVELOPMENT”: Introduction to object-oriented programming, Initializing & Accessing Array, extending interfaces, implementing interfaces - Organizing Sessions and Interfaces in Packages, life cycle of a thread, Collection Types, Sets**for Skill Development through Experiential Learning techniques. This is attained through assessment component mentioned in course handout.**					

Course Code: CSA4205	Course Title : Adaptive Software Engineering Type of Course: Program Core	L- T-P- C	3	0	0	3
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-requisites	NIL					
Course Description	This course is intended to provide the students with an overall view over the Software Engineering Discipline and with insight into the processes of software development. This course also provides students with a theoretical as well as practical understanding of agile software development practices, how small teams can apply them to create high-quality software and insights of different software architectures.					
Course Objectives	The objective of the course is EMPLOYBILITY of students by using PARTICIPATIVE LEARNING techniques.					
Course Out Comes	On successful completion of this course the students shall be able to: [CO1] Illustrate software engineering principles and activities involved in building large software programs. [Understand] [C02] Articulate the process of requirement gathering, requirement classification and requirement documentation. [Apply] [C03] Illustrate the importance of software architecture. [Apply] [C04] Classify the software quality parameters and software quality standards. [Apply]					
Module 1	Software and Software Engineering	Assignment	Theory		10 Sessions	
Introduction, Nature of software, Defining the discipline, Software Process, Software Application Domains, Software Myths, Terminologies, Role of management in software development Software Process Models: Generic Models, Defining Framework Activity, Process Assessment and Improvement, Prescriptive Process Models, Introduction to Agility and Process, Agility Principles and Agile Development , Scrum and Other Frameworks						
Module 2	System Models, Software Prototyping and Specification	Assignment	Case Studies		10 Sessions	

<table>
<tr><td colspan="5">Requirements Definition, preliminary Architectural Design, Resource Estimation, Prototype construction and evolution, Understanding Requirements, Requirements Engineering Types of Requirements, Feasibility Studies, Requirements Elicitation, Developing Use Cases, Requirements - Analysis Documentation, Software Requirement and Specification (SRS)</td></tr>
<tr><td><b>Module 3</b></td><td><b>Architectural Design</b></td><td>Assignment</td><td>Theory</td><td><b>12 Sessions</b></td></tr>
<tr><td colspan="5">Software Architecture, Importance of Software Architecture, Its Role, Views, Component & Connector View and its architecture style, Architectural Considerations, Assessing Alternative Architectural Designs, Architectural Reviews</td></tr>
<tr><td><b>Module 4</b></td><td><b>Software Quality and Security</b></td><td>Assignment</td><td>Theory</td><td><b>13 Sessions</b></td></tr>
<tr><td colspan="5">Software Quality, Quality Factors, the Software Quality Dilemma, Risk, Achieving Software Quality. Quality control and assurance, Elements of Software Quality Assurance, SQA Tasks, Goals, and Metrics, Formal Approaches to SQA</td></tr>
<tr><td colspan="5"><b>Targeted Applications & Tools that can be used: JIRA, Confluence</b></td></tr>
<tr><td colspan="5"><b>Project work/Assignment: Mention the Type of Project /Assignment proposed for this course</b></td></tr>
<tr><td colspan="5">Case Study : Health Care support System: Develop a SRS for the mentioned system</td></tr>
<tr><td colspan="5"><b>Text Book</b>
1. Roger S. Pressman, Bruce R. Maxim, “Software Engineering: A Practitioner's Approach”, 9th Edition, Tata McGraw Hill. 2023.</td></tr>
<tr><td colspan="5"><b>References</b>
1] K.K.Aggarwal, Yogesh Singh, ”Software engineering”, New Age International Publisher, Fourth edition,2022
2] James Shore, Shane Warden,”The Art of Agile Development”, Second Edition, O'Reilly,2021
3] Ari Takanen, Jared de Mott, Charlie Miller,"Fuzzing for Software Security Testing and Quality Assurance", Artech House Publishers , Second Edititon,2023
<b>Web resources:</b>
<b>1.https://onlinecourses.nptel.ac.in/noc20_cs68/preview</b> ●
<b>2.https://onlinecourses.nptel.ac.in/noc24_mg01/preview</b></td></tr>
</table>

Course Code: CSA4502	**Course Title: Machine Learning**	**L-T-P-C**	**2**	**0**	**0**	**2**
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-requisites	NIL					
Course Description	This course builds the foundational insight of understanding principles of machine learning process and to explore various ML algorithms and practices to accelerate strategic decision making with data. Overall, this course gives a comprehensive insight to appropriately choose ML algorithms on real time problems to construct an intelligent ML model.					
Course Objective	The objective of the course is to familiarize the learners to explore the competence and comprehend with potential machine learning algorithms and techniques to revolutionize with real-world problems and create prominent solutions (with ML models) to attain **Employability Skills** through **Experiential Learning** techniques.					
Course Out Comes	**On successful completion of the course the students shall be able to:** Co1: Infer the basic knowledge of Machine Learning concepts and apply ML techniques using Tree and Bayesian models Co2: Build Neural network with SVM algorithm, Evolutionary Learning Ensemble methods for solving classification problems Co3: Practice unsupervised models, Explanation-Based reinforcement learning methods using Python language Co4: Solve real-time applications using Machine Learning Algorithms					
Course Content:						
Module 1	Introduction to Machine Learning	Participative Learning		Brainstorming session/Quiz	**Sessions -6**	
Topics: Introduction to Machine Learning: Types of Machine Learning, Supervised Learning: The Machine Learning Process, Performance measures, The Bias-Variance Tradeoff, Learning with Trees: Using Decision Trees, Constructing Decision Trees, Classification and Regression Trees (CART), Turning Data into Probabilities: The Naïve Bayes' Classifier, Bayesian Networks.						
Module 2	Supervised Learning	Participative Learning		Fish bowl, Think-pair & share	**Sessions -8**	
Topics: Neural Networks: The Brain and The Neuron, Neural Networks, The Perceptron, Linear Separability, The Multi-Layer Perceptron: Going Forwards, Going Backwards Back-Propagation of Error, The Multi-Layer Perceptron in Practice, Deriving Back-Propagation. Support Vector Machines:						

Optimal Separation, Kernels, The Support Vector Machine Algorithm, Dimensionality Reduction: The Curse of Dimensionality, Linear Discriminant Analysis (LDA), Principal Components Analysis (PCA), Evolutionary Learning: The Genetic Algorithm (GA), Generating Offspring: Genetic Operators, Using Genetic Algorithms. Ensemble Learning: Boosting, Bagging and Random Forests.					
Module 3	Unsupervised Learning	Experiential Learning		Implementation of unsupervised learning algorithms	**Sessions -8**
Topics: Unsupervised Learning: The k-means algorithm, Hierarchical Clustering, The Self-Organising Feature Map, Explanation based Learning, Markov Decision Process- Reinforcement Learning and Evaluating Hypotheses: Introduction, Learning Task, Q Learning, Non Deterministic Rewards and Actions.					
Module 4	Symmetric weights and Deep Belief Networks	Project-based Learning		Realtime-problem solving using ML algorithms	**Sessions -8**
Topics: The EM Algorithm: Estimate Means of K Gaussians, General Statement of EM Algorithm, Extensions to the SVM, Active Reinforcement Learning, Energetic Learning: The Hopfield Network, Stochastic Neurons: The Boltzmann Machine, Deep Belief Networks (DBF)					
Text Books: **T1:** Machine Learning: An Algorithmic Perspective, Stephen Marshland, 2nd Edition, CRC Press, Taylor & Francis group, ISBN: 978-1-4665-8333-7, November 2014. **T2**: Machine Learning in Action, Peter Harrington, ISBN: 978-935-004-4131, April 2012. **T3:** Introduction to Machine Learning, Ethem AIpaydin, 4th Edition, The MIT press, ISBN:978-0-262-043-793, March 2020.					
Reference Books: **R1.** Hands-On Machine learning with Scikit-Learn, Keras, and Tensorflow: Concepts, Tools, and Techniques to Build Intelligent Systems, Aurelien Geron, 3rd Edition, O'Reilly Media, ISBN: 978-9355421982, October 2022. **R2**. Introduction to Machine Learning with Python: A Guide for Data Scientists, Andreas C. Muller,1st Edition, O'Reilly Media, ISBN: 978-1449369873, September 2016. **R3**. Machine Learning: A Probabilistic Perspective, Kevin P. Murphy, The MIT Press, ISBN: 978-0262018029, August 2012. **R4.** An Introduction to Machine Learning, Miroslav Kubat, Springer, ISBN: 978-3319876696, August 2018.					
MOOC's/Swayam Courses/NPTEL/Online Courses: **NPTEL Course:**					

Title of the course	Offering University	Course URL
Introduction to Machine Learning	IITM	https://nptel.ac.in/courses/106106139

MOOCs Courses

1. Machine Learning, Coursera, 12 weeks, course link: https://www.coursera.org/learn/machine-learning?utm_source=gg&utm_medium=sem&utm_campaign=07-StanfordML-IN&utm_content=B2C&campaignid=1950458127&adgroupid=113440892778&device=c&keyword=&matchtype=&network=g&devicemodel=&adpostion=&creativeid=475416041431&hide_mobile_promo&gclid=EAIaIQobChMIqMm7i7TK9wIVlwsrCh1C8QlhEAAYASAAEgJxwPD_BwE
2. Applied Machine Learning using Python, Coursera, 8 weeks, course link: https://www.coursera.org/programs/minor-programs-offered-by-cse-c4xxe/browse?productId=di4l_R0lEeaP7xL2JHHq4w&productType=course&query=machine+learning&showMiniModal=true
3. IBM Machine Learning, Coursera, 8 weeks, course link: https://www.coursera.org/programs/minor-programs-offered-by-cse-c4xxe/browse?productId=Q0Fc_Yl0EeqdTApgQ4tM7Q&productType=s12n&query=machine+learning&showMiniModal=true
4. Artificial Intelligence Foundations: Neural Networks, Linkedin, 4 weeks, course link: https://www.linkedin.com/learning/artificial-intelligence-foundations-neural-networks?u=89447330

Global Certification Course:

Professional ML Engineer Certification |Learn| Google Cloud:, Proctored exam, Link: https://cloud.google.com/learn/certification/machine-learning-engineer

Global Challenges-Hackathons:

W1.https://allhackathons.com/hackathons/?search=machine+learning&status=upcoming&location=in_person&themes=23

W2: https://microsoft.github.io/AI_Agents_Hackathon/

W3: https://machinehack.com/hackathons/predict_the_flight_ticket_price_hackathon/overview

W4: https://machinehack.com/hackathons

W6: https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques

W7: For beginners - https://www.kaggle.com/competitions/titanic

W8: https://www.kaggle.com/competitions/amex-default-prediction

Topics relevant to "EMPLOYABILITY SKILLS": Topics of all four modules will help in developing **Employability Skills** through **Experiential Learning techniques**. This is attained through assessment component mentioned in course handout.

Course Code: PPS 4008	**Course Title:** **Quantitative Skills and Logical Reasoning** **Type of Course:**		**L- T -P- C**	**1** **–**	**0**	**2**	**2**
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 the course the students shall be able to: CO1] **Recall** all the basic mathematical concepts they learnt in high school. CO2] **Identify** the principle concept needed in a question. CO3] **Solve** the quantitative and logical ability questions with the appropriate concept. CO4] **Analyze** the data given in complex problems. CO5] **Rearrange** the information to simplify the question						
Course Content:							
Module 1	Quantitative Ability	Assignment	Bloom's Level : Application	**24 Hours**			
Topics: Introduction to Aptitude, Percentages, Profit & Loss, Ratio & Proportion, Averages, Time & Work, Time speed & Distance, Permutation and Combination, Probability, Data Interpretation							

Module 2	Logical Reasoning	Assignment	Bloom's Level : Application	**21 Hours**

Topics:

Linear & Circular Arrangement Puzzle, Coding & Decoding, Blood Relations, Directions, Number Series, Wrong number series, Visual Reasoning, Critical thinking, Syllogism

Text Book

1. Quantitative Aptitude by R S Aggarwal
2. Verbal & Non-Verbal Reasoning by R S Aggarwal

References

1. www.indiabix.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: PPS3001	**Course Title: Problem Solving through Aptitude Type of Course: Practical Only Course**	**L- T-P- C**	0	0	2	1
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 the course the students shall be able to: CO1] **Recall** all the basic mathematical concepts they learnt in high school. CO2] **Identify** the principle concept needed in a question. CO3] **Solve** the quantitative and logical ability questions with the appropriate concept. CO4] **Analyze** the data given in complex problems.					
Course Content:						
Module 1	Quantitative Ability	Assignment	Bloom's Level : Application	**10 Hours**		
Topics: Introduction to Aptitude, working of Tables, Squares, Cubes, Number Series, Wrong number series, Letter series.						
Module 2	Logical Reasoning	Assignment	Bloom's Level : Application	**20 Hours**		

Topics: Linear & Circular Arrangement Puzzle, Coding & Decoding, Blood Relations, Directions, Ordering and Ranking, Clocks and Calendars
Targeted Application & Tools that can be used: Application area: Placement activities and Competitive examinations. Tools: LMS

Course Code: CSA4601	**Course Title: Cloud Computing lab** **Type of Course: Lab**	**L-T- P- C**	0	0	2	1
Version No.	1.0					
Course Pre-requisites	Computer Networks					
Anti-requisites	NIL					
Course Description	This course provides a hands-on comprehensive study of Cloud concepts and capabilities across the various Cloud service models including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). It dives into all of the details that a student needs to know in order to plan for developing applications on the cloud and what to look for when using applications or services hosted on a cloud.					
Course Objective	The objective of the course is to familiarize the learners with the concepts of Cloud Computing and attain Skill Development through Experiential Learning techniques.					
Course Outcomes	Upon successful completion of the course the students shall be able to: Co1: Implement appropriate virtualization techniques to virtualize infrastructures Apply Co2: Analyze cloud mechanisms to optimize the QoS parameters - Analyse Co3: Apply applications using Cloud services and VM instances - Apply					
List of Experiments / Exercises						
Experiment No 1: **Level 1:** Create a basic form in Zoho Creator to collect student details (Name, Roll No, Department, Email).						

Level 2: Add field validations and make the email field mandatory. Test the form for different inputs.

Experiment No 2:

Level 1: Create a new application in Zoho Creator to manage a library system (Books, Authors, Issued Books).

Level 2: Establish relationships between forms (Books and Authors) using lookup fields.

Experiment No 3:

Level 1: Create a leave management system with a form to apply for leave.

Level 2: Implement an approval workflow with notifications for pending and approved leaves.

Experiment No 4:

Level 1: Create an online appointment scheduling system in Zoho Creator. Design forms to capture customer details, appointment date/time, and service required. Build a calendar view to display booked appointments.

Level 2: Deploy the application on AWS and provide users with access through a custom domain. Set up Elastic Load Balancing (ELB) to ensure that the application can scale efficiently for multiple users.

Experiment No 5:

Level 1: Create a cloud inventory management application using Zoho Creator with forms for product name, quantity, and price. Set up reports to display the product list and low stock alerts.

Level 2: Design a cloud-based survey application in Zoho Creator to capture user responses with questions like Name, Age, and Feedback. Use Deluge scripting to automatically calculate the response count for each question.

Experiment No 6:

Level 1: Install Oracle VirtualBox/VMware Workstation. Create a virtual machine and install Ubuntu/Linux OS.

Level 2: Configure virtual hardware (RAM, CPU, storage) and observe performance changes after modification.

Experiment No 7:

Level 1: Set up and run a Windows virtual machine in VirtualBox. Enable Guest Additions and test features like shared clipboard and drag-and-drop.

Level 2: Share folders between host and guest operating systems and test access from both sides.

Experiment No 8:

Level 1: Install and configure Hyper-V on a Windows system. Create and run a virtual machine.

Level 2: Enable and test nested virtualization on Hyper-V.

Experiment No 9:

Level 1: Evaluate and compare different cloud storage systems to understand their features, performance, and pricing.

Level 2: Assess cloud Storage systems and Cloud security, the risks involved, its impact and develop cloud application.

Experiment No 10:

Level 1: Using the Cloud Analyst Tool, simulate traffic distribution across multiple regions (North America and Europe) for an e-commerce platform. Write a program to configure load balancing across these regions, ensuring the traffic is intelligently routed to the closest region based on latency and server load. Display the simulated load distribution and response times for each region.

Level 2: Your organization is planning to migrate services to the cloud. Using Cloud Analyst, describe the step-by-step process to simulate the deployment of two data centers and three user bases. Based on the simulation results, how would you estimate the operational cost, and what configuration changes could you make to reduce the overall cost?

Experiment No 11:

Level 1: Write a Reducer to aggregate and analyze the extracted data (e.g., count occurrences).

Level 2: Write a Mapper to extract key information (e.g., IP addresses) from log entries.

Experiment No 12:

Level 1: Install Google App Engine. Create hello world app and other simple web applications using python/java.

Level 2: Find a procedure to launch virtual machine using trystack (Online Openstack Demo Version)

Experiment No 13:

Level 1: Set up a basic Hadoop cluster in a local or virtualized environment. Configure HDFS (Hadoop Distributed File System) to store data across multiple nodes. Verify the setup by running basic Hadoop commands such as hdfs dfs -ls and hdfs dfs -put.

Level 2: Extend the setup to include YARN (Yet Another Resource Negotiator) for resource management. Implement Hadoop MapReduce to process data stored in HDFS. Create a simple program to process a text file, count the occurrences of words, and store the results back in HDFS.

Experiment No 14:

Level 1: Write a basic MapReduce program in Java or Python that processes a dataset (such as a text file) and performs a word count. Execute the program on a Hadoop cluster and store the output in HDFS.

Level 2: Modify the MapReduce program to process a larger, more complex dataset. For example, implement a program that computes the average temperature from a weather dataset, storing the result in HDFS. Test the program's scalability by running it on larger data sets.

References/Manual/Software:

Targeted Applications:

Developing applications on Cloud Platforms via Virtual machines

Cloud Tools:

- CloudSim
- VMWare
- Amazon EC2
- Google Compute Engine
- Microsoft Azure

1. Daniel Vaughan, "Cloud Native Development with Google Cloud". O'Reilly Media Publishers. 1st Edition 2023.
2. Rajkumar Buyya, Christian Vecchiola, and Thamarai Selvi, "*Mastering Cloud Computing*", McGraw Hill Education, 2017edition.
3. John Rittinghouse and James Ransome, "*Cloud Computing, Implementation, Management and Security*", CRC Press, 2010 edition**.**

Course Code: **CSA4304**	**Course Title: Object Oriented Programming using Java lab** **Type of Course: Lab**	**L-T- P- C**	0	0	4	2
Version No.	1.0					
Course Pre-requisites	**Basic programming Skills**					

Anti-requisites	**NIL**
Course Description	The main objective is to learn the basic concept and techniques which form the object-oriented programming paradigm. Object-oriented programming is a new way of thinking about problem using models organized around real world concept. It investigates the software engineering principles of encapsulation, information hiding and code reuse, and discusses how these concepts are used to build abstract data types. The object oriented programming features of Sessions, inheritance, polymorphism and composition are studied, along with constructors and method overloading. Students implement Java programs incorporating features from the Java programming language.
Course Objective	The objective of the course is to familiarize learners with Object-Oriented Programming concepts using Java, while developing general-purpose applications with database connectivity through experiential learning and object-oriented design principles.
Course Outcomes	**On successful completion of this course, the students shall be able to:** CO1: Apply Object-Oriented Programming principles in Java to design modula and reusable code that efficiently solve the real-world problems.(Apply) CO2: Utilize the concepts of Inheritance, Multithreading, and Exception Handling to develop efficient and robust code. (Apply) CO3: Develop Serverside java applications using Servlet and JSP concepts. (Apply) CO4: Construct basic applications that demonstrate efficient interaction with relational database systems through JDBC and Hibernate frameworks. (Apply)
List of Experiments / Exercises	

List of Laboratory Tasks:

Lab sheet -1 [5 Practical Sessions]

Experiment No 1:

Level1 -Programs using Control statements⍰ Methods with Parameters, Methods with control statements

Level2- Demonstrations of Class, Object, Constructor, Static member, Encapsulation, Inner Class

Experiment No. 2:

Level 1 –Simple Program for Understanding Arrays and Strings.

Level2 - Programs to implement array of objects, passing and returning objects as arguments.

Lab sheet – 2 [2 Practical Sessions]

Experiment No. 1:

Level1 - Programs to demonstrate concepts of constructors and destructors

Level2 - Write a program to create a database for a bank account contains Name, Account no, Account type, Balance, Including the following – any constructor, destructor and methods to set and get information for 10 people.

Experiment No. 2:

Level1 – Programs to implement methods of String and String Buffer Class.

Level2 - **Programs to implement Inheritance and Polymorphism, Programs to implements Interface.**

Lab sheet – 3 [3Practical Sessions]

Level 1 - Programs to demonstrate Exceptions Handlers.

Level 2 - Programs to implements nested handlers, Checked and Unchecked Exception Handlers.

Lab sheet – 4 [4 Practical Sessions]

Level 1 -Programs to implement Thread class and Runnable Interface.

Level 2 - Programs to implement priority, inter thread communication.

Lab sheet –5 [1 Practical Session]

Experiment No. 1:

Level 1 - Programs to implement Collections (List, Set, Map).

Level 2 - Programs to implement Comparable and Comparator Interface , Lambda Notation

Lab sheet 6 [2 Practical Session]]

Experiment No. 1:

Level 1 –Programs to implement concepts of GUI.

Level 2 – Programs to create Registration form using Swing.

Targeted Application & Tools that can be used: Notepad++, Eclipse IDE, NetBeans IDE

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

Programming: Implementation of given scenario using Java

1. Develop a Library management system with basic modules and users like

- Database module: This has two functions – Insertion of data and extraction of data.
- Report module: For the borrowed books list to display.

- Availability module: To view the availability of books.
- Search Module: search facility for books and members.

Users in the system:

- Librarian
- Student

User functions:

- Librarian: Add, view, delete the book details and user details, issue and return books.
- Student: view and requesting books, returning books.

2. Design an employee payroll management system with basic modules and its processes as

Admin:

- Admin can Add/Edit/delete the employees.
- Admin can Add/Edit/delete the schedule the work of the employees.
- Admin can Add and calculate/Edit/Delete the Salary of the employee.

Employee:

- Employees can view his/her schedule set by Admin.
- Employees can check his/her attendance.
- Employees can update his/her details.
- Employees can View their salary details.

3. Design an online Quiz system with basic modules and its processes as follows

Users of the System

- Teacher
- Student

Functional Requirements

Teacher:

- Can create quiz after getting logged in.
- Can enter subjects and enter question with its options and answer at the time of creating quiz.
- 10 Question for each quiz required to be completed.

Student:

- Can search quiz according to their interest.
- select the id of quiz and ready to start it.
- After completing all questions, result will be displayed automatically.

- Can view the description about each and every question in the respective quiz

References

1. Cay Horstmann,"Core Java -Volume 1: Fundamentals", 12th Edition, Oracle Press, 2021.
2. Bruce Eckel, **Thinking in Java. 4th ed**.
3. R. Nageswara Rao**, Core Java: An Integrated Approach, New: Includes All Versions upto Java 8**
4. Brett McLaughlin, **Head First Object-Oriented Analysis and Design: A Brain Friendly Guide to OOA&D**

Web References

1. NPTEL Course on "Java Programming", Prof. Debasis Samanta, https://archive.nptel.ac.in/courses/106/105/106105191/

2. "Head First Java" by Kathe Siera and Bert Bates, 2nd edition https://www.rcsdk12.org/cms/lib/NY01001156/Centricity/Domain/4951/Head_First_Java_Second_Edition.pdf.

3. "Building java programs" https://presiuniv.knimbus.com/user#/searchresult?searchId=java%20programming&_t=1662620793642

Course Code: **CSA4602**	Course Title: Machine Learning Lab Type of Course: Lab	L-T-P-C	0	0	2	1
Version No.	1.0					
Course Pre-requisites	**NIL**					
Anti-requisites	NIL					
Course Description	This course builds the foundational insight of understanding principles of machine learning process and to explore various ML algorithms and practices to accelerate strategic decision making with data. Overall, this course gives a comprehensive insight to appropriately choose ML algorithms on real time problems to construct an intelligent ML model.					
Course Objective	The objective of the course is to familiarize the learners to explore the competence and comprehend with potential machine learning algorithms and techniques to revolutionize with real-world problems and create prominent solutions (with ML models) to attain Employability Skills through Experiential Learning techniques.					
Course Out Comes	On successful completion of the course the students shall be able to: Co1: Apply SVM and Evolutionary Learning Ensemble methods for solving classification problems Co2: Practice unsupervised models, Explanation-Based reinforcement learning methods using Python language Co3: Solve real-time applications using Machine Learning Algorithms					
Course Content:						
List of Laboratory Tasks:						
Experiment No.1: Implementation and Evaluation of Simple Linear Regression Model Level 1: Use dataset for generating ML model for sales forecasting Level 2:						
Experiment No.2: Implementation and Evaluation of Multi-Linear regression Model Level 1: Practice multi-output regression model to capture non-linear relationships for energy consumption forecasting application						
Experiment No.3: Implement and Evaluate decision Tree Model using ID3 Algorithm with GINI Index Level 1: Use generating categorical inputs for Credit card Fraud detection Level 2: Apply the GINI Index to develop ML model						
Experiment No.4: Building a decision Tree using ID3 Algorithm and Information Gain and Evaluate the results.						

Level 1: Apply ID3 algorithm for Customer Churn Prediction (telecom/SaaS) Level 2: Apply lead code problems on Decision tree using ID3 algorithm
Experiment No.5: Implement Bayesian Classification Model Level 1: Develop ML model for classifying text/email
Experiment No.6: Apply Bayesian Belief Network Level 1: Develop BFN based model for disease prediction
Experiment No.7: Implement Support Vector Machine Model Level 1: Practice various types of Supervised learning algorithms Level 2: Practice different types of classification algorithms (Handwritten digits, classify tumors as malignant or benign.
Experiment No.8: Build an Artificial Neural Network Model Level 1: Use ANN to identify objects and make decisions/ detect anomalies in X-rays or MRIs
Experiment No.9: Build a Back Propagation Model Level 1: Apply back propagation to generate information retrieval model from custom document
Experiment No.10: Implement PCA on a given dataset and implement any classification model Level 1: Show how dimensionality reduction retains its accuracy and removes noise and redundancy features for facial feature dataset
Experiment No.11: Implement ADA Boosting Algorithm Level 1: Understand to handle imbalanced data, working with noisy data and non-linear relationships Level 2: Use misclassified fraudulent transaction data that uses AdaBoost to train the sequence of weak classifiers
Experiment No.12: Implement XG Boost Classifier Level 1: Demonstrate an online advertising via a click-through rate prediction and generate a customer churn prediction by applying various service apps.
Experiment No.13: Implement LDA on a Given Dataset and Implement any Classification model Level 1: List various classification/Regression models Level 2: Apply LDA on an 3D image to generate a
Experiment No.14: Implementation of Random Forest Algorithm Level 1: Level 1: List various classification/Regression models Level 2: Use a credit scoring dataset to generate a ML model for approve/reject loan
Experiment No.15: Implementation of K-means Clustering Level 1: Identify the generative model for clustering Level 2: Use to create models to group similar documents using features like TF-IDF.
Experiment No. 16: Implementation of Hierarchical Clustering Level 1: Identify the generative model for clustering Level 2: Use to create models on personalized treatment recommendations based on patient clusters
References/Manual/Software:

1. Jupyter Notebook/ JupyterLab, VS code(Code Editor), PyCharm (Python IDE) - IDEs
2. Flask/FastAPI – Python web frameworks
3. Docker – Containerize and ship ML apps
4. Google Colab / Kaggle Kernels – cloud-based GPU access
5. Pytorch, Scikit-learn, Tensorflow

T1: Machine Learning: An Algorithmic Perspective, Stephen Marshland, 2nd Edition, CRC Press, Taylor & Francis group, ISBN: 978-1-4665-8333-7, November 2014.

T2: Machine Learning in Action, Peter Harrington, ISBN: 978-935-004-4131, April 2012.

T3: Introduction to Machine Learning, Ethem AIpaydin, 4th Edition, The MIT press, ISBN:978-0-262-043-793, March 2020.

Topics relevant to “EMPLOYABILITY SKILLS”: Topics of all four modules will help in developing Employability Skills through Experiential Learning techniques. This is attained through assessment component mentioned in course handout.

Semester 3

<table>
<tr><td><b>Course Code:</b>
CSA4503</td><td colspan="3"><b>Course Title:</b> <b>Data Analytics and Visualization</b>
<b>Type of Course:</b> <b>Program Core</b></td><td>L-T- P- C</td><td>2</td><td>0</td><td>0</td><td>2</td></tr>
<tr><td><b>Version No.</b></td><td colspan="8">1.0</td></tr>
<tr><td><b>Course Pre-requisites</b></td><td colspan="8">Probability and Statistics, Advanced Python Programming</td></tr>
<tr><td><b>Anti-requisites</b></td><td colspan="8"><b>NIL</b></td></tr>
<tr><td><b>Course Description</b></td><td colspan="8">The Course consists of two parts where first Part covers advanced analytics that covers topics necessary to give businesses greater insight into their data than they could ordinarily, and the Second Part covers data visualization concepts. Primary concepts include machine learning, data mining, predictive analytics, location analytics, big data analytics, and location intelligence. Visualization for Time series, Geolocated data, Correlations, connections, Hierarchies, networks, and interactivity.</td></tr>
<tr><td><b>Course Objective</b></td><td colspan="8">The objective of the course is EMPLOYBILITY of student by using EXPERIENTIAL LEARNING techniques.</td></tr>
<tr><td><b>Course Outcomes</b></td><td colspan="8"><b>On successful completion of this course, the students shall be able to:</b>
<b>CO1</b>: Apply fundamental concepts of Data Analytics and the Data Science process using Python and essential libraries. <b>(Apply)</b>
<b>CO2:</b> Implement data manipulation and preprocessing techniques using NumPy and pandas to analyze and clean diverse datasets. (<b>Apply)</b>
<b>CO3</b>: Analyze data visualization techniques using matplotlib, seaborn, and pandas to interpret patterns, identify trends, and derive insights from datasets effectively. <b>(Analyze)</b>
<b>CO4:</b> Utilize advanced data aggregation and time series analysis methods to handle and analyze real-world time-dependent datasets. <b>(Apply)</b></td></tr>
<tr><td><b>Course Content:</b></td><td colspan="8">The objective of the course is EMPLOYBILITY of student by using PARTICIPATIVE LEARNING techniques.</td></tr>
<tr><td><b>Module 1</b></td><td><b>Introduction to Data Analytics - CO1</b></td><td>Assignment</td><td>Programming Task</td><td colspan="5"><b>Sessions: 7</b></td></tr>
<tr><td colspan="9">Introduction to Data Analytics, Exploratory Data Analysis and Data Science Process. Motivation for using Python for Data Analysis. Python Libraries: NumPy, pandas, matplotlib, SciPy, scikit-learn, statsmodels.</td></tr>
<tr><td><b>Module 2</b></td><td><b>Data Analytics with Python – CO2</b></td><td>Assignment</td><td>Programming Task</td><td colspan="5"><b>Sessions 8</b></td></tr>
</table>

<table>
<tr><td colspan="5">Arrays and vectorised computation, Introduction to pandas Data Structures, Essential Functionality, Summarizing and Computing Descriptive Statistics. Data Loading, Storage and File Formats. Reading and Writing Data in Text Format Data Cleaning and Preparation. Handling Missing Data, Data Transformation, String Manipulation.</td></tr>
<tr><td><b>Module 3</b></td><td><b>Introduction to Data Visualization – CO3</b></td><td>Assignment</td><td><b>Programming Task</b></td><td><b>Sessions 7</b></td></tr>
<tr><td colspan="5">Data Wrangling: Hierarchical Indexing, Combining and Merging Data Sets Reshaping and Pivoting. Data Visualization matplotlib: Basics of matplotlib, plotting with pandas and seaborn, other python visualization tools.</td></tr>
<tr><td><b>Module 4</b></td><td><b>Data Aggregation and Visualization – CO4</b></td><td>Assignment</td><td><b>Programming Tasks</b></td><td><b>Sessions: L8</b></td></tr>
<tr><td colspan="5">Data Aggregation and Group operations: Group by Mechanics, Data aggregation, General split-apply-combine, Pivot tables and cross tabulation. Time Series Data Analysis: Date and Time Data Types and Tools, Time series Basics, date Ranges, Frequencies and Shifting, Time Zone Handling, Periods and Periods Arithmetic, Resampling and Frequency conversion, Moving Window Functions.</td></tr>
<tr><td colspan="5"><b>Targeted Application & Tools that can be used:</b>
Exploratory Data Analysis, Data Wrangling, Statistical Analysis, Data Visualization, and Time Series Analysis using Python libraries such as NumPy, pandas, matplotlib, seaborn, SciPy, scikit-learn, and statsmodels.</td></tr>
<tr><td colspan="5"><b>Project work/Assignment:</b>
1. Customer Segmentation using EDA and Clustering
2. COVID-19 Data Analysis and Visualization
3. Stock Market Trend Analysis
4. Sales Data Cleaning and Reporting Dashboard
<b>5.</b> Social Media Sentiment Analysis (Basic Level)</td></tr>
<tr><td colspan="5"><b>Topics related to</b>
1. Problem Solving: Developing and optimizing machine learning algorithms for predicting stock market trends using real-time data.
2. Employability: Simulation of AI-based customer support chatbots for automating customer service workflows in e-commerce platforms using NLP and sentiment analysis.</td></tr>
<tr><td colspan="5"><b>Textbook(s):</b>
<b>T1.</b> McKinney, W. (2022). Python for Data Analysis: Data Wrangling with Pandas, NumPy and IPython. 3rd edition. O’Reilly Media.
<b>T2</b>. Zaki, Mohammed J., Wagner Meira Jr, and Wagner Meira. <i>Data mining and machine learning: Fundamental concepts and algorithms</i>. Cambridge University Press, 2020.
<b>T3</b>. O’Neil, C., & Schutt, R. (2013). Doing Data Science: Straight Talk from the Frontline O’Reilly Media.</td></tr>
<tr><td colspan="5"><b>References</b></td></tr>
</table>

R1. Mukhiya, Suresh Kumar, and Usman Ahmed. *Hands-On Exploratory Data Analysis with Python: Perform EDA techniques to understand, summarize, and investigate your data*. Packt Publishing Ltd, 2020.

R2. George, Nathan. *Practical data science with Python: learn tools and techniques from hands-on examples to extract insights from data*. Packt Publishing Ltd, 2021.

R3. Gutman, Alex J., and Jordan Goldmeier. *Becoming a data head: How to think, speak, and understand data science, statistics, and machine learning*. John Wiley & Sons, 2021.

E-Resources:

1. https://scikit-learn.org/stable/
2. https://www.istqb.org/
3. https://machinelearningmastery.com/time-series-forecasting-with-python/
4. https://matplotlib.org/
5. https://www.deeplearning.ai/

Course Code: **CSA4504**	**Course Title: MERN Full Stack Development** **Type of Course: 1]Program Core**	**L-T- P- C**	2	0	0	2
Version No.	1.0					
Course Pre-requisites	Web Development					
Anti-requisites	**NIL**					
Course Description	It provides students with a comprehensive understanding of full-stack development, covering both frontend and backend technologies. Mern Stack course includes MongoDB (database), Express.js (backend framework), ReactJS (frontend), and Node.js (runtime environment).The course offers hands-on experience in building real-world web applications using the Mern Stack. This practical approach allows students to apply theoretical concepts in a practical setting, enhancing their learning experience. This course incorporates modern technologies and practices such as NoSQL databases, RESTful APIs, and single-page application development. By studying Mern Stack development, students become familiar with these technologies, preparing them for the demands of the tech industry.					
Course Objective	This course aims to equip learners with industry-relevant skills in Full Stack Web Development using the MERN (MongoDB, Express.js, React.js, Node.js) stack.					

Course Outcomes	**On successful completion of this course, the students shall be able to:** 1]Experiment with Git, MongoDB Database and MySQL Database [Understand] 2]Utilize Node Modules and ReactJS[Apply] 3]Develop dynamic websites[Apply] 4]Build Web Applications using Scripts and deploy with tools[Apply]			
Course Content:	MongoDB Database, NodeJS, ReactJS, CSS, Bootstrap, & Responsive Design, NoSQL, Node Package Manager.			
Module 1	**Basics of Full Stack- CO1 (Experiment with Git, MongoDB Database and MySQL Database)**	Term paper/Assignment/Case Study	Programming	**7 Sessions**
Understanding the Basic Web Development Framework - User - Browser – Webserver - Backend Services – MVC Architecture - Understanding the different stacks –The role of Express – Node – Mongo DB – React				
Module 2	**Node JS Modules – CO2 (Utilize Node Modules and ReactJS)**	Term paper/Assignment/Case Study	Programming	**7 Sessions**
Basics of Node JS – Working with Node packages – Using Node package manager – Creating a simple Node.js application – Using Events – Listeners –Timers - Callbacks – Handling Data I/O – Implementing HTTP services in Node.js				
Module 3	**No SQL Database - Mongo DB– CO3 (Develop dynamic websites)**	Term paper/Assignment/Case Study	Programming	**8 Sessions**
Understanding NoSQL and MongoDB – Building MongoDB Environment – User accounts – Access control – Administering databases – Managing collections – Connecting to MongoDB from Node.js – simple applications				
Module 4	**Front End Library - React Js – CO4 (Build Web Applications using Scripts and deploy with tools)**	Term paper/Assignment/Case Study	Programming	**8 Sessions**

Mern Stack – Basic React applications – React Components – React State – Express REST APIs - Modularization and Webpack - Routing with React Router – Server-side rendering

Project work/Assignment:

Create dynamic, interactive, and scalable web applications.

Topics related to

1. MongoDB: Ensure proper schema design and indexing to optimize queries. Use tools like MongoDB Compass for visual debugging
2. Express.js: Implement middleware for error handling and logging. Use tools like Postman to test API endpoints
3. React: Utilize React Developer Tools for inspecting component hierarchies and state. Handle errors gracefully with error boundaries.
4. Node.js: Use debugging tools like Node Inspector and logging libraries like Winston to track server-side issues.

Textbook(s):

T1. Vasan Subramanian, 'Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React,
and Node', Second Edition, Apress, 2020.

T2 Brad Dayley, Brendan Dayley, Caleb Dayley, 'Node.js, MongoDB and Angular Web Development', Addison-Wesley, Second Edition, 2021.

References

R1. Chris Northwood, 'The Full Stack Developer: Your Essential Guide to the Everyday Skills Expected of a Modern Full Stack Web Developer', Apress; 3rd edition, 2023.

R2. Kirupa Chinnathambi, 'Learning React: A Hands-On Guide to Building Web Applications Using React and Redux', Addison-Wesley Professional, 3rd edition, 2022

E-Resources:

1. https://www.tutorialspoint.com/the_full_stack_web_development/index.asp
2. https://www.coursera.org/specializations/full-stack-react
3. https://www.udemy.com/course/the-full-stack-web-development/

Course Code: **PPS 3019**	**Course Title: Corporate Communication** Type of Course: Practical Only Course	L- T - P- C	0	0	2	1
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 understand concepts of corporate communication, improve confidence, and develop professional skills. This course will give the students a competitive advantage and increase their chances of success in the professional world by building their strong resumes and better online presence. This will benefit learners in effectively presenting their concepts and ideas in a corporate environment through various activities and learning methodologies.		
Course Objective	The objective of the course is to familiarize the learners with the concepts of "Corporate Communication" and attain SKILL DEVELOPMENT through PARTICIPATIVE LEARNING techniques.		
Course Out Comes	**On successful completion of this course, the students shall be able to:** CO1: Demonstrate effective communication while presenting oneself and others CO2: Design resume based on industrial standards CO3: Employ techniques to face interview and Group discussion		
Course Content:			
Module 1	Introduction to Corporate Communication	Classroom activity	6 Hours
Topics: Setting Expectations, Ice Breaker, Cs of communication, Significance of corporate communication, Communication process, barriers in communication and solutions, Structuring messages, non-verbal communication, written communication, Self introduction.			
Module 2	Presentation Skills	Group Presentation	8 Hours
Topics: Presentation Skills, Opening Body & Closing Body, Audibility, Speech Clarity, Fluency, Voice Modulation, Non-verbal Communication and Body Language.			
Module 3	Resume Writing	Individual Assessment	8 Hours
Topics: Importance of a Resume, Types of Resumes (Chronological, Functional, and Combination), Understanding Job Descriptions, Optimizing for Applicant Tracking Systems (ATS), Understanding ATS Requirements, Formatting Tips for ATS Compliance, The Digital Edge: Online Profiles (Aligning Resumes with LinkedIn Profiles)			

Module 4	Personal Interview	Individual Assessment	8 Hours

Understanding Personal Interviews, Self-Awareness and Preparation (SWOT Analysis: Knowing Your Strengths and Weaknesses), **Handling Common Questions (**Tell Me About Yourself: Structuring a Strong Response, Answering Situational and Behavioral Questions (STAR Method) **Practical Skills for Success (**Dressing for Success: Professional Appearance, Mock Interviews: Practice with Feedback, Follow-Up Etiquette: Thank You Emails and Calls)

Targeted Application & Tools that can be used:

1. TED Talks
2. You Tube Links
3. Activities

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

1) Resume
2) Self-Introduction

Personal Interview

Targeted Application & Tools that can be used:

1. TED Talks
2. YouTube Links

Assignments proposed for this course

1. Self-Introduction
2. Resume

YouTube Links: https://youtu.be/z_jxoczNWc (Steve Jobs Introducing the iPhone 4 in June 2010) References

1. “Talk Like TED - The 9 Public-Speaking Secrets of the World's Top Minds” By Carmine Gallo St. Martin's Press Copyright © 2014 Carmine Gallo All rights reserved. ISBN: 978-1-250-04112-8
2. “The Presentation Secrets of Steve Jobs: How to Be Insanely Great in Front of Any Audience” MP3 CD – Import, 22 April 2014
3. “The Definitive Book of Body Language: The Hidden Meaning Behind People's Gestures and Expressions” Hardcover – Illustrated, 25 July 2006
4. “Crucial Conversations: Tools for Talking When Stakes Are High” Paperback – Import, 1 July 2002

Web links:

1. https://www.wordstream.com/blog/ws/2014/11/19/how-to-improve-presentation-skills https://www.cbs.de/en/blog/15-effective-presentation-tips-to-

improve-presentation-skills/

2. https://hbr.org/2022/05/the-art-of-asking-great-questions

Topics relevant to development of "SKILL": Art of Presentation, Team building, Art of questioning and Personal Branding for Skill Development through Participative Learning Techniques. This is attained through assessment component mentioned in course handout.

Course Code: **MAT4002**	**Course Title: Introduction to Operation Research** **Type of Course:1] School Core**		**L-T- P- C**	**2**	**0**	**0**	**2**
Version No.		1.0					
Course Pre-requisites		MAT4001					
Anti-requisites		**NIL**					
Course Description		This course aims to aid the students to appreciate the need for fundamentals of Operational Research and basics of OR. Following this, various phases, Solving Linear Programming Problems using Various Methods. Special cases of LPP. Various real-life applications to make Optimum Decisions in various fields. The Course provides various Network systems will enable the students to carry out their research and development activities, placement opportunities and foundation to design the cellular architecture.					
Course Objective		This course provides a strong foundation for understanding the fundamental principles and laws of Operation Research to understand Linear programming, network Analysis and Game theory, Inventory models. Students can understand and solve the problems regarding management and application of various models in Engineering. The objective of the course is to **familiarize the learners with concepts of "Fundamentals of Operation Research"** and attain **Skill Development through Problem Solving Techniques**.					
Course Out Comes		On successful completion of the course the students shall be able to: CO1 - To Ability to solve the mathematical results and numerical techniques of optimization theory to concrete Petroleum Engineering problems by using computer software CO2 - To create analytical ability among students by using Mathematical techniques and decision-making ability.					

		CO3 - To solve problems based on transportation, Decision Theory & Networks. CO4 - To Ability to apply the theory of optimization methods and algorithms to develop and solve various types of optimizations.	
Course Content:			
Module 1	**Linear Programming**		**(7 Classes)**
Introduction to OR: Evolution of OR, Definitions of OR, Scope of OR, Applications of OR, Phases in OR study. Characteristics and limitations of OR, Models used in OR. Linear Programming Problem (LPP), Generalized LPP- Formulation of problems as L.P.P. Solutions to LPP by graphical method (Two Variables). Simplex method, Canonical and Standard form of LP problem, slack, surplus and artificial variables, Solutions to LPP by Simplex method using Software.			
Module 2	**Transportation and Assignment Problems**	**Assignment**	**(7 Classes)**
Transportation Problem: Formulation of transportation problem, types, initial basic feasible solution, Balanced and Unbalanced Problems by using North-West Corner rule, Least-cost cell method, Vogel's Approximation method. Optimality in the Transportation problem by Modified Distribution (MODI) method. Application of transportation problems. **Assignment Problem**- Formulation, Solutions to assignment problems by the Hungarian method, Travelling Salesman Problem (TSP).			
Module 3	**Network Models and Sequencing**		**(10 Classes)**
Shortest Path Problem (Dijkstra's Algorithm), Minimal Spanning Tree (Prim's & Kruskal's Algorithms), Maximal Flow Problem (Ford-Fulkerson Algorithm). Project Scheduling (PERT & CPM) - Construction of networks, Critical path method to find the expected completion time of a project, determination of floats in networks, PERT networks, determining the probability of completing a project, predicting the completion time of project; Cost analysis in networks. Crashing of networks – Problems **Sequencing:** Introduction, processing N jobs through two machines, processing N jobs through three machines, processing N jobs through m machines.			
Module 4	**Game Theory**	**Assignment**	**(6 Classes)**
Definition, Pure Strategy problems, Saddle point, Max-Min and Min-Max criteria, Principle of Dominance, Solution of games with Saddle point. Mixed Strategy problems. Solution of 2x2 games by Arithmetic method, Solution of 2xn m and mx2 games by graphical method. Formulation of games.			
Targeted Application & Tools that can be used: The objective of the course is to familiarize students with a variety of numerical techniques and the theoretical concepts of Operational Research to equip them with the necessary numerical approaches and basic statistical tools to tackle engineering and real-life problems. Tools Used: Lindo Software			

Assignment:

1. Simplex Method
2. Transportation
3. EOQ model with quantity discount
4. Solution of 2xn m and mx2 games by graphical method.

Text Book

1. Hamdy. A. Taha, Operations Research: An Introduction, Pearson Education, 10th Ed., 2022.
2. Frederick S Hiller & Gerald J Liberman, Introduction Operations Research, Bodhibrata Nag and Preetam Basu 11th Ed 2022.
3. J K Sharma, Operations Research Theory and Applications,6th Ed , 2020.: With Applications to Management, Decisions, Prentice-Hall of India, New Delhi.

References:

1. N D Vohra, Quantitative Techniques in management, Tata McGraw Hill.
2. E. K. P. Chong and S. H. Zak, An Introduction to Optimization, 2nd Edn., Wiley India Pvt. Ltd., 2010.
3. Hillier F S and Lieberman G J, Operations Research, Holden Day Inc., San Francisco.

E-resources/ Web links:

1. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=SCIENCEDIRECTSTANDARD_GLOBAL_ALLTITLES_2022_10_18_19102022_12338
2. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=JSTOR_GLOBAL_MUSEUMSCOLLECTION_2022_10_17_18102022_2632
3. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=JSTOR_GLOBAL_MUSEUMSCOLLECTION_2022_10_17_18102022_3006
4. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_216519
5. https://www.math.hkust.edu.hk/~maqian/ma006_0607F.html
6. https://www.scu.edu.au/study-at-scu/units/math1005/2022/

Topics relevant to SKILL DEVELOPMENT This course focuses on formulating and solving problems concerning real-world engineering applications numerically as well as statistically. This course provides an introduction to basic Operational research methods to deal with Linear Programming Problems, Finding Optimum Basic feasible solution, Transportation and Assignment problems, Finding Total duration of Completing Project, Inventory models and Theory of games for **Skill Development** **through** **Problem Solving methodologies**. This is attained through the assessment component mentioned in the course handout.

Course Code: **CSA4603**	**Course Title:** **Data Analytics and Visualization Lab** **Type of Course:** **Program Core**	**L-T- P- C**	0	0	2	1
Version No.	1.0					
Course Pre-requisites	Probability and Statistics, Advanced Python Programming					
Anti-requisites	**NIL**					
Course Description	This laboratory course is designed to provide hands-on experience in data analytics using Python and its powerful libraries such as NumPy, pandas, matplotlib, seaborn, SciPy, and statsmodels. The focus is on applying analytical techniques to real-world datasets, enabling learners to explore, clean, visualize, and interpret data effectively. Through structured experimentation, students will develop the practical skills necessary for implementing data science workflows, statistical analysis, and time series forecasting.					
Course Objective	The objective of the course is EMPLOYBILITY of student by using EXPERIENTIAL LEARNING techniques.					
Course Outcomes	**On successful completion of this course, the students shall be able to:** **CO1**: Apply NumPy and pandas to perform fundamental data manipulation, indexing, filtering, and cleaning tasks on structured datasets. (Apply) **CO2:** Use Python libraries to load, process, and transform datasets, including handling missing data, performing string operations, and converting formats. (Apply) **CO3**: Apply visualization techniques using matplotlib, pandas, and seaborn to generate graphical representations that highlight data patterns, distributions, and relationships for effective data interpretation. (Apply) **CO4:** Apply aggregation techniques, time series analysis, and hierarchical indexing to perform advanced analytical operations on real-time data**.** (Apply)					
List of Experiments / Exercises						
Experiment No. 1: **Level 1: Introduction to NumPy arrays – Creation, indexing, slicing** **Level 2: Mathematical operations and broadcasting using NumPy** **Experiment No. 2:** **Level 1: Getting started with pandas – Series and DataFrame creation** **Level 2: Basic data inspection, selection, filtering using pandas** **Experiment No. 3:** **Level 1: Introduction to matplotlib – Line and bar charts**						

Level 2: Customizing plots with titles, labels, and legends

Experiment No. 4:

Level 1: Overview of Python libraries – Exploring SciPy and statsmodels

Level 2: Use of simple statistical functions for descriptive analysis

Experiment No. 5:

Level 1: Data loading from CSV and Excel using pandas

Level 2: Writing cleaned data to different file formats

Experiment No. 6:

Level 1: Handling missing values – Detection and imputation techniques

Level 2: Data transformation and type conversions

Experiment No. 7:

Level 1: String operations and regular expressions in pandas

Level 2: Cleaning and formatting textual data

Experiment No. 8:

Level 1: Summary statistics using describe, mean, std functions

Level 2: Custom aggregations and value counts

Experiment No. 9:

Level 1: Combining datasets – Merge, join, and concat operations

Level 2: Reshape datasets using pivot and melt

Experiment No. 10:

Level 1: Plotting with pandas built-in visualization

Level 2: Create statistical plots using seaborn

Experiment No. 11:

Level 1: Hierarchical indexing – Creation and manipulation

Level 2: Group-wise transformations using multi-index

Experiment No. 12:

Level 1: GroupBy mechanics and applying aggregation functions

Level 2: Create pivot tables and cross-tab reports

Experiment No. 13:

Level 1: Time series basics – Parsing dates, datetime index

Level 2: Resampling, shifting, and moving average

Experiment No. 14:

Level 1: Working with time zones and frequency conversion

Level 2: Periods arithmetic and time series forecasting preparation

References/Manual/Software:

1. Python for Data Analysis by Wes McKinney – A comprehensive guide to using pandas, NumPy, and matplotlib for data analysis.

2. Think Stats: Exploratory Data Analysis in Python by Allen B. Downey – Focuses on statistical analysis using Python.
3. Pandas Documentation – https://pandas.pydata.org/docs/
4. Anaconda Distribution (includes Jupyter, Python, NumPy, pandas, matplotlib, seaborn) – https://www.anaconda.com/products/distribution
5. Jupyter Notebook Environment – https://jupyter.org/

Course Code: CSA4604	**Course Title: MERN Full Stack Development Lab** **Type of Course: Program Core**	**L-T- P- C**	0	0	4	2
Version No.	1.0					
Course Pre-requisites	Web Development					
Anti-requisites	**NIL**					
Course Description	It provides students with a comprehensive understanding of full-stack development, covering both frontend and backend technologies. Mern Stack course includes MongoDB (database), Express.js (backend framework), ReactJS (frontend), and Node.js (runtime environment).The course offers hands-on experience in building real-world web applications using the Mern Stack. This practical approach allows students to apply theoretical concepts in a practical setting, enhancing their learning experience. This course incorporates modern technologies and practices such as NoSQL databases, RESTful APIs, and single-page application development. By studying Mern Stack development, students become familiar with these technologies, preparing them for the demands of the tech industry.					
Course Objective	This course aims to equip learners with industry-relevant skills in Full Stack Web Development using the MERN (MongoDB, Express.js, React.js, Node.js) stack.					
Course Outcomes	**On successful completion of this course, the students shall be able to:** 1]Experiment with Git, MongoDB Database and MySQL Database [Understand] 2]Utilize Node Modules and ReactJS[Apply]					

	3]Develop dynamic websites[Apply] 4]Build Web Applications using Scripts and deploy with tools[Apply]

List of Experiments

List of Laboratory Tasks:

Experiment No. 1:

Level 1: Install Git on local machines. Initialize a local Git repository. Perform basic Git operations such as adding files, committing changes, and viewing the commit history.

Level 2: Practice creating branches and merging changes. Push changes to a remote repository (e.g., GitHub, GitLab).

Experiment No. 2:

Level 1: Install Node.js and npm on local machines.

Level 2: Create a simple Node.js application. Use npm to manage dependencies and install packages.

Experiment No. 3:

Level 1: Implement basic server functionalities such as handling HTTP requests.

Level 2: Implement basic server functionalities such as handling HTTP responses.

Experiment No. 4:

Level 1: Run and test the Node.js application locally.

Level 2: Creating Application with npm init and Installed Modules.

Experiment No. 5:

Level 1: Create a message.txt file in the same directory and add some content in node.js

Level 2: Creating Web-based Node Application.

Experiment No. 6:

Level 1: Creating a server in Node

Level 2: Read request and return response in Node.

Experiment No. 7:

Level 1: Create a Calculator Node.js Module with functions add, subtract and multiply. And use the Calculator module in another Node.js file.

Level 2: Create node.js program with the content ‘FS module’ to perform file operations.

Experiment No. 8:

Level 1: Install MongoDB on local machines or a virtual environment. Configure MongoDB to run as a service.

Level 2: Access the MongoDB shell and perform basic database operations (e.g., creating databases, collections, inserting documents).

Experiment No. 9:

Level 1 : Perform CRUD operations (Insert, Update, Delete and Query Documents) on 'Student' Database.

Level 2 : Do queries involving MongoDB update operator.

Experiment No. 10:

Level 1 : Perform different query modifiers on 'Student' Database

Level 2 : Implement different aggregation commands on 'Student' Database.

Experiment No. 11:

Level 1 : Perform CRUD operations (Insert, Update, Delete and Query Documents) on Employee Information Database.

Level 2 : Do queries involving MongoDB update operator.

.

Experiment No. 12:

Level 1 : Perform different query modifiers on Employee Database

Level 2 : Implement different aggregation commands on Employee Information Database.

Experiment No. 13:

Connect a Node.js application to MongoDB using the official MongoDB Node.js driver

Experiment No. 14:

Implement basic server functionalities such as handling HTTP requests and responses. Run and test the Node.js application locally

Experiment No. 15:

Level 1 : Learn about the basics of npm commands such as npm init, npm install, and npm publish.

Level 2 : Explore the npm registry to search for and install existing Node.js modules.

Experiment No. 16:

Create a simple Node.js project and install external modules using npm.

Experiment No. 17:

Level 1 : Create React Application implements input box for a floating number input.

Level 2 : Create a simple ReactJS Application to Pass Data from One Component to Another Component in.

Experiment No. 18:

Design responsive components such as navigation menus, cards, and tables using Bootstrap classes and utilities.

Experiment No. 19:

Create a ReactJs application for food delivery website where users can order food from a particular restaurant listed in the website.

Experiment No. 20:

Create a web application to manage the TO-DO list of users, where users can login and manage their to-do items using MERN stack (MongoDB, ReactJS, NodeJS).

Textbook(s):

T1. Vasan Subramanian, ‘Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React,
and Node’, Second Edition, Apress, 2020.

T2 Brad Dayley, Brendan Dayley, Caleb Dayley, ‘Node.js, MongoDB and Angular Web Development’, Addison-Wesley, Second Edition, 2021.

References/Manual/Software:

1. https://www.tutorialspoint.com/the_full_stack_web_development/index.asp
2. https://www.coursera.org/specializations/full-stack-react
3. https://www.udemy.com/course/the-full-stack-web-development/
4. **Software:** MongoDB, Express.js, React JS, Node.js, Visual Studio Code, Sublime Text, Atom, Git and GitHub.

Course Code: CSA4605	**Course Title: Mobile Application Development using Flutter** **Type of Course: Program Core** **Theory and Laboratory Integrated**	**L-T-P-C**	1	0	4	3
Version No.						
Course Pre-requisites	Python, Java, or C++, HTML, CSS, and JavaScript, Knowledge of SQL and NoSQL databases					
Anti-requisites	NIL					
Course Description	This course introduces students to Mobile Application development using Flutter, a modern cross-platform framework. It covers Dart programming, UI/UX design, state management, backend integration with Firebase, and deployment. Students will develop interactive mobile applications through hands-on lab sessions. The course emphasizes real-world app development, API integration, and best practices for performance optimization.					
Course Objectives	This course equips students with the skills to develop cross-platform mobile applications using Flutter and Dart. It covers UI/UX design, state management, backend integration with Firebase, and API communication. Students will gain hands-on experience in building, debugging, and optimizing mobile apps. The course also focuses on app deployment to the Google Play Store and Apple App Store.					
Course Out Comes	On successful completion of the course the students shall be able to: CO1: Apply the fundamentals of Flutter and Dart [Apply] CO2: Design and develop user-friendly mobile applications. [Create] CO3: Integrate backend services and manage application state effectively. [Apply]					
Course Outcome	Students will be able to **design and develop cross-platform mobile applications** using Flutter and Dart. They will effectively integrate **backend services, manage state, and implement APIs** to create dynamic applications. Additionally, students will **optimize, debug, and deploy** mobile apps on platforms like the Google Play Store and Apple App Store.					
Course Content:						
Module 1	**Introduction to Mobile App Development & Dart Programming**	Assignment	Programming	**25hours [5 T + 20P]**		

<table>
<tr><td colspan="5">Evolution of Mobile Application Development, Native vs. Hybrid vs. Cross-platform Development, Overview of Flutter and its Advantages, Setting up Flutter Development Environment, Understanding Dart: Variables, Data Types, Operators, Control Flow: Loops & Conditionals, Functions and Exception Handling, Object-Oriented Programming in Dart</td></tr>
<tr><td>**Module 2**</td><td>**Flutter Basics & UI Development**</td><td>Assignment</td><td>Programming</td><td>**25hours [5 T + 20P]**</td></tr>
<tr><td colspan="5">Flutter Architecture & Widget Tree, Understanding Stateful & Stateless Widgets, Commonly Used Widgets:Text, Image, Button, GridView, AppBar, Bottom Navigation Bar, Layouts & Styling: Column, Row, Stack, Container, Padding, Margin, Alignments, Themes & Custom Styling, User Input & Forms Handling</td></tr>
<tr><td>**Module 3**</td><td>**State Management & Firebase Integration**</td><td>Assignment</td><td>Programming</td><td>**25hours [5 T + 20P]**</td></tr>
<tr><td colspan="5">State Management Approaches:setState, Provider, Riverpod, Bloc, Working with API & JSON Data, Database Integration:Firebase Authentication (Google, Email-Password, OTP), Firebase Firestore for Real-time Database, Local Database (SQLite, Hive), Notifications: Push & Local, Advanced Flutter & Deployment, Animations & Custom UI, Gesture Detection & Touch Interactions, Maps & Location Services, Background Services & App Lifecycle.</td></tr>
<tr><td colspan="5">**List of Laboratory Tasks:**
Lab sheet -1
Level 1: Setup Flutter Development Environment
• Install Flutter SDK, Dart, and Android Studio
• Configure Emulator & Test a Sample App
Level 2: Dart Basics - Variables & Data Types
• **Create a Dart program to demonstrate different data types**
Lab sheet -2
Level 1: Dart Control Flow - Loops & Conditionals
• Implement a program that checks for prime numbers
Level 2: Functions & Exception Handling in Dart
• **Write a function to calculate factorial with error handling**
Lab sheet -3
Level 1: OOP in Dart - Classes & Objects
• Create a class for a Student Management System
Level 2: Understanding Asynchronous Programming
• **Implement async/await and Future in Dart**
Lab sheet -4
Level 1: Basic Flutter App - Hello World
• Create a simple Flutter app displaying text and images
Level 2: Using Stateless and Stateful Widgets</td></tr>
</table>

- **Develop an app that switches between themes dynamically**

Lab sheet -5

Level 1: Handling User Input & Forms

- Create a login form with validation (TextFormField, TextEditingController)

Level 2: ListView & GridView Implementation

- Build a Contacts List app using ListView.builder()

Lab sheet -6

Level 1: Navigation & Routing

- Implement multi-screen navigation using Navigator.push() and Navigator.pop()

Level 2: Styling & Theming

- Customize an app's appearance using ThemeData and Colors

Lab sheet -7

Level 1 Bottom Navigation & Drawer Implementation

- **Create an app with BottomNavigationBar and Drawer**

Level 2 Gesture Detection

- **Implement swipe detection and tap gestures on widgets**

Lab sheet – 8

Level 1: Basic Animation in Flutter

- **Create a fade-in effect using AnimatedContainer**

Level 2: Understanding State Management (setState)

- **Create a counter app using setState()**

Lab sheet – 9

Level 1: Provider State Management

- **Develop a shopping cart app using Provider**

Level 2: Fetching Data from REST API

- Display weather data using API integration and HTTP package

Lab sheet – 10

Level 1: **Working with Firebase Authentication**

- Implement Google Sign-In authentication in a Flutter app

Lab sheet – 11

Level 1: **Using Firebase Firestore (Database)**

- Create a CRUD app (Add, Update, Delete, Retrieve) using Firestore

Lab sheet – 12

Level 1: **Local Storage with Hive/Shared Preferences**

- Save user preferences (theme mode, language)

Lab sheet – 13

Level 1: **Push Notifications using Firebase Cloud Messaging (FCM)**

- Implement push notifications in a Flutter app

Lab sheet – 14

Level 1: **Image Upload & Storage in Firebase**

- Capture and upload images to Firebase Storage

Lab sheet – 15

Level 1: **Final Mini Project**

- Develop a complete app integrating Firebase, UI, and State Management.

Targeted Application & Tools that can be used:

- Android Studio (Recommended)
- Visual Studio Code (VS Code)
- Xcode (For iOS Development - macOS Only)

Project work/Assignment: Mention the Type of Project /Assignment proposed for this course The course includes **hands-on projects and assignments** focusing on real-world mobile application development. Students will work on: **o-Do List, Weather App, or Expense Tracker** to reinforce Flutter concepts. Fetch and display data from **REST APIs** (e.g., News, Weather, or Movie API). Implement **user authentication, real-time database, and cloud storage** using Firebase. Students will generate APK files and optionally **publish their apps** on the Google Play Store.

After completion of each module a programming-based Assignment/Assessment will be conducted. On completion of Module 2, student will be asked to develop a Project.

Textbook(s):

T1. **Marco L. Napoli**, *Beginning Flutter: A Hands-On Guide to App Development*, O'Reilly Media, 2019.

- Covers Flutter basics, UI design, state management, and app deployment.

T2. Alessandro Biessek, *Flutter for Beginners: An introductory guide to building cross-platform mobile applications with Flutter and Dart 2*, Packt Publishing, 2020.

- Provides step-by-step guidance for Flutter development with practical examples.

T3. Rap Payne, *Flutter & Dart Cookbook: Practical recipes for building cross-platform mobile apps*, Packt Publishing, 2022.

- Offers solutions for real-world Flutter app development challenges.

References

R1. Thomas Bailey, *Flutter for Dummies*, Wiley, 2020.

- A beginner-friendly book covering Flutter concepts, UI components, and app deployment.

R2. Carmine Zaccagnino, *Flutter Apprentice: Beyond the Basics*, Razeware LLC, 2022.

- Explores advanced Flutter concepts, including animations, custom widgets, and performance optimization.

R3. Martin Aguinis & Google Flutter Team, *Flutter Complete Reference: Create Modern Cross-Platform Apps*, BPB Publications, 2021.

- Comprehensive guide covering Flutter's framework, UI design, Firebase integration, and real-world projects.

Web Based Resources and E-books:

Digital Learning Resources (Library Resources)

W1: UDEMY : https://www.udemy.com/topic/google-flutter/?srsltid=AfmBOoqb5qprpD6TRaWLEH3iSSAIOEya6k4qOsy_ty4gHxJZamaCKOqm&utm_source=chatgpt.com

W2: https://flutter.dev/learn?utm_source=chatgpt.com

Topics relevant to "SKILL DEVELOPMENT": Flutter Framework & Dart Programming, UI/UX Design & Responsive Layouts, State Management Techniques, API Integration & Backend Connectivity.

Course Code: CSA8100	Course Title: Mini Project Type of Course: NTCC	L- T-P- C	-	-	-	3
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 method of scientific experimentation, and often get an opportunity to see, study and operate sophisticated and costly equipment. They also learn about the implementation of the principles of management they have learnt in class, when they observe multidisciplinary teams of experts from engineering, science, economics, operations research, and management deal with techno-economic problems at the micro and macro levels. Finally, it enables them to develop and refine their language, communication and inter-personal skills, both by its very nature, and by the various evaluation components, such as seminar, group discussion, project report preparation, etc. The broad-based core education, strong in mathematics and science and rich in analytical tools, provides the foundation necessary for the student to understand properly the nature of real-life problems. The students have options to pursue this course as either Project Work and Dissertation at the university, or Project Work in an Industry/ Company/ Research Laboratory, or Internship Program in an Industry/Company.					
Course Objectives	The objective of the course is to familiarize the learners with the concepts of Professional Practice and attain **Employability Skills** through **Experiential Learning** techniques.					
Course Outcomes	On successful completion of this course the students shall be able to: 1. Identify real world computing problems related to local, regional, national or global needs. 2. Apply appropriate techniques or modern tools for solving the intended problem. 3. Design the experiments as per the standards and specifications. 4. Interpret the events and results for meaningful conclusions. 5. Appraise project findings and communicate effectively through scholarly publications.					

COURSE BROAD SCHEDULE			
Topics	Mark		Rubrics
1. Domain, Technology, Project Scope and Planning	10	5%	Domain and technology (5) Scope and planning (5)
2. Importance and relevance of working technology in solving the problem	10	5%	Importance and relevance of working technology (8) and Identifying the applications (2)
3. Contributions to the project in solving real-world problems.	10	5%	Feature/characteristics/Applications selection and Constraints Identification (5), Analysis and Feature finalization subject to constraints and Design(5).
4. Application or usage of the project and impact on the society	20	10%	Application or usage of the project (10) and impact on the society (10) (SDG)
5. End Term Viva	100	50%	Technical Competency, Scope fulfilment, originality of problem solving, analysis and discussion of results, presentation content (delivery, clarity, supporting material quality), cost and impact analysis.
6. Project Report	10	5%	Compliance of report as per format (10)
7. *Supervisor	10	5%	Individual, Team work, discipline, Project Progression (10)
8. Publication/Patent	20	10%	Research Publication (15) and Patenting the project work (5)
9. Git Hub Repository	10	5%	Uploading the complete coding, Reports, publication proof, patent proof and other necessary documents in Git Hub(10)

Total	200	100%	** For S.No 1-4 , 5 marks should be allotted for communication and documentation skills*
Catalogue preparedby	Dr M. Anand Kumar		
Recommended b ythe Board of Studies on	BoS No: 16th, held on 27.03/2024		
Date of Approval bythe Academic Council	23rd Academic Council Meeting held on 27.03/2024		

Course Code: CSA8300	**Course Title: Major Project** **Type of Course: LTPC**	**L- T-P- C**	-	-	-	12
Version No.	**1.0**					
Course Pre-requisites	Knowledge and skills gained from all courses studied in previous semesters, including those applied in the mini-project.					
Anti-requisites	**NIL**					
Course Description	Students observe science and technology in action, develop an awareness of the method of scientific experimentation, and often get an opportunity to see, study and operate sophisticated and costly equipment. They also learn about the implementation of the principles of management they have learnt in class, when they observe multidisciplinary teams of experts from engineering, science, economics, operations research, and management deal with techno-economic problems at the micro and macro levels. Finally, it enables them to develop and refine their language, communication and inter-personal skills, both by its very nature, and by the various evaluation components, such as seminar, group discussion, project report preparation, etc. The broad-based core education, strong in mathematics and science and rich in analytical tools, provides the foundation necessary for the student to understand properly the nature of real-life problems. The students have options to pursue this course as either Project Work and Dissertation at the university, or Project Work in					

	an Industry/ Company/ Research Laboratory, or Internship Program in an Industry/Company.
Course Objectives	The objective of the course is to familiarize the learners with the concepts of Professional Practice and attain **Employability Skills** through **Experiential Learning** techniques.
Course Outcomes	On successful completion of this course the students shall be able to: 1. Identify real world computing problems related to local, regional, national or global needs. [**Understand]** **2.** Apply appropriate techniques or modern tools for solving the intended problem**. [Apply]** **3.** Design the experiments as per the standards and specifications. **[Apply]** 4. Interpret the events and results for meaningful conclusions. [**Analyze]** 5. Appraise project findings and communicate effectively through scholarly publications. [**Evaluate]**

COURSE BROAD SCHEDULE			
Topics	Mark		Rubrics
1. Domain, Technology, Project Scope and Planning	10	5%	Domain and technology (5) Scope and planning (5)
2. Importance and relevance of working technology in solving the problem	10	5%	Importance and relevance of working technology (8) and Identifying the applications (2)
3. Contributions to the project in solving real-world problems.	10	5%	Feature/characteristics/Applications selection and Constraints Identification (5), Analysis and Feature finalization subject to constraints and Design(5).
4. Application or usage of	20	10%	Application or usage of the project (10)

the project and impact on the society			and impact on the society (10) (SDG)
5. End Term Viva	100	50%	Technical Competency, Scope fulfilment, originality of problem solving, analysis and discussion of results, presentation content (delivery, clarity, supporting material quality), cost and impact analysis.
6. Project Report	10	5%	Compliance of report as per format (10)
7. *Supervisor	10	5%	Individual, Team work, discipline, Project Progression (10)
8. Publication/Patent	20	10%	Research Publication (15) and Patenting the project work (5)
9. Git Hub Repository	10	5%	Uploading the complete coding, Reports, publication proof, patent proof and other necessary documents in Git Hub(10)
Total	200	100%	** For S.No 1-4 , 5 marks should be allotted for communication and documentation skills*

Discipline Electives

Course Code: **CSA4701**	**Course Title: Agile Methodology and DevOps** **Type of Course: Discipline Elective**	**L-T- P- C**	3	0	0	3
Version No.	1.0					

Course Pre-requisites	**Adaptive Software Engineering**			
Anti-requisites	**NIL**			
Course Description	This course offers a solid foundation in Agile methodologies and DevOps practices, focusing on modern software development. It covers Agile principles, estimation techniques, and frameworks like Scrum, XP, Unified Process, and EVO. Students will learn DevOps concepts, lifecycle, and key tools, along with hands-on experience using Git for version control. Through assignments, case studies, and practical sessions, the course equips students to apply agile thinking and DevOps automation for efficient, high-quality software delivery.			
Course Objective	**The objective of the course is EMPLOYBILITY of student by using PARTICIPATIVE LEARNING techniques.**			
Course Outcomes	**On successful completion of this course, the students shall be able to:** **CO1**: Apply the fundamental concepts of Agile methodology, including Agile values, principles, and estimation techniques, to distinguish between Agile and traditional software development approaches. [**Apply**] **CO2:** Analyze the significance and evolution of Agile methodologies such as Scrum, Extreme Programming, EVO, and the Unified Process, and their impact on modern software development practices. [**Analyze**] **CO3**: Analyze the integration of DevOps principles and tools in software development life cycles, highlighting the differences between Waterfall, Agile, and Lean methodologies. [**Analyze**] **CO4:** Implement version control using Git by performing essential Git operations including repository management, staging, committing, and collaborating with remote repositories. [**Apply**]			
Course Content:				
Module 1	**Foundations of Agile Methodology and Estimation Techniques - CO1**	**Assignment**	**Theory Heading**	**10 Sessions**
Introduction to Agile technology, Iterative and Evolutionary Methods, Agile – Agile Development. Agile Values, Agile manifesto and principles – Agile project management – Agile team interactions, Compare and Contrast the agile with traditional methods. Agile Benefits. Agile Estimation Techniques.				
Module 2	**Agile Frameworks and Their Industry Significance – CO2**	**Assignment**	**Problem Solving**	**10 Sessions**
Agile Story: Evolutionary delivery, Scrum Demo, Planning game, Sprint back log, adaptive planning. Agile Motivation – Problems with The Waterfall - Research Evidence. Scrum: Method Overview, Life				

cycle phases and Work product roles and practices. Extreme Programming: Method Overview, Life cycle phases and Work product roles and practices. Unified process: Method Overview, Life cycle phases and Work product roles and practices. EVO: Method Overview, Life cycle phases and Work product roles and practices.

Module 3	**Introduction to DevOps and Software Development Lifecycle – CO3**	**Assignment**	**Case Study**	**12 Sessions**

Basic Linux Commands, Software Development Lifecycle, Waterfall Model, Agile Model, Lean Methodology, Waterfall Vs Agile Vs Lean, Overview of DevOps – Core elements of DevOps – Life cycle of DevOps –Adoption of DevOps -DevOps Tools – Build, Promotion and Deployment in DevOps.

Module 4	**Version Control and Git Operations in DevOps – CO4**	**Assignment**	**Programming**	**13 Sessions**

Devops and its tools. Version Control with Git, Introduction to Git, Features of Git, Benefits, Workflow, Git vs GitHub, Installation of Git on Windows/Linux and Environment set up, All Git Commands-Working with local and remote repositories, Running first Git command, Fundamentals of Repository structure and file status life cycle, Working locally with staging, unstaging and commit.

Project work/Assignment:

7. Compare Agile with Traditional Software Development Models.
8. Map the DevOps Lifecycle with Appropriate Tools.
9. Demonstrate the Use of Git for Version Control.
10. Project: AI-Powered Resume Analyzer
11. Project: DevOps Pipeline Simulation
12. Project: Smart Attendance System with Face Recognition

Topics related to

1. Problem Solving: Choose efficient algorithmic or system-level solutions for real-world challenges such as traffic management, scalable data processing, or sustainable computing.

2. Employability: Simulation of industry-relevant environments such as real-time network traffic, cloud deployment, agile development lifecycle, and cybersecurity scenarios to enhance practical skills and job readiness.

Textbook(s):

T1. Craig Larman, “Agile and Iterative Development – A Manager’s Guide”, Pearson Education – 2006.

T2. Edward Scatter “Brilliant Agile Project Management: A Practical Guide to Using Agile, Scrum and Kanban, 2015.

T3. Craig Berg, “DevOps For Beginners: A Complete Guide to DevOps Best Practices (Including How You Can Create World-Class Agility, Reliability, And Security In Technology Organizations With DevOps) (Code tutorials)”, Paperback – June 12, 2020.

References

R1. Chetankumar Patel, Muthu Ramachandran, Story Card Maturity Model (SMM): A Process Improvement Framework for Agile Requirements Engineering Practices, Journal of Software, Academy Publishers, Vol 4, No 5 (2009), 422-435, Jul 2009.

R2. Hazza& Dubinsky, Agile Software Engineering, Series: Undergraduate Topics in Computer Science, Springer 2009.

R3. Gaurav Agarwal, "Modern DevOps Practices: Implement and secure DevOps in the public cloud with cutting-edge tools, tips, tricks, and techniques", July 2021.

R4. Mikael Krief, "*Learning DevOps: The complete guide to accelerate collaboration with Jenkins, Kubernetes, Terraform and Azure DevOps*", October 2019.

E-Resources:

4. https://www.agilealliance.org/agile101
5. https://www.atlassian.com/agile
6. https://learn.microsoft.com/en-us/devops
7. https://www.atlassian.com/git/tutorials/atlassian-git-cheatsheet.

Course Code: **CSA4702**	**Course Title: Full Stack Development** **Type of Course: Discipline Elective**	**L-T- P- C**	1	0	4	3
Version No.	1.0					
Course Pre-requisites	**Web Technology, DBMS**					
Anti-requisites	**NIL**					
Course Description	This course provides a comprehensive understanding of **Java enterprise development**, covering both foundational and advanced Java programming concepts. Learners will gain hands-on experience with **Java I/O**, **Generics**, **Lambdas**, **Servlets**, **JSP**, and **JPA with Hibernate** for building robust web applications. The course delves into **Spring Core**, **Spring MVC**, and **Spring Boot REST APIs**, enabling rapid enterprise-level development. It also introduces **automation tools** such as **Apache Maven** and **Selenium**, equipping students to manage projects efficiently and perform automated web testing. A blend of theory and intensive programming practice ensures industry-readiness in full-stack Java application development.					
Course Objective	**The objective of the course is EMPLOYBILITY of student by using PARTICIPATIVE LEARNING techniques.**					
Course Outcomes	**On successful completion of this course, the students shall be able to:** **CO1**: Apply Java programming concepts including I/O streams, collections, generics, and lambda expressions to build functional components in enterprise applications. **(Analyze)** **CO2:** Analyze the behavior of Java EE web applications using servlets, JSP, and MVC architecture to manage client-server interactions effectively. **(Apply)** **CO3**: Apply JPA and Hibernate ORM features to develop scalable and maintainable data persistence layers in enterprise solutions. **(Apply)** **CO4:** Analyze the components of the Spring framework to develop secure, RESTful web services and enterprise applications. **(Analyze)** **CO5:** Apply automation tools like Maven and Selenium WebDriver to automate the build, test, and deployment process in real-time projects. **(Apply)**					
Course Content:						
Module 1	**Advanced Java Programming and Core Concepts – CO1**	Experimental Learning	Programming	**15 Sessions (3T + 12P)**		
Review of Java; Java I/O; Advanced concepts of Java and Java New Features of Java; Collection framework, Annotation, Java generics, Lambda Expression, JDBC.						

Module 2	**Java EE Web Development with Servlets and JSP – CO2**	Experimental Learning	Programming	**15 Sessions (3T + 12P)**
Introduction to Eclipse & Tomcat; Servlet API Fundamentals; Servlet Context, Session, Cookies; Request Redirection Techniques; JSP Fundamentals; Reading HTML form Data with JSP; State Management with Java; JSP Standard Tag Library - Core & Function Tags; Building MVC App with Servlets & JSP; Complete App- Integrating JDBC with MVC App				
Module 3	**Data Persistence with JPA and Hibernate ORM – CO3**	Experimental Learning	Programming	**15 Sessions (3T + 12P)**
Fundamentals of Java Persistence with Hibernate: JPA for Object/Relational Mapping, querying database using JPQL and Criteria API (JPA). Hibernate: Architecture, HQL, Querying, Caching, Performance and Concurrency; First & Second Level Caching, Batch Fetching, Optimistic Locking & Versioning; Entity Relationships, Inheritance Mapping & Polymorphic Queries.				
Module 4	**Spring Framework and RESTful Web Services – CO4**	Experimental Learning	Programming	**15 Sessions (3T + 12P)**
Spring Core, Spring MVC, Spring Boot REST API: Understanding Spring Framework; Spring AOP (Aspect Oriented Programming); Using Spring MVC; Building a Database Web App with Spring and Hibernate Implementing Spring Security; Developing Spring REST API; Using Spring Boot for Rapid Development				
Module 5	**Automation Testing with Maven and Selenium – CO5**	Experimental Learning	Programming	**15 Sessions (3T + 12P)**
Introduction to Automation Tools; Apache Maven: Maven Fundamentals, Software Setup – Command lineand 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 Web Elements, Driver Commands, Web Element Command				

List of Laboratory Tasks:

Experiment No. 1:

Level 1: Use Serialization and Deserialization mechanism to develop a console application.

Level 2: Build a console application by using Collection framework and Annotation.

Experiment No. 2:

Level 1: Build a console application by using Collection framework and Lambda Expression

Level 2: Develop a console application that connect with MySQL Database and perform database transactions.

Experiment No. 3:

Level 1: Build a web application to connect with a database using Servlet that perform database manipulations.

Level 2: Build web applications to connect with a database using JSP that perform database manipulations.

Experiment No. 4:

Level 1: Construct a login application in respect of the MVC model.

Level 2: Implement a web application based on the MVC design pattern, to create an Employee Registration module using JSP, Servlet, JDBC, and MySQL database.

Experiment No. 5:

Level 1: Create Student mark processing project using Hibernate with Maven.

Level 2: Create a User Registration project using JSP, Servlet, Hibernate Framework, and MySQL database.

Experiment No. 6:

Level 1: Develop a User Login Form and will validate username and password with the MySQL database using the Hibernate framework.

Level 2: Build a complete Hibernate application with HQL CRUD operations using MAVEN, JSP, Servlet, Hibernate Framework, JPQL and MySQL database.

Experiment No. 7:

Level 1: Build CRUD RESTful API using Spring Boot 3, Spring Data JPA (Hibernate), and MySQL database.

Level 2: Build login or sign-in and registration or signup REST API using Spring boot, Spring Security, Hibernate, and MySQL database.

Experiment No. 8:

Level 1: Create Spring web application to implement SpringMVC framework using eclipse IDELanguage identifier

Targeted Application & Tools that can be used:

Java SE & EE, Eclipse, Tomcat, JSP, Servlets, JDBC, JPA, Hibernate, Spring Boot, Spring MVC, Spring Security, Apache Maven, Selenium WebDriver, MySQL.

Project work/Assignment:

1. Build a console-based Library Management System using Java collections, generics, and I/O.
2. Develop a Student Registration Portal using Servlets and JSP.
3. Create an Employee Management System using JPA and Hibernate.
4. Build a Spring Boot-based Inventory Management System with RESTful API support.
5. Create a Test Automation Framework using Selenium WebDriver and Apache Maven.

Topics related to

1. Problem Solving: Develop a full-stack Java web application integrating JSP, Servlets, JDBC, and Spring to address real-world business challenges such as inventory management or customer tracking.

2. Employability: Simulate industry-level scenarios using tools like Apache Maven and Selenium for automated testing, and apply Spring Boot REST APIs to build scalable web services, enhancing job-readiness in enterprise Java development.

Textbook(s):

T1. Fender, Young, “Front-end Fundamentals”, Leanpub, 2021.

T2. Horstmann, “Core Java Volume II - Advanced Features”, 12th Edition – Pearson, 2023

References

R1. Soni, Ravi Kant. “Full Stack AngularJS for Java Developers: Build a Full-Featured Web Application fromScratch Using AngularJS with Spring RESTful.” ,Apress, 2021.

R2. Mardan, Azat. “Full Stack JavaScript: Learn Backbone.js, Node.js and MongoDB.”,Apress, 2015

Course Code: CSA4703	Course Title: Responsive Web Designing Type of Course: Discipline Elective	L-T- P- C	3	0	0	3
Version No.	1.0					
Course Pre-requisites	Web Technology					
Anti-requisites	NIL					
Course Description	This course introduces the fundamentals of modern web development, focusing on building structured, responsive, and interactive websites. Students will learn HTML for content structure, CSS for styling and layout, and JavaScript for dynamic behavior. The course also covers Responsive Web Design using tools like Sass, Flexbox, and CSS Grids for mobile-friendly interfaces. Finally, students explore Web Project Management, including planning, budgeting, development, and deployment processes. Practical assignments and projects enhance hands-on skills for real-world web development.					
Course Objective	The objective of the course is EMPLOYBILITY of student by using PARTICIPATIVE LEARNING techniques.					
Course Outcomes	**On successful completion of this course, the students shall be able to:** **CO1**: Design structured web pages using fundamental HTML and HTML5 elements including text, links, images, tables, and forms. [**Apply**] **CO2:** Apply CSS and JavaScript to style web pages and add interactive client-side functionalities. [**Apply**] **CO3**: Develop responsive websites using modern techniques like Flexbox, CSS Grid, Sass, and mobile-first design principles. [**Apply**] **CO4:** Demonstrate an understanding of the web project lifecycle, including planning, execution, documentation, testing, and deployment. [**Apply**]					
Course Content:						
Module 1	**Foundations of Web Design with HTML - CO1**	**Assignment**	**Programming**	**10 Sessions**		
Working of Web - HTML Markup for Structure - Creating simple page - Marking up text - Adding Links - Adding Images - Table Markup - Forms - HTML5.						
Module 2	**Styling and Scripting for Dynamic Web Pages – CO2**	**Assignment**	**Programming**	**10 Sessions**		
CSS - Formatting text - Colours and Background - Padding, Borders and Margins - Floating and positioning - Page Layout with CSS - Transition, Transforms and Animation - Javascript - Using Java Script**.**						

<table>
<tr><td><b>Module 3</b></td><td><b>Techniques for Responsive Web Design – CO3</b></td><td><b>Assignment</b></td><td><b>Programming</b></td><td><b>15 Sessions</b></td></tr>
<tr><td colspan="5">Sass for Responsive Web Design - Marking Content with HTML5 - Mobile-First or Desktop-First - CSS Grids, CSS Frameworks, UI Kits, and Flexbox for RWD - Designing small UIs by Large Finger - Images and Videos in Responsive Web Design - Meaningful Typography for Responsive Web Design.</td></tr>
<tr><td><b>Module 4</b></td><td><b>Managing Web Development Projects – CO4</b></td><td><b>Assignment</b></td><td><b>Programming</b></td><td><b>10 Sessions</b></td></tr>
<tr><td colspan="5">Project Life Cycle - Project Definition - Discovery and Requirements - Project Schedule and Budgeting - Running the project - Technical Documentation - Development, Communication, Documentation - QA and testing -Deployment - Support and operations.</td></tr>
</table>

Project work/Assignment:

1) Analyze an existing website’s HTML layout. Identify and document usage of semantic tags, forms, tables, and media embedding.
2) Compare different CSS layout models (float, flex, grid). Write a short report with code snippets showing use cases for each.
3) Draft a basic web project plan including objectives, timeline, technology stack, team roles, testing strategies, and deployment steps.
4) Project: Personal Portfolio Website
5) Project: Responsive Business Landing Page
6) Project: Mini E-commerce Product Page

Topics related to

1. Problem Solving: Choose *Responsive Web Design Techniques to Optimize User Experience Across Devices*.

2. Employability: Simulation of *Real-Time Web Page Development using HTML, CSS, and JavaScript Frameworks*.

Textbook(s):

T1. Jennifer Niederst Robbins, "Learning Web Design", O'REILLY 5th Edition, 2020.

T2. Ricardo Zea, "Mastering Responsive Web Design", PACKT Publishing, 2021.

T3. Justin Emond, Chris Steins, "Pro Web Project Management", Apress, 2020.

References

R1. Jon Duckett, "HTML and CSS: Design and Build Websites", John Wiley and Sons, edition 2021.

R2. Faithe Wempen, “Step by Step HTML 5”, South Asian Edition, Microsoft Press and PHI Learning.

R3. Adam Freeman Pro jQuery 2.0 Apress Latest Edition.

R4. Jeremy McPeak Beginning JavaScript Wrox Publication Latest Edition.

E-Resources:

1. https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Responsive_Design
2. https://www.w3schools.com/html/html_responsive.asp
3. https://developers.google.com/web/fundamentals/design-and-ux/responsive
4. https://jsfiddle.net/

5. https://www.freecodecamp.org/

Course Code: **CSA4704**	**Course Title: Data Modelling with NoSQL Databases** **Type of Course: 1] Elective_4**	**L-T-P-C**	**2**	**0**	**2**	**3**
Version No.	1.0					
Course Pre-requisites	**NIL**					
Anti-requisites	NIL					
Course Description	This course is designed to provide the core concepts of NoSQL databases. It provides a comprehensive introduction to the NoSQL database to design, create, manage, and query databases using MongoDB and Cassandra with practical hands-on experience, covering topics like database structure, installation, data manipulation, query optimization, security, and administration tools, allowing learners to effectively store and retrieve data within a database environment.					
Course Objective	The objective of the course is SKILL DEVELOPMENT of student by using PARTICIPATIVE LEARNING techniques.					
Course Out Comes	**On successful completion of the course the students shall be able to:** CO1: Understand the different types of NoSQL databases CO2: Understand the impact of the cluster on database design CO3: Apply Nosql development tools on different types of NoSQL Databases. CO4: Apply the different types of indexes in No SQL-DB for query execution.					
Course Content:						

Module 1	Introduction to NoSQL Database Modeling	Participative Learning		Brainstorming session/Quiz	**No. of classes** **L-6 P-8**

<table>
<tr><td colspan="6"><b>Topics:</b> Introduction-Different Databases for Different Requirement-:Relational Database Design-Early Database Management System- The Relational Database Revolution-Variety of NoSQL Database- Data Management with Distributed Database- ACID and BASE-CAP Theorem-SQL versus NoSQL</td></tr>
<tr><td><b>Module 2</b></td><td>NoSQL Database Types</td><td>Participative Learning</td><td></td><td>Coding Assignment</td><td><b>No. of classes L-8 P-6</b></td></tr>
<tr><td colspan="6"><b>Topics:</b> Four Types of NoSQL Databases: Key-Value Pair Databases: From Arrays to Key-Value-Essential Features of Key-Value Database- Hash Functions- Key-Value Dat abase Terminology-TTL Keys-Database-Document Database: Managing Multiple Documents in Collection-Operations-Collection-Embedded Document-Schemaless- Types of Partitions-Column Family Database:Architectures-Terminology-Graph Database:Graph and Network Modelling</td></tr>
<tr><td><b>Module 3</b></td><td>NoSQL Technologies</td><td>Experimental Learning</td><td></td><td>Coding Assignment/Quiz</td><td><b>No. of classes L-8 P-8</b></td></tr>
<tr><td colspan="6"><b>Topics:</b> Introduction to MongoDB-Developing with MongoDB- Data Modeling-Methods-Querying-Replication-CURD Operations-Sharding-Deployment-Application Administration-Server Administration-Map Reduce-CouchDB:Introduction to CouchDB-Curl and Futon-Database Creation-CouchDB HTTP API-Document-Connectivity</td></tr>
<tr><td><b>Module 4</b></td><td>Distributed Databases</td><td>Project based Learning</td><td></td><td>Project</td><td><b>No. of classes L-8 P-8</b></td></tr>
<tr><td colspan="6"><b>Topics:</b>Cassandra:Introduction-Architecture-Data Model-Keyspace Operations-Table Operations-CURD Operations-CQL Types-CQL Collections-CAP Theorem-Data modeling in Cassandra-Cassandra vs Hbase-Hbase:HBase Shell-Data Model-General Commands-Client API-Insert and Retrieve data in Hbase-Neoj Data Model-Overview-Querying Graph Databases-Redis Commands-Keys-Strings</td></tr>
<tr><td colspan="6"><b>List of Laboratory Tasks:</b></td></tr>
<tr><td colspan="6"><b>Experiment No.1: Installation and setup of MongoDB</b>
<b>Level 1</b>: <b>Connect</b> to MongoDB & explore your data
<b>Level 2: Set</b> up MongoDB Compass (GUI tool for MongoDB)</td></tr>
<tr><td colspan="6"><b>Experiment No.2: Basic commands to interact with MongoDB shell (mongo command)</b>
<b>Level 1</b>: <b>Creating</b> and managing databases and collections.
<b>Level 2: Apply</b> Collections, documents, and databases.</td></tr>
<tr><td colspan="6"><b>Experiment No.3:</b> MongoDB Basic Operations
<b>Level 1: Use</b> CRUD Operations (Create, Read, Update, Delete)
<b>Level 2: Create</b> and switch between databases and collections</td></tr>
<tr><td colspan="6"><b>Experiment No.4:Perform Insert and Delete Operations</b>
<b>Level 1: Use</b> insertOne() method for Single Document
<b>Level 2: Use</b> insertMany() method for Single Document</td></tr>
</table>

Level 3: Delete documents using deleteOne() and deleteMany()

Experiment No.5: Update and Retrieve Operations in MongoDB

Level 1: Develop and Retrieve documents using find(), findOne()

Level 2: Update documents using updateOne() and updateMany()

Experiment No.6: MongoDB Querying and Indexing

Level 1: Apply MongoDB Query Language (MQL)

Level 2: Create and manage indexes in MongoDB (createIndex())

Experiment No.7: MongoDB Creating Complex queries using operators

Level 1: Create and execute complex queries using operators

Level 2: Develop the Query operators: $gt, $lt, $in, $ne, $and, $or

Experiment No.8: Create aggregation pipelines to perform multi-step data transformation in MongoDB

Level 1: **Implement** Aggregation pipeline stages: $match, $group, $project, $sort

Level 2: **Filter and sort** data using $match and $sort

Experiment No.9: Advanced Aggregation and Data Transformation in MongoDB

Level 1: Join data from different collections using $lookup

Level 2: Develop complex data transformations

Experiment No.10: MongoDB Transactions and ACID Properties

Level 1: Use ACID properties and their relevance in MongoDB

Experiment No.11: Perform rollback and commit operations within a transaction

Level 1: Understand Rollback and Commit Commands

Experiment No.12: Integration of MongoDB with Applications

Level 1: Demonstrate and build a Python database with MongoDB

Experiment No.13: Demonstrate Text search using catalog data collection for a given word

Level 1:Excluding documents with certain words and phrases

Experiment No.14: Install Cassandra and Data Modeling

Level 1: Download Apache Cassandra and eclipse

Level 2: Create Ecommerce Customers and Products Table

Experiment No.15: Cassandra Query Language(CQL)

Level 1: Understand and practice statements INSERT, SELECT, UPDATE and DELETE

Targeted Application & Tools that can be used

-Install MongoDB Community Edition on windows and Apache Cassandra

-Verison: MongoDB 4.4 or higher (64-bit only).

Note:MongoDB does not support 32-bit x86 platforms.

Mini-Project work

Mini-Project Titles:

1. Blogging Platform with MongoDB-CRUD operations for blog posts.

2. Product Catalog Management System-Creating flexible schemas to accommodate varied product attributes.Implementing advanced querying and filtering in MongoDB
3. Inventory Management System-Track stock levels dynamically with real-time updates.Add, update, or delete items seamlessly.
4. Full-stack application using MongoDB as a database
5. Use advanced features like aggregation, transactions, and sharding
6. Present the project, showcasing MongoDB's features in a real-world scenario
7. To-Do List Application-Designing simple and effective data schemas in MongoDB.
8. Simple Social Media Application-Create a platform where users can connect, share posts, and engage with content. Imagine a lightweight version of Facebook or Instagram, customized to your learning needs.
9. Contact Management System-Build a database-driven app to store, update, and organize contact details.
10. Basic File Sharing Application-File storage and retrieval.
11. Task Tracker for Personal Projects-Dynamic data updates, ensuring every task reflects its status in real-time.
12. Bookstore Management System-To manage books, inventory, and sales, replicating the functionality of a small-scale bookstore.
13. Cassandra for Real-Time Analytics
14. Cassandra and Graph Databases: Exploring Data Relationships

TEXT BOOKS:

T1: Shannon Bradshaw,Eoin Brazil,MongoDB: The Definitive Guide - Powerful and Scalable Data Storage, Third Edition,2020

T2:Greyson Chesterfield, Mastering NoSQL Databases with MongoDB: Building Flexible, Scalable, and High-Performance Databases,November 2024

T3:Dan Sullivan,NoSQL For Mere Mortals, Pearson Education,2015 Shannon Bradshaw, Eoin Brazil, Kristina Chodorow,MongoDB: The Definitive Guide, 3rd Edition,O'Reilly Media,2019

T4: Shannon Bradshaw, Eoin Brazil, Kristina Chodorow,MongoDB: The Definitive Guide, 3rd Edition,O'Reilly Media,2019

REFERENCES:

R1. Eric Redmond and Jim R,Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement,O'Reilly ,2012

R2. Jeff Carpenter,Eben Hewitt,Cassandra: The Definitive Guide, 3rd Edition,April 2020, O'Reilly Media, Inc.,ISBN: 9781098115166

R2. Aaron Ploetz, Devram Kandhare, Sudarshan Kadambi, Xun Wu,Seven NoSQL Databases in a Week, March 2018,Packt Publishing,ISBN: 9781787288867

MOOC's/Swayam Courses/Online Courses:

https://onlinecourses.nptel.ac.in/noc22_cs91/preview
https://onlinecourses.swayam2.ac.in/cec25_ma16/preview

Certification Course by Google :

1. https://learn.mongodb.com/pages/certification-program
2. https://www.mongodb.com/resources/services/mongodb-professional-certification
3. https://www.simplilearn.com/free-mongodb-course-skillup
4. https://www.coursera.org/courses?query=mongodb
5. https://www.mygreatlearning.com/academy/learn-for-free/courses/mongodb-tutorial
6. https://www.coursera.org/learn/introduction-to-nosql-databases
7. https://www.simplilearn.com/free-nosql-course-skillup

ONLINE RESOURCES:

W1. https://datubaze.wordpress.com/wp-content/uploads/2021/03/nosql-for-mere-mortals.pdf
W2: https://repositorio-aberto.up.pt/bitstream/10216/61586/1/000148158.pdf
W3: https://dl.faghatketab.ir/Books/Computer/Database/NoSQL-for-dummies.pdf
W4: https://www.geeksforgeeks.org/mongodb-tutorial/

Topics relevant to "EMPLOYABILITY SKILLS": Topics of all four modules will help in developing **Employability Skills** through **Experiential Learning techniques**. This is attained through assessment component mentioned in course handout.

<table>
<tr><td><b>Course Code:</b>
<b>CSA4705</b></td><td colspan="3"><b>Course Title: Backend Development with Node.js</b>
<b>Type of Course: Discipline Elective</b></td><td><b>L-T- P- C</b></td><td>2</td><td>0</td><td>2</td><td>3</td></tr>
<tr><td><b>Version No.</b></td><td colspan="8">1.0</td></tr>
<tr><td><b>Course Pre-requisites</b></td><td colspan="8"><b>Web Technology</b></td></tr>
<tr><td><b>Anti-requisites</b></td><td colspan="8"><b>NIL</b></td></tr>
<tr><td><b>Course Description</b></td><td colspan="8">It provides students with a comprehensive understanding of Server-side Scripting backend development, how to build efficient and scalable backend applications using Node.js. Starting from the basics and moving towards advanced concepts, this course will equip you with the skills necessary to develop high-performance server-side applications. This course incorporates modern technologies and practices such as Node.js, RESTful APIs. By studying Server-side Scripting, students become familiar with these technologies, preparing them for the demands of the tech industry.</td></tr>
<tr><td><b>Course Objective</b></td><td colspan="8">The objective of the course is EMPLOYBILITY of student by using PARTICIPATIVE LEARNING techniques.</td></tr>
<tr><td><b>Course Outcomes</b></td><td colspan="8"><b>On successful completion of this course, the students shall be able to:</b>
<b>CO1</b>: Apply server-side JavaScript in web application development (Apply)
<b>CO2:</b> Analyze the web application development using HTTP, FS and Buffer modules (Analyze)
<b>CO3</b>: Develop dynamic websites [Apply]
<b>CO4:</b> Apply the node express, JSON, Socket.IO to allow high scalability with asynchronous code [Apply]</td></tr>
<tr><td><b>Course Content:</b></td><td colspan="8"></td></tr>
<tr><td><b>Module 1</b></td><td><b>Getting Started with Node.JS-CO1</b></td><td>Assignment</td><td colspan="2">Programming</td><td colspan="4"><b>14 Sessions (L6 + P8)</b></td></tr>
<tr><td colspan="9">Introducing Node.JS, Node Pacakage Manager (npm), Custom NPM modules, Installing Node, use Node.js REPL, Explore and use built-in modules of Node.js, Use of Node.JS and GitHub, collaborate on code with others using the git tool.</td></tr>
<tr><td><b>Module 2</b></td><td><b>Node JS Modules – CO2</b></td><td>Assignment</td><td colspan="2">Programming</td><td colspan="4"><b>14 Sessions (L8 + P6)</b></td></tr>
<tr><td colspan="9">Working with Node packages – Using Node package manager – Creating a simple Node.js application – Using Events – Listeners –Timers - Callbacks – Handling Data I/O – Implementing HTTP services in Node.js.</td></tr>
</table>

<table>
<tr><td>Module 3</td><td>Handling Data I/O in Node.js – CO3</td><td>Assignment</td><td>Programming</td><td>16 Sessions (L8 + P8)</td></tr>
<tr><td colspan="5">Working with fs module, working with JSON, Using Buffer Module to Buffer Data, Using Stream Module to Stream Data, Compressing and Decompressing Data with Zlib.
Implementing HTTP Services in Node.JS: Introduction to HTTP module, Processing URLs, Processing Query Strings and Form Parameters, Understanding Request , Response and Server Objects.</td></tr>
<tr><td>Module 4</td><td>Web Development with Node.JS – CO4</td><td>Assignment</td><td>Programming</td><td>16 Sessions (L8 + P8)</td></tr>
<tr><td colspan="5">Introducing Express, More on Express, GET, POST, body Parser Creating Middleware with Connect : What is Middleware?, Middleware in Connect, Access Control with Middleware Socket Services in Node.js: : Understanding Network Sockets, A Socket.IO Chat Server, , A Streaming Twitter Client.</td></tr>
</table>

List of Laboratory Tasks:

Experiment No. 1:

Level 1: Install Node.js and npm on local machines.

Level 2: Use npm to manage dependencies and install packages

Experiment No. 2:

Level 1: Install Git on local machines. Initialize a local Git repository.

Level 2: Perform basic Git operations such as adding files, committing changes, and viewing the commit history

Experiment No. 3:

Level 1: Learn about the basics of npm commands such as npm init, npm install and npm publish

Level 2: Explore the npm registry to search for and install existing Node.js modules.

Experiment No. 4:

Level 1: Connect a Node.js application to MySQL

Level 2: Connect a Node.js to MySQL using Node.js driver.

Experiment No. 5:

Level 1: Install MongoDB on local machines or a virtual environment.

Level 2: Configure MongoDB to run as a service.

Experiment No. 6:

Level 1 : Access the MongoDB shell and perform basic database operations

Level 2: MongoDB shell Commands - Creating databases, Collections, inserting documents, Update, Delete

Experiment No. 7:

Level 1 : install external modules using node package manager(npm).

Level 2: Create a simple Node.js project.

Experiment No. 8:

Level 1: Create JavaScript Objects and functions Using object literal

Level 2: Create JavaScript Objects and functions by By creating instance of Object directly

Experiment No. 9:

Level 1 : Working with the arrays using Node.js

Level 2 : Assessing file system from Node.js.

Experiment No. 10 :

Level 1 : Create a basic food delivery website using node.js.

Level 2: Upload the project to GitHub repository.

Experiment No. 11:

Level 1 : implement basic socket services.

Level 2: Enhanced Socket Service with Rooms and Broadcasting.

Experiment No. 12 :

Create a website to manage the TO-DO list of users, where users can login and manage their to-do items using NodeJS and MongoDB using the official MongoDB Node.js driver

Targeted Application & Tools that can be used: Web Applications. MongoDB, Express.js, Node.js, Visual Studio Code, Sublime Text, Atom, Git and GitHub.

Project work/Assignment:

Create dynamic, interactive, and scalable web applications

Topics related to

1. Problem Solving: Choose building scalable server-side applications using Node.js to handle asynchronous data processing and real-time communication.

2. Employability: Simulation of full-stack web applications using Node.js, Express, and Socket.IO to demonstrate industry-relevant backend development skills.

Textbook(s):

T1: Professional Node.js: Building JavaScript Based Scalable Software,2020

T2. Sams Teach Yourself Node.js in 24 Hours, 2021 **T3** : Learn PostgreSQL: Build and manage high-performance database solutions using PostgreSQL,2020.
References **R1.** Chris Northwood, 'The Full Stack Developer: Your Essential Guide to the Everyday Skills Expected of a Modern Full Stack Web Developer', Apress; 1st edition, 2021 **R2.** Kirupa Chinnathambi, 'Learning React: A Hands-On Guide to Building Web Applications Using React and Redux', Addison-Wesley Professional, 4th edition,2020 **E-Resources:** 1. https://www.tutorialspoint.com/nodejs/index.htm 2. https://www.w3schools.com/nodejs/ 3. https://www.udemy.com/course/the-full-stack-web-development/

Course Code: **CSA4706**	**Course Title: Computer Vision** **Type of Course: Discipline Elective**	**L-T- P-**	3	0	0	3
Version No.	1.0					
Course Pre-requisites	**Data Structures and Algorithms, Web Technology**					
Anti-requisites	**NIL**					
Course Description	This course explores the foundations and advancements in digital image processing and computer vision. Learners will gain hands-on experience in analyzing visual data, applying machine learning to vision tasks, and understanding geometric techniques for depth and motion analysis. Emphasis is placed on both theoretical underpinnings and practical implementations across real-world applications like face recognition, object tracking, and scene reconstruction.					
Course Objective	**The objective of the course is EMPLOYBILITY of student by using PARTICIPATIVE LEARNING techniques.**					
Course Outcomes	**On successful completion of this course, the students shall be able to:** **CO1**: Apply techniques of digital image processing to enhance, filter, and interpret visual data for real-world applications. [**Apply**]					

<table>
<tr><td></td><td colspan="4"><b>CO2:</b> Apply machine learning methods for object recognition, classification, and segmentation in computer vision systems. [<b>Apply</b>]
<b>CO3</b>: Apply mathematical modeling techniques to perform low, intermediate, and high-level image processing operations. [<b>Apply</b>]
<b>CO4: Analyze</b> vision-based algorithms to evaluate their performance and suitability in dynamic, real-time environments. [<b>Analyze</b>]</td></tr>
<tr><td><b>Course Content:</b></td><td colspan="4"></td></tr>
<tr><td><b>Module 1</b></td><td><b>Foundations of Digital Image Processing - CO1</b></td><td><b>Assignment</b></td><td><b>Theory Heading</b></td><td><b>11 Sessions</b></td></tr>
<tr><td colspan="5">Image Formation, Image Filtering, Edge Detection, Principal Component Analysis, Corner Detection SIFT, Applications: Large Scale Image Search. Correspondence and Pose consistency, finding templates using classifiers, Recognition by relations between templates, Applications - Pattern classification, Face Recognition.</td></tr>
<tr><td><b>Module 2</b></td><td><b>Geometric Approaches in Computer Vision – CO2</b></td><td><b>Assignment</b></td><td><b>Theory Heading</b></td><td><b>12 Sessions</b></td></tr>
<tr><td colspan="5">Image Transformations, Camera Projections, Camera Calibration, Depth from Stereo, Two View Structure from Motion, Object Tracking.</td></tr>
<tr><td><b>Module 3</b></td><td><b>Machine Learning Techniques for Vision Tasks – CO3</b></td><td><b>Assignment</b></td><td><b>Theory Heading</b></td><td><b>11 Sessions</b></td></tr>
<tr><td colspan="5">Introduction to Machine Learning, Image Classification, Object Detection, Semantic Segmentation, Linear filters, Edge detection, Filters and Features, Texture.</td></tr>
<tr><td><b>Module 4</b></td><td><b>Advanced Topics in Mid-Level Vision – CO4</b></td><td><b>Assignment</b></td><td><b>Theory Heading</b></td><td><b>11 Sessions</b></td></tr>
<tr><td colspan="5">The geometry of multiple views - Stereopsis, Affine structure from motion, Correspondence and Pose consistency, finding templates using classifiers, Recognition by relations between templates, Applications - Pattern classification, Face Recognition.</td></tr>
<tr><td colspan="5"><b>Project work/Assignment:</b>
13. <b>Image Filtering and Feature Extraction</b>
14. <b>Geometric Transformation and Camera Calibration</b>
15. <b>Machine Learning for Image Classification</b>
16. <b>Project: Face Recognition System Development</b>
17. <b>Project: 3D Reconstruction from 2D Images</b>
18. <b>Project: Object Detection and Segmentation using Machine Learning</b></td></tr>
<tr><td colspan="5"><b>Topics related to</b>
<b>1. Problem Solving:</b> Develop and implement innovative image processing techniques to address challenges in segmentation, feature extraction, and pattern recognition in complex visual datasets.
<b>2. Employability:</b> Design and simulate a complete computer vision application such as a face recognition or object detection system that demonstrates practical, industry-relevant skills.</td></tr>
</table>

Textbook(s):

T1. Richard Szeliski, Computer Vision: Algorithms and Applications, Springer-Verlag London Limited 2011.

T2. Richard Hartley and Andrew Zisserman, Multiple View Geometry in Computer Vision, 2ndEdition, Cambridge University Press, March 2004.

References

R1. R. Bishop; Pattern Recognition and Machine Learning, Springer, 2006.

E-Resources:

1. https://www.tutorialspoint.com/dip/index.htm
2. https://docs.opencv.org/master/d9/d0c/group__calib3d.html
3. https://machinelearningmastery.com/start-here/#cv
4. https://web.stanford.edu/class/cs231a/

Course Code: CSA4707	**Course Title:** Natural Language Processing **Type of Course:** Discipline Elective	**L-T- P- C**	1	0	4	3
Version No.	1.0					
Course Pre-requisites	Data Structures and Algorithms					
Anti-requisites	**NIL**					
Course Description	This course provides a comprehensive introduction to Natural Language Processing (NLP), combining theoretical foundations with hands-on programming. Learners will explore fundamental text preprocessing, word and text representation techniques, sequence labeling with probabilistic models, and practical NLP applications like sentiment analysis, machine translation, and text summarization. The course integrates machine learning concepts such as embeddings, RNNs, LSTMs, and attention mechanisms, preparing students to build real-world NLP systems using modern toolkits and language models.					
Course Objective	**The objective of the course is EMPLOYBILITY of student by using PARTICIPATIVE LEARNING techniques.**					

<table>
<tr><td>Course Outcomes</td><td colspan="4"><b>On successful completion of this course, the students shall be able to:</b>
<b>CO1</b>: Apply appropriate text preprocessing techniques such as tokenization, stemming, lemmatization, and stop word removal to clean and prepare textual data for NLP tasks. <b>(Apply)</b>
<b>CO2:</b> Apply word and sentence representation techniques including TF-IDF, Word2Vec, and n-gram models to extract meaningful features from text for downstream applications. <b>(Apply)</b>
<b>CO3</b>: Analyze sequence labeling problems using models like Hidden Markov Models and evaluate performance using algorithms like Viterbi for tasks such as Named Entity Recognition. <b>(Analyze)</b>
<b>CO4:</b> Analyze and implement real-world NLP applications such as sentiment analysis, text summarization, and machine translation by integrating linguistic features with machine learning models. <b>(Analyze)</b></td></tr>
<tr><td>Course Content:</td><td colspan="4"></td></tr>
<tr><td>Module 1</td><td><b>Foundations of Natural Language Processing</b></td><td>Experimental Learning</td><td>Problem Solving</td><td><b>19 Sessions (4T+15L)</b></td></tr>
<tr><td colspan="5">Introduction to Natural Language Processing, terminologies, empirical rules, why NLP is hard, why NLP is useful, NLP Processing pipeline, Corpus Cleaning techniques – word tokenization, sentence tokenization, word frequency distribution, stemming, lemmatization, dictionary, Introduction to Part of Speech Tagging, Textual Pre-Processing techniques – Stop words removal, regular expression, lower case, text standardization.</td></tr>
<tr><td>Module 2</td><td><b>Word and Text Representations in NLP</b></td><td>Experimental Learning</td><td>Programming</td><td><b>19 Sessions (4T+15L)</b></td></tr>
<tr><td colspan="5">Word relationships, Word Embeddings techniques- bag of words, TF-iDF, Word2Vec and optimization. Simple N-gram models. Estimating parameters and smoothing. Negative Sampling Evaluating language models. Logistic regression – Sigmoid and Softmax. Perceptron and backpropagation. RNN, LSTM, CNN. Attention. Pre-trained language models. Multilinguality.</td></tr>
<tr><td>Module 3</td><td><b>Sequence Labeling and Parsing Techniques</b></td><td>Experimental Learning</td><td>Programming</td><td><b>19 Sessions (4T+15L)</b></td></tr>
<tr><td colspan="5">Sequence Labeling, Hidden Markov Models. Best Emission Probability, Best Forward Probability and Viterbi Decoding Algorithms. Analysis of Viterbi Algorithm. Named Entity Recognition. Constituency Parsing.</td></tr>
<tr><td>Module 4</td><td><b>Real-World Applications of NLP</b></td><td>Project based Learning</td><td>Programming</td><td><b>19 Sessions (3T+15L)</b></td></tr>
<tr><td colspan="5">Application of NLP. Lexical Resource Creation. Machine Translation. Sentiment Analysis. Lexical Simplification. Text Summarization.</td></tr>
<tr><td colspan="5">List of Laboratory Tasks:</td></tr>
</table>

Experiment No. 1:

Level 1: Reading text files

Level 2: Text classification – models and evaluation metrics

Experiment No. 2:

Level 1: Sentiment Analysis using Naïve Bayes Classifier with Bag of Words

Level 2: Sentiment Analysis using VADER

Experiment No. 3:

Level 1: Document Retrieval Using TF-IDF

Level 2: Using NLTK PoS Tagger

Experiment No. 4:

Level 1: Building a PoS Tagger with Brown Corpus Universal Tagset and Viterbi Algorithm

Level 2: Generating a Parse Tree for a sentence

Experiment No. 5:

Level 1: Named Entity Recognition in a Document

Level 2: Machine Translation using Huggingface’s Helsinki Opus MT models.

Targeted Application & Tools that can be used:

Python (NLTK, spaCy, Gensim, Scikit-learn), TensorFlow, PyTorch, Hugging Face Transformers, Jupyter Notebook, Google Colab, Stanford NLP, TextBlob, Flair

Project work/Assignment:

1. Text Classification App
2. Sentiment Analysis Dashboard
3. Chatbot using RNN or Transformers
4. Named Entity Recognition System
5. Machine Translation App
6. Text Summarizer
7. Resume Parser for HR Systems
8. Topic Modeling on Research Papers
9. Speech-to-Text Processing Pipeline
10. Grammar and Spell Checker

Topics relevant to the development of Employability:

Problem Solving: Text preprocessing, POS tagging, NER, sentiment analysis, and machine translation challenges.

Employability: Building NLP pipelines, deploying models, chatbot development, word embeddings, and real-world applications using Python libraries.

Textbook(s):

T1. Daniel Jurafsky, and James H. Martin. Speech and Language Processing. (3rd Edition Draft, February 2024)

T2. Aditya Joshi, and Pushpak Bhattacharyya. Natural Language Processing. 1st Edition. Wiley Publishers. December 2023.

References

R1. Chris Manning and Hinrich Schutze, "Foundations of Statistical Natural Language Processing", 1st Edition, MIT Press. 1999.

R2. Pawan Goyal. "Natural Language Processing". 1st Edition, 2016.

E-Resources:

Weblinks

1. https://drive.google.com/file/d/10nbwAJd-dv6htOOZVBgAvLd1WscI0RqC/view
2. https://web.stanford.edu/~jurafsky/slp3/
3. https://nptel.ac.in/courses/106106211
4. https://nptel.ac.in/courses/106105158
5. https://nptel.ac.in/courses/106101007
6. https://nptel.ac.in/courses/106105572

Course Code: **CSA4708**	**Course Title: Reinforcement Learning** **Type of Course: Discipline Elective**	**L-T- P- C**	3	0	0	3
Version No.	1.0					
Course Pre-requisites	**Machine Learning**					
Anti-requisites	**NIL**					
Course Description	This course introduces the foundational concepts and practical applications of Reinforcement Learning (RL). Learners will explore core elements like agent-environment interactions, rewards, and policies through the lens of Markov Decision Processes (MDP). The course progresses through Monte Carlo methods, Temporal Difference learning (including SARSA and Q-Learning), and culminates with Multi-Armed Bandit (MAB) problems and an introduction to Deep Reinforcement Learning (DRL). Practical implementation using the OpenAI Gym environment is emphasized throughout to build hands-on skills in solving real-world sequential decision-making problems.					

Course Objective	The objective of the course is EMPLOYBILITY of student by using PARTICIPATIVE LEARNING techniques**.**			
Course Outcomes	**On successful completion of this course, the students shall be able to:** **CO1**: Infer the basic reinforcement learning environments to solve simple MDP problems using appropriate policy evaluation techniques. **CO2:** Build Monte Carlo prediction and control algorithms for model-free environments and evaluate their performance. **CO3**: Analyse the functionality of SARSA and Q-learning algorithms and determine their impact on policy optimization. **CO4:** Develop exploration strategies in Multi-Armed Bandit problems and demonstrate the selection of suitable algorithms for various real-time scenarios.			
Course Content:				
Module 1	**Foundations of Reinforcement Learning and MDPs - CO1**	**Assignment**	**Programming**	**11 Sessions**
Elements of RL, Agent, environment Interface, Goals and rewards, RL platforms, Applications of RL, Markov decision process (MDP), RL environment as a MDP, Maths essentials of RL, Policy and its types,episodic and continuous tasks, return and discount factor, fundamental functions of RL – value and Q functions, model-based and model-free learning, types of RL environments, Solving MDP using Bellman Equation, Algorithms for optimal policy using Dynamic Programming -Value iteration and policy iteration, Example : Frozen Lake problem, Limitations and Scope.				
Module 2	**Monte Carlo Methods for Prediction and Control – CO2**	**Assignment**	**Programming**	**11 Sessions**
Monte Carlo methods, prediction and control tasks, Monte Carlo prediction: algorithm, types of MC prediction, examples, incremental mean updates, Monte Carlo Control: algorithm, on-policy MC control, MCwith epsilon-greedy policy, off-policy MC control. Limitations of MC method.				
Module 3	**Temporal Difference Learning Techniques – CO3**	**Quiz**	**Programming**	**12 Sessions**
Temporal difference learning: TD Prediction, TD Control: On-policy TD control – SARSA, computing theoptimal policy using SARSA, Off-policy TD control – Q learning, computing optimal policy using Q learning, Examples, Difference between SARSA and Q-learning, Comparison of DP, MC and TD methods.				
Module 4	**Multi-Armed Bandits and Introduction to Deep Reinforcement Learning – CO4**	**Assignment**	**Programming**	**11 Sessions**
Understanding the MAB problem, Various exploration strategies – epsilon-greedy, softmax exploration, upper confidence bound and Thompson sampling, Applications of MAB - finding the				

best advertisement banner for a web site, Contextual bandits, introduction to Deep Reinforcement Learning (DRL) Algorithm – Deep Q Network (DQN).

Project work/Assignment:

Assignment on MDP Formulation.

Comparison Report: SARSA vs. Q-Learning.

Exploration Strategy Evaluation.

Project: Frozen Lake Navigation using Policy Iteration

Project: Train an Agent using Q-Learning to Play CartPole

Project: Contextual Multi-Armed Bandit for Ad Recommendation

Topics related to

1. Problem Solving: Design a reinforcement learning-based strategy to solve the Frozen Lake navigation problem using Value Iteration and Policy Iteration.

2. Employability: Simulate real-world decision-making environments using Multi-Armed Bandit algorithms to recommend optimal content or advertisements.

Textbook(s):

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, SecondEdition, 2020.

References

R1. Laurra Graesser and Wan Loon Keng, "Foundations of Deep Reinforcement Learning", Pearson, 2022.

E-Resources:

1. https://www.davidsilver.uk/teaching/
2. https://spinningup.openai.com/en/latest/
3. https://www.geeksforgeeks.org/temporal-difference-learning/
4. https://lilianweng.github.io/lil-log/2018/05/22/exploration-strategies-in-deep-reinforcement-learning.html

Course Code: **CSA4709**	**Course Title:** **Deep Learning** **Type of Course:** **Theory & Integrated Laboratory**	**L-T- P- C**	**2**	**0**	**2**	**3**
Version No.	1.0					
Course Pre-requisites	**Machine learning**					
Anti-requisites	**NIL**					

Course Description	This course provides a comprehensive introduction to Deep Learning, equipping students with foundational knowledge and practical experience in building and training deep neural networks. Starting with feedforward networks and optimization techniques, the course progresses through convolutional neural networks for image and text classification, sequence modeling using RNNs and LSTMs, and culminates in the exploration of autoencoders for unsupervised learning and representation. Emphasis is placed on hands-on assignments, quizzes, and real-world projects to ensure a deep understanding of both theory and application.			
Course Objective	The objective of the course is EMPLOYBILITY of student by using PARTICIPATIVE LEARNING techniques.			
Course Outcomes	**On successful completion of this course, the students shall be able to:** **CO1**: Apply optimization techniques such as stochastic gradient descent and momentum to train feedforward neural networks. **(Apply)** **CO2:** Implement convolutional neural networks for image and text classification tasks. **(Apply)** **CO3**: Implement recurrent neural networks and LSTM architectures for sequence modeling problems. **(Apply)** **CO4:** Develop autoencoder-based models to perform dimensionality reduction and feature extraction on real-world data. **(Apply)**			
Course Content:				
Module 1	**Foundations of Deep Learning and Feedforward Networks - CO1**	Assignments	Problem Solving	**16 Sessions (8T + 8 L)**
Introduction: Historical context and motivation for deep learning; basic supervised classification task, optimizing logistic classifier using gradient descent, stochastic gradient descent, momentum, and adaptive sub-gradient method. Neural Networks: Feedforward neural networks, deep networks, regularizing a deep network, model exploration, and hyper parameter tuning.				
Module 2	**Convolutional Neural Networks for Image and Text – CO2**	Assignments	Programming	**16 Sessions (8T + 8 L)**
Introduction to convolution neural networks: stacking, striding and pooling, applications like image, and text classification.				
Module 3	**Sequence Modeling with Recurrent Neural Networks – CO3**	Quiz	Problem Solving	**15 Sessions (7T + 8 L)**

Unfolding computational graphs, recurrent neural networks (RNNs), bidirectional RNNs, encoder-decoder sequence to sequence architectures, deep recurrent networks, LSTM networks.				
Module 4	**Autoencoders for Representation Learning – CO4**	Project	Programming	**13 Sessions (7T + 6 L)**
Autoencoders: Undercomplete autoencoders, regularized autoencoders, sparse autoencoders, denoising autoencoders, representational power, layer, size, and depth of autoencoders, stochastic encoders and decoders.				

List of Laboratory Tasks:

Experiment No. 1:

Level 1: Implement logistic regression classification with (a) gradient descent and (b) stochastic gradient descent method. Plot cost function over iteration.

Level 2: Experiment with logistic regression by adding momentum term, and adaptive sub gradient method

Experiment No. 2:

Level 1: Implement a feed-forward neural network for solving (a) regression and (b) 2-class classification problem. Also experiment with hyper-parameter tuning.

Level 2: Train and test a feed-forward neural network for multi-class classification using softmax layer as output.

Experiment No. 3:

Level 1: Create a 2D and 3D CNN for image classification. Experiment with different depth of network, striding and pooling values.

Level 2: CNN-based model for sentiment analysis on a text dataset such as movie reviews or tweets.

Experiment No. 4:

Level 1: Implement (a) RNN for image classification, (b) GRU network and (c) Implement LSTM networks

Level 2: Simple Recurrent Neural Network (RNN) for predicting next word in a sentence.

Experiment No. 5:

Level 1: Bidirectional RNN for Sequence Classification

Level 2: Encoder-Decoder Architecture for Machine Translation

Experiment No. 6:

Level 1: LSTM Networks for Time-Series Prediction

Level 2: Implement an auto-encoder, denoising autoencoders and sparse autoencoders.

Experiment No. 7:

Level 1: Design stochastic encoders and decoders.

Targeted Application & Tools that can be used:

Image classification, text classification, sequence prediction, and representation learning using tools such as TensorFlow, Keras, PyTorch, OpenCV, and Jupyter Notebook.

Project work/Assignment:

1. Handwritten Digit Recognition using CNN (MNIST Dataset)
2. Sentiment Analysis using RNN or LSTM
3. Image Denoising with Autoencoders
4. Real-Time Object Detection System
5. Stock Price Prediction using LSTM
6. Face Recognition System using Deep Neural Networks

Topics related to

1. Problem Solving: Choose appropriate deep learning architectures (CNN, RNN, Autoencoders) for specific real-world data challenges such as image classification or sequence prediction.

2. Employability: Simulation of deep learning-based systems such as image classification, object detection, or time-series forecasting to develop industry-relevant skills.

Textbook(s):

T1. Bunduma, N. (2017). Fundamentals of Deep Learning. O'reilly Books

T2. Heaton, J. (2015). Deep Learning and Neural Networks, Heaton Research Inc.

T3. Goodfellow, I. (2016). Deep Learning. MIT Press.

References

R1. Deng, L., & Yu, D. (2009). Deep Learning: Methods and Applications (Foundations and Trends in Signal Processing). Publishers Inc.

R2. Hall, M.L, (2011). Deep Learning. VDM Verlag

R3. David Foster, "Generative Deep Learning" O'Reilly Publishers, 2020.

R4. John D Kellehar, "Deep Learning", MIT Press, 2020.

Additional web-based resources

1. https://onlinecourses.nptel.ac.in/noc22_cs22/preview
2. https://www.coursera.org/learn/neural-networks-deep-learning?specialization=deep-learning
3. https://www.deeplearning.ai/
4. http://imlab.postech.ac.kr/dkim/class/csed514_2019s/DeepLearningBook.pdf

<table>
<tr><td>Course Code: CSA4710</td><td colspan="3">Course Title: Generative AI
Type of Course: Elective</td><td>L-T- P- C</td><td>2</td><td>0</td><td>2</td><td>3</td></tr>
<tr><td>Version No.</td><td colspan="8">1.0</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="8"><b>CSA4709 – Deep Learning</b>
<b>CSA4707 – Natural Language Processing</b>
<b>CSA4502 - Machine Learning</b></td></tr>
<tr><td>Anti-requisites</td><td colspan="8"><b>NIL</b></td></tr>
<tr><td>Course Description</td><td colspan="8">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.</td></tr>
<tr><td>Course Objective</td><td colspan="8">The objective of the course is EMPLOYBILITY of student by using Experiential LEARNING techniques.</td></tr>
<tr><td>Course Outcomes</td><td colspan="8"><b>On successful completion of this course, the students shall be able to:</b>
<b>CO1</b>: Infer the concepts of generative AI models and prompt engineering in tailoring customized outputs.
<b>CO2:</b> Demonstrate attention mechanism and transformers architecture with practical Applications
<b>CO3</b>: Practice advanced generative AI techniques using Langchain Python framework
<b>CO4:</b> Solve real-time applications using multi-modal generative AI models.</td></tr>
<tr><td>Course Content:</td><td colspan="8"></td></tr>
<tr><td>Module 1</td><td><b>Introduction to Generative AI-CO1</b></td><td>Participative Learning</td><td>Brainstorming session/Quiz</td><td colspan="5"><b>14 Sessions (L6+P8)</b></td></tr>
<tr><td colspan="9">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, Claud and its variants by Anthropic, Prompt Engineering-basic prompting</td></tr>
<tr><td>Module 2</td><td><b>Text-based Generative models-CO2</b></td><td>Participative Learning</td><td>Fish bowl, Think-pair &amp; share</td><td colspan="5"><b>14 Sessions (L8 + P6)</b></td></tr>
</table>

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: ChatGPTs, 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 Lang Chain – CO3**	Experiential Learning	Implementation of Gen AI models using Langchain Framework	**16 Sessions (L8 + P8)**

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-CO4**	Project based Learning	Multi-Modal Gen AI models for Realtime Applications	**16 Sessions (L8 + P8)**

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.

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.

Level 2:

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

Level 2:

Experiment No.5: Transformer based text and email classification

Level 1: **Develop** transformer-based AI models for classifying text/email

Level 2:

Experiment No.6: BERT for masked token generation

Level 1: **Develop** BERT based model for generating masked tokens

Level 2:

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

Level 2:

Experiment No.9: Custom Document loading and retrieval in LangChain using ChromaDB

Level 1: **Understand** ChromeDb

Level 2: **Apply** chromed with Langchain to generate information retrieval model from custom document

Experiment No.10: Create a GPT like Chatbot using the memory component and RALM in LangChain

Level 1: **Show** GPT like chatbot using memory component and retrieval augmented language model

Level 2:.

Experiment No.11: 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.12: : Implement GAN for neural style transfer

Level 1: **Demonstrate** a style transfer algorithm using generative models and experiment with the transformation of images by applying different artistic styles, assessing both the technical aspects and the aesthetic outcomes

Level 2:

Experiment No.13: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.14 : Image to Image generation using stable diffusion

Level 1: Apply stable diffusion to generate image from an image using prompts

Level 2:

Experiment No.15: 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

Targeted Application & Tools that can be used:

Open AI Generative AI models: GPT 3.5 Turbo, GPT 4.0 vision model, Dall-E 3.0, Lang Chain Framework in Python, Python IDE, Stable Diffusion, Gemini, Hugging Face,.

Project work/Assignment:

- Mini-Project Titles:
- Conversational Chatbot that interacts with documents: create a conversational chatbot to engage users in meaningful dialogues, answer queries, offer recommendations, and aid tasks using provided documents as inputs.
- Sentiment Analysis/Intent Analysis/Toxicity Analysis
- Natural Language Translation – Instruction Tuning using FLAN (Finetuned language Net) model
- Questions and Answering systems – Extractive & Generative
- Text Summarization – Medicine – Med-PaLM
- Given the Academic guidelines of the University, generate the student Handbook with FAQs and solutions.
- Generating Cartoon based story telling
- Simulate various driving conditions to improve safety and performance in Autonomous vehicles
- In Financial management, generate synthetic financial data for stress testing and scenario analysis
- Personalized recommendations/Product suggestions/tailored content based personalized design studio
- Simulate characters for Games
- Create conversational agents
- Tutor in a range of preferred subjects
- Generate codes
- Draft documents
- Answer questions about any knowledge base
- Create an application which uses LangChain to connect OpenAI API to DALL-E. This image generation application turns written descriptions into lifelike pictures and artwork.
- Embark on building a personalized language model with Falcon-7b. Utilize personalized LLM technique to explore text generation capabilities by providing task examples as inputs.
- Use OpenAI's DALL-E and Gradio UI to develop an innovative logo builder. Th app creates unique and stunning logos from text prompts, revolutionizing the logo design process.

- Crafting an AI powered HR Assistant: Develop a virtual assistant designed to answer queries related to Audi HR policy. Leverage Python libraries and OpenAI’s GPT model for accurate and efficient query responses.

Topics Related To :

Textbook(s):

T1. Generative AI with LangChain, Ist Edition by Ben Auffarth, Packt. Inc. ISBN: 978-1-83508-346-8, Decemeber 2023

T2. Generative Deep Learning,2nd Edition by David Foster, O'Reilly Media, Inc. ISBN: 9781098134181, May 2023.

T3. Prompt Engineering for Generative AI, by James Phoenix, Mike Taylor, O'Reilly Media, Inc., ISBN:9781098153373, July 2024

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.

R5. Salvaris, M., Dean, D., & Tok, W. H. (2018). Generative adversarial networks. In Apress eBooks (pp. 187–208). https://doi.org/10.1007/978-1-4842-3679-6_8

E-Resources:

W1. https://openai.com

W2: https://python.langchain.com/v0.2/docs/introduction/

W3: https://www.udemy.com/course/master-ai-image-generation-using-stable-diffusion/?kw=Image+generation+using&src=sac&couponCode=LETSLEARNNOWPP

W4: https://huggingface.co/google-t5/t5-base

W5: https://dominguezdaniel.medium.com/exploring-image-generative-ai-models-9359705b15d3

W6: https://cloud.google.com/use-cases/retrieval-augmented-generation?hl=en#

W7: https://ig.ft.com/generative-ai/

W8: https://medium.com/@samia.khalid/bert-explained-a-complete-guide-with-theory-and-tutorial-3ac9ebc8fa7c

MOOC’s/Swayam Courses/Online Courses:

https://onlinecourses.swayam2.ac.in/imb24_mg116/preview

Global Certification Course by Google :

1. https://www.cloudskillsboost.google
   - **a.** Introduction to Generative AI (Beginner)
   - **b.** Gemini for Google Cloud (Intermediate)
   - **c.** Generative AI for Developers (Advanced)
2. https://www.credly.com/badges/90e3eae0-87f3-44e3-af82-658e837aad3d/public_url
3. https://www.coursera.org/learn/generative-ai-with-llms
4. https://www.coursera.org/specializations/prompt-engineering

Course Code: CSA4711	Course Title: Cyber Security and Ethical Hacking Type of Course: Discipline Elective	L- T - P- C	3	0	0	3
Version No.	1.1					
Course Pre-requisites	Basic knowledge of Computer Networks, Operating Systems, and Programming Concepts					
Anti-requisites	**NIL**					
Course Description	This course offers a comprehensive introduction to cyber security and ethical hacking. It addresses core concepts in security architecture, threat analysis, penetration testing, and system defense mechanisms. Students will learn how to ethically identify vulnerabilities in systems and mitigate them using appropriate tools and techniques, guided by industry best practices and legal frameworks.					
Course Objective	The objective of the course is to familiarize the learners with the concepts of principles of cyber security and the practical skills and attain **SKILL DEVELOPMENT** through PARTICIPATIVE LEARNING techniques**.**					
Course Out Comes	On successful completion of this course the students shall be able to: 1. Understand and explain the fundamental concepts of cyber security, threats, vulnerabilities, and risk management. **[Understand]** 2. Apply techniques to secure networks and systems using tools such as firewalls, intrusion detection systems, and endpoint security solutions. **[Apply]** 3. Apply identity and access management practices, including cryptographic methods, to enforce secure authentication and data confidentiality. **[Apply]** 4. Apply ethical hacking methodologies to identify vulnerabilities and simulate cyber-attacks in a controlled environment. **[Apply]**					
Course Content:						

<table>
<tr><td><b>Module 1</b></td><td><b>Introduction to Cyber Security</b></td><td>Assignment</td><td>Programming Task</td><td><b>12 Sessions</b></td></tr>
<tr><td colspan="5"><b>Topics:</b>
Introduction to Cyber Security : Concept of Cyber Security, Types of Cyber Attacks, Threats and Vulnerabilities, Security Goals (CIA Triad), Information Assurance, Security Policies, Risk Management, and Cyber Security Standards (ISO 27001, NIST.</td></tr>
<tr><td><b>Module 2</b></td><td><b>System and Network Security</b></td><td>Assignment</td><td>Programming Task</td><td><b>12 Sessions</b></td></tr>
<tr><td colspan="5"><b>Topics:</b>
Operating System Security, File Permissions and Access Control, Malware (Viruses, Worms, Trojans), Network Protocols (TCP/IP, DNS, HTTP), Firewalls, Intrusion Detection/Prevention Systems (IDS/IPS), VPNs, Wireless Security (WPA2, WPA3), Network Scanning Tools (Nmap, Wireshark)</td></tr>
<tr><td><b>Module 3</b></td><td><b>Ethical Hacking Tools and Techniques</b></td><td>Assignment</td><td>Programming Tasks</td><td><b>10 Sessions</b></td></tr>
<tr><td colspan="5"><b>Topics:</b>
Phases of Ethical Hacking: Reconnaissance, Scanning, Gaining Access, Maintaining Access, Clearing Tracks. Password Cracking, Social Engineering, SQL Injection, XSS, Buffer Overflow, Metasploit Framework, Kali Linux tools (Aircrack-ng, John the Ripper), Enumeration Techniques</td></tr>
<tr><td><b>Module 4</b></td><td><b>Penetration Testing and Countermeasures</b></td><td>Assignment</td><td>Programming Tasks</td><td><b>11 Sessions</b></td></tr>
<tr><td colspan="5"><b>Topics:</b> Application
Penetration Testing Methodology, Vulnerability Assessment, Reporting and Documentation, Web Application Security, OWASP Top 10, Security Audit, Countermeasures, Patch Management, Incident Response, Digital Forensics Overview</td></tr>
<tr><td colspan="5"><b>Text Book</b>
1. Charles J. Brooks, Christopher Grow, Philip Craig, Donald Short, “Cybersecurity Essentials,” 2nd Edition, Jones & Bartlett Learning, 2022 [ISBN: 9781284235667]
2. Craig S. Wright, “Security Management Practices: A Step-by-Step Guide”, <i>1st Edition</i>, Auerbach Publications, USA, 2021. [ISBN-978-1032092223].
3. Erdal Ozkaya, <i>Cybersecurity – The Beginners Guide: A comprehensive guide to getting started in cybersecurity</i>, 2nd Edition, Packt Publishing, 2023. [ISBN: 9781804616436]</td></tr>
</table>

References

1. Ric Messier, *Certified Ethical Hacker (CEH) v12 Study Guide*, Wiley, 2023. [ISBN: 9781394162325]
2. Jon DiMaggio, *The Art of Cyberwarfare: An Investigator's Guide to Espionage, Ransomware, and Organized Cybercrime*, No Starch Press, 2022. [ISBN: 9781718502147]
3. Daniel G. Graham, *Practical Ethical Hacking: The Hands-On Guide to Breaking In*, No Starch Press, 2022. [ISBN: 9781718502192].

Web Based Resources and E-books:

1. https://www.jblearning.com/catalog/productdetails/9781284235667
2. https://nptel.ac.in/courses/106105031
3. https://www.cybrary.it/
4. https://www.hackthebox.com/
5. https://owasp.org/www-project-top-ten/

Topics relevant to "SKILL DEVELOPMENT Security Patterns and architectural elements, Managed Security Service Provider(for developing Skills through PARTICIPATIVE LEARNING techniques. This is attained through assessment component mentioned in the course handout.

Course Code: **CSA4712**	**Course Title: Web Application Security** **Type of Course: Discipline Elective**	**L-T- P- C**	1	0	4	3
Version No.	1.0					
Course Pre-requisites	**Web Technology**					
Anti-requisites	**NIL**					
Course Description	This course introduces foundational and advanced concepts in web application and API security. It explores real-world threats, secure coding practices, encryption techniques, and vulnerability assessments. Learners will engage with hands-on assignments and projects aimed at building secure systems and defending against evolving cyber threats.					
Course Objective	The objective of the course is EMPLOYBILITY of student by using PARTICIPATIVE LEARNING techniques.					
Course Outcomes	**On successful completion of the course, the students shall be able to:** **CO1:** Analyze the roles of authentication, authorization, session management, and input validation in securing web applications, and examine their interdependencies in mitigating common vulnerabilities. (Analyze) **CO2:** Apply secure development lifecycle models such as SDL and CLASP to design and implement secure web application architectures. (Apply) **CO3:** Apply security mechanisms including OAuth2, encryption, and service mesh for securing APIs in microservices and IoT-based systems. (Apply) **CO4:** Apply vulnerability assessment techniques such as scanning and penetration testing on platforms like web, wireless, and mobile to identify and address security flaws. (Apply)					
Course Content:						
Module 1	**Foundations of Web Application Security - CO1**	Participative Learning	Theory Heading	**19 Sessions (4T + 15P)**		
The history of Software Security-Recognizing Web Application Security Threats, Web Application Security, Authentication and Authorization, Secure Socket layer, Transport layer Security, Session Management-Input Validation						
Module 2	**Secure Software Development Lifecycle (SDLC) – CO2**	Assignments	Theory Heading	**19 Sessions (4T + 15P)**		

Web Applications Security - Security Testing, Security Incident Response Planning, The Microsoft Security Development Lifecycle (SDL), OWASP Comprehensive Lightweight Application Security Process (CLASP), The Software Assurance Maturity Model (SAMM)

Module 3	**API and Microservices Security – CO3**	Participative Learning	Theory Heading	**19 Sessions (4T + 15P)**

Encryption, Audit logging, securing service-to-service APIs: API Keys, OAuth2, Securing Micro service APIs: Service Mesh, Locking Down Network Connections, Securing Incoming Requests.

Module 4	**Vulnerability Assessment and Penetration Testing – CO4**	Project	Theory Heading	**18 Sessions (3T + 15P)**

Network-based vulnerability scanners, Database based vulnerability scanners, Types of Penetration Tests: External Testing, Web Application Testing, Internal Penetration Testing, SSID or Wireless Testing, Mobile Application Testing.

List of Laboratory Tasks:

Experiment No. 1:

Level 1: Install wireshark and explore the various protocols

Level 2: 2. Analyze the difference between HTTP vs HTTPS

Experiment No. 2:

Level 1: Analyze the various security mechanisms embedded with different protocols.

Level 2: Identify the vulnerabilities using OWASP ZAP tool

Experiment No. 3:

Level 1: Create simple REST API using python for following operation

1. GET 2. PUSH 3. POST 4. DELETE

Experiment No. 4:

Level 1: Install Burp Suite to do following vulnerabilities:

1. SQL injection 2. cross-site scripting (XSS)

Experiment No. 5:

Level 1: 7. Attack the website using Social Engineering method

Targeted Application & Tools that can be used:

OWASP ZAP, Burp Suite, Postman, Swagger, OAuth2, JWT, Nmap, Nikto, Metasploit, Spring Security, Node.js Security Packages, Docker, Kubernetes, Istio Service Mesh.

Project work/Assignment:

1. Secure Online Banking Portal

2. API Security for IoT Application
3. Vulnerability Assessment Report
4. Design a Secure E-commerce Platform
5. **Mobile App Security Analysis**

Topics related to

1. Problem Solving: Choose secure coding practices and vulnerability mitigation strategies to resolve common web and mobile application security issues.

2. Employability: Simulation of real-world cyber-attack scenarios using penetration testing and secure API implementation to enhance hands-on industry readiness.

Textbook(s):

T1. Andrew Hoffman, Web Application Security: Exploitation and Countermeasures for Modern Web Applications, First Edition, 2020, O'Reilly Media, Inc.

T2. Bryan Sullivan, Vincent Liu, Web Application Security: A Beginners Guide, 2012, The McGraw-Hill Companies

References

R1. Neil Madden, API Security in Action, 2020, Manning Publications Co., NY, USA.

E-Resources:

1. https://www.coursera.org/projects/web-application-security-testing-with-owsap-zap
2. https://www.coursera.org/learn/web-application-security

Course Code: CSA4713	Course Title: **Cybersecurity Testing** Type of Course: **Theory Course**	L-T-P-C	3	0	0	3
Version No.	1.0					
Course Pre-requisites	Students should have basic knowledge of Python programming, cybersecurity fundamentals, machine learning, and network security. Familiarity with ML frameworks like TensorFlow and scikit-learn, along with networking concepts such as TCP/IP and IDS/IPS, is recommended. A solid understanding of mathematics, particularly statistics, probability, and linear algebra, is essential. This background will help students engage effectively with cybersecurity testing. A foundation in Computer Science, IT, or Cybersecurity will be beneficial.					
Anti-requisites	**NIL**					
Course Description	The **Cybersecurity Testing** course provides a comprehensive overview of methods and tools used to assess and enhance the security of systems, networks, and applications. It focuses on testing techniques like penetration testing, vulnerability scanning, and security auditing to identify potential risks. The course covers automated testing tools, manual testing methodologies, and compliance with security standards. Students will gain hands-on experience with real-world cybersecurity testing scenarios, learning how to detect and mitigate threats effectively while ensuring system robustness and data protection.					
Course Objective	1. To understand the principles and methodologies of cybersecurity testing. 2. To explore various testing techniques, including penetration testing, vulnerability assessment, and security auditing. 3. To apply automated tools and manual methods for identifying and mitigating security vulnerabilities. 4. To ensure compliance with industry security standards and best practices during testing. 5. To provide hands-on experience with real-world scenarios for testing and securing systems, networks, and applications.					

Course Out Comes	**CO1**: Explain the importance of cybersecurity testing methods in identifying vulnerabilities and improving security. (Understand, Describe, Identify) **CO2**: Implement cybersecurity testing techniques such as penetration testing, vulnerability scanning, and security auditing. (Apply, Execute, Perform) **CO3**: Analyze test results to evaluate system security and identify potential threats. (Analyze, Evaluate, Assess) **CO4**: Develop and deploy mitigation strategies based on test outcomes to enhance overall security. (Develop, Deploy, Optimize)			
Course Content:				
Module 1	**Introduction to Cybersecurity**	Assignment	Theory Task	**10 Sessions**
Topics: Cybersecurity Fundamentals, Cyber Threats & Attack Types, Security Principles, Authentication & Access Control , Encryption Basics, Security in Network & Systems, Emerging Trends in Cybersecurity				
Assignment: Quiz I				
Module 2	**Security Testing Methodologies**	Assignment	Theory Task	**12 Sessions**
Topics: Introduction to Security Testing, Types of Security Testing, Threat Modeling & Risk Assessment, Security Testing Tools Overview, Application Security Testing, API Security & Mobile App Testing, Cloud Security Testing, Case Study on Security Breaches & Testing Approaches. **Assignment:** Quiz II				
Module 3	**OWASP and Web Security**	Assignment	Theory task	**12 Sessions**
Topics: Introduction to OWASP, OWASP Top 10 Vulnerabilities, Web Security Testing Methodologies, Best Practices for Secure Web Development, Industry Case Studies on Web Security Breaches. **Assignment:** Assignment 1, Test 1				

Module 4	**Security Implementation & Best Practices**	Assignment	Theory task	**11Sessions**

Topics:

Introduction to Risk Management in Cybersecurity, Cybersecurity Governance & Policies, Risk Assessment Frameworks, Security Compliance & Regulations, Incident Response & Disaster Recovery Planning, Security Audits & Vulnerability Management, Case Studies on Cyber Risk Management Failures

Assignment: Assignment 2, Test 2

Targeted Application & Tools that can be used:

NIL

Project work/Assignment:

To understand the application of cybersecurity Testing in daily lives the following assignments, Quizzes and Tests are included:

Assignment: 1] Module 3

Assignment: 2] Module 4

Text Book

1] **Kutub Thakur, Al-Sakib Khan Pathan**, *Cybersecurity Fundamentals: A Real-World Perspective*, 2020.

2] **Bryan Sullivan, Vincent Liu**, *Web Application Security: A Beginner's Guide*, Updated Edition, 2024.

3] **Krag Brotby**, *Information Security Governance: A Practical Development and Implementation Approach*, Updated Edition, 2024.

References

References

R1. Prakhar Prasad, *Mastering Modern Web Penetration Testing*, Packt Publishing, 2016.

R2. Alfred Basta, Nadine Basta, Mary Brown, *Computer Security and Penetration Testing* (2nd Edition), Cengage Learning, 2013.

Weblinks:

https://youtu.be/3DZLItfbqtQ

https://www.geeksforgeeks.org/last-minute-notes-computer-network/

Topics relevant to SKILL DEVELOPMENT: The **Cybersecurity Testing** course emphasizes skill development through practical exposure to key topics such as penetration testing, vulnerability scanning, and security auditing. Students will learn how to assess system vulnerabilities, evaluate the effectiveness of security measures, and ensure compliance with industry standards.

Course Code: **CSA4714**	**Course Title: Cloud Security** **Type of Course: Discipline Elective**		**L-T- P- C**	2	0	2 3
Version No.	1.0					
Course Pre-requisites	Cloud Computing					
Anti-requisites	**NIL**					
Course Description	The Cloud Security course provides an in-depth understanding of the principles, techniques, and best practices for securing cloud environments. The course covers a broad spectrum of security challenges associated with cloud computing, including data security, network security, application security, and infrastructure security. Students will learn to design, implement, and manage security solutions within public, private, and hybrid cloud environments, focusing on real-world scenarios and industry standards. The course emphasizes security frameworks such as NIST, ISO 27001, and OWASP, along with regulatory compliance including GDPR, HIPAA, and PCI DSS.					
Course Objective	The objective of the course is EMPLOYBILITY of student by using PARTICIPATIVE LEARNING techniques.					
Course Outcomes	**On successful completion of this course, the students shall be able to:** **CO1**: Apply encryption methods, including homomorphic encryption and Public Key Infrastructure to secure cloud data. **(Apply)** **CO2:** Develop secure virtualization environments, including hypervisor security and virtual machine (VM) isolation (**Apply)** **CO3**: Implement Security as a Service (SECaaS) solutions in cloud applications **(Apply)** **CO4:** Analyze security threats and intrusion detection in cloud environments **(Apply)**					
Course Content:	**The objective of the course is EMPLOYBILITY of student by using PARTICIPATIVE LEARNING techniques.**					
Module 1	**Cloud Security Fundamentals-CO1**	Assignment	Programming Task	**14 Sessions (L6 + P8)**		
Cloud Security Fundamentals Cloud computing security challenges – cloud computing security architecture – data security life-cycle - Security Patterns and architectural elements - Planning key Strategies for secure operation.						

<table>
<tr><td colspan="5">Cloud Application Security Encryption techniques – homomorphic encryption - securing data Redaction - secure bitcoin – Public key infrastructure (PKI) – key management - open web application security project (OWASP) Cloud Top 10 Security Risks - Security as a service (SECaaS)</td></tr>
<tr><td><b>Module 2</b></td><td><b>Cloud Infrastructure Security – CO2</b></td><td>Assignment</td><td>Programming Task</td><td><b>14 Sessions (L8 + P6)</b></td></tr>
<tr><td colspan="5">Security Management & Privacy Managed Security Service Provider (MSSP): Availability management – configuration management - vulnerability management - identity management. - Privacy: privacy, compliance and the cloud - privacy enhancing encryption.
Risk Management & Security Threats Risk management – principles - assessing the risk – strategies for managing risk – risk analysis framework – security threats - intrusion detection</td></tr>
<tr><td><b>Module 3</b></td><td><b>Security Management – CO3</b></td><td>Assignment</td><td><b>Programming Task</b></td><td><b>16 Sessions (L8 + P8)</b></td></tr>
<tr><td colspan="5">Security Management & Privacy Managed Security Service Provider (MSSP): Availability management – configuration management - vulnerability management - identity management. - Privacy: privacy, compliance and the cloud - privacy enhancing encryption.
Risk Management & Security Threats Risk management – principles - assessing the risk – strategies for managing risk – risk analysis framework – security threats - intrusion detection</td></tr>
<tr><td><b>Module 4</b></td><td><b>Cloud Standards and Compliance – CO4</b></td><td>Assignment</td><td><b>Programming Tasks</b></td><td><b>16 Sessions (L8 + P8)</b></td></tr>
<tr><td colspan="5">Cloud Standards and Compliance Cloud security alliance – cloud controls matrix - cloud security standards guidance – security compliance - NIST – PCI data security standards – SAS 70 - ISO 27001 – HIPAA – ITIL - FISMA - FIPS 140- 2. Cloud-Based IT Audit Process – System and Infrastructure lifecycle management for the cloud - governance, risk management and compliance (GRC)</td></tr>
</table>

List of Laboratory Tasks:

Experiment No. 1:

Level 1: Set up a cloud environment using AWS, Azure, or Google Cloud Platform (GCP).

Level 2: Implement basic security configurations, such as firewalls, access controls, and encryption.

Experiment No. 2:

Level 1: Store and retrieve data securely using cloud storage services.

Level 2: Apply encryption techniques like AES, RSA, and homomorphic encryption to secure data at rest and in transit

Experiment No. 3:

Level 1: Install and configure a hypervisor (e.g., VMware, Hyper-V, or KVM).

Level 2: Secure virtual machines (VMs) and ensure isolation between VMs using secure configurations.

Experiment No. 4:

Level 1: **Design a secure virtual network within a cloud environment**

Level 2: **Implement network security measures, including security groups, network access control lists (NACLs), and virtual private clouds (VPCs).**

Experiment No. 5:

Level 1: **Develop a simple web application and deploy it to the cloud**

Level 2: **Identify and mitigate security vulnerabilities using the OWASP Cloud Top 10 Security Risks.**

Experiment No. 6:

Level 1: **Configure IAM policies and roles to manage user access in the cloud**

Level 2: **Implement multi-factor authentication (MFA) and single sign-on (SSO) for secure access.**

Experiment No. 7:

Level 1: **Create and configure a virtual private network (VPN) within the cloud.**

Level 2: **Secure data transmission between on-premises and cloud environments using VPNs.**

Experiment No. 8:

Level 1: **Conduct vulnerability scanning on cloud infrastructure using tools like Nessus or OpenVAS.**

Level 2: **Remediate identified vulnerabilities and enhance security posture.**

Experiment No. 9:

Level 1: **Set up a cloud-based intrusion detection and prevention system (IDPS).**

Level 2: **Analyze and respond to security incidents using IDPS tools.**

Experiment No. 10:

Level 1: **Set up and configure cloud monitoring tools (e.g., AWS CloudWatch, Azure Monitor).**

Level 2: **Analyse logs and generate security reports for cloud environments**

Experiment No. 11:

Level 1: **Set up a cloud environment for IaaS, PaaS, and SaaS models and describe the security responsibilities of cloud providers and customers.**

Level 2: **Implement security measures in an IaaS setup on AWS/Azure/Google Cloud and simulate unauthorized access to verify mitigation.**

Experiment No. 12:

Level 1: Enable data encryption at rest for a file uploaded to cloud storage and verify encryption.
Level 2: Implement encryption in transit using SSL/TLS while transferring data to/from cloud storage.

Experiment No. 13:

Level 1: Set up a Virtual Private Network (VPN) in the cloud and test the secure communication between on-premises and cloud resources.
Level 2: Configure firewall rules and network security groups to restrict access based on IP and test unauthorized access attempts.

Experiment No. 14:

Level 1: Enable logging and monitoring for cloud resources and track activity using cloud-native tools (e.g., AWS CloudTrail, Azure Monitor).
Level 2: Set up an automated alert system for suspicious activities and test it by simulating security incidents.

Targeted Application & Tools that can be used:

AI-driven Intrusion Detection Systems, Fraud Detection Engines, Secure Email Filtering, Cyber Threat Intelligence Platforms using tools such as Python, Scikit-learn, TensorFlow, Keras, OpenCV, Wireshark, Splunk, and Jupyter Notebook.

Project work/Assignment:

1. **AI-Powered Intrusion Detection System**
2. **Phishing Email Detection Using NLP**
3. **Fraudulent Transaction Detector**
4. **Adversarial Attack Simulation**
5. **Anomaly Detection in IoT Devices**

Topics related to

1. Problem Solving: Designing and implementing AI models for real-time cybersecurity threat detection and response.

2. Employability: Simulation of AI-driven intrusion detection systems and fraud detection tools using machine learning and NLP techniques.

Textbook(s):

T1. John R. Vacca, "Cloud Computing Security: Foundations and Challenges", *1st Edition*, **CRC Press**, USA, **2020**. [ISBN-978-0367331656].

T2. Craig S. Wright, "Security Management Practices: A Step-by-Step Guide", 1st Edition, Auerbach Publications, USA, 2021. [ISBN-978-1032092223].

T3: Cloud Security Alliance (CSA), "Cloud Computing Compliance Controls Catalog (C5): A Compendium of Cloud Provider Requirements", 1st Edition, CSA, USA, 2021. [ISBN-978-0989567009].

References

R1. Ronald L. Krutz, Russell Dean Vines, "Cloud Security: A Comprehensive Guide to Secure Cloud Computing", *1st Edition*, Wiley, USA, 2010. [ISBN-978-0470589878

R2. Thomas Erl, "Cloud Computing: Concepts, Technology, Security, and Architecture", *2nd Edition*, Pearson, USA, 2024. [ISBN-978-0138052188].

E-Resources:

6. https://cloudsecurityalliance.org
7. https://owasp.org/www-project-cloud-security
8. https://www.nist.gov/topics/cloud-computing
9. https://www.vmware.com/security.html
10. https://learn.microsoft.com/en-us/security/azure

Course Code: CSA4715	**Course Title: AI in Cyber Security**	**L-T- P- C**	**2**	**0**	**2**	**3**
Version No.	1.0					
Course Pre-requisites	Students should have basic knowledge of **Python programming**, cybersecurity fundamentals, **machine learning**, and **network security**. Familiarity with **ML frameworks (TensorFlow, scikit-learn)**, networking concepts (TCP/IP, IDS/IPS), and **mathematics (statistics, probability, linear algebra)** is recommended. A background in **Computer Science, IT, or Cybersecurity** will be beneficial.					
Anti-requisites	**NIL**					
Course Description	The **AI in Cybersecurity course** explores how artificial intelligence enhances cybersecurity by detecting threats, preventing cyberattacks, and automating security responses. It covers AI-driven malware detection, intrusion detection systems, and secure AI implementation. Students will gain practical experience using AI models for security applications.					

Course Objective	1. To understand the role of AI in modern cybersecurity. 2. To explore AI techniques for threat detection and security automation. 3. To apply machine learning for cybersecurity applications like malware detection and fraud prevention. 4. To implement and evaluate AI-based security solutions.				
Course Out Comes	1. **CO1:** Explain AI applications in cybersecurity and security automation. *(Understand, Describe, Identify)* 2. **CO2:** Implement AI techniques for detecting and preventing cyber threats. *(Apply, Develop, Construct)* 3. **CO3:** Analyze AI-driven cybersecurity models for fraud detection and network security. *(Analyze, Evaluate, Visualize)* 4. **CO4:** Secure AI systems and optimize cybersecurity solutions. *(Secure, Deploy, Optimize)*				
Course Content:					
Module 1	Introduction to AI in Cybersecurity	Assignment	Programming activity		**8 Hours**
Cybersecurity Fundamentals: Overview of cybersecurity concepts, threat landscapes, vulnerabilities, and risk management. Discussion of common attack vectors (e.g., phishing, DDoS, ransomware) and defensive strategies. **AI and Machine Learning Basics:** Introduction to machine learning algorithms (supervised, unsupervised, reinforcement learning) used in security applications.Overview of deep learning, neural networks, and decision trees as applied to cybersecurity tasks.					
Module 2	AI for Threat Detection and Prevention	Assignment	Programming activity		**8Hours**
Intrusion Detection and Prevention Systems (IDS/IPS): Detailed study of IDS/IPS architectures and how AI algorithms improve detection rates, Comparison of signature-based versus anomaly-based detection systems. **Deep Learning for Network Traffic Analysis:** Use of deep neural networks, auto encoders, and recurrent neural networks (RNNs) to model network behaviour, Techniques for feature extraction and dimensionality reduction in large network datasets.					
Module 3	AI for Fraud Detection and Security Automation	Assignment	Programming activity		**8 Hours**
Financial Fraud Detection: Analysis of transactional data using classification and clustering techniques to uncover fraudulent patterns. Study of real-world case studies such as credit card fraud, insurance fraud, and identity theft. **Phishing Detection and Email Security:** Application of					

Natural Language Processing (NLP) and machine learning to analyse email content, URLs, and sender behaviour. Techniques for training models to differentiate between legitimate and phishing emails.

Module 4	Secure AI and Future Trends	Assignment	Programming activity		**6 Hours**

Adversarial Machine Learning: In-depth analysis of adversarial attacks such as evasion attacks, data poisoning, and model inversion. Strategies for designing robust AI models that can resist adversarial inputs. **Securing AI Systems:** Best practices for securing the training, deployment, and maintenance of AI models used in cybersecurity. Techniques for continuous monitoring and updating of AI models to prevent exploitation.

Targeted Application & Tools that can be used: Notepad++, Eclipse IDE, NetBeans IDE

Project work/Assignment: Mention the Type of Project /Assignment proposed for this course 30HRS

1. **Packet Filtering Firewall (Scapy & Netfilterqueue):**

   Leve;-1 Write a Python script to capture all incoming TCP packets and print their source IP addresses and destination ports.

   Level-2 :- Write a Python script to capture all incoming UDP packets and print their source IP addresses, destination IP addresses, and destination ports in real-time.

2. **Phishing URL Detection (Machine Learning):**

   Level-1: Basic Phishing URL Classifier using Decision Tree**.**

   Level-2: Advanced Phishing URL Classifier using Random Forest + Feature Engineering.

3. **Intrusion Detection System (IDS) Using K-Means Clustering:**

   Level-1:- Design an Intrusion Detection System (IDS) that uses K-Means clustering to classify network traffic as either normal or suspicious based on features like packet size, packet rate, and duration. Evaluate the performance of the model by calculating its accuracy and plotting the clustered traffic.

   Level-2:- Develop an anomaly-based IDS using K-Means clustering to detect suspicious network traffic. Implement a method to compute anomaly scores based on the distance from the cluster center, and classify traffic as normal or anomalous using a defined threshold.

4. **Static Malware Detection Using Decision Trees:**

   Level-1 :- Write a Python program that uses a Decision Tree Classifier to classify PE (Portable Executable) files as either malware or benign based on static features such as file size, entropy, and number of imports.

   Level 2:- Develop a more advanced Static Malware Detection System using a Decision Tree Classifier. The system should analyze additional features such as section headers, file metadata, and imports, and classify files based on these features. Implement a function to visualize the decision tree and interpret how specific features contribute to the classification.

5. **Keylogger Simulation for Ethical Hacking Awareness:**

   Level-1:- Write a basic Python program to simulate a keylogger that records keystrokes and saves them to a text file. The program should run in the background and log each keystroke until the user manually stops it.

   Level-2:- Enhance the **keylogger simulation** by adding features for **data encryption** and **stealth operation**. The keylogger should encrypt the keystroke data before saving it to a file and should hide its process from the task manager to prevent detection.

6. **Spam and Phishing Email Detection Using NLP:**

   Level-1: Write a Python program to build a spam email detection model using NLP techniques such as text preprocessing, TF-IDF, and Support Vector Machine (SVM).

   Level-2: Build an advanced spam and phishing email detection system using Natural Language Processing (NLP) and deep learning techniques. The system should classify emails as spam, phishing, or legitimate based on email content and metadata.

7. **Deep Learning for Network Anomaly Detection: Train a Recurrent Neural Network (RNN) :**

   Level-1: Develop a Recurrent Neural Network (RNN) for network anomaly detection using Keras. The model should classify network traffic as normal or anomalous based on features like packet size, protocol type, and duration.

   Level-2: Enhance the network anomaly detection system by integrating autoencoders for unsupervised learning. Use a deep learning model (RNN + Autoencoder) to detect anomalies in network traffic data and reconstruct the traffic patterns.

8. **Autoencoder for Cyber Threat Detection:**

   Level- 1: Implement an Autoencoder model for cyber threat detection in network traffic. The model should be trained on normal network traffic and detect anomalies that could indicate potential cyber threats based on reconstruction errors.

   Level-2: Develop an advanced cyber threat detection system using an Autoencoder with LSTM layers for detecting anomalies in time-series network traffic. The system should effectively distinguish between normal and anomalous traffic patterns and detect sophisticated cyber threats.

9. **Credit Card Fraud Detection Using AI:**

Level-1: - Build a Credit Card Fraud Detection System using XGBoost to classify transactions as fraudulent or legitimate based on features like transaction amount, time of transaction, and user behaviour.

Level-2: - Enhance the Credit Card Fraud Detection System by using ensemble methods (e.g., Random Forests or Stacking) and incorporate time-series analysis for transaction behavior patterns to improve fraud detection accuracy.

10. **AI-Based Password Strength Classifier:**

Level-1 :- Build a Password Strength Classifier using machine learning to classify passwords as weak, medium, or strong based on features like length, entropy, and the presence of special characters.

Level-2 :- Enhance the Password Strength Classifier by using a Neural Network and incorporating additional features such as common words (e.g., dictionary checks) and patterns (e.g., repeating characters) for more robust classification.

11. **Adversarial Attack on an AI Model (FGSM Method):**

Level-1: Implement an Adversarial Attack on a simple image classification model using the Fast Gradient Sign Method (FGSM) to generate adversarial examples and analyze their impact on model predictions.

Level-2: Enhance the FGSM attack by incorporating multiple adversarial examples and defense mechanisms (e.g., adversarial training or gradient masking) to make the model more resilient to adversarial perturbations.

12. **IoT Security - Anomaly Detection in Smart Home Devices:**

Level-1: Develop a basic anomaly detection system for IoT smart home devices using Isolation Forests to detect unusual behaviour based on features like device usage patterns, network traffic, and sensor data.

Level-2 : Enhance the IoT anomaly detection system by incorporating time-series analysis and deep learning techniques (e.g., LSTM Autoencoders) to detect more complex, subtle anomalies in the behaviour of smart home devices.

13. **Secure AI Model Deployment Using Homomorphic Encryption:**

Level-1: Implement Homomorphic Encryption to perform secure predictions using a simple AI model (e.g., Logistic Regression or Decision Tree) on encrypted data without decrypting it, ensuring data privacy.

Level-2: Enhance the AI model's deployment by implementing a more complex system with Homomorphic Encryption for neural networks and performing predictions on encrypted inputs using advanced encryption techniques such as Fully Homomorphic Encryption (FHE).

14. **Blockchain for Secure Cybersecurity Logging:**

Level-1: Implement a simple blockchain-based logging system using Python and Web3.py to securely store cybersecurity logs and ensure immutability, making it resistant to tampering.

Level-2:- Enhance the blockchain-based cybersecurity logging system by adding advanced features such as log verification, timestamping, and access control using smart contracts on a public Ethereum network.

15. Quantum Computing Simulation in Cybersecurity:

Level-1 : Simulate **Shor's Algorithm** using **Qiskit** to demonstrate how quantum computers could potentially break **RSA encryption**, illustrating the difference in computational power between classical and quantum systems.

Level-2 : Implement Quantum Key Distribution (QKD) using Qiskit to demonstrate how quantum communication could provide secure communication channels, immune to eavesdropping and interception, and analyze its implications for future cybersecurity.

Text Book

1. **I. Priyadarshini, R. Sharma, Eds.,** *Artificial Intelligence and Cybersecurity: Advances and Innovations*, Routledge, 2025.
2. **S. Mahajan, M. Khurana, and V. V. Estrela, Eds.,** *Applying Artificial Intelligence in Cybersecurity Analytics and Cyber Threat Detection*, Wiley, 2024.

References

[1] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*, 4th ed., Addison-Wesley, 2021.

[2] C. K. Hargreaves, *Machine Learning for Cybersecurity Cookbook*, Packt Publishing, 2020.

Web References

https://www.coursera.org/specializations/ai-for-cybersecurity?utm_source=chatgpt.com

Topics relevant to "SKILL DEVELOPMENT":

Course Code: **CSA4716**	**Course Title:** Applied AI in Financial Risk & Credit Modelling **Type of Course: PCC – Theory & Practical**	**L-T- P- C**	2	0	2	3
Version No.	1.0					
Course Pre-requisites	-					

<table>
<tr><td><b>Anti-requisites</b></td><td colspan="5"><b>NIL</b></td></tr>
<tr><td><b>Course Description</b></td><td colspan="5">This course explores the application of AI-driven techniques to model financial risk and creditworthiness in institutional and retail domains. It focuses on credit scoring, default prediction, regulatory modeling, and real-time risk stratification using structured and semi-structured datasets. Students will design model pipelines that comply with interpretability and auditability requirements under financial regulation.</td></tr>
<tr><td><b>Course Objective</b></td><td colspan="5">To equip students with the ability to engineer, validate, and deploy AI-powered models for financial risk assessment and credit decision-making in compliance-sensitive environments.</td></tr>
<tr><td><b>Course Outcomes</b></td><td colspan="5">On successful completion of this course the students shall be able to:
1. <b>Build</b> supervised and unsupervised models for credit and risk analytics
2. <b>Engineer</b> features aligned with borrower behavior and economic exposure
3. <b>Evaluate</b> model performance using risk-sensitive metrics
4. <b>Ensure</b> compliance through model explainability and fairness audits</td></tr>
<tr><td><b>Course Content:</b></td><td colspan="5"></td></tr>
<tr><td><b>Module 1:</b></td><td><b>Credit Risk Modeling Fundamentals and Data Structuring</b></td><td><b>Quiz</b></td><td></td><td>Knowledge based Quiz</td><td><b>9 Sessions</b></td></tr>
<tr><td colspan="6"><b>Topics:</b> Borrower-level data pipelines, financial ratios, bureau data parsing, credit event labeling, PD/LGD/EAD concepts
<b>Tools:</b> pandas, numpy, PyJanitor, Faker + JSON for synthetic credit data
<b>Use Cases:</b>
● <b>Extract and standardize loan repayment histories</b>
● <b>Simulate risk labeling using credit bureau-style attributes</b></td></tr>
<tr><td><b>Module 2:</b></td><td><b>Machine Learning Models for Risk and Credit Scoring</b></td><td><b>Quiz</b></td><td></td><td>Knowledge based Quiz</td><td><b>9 Sessions</b></td></tr>
<tr><td colspan="6"><b>Topics:</b> Logistic regression, decision trees, ensemble models, monotonic constraints, class imbalance handling (SMOTE, focal loss)
<b>Tools:</b> scikit-learn, XGBoost, LightGBM, CatBoost, imbalanced-learn
<b>Use Cases:</b>
● Build credit scoring models with monotonic risk features
● Simulate loan approval scenarios under changing default rates</td></tr>
</table>

Module 3:	**Model Calibration, Stability, and Drift Monitoring**	**Quiz**		Knowledge based Quiz	**9 Sessions**
Topics: Calibration curves, population stability index (PSI), binned score validation, segmentation bias detection, concept drift **Tools:** Yellowbrick, Evidently, scikit-plot, calibration.py **Use Cases:** ● Assess drift in credit scoring model after demographic shift ● Generate calibration plots and bin-based GINI analysis					
Module 4:	**Explainability, Fairness, and Regulatory Alignment**	**Quiz**		Knowledge based Quiz	**9 Sessions**
Topics: Local and global explainability, SHAP, feature contribution charts, fairness audits, reject inference, explainable AI in credit decisions **Tools:** SHAP, LIME, Aequitas, Fairlearn, Eli5 **Use Cases:** ● Explain adverse loan decision with SHAP visualization ● Detect gender or region-based bias in credit approvals					
Module 5:	**Capstone – Risk Model Design for a Lending Workflow**	**Quiz**		Knowledge based Quiz	**9 Sessions**
Topics: End-to-end scoring pipeline, staging environments, shadow deployment, compliance documentation, score override protocols **Tools:** FastAPI, Docker, MLflow, Streamlit, Seldon Core (optional) **Use Cases:** ● Capstone: Deploy credit scoring API with explainability interface ● Log and monitor approval overrides by loan officers					
Project work/Assignment: Students will develop a full-stack credit risk model using real or synthetic borrower data. It must include data preprocessing, feature engineering, supervised training, fairness validation, and model explanation. Deliverables will include notebooks, dashboards, regulatory briefs, and an API or dashboard prototype.					
Text Book T1. Co-branded Zell and Paytm Textbook for *Applied AI in Financial Risk & Credit Modelling* .					

References T2. Siddiqi – *Credit Risk Scorecards: Developing and Implementing Intelligent Credit Scoring* T3. scikit-learn documentation – https://scikit-learn.org/ T4. SHAP GitHub – https://github.com/slundberg/shap **Web References** W1. https://docs.evidentlyai.com/ W2. https://www.mlflow.org/
Topics relevant to SKILL DEVELOPMENT: This course prepares students for data science and AI roles in lending, credit risk, and regulatory technology. Students gain hands-on experience in developing audit-ready, bias-tested credit scoring pipelines, integrating explainable AI into lending workflows, and managing lifecycle performance of models in production. Emphasis is placed on compliance, transparency, and operational readiness, making graduates industry-prepared for high-stakes FinTech environments.

Course Code: CSA4717	**Course Title: Smart Contract Engineering & Decentralized Finance (DeFi)** **Type of Course: PCC – Theory & Practical**	**L-T- P- C**	2	0	2	3
Version No.	1.0					
Course Pre-requisites	-					
Anti-requisites	**NIL**					
Course Description	This course provides hands-on experience in designing, implementing, and testing secure smart contracts that power decentralized financial applications. Students will build financial logic using smart contracts and deploy them on permissionless blockchain networks. Topics include lending protocols, liquidity pools, staking systems, yield optimization contracts, and token standards — with emphasis on composability, security, and gas-efficiency.					
Course Objective	To develop technical proficiency in smart contract development for decentralized financial applications, ensuring sound logic, economic stability, and secure deployment using open-source frameworks.					

Course Outcomes	On successful completion of this course the students shall be able to: 1. **Build** smart contracts that encode financial instruments and workflows 2. **Implement** token standards and permissionless liquidity operations 3. **Simulate** financial use cases like lending, swaps, and staking 4. **Secure** contracts against known DeFi attack vectors				
Course Content:					
Module 1:	**Token Contracts and ERC Standards**	**Quiz**		Knowledge based Quiz	**9 Sessions**
Topics: ERC-20/721/1155 standards, mint/burn flows, allowances, reentrancy surfaces, modular token libraries **Tools:** Solidity, OpenZeppelin Contracts, Hardhat, Ganache **Use Cases:** ● **Create ERC-20 token with inflation and governance hooks** ● **Deploy NFT collection with metadata stored on IPFS**					
Module 2:	**B Liquidity Pools and AMM Design**	**Quiz**		Knowledge based Quiz	**9 Sessions**
Topics: Constant product formula (x*y=k), slippage, LP tokens, impermanent loss, swap routing, fee design **Tools:** Uniswap V2 (open-core), Foundry, EVM Tracing Tools, Graph Protocol **Use Cases:** ● Simulate swaps on constant product AMM using custom contracts ● Model LP returns under variable volatility					
Module 3:	**Lending Protocols and Risk Logic**	**Quiz**		Knowledge based Quiz	**9 Sessions**
Topics: Collateralization ratios, liquidation, interest models, time-based state changes, over-collateralization **Tools:** Compound V2 open contracts, Chainlink Price Feeds, Anvil (Foundry) **Use Cases:** ● Build borrow contract with time-sensitive liquidation logic ● Track collateral health using mock oracle price feeds					

Module 4:	**DeFi Contract Security and Upgradeability**	**Quiz**		Knowledge based Quiz	**9 Sessions**

Topics: Checks-effects-interactions, reentrancy guards, proxy patterns, privilege control, pausability

Tools: OpenZeppelin Defender, Slither, Mythril, Sourcify

Use Cases:

- Implement upgradeable contract for yield vault
- Run static analysis and fix insecure lending logic

Module 5:	**Capstone – Deploying a Multi-Contract DeFi System**	**Quiz**		Knowledge based Quiz	**9 Sessions**

Topics: Modular system design, testnet deployment, indexer integration, public dashboard connection, liquidity testing

Tools: The Graph (subgraph studio), Hardhat, IPFS, RainbowKit for frontend

Use Cases:

- Deploy a modular DApp combining token, pool, and loan contracts
- Create subgraph for dashboard analytics of TVL and APY

Project work/Assignment:

Students will design and deploy a small-scale DeFi protocol including a token, lending or staking logic, and user dashboard integration. Deliverables include contract source code, testnet addresses, event tracking subgraph, and documentation of economic parameters and security logic.

Text Book

T1. Co-branded Zell and Paytm Textbook for *Smart Contract Engineering & Decentralized Finance (DeFi)*

References

T2. Solidity Docs – https://docs.soliditylang.org/

T3. OpenZeppelin Contracts – https://docs.openzeppelin.com/contracts

T4. Compound Docs – https://docs.compound.finance/v2/

Web References

W1. https://defillama.com – DeFi Analytics and Protocol Comparison

W2. https://dune.com – Community-powered Blockchain Data Dashboarding

Topics relevant to SKILL DEVELOPMENT:

This course builds core competencies for blockchain engineers working in the DeFi space. Students gain the ability to model lending, staking, and liquidity logic using smart contracts; secure and optimize code using industry tools; and deploy and monitor modular financial apps. These capabilities directly support roles in DeFi engineering, protocol operations, and smart contract development across the Web3 ecosystem.

Course Code: CSA4718	**Course Title: Advanced FinTech Cybersecurity & Compliance Architecture** **Type of Course: PCC – Theory & Practical**	**L-T- P-C**	2	0	2	3
Version No.	1.0					
Course Pre-requisites	-					
Anti-requisites	**NIL**					
Course Description	This course explores the design and implementation of cybersecurity architectures tailored for modern financial systems and decentralized platforms. Students will work on integrating multi-layer security, secure API design, credential management, anomaly detection, and compliance-aware infrastructure in both centralized and Web3 environments. The course also includes cryptographic application in FinTech, secure transaction design, and real-time breach response planning aligned with RBI, GDPR, and PCI-DSS standards.					
Course Objective	To train students in building scalable, compliant, and secure FinTech system architectures that withstand evolving attack vectors and meet real-time regulatory requirements.					
Course Outcomes	On successful completion of this course the students shall be able to: 1. **Design** financial architectures with layered security and encryption 2. **Detect** threats and mitigate attacks in transaction processing systems 3. **Implement** real-time monitoring and incident response strategies 4. **Build** compliant infrastructure aligned with global regulations					
Course Content:						
Module 1:	**Threat Surfaces and Secure Architecture in FinTech**	**Quiz**		Knowledge based Quiz	**9 Sessions**	
Topics: API-level attack vectors, third-party risk, cloud-native threats, secure service segmentation, FinTech-specific threat modelling						

Tools: OWASP Threat Dragon, OpenVAS, OSQuery, Wazuh
Use Cases:

- Model threat surface of a digital wallet ecosystem
- Simulate API abuse in a loan management system

Module 2:	**Encryption, Access Control, and Secure Interfaces**	**Quiz**		Knowledge based Quiz	**9 Sessions**

Topics: TLS/HTTPS, mutual TLS, OAuth2, HMACs, JWT, end-to-end encryption, certificate pinning, KMS
Tools: Certbot, Keycloak, HashiCorp Vault, OpenSSL

Use Cases:

- Configure OAuth2 flows for user-permissioned UPI app
- Manage secret rotation for transaction signing service

Module 3:	**Real-Time Monitoring and Breach Response**	**Quiz**		Knowledge based Quiz	**9 Sessions**

Topics: SIEM systems, logging, anomaly detection, incident response plans, kill chains, alert prioritization
Tools: Wazuh, Graylog, Suricata, ELK Stack (open version)

Use Cases:

- Detect abnormal login patterns in investment platform
- Configure alerts for fraud behavior on savings app backend

Module 4:	**Regulatory Compliance and Data Security Engineering**	**Quiz**		Knowledge based Quiz	**9 Sessions**

Topics: RBI circulars, GDPR financial data provisions, PCI-DSS tokenization, data retention, consent trails, audit APIs

Tools: Apache Ranger, Airbyte (for data lineage), Open Banking API Specs, OpenRegulatory Templates

Use Cases:

- Enforce access policy on transaction dataset with audit log
- Map compliance flows for a BNPL (Buy Now Pay Later) service

Module 5:	**Capstone – Secure Financial Stack and Compliance Mapping**	**Quiz**		Knowledge based Quiz	**9 Sessions**

Topics: Integrated system design, DevSecOps automation, secure APIs, modular IAM, simulated compliance documentation

Tools: Docker + Trivy, OWASP ZAP, Terraform (Open), OpenAPI Spec tools

Use Cases:

- Build microservice-based financial backend with integrated security scanning
- Document and simulate a compliance walkthrough for audit readiness

Project work/Assignment:

Students will create a security-first FinTech system architecture that includes secure transaction APIs, access management layers, encryption setup, and real-time monitoring hooks. Deliverables include threat model, architectural diagram, configuration snippets, regulatory mapping, and incident response plan simulation.

-

Text Book

T1. Co-branded Zell and Paytm Textbook for *Advanced FinTech Cybersecurity & Compliance Architecture*

References

T2. OWASP API Security Project – https://owasp.org/www-project-api-security/

T3. GDPR Official Portal – https://gdpr-info.eu/

T4. RBI Circulars Archive – https://rbi.org.in/scripts/NotificationUser.aspx

Web References

W1. https://www.elastic.co/what-is/elk-stack
W2. https://github.com/hashicorp/vault

Topics relevant to SKILL DEVELOPMENT:

This course builds advanced capabilities in secure software architecture, breach prevention, and compliance integration for financial technology systems. Students acquire operational skills in managing real-world API security, enforcing encryption standards, architecting compliant backend infrastructure, and preparing for high-stakes audit and threat scenarios. The module prepares learners for cybersecurity and DevSecOps roles in regulated digital finance environments.

Course Code: CSA4719	**Course Title: Portfolio Optimization & Robo-Advisory Algorithms**	**L-T- P-C**	2	0	2	3
Version No.	1.0					

<table>
<tr><td><b>Course Pre-requisites</b></td><td colspan="5">-</td></tr>
<tr><td><b>Anti-requisites</b></td><td colspan="5"><b>NIL</b></td></tr>
<tr><td><b>Course Description</b></td><td colspan="5">This course equips students with the tools and techniques used in algorithmic portfolio construction, optimization, and automation through robo-advisory systems. It covers mathematical optimization techniques, risk-reward trade-offs, portfolio rebalancing strategies, and rule-based automation. Emphasis is placed on data integration, model explainability, compliance checks, and building deployable decision engines for digital investment platforms.</td></tr>
<tr><td><b>Course Objective</b></td><td colspan="5">To train students in designing optimization-driven and automated portfolio advisory systems aligned with client objectives, market conditions, and risk profiles using open-source frameworks.</td></tr>
<tr><td><b>Course Outcomes</b></td><td colspan="5">On successful completion of this course the students shall be able to:
1. <b>Build</b> quantitative portfolios using constraint-aware optimization techniques
2. <b>Evaluate</b> portfolios based on risk-return metrics and performance attribution
3. <b>Design</b> rebalancing, allocation, and glide-path strategies for robo-advisory workflows
4. <b>Integrate</b> investor profiling and suitability logic</td></tr>
<tr><td><b>Course Content:</b></td><td colspan="5"></td></tr>
<tr><td><b>Module 1:</b></td><td><b>Portfolio Theory and Constraint-Aware Optimization</b></td><td><b>Quiz</b></td><td></td><td>Knowledge based Quiz</td><td><b>9 Sessions</b></td></tr>
<tr><td colspan="6"><b>Topics:</b> Mean-variance optimization, Sharpe ratio, CVaR, risk parity, sector constraints, turnover minimization
<b>Tools:</b> PyPortfolioOpt, cvxpy, numpy, pandas, QuantLib
<b>Use Cases:</b>
● Construct a portfolio maximizing Sharpe ratio with sector constraints
● Simulate capital allocation under volatility ceiling</td></tr>
<tr><td><b>Module 2:</b></td><td><b>Factor Modeling and Smart Beta Strategy Design</b></td><td><b>Quiz</b></td><td></td><td>Knowledge based Quiz</td><td><b>9 Sessions</b></td></tr>
<tr><td colspan="6"><b>Topics:</b> Single and multi-factor models, Fama-French factors, alpha capture, risk attribution, factor exposure tuning
<b>Tools:</b> statsmodels, FactorLib, Alphalens, PyFolio</td></tr>
</table>

Use Cases: ● Analyze a smart-beta strategy targeting value and momentum ● Decompose portfolio risk using factor contribution					
Module 3:	**Rebalancing, Execution Models, and Transaction Cost Controls**	**Quiz**		Knowledge based Quiz	**9 Sessions**
Topics: Threshold-based rebalancing, tax-efficient harvesting, slippage simulation, order slicing, impact modelling **Tools:** ● **Backtrader**, **bt**, **Zipline**, **Empyrial**, **Quantstats** **Use Cases:** ● Automate quarterly rebalance minimizing transaction costs ● Simulate order slicing for illiquid small-cap execution					
Module 4:	**Robo-Advisory Decision Engines and Personalization**	**Quiz**		Knowledge based Quiz	**9 Sessions**
Topics: Investor risk profiling, suitability matrix, goal-based investing, glide-path allocation, age-risk automation **Streamlit**, **Flask**, **Scikit-learn (for clustering)**, **OpenAPI specs** **Use Cases:** ● Build investor profile using questionnaire + data clustering ● Generate a goal-based portfolio for retirement planning					
Module 5:	**Capstone – Deploying a Portfolio Recommendation System**	**Quiz**		Knowledge based Quiz	**9 Sessions**
Topics: Integration with brokerage APIs, compliance rules, rebalancing logic, dashboard generation, alert triggers **Tools:** FastAPI, YFinance, Dash, Jupyter, Plotly **Use Cases:** ● Capstone: Build end-to-end portfolio optimizer and deploy via API ● Visualize risk-return metrics and rebalance timeline					
Project work/Assignment: Students will create a personalized robo-advisory platform that uses optimization logic to generate portfolio recommendations. It must include profiling logic, risk-adjusted asset					

selection, rebalancing strategies, and deployable interface or API. Deliverables include source code, UI prototype, optimization backtests, and client recommendation reports.
Text Book T1. Co-branded Zell and Paytm Textbook for *Portfolio Optimization & Robo-Advisory Algorithms*
References T2. Marcos López de Prado – *Advances in Financial Machine Learning* T3. Open Source Portfolio Optimization Library – https://github.com/robertmartin8/PyPortfolioOpt **Web References** W1. https://quantlib.org/ W2. https://fastapi.tiangolo.com/
Topics relevant to SKILL DEVELOPMENT: This course prepares students for technical roles in digital wealth platforms, investment advisory technology, and quant-driven FinTech. It emphasizes constraint-based portfolio construction, real-world execution modeling, and automation for client-centric investment workflows. Students develop practical fluency in portfolio logic, backtesting, risk attribution, and deploying scalable APIs for financial decision systems

Course Code: CSA4720	**Course Title: RegTech Systems, Fraud Analytics & Compliance Automation**	**L-T- P- C**	2	0	2	3
Version No.	1.0					
Course Pre-requisites	-					
Anti-requisites	**NIL**					
Course Description	This course focuses on the design and automation of regulatory technology (RegTech) solutions for fraud detection, AML/KYC verification, and real-time compliance tracking in digital finance. Students will build pipelines for suspicious activity monitoring, rule-based enforcement, transaction anomaly detection, and automated audit log generation aligned with RBI, FATF, SEBI, and global frameworks. Emphasis is placed on auditability, real-time alerts, explainability, and compliance reporting workflows.					
Course Objective	To train students in designing regulatory automation systems that operate across high-volume financial systems while satisfying modern compliance and fraud control mandates.					

Course Outcomes	On successful completion of this course the students shall be able to: **5. Build** fraud detection and compliance rules using structured datasets **6. Integrate** real-time anomaly detection engines into transaction systems **7. Automate** KYC validation, sanctions screening, and SAR/STR filing **8. Ensure** full auditability and regulatory traceability in financial systems				
Course Content:					
Module 1:	**Regulatory Landscapes and Digital Rule Automation**	**Quiz**		Knowledge based Quiz	**9 Sessions**
Topics: FATF guidelines, RBI AML/KYC rules, SEBI mandates, RegTech taxonomy, rule-based automation engines, audit triggers **Tools: Drools (open-source rule engine)**, **Apache NiFi**, **OpenRegTech Templates** **Use Cases:** ● Automate customer classification under RBI KYC framework ● Create rule engine to flag transactions violating SEBI margin rules					
Module 2:	**Fraud Pattern Detection and Anomaly Signals**	**Quiz**		Knowledge based Quiz	**9 Sessions**
Topics: Behavioral anomaly detection, peer group analysis, entity resolution, synthetic identity fraud, velocity checks **Tools:** pandas, scikit-learn, PyOD, Jupyter, Dask **Use Cases:** ● Detect high-frequency low-value attack patterns ● Flag location-inconsistent spending for prepaid cards					
Module 3:	**Sanctions Screening and Customer Risk Scoring**	**Quiz**		Knowledge based Quiz	**9 Sessions**
Topics: Watchlist matching, fuzzy name matching, Politically Exposed Persons (PEPs), customer risk tiers, false positive suppression **Tools:** FuzzyWuzzy, Whoosh, OpenRefine, AML-CFT open data sources **Use Cases:**					

- Match customer list against OFAC or UN watchlists
- Score customers based on origin, transaction pattern, and exposure

Module 4:	**Auditability, Explainability, and Regulatory Interfaces**	**Quiz**		Knowledge based Quiz	**9 Sessions**

Topics: Immutable logs, event reconstruction, evidence trail generation, explainable flags, complaint workflows, audit APIs
Tools: ELK Stack, OpenMetadata, Marquez, Great Expectations

Use Cases:

- Generate full evidence trail for frozen account
- Produce automated compliance report from transaction log events

Module 5:	**Capstone – RegTech System Simulation for a Digital Bank**	**Quiz**		Knowledge based Quiz	**9 Sessions**

Topics: Rule orchestration, fraud rules vs compliance triggers, reactive vs proactive monitoring, dashboard deployment
Tools: Apache Superset, Streamlit, Airflow, SQLMesh, FastAPI

Use Cases:

- Capstone: Simulate end-to-end RegTech stack for a mock neobank
- Combine rules, ML triggers, and explainability layer

Project work/Assignment:
Students will develop a modular RegTech system capable of ingesting transaction data, running fraud/compliance rules, scoring risks, logging audit trails, and outputting alert and report dashboards. Deliverables include data pipeline code, live or mock dashboards, logs, alert traces, and a mock audit summary for internal review.

Text Book

T1. T1. Co-branded Zell and Paytm Textbook for *RegTech Systems, Fraud Analytics & Compliance Automation*

References

T2. FATF Guidelines – https://www.fatf-gafi.org/publications/

T3. RBI Master Directions on KYC – https://www.rbi.org.in/Scripts/BS_ViewMasDirections.aspx

Web References

W1. https://superset.apache.org/

W2. https://www.openmetadata.org/

Topics relevant to SKILL DEVELOPMENT:

This course develops hands-on skills in regulatory automation and digital fraud analytics. Students learn to engineer compliant transaction systems, build explainable anomaly detection engines, and develop modular RegTech workflows for real-time governance. The course prepares learners for technical roles in FinTech regulatory ops, compliance tech architecture, and fraud analytics engineering.