VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

CURRICULUM AND SYLLABI

(2025-2026)

M. Tech Computer Science and Engineering

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

M. Tech Computer Science and Engineering

VISION STATEMENT OF VELLORE INSTITUTE OF TECHNOLOGY

Transforming life through excellence in education and research.

MISSION STATEMENT OF VELLORE INSTITUTE OFTECHNOLOGY

World class Education: Excellence in education, grounded in ethics andcritical thinking, for improvement of life.

Cutting edge Research: An innovation ecosystem to extend knowledge andsolve critical problems.

Impactful People: Happy, accountable, caring and effective workforce andstudents.

Rewarding Co-creations: Active collaboration with national & international industries & universities for productivity and economic development.

Service to Society: Service to the region and world through knowledge andcompassion.

VISION STATEMENT OF THE SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

To be a world-renowned centre of education, research and service in computing andallied domains.

MISSION STATEMENT OF THE SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

- To offer computing education programs with the goal that the students become technically competent and develop lifelong learning skill.
- To undertake path-breaking research that creates new computing technologies and solutions for industry and society at large.
- To foster vibrant outreach programs for industry, research organizations, academiaand society.

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

M. Tech Computer Science and Engineering

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

PEO 1: Graduates will acquire a core competency in technology development and deployment with social awareness and responsibility.

PEO 2: Graduates will pursue career as successful engineers, researchers or entrepreneurs by developing sustainable solutions across diverse industry sectors.

PEO 3: Graduates will develop the ability to work in teams, uphold ethical standards, and communicate effectively to succeed in their professional careers.

PEO 4: Graduates will adopt a holistic approach to address the challenges in technology development and evolving market trends.

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

M. Tech Computer Science and Engineering

PROGRAMME OUTCOMES (POs)

PO1: Independently carry out research/investigation and development work to solve practical problems

PO2: Write and present a substantial technical report/document

PO3: Demonstrate a degree of mastery in designing and developing industry ready components or products applying all the relevant standards with realistic constraints

PO4: Analyse the complex engineering problems and design sustainable solutions

PO5: Communicate effectively at the work place and use engineering tools and techniques with ethical considerations.

PO6: Recognize the need for independent and life-long learning in the broadest context of evolving technology.

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

M. Tech Computer Science and Engineering

CURRICULAM AND SYLLABUS
2025-2026

Category Credit Detail			
Sl.No.	**Description**	**Credits**	**Maximum Credit**
1	UCC - University Core Courses	39	39
2	OEC - Open Elective Courses	3	3
3	PFCC - Professional Core Courses	24	24
4	PFEC - Professional Elective Courses	14	14
Total Credits		80	

University Core Courses

sl.no	Course Code	Course Title	Course Type	Version	L	T	P	J	Credits
1	MACSE698	Internship I/ Dissertation I	Project	1.0	0	0	0	0	10.0
2	MACSE699	Internship II/ Dissertation II	Project	1.0	0	0	0	0	10.0
3	MAENG501	Technical Report Writing	Embedded Theory and Lab	1.0	1	0	4	0	3.0
4	MASET697	Project Work	Project	1.0	0	0	0	0	10.0
5	MASTS601	Competitive Coding I	Soft Skill	1.0	3	0	0	0	3.0
6	MASTS602	Competitive Coding II	Soft Skill	1.0	3	0	0	0	3.0

Open Elective Courses

sl.no	Course Code	Course Title	Course Type	Version	L	T	P	J	Credits
1	MASTS501	Qualitative and Quantitative Skills Practice I	Soft Skill	1.0	3	0	0	0	3.0

Professional Core Courses

sl.no	Course Code	Course Title	Course Type	Version	L	T	P	J	Credits
1	MACSE501	Data Structures and Algorithm Analysis	Embedded Theory and Lab	1.0	3	0	2	0	4.0

2	MACSE512	Operating Systems	Embedded Theory and Lab	1.0	3	0	2	0	4.0
3	MACSE513	Computer Networks	Embedded Theory and Lab	1.0	3	0	2	0	4.0
4	MACSE517	Computer Architecture and Organization	Embedded Theory and Lab	1.0	3	0	2	0	4.0
5	MACSE518	Database Modelling and Design	Embedded Theory and Lab	1.0	3	0	2	0	4.0
6	MACSE519	Machine Learning and Applications	Embedded Theory and Lab	1.0	3	0	2	0	4.0

Professional Elective Courses									
sl.no	**Course Code**	**Course Title**	**Course Type**	**Version**	**L**	**T**	**P**	**J**	**Credits**
1	MACSE610	Image and Video Analytics	Embedded Theory and Lab	1.0	3	0	2	0	4.0
2	MACSE635	Blockchain Technology and Applications	Embedded Theory and Lab	1.0	3	0	2	0	4.0
3	MACSE636	Cybersecurity	Theory Only	1.0	3	0	0	0	3.0
4	MACSE637	Digital Forensics	Theory Only	1.0	3	0	0	0	3.0
5	MACSE638	Software Engineering and Modelling	Theory Only	1.0	3	0	0	0	3.0
6	MACSE639	DevOps	Embedded Theory and Lab	1.0	3	0	2	0	4.0
7	MACSE640	Full Stack Application and Development	Embedded Theory and Lab	1.0	3	0	2	0	4.0

Course Code	Course Title	L	T	P	C
MACSE698	Internship I/ Dissertation I	0	0	20	10
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

To provide sufficient hands-on learning experience in the design, development, and analysis of appropriate products or processes.
To enhance technical skill sets in the student's chosen field of study.
To foster a research-oriented mindset through practical engagement.

Course Outcomes

Gain in-depth expertise in the major subject or field of study, including comprehensive understanding and critical insight into ongoing research and development trends.
Develop the ability to approach complex problems holistically, enabling critical, independent, and creative thinking in identifying, formulating, and addressing multifaceted challenges.
Cultivate a strong awareness of the ethical dimensions associated with research and development activities, promoting integrity and responsible conduct in scholarly and professional work.
Enhance research communication and dissemination skills, including the ability to effectively present findings through technical writing, oral presentations, and scholarly discussions.
Achieve recognition through publications in peer-reviewed journals and international conferences, which serve as valuable indicators of research quality and impact.

General Guidelines

(PROJECT DURATION - One Semester): - The dissertation may encompass a wide range of scholarly activities, including theoretical analysis, modeling and simulation, experimental investigation, prototype design, equipment fabrication, data correlation and analysis, software development, or applied research. It is expected to be an individual effort, demonstrating the student's independent capability in their area of specialization. The work may be conducted either within the university or externally, such as in a relevant industry or research institution. While not mandatory, publications resulting from the dissertation in peer-reviewed journals or international conferences will be considered a valuable addition, reflecting the quality and impact of the research undertaken.

Mode of Evaluation : Project Reviews, Project Reports, VivaVoce/Student Interactions

Recommended by Board of Studies :	02-06-2025
Approved by Academic Council : No. 78	12-06-2025

Course Code	Course Title	L	T	P	C
MACSE699	Internship II/ Dissertation II	0	0	20	10
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

To provide hands-on learning experience in the design of appropriate products or processes.
To develop practical skills related to the development and analysis of engineering solutions.
To enhance technical competencies in the student's chosen field through experiential learning.

Course Outcomes

Formulate well-defined problem statements for complex, real-world scenarios by applying appropriate assumptions and constraints.
Conduct comprehensive literature and patent searches to establish a strong foundation and context for the identified problem.
Design and implement experimental or analytical approaches, including iterative solution development, and systematically document findings.
Perform critical evaluations such as error analysis, bench marking, and cost estimation to assess the feasibility and efficiency of proposed solutions.
Synthesize results to derive meaningful scientific conclusions or practical solutions, and effectively communicate the outcomes through technical reports and presentations.

General Guidelines

(PROJECT DURATION - One Semester): - The dissertation may encompass a wide range of scholarly activities, including theoretical analysis, modeling and simulation, experimental investigation, prototype design, equipment fabrication, data correlation and analysis, software development, or applied research. It is expected to be an individual effort, demonstrating the student's independent capability in their area of specialization. The work may be conducted either within the university or externally, such as in a relevant industry or research institution. While not mandatory, publications resulting from the dissertation in peer-reviewed journals or international conferences will be considered a valuable addition, reflecting the quality and impact of the research undertaken.

Mode of Evaluation : Project Reviews, Project Reports, VivaVoce/Student Interactions

Recommended by Board of Studies :	02-06-2025
Approved by Academic Council : No. 78	12-06-2025

Course Code	Course Title	L	T	P	C
MAENG501	Technical Report Writing	1	0	4	3
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

To develop communicative competence in students.
To apply advanced technical communication principles in practical writing tasks for diverse professional contexts.
To execute research and analytical skills to produce coherent and evidence-based technical documents through practical exercises.

Course Outcomes

Apply the principles of effective report writing.
Demonstrate the ability to draft and present technical reports.
Produce key components of technical documents, including sections focusing on purpose, audience, and structure, through writing assignments.
Create technical summaries and abstracts by practicing techniques for summarizing and paraphrasing complex information.

Module:1	**Fundamentals of Technical Writing**	**3 hours**
Introduction to Technical Writing: Definition and typical forms (reports, instructions, proposals); Key Factors in Technical Writing: Purpose, Audience, and Tone; General Writing Basics: Clarity, fluency, effectiveness; The Process of Writing: Pre-writing, Writing, and Post-writing stages; Organization in Writing: Having an outline, using introductions, headings, lists, figures, and summaries.		
Module:2	**Technical Grammar**	**2 hours**
Concord; Tense Shifts		
Module:3	**Introduction to Reports and Report Structure Front Matter**	**2 hours**
Reports in Organizations: Role and importance of reporting in corporate and industrial segments.; Purpose of Reports: Conveying decisions, facts, and information accurately and up-to-date.; Report Structure Overview: Division into Front Matter, Main Body, and Back Matter; Components of the Front Matter		
Module:4	**Report Structure Main Body and Back Matter**	**4 hours**

Components of the Main Body: Introduction, Discussion or Description, Conclusions, and Recommendations; Writing Introduction, Discussion or Description, Conclusions and Recommendations, Methods of Reporting: The Letter Method and The Schematic Method; Routine Reports: Nature, frequency, and function in organizations (often statistical, fixed intervals).		
Module:5	**Technical Proposals**	**2 hours**
Punctuation Right words and phrases; avoiding cliches, jargons, foreign words and phrases, ambiguity, redundancy, circumlocution - Developing hints		
Module:6	**Contemporary Issues**	**2 hours**
Guest Lecture		
	Total Lecture Hours:	**15 hours**

Text Book(s)

Kumar. S & Pushplata. , "**Effective Communication Skills.** ", New Delhi OUP, **2018**
Muralikrishna and Sunita Mishra, "**Communication Skills For Engineers**", Pearson, 2nd Edition, **2011**
Shirley Mathew., "**Effective Communication Skills.**", Nirali Prakashan, **2025**

Reference Books

Indicative Experiments

1. **Introduction to the Technical Writing** Introduction to Technical Report Writing**;** Analyzing and identifying the characteristics of effective technical writing in sample documents.; Short writing exercises focusing on clarity, conciseness, and identifying purpose, audience, and tone in simple technical scenarios. **Activity:** Reviewing and providing feedback on short technical descriptions.	2 hours
2. **Analyzing Audience and Context for Practical Writing** Applying reader and stakeholder analysis techniques to specific engineering report scenarios; Drafting content segments tailored for different technical and non-technical audiences (e.g., writing an executive summary for managers vs. a technical description for fellow engineers); Simulating audience needs assessment based on given project descriptions. **Activity:** Developing a 'reader profile' for a major report project.	4 hours
3. **Technical Grammar and Style in Practice**	2 hours

Intensive practical exercises on complex grammatical structures, sentence syntax, and common errors in technical writing; Exercises in applying principles of clarity, conciseness, and precision to improve technical sentences and paragraphs; Practicing the appropriate use of active and passive voice in different report sections. **Activity:** Peer-editing session focusing on grammar, mechanics of writing, and technical style.	
4. **The Writing Process and Report Outlining Lab** Practicing prewriting techniques for complex technical topics: brainstorming, mind mapping, and systematic outlining; Developing detailed hierarchical outlines for a major technical report project, including main and sub-points; Planning content organization based on report type (e.g., feasibility, empirical research, lab report) and audience needs. **Activity:** Group exercise to compare and refine report outlines.	4 hours
5. **Writing the Report Front Matter and Introduction** Detailed practical guidance on drafting all components of the report Front Matter; Creating a professional Title Page, Forwarding Letter/Preface, Acknowledgements, and Table of Contents for the ongoing report project; Drafting effective Abstracts and Summaries (Executive Summaries) based on provided technical content, focusing on capturing the essence. **Activity:** Writing the Introduction section of the report project, including background, scope, and objectives.	4 hours
6. **Writing the Report Body: Data, Descriptions, and Discussion** Techniques for presenting data, technical descriptions, and analysis in the main body; Drafting sections of the report body focusing on presenting organized data and technical details; Writing the "Discussion" section, focusing on interpreting results and explaining findings based on provided or self-generated data. **Activity:** Integrating data and analysis points into the draft of the report body.	4 hours
7. **Writing the Report Body: Conclusions and Recommendations** Practical methods for drawing logical conclusions and formulating actionable recommendations; Drafting the "Conclusions" section based on the data and	4 hours

discussion from Module 6; Developing clear and practical recommendations based on the conclusions, considering the report's purpose and audience. **Activity:** Writing the "Recommendations" section of the report project.	
8. **Transcribing Visuals** Using charts, graphs and tables; Transcribing visuals that are clear, accurate, and effectively support the report's text; Integrating created visuals into the report draft, ensuring proper placement, captions, and referencing within the text. **Activity:** Peer review focusing on the effectiveness and integration of visuals.	2 hours
9. **Report Back Matter, Condensation, and Final Review** Practical session on creating the Back Matter: Appendices, Bibliography, Glossary, and Index (if applicable). Emphasis on consistent citation and referencing styles; Compiling Appendices and formatting a Bibliography for the report project; Practicing techniques for précis writing and summarization to condense longer texts. **Activity:** Final review of the complete report draft, focusing on overall structure, flow, formatting, and coherence.	2 hours
10. **Presentation Skills** Short Presentations, Formal Presentation with PPT Analytical Presentation of Charts, Graphs and Tables **Activity**: Presentations – Individual and Group	2 hours
Total Laboratory Hours:	**60 hours**
Mode of Evaluation : Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment, Presentaion	
Recommended by Board of Studies :	16-05-2025
Approved by Academic Council : No. 78	12-06-2025

Course Code	Course Title	L	T	P	C
MASTS601	Competitive Coding I	3	0	0	3
Pre-requisite	NIL	Syllabus version			
		1.0			

Course Objectives

Equip learners with an in-depth understanding of linear data structures (linked lists, stacks, queues, deques, etc.). Enabling them to design, implement, and optimize these structures for various problem-solving scenarios. Emphasis is placed on both theory (complexities, operations) and hands-on coding to master data structure manipulation and usage.

Course Outcomes

By the end of this course, participants will be able to:

● Confidently implement and utilize linked lists, stacks, queues, and deques

● Analyse and optimize the time/space complexities of linear data structure operations (insertion, deletion, traversal)

● Apply problem-solving strategies to real-world tasks and competitive coding challenges, leveraging linear data structures effectively

● Integrate debugging, testing, and best coding practices when working with linear data structures

Module:1	Linked List-1	6 hours

Introduction to Linked Lists: Definition, structure, advantages; Types of Linked Lists: Singly, Doubly, Circular.

Basic Operations: Creating a linked list; Insertion (at beginning, middle, end); Deletion (from beginning, middle, end); Traversal (iterative, recursive).

Module:2	Linked List-2	6 hours

Advanced Topics: Reversing a linked list (iterative, recursive); Detecting and removing loops; Merging and splitting linked lists; Intersection of two linked lists.

Module:3	Stack-1	5 hours

Introduction to Stack: Definition, LIFO property, real-world applications (e.g., undo functionality, browser history); Static vs dynamic stacks.

Operations on Stack: Push, pop, peek; Checking stack overflow and underflow.

Implementation: Using arrays; Using linked lists.

Module:4	Stack-2	4 hours

Applications of Stack: Balancing parentheses in expressions; Converting infix expressions to postfix; Evaluating postfix expressions; Solving backtracking problems (e.g., maze solving).

Module:5	Queue	7 hours

Introduction to Queue: Definition, FIFO property; Types of Queues: Simple, Circular, Double-Ended.

Operations on Queue: Enqueue (insertion), Dequeue (removal), Peek (front element); Checking queue overflow and underflow.

Implementation: Using arrays; Using linked lists.

Applications of Queue: Task scheduling (e.g., CPU scheduling, print queue); Breadth-First Search (BFS) implementation; Managing resources in real-time systems (e.g., job queues).

Module:6	Priority Queue	5 hours

Introduction to Priority Queue: Definition, importance in task prioritization; Differences between Priority Queue and Simple Queue.

Operations on Priority Queue: Enqueue and Dequeue operations based on priority.

Implementation: Using arrays; Using heaps (Binary Heaps: Min-Heap and Max-Heap).

Module:7	Hash map	9 hours
Introduction to Hash map: Definition, key-value pair structure, advantages over arrays. Hash Functions and Collision Handling: Direct addressing, simple hash functions, modulo operation; Collision resolution techniques: Chaining, Linear Probing, Quadratic Probing. Applications of Hash map: Frequency counting (e.g., counting occurrences of elements); Caching mechanisms (e.g., implementing LRU cache); Finding duplicates in arrays or strings; Solving anagram problems; Building indexes for fast lookup.		
Module:8	**Interview Prep - Networking**	**3 hours**
Networking Fundamentals: OSI model TCP/IP basics Network Protocols: HTTP/HTTPS DNS and DHCP FTP and SMTP - Practical Interview Questions: - Debugging network issues, Firewall, DNS, Ping and Trace route - Explaining client-server architecture - Designing scalable systems		
Total Lecture hours:		**45 hours**

Text Book	
1.	A Textbook of Data Structures and Algorithms 1: Mastering Linear Data Structures – 1st Edition by G A Vijayalakshmi Pai – Wiley Publication
Reference Books	
1.	Data Structure and Algorithms Made Easy by Narasimha Karumanchi -5th edition

Mode of Evaluation: Written assignment, Quiz, Project & FAT.			
Recommended by Board of Studies			
Approved by Academic Council		Date	

Course Code	Course Title	L	T	P	C
MASTS501	Qualitative and Quantitative Skills Practice I	3	0	0	3
Pre-requisite	NIL	Syllabus version			

Course Objectives:

1. To enhance the logical reasoning skills of students and improve problem-solving abilities
2. To strengthen the ability of solving quantitative aptitude problems
3. To enrich the verbal ability of the students for academic purposes

Course Outcomes:

1. Become experts in solving problems of quantitative Aptitude
2. Learn to defend and critique concepts of logical reasoning
3. Integrate and display verbal ability effectively

Module:1	Speed Maths	12 hours

Speed Maths

- Addition and Subtraction of bigger numbers
- Square and square roots
- Cubes and cube roots
- Vedic Maths
- Multiplication Shortcuts
- Multiplication of 3 and higher digit numbers
- Simplifications
- Comparing fractions
- Shortcuts to find HCF and LCM
- Divisibility tests shortcuts

Module:2	Percentages, Simple and compound Interest	5 hours

- Percentages
- Simple Interest
- Compound Interest
- Realtion between Simple and Compound Interest

Module:3	Profit and loss, Partnership	5 hours

- Profit and loss
- Partnership
- Averages
- Mixtures and Alligations

Module:4	Data Arrangements and Blood relations	5 hours

- Linear Arrangement
- Circular Arrangement
- Multi-dimensional Arrangement
- Blood relations

Module: 5	**Cryptarithmetic and Attention to detail**	**5 hours**
• Crypt Arithmetic • Attention to detail		
Module: 6	**Life skills**	**8 hours**
• Profile building • Decision making • Critical thinking • Problem solving		
Module:7	**Reading Comprehension for placements**	**3 hours**
• Types of questions • Comprehension Strategies • Practice exercises		
Module:8	**Critical Reasoning and Voices**	**2 hours**
• Critical Reasoning • Active voice • Passive voice		
	Total Lecture hours:	**45 hours**

Text Book(s)

1.	Aggarwal R.S. (2017). *Quantitative Aptitude for Competitive Examinations* 3rd (Ed.). New Delhi: S. Chand Publishing.
2.	FACE. (2016). *Aptipedia Aptitude Encyclopedia* pt (Ed.). New Delhi: Wiley Publications.
3.	ETHNUS. (2016). *Aptimithra,* pt (Ed.) Bangalore: McGraw-Hill Education Pvt.Ltd.

Reference Books

1.1 Sharma Arun. (2016). *Quantitative Aptitude,* 7th(Ed.). Naida: McGraw Hill Education Pvt. Ltd.

Mode of evaluation: CAT, Assessments and FAT (Computer Based Test)

Recommended by Board of Studies |		
Approved by Academic Council |	Date	

Course Code	Course Title	L	T	P	C
MACSE501	Data Structures and Algorithm Analysis	3	0	2	4
Pre-requisite	Nil	Syllabus Version			
		1			

Course Objectives

1. To familiarize the concepts of data structures and provide a mathematical framework for the design and analysis of algorithms.
2. To provide a deeper insight into linear and non-linear data structures.
3. To disseminate knowledge on how to develop algorithms using various design strategies to solve real-world problems.

Course Outcomes

1. Develop the ability to analyse the efficiency of algorithms using asymptotic notations and implement linear data structures.
2. Apply tree and graph-based data structures to design and implement efficient traversal, search, and pathfinding algorithms.
3. Demonstrate the ability to design and implement efficient solutions to complex problems using divide-and-conquer and greedy strategies.
4. Formulate optimal and exhaustive search-based solutions for computationally intensive problems using dynamic programming, backtracking, and branch-and-bound technique.
5. Critically analyse computational complexity, differentiate between P, NP, and NP-complete problems, and apply approximation algorithms to develop near-optimal solutions for intractable problems.

Module:1	Complexity Analysis and Elementary Data Structures	8 hours

Overview and importance of algorithms and data structures- Algorithm specification, Asymptotic Notation - The Big-O, Omega and Theta notation, Recursion, Performance analysis, Array, Stack, Queue, Linked-list and its types, various representations, operations and applications of Linear Data Structures.

Module:2	Non Linear Data Structures	10 hours

Trees: Binary trees, Properties of binary trees, Binary Search Tree, AVL, Red-Black Trees, B- Trees, Segment Trees - Heaps: Binary heaps, Min-Max heap – Graphs: Representation, Breadth first Search, Depth First Search, Shortest path algorithm –Bellman Ford algorithm - Network flow algorithm: Ford-Fulkerson and Push-relabel algorithm.

Module:3	Divide and Conquer and Greedy Techniques	8 hours

Divide and Conquer: Merge Sort, Quick Sort, Karatsuba's fast multiplication method, Strassen's Matrix Multiplication - Greedy Strategy: Job Sequencing Problem with Deadlines, Minimum Spanning Tree: Prims algorithm, Kruskal's algorithm, Huffman coding.		
Module:4	**Dynamic Programming and Backtracking Branch and Bound Techniques**	**9 hours**
Dynamic programming: Matrix Chain Multiplication, Longest Common Subsequence - Backtracking: N-Queens problem, Subset Sum problem, Graph coloring problem Branch and Bound: Subset sum, A-Star, LIFO-BB and FIFO BB methods.		
Module:5	**Complexity Classes and Approximation Algorithms**	**8 hours**
Class P - Class NP - Reducibility and NP-completeness - Circuit Satisfiability problem- 3CNF, Independent Set, Clique, Approximation Algorithms: Vertex Cover, Set Cover and Travelling salesman.		
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. , "**Introduction to Algorithms** ", MIT Press, **2022**
Allen, Weiss Mark. , "**Data structures and algorithm analysis in C++**", Pearson Education India, **2007**

Reference Books

Skiena, Steven S. , "**The Algorithm Design Manual (Texts in Computer Science)**", Springer, 3rd Edition, **2020**
Jon Kleinberg and EvaTardos, "**Algorithm Design**", Pearson Education, 1st Edition, **2014**
Brass, Peter., "**Advanced data structures**", Cambridge University Press, **2008**

Indicative Experiments

1. Linear data structures: stack, queue, Linked lists In the situation where there are multiple users or a network computer system, you probably share a printer with other users. When you request to print a file, your request is added to the print queue. When your request reaches the front of the print queue, your file is printed. This ensures that only one person at a time has access to the printer and that this access is given on a first-come, first served basis. Design and implement the algorithm for this scenario.	4 hours

2. Non-linear data structures: Trees & Graphs You are making an iPod playlist to hear the songs. Assuming that shuffle functions are not applicable, choose an appropriate data structure that will add and delete songs onto you're your iPod in such a way that the recently inserted song will always be the first song currently on the iPod.	4 hours
3. Graph algorithms: BFS, DFS, network flow You are developing a diagnostic tool for a large, complex communication network (e.g., an internet service provider's infrastructure, or a peer-to-peer file-sharing network). The network shown below is an undirected, unweighted graph, where devices are nodes (routers, servers, end-points) which are connected directly by edges. Nodes: A, B, C, D, E, F, G, H, I, J Edges: A-B, A-C B-D, B-E C-E, C-F, D-G, E-H, F-I, G-J, H-J, I-J. Implement functionalities primarily using Breadth-First Search (BFS) with required assumptions to answer the following queries: •□□□□□□ Given a source_device and a target_device, find the minimum number of hops (direct connections) required to establish a path between them. If no path exists, indicate so. •□□□□□□ Given a starting_device and a max_hops limit, determine all devices that can be reached from the starting_device within max_hops. This could represent the reach of a network broadcast or a virus. •□□□□□□ Given a specific pair of source_device and target_device, identify all "bridge devices" along every shortest path between them. A device is a "bridge device" in this context if its removal would increase the shortest path distance between the source_device and target_device or disconnect them entirely. This requires careful application or multiple runs of BFS. •□□□□□□ Given a starting_device, find the diameter of the connected component to which it belongs. The diameter is the longest shortest path between any two nodes in that component.	4 hours
4. Trees: AVL, Red-Black tree You are tasked with designing the core data structure for a large e-commerce platform's product catalog. Products are organized in a strict multi-level hierarchy (e.g., Electronics -> Laptops -> Gaming Laptops -> BrandX Gaming Laptop Model Y). Each product has a unique product_ID, a price, and a	4 hours

quantity_in_stock. The catalog is highly dynamic, with frequent additions, removals, price changes, and stock updates. Implement a robust and efficient tree-based data structure (e.g., a balanced Binary Search Tree like an AVL Tree or Red-Black Tree, where nodes can represent categories or individual products) to manage this hierarchical product catalog. Your implementation must support the CRUD operations.	
5. Divide & Conquer: Merge sort, Quick sort, Karatsuba's fast multiplication method You have n coins, all of which are gold except one coin which appears to be a gold coin, but it is fake. All gold coins are of the same weight, the fake coin weighs less than the others. You have a balance scale, you can put any number of coins on each side of the scale at one time and it will tell you if the two sides weight the same, or which side is lighter if they don't weight the same. The problem is to identify the fake coin. Design an algorithm to find the fake coin in the given n coins.	4 hours
6. Dynamic programming: MCM, LCS A chain of matrices A is called as Add-multiply-chain(AM-chain) if all the matrices of the chain are sqaure matrices of the same size and both addition & multiplication operations are involved in the chain. For example, $A = A_1 * A_2 * A_3 + A_4 * A_5 * A_6 * A_7 + A_8 * A_9 * A_{10}$ is an AM-chain. The above AM-chain A is written as (A, 10, 3, 7), which conveys that the chain A has 10 matrices, with two '+'operations, one '+'operation after A3, another '+' operation after A7, and multiplication is the operation in all the required places in the chain. Multiplication (or addition) between any two numbers is called a scalar operation. The number of scalar operations involved in the chain $A_1A_2 + A_3$ is 12, where all three matrices are square matrices of size 2. (A, n, i, j) is a chain A with n matrices with a '+' operation after A_i, another '+' operation after A_j, and all the other operations are multiplication. Given an AM-chain (A, n, i, j), design a pseudocode and implement it to parenthesize the chain A such that the number of scalar operations to evaluate the chain A is minimum.	2 hours
7. Backtracking: N-Queens problem, Subset Sum, Graph Coloring You have a standard 8x8 chessboard (if your are not familiar with game of chess, please get to know), empty but for a single knight on some square. Your task is to generate a series of legal knight moves that result in the knight visiting every square on the chessboard exactly once. In addition, the knight	2 hours

must end on a square that is one knight's move from the beginning square. The output of your program should indicate the order in which the knight visits the squares, starting with the initial position. Generalize your program for an n x n board where n>8.	
8. Branch & Bound: A-Star, Subset Sum You are developing the pathfinding core for a logistics company using autonomous delivery vehicles in a dense urban environment. The city map is represented as a grid, where each cell has an associated "traffic cost" (representing time delay due to congestion, road conditions, etc.). Some cells may be temporarily designated as "restricted zones" or "accident sites" with extremely high or infinite costs. Vehicles can move horizontally, vertically, and diagonally. You need to find the most cost-effective path from a starting depot to a delivery destination. Implement the A* search algorithm to find the minimum-cost path from a start_cell (start_row, start_col) to a target_cell (target_row, target_col) within a given M x N integer grid.	2 hours
9. Approximation Algorithm: Vertex Cover, Set Cover You are a network security engineer responsible for monitoring a critical communication network. The network consists of various communication nodes (e.g., servers, routers, switches) and the direct communication links between them. To detect potential intrusions or failures, you need to install monitoring agents on a subset of these nodes. An agent on a node can monitor all direct communication links connected to that node. Your goal is to select the minimum number of nodes to place agents on, such that every communication link in the network is monitored. However, finding the absolute minimum is a known NP-hard problem for large networks, so an efficient approximation is required. Implement a 2-approximation algorithm for the Minimum Vertex Cover problem.	4 hours
Total Laboratory Hours:	**30 hours**

Mode of Evaluation : Continuous Assessment Test, Final Assessment Test	
Recommended by Board of Studies :	23-05-2025
Approved by Academic Council : No. 78	12-06-2025

Course Code	Course Title	L	T	P	C
MACSE512	Operating Systems	3	0	2	4
Pre-requisite	Nil	Syllabus Version			
		1			

Course Objectives

1. To provide a comprehensive understanding of the evolution and fundamental components of modern operating systems, including process/thread management and virtualization technologies.
2. To equip students with the ability to design and analyze system-level mechanisms such as CPU scheduling, memory and file management, and synchronization in both traditional and real-time environments.
3. To prepare students for modern computing environments by exploring cloud OS concepts, containerization, and security techniques in distributed and edge systems.

Course Outcomes

1. Explain the structure and key components of an operating system including system calls, interfaces, and virtualization techniques.
2. Analyze process lifecycle, scheduling techniques, and implement synchronization mechanisms in multi-threaded environments.
3. Demonstrate memory management techniques such as segmentation, paging, and virtual memory across architectures.
4. Evaluate storage management approaches, disk scheduling algorithms, and file system implementation strategies.
5. Apply protection and security mechanisms in OS and explain the role of containers and distributed system management.

Module:1	Introduction	5 hours

Operating System Components – Design Architectures: monolithic, Layered, Microkernel, Modular, Hybrid – Operating System Operations: Interrupt Handling, Dual Mode Operation, Multitasking, Context Switching - Process Management -Memory Management - Storage Management - Protection and Security - Kernel Data Structures – Open Source Operating Systems: Definition, Types, Structures, Uses - Operating System Interface Types: Command Line, Graphical User, Touch Based - System Calls -Types of System Calls - System Programs - Virtualization Basics: OS-Level vs Hardware Level Virtualization – Hypervisors

Module:2	Process Management	11 hours

Process Concept - Process Scheduling - Operations on Processes - Inter-process Communication - Multicore Programming - Multithreading Models -Thread Libraries - CPU Scheduling - Scheduling Criteria - Scheduling Algorithms - Thread Scheduling -Multiple-Processor Scheduling - Real-Time OS Scheduling (Rate Monotonic, EDF) and Schedulability analysis - Process Synchronization - The Critical-Section Problem - Peterson's Solution - Synchronization Hardware -

Mutex Locks – Semaphores – Monitors -Classic Problems of Synchronization – Deadlock – Prevention – Avoidance- Detection – Recovery		
Module:3	**Memory Management**	**9 hours**
Contiguous Memory Allocation – Segmentation - Paging - Structure of the Page Table - Intel 32 and 64-bit Architectures - ARM Architecture - Virtual Memory - Demand Paging -Copy-on-Write - Page Replacement- Allocation of Frames - Thrashing - Allocating Kernel Memory		
Module:4	**Storage Management**	**9 hours**
Mass-Storage Structure - Disk Structure - Disk Attachment - Disk Scheduling - Disk Management - Swap-Space Management - File-System Interface - File Concept - Access Methods - Directory and Disk Structure - File-System Mounting - File Sharing - Protection - File-System Implementation -File-System Structure - Implementation - Directory - Allocation methods - NFS , EXT4 File Systems - I/O Systems - I/O Hardware - Application I/O Interface - Kernel I/O Subsystem - Distributed Storage Management		
Module:5	**Protection Security and Containerization**	**9 hours**
Protection - Domain of Protection - Access Matrix - Implementation of the Access Matrix -Access Control - Revocation of Access - Security - Program Threats- System and Network Threats - User Authentication - Implementing Security Defenses - Firewalling - Systems and Networks - Linux System Case study - Distributed Systems Basics - Containerization: Docker and Kubernetes		
Module:6	**Contemporary Issues**	**2 hours**
Contemporary Issues		
	Total Lecture Hours:	**45 hours**
Text Book(s)		
Abraham Silberschatz, Peter B. Galvin, Greg Gagne, " **“Operating System Concepts”"**, Wiley United States, 10th Edition, **2018** Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau , **"“Operating Systems: Three Easy Pieces”"**, Arpaci Dusseau Books, 1st Edition, **2023**		
Reference Books		
Thomas Anderson, Michael Dahlin, " **“Operating Systems: Principles and Practice”** ", Recursive Books Austin Texas USA, 2nd Edition, **2020** Kaiwan N. Billimoria , **"“Linux Kernel Programming”"**, Packt Publishing Birmingham United Kingdom, 2nd Edition, **2022**		
Indicative Experiments		

1. To familiarize students with basic Linux/Unix shell commands used for file handling, process management, user administration, and system navigation.	3 hours
2. To set up and configure a virtualization environment using tools such as Virtual Box, VMware, or KVM, and understand the basics of virtual machines and hypervisors.	3 hours
3. To implement and observe process creation and control using system calls such as fork(), exec(), wait(), and exit() in a Linux environment.	3 hours
4. To demonstrate the creation and execution of threads using the POSIX pthread library, and understand the basic concepts of multithreading and thread management.	3 hours
5. To simulate and compare CPU scheduling algorithms such as FCFS, SJF, Round Robin, and Priority Scheduling and analyze their performance based on turnaround time and waiting time.	3 hours
6. To simulate real-time scheduling algorithms such as Rate Monotonic Scheduling (RMS) and Earliest Deadline First (EDF), and evaluate their schedulability in real-time systems.	3 hours
7. To simulate deadlock detection in a multi-process system using resource allocation matrices and implement a deadlock detection algorithm (e.g., Banker's Algorithm or Wait-for Graph).	3 hours
8. To implement process synchronization using semaphores in producer-consumer or reader-writer problem scenarios and understand race conditions and critical section handling.	3 hours
9. To simulate dynamic memory allocation techniques such as First Fit, Best Fit, and Worst Fit and evaluate memory utilization and fragmentation.	3 hours
10. To simulate and evaluate page replacement algorithms such as FIFO, LRU, and Optimal and compare their performance in terms of page faults.	3 hours
Total Laboratory Hours:	**30 hours**
Mode of Evaluation : Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment	
Recommended by Board of Studies :	23-05-2025
Approved by Academic Council : No. 78	12-06-2025

Course Code	Course Title	L	T	P	C
MACSE513	Computer Networks	3	0	2	4
Pre-requisite	Nil	Syllabus Version			
		1			

Course Objectives

1. To impart knowledge of various network models, their layered architecture, and associated protocols.
2. To equip with the foundational knowledge of routing algorithms and their applications in networking.
3. To facilitate the comprehension of the fundamentals, architecture, and characteristics of wireless and mobile networks.

Course Outcomes

1. Explore the Internet model and various network performance metrics.
2. Interpret the application layer protocols, services and evaluate the requirements for reliable services and implications of congestion at the transport layer services.
3. Analyze various functionalities required in the control and data plane at network layer services.
4. Explain the functions of data link layer including error control and access control.
5. Evaluate the characteristics of wireless and mobile networks, including their security protocols and standards.

Module:1	Computer Networks and Internet	9 hours

Internet: A Nuts-and-Bolts Description - Network Protocols - The Network Edge: Access Networks and Physical Media - The Network Core: Packet Switching, Circuit Switching - Network of Networks - Delay, Loss and Throughput in Packet-Switched Networks – Protocol Layers and Their Service Models(TCP/IP)- Principles of Network Applications: Architectures

Module:2	Application Layer and Transport Layer	10 hours

Application Layer: The Web and HTTP - Electronic Mail in the Internet, FTP - DNS—The Internet's Directory Service - Peer-to-Peer File Distribution - Transport Layer: Relationship Between Transport and Network Layers - Overview of the Transport Layer in the Internet - Multiplexing and Demultiplexing - Connectionless Transport: UDP – Reliable Data Transfer: Go-Back-N (GBN) and Selective Repeat (SR) - Connection-Oriented Transport: TCP, Flow Control and Congestion Control - Socket Programming: Creating Network Applications: TCP/UDP socket programming

Module:3	Network Layer	10 hours

Network Layer – Router - The Internet Protocol (IP): IPv4, Addressing and IPv6 – Generalized Forwarding and SDN - Control Plane: Per-router control and logically centralized control - Routing Algorithms - Link-State (LS) Routing Algorithm,

Distance-Vector (DV) Routing Algorithm, Intra-AS Routing in the Internet: OSPF and Routing Among the ISPs: BGP - SDN Control Plane		
Module:4	**Link Layer and LANs**	**8 hours**
Overview of Link Layer Services - Error-Detection and -Correction Techniques: Checksum and CRC - Multiple Access Links and Protocols: Channel Partitioning Protocols and Random-Access Protocols - Switched Local Area Networks: Link-Layer Addressing and ARP - Virtual Local Area Networks		
Module:5	**Wireless and Mobile Networks Security**	**6 hours**
Elements of a wireless network - Wireless Links and Network Characteristics - WiFi: 802.11 Wireless LANs - Mobility Management: Principles - Wireless and Mobility: Impact on Higher Layer Protocol- Security in Computer Network- Message Integrity and Digital Signatures - Network-Layer Security: IPsec and Virtual Private Networks		
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

James F. Kurose, Keith W. Ross, "**Computer Networking: A Top-Down Approach**", Pearson, United Kingdom., 8th Edition, **2022**

Reference Books

Andrew S. Tanenbaum, "**Computer Networks**", Pearson, Singapore, 6th Edition, **2022**

Larry Peterson and Bruce Davie, "**Computer Networks: A Systems Approach**", Morgan Kaufmann, United States of America., 6th Edition, **2021**

Larry Peterson and Bruce Davie, "**Computer Networks: A Systems Approach**", Morgan Kaufmann, United States of America, 6th Edition, **2021**

Indicative Experiments

1. Hardware Demo (Demo session of all networking hardware and Functionalities) OS Commands (Network configuration commands): □ □ □ □□Identify and understand the functionalities of networking hardware (e.g., routers, switches, hubs, cables). □ Execute network configuration commands (e.g., ifconfig, ipconfig, ping, tracert, netstat) on operating systems like Linux/Windows to configure and troubleshoot network settings.	2 hours

□ □□Document the purpose and output of at least ten commands and describe the role of each hardware component.	
2. Network Packet Analysis using Wireshark: □ □ □ □ □ □ □ □ □ □**Packet Capture**: Capture live network traffic from a specified interface (e.g., Ethernet, Wi- Fi) for at least 5 minutes. □ □ □ □ □**Starting Wireshark**: Configure Wireshark with appropriate capture filters to focus on specific protocols (e.g., HTTP, TCP). □ □ □ □ □□**Viewing Captured Traffic**: Analyze captured packets, identifying key fields like source/destination IP, protocol, and payload. □ □ □ □ □**Analysis, Statistics, and Filters**: Generate statistical reports (e.g., protocol hierarchy, conversations) and apply display filters to isolate specific traffic (e.g., tcp.port == 80).	3 hours
3. Socket programming (TCP and UDP), Multi-client chatting: □ □ □ □□Develop a client-server application using socket programming in C language. Implement a TCP/UDP based server that handles multiple clients simultaneously for a chat application. □ □ □ □ □Implement a TCP/UDP based client-server application for simple message exchange. □ □ □ □ □Test the applications by running multiple clients and verifying message delivery.	4 hours
4. IP addressing, Classless addressing: □ □ □ □ □Configure IP addresses on a simulated network using tools like Cisco Packet Tracer. □ □ □ □ □Assign IP addresses to devices using both classful and classless (CIDR) addressing schemes. □ □ □ □ □Calculate subnets for a given network (e.g., divide a /24 network into subnets). □ □ □ □ □Verify connectivity using ping and document the addressing scheme. □□□□□□ Implement a simple client-server program where the server listens on a configured IP address and port. The client sends a message containing classless IP addresses to the server. The server	3 hours

calculates the first address, last address, and other related information. Test the program to ensure successful communication.	
5. Performance evaluation of routing protocols using simulation tools: □ □ □ □□Use the NS-3 simulator to evaluate the performance of routing protocols (e.g., RIPng, OSPFv3). □ □ □Write a C++ program in NS-3 to set up a network topology with at least five routers, configuring IPv6 addresses and enabling routing protocols (e.g., RIPng and OSPFv3). □ □ □ □Simulate traffic using NS-3's traffic generation tools (e.g., OnOffApplication) and measure performance metrics such as convergence time (using routing table updates), packet loss, and throughput (using FlowMonitor). □ □ □ □ □□Analyze the routing tables generated by each protocol to determine the shortest path between nodes. Compare the performance of at least two protocols (e.g., RIPng vs. OSPFv3) based on the collected metrics.	3 hours
6. SDN Applications and Use Cases Explore Software-Defined Networking (SDN) using a tool like Mininet. □□□□□□□ Create a simple SDN topology with a controller (e.g., OpenDaylight, Ryu). •□□□□□□□□ Implement a use case, such as traffic engineering or load balancing. •□□□□□□□□ Test the setup by generating traffic and analyzing controller behavior.	2 hours
7. Error detection and correction mechanisms, Flow control mechanisms: □ □ □ □□Implement and analyze error detection/correction and flow control mechanisms. □ □ □ □ □Write a program to simulate error detection (e.g., CRC, checksum) and correction (e.g., Hamming code) on sample data. □ □ □ □ □Implement a flow control mechanism (e.g., sliding window protocol) in a simulated environment. □ □ □ □ □Test the implementations with different error rates and window sizes.	4 hours

8. Wired and Wireless Network Simulation Using Cisco Packet Tracer: □ □ □ □ □ □Use Cisco Packet Tracer to design and simulate a network. □ □ □ □ □Create a topology with both wired (e.g., Ethernet) and wireless (e.g., Wi-Fi) components, including at least two routers, two switches, and five end devices. □ □ □ □ □Configure network settings (e.g., IP addresses, VLANs, SSIDs). Test connectivity and simulate data transfer.	3 hours
9. Security in Network - Use cases: □ □ □ □ □Explore network security use cases. □ □ □ □ □Set up a network in a simulation tool (e.g., GNS3, Packet Tracer) and implement at least two security mechanisms (e.g., ACLs, VPN, firewall rules). □ □ □ □ □Simulate an attack (e.g., unauthorized access attempt) and demonstrate how the security measures prevent it. □ □ □ □ □Document the setup and results.	3 hours
10. Network Security with IPsec and Message Integrity: □ □ □ □ □Use GNS3 to configure a secure network and implement message integrity checks. □ □ □ □ □Set up a network topology with two routers and two end devices, configuring an IPsec VPN tunnel between the routers to secure communication using ESP (Encapsulating Security Payload) in tunnel mode. □ □ □ □ □Write a C program to implement message integrity using HMAC-SHA256 for a simple client-server application running on the end devices. The program will compute and verify digital signatures for messages exchanged over the VPN. □ □ □ Simulate data transfer through the VPN and verify that the IPsec tunnel encrypts traffic (using Wireshark to inspect packets). Test the HMAC implementation by introducing tampered messages and checking detection. Submit the GNS3 configuration file, the C source code for the HMAC implementation, a report detailing the IPsec setup, packet capture analysis, and results of the message integrity tests.	3 hours

Submit the GNS3 configuration file, the C source code for the HMAC implementation, a report detailing the IPsec setup, packet capture analysis, and results of the message integrity tests.	
Total Laboratory Hours:	**30 hours**

Text Book(s)
James F. Kurose, Keith W. Ross, " **Computer Networking: A Top-Down Approach**", Pearson, United Kingdom, 8th Edition, **2022**
Reference Books
Andrew S. Tanenbaum, "**Computer Networks**", Pearson, Singapore, 6th Edition, **2022**
Mode of Evaluation : Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment, Oral Examination, Seminar, Group Discussion

Recommended by Board of Studies :	23-05-2025
Approved by Academic Council : No. 78	12-06-2025

Course Code	Course Title	L	T	P	C
MACSE517	Computer Architecture and Organization	3	0	2	4
Pre-requisite	Nil	Syllabus Version			
		1			

Course Objectives

1. To provide knowledge on the basics of computer architectures and organization that lays the foundation to study high-performance architectures.
2. To impart skills to design and develop parallel programs using parallel computing platforms such as OpenMP, CUDA.
3. To familiarize on the domain-specific architectures and real-world implementations in modern processors.

Course Outcomes

1. Outline the developments in the evolution of computer architectures and parallel programming paradigms.
2. Design and evaluate memory hierarchies including multi-level cache systems and virtual memory.
3. Develop parallel programs using OpenMP and analyze performance parameters such as speed-up, and efficiency for parallel programs against serial programs.
4. Analyze GPU and SIMD architectures and implement data-level parallelism using CUDA.
5. Compare domain-specific architectures and evaluate their impact in modern data center applications.

Module:1	Fundamentals of Computer Architecture	5 hours

Introduction to Parallelism-Flynn's Classification: SISD, SIMD, MISD, MIMD- Pipelining- Instruction and Arithmetic Pipelines, Superscalar and Super Pipelined Architectures- Metrics for Performance Measurement.

Module:2	Memory Hierarchy Design	8 hours

Optimizations of Cache Performance - Memory Technology and Optimizations- Virtual Memory and Virtual Machines -The Design of Memory Hierarchy-Case Study: Memory Hierarchies in Intel Core i7 and ARM Cortex-A8.

Module:3	Instruction-Level and Thread-Level Parallelism	11 hours

Instruction-level Parallelism: Concepts and Challenges- Basic Compiler Techniques for Exposing ILP –Dynamic Scheduling-Limitations of ILP- Multithreading: Exploiting Thread-Level Parallelism –Shared Memory Multicore Systems, Performance Metrics for Shared-Memory Multicore Systems- Multithreaded Programming using OpenMP.

Module:4	Data-Level Parallelism in Vector, SIMD, and GPU Architectures	10 hours

Introduction, Vector Architecture, SIMD Instruction Set Extensions for Multimedia, Graphics Processing Units, GPU Memory Hierarchy, Detecting and Enhancing Loop- Level Parallelism, CUDA Programming, Case Study: Nvidia Maxwell.		
Module:5	**Domain-Specific Architectures**	**9 hours**
Introduction - Guidelines for DSAs - Google's Tensor Processing Unit: An Inference Data Center Accelerator - Microsoft Catapult: a Flexible Data Center Accelerator - Intel Crest: a Data Center Accelerator for Training.		
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

J.L. Hennessy and D.A. Patterson. , "**Computer Architecture: A Quantitative Approach.**", Morgan Kauffmann Publishers, 6th Edition, **2019**
Gerassimos Barlas, "**Multicore and GPU Programming: An Integrated Approach.**", Morgan Kaufmann, 2nd Edition, **2022**

Reference Books

William Stallings., "**Computer Organization and Architecture: Designing for Performance.**", Pearson, 11th Edition, **2022**
T. Deakin and T.G. Mattson. , "**Programming Your GPU with OpenMP: Performance Portability for GPUs.**", MIT Press Cambridge, 1st Edition, **2023**
J. Sanders and E. Kandrot, " **CUDA by Example: An Introduction to General-Purpose GPU Programming.**", Addison-Wesley Boston, 1st Edition, **2010**

Indicative Experiments

1. Create a new project using Visual Studio or an equivalent IDE that supports OpenMP. Configure the project properties to enable OpenMP support, compile, and execute the basic "hello world" program.	2 hours
2. Construct an OpenMP program using the parallel construct that prints the thread number from each thread in the parallel region to demonstrate how multiple threads execute concurrently.	2 hours
3. Develop a OpenMP program that calculates the execution time of a computational task using timing function and performance counters. Compare their properties.	2 hours
4. Implement an OpenMP program using various environment routines to access processor run-time information that simulates a group of students (threads) solving a set of problems simultaneously. Each student reports their ID, total students participating, maximum possible students, and processor details.	2 hours
5. Write an OpenMP program that simulates operations on a deck of cards. The program should use the loop construct to shuffle the deck in	4 hours

parallel, the sections construct to deal cards concurrently to two players, and the single construct to print a game start announcement exactly once. Use these constructs to demonstrate how parallel tasks can be organized efficiently. Also, explain the purpose and behavior of each construct.	
6. Generate an OpenMP program to simulate a bakery model with 20 customer orders for varying preparation times. Use the static, dynamic, and guided scheduling clauses to assign orders to multiple threads. Observe and compare how each scheduling method affects workload distribution and overall execution time.	2 hours
7. Use Pin instrumentation and Cache grind profiling to analyze cache usage and identify where cache misses happen.	2 hours
8. Track branch predictions and mispredictions during execution and identify inefficient branching and improve performance.	2 hours
9. Develop a CUDA-platform setup on NVIDIA / Google Colab.	2 hours
10. Build a CUDA program in C/C++ environment to add two large arrays using unified memory, validate the result with a CPU implementation, and measure GPU kernel execution time using CUDA events.	2 hours
11. Write a CUDA program in C/C++ environment to reverse the elements of a large array using a single thread block, utilize shared memory for efficient access, include boundary checks, and measure the execution time using CUDA events.	2 hours
12. Implement a CUDA program in C/C++ environment to perform matrix addition and multiplication using shared memory optimization, validate the results with CPU computation, and compare execution times using CUDA event profiling.	2 hours
13. Hands-on Lab: CNN Training with TensorFlow on TPUs.	4 hours
Total Laboratory Hours:	**30 hours**
Mode of Evaluation : Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment	
Recommended by Board of Studies :	23-05-2025
Approved by Academic Council : No. 78	12-06-2025

Course Code	Course Title	L	T	P	C
MACSE518	Database Modeling and Design	3	0	2	4
Pre-requisite	Nil	Syllabus Version			
		1			

Course Objectives

1. To inculcate the foundational knowledge of database concepts, ER modeling, relational algebra, and optimization techniques for effective database design.
2. To equip with advanced skills in EER modeling, complex relationship modeling, schema mapping, and transactional programming using SQL and PL/SQL.
3. To impart skills in normalization, distributed systems, advanced data modeling, and cutting-edge NoSQL and Big Data technologies.

Course Outcomes

1. Comprehend the fundamental of database systems and its architectures.
2. Model the database systems using ER and EER to design complex database systems.
3. Analyze and apply normalization theory to relational schemas and optimize the design for minimal redundancy.
4. Gain insight into advanced database technologies for handling complex problems.
5. Design NoSQL databases and handle big data to solve real-world problems.

Module:1	Introduction to Database Systems	6 hours

Data, Information, Metadata, Data Management - Purpose of Database– Data Models and concepts- Database System Components - Three Schema Architecture– Centralized and Client/Server Architectures–Conceptual Design using ER Modeling: Entities, Attributes, Relationships, Integrity Constraints, Strong Entities, Weak Entities– ER Modeling Process.

Module:2	Database Modeling and Database Languages	10 hours

Enhanced Entity-Relationship (EER) Modeling: Subclasses, Superclasses, and Inheritance– Specialization and Generalization– Constraints– Modeling of UNION Types Using Categories- Ternary and Higher Order Relationships, Recursive Relationships– Mapping ER and EER Models to Relational Schema– Introduction to Relational Algebra: Basic Operators: Selection, Projection, Join, Union, Intersection– Views and Materialized Views– Query Optimization– Indexing. Database Languages: Data Definition Language (DDL)– Data Manipulation Language (DML)– Data Control Language (DCL)– Display Queries (DQL)– Joins, Subqueries– Transaction Control Language (TCL)– COMMIT, ROLLBACK, SAVEPOINT– PL/SQL– Cursors, Triggers.

Module:3	Relational Database Design and Transaction Management	10 hours

Normalization – Functional Dependencies: Concept, Equivalence Set– Minimal Cover, Attribute Closure– Deriving Candidate Keys and Purpose of Keys– Normal Forms: 1NF, 2NF, 3NF, BCNF– Comprehensive Normalization Examples– Dependency Preserving Decomposition– Lossless Join Decomposition– Higher Normal Forms: Multi-valued Dependencies, 4NF, Join Dependencies, 5NF, Domain-Key Normal Form (DKNF) –Transactions – Concepts of Transactions– Serializability– Concurrency Control – Database Recovery.		
Module:4	**Advanced Database Systems**	**10 hours**
Parallel Databases: Architecture, Interoperation Parallelism and Intraoperation parallelism – Data Partitioning Strategies– Query Optimization for Parallel Execution– Distributed Database Concepts– Features– Fragmentation, Allocation, and Replication– Concurrency Control and Recovery in Distributed Databases– Query Processing and Optimization in Distributed Databases– Spatial Database– Temporal Database– Multimedia Database Concepts– Graph Databases - Knowledge Graphs.		
Module:5	**NoSQL Databases and Big Data Systems**	**7 hours**
Semi-Structured Data Models: JSON, XML, RDF– NoSQL Databases: MongoDB, Key–Value Stores– Big Data Systems: HDFS Overview, Apache Spark– Cloud Data Storage.		
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

R. Elmasri and S. Navathe, "**Fundamentals of Database Systems**", Addison-Wesley, 7th Edition, **2021**

D Abraham Silberschatz, Henry F. Korth, S. Sudarshan , "**Database System Concepts**", McGraw Hill, 7th Edition, **2021**

Reference Books

Edward Sciore, "**Database Design and Implementation: Second Edition (Data-Centric Systems and Applications)**", Springer, 2nd Edition, **2020**

Sadalage & Fowler, "**NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence**", Addison-Wesley, 2nd Edition, **2021**

Guy Harrison, "**Next Generation Databases: NoSQL, NewSQL, and Big Data**", Apress, 2nd Edition, **2020**

Indicative Experiments

1. Design an EER diagram for a Research Collaboration Portal. • Entities: Professors, Students, Projects, Publications, FundingAgencies. • Generalization: Person (Professor, Student).	2 hours

• Specialization: Professors can be Principal Investigator (PI) or Co-Investigator (Co-PI). • Recursive Relationship: Professors mentoring junior faculty or students. • Ternary Relationship: A project is supervised by a PI, funded by a FundingAgency, and has at least one student. • Constraints: One project cannot exceed ₹10,00,000 in funding per year. A student cannot be on more than 2 active projects. Cardinality, Participation Tools: Lucidchart, Draw.io, MySQL Workbench.	
2. Relational Schema Mapping: Transform the EER diagram from lab 1 into its corresponding relational schema representation. List all resulting tables with primary keys, foreign keys, and attributes. • Create and populate database tables using appropriate DDL and DML commands. • Specify how the following are handled: Superclass-subclass mapping strategy (e.g., one table per subclass). Recursive relationship. Ternary relationship. Constraints (primary key, foreign key, check, default, not null, unique). Platform: MySQL / PostgreSQL / Oracle.	2 hours
3. Execute advanced SQL queries on the relational schema derived from the EER diagram. This includes retrieving, filtering, and analyzing data using: • Aggregate functions (SUM, AVG, MAX, MIN, COUNT) • GROUP BY, HAVING clauses • Nested subqueries (scalar, correlated, multi-row) • Window functions (ROW_NUMBER, RANK) Platform: MySQL / PostgreSQL / Oracle.	2 hours
4. Develop and execute advanced SQL queries on the relational schema derived from the EER diagram, focusing on advanced data retrieval, filtering, and analysis. This includes the use of: • Standard Joins: Inner Join, Outer Join (Left, Right, Full), Self-Join • Special Joins: Cross Join, Natural Join • Other Joins: Equi Join, Non-Equi Join, Semi Join, Anti Join (using EXISTS / NOT EXISTS) Platform: MySQL / PostgreSQL / Oracle.	2 hours

5. DCL and TCL operations on the relational schema derived from the Research Collaboration Portal. This includes managing user permissions and ensuring consistency of transactional operations using: • DCL Commands: GRANT, REVOKE to control access privileges on tables such as Professors, Projects, and Publications. • TCL Commands: COMMIT, ROLLBACK, and SAVEPOINT to manage and recover database states during multi-step operations like funding updates or project allocations. Platform: MySQL / PostgreSQL / Oracle.	2 hours
6. Using the relational schema of the Research Collaboration, perform the following tasks: • Create indexes based on query requirements. • Define sequences for automatically generating unique IDs (Oracle uses sequences, MySQL uses AUTO_INCREMENT, and PostgreSQL supports both (SERIAL / SEQUENCE)). • Create views to simplify common queries. Platform: MySQL / PostgreSQL / Oracle.	2 hours
7. PL/SQL – Procedures and Functions: Implement and execute stored procedures and functions. Using the Research Collaboration Portal relational schema, implement and execute stored procedures and functions to automate and encapsulate common operations. Tasks: • Create a stored procedure to add a new project. The procedure should accept project details (title, PI ID, funding agency ID, funding amount, start and end dates) and validate that the funding amount does not exceed ₹10,00,000. If the validation fails, rollback the transaction and raise an appropriate error. • Create a function to count the number of active projects for a given student ID. The function should return the count of projects where the current date falls between the project start and end dates. • Create a stored procedure to assign a student to a project only if the student is currently involved in fewer than 2 active projects. If the student is eligible, insert the assignment; otherwise, raise an error. • Create a function that returns the total funding amount received by a specific professor (as PI) across all projects. Platform: MySQL / PostgreSQL / Oracle.	2 hours
8.	2 hours

PL/SQL – Cursors and Triggers: Using the Research Collaboration Portal schema, implement and execute PL/SQL cursors and triggers to handle complex data retrieval and enforce data integrity. Tasks: • Write an explicit cursor to fetch and display all projects along with their Principal Investigator's name and total funding amount. Include cursor open, fetch, and close operations with appropriate exception handling. • Use an implicit cursor to count the number of publications authored by each professor and display the results. • Create a BEFORE INSERT trigger on the Projects table to validate that the funding amount does not exceed ₹10,00,000. If the validation fails, raise an application error to prevent insertion. • Create an AFTER INSERT trigger on the Project_Student table to check that a student is not assigned to more than 2 active projects. If violated, raise an error and roll back the insert. • Create a BEFORE UPDATE trigger on the Professors table to prevent changing the designation of a professor from "Principal Investigator" to any other value if they currently supervise active projects. Platform: MySQL / PostgreSQL / Oracle.	
9. Design a MongoDB-based Student Management System where: • Each student has a unique ID, name, age, and department. • Each student is enrolled in multiple courses. • Each course includes course_code, course_name, credits Platform: MongoDB	2 hours
10. Spatial Databases using GIS Coordinates: Design a spatial database for managing university campuses, department buildings, and student hostels. Tasks: • Store GPS locations of all buildings • Find hostels within 1 km of academic buildings • Identify the nearest cafeteria to each department • Visualize results using GeoJSON exports Platform: PostgreSQL/PostGIS or MongoDB with GeoJSON	2 hours
11. Build a basic application that interacts with cloud-based databases or Big Data frameworks for scalable data storage, retrieval, and analysis. Tasks: • Connect and interact with a cloud database (e.g., Firebase, RDS, BigQuery) • Perform CRUD operations over the cloud	2 hours

• Integrate data ingestion and processing with Big Data tools (e.g., Spark or Hadoop) • Analyze large datasets in a distributed environment • Understand deployment, scalability, and cloud storage optimization Tools: Firebase Realtime DB / Firestore or AWS RDS / MongoDB Atlas or Google BigQuery / Azure Cosmos DB.	
Total Laboratory Hours:	**30 hours**
Mode of Evaluation : Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment, Presentaion	
Recommended by Board of Studies :	23-05-2025
Approved by Academic Council : No. 78	12-06-2025

Course Code	Course Title	L	T	P	C
MACSE519	Machine Learning and Applications	3	0	2	4
Pre-requisite	Nil	Syllabus Version			
		1			

Course Objectives

1. To deliver insights into the foundational and mathematical principles of machine learning algorithms.
2. To impart skills to select and apply appropriate machine learning models to address real-world problems across diverse domains.
3. To explore emerging trends in machine learning and its advancements emphasizing societal impact and ethical considerations

Course Outcomes

1. Comprehend the fundamental properties of machine learning techniques.
2. Design and Develop supervised learning techniques for real-world data challenges.
3. Evaluate ensemble methods to improve model accuracy and robustness.
4. Develop an appropriate unsupervised learning model to manage unknown patterns.
5. Apply Reinforcement Learning concepts and algorithms for intelligent system design.

Module:1	Introduction	6 hours

Overview of Machine Learning (ML) Concepts - Applications - Types of ML algorithms - Hypothesis space and Inductive Bias - Data - Data handling - Model Selection and Evaluation: Train test split and cross-validation - Evaluation metrics

Module:2	Supervised Learning	10 hours

Linear Regression: Simple and Multiple Linear regression - Evaluating regression fit – Non-Linear Regression – Applications: Predicting Sales, Predicting price of Real Estate, Predicting Concrete Compression Strength - Multi-Class and Multi-Label classification -Decision tree learning: Decision tree representation - Decision tree algorithms- Hypothesis space search in decision tree learning - Inductive bias in decision tree learning - Instance based Learning: K nearest neighbor - Support Vector Machine (SVM) - Dual formulation - Maximum margin with noise - Nonlinear SVM and Kernel function.

Module:3	Probabilistic and Ensemble Learning	9 hours

Probability and Bayes Learning: Logistic Regression - Bayesian Learning, Naïve Bayes - Bayesian Belief Networks - Artificial Neural Networks (ANN): Biological motivation - ANN representation - Perceptron - Multilayer perceptron and the back propagation algorithm - Ensembles: Bias-Variance Tradeoff – Bagging - Random

Forest, Boosting – Adaboost - Applications: Predicting Fraud Insurance Claims, Email Spam filtering		
Module:4	**Unsupervised Learning**	**10 hours**
Clustering- Partitional Clustering: k-means - k-medoids - Hierarchical Clustering: Agglomerative - Divisive - Density based clustering: DBSCAN-Spectral clustering - Dimensionality Reduction: Principal Component Analysis - Linear Discriminant Analysis - Expectation Maximization - Mixture of Gaussians- Outliers- Anomaly Detection – Applications :Document Clustering - Customer Segmentation		
Module:5	**Advanced Machine Learning Concepts**	**8 hours**
Self-supervised Learning - Reinforcement Learning (RL) - RL Framework - Markov Decision Process - Policy - Planning algorithms - Q Learning algorithm - Online learning algorithms : Winnow algorithm - On-line to batch conversion - Overview of MLOps - Explainability in ML - Ethics in ML.		
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

Mehryar Mohri, Afshin Rostamizadeh, Ameet Talwalkar , "**Foundations of Machine Learning**", MIT Press, 2nd Edition, **2018**
Ethem Alpaydin, "**Introduction to Machine Learning**", MIT Press Prentice Hall of India, 4th Edition, **2020**

Reference Books

Tom Mitchell, "**Machine Learning**", McGraw Hill, 3rd Edition, **1997**
Shai Shalev-Shwartz and Shai Ben-David, "**Understanding Machine Learning: From Theory to Algorithm**", Cambridge University Press, **2014**
Stephen Marsland , "**Machine Learning: An Algorithmic Perspective**", Chapman and Hall CRC press, 2nd Edition, **2015**
Danish Haroon, "**Python Machine Learning Case Studies: Five Case Studies for the Data Scientist**", Apress, **2017**

Indicative Experiments

1. To effectively **predict alumni donation likelihood**, the university employs a comprehensive preprocessing strategy on its dataset. Numerical features such as annual income (in lakhs) and previous donation amounts (in rupees) are scaled to ensure all features contribute equally to the model. Categorical variables including employment sector, degree program, and marital status are encoded into numerical representations suitable for machine learning algorithms. Furthermore, the geographical location of alumni is transformed, potentially into binary features indicating specific regions or through one-hot encoding to capture more granular location information. These	2 hours

meticulous preprocessing steps are crucial for preparing the data for robust model training and ultimately enhancing the accuracy of predicting which alumni are most likely to donate.	
2. Use **linear regression** to predict a student's final academic performance based on features such as attendance percentage, internal assessment scores, assignment submissions, project grades, previous semester GPA, participation in extracurricular activities, study hours per week, socio-economic background, number of courses taken in the semester, involvement in academic clubs, time spent on campus, previous academic performance, and faculty feedback ratings.	2 hours
3. You have Computational Fluid Dynamics (CFD) simulation results for turbine blades under varying loading and flow conditions. Each simulation provides features like maximum principal stress, Von Mises stress, maximum temperature, displacement magnitude, surface pressure, fatigue life estimate, Reynolds number, and wall shear stress. Using this data, **formulate a decision tree approach** to classify whether a blade is likely to "Fail" or remain "Safe."	2 hours
4. Use the **KNN classifier** to predict the biochemical composition of a sample into enzyme concentration levels (e.g., Low, Medium, High) based on features such as pH level, enzyme activity, substrate concentration, temperature, protein concentration, absorbance at specific wavelengths, reaction time, catalyst concentration, substrate type, ion concentration, viscosity of the solution, and absorbance at 280 nm	2 hours
5. Apply **Support Vector Machines (SVM)** to predict whether a university student will stay enrolled or drop out based on features such as high school GPA, entrance exam score, extracurricular activities, letter of recommendation score, work experience, number of club activities, financial aid status, number of failed courses, mental health status, attendance percentage, family support, and work-study participation.	2 hours
6. **Implement logistic regression** to predict whether a hotel guest will cancel their booking based on features such as lead time, previous cancellations, deposit type (No Deposit, Non-Refundable, Refundable), customer type (Transient, Group, Contract, Transient-Party), number of special requests, average daily rate (ADR), reserved room type, total nights stayed, booking changes, market segment (Direct, Corporate, Online Travel Agent), assigned room type, repeated guest status, and distribution channel.	2 hours
7.	2 hours

A student at a multimedia school has created a web design project, which includes an animated logo, interactive media elements, and is designed using HTML/CSS. You need to predict whether this project will meet the design standards of a client and industry expectations based on the following features: **Features**: **Number of Pages Designed** (numeric): The number of pages in the website design. **Animation Complexity** (categorical): The complexity of the animation used (e.g., simple, moderate, complex). **Client Feedback Rating** (numeric): A rating (1-10 scale) provided by the client after previewing the project. **Interactive Features Implemented** (categorical): Whether the website includes interactive media elements like hover effects, videos, or dynamic content (yes/no). **HTML/CSS Coding Quality** (categorical): The quality of the HTML/CSS code written (e.g., high, medium, low). **Use of Multimedia Tools** (categorical): The tools used to create the multimedia content (e.g., Adobe After Effects, Blender, Final Cut Pro). **Project Deadline Met** (binary): Whether the project was completed on time (1 = yes, 0 = no). **Client Requirements Met** (binary): Whether the project met all client requirements (1 = yes, 0 = no). **Target Variable**: **Project Success** (binary): Whether the project is deemed successful (1 = success, 0 = failure) Using **Naive Bayes classification**, train a model to predict whether a multimedia project will be successful or unsuccessful based on the provided features. Evaluate the model's performance using accuracy, precision, recall, and F1-score	
8. Apply **K-Means clustering** to categorize cricket players based on their performance metrics using features such as batting average, strike rate, number of boundaries per match, total runs, bowling average, economy rate, number of wickets taken, dot ball percentage, fielding efficiency, and match impact score. The clustering can help group players into roles like power hitters, all-rounders, specialist bowlers, consistent batsmen, or fielding specialists.	2 hours
9.	2 hours

Apply DBSCAN clustering to categorize building construction projects based on features such as total construction cost, project duration, number of labor hours, material cost percentage, number of floors, safety incident count, construction method (traditional, modular, prefab), site complexity score, weather delays, and energy efficiency rating, to identify clusters of typical, high-risk, or high-efficiency construction projects.	
10. Use **Principal Component Analysis (PCA)** to analyze and reduce the dimensionality of participant and event data collected during a university techfest, based on features such as number of events participated, event types (coding, robotics, design, management), total scores earned, feedback ratings, number of workshops attended, team size, project submissions, number of prizes won, volunteer hours, and previous techfest experience.	2 hours
11. In the biochemical lab, researchers often run multi-step experiments like protein purification, DNA extraction, or enzymatic reactions. The experimental results depend heavily on parameters like temperature, pH, reaction time, reagent concentration and mixing speed Rather than manually tuning these parameters, you can use **Reinforcement Learning** to learn an optimal sequence of experimental settings that maximizes the experiment's success rate (e.g., purity of protein, yield of DNA, or enzymatic efficiency).	2 hours
12. Implement a **Random Forest classifier** to predict whether a patient is likely to have Tuberculosis (TB) based on clinical and diagnostic features. Use explainability techniques such as feature importance and individual prediction interpretation to ensure the model's decisions are understandable and medically relevant. **Features to include:** ● Chest X-ray abnormalities ● Duration of cough ● Weight loss ● Night sweats	2 hours

• Fever • Sputum test results • Previous TB history	
Total Laboratory Hours:	**30 hours**

Text Book(s)

Mehryar Mohri, Afshin Rostamizadeh, Ameet Talwalkar , "**Mehryar Mohri, Afshin Rostamizadeh, Ameet Talwalkar** ", MIT Press, 2nd Edition, **2018**
Ethem Alpaydin , "**Introduction to Machine Learning**", MIT Press Prentice Hall of India, 4th Edition, **2020**

Reference Books

Tom Mitchell, "**Machine Learning**", McGraw Hill, 3rd Edition, **1997**
Shai Shalev-Shwartz and Shai Ben-David, "**Understanding Machine Learning: From Theory to Algorithms**", Cambridge University Press, **2014**
Stephen Marsland , "**Machine Learning: An Algorithmic Perspective**", Chapman and Hall CRC press, 2nd Edition, **2015**

Mode of Evaluation : Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment, Seminar

Recommended by Board of Studies :	23-05-2025
Approved by Academic Council : No. 78	12-06-2025

Course Code	Course Title	L	T	P	C
MACSE610	Image and Video Analytics	3	0	2	4
Pre-requisite	Nil	Syllabus Version			
		1			

Course Objectives

1. To provide a strong foundation in image and video processing techniques.
2. To equip students with methods for feature extraction, pattern analysis, and classification.
3. To introduce modern deep learning approaches for visual understanding and vision-language tasks.

Course Outcomes

1. Apply image enhancement and transformation techniques in various domains.
2. Extract and interpret meaningful features from images and videos.
3. Analyze and implement pattern classification and object detection using deep learning models
4. Apply motion estimation, segmentation, and tracking techniques for dynamic scene analysis.
5. Build intelligent systems combining vision language models.

Module:1	Fundamentals of Image and Video Processing	9 hours

Basic steps of image processing system- Pixel relationship - Image transforms - Fourier, DCT, Wavelet- Image Noise- Image Blur- Image enhancement - Histogram equalization, Spatial and frequency domain filtering- Color models - YUV, RGB, HSV, YCbCr, LAB - Digital video fundamentals – Spatial Resolution and Frame Rate- Digital Video Standards- Video Streaming.

Module:2	Feature Extraction	9 hours

Boundary Preprocessing- Boundary Feature Descriptors- Region Feature Descriptors- Principal Components as Feature Descriptors- Whole-Image Features- Scale-Invariant Feature Transform- Histogram of Oriented Gradients.

Module:3	Pattern Analysis and Object Detection	9 hours

Image Pattern Classification - Patterns and Pattern Classes- Pattern Classification by Prototype Matching- Optimum Statistical Classifiers: Bayes - Neural Networks and Deep Learning - Deep Convolutional Neural Networks - Region-based Convolutional Neural Networks – YOLO – Vision and Detection Transformer.

Module:4	Motion Estimation, Segmentation and Tracking	9 hours

Video Understanding- Motion estimation- Motion Models, Model Estimation, Matching Methods - Image Segmentation - Change Detection- Background Subtraction - Motion Segmentation- Motion Tracking- Image and Video Matting.

Module:5	**Video Analysis**	**7 hours**
Video super resolution- Pose Estimation- Human activity recognition- Simultaneous localization and mapping - Autonomous Navigation- Vision Language - Video Captioning- Visual Question Answering and Reasoning.		
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

Gonzalez, Rafael C., and Woods, Richard E., "**Digital Image Processing** ", Pearson Education, Inc, 4th Edition, **2018**
Richard Szeliski, "**Computer Vision: Algorithms and Applications**", Springer, 2nd Edition, **2022**

Reference Books

A. Murat Tekalp, "**Digital Video Processing**", Prentice Hall, 2nd Edition, **2015**
Mohamed Elgendy, "**Deep Learning for Vision Systems**", Manning Publications, 1st Edition, **2020**
Ayyadevara, V. Kishore and Reddy, Yeshwanth, "**Modern Computer Vision with PyTorch**", Packt Publishing, 1st Edition, **2020**

Indicative Experiments

1. Implement histogram equalization and frequency domain filtering techniques to enhance contrast and clarity in medical images such as X-rays and MRIs.	3 hours
2. Extract relevant features from images of different vehicles to develop an automated vehicle recognition system for traffic monitoring.	3 hours
3. Implement an image processing pipeline using edge detection and template matching techniques to identify and locate surface defects (e.g., cracks, stains, irregularities) in material images.	3 hours
4. Perform K-Means clustering to segment land, vegetation, and water bodies in satellite images for remote sensing application	3 hours
5. Implement object detection using YOLO to identify pedestrians, vehicles, and traffic signs in road scenes for ADAS applications.	3 hours
6. Implement KLT tracking to follow players' movements in sports videos	3 hours
7. Build a Transformer-based image captioning model that generates descriptive sentences to assist visually impaired users	3 hours
8. Develop a basic Visual Question Answering (VQA) system that can answer questions about an input image, enabling human-machine interaction.	3 hours

9. Build a Transformer-based image captioning model that generates descriptive sentences to assist visually impaired users	3 hours
10. Develop a basic Visual Question Answering (VQA) system that can answer questions about an input image, enabling human-machine interaction	3 hours
Total Laboratory Hours:	**30 hours**
Mode of Evaluation : Continuous Assessment Test, Digital Assignment, Final Assessment Test	
Recommended by Board of Studies :	23-05-2025
Approved by Academic Council : No. 78	12-06-2025

Course Code	Course Title	L	T	P	C
MACSE635	Blockchain Technology and Applications	3	0	2	4
Pre-requisite	Nil	Syllabus Version			
		1			

Course Objectives

1. To impart knowledge on the basics of blockchain technology.
2. To articulate the significance of Distributed Ledger and Smart Contracts.
3. To provide insights on the need for security in Blockchain.

Course Outcomes

1. Understand the data structure and technology of Blockchain.
2. Analyze the use of DLT, Smart Contracts and apply them in real world scenarios.
3. Gain knowledge on the components of Blockchain platforms and Ecosystems.
4. Apply the security and privacy in a Blockchain environment.
5. Analyze the requirements of real-world applications and apply Blockchain.

Module:1	Introduction to Blockchain Technology	8 hours

History and evolution of blockchain –Types - components of blockchain: nodes, block, Cryptographic Hash Functions - Hash Pointers - Data Structures- Merkle Tree -Digital Signatures - Public Keys as Identities - Cryptocurrency - Distributed consensus - proof of work - Bitcoin transactions - Bitcoin Mining - Incentives for miners - Blockchain Architecture - Challenges - Blockchain Design Principles, Asynchronous Byzantine Agreement

Module:2	Distributed Ledger Technology and Smart Contracts	9 hours

Origin of Ledgers – Types and Features of Distributed Ledger Technology - Role of Consensus Mechanism - Public and Private Ledgers- Zero Knowledge Proofs - Anatomy of a Smart Contracts - Life Cycle - Usage Patterns - Basics of smart contracts – Writing smart contracts using Solidity : Solidity fundamentals , Structure of a contracts : Data Types, Variables, Control structures, functions, modifiers, events, Visibility and Getters- Formal verification and testing frameworks – MythX or Hardhat

Module:3	Blockchain Platforms and Ecosystems	9 hours

Blockchain and full ecosystem - Basics of Ethereum, Ethereum Virtual Machine (EVM), Transaction fee, Ether, Gas – GoEthereum (Geth), Truffle – Ganache – Remix-IDE – Wallet Software - Architecture of Hyperledger fabric v1.1-Introduction to hyperledger fabric v1.1 -chain code- Web3- Emerging systems – polkadot or cosmos.

Module:4	Blockchain Security and Privacy	9 hours

Security principles in Blockchain technology - Threats and vulnerabilities in Blockchain systems- Consensus and Security: Role of consensus mechanisms in security - Attacks on consensus- Smart Contract Security: Common vulnerabilities in smart contracts- Privacy in Blockchain : Privacy challenges in public vs private Blockchains.		
Module:5	**Use Cases and Applications**	**8 hours**
Overview of Blockchain adoption in various sectors - Benefits and challenges of Blockchain integration- Financial Services and Banking: Cryptocurrency and decentralized finance (DeFi) - Cross-border payments and remittances- Central Bank Digital Currencies (CBDCs)- Supply Chain Management Provenance and traceability using Blockchain- Case studies: Walmart- IBM Food Trust, etc., Healthcare: Secure medical records management- Data sharing and patient consent via Blockchain - Tokenization of assets and Identity Systems.		
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

Ajay Prasad, Shaurya Gupta, Sonali Vyas, Vinod Kumar Shukla, "**Blockchain Technology: Exploring Opportunities, Challenges, and Applications**", CRC Press, 1st Edition, **2021**

Narayanan, A., Bonneau, J., Felten, E., Miller, A., and Goldfeder, S, "**Bitcoin and cryptocurrency technologies: a comprehensive introduction.** ", Princeton University Press, 1st Edition, **2016**

Reference Books

Imran Bashir, "**Mastering Blockchain**", Packt Publishing Limited, 4th Edition, **2023**

Garg Wiley-Blackwell, "**Blockchain for Real World Applications** ", Wiley-Blackwell, 1st Edition, **2023**

Niaz Chowdhury, Ganesh Chandra Deka, "**Blockchain and AI The Intersection of Trust and Intelligence**", CRC Press, 1st Edition, **2024**

Udai Pratap Rao, Piyush Kumar Shukla, Chandan Trivedi, Sweta Gupta, Zelalem Sintayehu Shibeshi, "**Blockchain for Information Security and Privacy**", CRC Press, 1st Edition, **2021**

Indicative Experiments

1. Simulate a simple Blockchain using JavaScript. Perform the following functions: 1. Create genesis block and display the content of the blockchain 2. Add two subsequent blocks and display the content of the blockchain 3. Modify the data in any one block and check whether the blockchain is valid or not.	3 hours
2. Implement Proof of Work consensus protocol using Javascript and validate the consensus protocol.	3 hours
3. Setting up Ethereum network: 1. By using Geth command line interface	3 hours

2. Identifying and setting up a testnet, like Ropsten or Kovan, so that free ethers can be used as transaction. 3. Transfer ethers from one account to another on an Ethereum testnet.	
4. Using Remix: 1. Solidity code to develop, compile and deploy the contracts. 2. Using setter and getter functions to interact with the contracts	3 hours
5. Implement a simple Smart Contact using Truffle, Ganache, Metamask	3 hours
6. Implement E-Voting Application using Blockchain.	3 hours
7. Implement Academic certificate verification application using blockchain.	3 hours
8. Implement agriculture goods tracking using blockchain for supply chain management application.	3 hours
9. Implement any one healthcare application using blockchain.	3 hours
10. Create a Hyperledger Fabric Blockchain service on Cloud.	3 hours
Total Laboratory Hours:	**30 hours**
Mode of Evaluation : Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment, Seminar, Presentation	
Recommended by Board of Studies :	23-05-2025
Approved by Academic Council : No. 78	12-06-2025

Course Code	Course Title	L	T	P	C
MACSE636	Cyber Security	3	0	0	3
Pre-requisite	Nil	**Syllabus Version**			
		1			

Course Objectives

To introduce the fundamental principles of cybersecurity
To impart comprehensive knowledge and practical skills in cryptographic techniques
To develop the ability to identify, understand, and mitigate various types of cybercrimes

Course Outcomes

Understand and apply core principles of cybersecurity, including data integrity and authentication mechanisms
Demonstrate knowledge of classical and modern cryptographic techniques and their practical applications.
Identify, analyze, and evaluate various types of cybercrimes, threats, and attack vectors.
Develop, implement, and assess effective cybersecurity policies and strategies for organizations.

Module:1	**Overview of Cyber Security and Risk Management**	**7 hours**

Key principles: Confidentiality, Integrity, and Availability (CIA), Threats and vulnerabilities: attacks, defense and weaknesses-Risk analysis and assessment: Identifying and evaluating potential risks

Module:2	**Cryptography Techniques**	**9 hours**

Encryption methods: Classical techniques, Symmetric cryptography- block ciphers: DES-AES, DES, stream ciphers: RC4, and Asymmetric cryptography: RSA, ElGamal, Elliptic Curve, Key Distribution

Module:3	**Integrity and Authentication**	**9 hours**

Hash functions: Message Digest (MD5), Secure Hash Algorithm (SHA-256), Message Authentication, Message Authentication Code (MAC), Digital Signature Algorithm: RSA ElGamal based, HMAC

Module:4	**Cybercrimes and Cyber Offenses**	**9 hours**

Types of cybercrimes: Classification and characteristics, Attack planning and execution, social engineering tactics: Human-based and computer-based attacks: Cyber stalking and Cyber Cafe, phishing, password cracking, keyloggers, spyware, DoS/DDoS attacks, and SQL injection, reconnaissance and disruption methods, malicious codes		
Module:5	**Cyber Security Policy Development and Enforcement**	**9 hours**
Identifying Organizational policy needs: Determining requirements for security policies, Policy creation for Internet Use, Compliance and enforcement: Ensuring adherence to policies and procedures, Policy review and updates		
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

William Stallings, "**Cryptography and Network Security: Principles and Practice**", Pearson Education, 8th Edition, **2023**

Nina Godbole & Sunit Belapure, "**Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives**", Wiley, 1st Edition, **2011**

Scott Barman, "**Writing Information Security Policies**", New Riders Publishing, 1st Edition, **2002**

Reference Books

Brian Underdahl, "**Cybersecurity for Dummies**", Wiley Publishing Inc , 1st Edition, **2011**

Joseph Steinberg, Kevin Beaver, Ira Winkler, "**Cybersecurity All-in-One For Dummies**", John Wiley and Sons Inc, 1st Edition, **2023**

Behrouz A. Forouzan, Debdeep Mukhopadhyay, "**Cybersecurity All-in-One For Dummies**", Mcgraw Hill Education, 3rd Edition, **2015**

Mode of Evaluation : Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test

Recommended by Board of Studies :	23-05-2025
Approved by Academic Council : No. 78	12-06-2025

Course Code	Course Title	L	T	P	C
MACSE637	Digital Forensics	3	0	0	3
Pre-requisite	Nil	Syllabus Version			
		1			

Course Objectives

1. To introduce the fundamental concepts, challenges, and methodologies in digital forensic investigations.
2. To enable the students to acquire knowledge on various data acquisition, recovery processes in disk forensics and techniques used in memory forensics, Disk Forensics, Network, Mobile and Cloud Forensics.
3. To make usage of appropriate tools for conducting forensic investigations in network, mobile, cloud environment and malware analysis with vulnerability assessment.

Course Outcomes

1. Understand the concepts, challenges, and methods of digital forensic investigations.
2. Create the disk images, recover deleted files and extract hidden data in disk forensics.
3. Examine and extract forensic artifacts from memory forensics.
4. Demonstrate the forensic tools to investigate network traffic, mobile devices (Android and IOS), cloud-based systems, conduct effective malware analysis and assess the system vulnerabilities.

Module:1	Digital Forensics Essentials	7 hours

Digital forensics and its facts - Trends and Challenges in digital forensics -Digital forensics approaches and best practices - Types of Cyber-crimes - Digital evidence and rules of evidence - Forensic readiness planning and business continuity - roles and responsibilities of a forensic investigator - Legal compliance, Forensic Investigation process.

Module:2	Disk Forensics	9 hours

Data Acquisition - Types and format of Data Acquisition – Methodology - Logical Structure of a Disk - Booting Process of Windows and Linux Operating Systems - File Systems of Windows and Linux Operating Systems - Partition Structure - Analyzing File Systems using Autopsy and The Sleuth Kit - SMART Metrics -File and Data Carving.

Module:3	Memory Forensics	9 hours

Acquisition and Analysis with tools: WinPmem, LiME and Memoryze - Collection of Volatile and Non-volatile Information in windows and Linux systems - Filesystem Images Analysis using The Sleuth Kit, Memory and Registry Analysis - Memory Forensics using Volatility and PhotoRec - Examine Cache - Cookie and History Recorded in Web Browsers- Examining Windows Files and Metadata- Timestamps- Temporary files- Deleted Content.

Module:4	**Network, Mobile and Cloud Forensics**	**9 hours**
Data Acquisition – PCAP Analysis- File extraction from Network- Geolocation with Wireshark- NetFlow Logs and Analysis- Connected Devices- Statistical Analysis- Protocol and Connection Architecture- Encrypted Network Activity Classification- Identifying Network Beacons using historical Network Data with RITA- Investigating Network traffic- Mobile Forensics: Acquisition Procedures for Mobile, Equipment and Tools- Cloud Forensics: Legal and Technical Challenges- Acquisition and Investigation.		
Module:5	**Malware Analysis**	**9 hours**
Understanding Malware - Techniques to spread malware- Types of Malware analysis - Acquiring and handling Malware Samples - Virtual environment and Analysis - Static and Dynamic Analysis- Analysis of Malware Behavior on System Properties and on Network - Sandbox with Cuckoo- Volatility Framework- Capa, Code Analysis.		
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture Hours:	**45 hours**
Text Book(s)		
Greg Gogolin, "**Digital Forensics Explained**", CRC Press, Taylor & Francis Group, LLC, Boca Raton. , 2nd Edition, **2021** Bill nelson, Amelia Phillips, Christopher steuart, "**Guide to Computer Forensics investigations processing of digital Evidence**", Cengage Learning, 5th Edition, **2015**		
Reference Books		
Adam Tilmar Jakobsen, "**Practical Cyber Intelligence: A Hands-on Guide to Digital Forensics**", Wiley, 1st Edition, **2024** Gerard Johansen, "**Digital Forensics and incident response: incident response techniques and procedures to respond to modern Cyber Threats**", Packt publishing, 2nd Edition, **2020**		
Mode of Evaluation : Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Seminar		
Recommended by Board of Studies :	23-05-2025	
Approved by Academic Council : No. 78	12-06-2025	

Course Code	Course Title	L	T	P	C
MACSE638	Software Engineering and Modeling	3	0	0	3
Pre-requisite	Nil	**Syllabus Version**			
		1			

Course Objectives

To understand software engineering fundamentals and apply agile methods and modeling techniques for effective project planning and requirement analysis
To design, build, and test reliable web and mobile applications using core principles and patterns.
To evaluate software measurements and management methods to estimate effort, assess risks, improve processes, and ensure quality during development

Course Outcomes

Apply software engineering principles and development frameworks to design solutions for real-world applications.
Develop and document functional and non-functional requirements using structured analysis and modeling techniques.
Design and implement robust software architectures and user interfaces by applying suitable design patterns and models
Execute effective testing strategies and manage software quality using industry-standard metrics and project management practices

Module:1	**Foundations of Software and Engineering Practices**	**8 hours**

The Nature of Software- The Software Process- Software Engineering Practice- Defining a Framework Activity- Prescriptive process Models- Product and Process- Agility and Cost of Change- Agile Process- Scrum- Other Agile Frameworks- Resource Estimation- Core Principles- Principles that guide each framework activity.

Module:2	**Modeling Software Requirements**	**9 hours**

Requirements Engineering- Establishing the groundwork- Building the Analysis model- Requirements Analysis- Scenario-Based Modeling- Class-Based Modeling- Functional Modeling- Behavioral Modeling.

Module:3	**Software Design Principles and Architectures**	**9 hours**

The Design Process- Design Concepts- The Design Model- Software Architecture- Architectural Styles- Architectural Design- User Interface Analysis and Design- User Interface Design- Design Evaluation- Mobile Development Life Cycle- Web Design Pyramid- Design Patterns- Architectural Patterns- User Interface Design Patterns- Mobility Design Patterns.		
Module:4	**Advanced Techniques in Software Testing**	**9 hours**
A strategic approach to software testing- Test-case Design- White-Box Testing- Black-Box Testing- Object-Oriented Testing- Integration Testing- Regression Testing- Validation Testing- Testing Patterns- Mobile Testing Guidelines- Web Application Testing- Strategies- Testing AI Systems- Testing Documentation		
Module:5	**Software Metrics, Management, and Process Improvement**	**8 hours**
Software Measurement- Product Metrics- Metrics for Testing- Metrics for Maintenance- Process and Project Metrics- Metrics for Software Quality- The Management Spectrum- Agile Estimation- Project Scheduling- Scheduling- Software Risks- The RMMM Plan- Software Maintenance- Software Evolution- The SPI Process- The CMMI.		
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture Hours:	**45 hours**
Text Book(s)		
Pressman, Roger S., & Maxim, Bruce R. , "**Software Engineering: A Practitioner's Approach**", McGraw Hill Education, 9th Edition, **2019**		
Reference Books		
Sommerville, I., "**Software Engineering.**", Pearson Education Limited, 10th Edition, **2015** Bruegge, B., &Dutoit, A. H., "**Object-Oriented Software Engineering: Using UML, Patterns, and Java** ", Prentice Hall., 3rd Edition, **2010**		
Mode of Evaluation : Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Seminar, Presentation		
Recommended by Board of Studies :	23-05-2025	
Approved by Academic Council : No. 78	12-06-2025	

Course Code	Course Title	L	T	P	C
MACSE639	DevOps	3	0	2	4
Pre-requisite	Nil	**Syllabus Version**			
		1			

Course Objectives

To explain the principles of DevOps culture and discuss its significance in improving software development and delivery processes
To enable the skills to use DevOps tools and technologies to automate tasks, streamline workflows, and enhance team collaboration.
To develop the knowledge in security, scalability, integration and deployment of modern software applications.

Course Outcomes

Articulate the core principles of SDLC and evolution of DevOps.
Implement CI/CD pipelines and security practices into the DevOps lifecycle to automate secure software delivery.
Utilize Infrastructure as Code (IaC) to provision and manage infrastructure.
Demonstrate code automation using industry-standard tools.
Deploy, manage and monitor containerized applications using Docker and Kubernetes.

Module:1	**INTRODUCTION TO DEVOPS**	**7 hours**

Introduction to Software Engineering - Traditional Process models - Evolution of DevOps - Principles - Practices - Cloud Infrastructure and Services- Virtual Machine (VM) Creation- DevOps in Cloud Environments- Cloud Service Providers: AWS, Azure and GCP.

Module:2	**DEVOPS ESSENTIALS**	**9 hours**

DevOps Lifecycle - Concepts of CI/CD- Overview of Version Control System: Git workflows - Working with Remote Repositories – Github- Implement CI/CD concepts using Git workflows and collaborate on remote repositories through GitHub - Security practices in DevOps

Module:3	**CONFIGURATION MANAGEMENT**	**9 hours**

Infrastructure as Code (IaC) - IaC Principles and Best Practices - Configuration Tools: Ansible, Puppet and Maven - Deploying Infrastructure using configuration tools - CI/CD Integration with IaC.

Module:4	**JENKINS**	**9 hours**

Jenkins Overview & Architecture - Installation & Basic Configuration - Creating and Configuring Jenkins Jobs - Pipeline – Plugins - Implement Jenkins pipeline script to automate code building and deployment.		
Module:5	**CONTAINERIZATION**	**9 hours**
Docker overview - Installing docker - Pulling images - Running images - Docker build and deployment, Exposing volumes and ports - Docker Compose- Kubernetes and its architecture - Performance monitoring and its tools		
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

K K Aggarwal. Yogesh Singh, "**Software Engineering**", New Age International Private Limited publications, 4th Edition, **2022**

Ekambar Kumar Singirikonda, "**DevOps Automation Cookbook**", BPB Publications, 1st Edition, **2024**

Reference Books

Rama Bedarkar, "**Agile Scrum: Improving Practices for Business Gains**", Wiley publications, 1st Edition, **2020**

Indicative Experiments

1. Cloud-based application deployment and management using AWS or Azure or GCP.	3 hours
2. Version Control Systems - Git: For version control and collaborative development.	3 hours
3. Integrating Git repository with Github	3 hours
4. Continuous Integration & Continuous Deployment (CI/CD) using: Jenkins (or) GitLab CI (or) CircleCI.	3 hours
5. Configuration management and automation (IaC) using Ansible (or) Puppet (or) Maven.	3 hours
6. Jenkins pipeline creation	3 hours
7. Deployment of software application using Docker	3 hours
8. Containerization and Orchestration of software application using Kubernetes	3 hours
9. Monitoring and Logging using Prometheus or Grafana	3 hours

10. Security in DevOps using OWASP ZAP (or) SonarQube	3 hours
Total Laboratory Hours:	**30 hours**
Mode of Evaluation : Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment	
Recommended by Board of Studies :	23-05-2025
Approved by Academic Council : No. 78	12-06-2025

Course Code	Course Title	L	T	P	C
MACSE640	Full Stack Application and Development	3	0	2	4
Pre-requisite	Nil	**Syllabus Version**			
		1			

Course Objectives

To enable students to design dynamic web applications using HTML, CSS, JavaScript, React, and Angular
To impart knowledge of backend development with Node.js, Express.js, and MongoDB for full-stack solutions
To develop skills to integrate front-end and back-end for seamless data handling, validation, and server communication

Course Outcomes

Understand and apply HTML, CSS, and JavaScript techniques to create structured, styled, and interactive web pages
Design and develop component-based applications using React with efficient state management, event handling, routing, and API integration
Implement and analyze AngularJS applications by utilizing templates, services, routing, and form validation through TypeScript
Construct and manage server-side applications and databases using Node.js and MongoDB, handling asynchronous data operations
Develop and secure web servers using ExpressJS, including RESTful APIs, session management, and database connectivity

Module:1	**Web Development Basics HTML CSS JavaScript**	**8 hours**

HTML Tags HTML Attributes, Meta Tags, Lists, Table, Images, forms- Introduction to CSS- CSS Selectors Properties and Values, CSS Box model, CSS Animation, CSS Frameworks- JavaScript Fundamentals- Objects and Datatypes, Classes and Functions, Document Object Model- Forms: form Element, Controls, Text Control Accessing a Form's Control Values, reset and focus Methods – Event handling in JavaScript: onload, onclick, onchange, onmouseover, onmouseout, onmouseup, onkeyup, onkeydown, onkeypress, onselect- Form Validation using Regular Expression - Hosting a website.

Module:2	**Client Side Scripting ReactJS**	**8 hours**

Introduction to React, Architecture and Installation, JSX and Components, Props and State Management, React JS styling-Inline JSON format and inline JavaScript literal notation, using modules and external style sheet, Express REST APIs, Event

Management, Modularization and Webpack, React Hooks, React Routing, Client and Form Programming, Server-side rendering - Deployment using GitHub.		
Module:3	**Client Side Scripting AngularJS**	**9 hours**
Typescript – Invoking Angular- Angular Components – Expressions – Templates and Data binding – Directives, changing the DOM with Directives, dynamic views – services and routing-Forms-Directives and HTML Validation- Communicating with Servers-Working with RESTful Resources.		
Module:4	**Server Side Scripting NodeJS MongoDB**	**9 hours**
Basics of Node JS – Installation – 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 -Building MongoDB Environment – User accounts – Access control – Administering databases – Managing collections – Connecting to MongoDB from Node.js –Applications		
Module:5	**Backend Web Application Framework ExpressJS**	**9 hours**
Setting up ExpressJS - Restful APIs - HTTP Methods and Routing –Templating - Data Handling and Database - Form Handling - Authentication using Cookies and Sessions - Implementing Express in Node.js.		
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

Martin Krause, "**The Complete Developer Master the Full Stack with TypeScript, React, Next.js, MongoDB, and Docker**", No Starch Press, 1st Edition, **2024**

Nabendu Biswas, "**Ultimate Full-Stack Web Development with MERN: Design, Build, Test and Deploy Production-Grade Web Applications with MongoDB, Express, React and NodeJS**", Orange Education Pvt Ltd, 1st Edition, **2023**

Reference Books

Daniel Bugl , Matthias Zronek, "**Modern Full-Stack React Projects**", Packt Publishing, 1st Edition, **2024**

Mykyta Chernenko, Artem Korchunov, "**Full-Stack Web Development with TypeScript 5**", Packt Publishing, 1st Edition, **2024**

Philip Ackermann, "**Full Stack Web Development: The Comprehensive Guide**", Shroff Rheinwerk Computing, 1st Edition, **2023**

Vasan Subramanian, "**Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React, and Node**", Apress, 2nd Edition, **2019**

Indicative Experiments	
1. You are leading the front-end development for an educational game platform aimed at teaching environmental awareness. One of the modules is a “Plant a Tree” simulation. In this simulation: I. When a user clicks on a patch of soil (an image), a seed appears, and a growth animation begins (the seed sprouting into a small tree). II. The sprouting uses a CSS animation that changes height, opacity, and color shade over time. III. After the sprouting animation completes, a message panel slides in (slide-in animation) congratulating the user and offering a random environmental tip. IV. If the user clicks again on the soil, a new tree must grow, and a new message must appear, but the old tree should fade out first before the new seed is planted and the cycle restarts. V. All transitions must feel smooth and natural, without overlapping animations or sudden jumps. VI. Different types of trees (with different growth animations) should randomly be selected for each new planting. VII. If the user double-clicks rapidly, the system should queue the actions properly so that animations still flow sequentially, without interruption.	3 hours
2. Design a Water Intake Tracker using HTML, CSS, and JavaScript where users can log their daily water consumption. The interface should include a form where users can input the amount of water they have consumed (in liters), select the time of day (Morning, Afternoon, or Evening), and click a "Log Water" button to add the intake. Once the "Log Water" button is clicked, the application will update a progress bar that visually represents the percentage of the daily goal (3 liters) achieved based on the user’s input. The progress bar color should change depending on the time of day when the intake is logged: Morning should set the progress bar color to blue, Afternoon to yellow, and Evening to purple.	3 hours

If the user reaches or exceeds the goal of 3 liters, the progress bar should turn green. In addition to the progress bar, the application should display the total water intake logged and provide a summary list of all the water intakes entered, showing the amount and the time of day selected for each entry. The solution should be implemented using plain HTML, CSS, and JavaScript, with no use of a database or persistent storage, ensuring everything resets when the page is reloaded. The application must dynamically update the total intake and the progress bar color according to the user's input, and it should also calculate and show the percentage of the daily goal reached after each entry.	
3. Design a form using ReactJS JSX with the parameters for Employee no (Number Field from 1 to 100), employee name (Text Field), designation (Select field with minimum 4 options) and basic pay (Text Field). Obtain the above values and calculate 40% DA (Text Field), 70% HRA (Text Field) on the basic as the allowances. Also keep 20% PF (Text Field) and 30% LIC (Text Field) on the basic pay as the deductions. Read all the above inputs. Using functional component in ReactJS, the Net pay has to be computed and is displayed in a text field. And, print in a separate text field "Good" if the computed net pay is >15000, else print "bad". These 2 outputs are to be generated by triggering the event on a button namely "Submit". Formula for calculating net pay=basic pay + DA + HRA - (PF + LIC) Note: The net pay is to be printed in first text field and the message "good" or "bad" (based on the above condition) is to be printed in second text field.	3 hours
4. Design a ReactJS application that allows users to track their daily expenses. The interface should have a form where users can input the expense description in a text field, amount in a text field and category (such as food, entertainment, or bills) in a drop down/select box. Once the "Add Expense" button is clicked, the expense should be added to a list displayed below the form, showing the description, amount, and category of each expense. The application should also include a button to calculate the total expenses for the day, which will be displayed at the bottom of the expense list. Use a class component and handle the events for adding expenses and calculating the total using arrow functions. Make sure that the expense list is updated dynamically after each addition and the total expenses are calculated and displayed. Also, use suitable CSS based ReactJS code provide the styling.	3 hours
5.	3 hours

Design an Angular application that allows users to manage and track a list of books in a library. The interface should contain a form where users can input the book title, author, genre, and year of publication. There should be a button that, when clicked, adds the book to a list displayed below the form, showing the book's title, author, genre and year of publication. The application should also include a button to search the list of books by genre or author and only the books that match the search criteria should be displayed. Use Angular components to handle the form, display the book list, and search functionality. Implement a service to manage the data, including adding books and filtering by search criteria. The list should be dynamically updated after each book addition and the search results should reflect any changes immediately. Use Angular directives to handle form validation and ensure that all fields are filled out before submission. Make sure the application is styled with CSS to make it visually appealing.	
6. It is planned to build a Smart City Dashboard called CityPulse using AngularJS. The dashboard should display real-time data from various city services through a RESTful API: • Traffic congestion updates (GET /traffic) • Weather data (GET /weather) • Waste collection schedules (GET /waste) • Citizen complaints (GET /complaints) • Submit a new complaint (POST /complaints) • Update a complaint status (PUT /complaints/:id) • Delete a resolved complaint (DELETE /complaints/:id) Requirements: a. Each section of the dashboard (traffic, weather, waste, complaints) should update independently without refreshing the whole page. b. When users submit a complaint, it should instantly appear in the “Citizen Complaints” section.	3 hours

c. When authorities update or resolve complaints, the dashboard must automatically reflect the changes. d. The app must show loading indicators for each individual section. e. If any section fails to load (e.g., waste schedule API is down), it should only affect that section, not the whole dashboard. f. Implement retry logic for important data like traffic and weather updates if the first attempt fails.	
7. Design a Customer Service Ticket Management System using Node.js and MongoDB. The system should have a home page with navigation links such as "Create Ticket," "Update Ticket Status," "Search Ticket by Customer Name," and "Delete Old Closed Tickets." The "Create Ticket" form should collect customer name, issue description, and status (open/closed), and upon submission, a unique Ticket ID should be generated automatically and stored in a MongoDB database along with the ticket information. The system should allow users to update the status of a ticket by entering the Ticket ID and selecting a new status. A search functionality should enable users to retrieve and display ticket details based on the customer name. The "Delete Old Closed Tickets" feature should allow for the deletion of tickets that have been closed for more than 30 days by querying the MongoDB database and removing those entries. Note: The entire system should be implemented using Node.js for backend functionality and MongoDB for data storage, without the use of other frameworks like Express.	3 hours
8. Using Node.js create a custom event emitter named StudentEmitter that emits a 'registered' event whenever a new student is registered. The event handler should display the student's name and registration number. Additionally, create another event 'courseSelected' that logs the name of the course selected by the student. Use the .on() method to register listeners for both events and ensure that when the events are emitted with appropriate data, the handlers respond correctly by displaying formatted output in the console.	3 hours

9. ABC university is one of the renowned universities in the world. They are in need to deploy a full-stack University Student Portal called CampusConnect using ExpressJS for the backend and MongoDB as the database. The system must handle and fulfill these requirements: a. Student Registration and Login • Students must be able to register and log in through responsive, validated form. • Registration must require a strong password (minimum length, complexity checks). • Implement email confirmation after registration. Users must verify their email before they can log in. b. Session Management • After login, users must stay authenticated using express-session and secure cookies. • Sessions should expire automatically after 30 minutes of inactivity (using session idle timeout). c. Database Models • Design three collections: students, courses, and enrollments. • Each enrollment document must link a student to a course with timestamps. d. Course Enrollment • Logged-in students can enroll in available courses via a dynamic form. • Prevent students from enrolling in the same course twice. • Enrollment history must be viewable in a dashboard sorted by most recent. e. Security Measures • Protect all sensitive routes using authentication middleware. • Redirect unauthenticated users to the login page when they attempt to access protected resources.	3 hours

• Use helmet.js to set secure HTTP headers. • Sanitize all user inputs to prevent NoSQL injections and XSS attacks.	
10. Assume that the following employee collection sample: { "_id": 1, "name": "Alice", "role": "Developer", "projects": ["Alpha", "Beta"], "salary": 60000 } { "_id": 2, "name": "Bob", "role": "Manager", "projects": ["Gamma"], "salary": 80000 } { "_id": 3, "name": "Charlie", "role": "Developer", "projects": ["Delta"], "salary": 55000 } Perform the following using MongoDB: a. Retrieve all Developers working on the "Alpha" project b. Promote all Developers to "Senior Developer" c. Remove the "Beta" project from everyone's project list d. Increase salary of all Developers by 10% e. Find employees working on more than one project	3 hours
Total Laboratory Hours:	**30 hours**
Mode of Evaluation : Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment, Seminar	
Recommended by Board of Studies :	23-05-2025
Approved by Academic Council : No. 78	12-06-2025