PERIYAR UNIVERSITY

PERIYAR PALKALAI NAGAR SALEM 636 011

MASTER OF SCIENCE IN COMPUTER SCIENCE

SEMESTER PATTERN

Under Choice Based Credit System

REGULATIONS AND SYLLABUS

FOR AFFILIATED COLLEGES

(Effective from the Academic year 2026 - 2027 onwards)

PERIYAR UNIVERSITY, SALEM – 636 011

M.Sc. COMPUTER SCIENCE

REGULATIONS AND SYLLABUS – 2026 – 2027

(Affiliated Colleges)

Effective from the Academic year 2026 - 2027 onwards

MASTER OF SCIENCE IN COMPUTER SCIENCE (M.Sc. Computer Science)

The Master of Science in Computer Science is a postgraduate programme designed to provide students with an advanced and comprehensive understanding of the theoretical foundations, computational principles, and practical applications that underpin modern computing systems. The programme builds upon undergraduate knowledge of computer science and mathematics, extending it into specialized areas such as algorithms, data structures, operating systems, computer networks, database management systems, software engineering, artificial intelligence, and emerging technologies. It aims to develop analytical and problem-solving skills among students, enabling them to design, develop, and evaluate complex software and hardware systems that meet real-world requirements.

In addition to strengthening core technical competencies, the programme emphasizes research orientation, innovation, and hands-on learning through laboratory work, projects, and seminars. Students are encouraged to explore contemporary domains including cloud computing, cybersecurity, machine learning, and mobile application development, thereby preparing them for diverse career opportunities in the IT industry, research organizations, and academia. By the end of the programme, graduates are expected to possess the technical expertise, critical thinking ability, and ethical grounding necessary to contribute effectively to the rapidly evolving field of computer science.

A) OBJECTIVE OF THE PROGRAMME

To develop the Post Graduate in Computer Science with strong computational knowledge, industry-relevant software engineering skills, and research aptitude, enabling employment in research and development units of industries, software organizations, and academic institutions.

B) CONDITIONS FOR ADMISSION

A candidate who has passed B.Sc. Computer Science / BCA / B.Sc. Computer Technology / B.Sc. Information Science / B.Sc. Information Technology / B.Sc. Data Analytics / B.Sc. Data Science / B.Sc. Artificial Intelligence and Data Science / B.Sc. Cyber Security / B.Sc. Internet of Things degree of this University, or any other degree accepted by the Syndicate as equivalent thereto, subject to such conditions as may be prescribed, shall be permitted to appear and qualify for the M.Sc. Computer Science degree examination of this University after a course of study of two academic years.

C) DURATION OF THE PROGRAMME

The programme for the degree of Master of Science in Computer Science shall consist of two academic years divided into four semesters.

D) EXAMINATIONS

The examination shall be of three hours duration for each course at the end of each semester. A candidate failing in any course(s) will be permitted to appear in the subsequent examination. The practical / project work should be individual work, and the University examination for practical / project work will be conducted jointly by the internal and external examiners at the end of the respective semester.

E) PROGRAMME OUTCOMES (POs) AND PROGRAMME SPECIFIC OUTCOMES (PSOs)

PO1: Computational Knowledge: Apply advanced algorithmic structures, mathematical foundations, and data engineering principles to design robust, scalable computational systems solutions.

PO2: Problem Analysis & Design: Identify, analyze, and formulate complex computing problems, designing full-stack frameworks, microservices, and secure distributed protocols to meet industry specifications.

PO3: Modern Tool Usage: Master advanced computational tools and environments including Python Data Engineering libraries, MLOps orchestration frameworks, Figma UI/UX engines, and Quantum simulation suites.

PO4: Professional Ethics & Human Rights: Adhere to regulatory, corporate compliance, data privacy laws, and ethical standards in AI deployment, cybersecurity, and advanced technical research documentation while valuing fundamental human rights.

PO5: Innovation & Lifelong Learning: Engage in continuous professional evolution by executing independent technical research projects, full-stack application deployment, and adaptation to transformative emerging technologies like LLMs.

PSO1: Programming & Software Development: Design, develop, and deploy efficient, scalable software solutions using modern programming languages, frameworks, and software engineering practices.

PSO2: Systems & Emerging Technologies: Apply knowledge of distributed systems, databases, artificial intelligence, machine learning, and cloud-based emerging technologies to solve real-world computing problems.

PSO3: Research & Innovation: Conduct independent technical research, document findings professionally, and adapt to evolving technologies through continuous self-directed learning.

PSO4: Professional & Industry Readiness: Demonstrate ethical, collaborative, and industry-relevant competencies, including proficiency with modern computational tools, for employability and entrepreneurship.

F) METHODS OF EVALUATION AND METHODS OF ASSESSMENT

Methods of Evaluation:

Component	Marks
Continuous Internal Assessment Test	10 Marks
Assignments / Seminar / Quiz	10 Marks
Attendance and Class Participation	5 Marks
Internal Total	25 Marks
End Semester Examination	75 Marks
Grand Total	100 Marks

Methods of Assessment (Bloom's Taxonomy Levels):

Level	Description
Remembering (K1)	The lowest level of questions requires students to recall information from the course content, define terminologies, list core properties, or state

	foundational formulas without manipulation.
Understanding (K2)	Students comprehend, organize, compare, and interpret facts and ideas in their own words, going beyond recall to explain workflows, illustrate concepts, or contrast technical architectures.
Application (K3)	Students apply concepts learned in the classroom to solve problems, using structured principles to write code, execute queries, or calculate precise analytical outcomes.
Analyze (K4)	Students break a problem into its component parts, identify reasons, causes, or dependencies, and investigate system failures, isolate bugs, or dissect complex structural relationships.
Evaluate (K5)	Students make reasoned judgments on the value of an idea, design, or solution, engaging in decision-making and problem-solving where questions may not have a single right answer.
Create (K6)	Students engage in original thinking, assembling diverse elements into a new functional architecture, designing custom full-stack frameworks or novel computing solutions.

The Passing minimum shall be 50% out of 75 marks (38 marks) in the End Semester Examination for theory courses.

G) STRUCTURE OF M.Sc. (COMPUTER SCIENCE) PROGRAMME

Course Code	Title of the Course	Credits	Hours		Maximum Marks		
			Theory	Practical	CIA	EA	Total
FIRST SEMESTER							
26MCS01	Core Paper I – Algorithmic Design and Complex Data Structures	5	6		25	75	100
26MCS02	Core Paper II – Mathematical Foundations of Computer Science & Automata	5	6		25	75	100
26MCSE01 /02	Elective Paper I (Prompt Engineering / Ethical Hacking and Cyber Security)	3	6		25	75	100

26MCSP01	Practical I – Algorithmic Paradigm Lab	4		4	40	60	100
26MCSP02	Practical II – Mathematical Foundations Lab (Using C++)	4		4	40	60	100
26MCSS01	Skill Enhancement Course I – UI/UX and Prototyping Tools (FIGMA)	2	4		25	75	100
Total		**23**	**22**	**8**	**600**		
SECOND SEMESTER							
26MCS03	Core Paper III – Distributed Database Systems and NoSQL Paradigms	5	6		25	75	100
26MCS04	Core Paper IV – Java Architecture and Frameworks	5	6		25	75	100
26MCSE03 /04	Elective Paper II (Advanced Distributed Systems / Deep Algorithmic Learning)	3	6		25	75	100
26MCSP03	Practical III – Distributed Database Systems & NoSQL Lab	4		4	40	60	100
26MCSP04	Practical IV – Java Architecture and Frameworks Lab	4		4	40	60	100
26MCSN01	Non-Major Elective Course I – Fundamentals of Internet & Office Automation	2	3		25	75	100
26MHR01	Fundamentals of Human Rights	1	1		100	-	100 (Internal)

Total		**24**	**22**	**8**	**700**		
THIRD SEMESTER							
26MCS05	Core Paper V – Python for Data Engineering and Automation	5	6		25	75	100
26MCS06	Core Paper VI – MERN Full Stack Development	5	6		25	75	100
26MCSE05 /06	Elective Paper III (Foundation of Quantum Computing Techniques / Large Language Models)	3	6		25	75	100
26MCSP05	Practical V – Python for Data Engineering and Automation Lab	4		4	40	60	100
26MCSP06	Practical VI – MERN Full Stack Development Lab	4		4	40	60	100
26MCSS03	Skill Enhancement Course Lab – Figma Design Lab	2		2	40	60	100
26MCSN02	Non-Major Elective Course II – Web Design for Beginners (HTML & CSS)	2	2		25	75	100
26MCSI01	Contextual Learning: Internship	1			-	-	Commended/ Highly Commended
Total		**26**	**20**	**10**	**700**		
FOURTH SEMESTER							
26MCS07	Core Paper VII – Artificial Intelligence & MLOps	5	6		25	75	100
26MCSP07	Practical VII – Artificial	4		4	40	60	100

	Intelligence & MLOps Lab						
26MCSPR01	Project with Viva Voce	5		16	50	150	200
26MCSS02	Skill Enhancement Course II – Research Documentation (LaTeX)	2	4		25	75	100
26MCSEX01	Extension Activity	1			100	-	100 (Internal)
Total		**17**	**10**	**20**	**600**		

H) ELECTIVES, SKILL ENHANCEMENT, AND NON-MAJOR ELECTIVE COURSES LIST

Core Papers:

26MCS01 – Core Paper I – Algorithmic Design and Complex Data Structures
26MCS02 – Core Paper II – Mathematical Foundations of Computer Science and Automata
26MCS03 – Core Paper III – Distributed Database Systems and NoSQL Paradigms
26MCS04 – Core Paper IV – Java Architecture and Frameworks
26MCS05 – Core Paper V – Python for Data Engineering and Automation
26MCS06 – Core Paper VI – MERN Full Stack Development
26MCS07 – Core Paper VII – Artificial Intelligence and MLOps

Practical Papers:

26MCSP01 – Practical I – Algorithmic Paradigm Lab
26MCSP02 – Practical II – Mathematical Foundations Lab (Using C++)
26MCSP03 – Practical III – Distributed Database Systems & NoSQL Lab
26MCSP04 – Practical IV – Java Architecture and Frameworks Lab
26MCSP05 – Practical V – Python for Data Engineering and Automation Lab
26MCSP06 – Practical VI – MERN Full Stack Development Lab
26MCSP07 – Practical VII – Artificial Intelligence & MLOps Lab

Elective Course – I (choose one):

26MCSE01 – Prompt Engineering
26MCSE02 – Ethical Hacking and Cyber Security

Elective Course – II (choose one):

26MCSE03 – Advanced Distributed Systems and Modern Architectures
26MCSE04 – Deep Algorithmic Learning

Elective Course – III (choose one):

26MCSE05 – Foundation of Quantum Computing Techniques
26MCSE06 – Large Language Models (LLM)

Skill Enhancement Courses:

26MCSS01 – UI/UX and Prototyping Tools (FIGMA) – Semester I
26MCSS03 – Figma Design Lab (Skill Enhancement Course Lab) – Semester III
26MCSS02 – Research Documentation and Professional Layout Design (LaTeX) – Semester IV

Non-Major Elective Courses (offered to other departments):

26MCSN01 – Fundamentals of Internet and Office Automation – Semester II
26MCSN02 – Web Design for Beginners (HTML & CSS) – Semester III

I) EXTERNAL ASSESSMENT QUESTION PAPER PATTERN (THEORY)

Time: 3 Hours | Maximum Marks: 75

- Part A: 15 × 1 = 15 Marks – Answer all questions (three from each unit, Multiple Choice Questions).
- Part B: 2 × 5 = 10 Marks – Answer any two questions, one from each unit.
- Part C: 5 × 10 = 50 Marks – Answer all questions, one question from each unit (either/or type).

The Passing minimum shall be 50% out of 75 marks (38 marks) in the End Semester Examination for theory courses.

J) CONTINUOUS INTERNAL ASSESSMENT FOR PRACTICAL

Test 1	15 Marks
Test 2	15 Marks
Record	10 Marks
Total	40 Marks

Record Note must be compulsorily submitted while attending the Practical Examination. No passing minimum is prescribed for the Record.

K) EXTERNAL ASSESSMENT QUESTION PAPER PATTERN (PRACTICAL)

Exam Duration: 3 Hours; Maximum Marks: 60. Two questions, with or without sub-sections, will be given for the practical examination, each chosen from the question bank prepared by the examiner(s).

Distribution of Marks per Question (30 Marks each):

- Problem Understanding: 5 Marks
- Program Writing: 10 Marks
- Debugging: 10 Marks
- Correct Results: 5 Marks

L) ASSESSMENT OF PROJECT WORK

Component	Marks
Continuous Internal Assessment	Review I – 25 Marks; Review II – 25 Marks (Total: 50 Marks)
External Assessment	Evaluation & Viva-Voce (Jointly) – 150 Marks

Common instructions for the project work:

- The candidate should submit the filled-in format to the Department for approval during the first week of Semester IV.
- The project should be periodically reviewed.
- The student should submit three copies of the project work.
- Students may use a presentation during the Viva-Voce examination.

M) PASSING MINIMUM

A candidate shall be declared to have passed in the Theory / Practical / Project Work examination if the candidate secures not less than 50% of marks in the End Semester Examination and also in the aggregate of the prescribed marks. Submission of a duly certified Record Note Book is compulsory for the Practical examinations.

N) CLASSIFICATION OF SUCCESSFUL CANDIDATES

Candidates who secure not less than 75% of the aggregate marks in the whole examination shall be declared to have passed the examination in the First Class with Distinction, provided they pass all the examinations prescribed for the programme at the first appearance. Candidates, other than the above, who secure not less than 60% of the aggregate marks in the whole examination, shall be declared to have passed the examination in First Class. The remaining successful candidates shall be declared to have passed in Second Class. Candidates who pass all the examinations prescribed for the programme at the first instance and within a period of two academic years from the year of admission alone are eligible for University Ranking.

O) MAXIMUM DURATION FOR COMPLETION OF THE PROGRAMME

The maximum duration permitted for completion of the programme shall be three academic years from the date of normal commencement of the programme.

P) COMMENCEMENT OF THIS REGULATION

These Regulations shall take effect from the academic year 2026-2027, that is, for students who are admitted to the first year of the programme during the academic year 2026-2027 and thereafter.

ANNEXURE - I

PERIYAR UNIVERSITY

Name of the College :

Programme :

Name of the Student :

Register Number :

Title of the Project Work :

Address of Organization / Institution :

Name of the External Guide :

Designation :

Place:

Date: Signature of External Guide

(With seal)

Name of the Internal Guide :

Qualification :

Teaching Experience :

Place:

Date: Signature of Internal Guide

ANNEXURE - II

PROJECT DOCUMENTATION CURRICULUM FRAMEWORK

DEFINITIONS & LOGICAL SYLLABUS BREAKDOWN

Preliminary Pages (i to vii)

- **College Bonafide Certificate:** Official institutional document signed by the Project Supervisor, Head of the Department (HOD), and the Principal, formally certifying the originality of the research work completed by the candidate to fulfil degree regulations.
- **Company Attendance / Internship Certificate:** Mandatory corporate document issued on official industry letterhead validating the student's external technical training execution, attendance timelines, and project module completion.
- **Declaration:** Formal statement signed directly by the student affirming that the software application and corresponding documentation are entirely original works and have not been submitted for any other academic degree.
- **Acknowledgement:** Dedicated section expressing formal gratitude to the institutional administration, faculty guides, industry supervisors, and supporting individuals who provided necessary infrastructure and academic counselling.
- **Synopsis:** Comprehensive executive summary of the technical problem statement, the software development stack, structural architecture components, and definitive project outcomes.
- **List of Figures:** Systematic index mapping every graphic element, architecture block diagram, UML chart, and user interface snapshot alongside the exact page allocation.

- **List of Tables:** Structured index referencing data frameworks, technical specification grids, database schemas, field parameters, and system testing evaluation logs with cross-referenced page entries.

Chapter 1: Introduction

- **1.1 Project Overview:** High-level introduction clarifying the problem domain boundaries, intended target user audience, real-world utility, and overarching system design goals.
- **1.2 Organization / Company Profile:** Background details of the host organization or department, its field of operation, commercial domain, and the operational setting associated with the internship deployment.
- **1.3 System Specifications:** The engineering constraint envelope dictating hardware capability and developer dependency environments.
- **1.3.1 Hardware Configuration:** Infrastructure requirements including memory allocation, processor throughput, and storage subsystem requirements.
- **1.3.2 Software Specification:** Compilation variables including operating environment, runtime platform, build tools, and the database engine used.

Chapter 2: System Analysis

- **2.1 Existing System Description:** Documents the precise logic of current manual or legacy workflows in place before implementation.
- **2.2 Drawbacks and Limitations:** Critical diagnosis of data transmission issues, processing latency, storage redundancy, and access control gaps in the current system.
- **2.3 Proposed System Description:** Describes the automated solution and how procedural operations are modernized, hosted, and resolved through structured data control.
- **2.4 Core Features & System Advantages:** Tabulates functional improvements such as faster queries, automated alerts, secure authentication, and dynamic tracking.

Chapter 3: System Design

- **3.1 Input Interface Design:** Defines the form models, data capture fields, event triggers, and front-end validation rules.
- **3.2 Output Interface Design / Report Formats:** Lays out presentation layers, dashboards, downloadable document structures, and reporting formats.
- **3.3 Code Design & Methodology:** Defines the modular programming model, backend architecture style (e.g. MVC), class relationships, and file organization.
- **3.4 Logical Database Design (ER Diagrams):** Constructs the relationship schema mapping table definitions, primary/foreign keys, cardinalities, and normal forms.

Chapter 4: System Development & Testing

- **4.1 Description of Functional Modules:** Breaks down the software into distinct operational units, showing each module's pipeline.

- **4.2 Testing Strategies:** Establishes code compliance validation workflows, covering Unit Testing (independent verification of functions and routines using test data) and Integration Testing (verifying data exchange across application boundaries and endpoints).

Chapter 5: Conclusion & Future Enhancements

- **5.1 Conclusion:** Evaluates whether the project's objectives, as defined during the initial analysis phase, were successfully achieved.
- **5.2 Scope for Future Enhancements:** Identifies potential extensions, such as further automation, mobile scaling, or migration to a microservices architecture.

Chapter 6: Bibliography / References

- **Bibliography:** An indexed list of textbooks, research papers, academic manuals, and online documentation, cited in a consistent reference style.

Appendices

- **A. Data Flow Diagrams (DFD) / UML Diagrams:** Full-page diagrams charting contextual, Level-1, and Level-2 system dependencies, or class/use-case mappings.
- **B. Database Table Structures & Field Specs:** Dictionaries detailing field boundaries, data types, default values, keys, and storage parameters for every table.
- **C. Sample Source Code Snippets:** Highlights of key backend routes, core algorithms, or data security implementations.
- **D. Sample Input Screens:** Screen captures showing filled UI entry views during standard system interactions.
- **E. Sample Output Reports:** Visual reporting logs showing generated summaries, dashboards, and transactional reports.

**Subsections may be selectively adapted, omitted, or substituted with the guide's prior approval if a particular structural phase is not technically relevant to the core development architecture.*

ANNEXURE - III

A. Format of the Title Page

TITLE OF THE PROJECT WORK

A Project Work submitted in partial fulfilment of the requirements for the degree of

Master of Science in Computer Science

to the

Periyar University, Salem – 636 011

Submitted by

Name of the Student

Reg. No.

Under the Guidance of

Name of the Guide
(Designation, Name of the Department)

College emblem

Name of the Department

College Name
(Affiliated to Periyar University)
Place with Pin Code

Month – Year

B. Format of the Certificate

College Name
(Affiliated to Periyar University)
Place with Pin Code

College emblem

This is to certify that the Project Work entitled **Title of the Project** submitted in partial fulfilment of the requirements of the degree of **Master of Science in Computer Science** to the Periyar University, Salem is a record of Bonafide work carried out by **Name of the Student**, **Reg. No.** _____ under my supervision and guidance.

Head of the Department Internal Guide

Submitted for Viva-Voce Examination held on _____ at

Name of the College, Place with Pin Code.

External Examiner Internal Examiner

SEMESTER - I

CORE – I - ALGORITHMIC DESIGN AND COMPLEX DATA STRUCTURES

Course Objectives:

The main objectives of this course are to:

1. Understand foundations of linear structures, XOR linked layouts, and space-time complexity bounds.
2. Design advanced balance-tree algorithms and indexing matrices for micro-indexing frameworks.
3. Optimize sorting paradigms and priority elements using complex tree structures and heap properties.
4. Formulate graph layouts, network matrices, and shortest path traversals for distributed setups.
5. Model programmatic instances across polynomial complexities using standard structural paradigms.

Expected Course Outcomes:

On the successful completion of the course, student will be able to:

1	Analyze and contrast asymptotic spaces and amortized complexity profiles of linear / non-linear data structures.	K1, K2
2	Evaluate tree structures including self-balancing models, multi-way indexing paths, and prefix lookups.	K2, K3
3	Construct specialized priority queues and compute amortized run-times of multi-way heaps.	K3,K4
4	Implement complex graph optimization algorithms and network flow models for resource allocation.	K4, K5
5	Standardize advanced paradigms to resolve algorithmic completeness limits using formal analysis models.	K5,K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Unit:1	**Foundations of Algorithms & Linear Structures**	**15 hours**

Asymptotic Notations (Big-O, Omega, Theta) - Amortized Analysis - Space-Time Complexity Trade-offs - Skip-Lists Operations - XOR Linked Lists - Double-Ended Queues

(Deque) Implementation.		
Unit:2	**Advanced Tree Structures**	**15 hours**
AVL Trees & Balancing Rotations - Red-Black Trees Insertion & Deletion - B-Trees & B+ Trees for Database Indexing - Radix Tries & Prefix String Lookups.		
Unit:3	**Heap Structures & Priority Queues**	**15 hours**
Binary Min-Max Heaps - Priority Queue Implementations - Leftist Heaps & Merge Optimization - Fibonacci Heaps Amortized Performance - Binomial Heaps - Skew Heaps.		
Unit:4	**Graph Algorithms & Optimization**	**15 hours**
Graph Representations (Adjacency Matrix & Lists) - Tarjan's Strongly Connected Components - Minimum Spanning Trees (Kruskal & Prim) - Bellman-Ford Algorithm & Negative Cycles - Floyd-Warshall All-Pairs Shortest Path - Ford-Fulkerson Network Flow.		
Unit:5	**Algorithmic Paradigms & Complex Analysis**	**15 hours**
Master Theorem for Divide and Conquer - Greedy Strategies (Fractional Knapsack) - Dynamic Programming (Matrix Chain Multiplication) - Backtracking (N-Queens Problem) - Introduction to NP-Completeness.		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5
CO1	3	3	2	1	1
CO2	3	3	2	1	1
CO3	3	3	2	1	1
CO4	3	3	2	1	1
CO5	3	3	2	3	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4
CO1	3	2	2	1
CO2	3	2	2	1
CO3	3	2	2	1
CO4	3	3	2	1
CO5	2	2	3	1

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Text Books:

- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, 'Introduction to Algorithms', 4th Ed., MIT Press, 2022. (Covers Units: I - V) https://books.google.com/books?q=Introduction+to+Algorithms+Thomas+H.+Cormen%2C+Charles+E.+Leiserson%2C+Ronald+L.+Rivest%2C+Clifford+Stein
- Dr. G. A. Vijayalakshmi Pai, 'Data Structures and Algorithms: Concepts, Techniques and Applications', Tata McGraw-Hill, 2008. (Covers Units: I - V) https://books.google.com/books?q=Data+Structures+and+Algorithms%3A+Concepts%2C+Techniques+and+Applications+Dr.+G.+A.+Vijayalakshmi+Pai

Reference Books:

- Robert Sedgewick, Kevin Wayne, 'Algorithms', 4th Ed., Addison-Wesley, 2011. https://books.google.com/books?q=Algorithms+Robert+Sedgewick%2C+Kevin+Wayne
- Narasimha Karumanchi, 'Data Structures and Algorithms Made Easy', 5th Ed., CareerMonk Publications, 2017. https://books.google.com/books?q=Data+Structures+and+Algorithms+Made+Easy+Narasimha+Karumanchi

Free Online Study Resources:

- NPTEL - Design and Analysis of Algorithms (IIT Bombay/CMI), free video lectures with downloadable PDF notes (Free) - https://nptel.ac.in/courses/106106131
- Open Data Structures (Pat Morin), free open-access textbook (PDF/HTML) (Free) - https://opendatastructures.org/

SEMESTER – I

CORE – II - MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE AND AUTOMATA

Course Objectives:

The main objectives of this course are to:

1. Formalize predicate calculus and mathematical induction strategies for automated proving.
2. Map relations, algebraic lattices, and combinatorics vectors onto structural databases.
3. Interpret finite automata operations, expressions, and string minimization paths.
4. Standardize parsing architectures and structural languages into standardized normal forms.
5. Distinguish halting challenges, compute functions, and define Church-Turing thresholds.

Expected Course Outcomes:

On the successful completion of the course, student will be able to:

1	Utilize mathematical logic, syntax structures, and predicates for validation schemas.	K1, K2
2	Apply structural combinatorics and recurrence relations to solve database indexing properties.	K2, K3
3	Construct finite state machines and minimize expressions utilizing standard computational lemmas.	K3, K4
4	Compile context-free grammar constraints and execute mapping configurations using pushdown metrics.	K4, K5
5	Map the undecidability limits of modern computing architectures via custom Turing models.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Unit:1	**Mathematical Logic & Predicates**	**15 hours**

Propositional Logic Syntax - Truth Tables - CNF and DNF Normal Forms - Predicate Calculus - Universal & Existential Quantifiers - Rules of Inference - Mathematical Induction - Well-Formed Formulas for Automated Proving.

Unit:2	**Discrete Structures & Combinatorics**	**15 hours**

Set Identities & Operations - Cartesian Products - Relational Matrices - Equivalence Relations & Partial Orderings - Lattices - Functions (Injection, Surjection, Bijection) - Recurrence Relations & Combinatorics.		
Unit:3	**Regular Languages & Finite Automata**	**15 hours**
Deterministic Finite Automata (DFA) - Non-deterministic Finite Automata (NFA) - NFA to DFA Subset Construction - DFA State Minimization - Regular Expressions - Pumping Lemma for Regular Languages - Myhill-Nerode Theorem.		
Unit:4	**Context-Free Languages & Pushdown Automata**	**15 hours**
Context-Free Grammars (CFG) - Parse Trees & Ambiguity - CFG Simplification - Chomsky Normal Form (CNF) - Greibach Normal Form (GNF) - Pushdown Automata (PDA) - Pumping Lemma for CFLs.		
Unit:5	**Turing Machines & Computability**	**15 hours**
Turing Machine Formal Definition & Tape Systems - Multi-tape Turing Machines - Universal Turing Machines - The Halting Problem - Undecidability - Post Correspondence Problem (PCP) - Church-Turing Thesis.		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5
CO1	3	2	1	1	1
CO2	3	2	1	1	1
CO3	3	2	1	1	1
CO4	3	2	1	1	1
CO5	3	2	1	2	2

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4
CO1	2	2	3	1
CO2	2	2	3	1
CO3	3	2	2	1
CO4	3	2	2	1

CO5	2	2	3	1

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Text Books:

- Kenneth H. Rosen, 'Discrete Mathematics and its Applications', 8th Ed., McGraw-Hill, 2019. (Covers Units: I - V) https://books.google.com/books?q=Discrete+Mathematics+and+its+Applications+Kenneth+H.+Rosen
- John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, 'Introduction to Automata Theory, Languages, and Computation', 3rd Ed., Pearson, 2006. (Covers Units: I - V) https://books.google.com/books?q=Introduction+to+Automata+Theory%2C+Languages%2C+and+Computation+John+E.+Hopcroft%2C+Rajeev+Motwani%2C+Jeffrey+D.+Ullman

Reference Books:

- Michael Sipser, 'Introduction to the Theory of Computation', 3rd Ed., Cengage Learning, 2012. https://books.google.com/books?q=Introduction+to+the+Theory+of+Computation+Michael+Sipser
- C. L. Liu, 'Elements of Discrete Mathematics', 2nd Ed., McGraw-Hill, 1985. https://books.google.com/books?q=Elements+of+Discrete+Mathematics+C.+L.+Liu

Free Online Study Resources:

- NPTEL - Discrete Mathematics (IIT Madras/Ropar), free video lectures with downloadable PDF notes (Free) - https://nptel.ac.in/courses/111106135
- Discrete Mathematics: An Open Introduction (Oscar Levin), free open textbook (PDF) (Free) - https://discrete.openmathbooks.org/dmoi3.html

SEMESTER – I
ELECTIVE – I - PROMPT ENGINEERING

Course Objectives:

The main objectives of this course are to:

1. Comprehend vector embedding layouts, tokenizations, and transformer context fields.
2. Design zero-shot, few-shot, and constrained structural syntax scripts without drift.
3. Structure programmatic loop reasoning models using self-consistency validation benchmarks.
4. Construct automated context tools, semantic indexing schemes, and functional pipeline maps.
5. Define compliance frameworks, evaluate hallucinations, and apply multi-tier threat blocks.

Expected Course Outcomes:

On the successful completion of the course, student will be able to:

1	Map transformer internal states and tune hyperparameters for natural sequence alignment.	K1, K2
2	Formulate dynamic in-context instruction blueprints to yield structural data returns.	K2, K3
3	Execute tree-of-thought graph traversals and advanced error-driven prompt validation optimizations.	K3, K4
4	Configure programmatic workflow engines via LangChain and structure optimized multi-source RAG architectures.	K4, K5
5	Deploy safety guardrails, screen out structural toxicity, and neutralize injection vectors within target nodes.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Unit:1	**Introduction to Generative AI and LLMs**	**12 hours**

Transformer Architecture - Encoder-Decoder Foundations - Attention Mechanisms - Byte-Pair Encoding & Tokenization - Context Window Scaling - Auto-regressive Text Generation - Hyperparameter Tuning (Temperature, Top-p) - Vector Embeddings & Similarity Maps.

Unit:2	**Core Prompting Methodologies**	**12 hours**

Zero-Shot Prompting - Few-Shot In-Context Learning - Structured Output Design (JSON & XML) - System-Level Constraint Enforcement - Negative Prompting & Delimiters - Context Drift Mitigation.		
Unit:3	**Advanced Reasoning Paradigms**	**12 hours**
Chain-of-Thought (CoT) Prompting - Self-Consistency Validation - Tree-of-Thoughts (ToT) Graph Searches - Graph-of-Thoughts Parallel Execution - Directional Stimulus Prompting - Iterative Error-Driven Optimization.		
Unit:4	**Programmatic Prompting & Tools**	**12 hours**
Dynamic Template Variables - Automated Context Injection - Model Orchestration via LangChain - Semantic Indexing with LlamaIndex - External API Tool Calling - Retrieval-Augmented Generation (RAG) Pipelines.		
Unit:5	**Vulnerabilities, Bias, and Ethics**	**12 hours**
Hallucination Quantification & Filtering - Toxicity Screening & Guardrails - Prompt Injection Defense & Firewalls - Jailbreaking Mitigation - Data Privacy Scrubbing - Enterprise Compliance & Audit Logging.		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5
CO1	2	2	3	1	3
CO2	2	2	3	1	3
CO3	2	2	3	1	3
CO4	2	2	3	1	3
CO5	2	2	3	3	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4
CO1	2	3	2	2
CO2	3	2	2	2
CO3	2	2	3	2
CO4	3	3	2	2

CO5	2	2	2	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Text Books:

- John Berryman, Albert Ziegler, 'Prompt Engineering for LLMs', 1st Ed., O'Reilly Media, 2024. (Covers Units: I - V) https://books.google.com/books?q=Prompt+Engineering+for+LLMs+John+Berryman%2C+Albert+Ziegler
- James Phoenix, Mike Taylor, 'Prompt Engineering for Generative AI', 1st Ed., O'Reilly Media, 2024. (Covers Units: I - V) https://books.google.com/books?q=Prompt+Engineering+for+Generative+AI+James+Phoenix%2C+Mike+Taylor

Reference Books:

- Sinan Ozdemir, 'Quick Start Guide to Large Language Models', 2nd Ed., Addison-Wesley, 2024. https://books.google.com/books?q=Quick+Start+Guide+to+Large+Language+Models+Sinan+Ozdemir
- Sandra Kublik, Shubham Saboo, 'GPT-3: Building Innovative NLP Products Using Large Language Models', O'Reilly Media, 2023. https://books.google.com/books?q=GPT-3%3A+Building+Innovative+NLP+Products+Using+Large+Language+Models+Sandra+Kublik%2C+Shubham+Saboo

Free Online Study Resources:

- Prompt Engineering Guide, free open-access online resource (Free) - https://www.promptingguide.ai/
- OpenAI Prompt Engineering official guide, free documentation (Free) - https://platform.openai.com/docs/guides/prompt-engineering

SEMESTER – I
ELECTIVE – I (OPTION) - ETHICAL HACKING AND CYBER SECURITY

Course Objectives:

The main objectives of this course are to:

1. Apply enumeration models, scan network parameters, and validate security structures.
2. Implement dynamic logic auditing, execute privilege containment, and analyze internal log paths.
3. Uncover network loops, evaluate routing hazards, and deploy intrusion tracking signatures.
4. Construct server configuration filters to neutralize automated scripting injections.
5. Define shared cloud risk models, isolate container escapes, and document audit records.

Expected Course Outcomes:

On the successful completion of the course, student will be able to:

1	Execute active and passive infrastructure footprinting configurations within standardized boundaries.	K1, K2
2	Perform comprehensive system vulnerability diagnostics and compute metrics using CVSS tools.	K2, K3
3	Capture and parse transport packet frameworks using Wireshark to neutralize loop overflows.	K3, K4
4	Identify and patch multi-tier database injection vectors and target cross-site validation flaws.	K4, K5
5	Manage access configurations over distributed sandbox environments, ensuring reporting compliance.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Unit:1	**Security Fundamentals & Footprinting**	**12 hours**

Information Security Fundamentals - CIA Triad Validation - Passive Footprinting & OSINT - Active Network Scanning & Nmap Routines - OS Fingerprinting - DNS Zone Transfer - Service Enumeration.

Unit:2	**Vulnerability Assessment & System Hacking**	**12 hours**

Vulnerability Identification & CVE Listings - CVSS Matrix Calculations - Password Cracking - Privilege Escalation - Steganography & Payload Injection - Log Clearing &

Footprint Erasure.		
Unit:3	**Malware Threats & Sniffing**	**12 hours**
Trojan Delivery Frameworks - Worms & Ransomware Encryption Loops - Packet Sniffing via Wireshark - ARP Poisoning Vectors - Switch MAC Table Overflow - Host Intrusion Detection Signatures.		
Unit:4	**Web Server & Application Attacks**	**12 hours**
Web Server Vulnerability Audits - SQL Injection & Database Extraction - Blind & Timing SQLi - Persistent Cross-Site Scripting (XSS) - Cross-Site Request Forgery (CSRF) - Automated Exploitation using Burp Suite.		
Unit:5	**Wireless, Mobile, and Cloud Security**	**12 hours**
Wireless Encryption & WPA2 - Four-Way Handshake Capture - Mobile OS Sandbox Vulnerabilities - Cloud Shared Responsibility Model - Virtual Machine Escape - Pen Testing Compliance & Reporting.		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5
CO1	2	2	3	3	2
CO2	2	2	3	3	2
CO3	2	2	3	3	2
CO4	2	2	3	3	2
CO5	2	2	3	3	2

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4
CO1	2	3	1	2
CO2	2	3	2	2
CO3	3	3	1	2
CO4	3	2	2	3
CO5	2	2	1	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Text Books:

- Ric Messier, 'CEH v12 Certified Ethical Hacker Study Guide', 1st Ed., Sybex, 2023. (Covers Units: I - V) https://books.google.com/books?q=CEH+v12+Certified+Ethical+Hacker+Study+Guide+Ric+Messier
- Atul Kahate, 'Cryptography and Network Security', 4th Ed., McGraw Hill India, 2019. (Covers Units: I - V) https://books.google.com/books?q=Cryptography+and+Network+Security+Atul+Kahate

Reference Books:

- William Stallings, 'Cryptography and Network Security: Principles and Practice', 8th Ed., Pearson, 2020. https://books.google.com/books?q=Cryptography+and+Network+Security%3A+Principles+and+Practice+William+Stallings
- Georgia Weidman, 'Penetration Testing: A Hands-On Introduction to Hacking', No Starch Press, 2014. https://books.google.com/books?q=Penetration+Testing%3A+A+Hands-On+Introduction+to+Hacking+Georgia+Weidman

Free Online Study Resources:

- NPTEL - Ethical Hacking (IIT Kanpur), free video lectures with downloadable PDF notes (Free) - https://nptel.ac.in/courses/106105217
- OWASP Foundation, free official web security resources and guides (Free) - https://owasp.org/www-project-top-ten/

SEMESTER – I
PRACTICAL – I - ALGORITHMIC PARADIGM LAB

Learning Objective (LO):
To understand the code implementation of basic linear, tree, and graph structures.

List of Programs:

01. Write a C++ program to implement search operations in a simple Skip-List layout.
02. Write a program to create an XOR Linked List and display its elements forward and backward.
03. Write a program to construct an AVL Tree and perform left and right balancing rotations.
04. Write a program to insert a node into a Red-Black Tree and verify node colour rules.
05. Write a program to simulate a B-Tree index structure with simple node splitting.
06. Write a program to implement a Leftist Heap and merge two separate heaps.
07. Write a program to implement a standard Min-Max Heap and extract the minimum element.
08. Write a program to find Strongly Connected Components in a graph using Tarjan's method.
09. Write a program to find the shortest path between all pairs of nodes using Floyd-Warshall.
10. Write a program to find the maximum network flow using the Ford-Fulkerson method.

Expected Course Outcomes:
On the successful completion of this practical course, student will be able to:

1	CO1: Implement skip-list and XOR linked-list operations for efficient linear data access.	K1,K2
2	CO2: Construct and balance AVL and Red-Black Trees through node rotation and colour rule verification.	K2,K3
3	CO3: Simulate B-Tree index structures and implement Leftist and Min-Max Heap operations.	K3,K4
4	CO4: Identify strongly connected components and compute all-pairs shortest paths in graphs.	K4, K5
5	CO5: Determine maximum network flow using the Ford-Fulkerson method for resource	K5, K6

	allocation problems.	
K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5
CO1	3	2	2	1	1
CO2	3	3	2	1	1
CO3	3	3	2	1	1
CO4	3	3	2	1	1
CO5	3	3	2	2	2

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4
CO1	3	2	2	1
CO2	3	2	2	1
CO3	3	2	2	1
CO4	3	3	2	1
CO5	3	3	2	1

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

SEMESTER – I

PRACTICAL – II - MATHEMATICAL FOUNDATIONS LAB (USING C++)

Learning Objective (LO):

To convert discrete mathematical rules and state machines into executable C++ code.

List of Programs:

01. Write a C++ program to generate a Truth Table for any given two-variable logic statement.
02. Write a program to implement basic inference rules (Modus Ponens and Modus Tollens).
03. Write a program to check if an input matrix represents an Equivalence Relation.
04. Write a program to draw a simple Hasse Diagram from a given partial order set.
05. Write a program to solve second-order linear recurrence relations using a formula method.
06. Write a program to simulate a Deterministic Finite Automata (DFA) string checker.
07. Write a program to convert a basic Non-deterministic Finite Automata (NFA) to a DFA.
08. Write a program to reduce the number of states in a DFA using state minimization.
09. Write a program to remove null (epsilon) productions from a Context-Free Grammar.
10. Write a program to implement a Pushdown Automata (PDA) for matching open and closed brackets.

Expected Course Outcomes:

On the successful completion of this practical course, student will be able to:

1	CO1: Generate truth tables and implement basic logical inference rules in C++.	K1, K2
2	CO2: Verify equivalence relations and construct Hasse diagrams for partial order sets.	K2, K3
3	CO3: Solve linear recurrence relations using systematic formula-based methods.	K3, K4
4	CO4: Simulate and convert finite automata, including DFA minimization and NFA-to-DFA conversion.	K4, K5
5	CO5: Process context-free grammars and implement pushdown automata for language recognition.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5
CO1	3	2	3	1	1
CO2	3	2	3	1	1
CO3	3	2	2	1	1
CO4	3	3	3	1	1
CO5	3	3	3	1	1

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4
CO1	2	2	3	1
CO2	2	2	3	1
CO3	2	2	3	1
CO4	3	2	3	1
CO5	3	2	3	1

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

SEMESTER – I

SEC – I - UI/UX AND PROTOTYPING TOOLS (FIGMA)

Course Objectives:

The main objectives of this course are to:

1. Understand the basics of UI/UX design principles.
2. Learn to use the Figma design tool and its basic features.
3. Learn to create simple design systems using colors, fonts, and components.
4. Learn to create simple interactive prototypes.
5. Learn to collaborate and share designs using Figma.

Expected Course Outcomes:

On the successful completion of the course, student will be able to:

1	Explain the basic principles of UI/UX design.	K1, K2
2	Use the Figma workspace to create simple design layouts.	K2, K3
3	Create a simple design system using colors, fonts, and reusable components.	K3, K4
4	Create a simple interactive prototype with basic navigation between screens.	K4, K5
5	Collaborate on a shared design file and export design assets.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Unit:1	**UI/UX Design Basics**	**6 hours**

Introduction to User Interface (UI) and User Experience (UX) - Basic design layouts - Typography basics - Introduction to accessibility in design.

Unit:2	**Getting Started with Figma**	**6 hours**

Introduction to the Figma workspace - Basic shapes and tools - Working with frames - Layers panel - Basic alignment and spacing.

Unit:3	**Design Systems and Typography**	**6 hours**

Using color palettes - Setting up text styles - Introduction to grids - Introduction to Auto-Layout - Creating reusable components.

Unit:4	**Simple Interactive Prototypes**	**6 hours**

Linking frames together - Creating basic navigation between screens - Simple button interactions - Introduction to transitions.

Unit:5	Collaboration and sharing	6 hours
Using Figma plugins (overview) - Sharing a design file with others - Basic commenting and feedback - Exporting design assets for developers.		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5
CO1	1	2	3	1	1
CO2	1	2	3	1	1
CO3	1	2	3	1	1
CO4	1	2	3	1	1
CO5	1	2	3	1	2

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4
CO1	2	1	2	2
CO2	2	1	2	2
CO3	2	1	2	2
CO4	2	1	2	2
CO5	1	1	2	2

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Text Books:

- Anthony E. Sanchez, '2024-2025 Newbies Guide to UI/UX Design Using Figma', Inigi Publishers, 2024. (Covers Units: I - V) https://books.google.com/books?q=2024-2025+Newbies+Guide+to+UI%2FUX+Design+Using+Figma+Anthony+E.+Sanchez
- Tom Green, Kevin Brandon, 'UX Design with Figma: User-Centered Interface Design and Prototyping with Figma', Apress, 2024. (Covers Units: I - V) https://books.google.com/books?q=UX+Design+with+Figma%3A+User-Centered+Interface+Design+and+Prototyping+with+Figma+Tom+Green%2C+Kevin+Brandon

Reference Books:

- Fabio Staiano, 'Designing and Prototyping Interfaces with Figma', 2nd Ed., Packt Publishing, 2023. https://books.google.com/books?q=Designing+and+Prototyping+Interfaces+with+Figma+Fabio+Staiano
- Aslan Walker, 'Figma for Beginners: A Beginner's Guide to New Features for Designers, Artists, and Developers', Independent Publisher, 2024. https://books.google.com/books?q=Figma+for+Beginners%3A+A+Beginner+Aslan+Walker

Free Online Study Resources:

- Figma official Learn Design resources, free (Free) - https://www.figma.com/resources/learn-design/
- Figma Help Center, free official tutorials and documentation (Free) - https://help.figma.com/

SEMESTER - II

CORE – III - DISTRIBUTED DATABASE SYSTEMS AND NOSQL PARADIGMS

Course Objectives:

The main objectives of this course are to:

1. Understand the basics of distributed database architecture and data distribution.
2. Learn how transactions are managed across distributed systems.
3. Understand the need for NoSQL databases and the CAP theorem.
4. Build applications using a document-based NoSQL database (MongoDB).
5. Understand the basic concepts of column-family and graph databases.

Expected Course Outcomes:

On the successful completion of the course, student will be able to:

1	Explain the architecture of distributed databases and basic data distribution methods.	K1, K2
2	Describe how distributed transactions are managed, including the Two-Phase Commit protocol.	K2, K3
3	Explain the CAP theorem and the reasons for using NoSQL databases.	K3, K4
4	Design and query a document database using MongoDB.	K4, K5
5	Describe the basic concepts of column-family and graph databases with simple examples.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Unit:1	**Distributed Database Basics**	**15 hours**

Introduction to distributed databases - Data fragmentation (horizontal and vertical) - Data replication basics - Distributed query processing overview - Client-server architecture in distributed databases.

Unit:2	**Distributed Transaction Management**	**15 hours**

Transaction basics and ACID properties in a distributed setting - Concurrency control overview - Distributed deadlock basics - Two-Phase Commit (2PC) protocol.

Unit:3	**Introduction to NoSQL Databases**	**15 hours**

Limitations of relational databases for big data - Types of NoSQL databases (overview) - The CAP Theorem - Eventual consistency - When to use NoSQL versus relational

databases.		
Unit:4	**Document and Key-Value Databases**	**15 hours**
Introduction to MongoDB - Documents and collections - Basic CRUD operations in MongoDB - Simple queries and filters - Introduction to key-value stores.		
Unit:5	**Column-Family and Graph Databases (Overview)**	**15 hours**
Introduction to column-family databases with Apache Cassandra (concepts) - Introduction to graph databases with Neo4j (concepts) - Comparing NoSQL database types - Choosing the right database for an application.		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5
CO1	3	2	2	1	1
CO2	3	2	2	1	1
CO3	2	2	2	1	1
CO4	3	3	2	1	1
CO5	2	2	2	1	1

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4
CO1	2	2	2	1
CO2	2	2	2	1
CO3	2	2	2	1
CO4	3	2	2	1
CO5	2	2	2	1

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Text Books:

- M. Tamer Ozsu, Patrick Valduriez, 'Principles of Distributed Database Systems', 4th Ed., Springer, 2020. (Covers Units: I - V) https://books.google.com/books?q=Principles+of+Distributed+Database+Systems+M.+Tamer+Ozsu%2C+Patrick+Valduriez

- Shannon Bradshaw, Eoin Brazil, Kristina Chodorow, 'MongoDB: The Definitive Guide', 3rd Ed., O'Reilly Media, 2019. (Covers Units: I - V) https://books.google.com/books?q=MongoDB%3A+The+Definitive+Guide+Shannon+Bradshaw%2C+Eoin+Brazil%2C+Kristina+Chodorow

Reference Books:

- Pramod J. Sadalage, Martin Fowler, 'NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence', Addison-Wesley, 2012. https://books.google.com/books?q=NoSQL+Distilled%3A+A+Brief+Guide+to+the+Emerging+World+of+Polyglot+Persistence+Pramod+J.+Sadalage%2C+Martin+Fowler
- Eben Hewitt, 'Cassandra: The Definitive Guide', 1st Ed., O'Reilly Media, 2010 https://books.google.com/books?q=Cassandra%3A+The+Definitive+Guide+Eben+Hewitt

Free Online Study Resources:

- MongoDB University, free official self-paced courses (Free) - https://learn.mongodb.com/
- NPTEL - Distributed Systems (IIT Kharagpur), free video lectures with downloadable PDF notes (Free) - https://nptel.ac.in/courses/106106168

SEMESTER – II
CORE – IV - JAVA ARCHITECTURE AND FRAMEWORKS

Course Objectives:

The main objectives of this course are to:

1. Understand JVM memory structure and apply Java's functional programming features.
2. Learn the core concepts of the Spring Framework, including Dependency Injection.
3. Build database-driven applications using Spring Boot and Hibernate.
4. Design and develop REST APIs using Spring Boot.
5. Apply basic security practices and deploy a Spring Boot application.

Expected Course Outcomes:

On the successful completion of the course, student will be able to:

1	Explain JVM memory management and apply Java reflection, lambda expressions, and streams.	K1,K2
2	Apply Dependency Injection and Aspect-Oriented Programming concepts using the Spring Framework.	K2,K3
3	Develop database-driven applications using Spring Boot and Hibernate ORM.	K3,K4
4	Design and implement REST APIs with proper request handling and exception handling.	K4,K5
5	Apply basic authentication security and deploy a Spring Boot application.	K5,K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Unit:1	**Java and the JVM**	**15 hours**

JVM architecture and memory areas (heap, stack, method area) - Java Reflection API basics - Lambda expressions - Stream API for collections - Introduction to multithreading and concurrency.

Unit:2	**Spring Framework Fundamentals**	**15 hours**

Introduction to the Spring Framework - Inversion of Control (IoC) and Dependency Injection - Spring Beans and the Application Context - Common Spring annotations (@Component, @Autowired, @Service) - Introduction to Aspect-Oriented Programming (AOP).

Unit:3	**Spring Boot and Database Connectivity**	**15 hours**

Introduction to Spring Boot and project setup - Spring Boot starter dependencies -

Connecting Spring Boot to a database - Introduction to Hibernate and Object-Relational Mapping (ORM) - Spring Data JPA repositories.		
Unit:4	**Building REST APIs**	**15 hours**
REST architecture principles - HTTP methods (GET, POST, PUT, DELETE) - Building REST controllers in Spring Boot - Handling requests and responses - Basic exception handling in REST APIs.		
Unit:5	**Application Security and Deployment**	**15 hours**
Basic authentication and authorization concepts - Introduction to Spring Security - Overview of JSON Web Tokens (JWT) - Packaging a Spring Boot application - Basic deployment using Docker.		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5
CO1	2	2	2	1	1
CO2	2	3	2	1	1
CO3	2	3	3	1	1
CO4	2	3	2	1	1
CO5	2	2	2	1	2

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4
CO1	3	1	1	1
CO2	3	2	1	1
CO3	3	2	1	1
CO4	3	1	1	1
CO5	2	2	1	2

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Text Books:

- Craig Walls, 'Spring in Action', 6th Ed., Manning Publications, 2022. (Covers Units: I - V) https://books.google.com/books?q=Spring+in+Action+Craig+Walls

- Cay S. Horstmann, 'Core Java Volume I - Fundamentals', 12th Ed., Pearson, 2021. (Covers Units: I - V) https://books.google.com/books?q=Core+Java+Volume+I+-+Fundamentals+Cay+S.+Horstmann

Reference Books:

- Joshua Bloch, 'Effective Java', 3rd Ed., Addison-Wesley, 2018. https://books.google.com/books?q=Effective+Java+Joshua+Bloch
- Craig Walls, 'Spring Boot in Action', Manning Publications, 2016. https://books.google.com/books?q=Spring+Boot+in+Action+Craig+Walls

Free Online Study Resources:

- Oracle - The Java Tutorials, free official documentation (Free) - https://docs.oracle.com/javase/tutorial/
- Spring Official Guides, free official documentation and tutorials (Free) - https://spring.io/guides

SEMESTER – II

ELECTIVE – II - ADVANCED DISTRIBUTED SYSTEMS AND MODERN ARCHITECTURES

Course Objectives:

The main objectives of this course are to:

1. Understand the basics of communication between distributed systems.
2. Learn how distributed systems agree on shared decisions (consensus).
3. Understand the basics of microservices architecture.
4. Learn the fundamentals of containers and basic orchestration.
5. Understand the basics of monitoring distributed applications.

Expected Course Outcomes:

On the successful completion of the course, student will be able to:

1	Explain how remote procedure calls and basic messaging work between distributed systems.	K1, K2
2	Describe how distributed systems reach agreement using basic consensus algorithms.	K2, K3
3	Explain the principles of microservices architecture and event-driven design.	K3, K4
4	Use Docker to containerize an application and explain basic container orchestration concepts.	K4, K5
5	Describe basic approaches to monitoring and observing distributed applications.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Unit:1	**Foundations of Distributed Computing**	**12 hours**

Remote Procedure Call (RPC) basics - Introduction to gRPC - Message-based communication between services - Basic clock synchronization concepts in distributed systems.

Unit:2	**Distributed Consensus (Overview)**	**12 hours**

Why consensus is needed in distributed systems - Basic idea of the Paxos algorithm - Basic idea of the Raft algorithm - Leader election concepts.

Unit:3	**Microservices Architecture Basics**	**12 hours**

Monolithic versus microservices architecture - Principles of microservices design - Event-driven communication between services - Basic load balancing concepts.

Unit:4	Containers and Basic Orchestration	12 hours
Introduction to containers and Docker - Building and running Docker images - Introduction to container orchestration - Basic concepts of Kubernetes.		
Unit:5	**Monitoring Distributed Applications**	**12 hours**
Why monitoring is important in distributed systems - Basic logging practices - Introduction to monitoring tools (overview) - Introduction to serverless computing.		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5
CO1	2	2	2	1	1
CO2	2	2	2	1	1
CO3	2	3	2	1	1
CO4	2	2	2	1	1
CO5	2	2	2	1	1

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4
CO1	2	2	2	1
CO2	2	2	2	1
CO3	2	3	2	1
CO4	2	2	2	1
CO5	2	2	2	1

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Text Books:

- Brendan Burns, 'Designing Distributed Systems', 1st Ed., O'Reilly Media, 2018. (Covers Units: I - V) https://books.google.com/books?q=Designing+Distributed+Systems+Brendan+Burns
  Martin Kleppmann, 'Designing Data-Intensive Applications', 1st Ed., O'Reilly Media, 2017. (Covers Units: I - V)
  https://books.google.com/books?q=Designing+Data-Intensive+Applications+Martin+Kleppmann

Reference Books:

- Sam Newman, 'Building Microservices', 2nd Ed., O'Reilly Media, 2021 https://books.google.com/books?q=Building+Microservices+Sam+Newman
- Kelsey Hightower, Brendan Burns, Joe Beda, 'Kubernetes: Up and Running', 1st Ed., O'Reilly Media, 2017. https://books.google.com/books?q=Kubernetes%3A+Up+and+Running+Kelsey+Hightower%2C+Brendan+Burns%2C+Joe+Beda

Free Online Study Resources:

- Kubernetes Documentation, free official documentation (Free) - https://kubernetes.io/docs/home/
- Martin Kleppmann's free lecture series, 'Distributed Systems', University of Cambridge (Free) - https://www.cl.cam.ac.uk/teaching/2021/ConcDisSys/dist-sys-notes.pdf

SEMESTER – II
ELECTIVE – II (OPTION) - DEEP ALGORITHMIC LEARNING

Course Objectives:

The main objectives of this course are to:

1. Understand the basics of neural networks and how they learn.
2. Learn the fundamentals of Convolutional Neural Networks for image data.
3. Learn the fundamentals of Recurrent Neural Networks for sequence data.
4. Get an introduction to generative models in deep learning.
5. Understand basic approaches to optimizing and deploying deep learning models.

Expected Course Outcomes:

On the successful completion of the course, student will be able to:

1	Explain how neural networks learn using backpropagation and basic optimization.	K1, K2
2	Build a simple Convolutional Neural Network for image classification.	K2, K3
3	Build a simple Recurrent Neural Network or LSTM for sequence data.	K3, K4
4	Explain the basic idea of generative models such as Autoencoders and GANs.	K4, K5
5	Describe basic techniques for optimizing and deploying a trained deep learning model.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 – Create

Unit:1	**Neural Network Foundations**	**15 hours**

Perceptron and activation functions - Multi-Layer Perceptron (MLP) - Loss functions - Backpropagation basics - Gradient descent and the Adam optimizer.

Unit:2	**Convolutional Neural Networks (CNNs)**	**15 hours**

Convolution and pooling operations - CNN architecture basics - Introduction to popular CNN models (overview) - Applications in image classification.

Unit:3	**Recurrent and Sequence Models**	**15 hours**

Recurrent Neural Networks (RNN) basics - Long Short-Term Memory (LSTM) networks - Introduction to attention - Basic idea of Transformer models.

Unit:4	**Introduction to Generative Models**	**15 hours**

Autoencoders - Introduction to Variational Autoencoders (VAE) - Basic idea of Generative

Adversarial Networks (GAN) - Simple applications of generative models.		
Unit:5	**Model Optimization and Deployment Basics**	**15 hours**
Hyperparameter tuning basics - Introduction to model compression - Basics of deploying a trained model - Overview of GPU usage for training.		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5
CO1	2	2	2	1	1
CO2	2	2	2	1	1
CO3	2	2	2	1	1
CO4	2	2	2	1	1
CO5	2	2	2	1	1

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4
CO1	2	2	2	1
CO2	2	2	2	1
CO3	2	2	2	1
CO4	2	2	2	1
CO5	2	2	2	1

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Text Books:

- Ian Goodfellow, Yoshua Bengio, Aaron Courville, 'Deep Learning', MIT Press, 2016. (Covers Units: I - V) - Free full text: https://www.deeplearningbook.orghttps://www.deeplearningbook.org/
- Aurelien Geron, 'Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow', 3rd Ed., O'Reilly Media, 2022. (Covers Units: I - V) https://books.google.com/books?q=Hands-On+Machine+Learning+with+Scikit-Learn%2C+Keras%2C+and+TensorFlow+Aurelien+Geron

Reference Books:

- Francois Chollet, 'Deep Learning with Python', 2nd Ed., Manning Publications, 2021. https://books.google.com/books?q=Deep+Learning+with+Python+Francois+Chollet
- Christopher M. Bishop, 'Pattern Recognition and Machine Learning', Springer, 2006. https://books.google.com/books?q=Pattern+Recognition+and+Machine+Learning+Christopher+M.+Bishop

Free Online Study Resources:

- Deep Learning (Goodfellow, Bengio, Courville), official free full-text edition (Free) - https://www.deeplearningbook.org/
- Stanford CS231n - Convolutional Neural Networks, free official course notes (Free) - https://cs231n.github.io/

SEMESTER – II

PRACTICAL – III - DISTRIBUTED DATABASE SYSTEMS AND NOSQL LAB

Learning Objective (LO):

To practice basic data distribution concepts and build simple applications using MongoDB.

List of Programs:

01.	Write a script to demonstrate horizontal data fragmentation using two simple tables.
02.	Write a program to simulate a basic Two-Phase Commit between two database nodes.
03.	Install MongoDB and create a database with a sample collection.
04.	Write MongoDB queries to insert, read, update, and delete documents (CRUD).
05.	Write a MongoDB query to filter documents using conditions.
06.	Write a MongoDB aggregation query using $match and $group.
07.	Write a simple program to store and retrieve key-value pairs using Redis.
08.	Create a simple table in Cassandra and insert sample records (conceptual demo).
09.	Create a simple graph with a few nodes and relationships in Neo4j (conceptual demo).
10.	Compare query results between MongoDB and a relational database for the same dataset.

Expected Course Outcomes:

On the successful completion of this practical course, student will be able to:

1	CO1: Demonstrate basic data fragmentation and a simple Two-Phase Commit process.	K1, K2
2	CO2: Perform basic CRUD operations in MongoDB.	K2, K3
3	CO3: Write filter and aggregation queries in MongoDB.	K3, K4
4	CO4: Use a key-value store (Redis) for simple data storage and retrieval.	K4, K5
5	CO5: Demonstrate basic concepts of column-family and graph databases.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5

CO1	2	2	2	1	1
CO2	3	2	2	1	1
CO3	3	2	2	1	1
CO4	2	2	2	1	1
CO5	2	2	2	1	1

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4
CO1	2	2	2	1
CO2	2	2	2	1
CO3	2	2	2	1
CO4	2	2	2	1
CO5	2	2	2	1

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

SEMESTER – II

PRACTICAL – IV - JAVA ARCHITECTURE AND FRAMEWORKS LAB

Learning Objective (LO):

To build simple database-driven applications and REST APIs using Spring Boot and Hibernate.

List of Programs:

01. Write a Java program to read class details (methods and fields) using Reflection.
02. Write a program to filter and process a list of numbers using Java Lambda expressions and Streams.
03. Create a simple Spring Boot 'Hello World' application.
04. Demonstrate Dependency Injection in Spring using the @Autowired annotation.
05. Write a simple Spring AOP program to log the execution time of a method.
06. Configure Hibernate to map a Java class to a database table and perform a basic insert.
07. Build a simple CRUD REST API in Spring Boot using @RestController.
08. Add exception handling to a REST API to return proper error messages.
09. Implement basic login authentication using Spring Security.
10. Package a Spring Boot application as a runnable JAR file.

Expected Course Outcomes:

On the successful completion of this practical course, student will be able to:

1	CO1: Apply Java Reflection and use Lambda expressions and Streams for data processing.	K1, K2
2	CO2: Build a simple Spring Boot application and demonstrate Dependency Injection and AOP logging.	K2, K3
3	CO3: Configure Hibernate ORM to connect a Java application to a database.	K3, K4
4	CO4: Develop a CRUD REST API in Spring Boot with basic exception handling.	K4, K5
5	CO5: Implement basic authentication using Spring Security and package a deployable application.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5
CO1	2	2	1	1	1
CO2	2	2	2	1	1
CO3	2	2	2	1	1
CO4	2	3	2	1	1
CO5	2	2	2	1	2

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4
CO1	3	1	1	1
CO2	3	2	1	1
CO3	3	2	1	1
CO4	3	2	1	1
CO5	2	2	1	2

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

SEMESTER – II

NME – I - FUNDAMENTALS OF INTERNET AND OFFICE AUTOMATION

Course Objectives:

The main objectives of this course are to:

1. Understand domain registry mappings, routing protocols, and multi-factor validation fields.
2. Master dynamic mail merges, document typography steps, and version control features.
3. Utilize complex lookup logic, configure structural aggregates, and build chart elements.
4. Format multimedia streams, define presentation schedules, and export handouts.
5. Establish collaborative document systems, manage notifications, and ensure recovery rules.

Expected Course Outcomes:

On the successful completion of the course, student will be able to:

1	Explain web layout architectures, parsing indices, and secure transmission protocols.	K1, K2
2	Produce structured documentation templates featuring automated dynamic indexing and database linkages.	K2, K3
3	Build analytical spreadsheets incorporating cross-matrix filters, functions, and layout validation boundaries.	K3, K4
4	Design unified slide master presentations using configured transition triggers.	K4, K5
5	Coordinate real-time cloud data spaces, configuring custom validation permissions and backups.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Unit:1	**Internet & Web Basics**	**6 hours**

Web Client-Server Architecture - URL Structure & Parsing - Domain Name System (DNS) Mapping - SMTP & IMAP Email Protocols - Search Engine Indexing & Keyword Retrieval - E-Commerce Security & Multi-Factor Authentication.

Unit:2	**Word Processing Operations**	**6 hours**

Text Typography & Style Formatting - Table Configurations & Layouts - Mail Merge with External Databases - Automated Table of Contents - Track Changes & Document Version

Control - Page Geometry & Template Design.		
Unit:3	**Spreadsheet Analysis**	**6 hours**
Relative and Absolute Cell Addressing - Logical and Mathematical Formulas - Data Retrieval via VLOOKUP and XLOOKUP - Pivot Tables & Data Aggregation - Dynamic Charts & Graphs - Data Filtering & Validation Rules.		
Unit:4	**Presentation Design**	**6 hours**
Slide Master & Global Layout Templates - Slide Transitions & Animation Timing - Multimedia Integration (Audio & Video) - Presenter View Dual-Screen Configurations - Handout Generation & Exporting Formats.		
Unit:5	**Cloud Automation & Collaboration**	**6 hours**
Cloud Drive Permissions & File Sharing - Real-Time Collaborative Document Editing - Online Forms & Questionnaire Data Validation - Automated Notification Triggers - Multi-Device Cloud Synchronization - Data Backup & Recovery Scheduling.		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5
CO1	1	1	3	1	1
CO2	1	1	3	1	1
CO3	1	1	3	1	1
CO4	1	1	3	1	1
CO5	1	1	3	1	1

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4
CO1	1	2	1	3
CO2	2	1	1	3
CO3	2	1	1	3
CO4	1	1	1	3
CO5	1	2	1	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Text Books:

- Joan Lambert, 'Microsoft Office 365 and Office 2021 Step by Step', Microsoft Press, 2022. (Covers Units: I - V) https://books.google.com/books?q=Microsoft+Office+365+and+Office+2021+Step+by+Step+Joan+Lambert
- June Jamrich Parsons, 'New Perspectives Computer Concepts', Cengage Learning, 2021. (Covers Units: I - V) https://books.google.com/books?q=New+Perspectives+Computer+Concepts+June+Jamrich+Parsons

Reference Books:

- Sanjay Saxena, 'A First Course in Computers', Vikas Publishing House, 2021. https://books.google.com/books?q=A+First+Course+in+Computers+Sanjay+Saxena
- ITL Education Solutions Limited, 'Introduction to Information Technology', 2nd Ed., Pearson, 2012. https://books.google.com/books?q=Introduction+to+Information+Technology+ITL+Education+Solutions+Limited

Free Online Study Resources:

- Microsoft Office Support, free official how-to documentation (Free) - https://support.microsoft.com/en-us/office
- NPTEL - Computer Fundamentals and Office Automation, free video lectures with downloadable PDF notes (Free) - https://onlinecourses.swayam2.ac.in/

SEMESTER -II
FUNDAMENTALS OF HUMAN RIGHTS

Course Objectives:

The main objectives of this course are to:

1. Understand the meaning, nature, and classification of human rights.
2. Learn about human rights protections under the Indian Constitution.
3. Understand the major international human rights instruments.
4. Learn about the institutions responsible for protecting human rights in India.
5. Understand human rights issues in the context of modern technology and society.

Expected Course Outcomes:

On the successful completion of the course, student will be able to:

1	Explain the meaning, nature, and classification of human rights.	K1, K2
2	Describe the Fundamental Rights and Directive Principles under the Indian Constitution.	K2, K3
3	Explain major international human rights instruments and the role of the United Nations.	K3, K4
4	Describe the role of the National Human Rights Commission and other institutions in protecting rights.	K4, K5
5	Discuss human rights issues related to technology and contemporary society.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Unit:1	**Introduction to Human Rights**	**6 hours**

Meaning, definition, and nature of human rights - Historical evolution of human rights - Classification of human rights (civil, political, economic, social, and cultural) - Characteristics of human rights.

Unit:2	**Human Rights and the Indian Constitution**	**6 hours**

Fundamental Rights under the Indian Constitution - Directive Principles of State Policy - Fundamental Duties - Constitutional remedies for the protection of rights (Article 32 and Article 226).

Unit:3	**International Human Rights Instruments**	**6 hours**

Universal Declaration of Human Rights (UDHR) - International Covenant on Civil and Political Rights (ICCPR) - International Covenant on Economic, Social and Cultural Rights

(ICESCR) - Role of the United Nations in promoting human rights.		
Unit:4	**Human Rights Institutions in India**	**6 hours**
National Human Rights Commission (NHRC) - State Human Rights Commissions - National Commission for Women - National Commission for Scheduled Castes and Scheduled Tribes - Role of the judiciary in protecting human rights.		
Unit:5	**Human Rights and Contemporary Issues**	**6 hours**
Women's rights and child rights - Rights of persons with disabilities - Human rights in the context of technology (privacy, data protection, and AI ethics) - Common human rights violations and redressal mechanisms.		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5
CO1	1	1	1	3	1
CO2	1	1	1	3	1
CO3	1	1	1	3	1
CO4	1	1	1	3	1
CO5	1	1	2	3	1

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4
CO1	1	1	1	2
CO2	1	1	1	2
CO3	1	1	1	2
CO4	1	1	1	2
CO5	1	1	2	2

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Text Books:

- Dr. H. O. Agarwal, 'International Law and Human Rights', 21st Ed., Central Law Publications, 2022. (Covers Units: I - V) https://books.google.com/books?q=International+Law+and+Human+Rights+Dr.+H.+O.+Agarwal

- Dr. J. N. Pandey, 'Constitutional Law of India', 60th Ed., Central Law Agency, 2023. (Covers Units: I - V) https://books.google.com/books?q=Constitutional+Law+of+India+Dr.+J.+N.+Pandey

Reference Books:

- C. J. Nirmal (Ed.), 'Human Rights in India: Historical, Social and Political Perspectives', Oxford University Press, 2000. https://books.google.com/books?q=Human+Rights+in+India%3A+Historical%2C+Social+and+Political+Perspectives+C.+J.+Nirmal+%28Ed.%29
- Upendra Baxi, 'The Future of Human Rights', 4th Ed., Oxford University Press, 2018. https://books.google.com/books?q=The+Future+of+Human+Rights+Upendra+Baxi

Free Online Study Resources:

- United Nations Human Rights Office of the High Commissioner (OHCHR), free publications (Free) - https://www.ohchr.org/en/publications
- National Human Rights Commission, India, free publications (Free) - https://nhrc.nic.in/publications/latest-released-publications

E-Book / Online Resource Website:

United Nations Human Rights Office of the High Commissioner (OHCHR) – Free Publications: https://www.ohchr.org/en/publications | National Human Rights Commission, India – Free Publications: https://nhrc.nic.in/publications/latest-released-publications

SEMESTER - III
CORE – V - PYTHON FOR DATA ENGINEERING AND AUTOMATION

Course Objectives:

The main objectives of this course are to:

1. Utilize advanced computational constructs, lambda transformations, and memory optimizations.
2. Clean large datasets, execute relational merges, and build aggregated insights.
3. Organize pipeline transformations, synchronize system calls, and automate routine tasks.
4. Manage HTTP request flows, parse document strings, and build browser automations.
5. Code unit test modules, isolate system parameters, and construct CLI wrappers.

Expected Course Outcomes:

On the successful completion of the course, student will be able to:

1	Implement memory-optimized abstract data interfaces and thread-safe custom stream generators.	K1, K2
2	Manipulate high-dimensional vectors and structure transactional aggregations on distributed DataFrames.	K2, K3
3	Construct robust ETL orchestration pipelines managed via Airflow directed acyclic tracking graphs.	K3, K4
4	Build asynchronous web extractors featuring automated execution retries and rate-limiting blocks.	K4, K5
5	Build comprehensive test suites, mock simulation data nodes, and compile modular extensions.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Unit:1	**Advanced Python Foundations**	**15 hours**

Memory optimization - Lambda functional transformations - Custom generator streams - Thread safety - Abstract Base Classes.

Unit:2	**Data Manipulation and Aggregation**	**15 hours**

NumPy array operations - Pandas DataFrame scaling - Data cleaning - GroupBy summaries - Dataset relational merging.

Unit:3	Data Pipeline Engineering (ETL)	15 hours
Data format conversions - Database transaction connections - Operating System automation - Apache Airflow DAGs - Multiprocessing pipelines.		
Unit:4	**Web Scraping & API Automation**	**15 hours**
HTTP request handling - BeautifulSoup HTML extraction - Playwright browser automation - API rate-limiting - Automated exception retries.		
Unit:5	**Testing, Packaging, and Deployment**	**15 hours**
Unit testing frameworks - Pipeline mock simulators - Click CLI construction - Poetry dependency management - Docker containerization.		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5
CO1	2	3	3	1	2
CO2	2	3	3	1	2
CO3	2	3	3	1	2
CO4	2	3	3	1	2
CO5	2	3	3	1	2

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4
CO1	3	2	1	2
CO2	3	3	1	2
CO3	3	3	1	2
CO4	3	2	1	2
CO5	2	2	2	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Text Books:

- Wes McKinney, 'Python for Data Analysis', 3rd Ed., O'Reilly Media, 2022. (Covers Units: I - V) - Free full text: https://wesmckinney.com/book/
- Bas P. Harenslak, Julian Rutger de Ruiter, 'Data Pipelines with Apache Airflow', Manning Publications, 2021. (Covers Units: I - V) https://books.google.com/books?

q=Data+Pipelines+with+Apache+Airflow+Bas+P.+Harenslak%2C+Julian+Rutger+de+Ruiter

Reference Books:

- Luciano Ramalho, 'Fluent Python', 2nd Ed., O'Reilly Media, 2022. https://books.google.com/books?q=Fluent+Python+Luciano+Ramalho
- Eric Matthes, 'Python Crash Course', 3rd Ed., No Starch Press, 2023.– https://books.google.com/books?q=Python+Crash+Course+Eric+Matthes

Free Online Study Resources:

- Automate the Boring Stuff with Python, free open-access online book (Free) - https://automatetheboringstuff.com/
- Python for Data Analysis (Wes McKinney), free official online edition (Free) - https://wesmckinney.com/book/

SEMESTER – III
CORE – VI - MERN FULL-STACK DEVELOPMENT

Course Objectives:

The main objectives of this course are to:

1. Understand the JavaScript engine event loop mechanics and manage system environment security variables.
2. Model standard HTTP endpoints, set up path parameter captures, and manage CORS policies.
3. Enforce object-document structural relations, execute CRUD logs, and connect cloud databases.
4. Master functional hooks, manage component rendering scopes, and configure state parameters.
5. Orchestrate frontend-backend API bridges, apply encryption standards, and organize production builds.

Expected Course Outcomes:

On the successful completion of the course, student will be able to:

1	Create asynchronous non-blocking runtime layers in Node.js using ES6+ features.	K1,K2
2	Construct robust REST APIs using Express.js incorporating custom global error boundaries.	K2, K3
3	Design Mongoose data schemas and execute complex pipeline aggregations on MongoDB nodes.	K3, K4
4	Structure modular single-page React environments with state synchronizations and dynamic routes.	K4, K5
5	Enforce stateless token authorizations, neutralize scripting vectors, and organize full-stack deployments.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Unit:1	**JavaScript Basics and Node.js**	**15 hours**

ES6+ language features and array methods - Asynchronous execution using promises and async/await - Node.js architecture and event loop mechanics - Node Package Manager (NPM) dependency tracking - Environment variable security management using dotenv.

Unit:2	**Backend Development with Express.js**	**15 hours**

Express server initialization and request-response cycles - REST API design using standard HTTP verbs - Dynamic routing via path parameters and query strings - Built-in request body parsing middleware - Cross-Origin Resource Sharing (CORS) and global error handling.		
Unit:3	**Database Operations with MongoDB**	**15 hours**
NoSQL Fundamentals: document-based storage, collections, and a free cloud database on MongoDB Atlas. Basic CRUD Operations: insert, read, update, and delete documents. Introduction to Mongoose: connecting Express to MongoDB, defining data schemas, and validation rules. Data Relationships: embedding versus referencing documents across collections. Database Queries: filters using matching, sorting, and basic grouping.		
Unit:4	**Frontend Development with React.js**	**15 hours**
Single Page Application architecture and component rendering - State management via useState and data transfer using props - Lifecycle data fetching using the useEffect hook - Event handling and form input synchronization - Client-side page navigation with React Router.		
Unit:5	**App Integration and Security**	**15 hours**
Frontend-to-backend communication using Axios and Fetch API - Stateless user authentication via JSON Web Tokens (JWT) - Authorization mechanics and route protection - Application security safeguards against injection attacks - Production builds compilation and full-stack project deployment.		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5
CO1	2	2	3	1	3
CO2	2	2	3	1	3
CO3	2	2	3	1	3
CO4	2	2	3	1	3
CO5	2	2	3	1	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4
CO1	3	1	1	2
CO2	3	1	1	2

CO3	3	2	1	2
CO4	3	1	1	2
CO5	3	2	1	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Text Books:

- Ethan Brown, 'Web Development with Node and Express', 2nd Ed., O'Reilly Media, 2019. (Covers Units: I - V) https://books.google.com/books?q=Web+Development+with+Node+and+Express+Ethan+Brown
- Robin Wieruch, 'The Road to React', Independent Publisher, 2023. (Covers Units: I - V) https://books.google.com/books?q=The+Road+to+React+Robin+Wieruch

Reference Books:

- Brad Dayley, Brendan Dayley, Caleb Dayley, 'Node.js, MongoDB and Angular Web Development', 3rd Ed., Addison-Wesley, 2018. https://books.google.com/books?q=Node.js%2C+MongoDB+and+Angular+Web+Development+Brad+Dayley%2C+Brendan+Dayley%2C+Caleb+Dayley
- Azat Mardan, 'Full Stack JavaScript', 3rd Ed., Apress, 2018. https://books.google.com/books?q=Full+Stack+JavaScript+Azat+Mardan

Free Online Study Resources:

- The Odin Project - Full Stack JavaScript, free open-access curriculum (Free) - https://www.theodinproject.com/
- freeCodeCamp - Back End Development and APIs, free open-access curriculum (Free) - https://www.freecodecamp.org/learn/back-end-development-and-apis/

SEMESTER – III
ELECTIVE – III - FOUNDATION OF QUANTUM COMPUTING TECHNIQUES

Course Objectives:

The main objectives of this course are to:

1. Understand Dirac bra-ket notation systems, projection matrices, and expectation values.
2. Initialize state superpositions using quantum gates and build entangled Bell matrices.
3. Model hardware layouts, map CNOT operations, and establish teleportation channels.
4. Analyze Deutsch-Jozsa balance queries, analyze Shor matrices, and trace Fourier steps.
5. Isolate environmental noise patterns, analyze correction thresholds, and execute code packages.

Expected Course Outcomes:

On the successful completion of the course, student will be able to:

1	Formulate linear transformation operators and complex vector spectral decompositions inside Hilbert spaces.	K1, K2
2	Map unitary single-qubit transformations and coordinate paths upon geometric Bloch spheres.	K2, K3
3	Construct controlled multi-qubit logic matrices and verify the mathematical limits of the No-Cloning theorem.	K3, K4
4	Execute amplitude amplification routines and trace period identification steps for factoring algorithms.	K4, K5
5	Implement error tracking transformation matrices and assemble algorithms programmatically inside Qiskit.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Unit:1	**Quantum Mechanics & Linear Algebra Foundations**	**15 hours**

Complex vector spaces and Hilbert space transformations - Linear operator spectral decomposition - Dirac bra-ket mathematical notation systems - Unitary matrix state space characteristics - Hermitian projection eigenvalues and expectation values.

Unit:2	**Qubits and Quantum Gates**	**15 hours**

Single qubit state vector representations - Bloch Sphere geometric coordinate tracking -

Pauli-X, Y, and Z matrix transformations - Hadamard quantum gate superposition initialization - Multi-qubit entanglement construction via Bell states.		
Unit:3	**Quantum Circuits & Architectures**	**15 hours**
Quantum hardware circuit layout designs - Controlled-NOT (CNOT) logic mapping matrices - Three-qubit Toffoli gate configurations - No-Cloning Theorem mathematical verification - Superdense coding and Quantum Teleportation protocols.		
Unit:4	**Quantum Algorithms**	**15 hours**
Deutsch-Jozsa balanced function query logic - Simon period identification step sequences - Shor prime integer factorization algorithms - Grover database search amplitude amplification loops - Quantum Fourier Transform (QFT) phase matrices.		
Unit:5	**Quantum Error Correction & Programming**	**15 hours**
Quantum environmental noise and phase relaxation channels - Phase-flip error tracking transformation matrices - Shor nine-qubit error syndrome validation maps - Fault-tolerant execution threshold logic - Qiskit programmatic quantum algorithm assembly.		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5
CO1	3	2	2	1	1
CO2	3	2	2	1	1
CO3	3	2	2	1	1
CO4	3	2	2	1	1
CO5	3	2	2	1	1

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4
CO1	1	2	3	1
CO2	1	2	3	1
CO3	2	2	3	1
CO4	2	3	3	1
CO5	2	3	2	2

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Text Books:

- Michael A. Nielsen, Isaac L. Chuang, 'Quantum Computation and Quantum Information', 10th Anniversary Ed., Cambridge University Press, 2010. (Covers Units: I - V) https://books.google.com/books?q=Quantum+Computation+and+Quantum+Information+Michael+A.+Nielsen%2C+Isaac+L.+Chuang
- Eric R. Johnston, Nic Harrigan, Mercedes Gimeno-Segovia, 'Programming Quantum Computers', 1st Ed., O'Reilly Media, 2019. (Covers Units: I - V) https://books.google.com/books?q=Programming+Quantum+Computers+Eric+R.+Johnston%2C+Nic+Harrigan%2C+Mercedes+Gimeno-Segovia

Reference Books:

- Chris Bernhardt, 'Quantum Computing for Everyone', MIT Press, 2019. https://books.google.com/books?q=Quantum+Computing+for+Everyone+Chris+Bernhardt
- Jack D. Hidary, 'Quantum Computing: An Applied Approach', 2nd Ed., Springer, 2021
- https://books.google.com/books?q=Quantum+Computing%3A+An+Applied+Approach+Jack+D.+Hidary

Free Online Study Resources:

- IBM Quantum Learning, free official courses and digital textbook (Free) - https://quantum.cloud.ibm.com/learning/en
- Qiskit Textbook - Learn Quantum Computation using Qiskit, free open-access textbook (Free) - https://qiskit.org/learn

SEMESTER – III
ELECTIVE – III (OPTION) - LARGE LANGUAGE MODELS (LLM)

Course Objectives:

The main objectives of this course are to:

1. Analyze multi-head transformer layers, positional encoding formulas, and sequence normalizations.
2. Design vocabulary tokenizers, track model pre-training updates, and manage gradient steps.
3. Format instruction tuning datasets, run PEFT prompts, and measure evaluation benchmarks.
4. Model human preference tracking feedback loops and apply policy optimizations.
5. Construct vector indices, organize vLLM continuous batches, and configure automated evaluation logs.

Expected Course Outcomes:

On the successful completion of the course, student will be able to:

1	Calculate scaled dot-product attention sequence alignment and map causal masking vectors.	K1, K2
2	Manage distributed corpus data pipelines and optimize pre-training constraints via Chinchilla scaling laws.	K2, K3
3	Implement parameter-efficient low-rank adaptation matrix updates using quantized scaling.	K3, K4
4	Configure direct preference reward structures to neutralize semantic hallucination paths.	K4, K5
5	Deploy context-injected retrieval engines managed through multi-agent orchestration frameworks.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Unit:1	**The Transformer Architecture**	**12 hours**

Scaled Dot-Product Self-Attention mechanisms - Multi-head attention sequence matrix alignment - Sinusoidal positional coordinate encoding formulas - Transformer layer normalization structures - Decoder causal lookahead masking operations.

Unit:2	**LLM Pre-training Paradigms**	**12 hours**

Massive corpus filtering and deduplication pipelines - Byte-Pair Encoding (BPE)

vocabulary tokenization - Autoregressive language modeling sequence training - Compute-optimal scaling via Chinchilla laws - Distributed pre-training gradient synchronization.		
Unit:3	**Downstream Adaptation & Fine-Tuning**	**12 hours**
Supervised Instruction Fine-Tuning (SFT) data formatting - Parameter-Efficient Fine-Tuning (PEFT) prompt token insertions - Low-Rank Adaptation (LoRA) matrix updates - QLoRA 4-bit normal float quantization scaling - Standard evaluation benchmarking metrics.		
Unit:4	**Alignment & Reinforcement Learning**	**12 hours**
Human preference feedback ranking classifications - Reward model scoring head configurations - Proximal Policy Optimization (PPO) clipped loops - Direct Preference Optimization (DPO) log-likelihood objectives - Semantic hallucination penalty constraints.		
Unit:5	**Enterprise Integration & Deployment**	**12 hours**
Vector database cosine similarity search indexing - Retrieval-Augmented Generation (RAG) context injection - High-throughput inference via vLLM continuous batching - INT4 and INT8 engine precision model quantization - Automated LLM-as-a-judge multi-agent evaluation frameworks.		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5
CO1	2	2	3	1	3
CO2	2	2	3	1	3
CO3	2	2	3	1	3
CO4	2	2	3	1	3
CO5	2	2	3	1	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4
CO1	2	3	3	1
CO2	2	3	2	1
CO3	3	2	2	2
CO4	2	2	3	2

CO5	2	3	2	2

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Text Books:

- Sebastian Raschka, 'Build a Large Language Model (From Scratch)', Manning Publications, 2024. (Covers Units: I - V) https://books.google.com/books?q=Build+a+Large+Language+Model+%28From+Scratch%29+Sebastian+Raschka
- Sinan Ozdemir, 'Quick Start Guide to Large Language Models', 2nd Ed., Addison-Wesley, 2024. https://books.google.com/books?q=Quick+Start+Guide+to+Large+Language+Models+Sinan+Ozdemir (Covers Units: I - V)

Reference Books:

- Jay Alammar, Maarten Grootendorst, 'Hands-On Large Language Models', 1st Ed., O'Reilly Media, 2024. |
- https://books.google.com/books?q=Hands-On+Large+Language+Models+Jay+Alammar%2C+Maarten+Grootendorst
- Denis Rothman, 'Transformers for Natural Language Processing', 2nd Ed., Packt Publishing, 2022. https://books.google.com/books?q=Transformers+for+Natural+Language+Processing+Denis+Rothman

Free Online Study Resources:

- Hugging Face NLP Course, free official course (Free) - https://huggingface.co/learn/nlp-course
- Sebastian Raschka - free chapters and code for 'Build a Large Language Model (From Scratch)' (Free) - https://github.com/rasbt/LLMs-from-scratch

SEMESTER – III

PRACTICAL V - PYTHON FOR DATA ENGINEERING AND AUTOMATION LAB

Learning Objective (LO):

To automate systems, manage data pipelines, and clean large datasets in Python.

List of Programs:

01. Write a Python program to read a 1GB log file chunk by chunk using custom Generators.
02. Write a NumPy script to perform matrix multiplication, inversion, and row slicing.
03. Write a Pandas script to merge two CSV files, drop null values, and group entries by category.
04. Create a simple automated data workflow task (DAG) using Apache Airflow.
05. Write a script to speed up file processing using Python's multiprocessing library.
06. Write a web scraper using BeautifulSoup to extract headlines from an online news portal.
07. Write a program to request data from a public REST API and handle HTTP timeout errors.
08. Write a script that monitors a directory and copies files automatically when added.
09. Write a set of automated unit test scripts for mathematical functions using PyTest.
10. Create a Command Line Interface (CLI) application that accepts text flags using Click.

Expected Course Outcomes:

On the successful completion of this practical course, student will be able to:

1	CO1: Implement memory-efficient file processing using Python Generators and NumPy matrix operations.	K1, K2
2	CO2: Clean and merge datasets using Pandas and orchestrate workflows using Apache Airflow DAGs.	K2, K3
3	CO3: Apply multiprocessing techniques to optimize file processing performance.	K3, K4
4	CO4: Build web scrapers and REST API clients with robust error handling.	K4, K5
5	CO5: Develop automated test suites and Command Line Interface applications using PyTest and Click.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5
CO1	3	2	1	1	1
CO2	3	3	1	1	1
CO3	3	2	1	1	1
CO4	3	3	1	1	2
CO5	3	2	2	1	2

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4
CO1	3	2	1	2
CO2	3	3	1	2
CO3	3	2	1	2
CO4	3	2	1	2
CO5	2	2	2	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

SEMESTER – III

PRACTICAL -VI - MERN FULL STACK DEVELOPMENT LAB

Learning Objective (LO):

To build single-page web applications from scratch using Node, Express, MongoDB, and React.

List of Programs:

01. Write an asynchronous Node.js script to read, write, and rename a file using Promises.
02. Set up an Express.js server that listens on port 5000 and returns a welcome string.
03. Build a standard backend REST API in Express with endpoints for POST, GET, PUT, and DELETE.
04. Define a product data structure table (Schema) in Mongoose with simple text validations.
05. Write a script to connect an Express application to a cloud MongoDB Atlas database.
06. Create a simple React login form component that manages state using the useState hook.
07. Write a React component that fetches user lists from a backend API using useEffect.
08. Configure page routing paths (Home, About, Profile) inside a React app using React Router.
09. Connect a React frontend to an Express backend using Axios to pass secure headers.
10. Build and compile a single deployment package containing both backend and frontend code.

Expected Course Outcomes:

On the successful completion of this practical course, student will be able to:

1	CO1: Implement asynchronous file operations and a basic Express.js server in Node.js.	K1, K2
2	CO2: Build a complete REST API in Express with CRUD endpoints and Mongoose schema validation.	K2, K3
3	CO3: Connect an Express backend to a cloud MongoDB Atlas database.	K3, K4
4	CO4: Develop React components using state and effect hooks for dynamic user interfaces.	K4, K5
5	CO5: Integrate React frontend with Express backend using routing, Axios, and a unified deployment package.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5
CO1	3	1	1	1	1
CO2	3	2	1	1	1
CO3	3	2	2	1	1
CO4	3	2	1	1	1
CO5	3	2	1	2	2

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4
CO1	3	1	1	2
CO2	3	2	1	2
CO3	3	2	1	2
CO4	3	1	1	2
CO5	3	2	1	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

SEMESTER – III
SEC - FIGMA DESIGN LAB

Learning Objective (LO):
To gain hands-on, practical familiarity with the Figma design tool through basic design exercises.

List of Programs:

01.	Create a Figma account and explore the workspace interface and toolbar.
02.	Create a new design file and draw basic shapes (rectangle, ellipse, line).
03.	Use the Frame tool to create screens of different sizes (mobile, tablet, desktop).
04.	Add and format text on a design using the Text tool (font, size, and color).
05.	Apply solid colors and a simple gradient to shapes in a design.
06.	Group multiple objects together and align them using Figma's alignment tools.
07.	Import an image into Figma and place it within a design layout.
08.	Create a simple reusable component, such as a button, with one variant.
09.	Link two frames together to create a simple two-screen clickable prototype.
10.	Export a completed design as a PNG or PDF file.

Expected Course Outcomes:
On the successful completion of this practical course, student will be able to:

1	CO1: Navigate the Figma workspace and create basic shapes.	K1, K2
2	CO2: Use frames and text tools to design simple screen layouts.	K2, K3
3	CO3: Apply colors, alignment, and grouping to organize a design.	K3, K4
4	CO4: Create a simple reusable design component.	K4, K5
5	CO5: Build a simple interactive prototype and export design files.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5
CO1	1	1	3	1	1
CO2	1	1	3	1	1
CO3	1	2	3	1	1
CO4	1	2	3	1	1
CO5	1	2	3	1	2

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4
CO1	1	1	1	2
CO2	1	1	1	2
CO3	2	1	1	2
CO4	2	1	1	2
CO5	2	1	1	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

SEMESTER – III

NME – II - WEB DESIGN FOR BEGINNERS (HTML & CSS)

Course Objectives:

The main objectives of this course are to:

1. Understand browser parsing systems, text structures, and semantic character standards.
2. Master absolute and relative linking strategies and configure background multimedia files.
3. Apply table parameters (rowspan/colspan) and design functional input form templates.
4. Manage element text typographic stylings, select layouts, and manipulate margins.
5. Implement layout tracking modes, set float values, and structure grid frameworks.

Expected Course Outcomes:

On the successful completion of the course, student will be able to:

1	Construct structurally valid HTML5 document hierarchies using semantic formatting rules.	K1, K2
2	Integrate nested multi-level lists, hyperlinks, and complex absolute/relative media streams.	K2, K3
3	Formulate organized data tables and implement input validation fields for interactive forms.	K3, K4
4	Configure modular CSS style sheets, applying proper precedence selectors and box modeling rules.	K4, K5
5	Design fluid layouts using Flexbox structures and cross-device responsive media queries.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Unit:1	**Introduction to the Web & HTML5 Foundations**	**6 hours**

Web Browser Rendering Basics - Document Structure & Metadata (DOCTYPE, html, head, body) - Text Formatting (Headings, Paragraphs) - HTML5 Semantic Elements (article, section, nav, aside) - Character Encoding Standards (UTF-8).

Unit:2	**Lists, Links, and Media Integration**	**6 hours**

Ordered and Unordered Lists - Nested Lists - Hyperlinks (anchor tag) & Target Attributes - Relative vs. Absolute File Paths - Multimedia Integration (Inserting Images, Audio, and Video).

Unit:3	**Tables and Interactive Forms**	**6 hours**

HTML Data Tables (table, tr, td) - Table Rowspan and Colspan Attributes - Form Elements (form, input, textarea) - Input Types & Client-Side Form Validation Attributes.		
Unit:4	**Cascading Style Sheets (CSS) Basics**	**6 hours**
External, Internal, and Inline CSS Linking - CSS Selector Types (Element, Class, ID) & Precedence - The CSS Box Model (Margin, Border, Padding, Content Area) - Typography & Text Styling (Fonts, Colors).		
Unit:5	**Layout Design and Responsiveness**	**6 hours**
CSS Positioning Modes (Static, Relative, Absolute, Fixed) - Float & Clear Properties - Flexbox Layout (Flex-Direction, Alignment, Positioning) - CSS Grid Tracking Systems - Responsive Web Design via Media Queries.		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5
CO1	1	1	3	1	1
CO2	1	1	3	1	1
CO3	1	1	3	1	1
CO4	1	1	3	1	1
CO5	1	1	3	1	1

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4
CO1	2	1	1	3
CO2	2	1	1	3
CO3	2	1	1	3
CO4	2	1	1	3
CO5	2	2	1	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Text Books:

- Jon Duckett, 'HTML and CSS: Design and Build Websites', 1st Ed., Wiley, 2011. (Covers Units: I - V) https://books.google.com/books?q=HTML+and+CSS%3A+Design+and+Build+Websites+Jon+Duckett

- Jennifer Niederst Robbins, 'Learning Web Design', 5th Ed., O'Reilly Media, 2018. (Covers Units: I - V) https://books.google.com/books?q=Learning+Web+Design+Jennifer+Niederst+Robbins

Reference Books:

- Ben Frain, 'Responsive Web Design with HTML5 and CSS', 4th Ed., Packt Publishing, 2023. https://books.google.com/books?q=Responsive+Web+Design+with+HTML5+and+CSS+Ben+Frain
- Jon Duckett, 'JavaScript and JQuery: Interactive Front-End Web Development', 1st Ed., Wiley, 2014. https://books.google.com/books?q=JavaScript+and+JQuery%3A+Interactive+Front-End+Web+Development+Jon+Duckett

Free Online Study Resources:

- MDN Web Docs - HTML and CSS, free official documentation (Free) - https://developer.mozilla.org/en-US/docs/Learn/HTML
- freeCodeCamp - Responsive Web Design, free open-access curriculum (Free) - https://www.freecodecamp.org/learn/2022/responsive-web-design/

SEMESTER – IV

CORE – VII - ARTIFICIAL INTELLIGENCE AND MLOPS

Course Objectives:

The main objectives of this course are to:

1. Understand basic AI problem-solving and search techniques.
2. Learn the core machine learning lifecycle, from data to evaluation.
3. Understand why MLOps is needed and how to version code and data.
4. Learn the basics of tracking machine learning experiments.
5. Understand basic approaches to deploying and monitoring ML models.

Expected Course Outcomes:

On the successful completion of the course, student will be able to:

1	Apply basic search techniques to solve simple AI problems.	K1, K2
2	Apply the machine learning lifecycle, including data preparation and model evaluation.	K2, K3
3	Use Git for code versioning and explain the basics of data versioning.	K3, K4
4	Track machine learning experiments using a simple tracking tool.	K4, K5
5	Describe basic approaches to deploying and monitoring a machine learning model.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Unit:1	**AI Problem Solving Basics**	**15 hours**

State-space search problem modeling - Uninformed search (BFS, DFS) - Informed search (A*) basics - Alpha-Beta pruning overview.

Unit:2	**Core Machine Learning Lifecycle**	**15 hours**

Dataset preparation and train-test split - Decision Trees - Support Vector Machines (basics) - Clustering basics - Model evaluation metrics.

Unit:3	**Introduction to MLOps**	**15 hours**

Basics of MLOps - The machine learning project lifecycle - Code versioning with Git - Basic idea of data versioning.

Unit:4	**Experiment Tracking Basics**	**15 hours**

Why experiment tracking matters - Introduction to a simple experiment tracking tool (e.g.

MLflow) - Logging model parameters and results.		
Unit:5	**Model Deployment and Monitoring Basics**	**15 hours**
Deploying a model as a simple REST API - Basic idea of containerizing a model with Docker - Introduction to monitoring deployed models - Basic idea of model drift.		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5
CO1	3	2	2	1	1
CO2	3	3	2	1	1
CO3	2	2	2	1	1
CO4	2	2	2	1	1
CO5	2	2	2	1	2

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4
CO1	2	2	2	1
CO2	2	2	2	1
CO3	2	2	2	1
CO4	2	2	2	1
CO5	2	2	2	2

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Text Books:

- Stuart Russell, Peter Norvig, 'Artificial Intelligence: A Modern Approach', 4th Ed., Pearson, 2020. (Covers Units: I - V) https://books.google.com/books?q=Artificial+Intelligence%3A+A+Modern+Approach+Stuart+Russell%2C+Peter+Norvig
- Noah Gift, Alfredo Deza, 'Practical MLOps', 1st Ed., O'Reilly Media, 2021. (Covers Units: I - V) https://books.google.com/books?q=Practical+MLOps+Noah+Gift%2C+Alfredo+Deza

Reference Books:

- Mark Treveil et al., 'Introducing MLOps', 1st Ed., O'Reilly Media, 2020. https://books.google.com/books?q=Introducing+MLOps+Mark+Treveil+et+al.

- Andriy Burkov, 'Machine Learning Engineering', True Positive Inc, 2020. https://books.google.com/books?q=Machine+Learning+Engineering+Andriy+Burkov

Free Online Study Resources:

- Google - Machine Learning Crash Course, free official course (Free) - https://developers.google.com/machine-learning/crash-course
- Artificial Intelligence: A Modern Approach - free companion code and resources (Free) - https://aima.cs.berkeley.edu/

SEMESTER – IV

PRACTICAL – VII - ARTIFICIAL INTELLIGENCE AND MLOPS LAB

Learning Objective (LO):

To implement basic AI search algorithms and practice simple MLOps tracking and deployment.

List of Programs:

01. Write a Python program to solve the 8-Puzzle problem using the A* Search algorithm.
02. Write a program to implement Alpha-Beta Pruning for a simple Tic-Tac-Toe game.
03. Train a Decision Tree classification model on a sample dataset using Scikit-Learn.
04. Train a Support Vector Machine (SVM) model to classify a simple dataset.
05. Create a Git repository and practice committing changes to a simple ML script.
06. Use a simple experiment tracking tool (e.g. MLflow) to log model accuracy across two training runs.
07. Write a basic Dockerfile to package a simple Python ML script.
08. Deploy a trained model as a simple REST API using Flask or FastAPI.
09. Write a script to test the deployed model API with sample input data.
10. Write a simple script to compare predictions before and after retraining a model (basic drift check).

Expected Course Outcomes:

On the successful completion of this practical course, student will be able to:

1	CO1: Implement basic AI search algorithms including A* Search and Alpha-Beta Pruning.	K1, K2
2	CO2: Train and evaluate basic classification models using Scikit-Learn.	K2, K3
3	CO3: Use Git for version control of machine learning code.	K3, K4
4	CO4: Track experiments and log model metrics using a simple tracking tool.	K4, K5
5	CO5: Deploy a trained model as a REST API and perform a basic drift check.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5
CO1	3	2	2	1	1
CO2	3	2	2	1	1
CO3	2	2	2	1	1
CO4	2	2	2	1	1
CO5	2	2	2	1	2

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4
CO1	2	2	2	1
CO2	2	2	2	1
CO3	2	2	2	1
CO4	2	2	2	1
CO5	2	2	2	2

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

SEMESTER – IV

SEC – II - RESEARCH DOCUMENTATION AND PROFESSIONAL LAYOUT DESIGN (LaTeX)

Course Objectives:

The main objectives of this course are to:

1. Understand the basics of LaTeX and how it differs from regular word processors.
2. Learn to create well-structured documents with sections, lists, and references.
3. Learn to typeset mathematical equations using LaTeX.
4. Learn to insert images, tables, and figures into a LaTeX document.
5. Learn to manage references and citations using BibTeX.

Expected Course Outcomes:

On the successful completion of the course, student will be able to:

1	Explain the basics of LaTeX and create a simple academic document.	K1, K2
2	Create structured documents with proper formatting, lists, and cross-references.	K2, K3
3	Typeset mathematical equations and expressions using LaTeX.	K3, K4
4	Insert and format images, tables, and figures in a LaTeX document.	K4, K5
5	Manage references and citations in a research document using BibTeX.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Unit:1	**Introduction to LaTeX**	**6 hours**

What is LaTeX and how it is different from Word - Setting up a LaTeX environment (Overleaf) - Basic document structure - Document classes (article, report) - Basic text formatting (bold, italic).

Unit:2	**Document Structuring**	**6 hours**

Sections and subsections - Creating lists (numbered and bullet) - Adding hyperlinks - Setting page margins - Adding headers and footers - Cross-referencing sections and figures.

Unit:3	**Typing Mathematical Equations**	**6 hours**

Basic math mode - Writing simple equations - Fractions, powers, and roots - Writing multi-line equations - Basic matrices.

Unit:4	Working with Images and Tables	6 hours
Inserting images - Resizing and positioning images - Creating simple tables - Adding captions to figures and tables - Placing figures and tables on the page.		
Unit:5	**References and Project Organization**	**6 hours**
Introduction to BibTeX - Creating a simple bibliography - Citing references in text - Organizing a large document using multiple files - Compiling the final document.		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5
CO1	2	1	3	1	1
CO2	2	1	3	1	1
CO3	2	1	3	1	1
CO4	2	1	3	1	1
CO5	2	1	3	1	2

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4
CO1	1	1	2	2
CO2	1	1	2	2
CO3	1	1	2	2
CO4	1	1	2	2
CO5	1	1	2	2

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Text Books:

- Leslie Lamport, 'LaTeX: A Document Preparation System', 2nd Ed., Addison-Wesley, 1994. (Covers Units: I - V) https://books.google.com/books?q=LaTeX%3A+A+Document+Preparation+System+Leslie+Lamport
- Frank Mittelbach, Michel Goossens, 'The LaTeX Companion', 2nd Ed., Addison-Wesley, 2004. (Covers Units: I - V) https://books.google.com/books?q=The+LaTeX+Companion+Frank+Mittelbach%2C+Michel+Goossens

Reference Books:

- Tobias Oetiker, Hubert Partl, Irene Hyna, Elisabeth Schlegl, 'The Not So Short Introduction to LaTeX2e', Open Publication, 2022. - Free full text: https://tobi.oetiker.ch/lshort/lshort.pdf
- Stefan Kottwitz, 'LaTeX Beginner's Guide', 2nd Ed., Packt Publishing, 2021. https://books.google.com/books?q=LaTeX+Beginner+Stefan+Kottwitz

Free Online Study Resources:

- The Not So Short Introduction to LaTeX2e, free official PDF guide (Free) - https://tobi.oetiker.ch/lshort/lshort.pdf
- Overleaf Documentation, free official LaTeX learning resource (Free) - https://www.overleaf.com/learn