

MCA PROGRAM CURRICULUM - 2025

(Applicable for the batches admitted from A.Y. 2025-26)

Version 2.0

MASTER OF COMPUTER APPLICATIONS

Version 1.0–Approved in the BoS meeting on 17-05-2025 & Academic Council meeting on 14-06-2025

Version 2.0–Approved in the BoS meeting on 13-02-2026 & Academic Council meeting on 28-02-2026



A D I T Y A
U N I V E R S I T Y

Aditya Nagar, ADB Road, Surampalem - 533 437

VISION & MISSION OF THE UNIVERSITY

VISION:

To be a globally recognized University through excellence in Education, Innovation and Sustainable growth.

MISSION:

Deliver collaborative education to prepare students for global challenges through

- Transformative learning
- Vibrant research ecosystem
- Sustainable and inclusive community

VISION & MISSION OF THE DEPARTMENT

VISION:

To become a globally recognized center of excellence in computer applications, fostering innovation, transformative education, and sustainable technological advancement for societal benefit.

MISSION:

- M1:** Deliver transformative and industry-relevant education through cutting-edge pedagogy and collaborative learning.
- M2:** Advance impactful research in computing, fostering innovation and knowledge creation.
- M3:** Develop sustainable and inclusive technological solutions that address societal and global challenges.

PROGRAM OUTCOMES (POs)

After successful completion of the program, the Post Graduates will be able to

- PO 1 Foundation Knowledge:** Apply knowledge of mathematics, programming logic and coding fundamentals for solution architecture and problem solving.
- PO 2 Problem Analysis:** Identify, review, formulate and analyse problems for primarily focussing on customer requirements using critical thinking frameworks.
- PO 3 Development of Solutions:** Design, develop and investigate problems with as an innovative approach for solutions incorporating ESG/SDG goals.
- PO 4 Modern Tool Usage:** Select, adapt and apply modern computational tools such as development of algorithms with an understanding of the limitations including human biases.
- PO 5 Individual and Teamwork:** Function and communicate effectively as an individual or a team leader in diverse and multidisciplinary groups. Use methodologies such as agile.
- PO 6 Project Management and Finance:** Use the principles of project management such as scheduling, work breakdown structure and be conversant with the principles of Finance for profitable project management.
- PO 7 Ethics:** Commit to professional ethics in managing software projects with financial aspects. Learn to use new technologies for cyber security and insulate customers from malware
- PO 8 Life-Long Learning:** Change management skills and the ability to learn, keep up with contemporary technologies and ways of working.

PROGRAM SPECIFIC OUTCOMES (PSOs)

After successful completion of the program, the Post Graduates will be able to

- PSO 1** Design, develop and deploy efficient and secure software solutions by applying modern programming practices, system architecture, databases and networking principles to address the real-world problems.
- PSO 2** Apply emerging technologies such as artificial intelligence, cloud computing and data analytics to create innovative and scalable solutions for diverse industry sectors.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

Post Graduates of the program will

- PEO 1** Engage in professional practice and promote the development of innovative systems to optimize the solutions for Computer Science and Engineering problems.
- PEO 2** Achieve peer-recognition, as an individual or in a team through good analytical, research, design and implementation skills.
- PEO 3** Contribute to society as broadly educated, expressive, ethical and responsible citizens with proven expertise.

DEPARTMENT OF COMPUTER APPLICATIONS

Master of Computer Applications

(MCA)Program Curriculum - 2025

Credit Division:

S.No.	Category of Course	Credits
1	Program Core Courses (PCC)	48
2	Program Elective Courses (PEC)	15
3	Ability Enhancement Courses (AEC)	03
4	Technical Paper Publication (TPP)	02
5	Summer Internship (SI)	02
6	Project Work (PROJ)	10
7	Mandatory Courses (MC)	00
Total Credits		80

Program Core Courses (PCC)

Course Code	Course Title	L	T	P	C	CIE	SEE	Total	Pre-requisite
2504MC01	Computational Mathematics	2	1		3	50	50	100	-
2504MC02*/ 2604MC01#	Object Oriented Software Engineering	2	1		3	50	50	100	-
		2			2				
2504MC03	Artificial Intelligence	2		1	3	50	50	100	-
2504MC04	Data Warehousing and Mining	2		1	3	50	50	100	-
2504MC05*/ 2604MC02#	Advanced Java Programming	2		2	4	50	50	100	-
2504MC06*/ 2604MC03#	Python Programming	1		2	3	50	50	100	-
		2		2	4				
2504MC07	Advanced Data Structures and Algorithms	2		1	3	50	50	100	-
2504MC08	Computer Networks	2		1	3	50	50	100	-
2504MC09	Software Project and Testing Management	2		1	3	50	50	100	-
2504MC10*/ 2604MC04#	Machine Learning with Python	2		2	4	50	50	100	Python
2504MC11*/ 2604MC05#	MERN Stack Technologies	1		2	3	50	50	100	-
		2		1	3				
2504MC21*/ 2604MC06#	Fundamentals of DataScience			2	2	50	50	100	Python
2504MC12	Internet of Things (IoT)			2	2	50	50	100	-
2504MC13	Agile Methodologies	1		2	3	50	50	100	-
2504MC14	Cloud Computing	3			3	50	50	100	-
2504MC15	Operating Systems	2	1		3	50	50	100	-

Program Elective Courses (PEC)

Specialization: Machine Intelligence (MI)

Course Code	Course Title	L	T	P	C	CIE	SEE	Total	Pre-requisite
2504MC16	Information Retrieval Systems	3			3	50	50	100	ADSA
2504MC17	Recommender Systems	3			3	50	50	100	MLP
2504MC18	Neural Networks	2	1		3	50	50	100	AI
2504MC19	Deep Learning	2		1	3	50	50	100	MLP
2504MC20	Natural Language Processing	2	1		3	50	50	100	AI

Specialization: Data Engineering (DE)

Course Code	Course Title	L	T	P	C	CIE	SEE	Total	Pre-requisite
2504MC49	Foundations of Data Engineering	2		1	3	50	50	100	Python/AJP
2504MC22	NoSQL Databases	2		1	3	50	50	100	-
2504MC23	Big Data Analytics	3			3	50	50	100	-
2504MC24	Data Visualization	2		1	3	50	50	100	-
2504MC25	Gen AI & Prompt Engineering	2		1	3	50	50	100	-

Specialization: Network Security (NS)

Course Code	Course Title	L	T	P	C	CIE	SEE	Total	Pre-requisite
2504MC26	Ethical Hacking	2	1		3	50	50	100	-
2504MC27	Cyber Security	3			3	50	50	100	-
2504MC28	Network Security and Cryptography	2	1		3	50	50	100	CN
2504MC29	Block Chain Technologies	3			3	50	50	100	-
2504MC30	Secure Coding Techniques	3			3	50	50	100	CN

Specialization: Full Stack Development (FSD)

Course Code	Course Title	L	T	P	C	CIE	SEE	Total	Pre-requisite
2504MC31	Applied Modern Technologies	3			3	50	50	100	-
2504MC32	Web Design and Development	2		1	3	50	50	100	-
2504MC33	Advanced JavaScript	2		1	3	50	50	100	-
2504MC34	Client-side Scripting	2		1	3	50	50	100	-
2504MC35	Server-side Scripting	2		1	3	50	50	100	-

Specialization: Foundations of Cloud Technology (FCT)

Course Code	Course Title	L	T	P	C	CIE	SEE	Total	Pre-requisite
2504MC31	Applied Modern Technologies	3			3	50	50	100	-
2504MC36	Cloud Essentials	2		1	3	50	50	100	-
2504MC37	Cloud Architecture, Services and Storage	2		1	3	50	50	100	-
2504MC38	Cloud and DevOps	2		1	3	50	50	100	-
2504MC39	Cloud Engineering	2		1	3	50	50	100	-

Specialization: Software Testing & Quality Assurance (STQA)

Course Code	Course Title	L	T	P	C	CIE	SEE	Total	Pre-requisite
2504MC31	Applied Modern Technologies	3			3	50	50	100	-
2504MC40	Fundamentals of Product Management	2		1	3	50	50	100	-
2504MC41	Software Quality Assurance	2		1	3	50	50	100	-
2504MC42	Software Architecture	2		1	3	50	50	100	-
2504MC43	Software Verification, Validation and Testing	2		1	3	50	50	100	SPTM

Ability Enhancement Courses (AEC)

Course Code	Course Title	L	T	P	C	CIE	SEE	Total	Pre-requisite
2504MC44	Design Thinking			1	1	100	0	100	-
2504MC45	Student Activity-Based Learning				2	100	0	100	-

Technical Paper Publication (TPP)

Course Code	Course Title	L	T	P	C	CIE	SEE	Total	Pre-requisite
2504MC46	Technical Paper Publication			2	2	100	0	100	-

Summer Internship (SI)

Course Code	Course Title	L	T	P	C	CIE	SEE	Total	Pre-requisite
2504MC47	Summer Internship			2	2	100	0	100	-

Project (PROJ)

Course Code	Course Title	L	T	P	C	CIE	SEE	Total	Pre-requisite
2504MC48	Project Work			10	10	50	50	100	-

Mandatory Courses (MC)

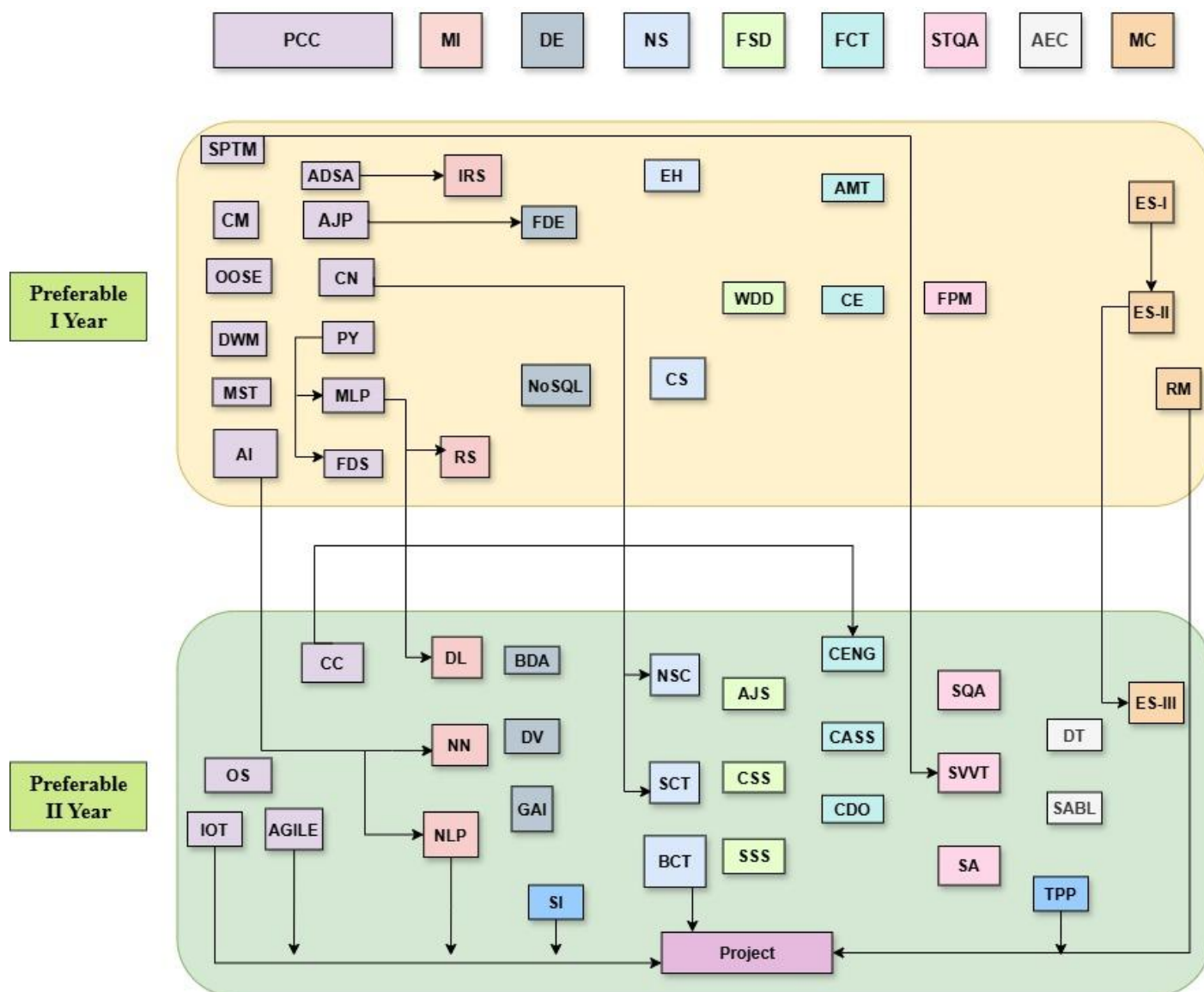
Course Code	Course Title	L	T	P	C	CIE	SEE	Total	Pre-requisite
2504AC01	Research Methodology	2				100	0	100	-
2504AC02	Employability Skills-I			3		100	0	100	-
2504AC03	Employability Skills-II			3		100	0	100	ES-I

Note: *Courses are applicable only to students admitted in the year 2025.

#Courses are applicable to the students admitted from the year 2026 onwards.

All the other courses are applicable to students admitted from the year 2025 onwards.

2025 MCA CURRICULUM PREREQUISITE FLOW CHART



Preferable I Year Courses	
CM	Computational Mathematics
OOSE	Object Oriented Software Engineering
AI	Artificial Intelligence
DWM	Data Warehousing and Mining
AJP	Advanced Java Programming
PY	Python Programming
ADSA	Advanced Data Structures and Algorithms
CN	Computer Networks
SPTM	Software Project and Testing Management
MLP	Machine Learning with Python
MST	MERN Stack Technologies
IRS	Information Retrieval Systems
RS	Recommender Systems
FDS	Fundamentals of Data Science
NoSQL	NoSQL Databases
EH	Ethical Hacking
CS	Cyber Security
AMT	Applied Modern Technologies
WDD	Web Design and Development
CE	Cloud Essentials
FPM	Fundamentals of Product Management
FDE	Foundations of Data Engineering
RM	Research Methodology
ES-I	Employability Skills-I

Preferable II Year Courses	
AGILE	Agile Methodologies
OS	Operating Systems
CC	Cloud Computing
IOT	Internet of Things (IoT)
NN	Neural Networks
DL	Deep Learning
NLP	Natural Language Processing
BDA	Big Data Analytics
DV	Data Visualization
GAI	Gen AI & Prompt Engineering
NSC	Network Security and Cryptography
SCT	Secure Coding Techniques
BCT	Block Chain Technologies
AJS	Advanced JavaScript
CSS	Client-side Scripting
SSS	Server-side Scripting
CASS	Cloud Architecture, Services and Storage
CDO	Cloud and DevOps
CENG	Cloud Engineering
SQA	Software Quality Assurance
SVVT	Software Verification, Validation and Testing
SA	Software Architecture
DT	Design Thinking
ES-II	Employability Skills-I
SABL	Student Activity Based Learning
TPP	Technical Paper Publication
SI	Summer Internship
Project	Project Work

Suggestive Semester wise Curriculum

I Semester

Course Code	Course Title	Course Category	Credits				Total Hours
			L	T	P	Total	
2504MC01	Computational Mathematics	PCC	2	1		3	3
2504MC02*/ 2604MC01#	Object Oriented Software Engineering	PCC	2	1		3	3
			2			2	2
2504MC03	Artificial Intelligence	PCC	2		1	3	4
2504MC04	Data Warehousing and Mining	PCC	2		1	3	4
2504MC05*/ 2604MC02#	Advanced Java Programming	PCC	2		2	4	6
2504MC06*/ 2604MC03#	Python Programming	PCC	1		2	3	5
			2		2	4	6
2504AC01	Research Methodology	MC	2				
Total			13	2	6	19	25

II Semester

Course Code	Course Title	Course Category	Credits				Total Hours
			L	T	P	Total	
2504MC07	Advanced Data Structures and Algorithms	PCC	2		1	3	3
2504MC08	Computer Networks	PCC	2		1	3	3
2504MC09	Software Project and Testing Management	PCC	2		1	3	4
2504MC10*/ 2604MC04#	Machine Learning with Python	PCC	2		2	4	6
2504MC11*/ 2604MC05#	MERN Stack Technologies	PCC	1		2	3	5
			2		1		4
2504MC21*/ 2604MC06#	Fundamentals of Data Science	PCC			2	2	4
	Elective – I	PEC				3	4
2504AC02	Employability Skills-I	MC			3		
Total			9	2	10	21	29

III Semester

Course Code	Course Title	Course Category	Credits				Total Hours
			L	T	P	Total	
2504MC12	Internet of Things (IoT)	PCC			2	2	4
2504MC13	Agile Methodologies	PCC	1		2	3	5
2504MC14	Cloud Computing	PCC	3			3	3
2504MC15	Operating Systems	PCC	2	1		3	3
	Elective – II	PEC				3	4
	Elective- III	PEC				3	4
2504MC44	Design Thinking	AEC			1	1	2
2504MC47	Summer Internship	SI			2	2	
2504AC03	Employability Skills-II	MC			3		
Total			6	1	10	20	25

IV Semester

Course Code	Course Title	Course Category	Credits				Total Hours
			L	T	P	Total	
	Elective - IV	PEC				3	4
	Elective - V	PEC				3	4
2504MC45	Student Activity-Based Learning	AEC				2	
2504MC46	Technical Paper Publication	TPP			2	2	
2504MC48	Project Work	PROJ			10	10	
Total					12	20	08

Note: *Courses are applicable only to students admitted in the year 2025.

#Courses are applicable to the students admitted from the year 2026 onwards.

All the other courses are applicable to students admitted from the year 2025 onwards.

COMPUTATIONAL MATHEMATICS

Course Code: 2504MC01

L	T	P	C
2	1	0	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Apply various probability distributions for both discrete and continuous random variables
- CO2:** Compute mean and variance of sample means with replacement, without replacement and maximum errors
- CO3:** Apply various tests to test the hypothesis concerning mean, Proportion and variance.
- CO4:** Apply the concepts of group theory and number theory.
- CO5:** Apply the knowledge of graph theory to solve real world problems like minimum Spanning Tree - traversal of binary tree.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	2	-	-	-	-	-	-
CO2	3	2	-	-	-	-	-	-
CO3	2	3	-	-	-	-	-	-
CO4	3	2	-	-	-	-	-	-
CO5	3	2	-	-	-	-	-	-

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1	1	-
CO2	1	-
CO3	1	-
CO4	1	-
CO5	1	-

UNIT – I

Random variables and probability distributions: Conditional Probability-Bayes's theorem- Random variables – Discrete and Continuous random variables – Distribution function – Mathematical Expectation, Variance - Binomial, Poisson, Normal distributions.

UNIT – II

Descriptive Statistics and Sampling Theory: Computation of mean, median, mode and variance of grouped and ungrouped data–Population and samples – Sampling with and without replacement-Sampling distribution of Means and Variance – Central limit theorem (without proof)-Point and Interval estimations – Maximum error of estimate.

UNIT – III

Tests of Hypothesis: Introduction – Hypothesis – Null and Alternative Hypothesis – Type I and Type II errors – Level of significance – One tail and two-tail tests – Tests concerning one mean and two means (Large and Small samples) – Tests on proportions, F test and Chi-Square test.

UNIT – IV

Algebraic Structures and Number Theory: Algebraic Structures, Semi-Groups, Monoids, Groups, Subgroups, Abelian Group-Homomorphism, Isomorphism. Number Theory: Properties of Integers, Division Theorem, Greatest Common Divisor, Euclidean Algorithm, Least Common Multiple, Testing for Prime Numbers, The Fundamental Theorem of Arithmetic, Modular Arithmetic, Fermat's and Euler's Theorems.

UNIT – V

Graph Theory: Basic Concepts of Graphs, Matrix Representation of Graphs: Adjacency Matrix, Incidence Matrix, Isomorphic Graphs, Paths and Circuits, Euler and Hamilton Graphs, Planar Graphs, and Euler's Formula. Tree Properties, Spanning Trees, BFS Algorithm, DFS Algorithm, Minimal Spanning Trees and Kruskal's Algorithm, Graph coloring, chromatic number.

Textbooks:

1. Probability and Statistics for Engineers, Miller and Freund's, 7/e, Pearson, ISBN-13: 978-8120328341
2. Discrete Mathematics and its Applications with Combinatorics and Graph Theory, Kenneth H. Rosen, McGraw Hill Education (India) Private Limited. ISBN: 978-9390727353

Reference Books:

1. Fundamentals of Mathematical Statistics, S. C. Gupta and V.K. Kapoor, 11/e, Sultan Chand & Sons Publications, ISBN-13: 978-8180540042
2. Probability and statistics by T.K.V. Iyengar, S. Chand publishers, ISBN-13: 978-9355010643
3. Discrete Mathematics for Computer Scientists and Mathematicians, J. L. Mott, A. Kandel, T.P, ISBN-13: 978-81-203-1502-0

Web Links:

1. <http://mathworld.wolfram.com/topics/ProbabilityandStatistics.html>
2. <http://nptel.ac.in/courses/111105041/1>
3. <http://mathworld.wolfram.com/classroom/classes/DiscreteMathematics.html>
4. <https://ggc-discrete-math.github.io/>
5. <http://nptel.ac.in/courses/106106094/>

OBJECT ORIENTED SOFTWARE ENGINEERING

Course Code: 2504MC02

L	T	P	C
2	1	0	3

Course Outcomes:

At the end of the Course, Student will be able to:

CO1: Understand classes, objects, inheritance, aggregation, polymorphism.

CO2: Create and analyze use cases.

CO3: Construct class diagrams with associations and multiplicity

CO4: Implement classes based on interaction and state diagrams.

CO5: Implement Model Driven Development.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	-	-	2	-	-	-	-
CO2	2	2	2	-	-	-	-	-
CO3	2	2	-	2	-	-	-	-
CO4	2	2	2	-	-	2	-	-
CO5	2	1	2	-	2	2	-	-

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1	3	-
CO2	-	2
CO3	-	2
CO4	2	2
CO5	-	-

UNIT – I

Introduction to Object Oriented Software Engineering: The nature of software-Types of software- Characteristic of software-Stakeholders in software engineering – SDLC Process Models- Waterfall, RAD, Agile Software Development, Classes and objects, Inheritance, types of inheritance, Aggregation, Instance variables, Methods, operations and polymorphism, Organizing classes into inheritance hierarchies.

UNIT – II

Developing Requirements and Domain Analysis: Functional Requirements and Non-Functional requirements, Requirements gathering, object-based requirements analysis, Use cases: describing how the user will use the system, techniques for gathering requirements, managing changing requirements, class-based requirements design.

UNIT – III

Introduction to UML: Essentials of UML class diagrams – Use case diagram, Activity diagram, Class diagram with Associations and multiplicity, Generalization and More advanced features of class diagrams.

UNIT – IV

User-Centered Design: Characteristics of users, The basics of user interface design, Usability principles, Evaluating user interfaces, Modeling interactions and behavior - Interaction diagrams, State diagrams, Activity diagrams, Implementing classes based on interaction and state diagrams, Difficulties and risks in modeling interactions and behavior.

UNIT – V

Architecting and Designing Software: The process of design - Principles leading to good design - Design Principles- Techniques for making good design decisions - Model Driven Development, Incorporating reusability and reuse into software engineering, Frameworks: reusable subsystems, the client-server architecture, Technology needed to build client-server systems -The Object Client-Server Framework (OCSF).

Textbooks:

1. Object-Oriented Software Engineering Practical Software Development using UML and Java, Timothy C Lethbridge, 3rd Edition, McGraw-Hill Higher Education, ISBN-13: 978-0077109080.
2. Object-Oriented Software Engineering: A Use Case Driven Approach, Ivar Jacobson, 1st Edition, Addison Wesley Longman Publishing, ISBN-13: 978-0201544350.

Reference Books:

1. "Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development" by Craig Larman, 3rd Edition, Prentice Hall, ISBN-13: 978-0131489066.
2. "Design Patterns: Elements of Reusable Object-Oriented Software" by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides (The Gang of Four), 1st Edition, Addison-Wesley Professional, ISBN-13: 978-0201633610.
3. "Object-Oriented Software Engineering Using UML, Patterns, and Java" by Bernd Bruegge and Allen H. Dutoit, 3rd Edition, Pearson, ISBN-13: 978-0136061250.

Web Links:

1. <https://www.javatpoint.com/software-engineering-object-oriented-design>.
2. <https://www.institutedata.com/blog/object-oriented-programming>.
3. <https://www.scaler.com/topics/software-engineering/object-oriented-design/>

OBJECT ORIENTED SOFTWARE ENGINEERING

Course Code: 2604MC01

L	T	P	C
2	0	0	2

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Understand classes, objects, inheritance, aggregation, polymorphism.
- CO2:** Create and analyze use cases.
- CO3:** Construct class diagrams with associations and multiplicity
- CO4:** Implement classes based on interaction and state diagrams.
- CO5:** Implement Model Driven Development.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	-	-	2	-	-	-	-
CO2	2	2	2	-	-	-	-	-
CO3	2	2	-	2	-	-	-	-
CO4	2	2	2	-	-	2	-	-
CO5	2	1	2	-	2	2	-	-

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1	3	-
CO2	-	2
CO3	-	2
CO4	2	2
CO5	-	-

UNIT – I

Introduction to Object Oriented Software Engineering: The nature of software-Types of software- Characteristic of software-Stakeholders in software engineering – SDLC Process Models- Waterfall, RAD, Agile Software Development, Classes and objects, Inheritance, types of inheritance, Aggregation, Instance variables, Methods, operations and polymorphism, Organizing classes into inheritance hierarchies.

UNIT – II

Developing Requirements and Domain Analysis: Functional Requirements and Non-Functional requirements, Requirements gathering, object-based requirements analysis, Use cases: describing how the user will use the system, techniques for gathering requirements, managing changing requirements, class-based requirements design.

UNIT – III

Introduction to UML: Essentials of UML class diagrams – Use case diagram, Activity diagram, Class diagram with Associations and multiplicity, Generalization and More advanced features of class diagrams.

UNIT – IV

User-Centered Design: Characteristics of users, The basics of user interface design, Usability principles, Evaluating user interfaces, Modeling interactions and behavior - Interaction diagrams, State diagrams, Activity diagrams, Implementing classes based on interaction and state diagrams, Difficulties and risks in modeling interactions and behavior.

UNIT – V

Architecting and Designing Software: The process of design - Principles leading to good design - Design Principles- Techniques for making good design decisions - Model Driven Development, Incorporating reusability and reuse into software engineering, Frameworks: reusable subsystems, the client-server architecture, Technology needed to build client-server systems -The Object Client-Server Framework (OCSF).

Textbooks:

1. Object-Oriented Software Engineering Practical Software Development using UML and Java, Timothy C Lethbridge, 3rd Edition, McGraw-Hill Higher Education, ISBN-13: 978-0077109080.
2. Object-Oriented Software Engineering: A Use Case Driven Approach, Ivar Jacobson, 1st Edition, Addison Wesley Longman Publishing, ISBN-13: 978-0201544350.

Reference Books:

1. "Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development" by Craig Larman, 3rd Edition, Prentice Hall, ISBN-13: 978-0131489066.
2. "Design Patterns: Elements of Reusable Object-Oriented Software" by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides (The Gang of Four), 1st Edition, Addison-Wesley Professional, ISBN-13: 978-0201633610.
3. "Object-Oriented Software Engineering Using UML, Patterns, and Java" by Bernd Bruegge and Allen H. Dutoit, 3rd Edition, Pearson, ISBN-13: 978-0136061250.

Web Links:

1. <https://www.javatpoint.com/software-engineering-object-oriented-design>.
2. <https://www.institutedata.com/blog/object-oriented-programming>.
3. <https://www.scaler.com/topics/software-engineering/object-oriented-design/>

ARTIFICIAL INTELLIGENCE

Course Code: 2504MC03

L	T	P	C
2	0	1	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Use different search methods like BFS, DFS, and heuristic search to solve problems like the water jug problem and Traveling Salesman Problem.
- CO2:** Represent knowledge and use logical reasoning with propositional and first-order predicate logic.
- CO3:** Build and use Bayesian networks and fuzzy logic systems to handle uncertain information.
- CO4:** Create expert systems using techniques like forward chaining and blackboard architecture.
- CO5:** Apply techniques for understanding and processing human language.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	1	2	3	-	-	-	1
CO2	2	-	2	-	-	-	-	-
CO3	3	3	-	1	-	-	2	2
CO4	3	1	3	2	-	-	2	-
CO5	-	3	3	3	1	-	1	1

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1	3	2
CO2	3	3
CO3	-	2
CO4	3	-
CO5	-	3

UNIT – I

Problem Solving Techniques: State Space Search: Production systems, Problem characteristics, Production system Characteristics Uninformed Search: Breadth first search, Depth first search, comparing DFS and BFS Informed Search: Heuristic search, Hill climbing, Best First Search-A* algorithm.

Practice:

1. Explain the A* algorithm with an example. Why is it considered optimal and complete?
2. Apply Hill climbing on a sample heuristic function and show the search path. Discuss its limitations.
3. Compare and contrast Breadth First Search (BFS) and Depth First Search (DFS) in terms of time complexity, completeness, and space requirements.
4. Differentiate between production systems and problem characteristics with suitable examples.

UNIT – II

Knowledge Representation: Procedural Vs Declarative Knowledge, Representations and Mappings, Approaches to Knowledge Representation, Issues in Knowledge Representation, Logic Programming Forward Vs Backward Reasoning Symbolic Logic: Propositional Logic, First Order Predicate Logic: Representing Instance and is a Relationships, Computable Functions and Predicates, Syntax & Semantics of FOPL, Normal Forms, Unification & Resolution, Representation Using Rules, and Natural Deduction.

Practice:

1. Describe forward chaining and backward chaining with inference rule examples.
2. Represent the following statement using First-Order Predicate Logic (FOPL): “All humans are mortal. Socrates is a human.”
3. Convert the following into CNF (Conjunctive Normal Form) and apply resolution to prove a conclusion.
 - If it rains, the ground gets wet.
 - If the ground gets wet, the match is cancelled.
 - It is raining. Prove: The match is cancelled.
4. Define unification in FOPL. Show how unification works on two example expressions.

UNIT – III

Reasoning Under Uncertainty: Introduction to Non-Monotonic Reasoning, Truth Maintenance Systems, Statistical Reasoning: Bayes Theorem, Certainty Factors & Rule-Based Systems, Bayesian Probabilistic Inference, Bayesian Networks, Dempster-Shafer Theory, Fuzzy Logic & Fuzzy Systems.

Practice:

1. Construct a basic Bayesian Network for a medical diagnosis scenario and compute posterior probabilities.
2. Compare certainty factors and Dempster-Shafer theory in handling uncertain knowledge.
3. Explain Bayes’ Theorem & solve a real-life conditional probability problem using it.

UNIT – IV

Experts Systems: Overview of an Expert System, Structure of an Expert Systems, Different Types of Expert Systems- Rule Based, Model Based, Case Based and Hybrid Expert Systems, Knowledge Acquisition and Validation Techniques, Black Board Architecture, Knowledge Building System Tools, Expert System Shells.

Practice:

1. Differentiate between case-based, model-based, and hybrid expert systems with examples.
2. Write short notes on any two expert system shells and their use in application development.
3. What is the role of blackboard architecture in expert systems? Illustrate its workflow.

UNIT – V

Natural Language Processing: Role of Knowledge in Language Understanding, Approaches Natural Language Understanding, Steps in The Natural Language Processing.

Practice:

1. Compare top-down and bottom-up approaches in NLP.
2. Explain syntactic and semantic analysis in NLP with suitable examples.
3. What is the role of knowledge in natural language understanding? Give examples.

Textbooks:

1. Artificial Intelligence by Elaine Rich, McGraw-Hill Publications, 3rd edition, ISBN-10: 0070087709, ISBN-13: 978-0070087705
2. Deep Learning: A practitioner's approach, Josh Patterson and Adam Gibson, First edition, Shroff Publishers and Distributors Pvt. Ltd., ISBN-10: 1491914254, ISBN-13: 978-1491914250

Reference Books:

1. Artificial Intelligence: A Modern Approach by Stuart Russell, Peter Norvig, 3rd edition, Pearson Education Publications, ISBN-10: 0136042597, ISBN-13: 978-0136042594
2. Artificial Intelligence: Structures and Strategies for Complex Problem Solving by George F Luger, 5th edition, Pearson Education Publications, ISBN-10: 0321545893, ISBN-13: 978-0321545893
3. Artificial Intelligence- Deepak Khemani, Tata McGraw Hill (TMH), ISBN-10: 1259029982, ISBN-13: 978-1259029981

Web Links:

1. <https://nptel.ac.in/courses/106/105/106105079/>
2. <https://www.mindmeister.com/44054594/expert-systems/>
3. https://onlinecourses.nptel.ac.in/noc17_cs30/
4. <https://www.slideshare.net/girishnaik/artificial-intelligence-3638681>

DATA WAREHOUSING AND MINING

Course Code: 2504MC04

L	T	P	C
2	0	1	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Understand different types of data to be mined and different preprocessing techniques.
- CO2:** Analyze OLTP and OLAP.
- CO3:** Analyze Association Rule generation using Apriori and FP growth algorithms.
- CO4:** Build Classification Model using learning algorithms and evaluate performance of it.
- CO5:** Apply Clustering and Outlier detection techniques for mining and analyzing complex data in real-world applications.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	2	3	-	3	2	-	-	-
CO2	3	2	2	2	3	-	-	-
CO3	2	3	3	2	3	-	2	-
CO4	2	2	2	3	3	-	-	-
CO5	3	3	-	2	2	-	2	-

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT – I

Introduction to Data Mining and Preprocessing: An overview of Data Mining – Kinds of data and pattern to be mined –Technologies – Targeted Applications - Major Issues in Data Mining – Data Objects and Attribute Types – Measuring Data Similarity and Dissimilarity.

Data Preprocessing: Data Cleaning – Data Integration – Data Reduction – Data Transformation – Data Discretization.

Practice:

1. Implement data cleaning techniques on a given dataset
2. Apply data transformation techniques to a dataset.

UNIT – II

Data Warehouse and OLAP Technology an Overview: Data Warehouse, A Multidimensional Data Model, Data Warehouse Architecture, Data Warehouse Implementation, From Data Warehousing to Data Mining.

Practice:

1. Build Data Warehouse/Data Mart (using open source tools like Pentaho Data Integration Tool, Pentaho Business Analytics).
2. Design multi-dimensional data models namely Star, Snowflake and Fact Constellation schemas for any one enterprise (ex. Banking, Insurance, Finance, Healthcare, manufacturing, Automobiles, sales etc).
3. Write ETL scripts and implement using data warehouse tools.
4. Perform Various OLAP operations such slice, dice, roll up, drill up and pivot.

UNIT – III

Mining Frequent Patterns and Advanced Pattern Mining: Basic Concepts – Frequent Itemset Mining Methods – Apriori Algorithm-Generating Association Rules from Frequent Itemsets Pattern Evaluation Methods – Pattern Mining in Multilevel, Multidimensional space – Constraint-Based Frequent Pattern Mining – Mining Compressed or Approximate Patterns – Pattern Exploration and Application.

Practice:

1. Implementing the Apriori Algorithm.
2. Generate association rules from frequent itemsets.

UNIT – IV

Classification Techniques: Classification – Model Evaluation and Selection – Techniques to Improve Classification Accuracy – Classification by Backpropagation – Support Vector Machines – Learning from Neighbors.

Practice:

1. Implement a decision tree classifier on a sample dataset.
2. Implement SVM for classification.

UNIT – V

Clustering Techniques: Cluster Analysis – Definition – Types of Data in Cluster Analysis, Clustering methods– Partitioning Methods – k-Means– k-Medoids– Hierarchical Methods – Agglomerative versus Divisive Hierarchical Clustering –BIRCH–Density-Based Methods– DBSCAN. Outlier Detection and Applications: Outliers and Outlier Analysis – Clustering-Based Approach – Classification-Based Approach – Mining Complex Data Types – Data Mining Applications.

Practice:

1. Implement k-Means clustering on a dataset.
2. Identify outliers using a clustering approach (e.g., DBSCAN).
3. Implement outlier detection using the Z-score method.

Textbooks:

1. Data Mining Concept and Techniques, Jiawei Han, Micheline Kamber, Jian Pie, Third Edition, Morgan and Kaufmann Publisher, ISBN-13: 9780123814807.
2. Data Mining Techniques, Arun K Pujari, Second Edition, Universities Press India Pvt. Ltd. ISBN-10: 8173718123.

Reference Books:

1. Introduction to Data Mining - Pang-Ning Tan, Michael Steinbach, and Vipin Kumar, 2nd Edition, ISBN: 978-0138143810.
2. Data Mining: Practical Machine Learning Tools and Techniques, Ian H. Witten, Eibe Frank, Mark A. Hall, Third Edition, Morgan and Kaufmann Publisher, ISBN-10: 0124046512.

Web Links:

1. <https://data-flair.training/blogs/data-mining-tutorial/>.
2. <https://www.geeksforgeeks.org/difference-between-data-warehousing-and-data-mining>.
3. <https://www.javatpoint.com/data-mining-cluster-vs-data-warehousing>.

ADVANCED JAVA PROGRAMMING

Course Code: 2504MC05

L	T	P	C
2	0	2	4

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Utilize fundamental OOP concepts and Java basics, including data types, control statements, and class structures.
- CO2:** Explore inheritance, interfaces, and package management in Java.
- CO3:** Implement effective exception handling and multithreading techniques to create robust and efficient Java applications.
- CO4:** Gain proficiency in JDBC to connect and perform CRUD operations on databases
- CO5:** Develop dynamic web applications using Java Servlets and JSP by understanding their life cycles and core components.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	-	-	3	-	-	-	-
CO2	3	-	-	3	-	-	-	-
CO3	3	2	-	-	-	-	-	-
CO4	3	-	-	3	-	-	-	-
CO5	3	2	-	3	-	-	-	-

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT – I

Introduction to Oops and Classes: Java Features, Programming Paradigm, Basics of Java, Data Types, Variables, Operators, Control Statements, Loops.

Classes and Objects: Introduction of Class and Objects - Introducing Methods - Overloading methods – Constructors - this Keyword.

Class Features: Garbage Collection - the finalize () Method - Understanding static - Introducing nested and inner classes - String class - String Buffer Class - Command Line Arguments.

Practice:

1. Write a Java Program to generate Fibonacci sequence for n numbers
2. Write a Java Program that prompts the user for an integer and then prints out all the prime numbers up to that Integer
3. Write a Java Program to demonstrate String and String Buffer class methods.
4. Implement the concept of constructor overloading

UNIT – II

Inheritance: Inheritance Basics - Using super - Method overriding - Dynamic Method Dispatch- Abstract keyword- Using final with inheritance - The Object Class.

Interfaces and Packages: Inheritance in java with Interfaces – Defining Interfaces - Implementing Interfaces - Extending Interfaces- Creating Packages - Introducing Access Control- Access protection - Importing Packages - Interfaces in a Package.

Practice:

1. Implement the concept of inheritance, super, abstract and final keywords.
2. Write a java program for dynamic dispatch of methods.
3. Write a Java Program to create and demonstrate packages.

UNIT – III

Exception Handling: Exception Handling Fundamentals, Exception Types, Uncaught Exceptions, Using Try and Catch, Multiple Catch clauses, Nested try Statements, Java's Built-in Exceptions, Creating your own Exception subclasses,

Multithreading Java: Thread Model - Life cycle of a Thread - Java Thread Priorities - Runnable interface and Thread Class- Thread Synchronization – Inter Thread Communication.

Practice:

1. Write a Java program to illustrate exception handling mechanism using multiple catch clauses.
2. Implement the concept of multi-threading.

UNIT – IV

JDBC: Introduction to JDBC- Connecting to the database- Basic JDBC Operations – Essential JDBC Classes – JDBC Drivers- Connecting to a database with driver manager – JDBC database URL, CRUD Operations.

Practice:

1. Write a JDBC program to perform CRUD operations

UNIT – V

Java Servlets: Servlets Basics – Life Cycle of a Servlet –A Simple Servlet - The Servlet API – Servlet Interfaces – Generic Servlet Class- HttpServletRequest Interface – HttpServletResponse

JSP: component of jsp page – Page directive – Action – scriptlet – JSP expression, JSP Syntax and semantics.

Practice:

1. Write a program to receive two numbers from a HTML form and display their sum in HTTPServlet.
2. Write a program to Authenticate using JSP.
3. Write a JSP program calculates factorial values for an integer number, while the input is taken from an HTML form.

Additional Practice:

1. Write a Java Program, using String Tokenizer class, which reads a line of integers and then displays each integer and the sum of all integers.
2. Write a program to Authenticate using Java Servlet
3. Write a program to store the user information into cookies, and to display the above stored information by retrieving from cookies.
4. Write a Java program to solve Producer-Consumer problem using synchronization

Textbooks:

1. The Complete Reference Java, Schildt Herbert, 9th Edition, Tata McGraw-Hill, 2019, ISBN-101259003462
2. The Complete Reference JSP 2.0, Phil Anna, 2nd Edition, Tata McGraw-Hill, ISBN-10: 0070580010

Reference Books:

1. Introduction to java programming, by Y Daniel Liang, 7th Edition, Pearson, ISBN-10:1259003462
2. Core Java Volume I Fundamentals, Cay S. Horstmann, 11th Edition, Pearson, ISBN-10: 0137081896
3. "Programming with Java," E Balagurusamy, 6th Edition, McGraw-Hill, ISBN-10:1260571342

Web Links:

1. <https://nptel.ac.in/courses/106/105/106105191/>
2. <https://docs.oracle.com/javase/tutorial/>
3. <https://www.programiz.com/java-programming>
4. <https://cse.iitkgp.ac.in/~dsamanta/java/index.htm>
5. https://onlinecourses.swayam2.ac.in/aic20_sp13/preview

ADVANCED JAVA PROGRAMMING

Course Code: 2604MC02

L	T	P	C
2	0	2	4

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Utilize fundamental OOP concepts and Java basics, including data types, control statements, and class structures.
- CO2:** Explore inheritance, interfaces, and package management in Java.
- CO3:** Implement effective exception handling and multithreading techniques to create robust and efficient Java applications.
- CO4:** Gain proficiency in JDBC to connect and perform CRUD operations on databases
- CO5:** Develop dynamic web applications using Java Servlets and JSP by understanding their life cycles and core components.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	-	-	3	-	-	-	-
CO2	3	-	-	3	-	-	-	-
CO3	3	2	-	-	-	-	-	-
CO4	3	-	-	3	-	-	-	-
CO5	3	2	-	3	-	-	-	-

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT – I

Introduction to Oops and Classes: Java Features, Programming Paradigm, Basics of Java, Data Types, Variables, Operators, Control Statements, Loops.

Classes and Objects: Introduction of Class and Objects - Introducing Methods - Overloading methods – Constructors - this Keyword.

Class Features: Garbage Collection - the finalize () Method - Understanding static - Introducing nested and inner classes - String class - String Buffer Class - Command Line Arguments.

Practice:

1. Write a Java Program to generate Fibonacci sequence for n numbers
2. Write a Java Program that prompts the user for an integer and then prints out all the prime numbers up to that Integer
3. Write a Java Program to demonstrate String and String Buffer class methods.
4. Implement the concept of constructor overloading

UNIT – II

Inheritance: Inheritance Basics - Using super - Method overriding - Dynamic Method Dispatch- Abstract keyword- Using final with inheritance - The Object Class.

Interfaces and Packages: Inheritance in java with Interfaces – Defining Interfaces - Implementing Interfaces - Extending Interfaces- Creating Packages - Introducing Access Control- Access protection - Importing Packages - Interfaces in a Package.

Practice:

1. Implement the concept of inheritance, super, abstract and final keywords.
2. Write a java program for dynamic dispatch of methods.
3. Write a Java Program to create and demonstrate packages.

UNIT – III

Exception Handling: Exception Handling Fundamentals, Exception Types, Uncaught Exceptions, Using Try and Catch, Multiple Catch clauses, Nested try Statements, Java's Built-in Exceptions, Creating your own Exception subclasses,

Multithreading Java: Thread Model - Life cycle of a Thread - Java Thread Priorities - Runnable interface and Thread Class- Thread Synchronization – Inter Thread Communication.

Practice:

1. Write a Java program to illustrate exception handling mechanism using multiple catch clauses.
2. Implement the concept of multi-threading.

UNIT – IV

JDBC: Introduction to JDBC- Connecting to the database- Basic JDBC Operations – Essential JDBC Classes – JDBC Drivers- Connecting to a database with driver manager – JDBC database URL, CRUD Operations.

Practice:

1. Write a JDBC program to perform CRUD operations

UNIT – V

Java Servlets: Servlets Basics – Life Cycle of a Servlet –A Simple Servlet - The Servlet API – Servlet Interfaces – Generic Servlet Class- HttpServletRequest Interface – HttpServletResponse

JSP: component of jsp page – Page directive – Action – scriptlet – JSP expression, JSP Syntax and semantics.

Practice:

1. Write a program to receive two numbers from a HTML form and display their sum in HTTPServlet.
2. Write a program to Authenticate using JSP.
3. Write a JSP program calculates factorial values for an integer number, while the input is taken from an HTML form.

Additional Practice:

1. Write a Java Program, using String Tokenizer class, which reads a line of integers and then displays each integer and the sum of all integers.
2. Write a program to Authenticate using Java Servlet
3. Write a program to store the user information into cookies, and to display the above stored information by retrieving from cookies.
4. Write a Java program to solve Producer-Consumer problem using synchronization

Cornerstone Project:**Project – 1: Student Management System****Implementation Mainly Focuses On**

- Java Fundamentals and OOP Concepts
- Classes, Objects, Constructors, and Methods
- Inheritance and Interfaces
- Exception Handling
- Multithreading
- JDBC Connectivity
- Servlets and JSP for Student Registration and Login

Documentation Must Contain

- Problem Statement
- System Architecture
- Use Case Diagram
- Database Design
- Screenshots
- Source Code
- Test Cases
- Future Enhancements

Project – 2: Library Management System**Implementation Mainly Focuses On**

- Object-Oriented Programming
- Constructor Overloading and Method Overriding
- Packages and Interfaces
- Exception Handling
- Multithreading for Book Issue/Return Processing
- JDBC CRUD Operations
- Servlet and JSP-based User Interface

Documentation Must Contain

- Objectives
- ER Diagram
- Database Schema
- Module Description
- Algorithms
- Screenshots
- Testing
- Conclusion

Project – 3: Employee Payroll Management System**Implementation Mainly Focuses On**

- Classes and Objects
- Inheritance and Runtime Polymorphism
- Abstract Classes and Interfaces
- Exception Handling
- Multithreading
- JDBC Database Operations
- Servlet-Based Employee Portal
- JSP Authentication

Documentation Must Contain

- Requirement Analysis
- UML Class Diagram
- Database Design
- Implementation Details
- Sample Outputs
- Testing Results
- Future Scope

Project – 4: Online Banking System**Implementation Mainly Focuses On**

- Java OOP Concepts
- Inheritance and Interfaces
- User-Defined Exceptions
- Thread Synchronization
- JDBC for Account Management
- Servlets and JSP for Internet Banking
- Session Management

Documentation Must Contain

- Introduction
- System Design
- Activity Diagram
- Database Schema
- Screenshots
- Test Cases
- Conclusion

Project – 5: Hospital Management System**Implementation Mainly Focuses On**

- Java Classes and Objects
- Constructor Overloading
- Inheritance and Packages
- Exception Handling
- Multithreading
- JDBC CRUD Operations
- Servlet and JSP-Based Appointment System

Documentation Must Contain

- Problem Definition
- Use Case Diagram
- Database Tables
- Module Description
- Program Screenshots
- Testing
- Future Enhancements

Project – 6: Online Examination Management System**Implementation Mainly Focuses On**

- Java Programming Fundamentals
- OOP Concepts
- Interfaces and Packages
- Exception Handling
- Multithreading for Timer-Based Exams
- JDBC Database Connectivity
- Java Servlets
- JSP Authentication and Result Generation

Documentation Must Contain

- Objectives
- System Architecture
- UML Diagrams
- Database Design
- Module Description
- Screenshots
- Testing Results
- Conclusion

Textbooks:

1. The Complete Reference Java, Schildt Herbert, 9th Edition, Tata McGraw-Hill, 2019, ISBN-101259003462
2. The Complete Reference JSP 2.0, Phil Anna, 2nd Edition, Tata McGraw-Hill, ISBN-10: 0070580010

Reference Books:

1. Introduction to java programming, by Y Daniel Liang, 7th Edition, Pearson, ISBN-10:1259003462
2. Core Java Volume I Fundamentals, Cay S. Horstmann, 11th Edition, Pearson, ISBN-10: 0137081896
3. “Programming with Java,” E Balagurusamy, 6th Edition, McGraw-Hill, ISBN-10:1260571342

Web Links:

1. <https://nptel.ac.in/courses/106/105/106105191/>
2. <https://docs.oracle.com/javase/tutorial/>
3. <https://www.programiz.com/java-programming>
4. <https://cse.iitkgp.ac.in/~dsamanta/java/index.htm>
5. https://onlinecourses.swayam2.ac.in/aic20_sp13/preview

Note:

- The cornerstone project team size shall be four (4) students.
- Students may select any one of the above prescribed projects or a project of their own choice with the prior permission of the Course instructor.
- The evaluation of the Cornerstone Project shall be based on the project implementation, oral presentation, 10-minute video presentation, report and viva voce.
- The video presentation should consist of the working procedure of the project along with contribution of each student for a minimum of 2 minutes.
- Finally, the Source code of the cornerstone Project has to be pushed into the Students GitHub repository (if applicable).

PYTHON PROGRAMMING

Course Code: 2504MC06

L	T	P	C
1	0	2	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Apply Python operators and expressions to develop conditional and iterative logic structures
- CO2:** Effectively utilize Python's data structures to perform operations and manage data efficiently
- CO3:** Analyze and manipulate strings in Python using built-in methods.
- CO4:** Develop Python programs using user-defined functions and nltk for advanced data processing and analysis.
- CO5:** To define a class with attributes and methods in python

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	2	1	3	1	-	-	1
CO2	3	3	2	1	-	-	1	-
CO3	2	1	2	3	2	1	-	1
CO4	3	3	1	-	1	2	1	-
CO5	2	1	3	3	1	-	-	2

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT – I

Introduction to Python: History of Python, Unique features of Python, Demo on IDE, Ipython, Spyder etc., “Hello world” program in Python, Keywords, Identifiers, reading input from user-Demo, Python Data Types, Declaring and using Numeric data types: int, float, complex and string.

Python Operators, Expressions and Flow Controls: All Operations and simple expressions, Conditional blocks using if, else and elif, Simple for loops in python, for loop using ranges, Use of while and do while-loop in python, Loop manipulation using pass, continue, break and else.

Practice:

- Write a Python program that takes input from the user and determines if the number is positive, negative, or zero using conditional statements (if, else, elif).
- Develop a Python script that calculates the factorial of a number using both for and while loops, demonstrating the use of loop control statements (pass, continue, break, and else).

UNIT – II

Pythons List, Tuples, Dictionaries & Sets: Lists and its operations, Ranges: Iterators and its purpose, Tuples: Operation and usage, Python Dictionaries, examples on Dictionaries, Sets and its operations, File handling.

Practice:

1. Create a Python program that performs various operations on lists (such as adding, removing, sorting, and reversing) and tuples (accessing elements, slicing, and unpacking).
2. Develop a Python script that creates a dictionary to store and retrieve contact details (name and phone number) and demonstrates adding, updating, and deleting entries.

UNIT-III

Python Strings & Regular Expressions: Strings: Understanding string in build methods and Operations[slicing],

Regular Expressions: Powerful pattern matching and searching, Power of pattern searching using regex in python, Real time parsing of networking or system data using regex, Password, email, URL validation using regular expression, Pattern finding programs using regular expressions.

Practice:

1. Write a Python program to demonstrate different string operations such as concatenation, slicing, and finding the length of a string.
2. Create a Python script that uses regular expressions to validate user input for an email address and a password, ensuring they meet specific criteria (e.g., email format and password strength).

UNIT – IV

Python Functions, Exceptions and Packages: Python user defined functions, Python packages functions, Defining and calling Function, powerful Lamda function in python, organizing python codes using functions, Programming using Exception handling, pandas, NumPy, Scikit, nltk etc.

Practice:

1. Develop a Python function that takes a list of numbers as input and returns the sum of the squares of the numbers. Include error handling to manage incorrect inputs.
2. Create a Python script that uses Pandas to read a CSV file and perform basic data analysis, such as finding the mean, median, and mode of a dataset.

UNIT – V

Object Oriented programming in python: Creating a Class, Class methods, Class Inheritance, Encapsulation, Polymorphism, class method vs. static methods, Python object persistence.

Practice:

1. Develop a python program for Bank Class to manage the Customer Accounts and

Transactions

2. Develop a python program for Student management system using inheritance

Additional Practice:

1. Design a Python project that integrates lists, dictionaries, and file handling to create a simple contact management system that can add, view, update, and delete contacts, with data stored in a file.
2. Write a Python program that uses functions, loops, and regular expressions to search through a text file for lines matching a specific pattern, such as IP addresses or URLs, and output the results.
3. Develop a Python script that uses NumPy to perform matrix operations

Case Studies:

1. Student Grade Calculator
2. Simple ATM Simulation
3. Library Book Management System
4. Online Quiz Application
5. To-Do List Manager
6. Basic Calculator
7. Employee Salary Slip Generator
8. Simple Chatbot
9. Automated Email Sender
10. Tic Tac Toe Game with Minimax AI

Textbooks:

1. Core Python Programming, Wesley J. Chun, Second Edition, Pearson. ISBN-13, 978-0132269933.
2. Think Python How to Think Like a Computer Scientist, Allen B. Downey 2nd Edition, Version 2.4.0. ISBN-13 978-1491939369.

Reference Books:

1. An Introduction to Python Tutorial Guido van Rossum and the Python development team, version 3.2.1, Python Software Foundation. ISBN-13: 978-9355425218
2. A Practical Introduction to Python Programming, Brian Heinold, Generic Publication ISBN-13: 9798848271577.
3. Python For Everybody: Python Programming Made Easy Kindle Edition, ISBN 13: 9781329427426.

Web Links:

1. <https://www.w3schools.com/python/default.asp>
2. <http://www.cse.iitm.ac.in/~rupesh/books/python>

PYTHON PROGRAMMING

Course Code: 2604MC03

L	T	P	C
2	0	2	4

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Apply Python operators and expressions to develop conditional and iterative logic structures
- CO2:** Effectively utilize Python's data structures to perform operations and manage data efficiently
- CO3:** Analyze and manipulate strings in Python using built-in methods.
- CO4:** Develop Python programs using user-defined functions and nltk for advanced data processing and analysis.
- CO5:** To define a class with attributes and methods in python

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	2	1	3	1	-	-	1
CO2	3	3	2	1	-	-	1	-
CO3	2	1	2	3	2	1	-	1
CO4	3	3	1	-	1	2	1	-
CO5	2	1	3	3	1	-	-	2

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT – I

Introduction to Python: History of Python, Unique features of Python, Demo on IDE, Ipython, Spyder etc., “Hello world” program in Python, Keywords, Identifiers, reading input from user-Demo, Python Data Types, Declaring and using Numeric data types: int, float, complex and string.

Python Operators, Expressions and Flow Controls: All Operations and simple expressions, Conditional blocks using if, else and elif, Simple for loops in python, for loop using ranges, Use of while and do while-loop in python, Loop manipulation using pass, continue, break and else.

Practice:

- Write a Python program that takes input from the user and determines if the number is positive, negative, or zero using conditional statements (if, else, elif).
- Develop a Python script that calculates the factorial of a number using both for and while loops, demonstrating the use of loop control statements (pass, continue, break, and else).

UNIT – II

Pythons List, Tuples, Dictionaries & Sets: Lists and its operations, Ranges: Iterators and its purpose, Tuples: Operation and usage, Python Dictionaries, examples on Dictionaries, Sets and its operations, File handling.

Practice:

1. Create a Python program that performs various operations on lists (such as adding, removing, sorting, and reversing) and tuples (accessing elements, slicing, and unpacking).
2. Develop a Python script that creates a dictionary to store and retrieve contact details (name and phone number) and demonstrates adding, updating, and deleting entries.

UNIT-III

Python Strings & Regular Expressions: Strings: Understanding string in build methods and Operations[slicing],

Regular Expressions: Powerful pattern matching and searching, Power of pattern searching using regex in python, Real time parsing of networking or system data using regex, Password, email, URL validation using regular expression, Pattern finding programs using regular expressions.

Practice:

1. Write a Python program to demonstrate different string operations such as concatenation, slicing, and finding the length of a string.
2. Create a Python script that uses regular expressions to validate user input for an email address and a password, ensuring they meet specific criteria (e.g., email format and password strength).

UNIT – IV

Python Functions, Exceptions and Packages: Python user defined functions, Python packages functions, Defining and calling Function, powerful Lamda function in python, organizing python codes using functions, Programming using Exception handling, pandas, NumPy, Scikit, nltk etc.

Practice:

1. Develop a Python function that takes a list of numbers as input and returns the sum of the squares of the numbers. Include error handling to manage incorrect inputs.
2. Create a Python script that uses Pandas to read a CSV file and perform basic data analysis, such as finding the mean, median, and mode of a dataset.

UNIT – V

Object Oriented programming in python: Creating a Class, Class methods, Class Inheritance, Encapsulation, Polymorphism, class method vs. static methods, Python object persistence.

Practice:

1. Develop a python program for Bank Class to manage the Customer Accounts and Transactions
2. Develop a python program for Student management system using inheritance

Additional Practice:

1. Design a Python project that integrates lists, dictionaries, and file handling to create a simple contact management system that can add, view, update, and delete contacts, with data stored in a file.
2. Write a Python program that uses functions, loops, and regular expressions to search through a text file for lines matching a specific pattern, such as IP addresses or URLs, and output the results.

Cornerstone Project:**Project – 1: Student Academic Performance Analyzer****Implementation Mainly Focuses On**

- Student details entry and management.
- Subject-wise marks entry and validation.
- Calculation of total marks, percentage, and grade.
- Use of lists and dictionaries for storing student records.
- File handling for saving and retrieving student data.
- Exception handling for invalid inputs.

Documentation Must Contain

- Objective & problem definition
- Student record structure
- Flowchart for result generation
- Module explanations: Add Student, Enter Marks, Generate Result, View Report
- Grade calculation logic
- Screenshots of sample input/output
- Enhancements added (rank generation, subject-wise analysis)

Project – 2: Library Book Management System**Implementation Mainly Focuses On**

- Adding, searching, updating, and deleting book records.
- Managing book details using dictionaries and lists.
- File handling for maintaining book inventory.
- Search operations based on book ID and title.
- Validation for duplicate book IDs.
- Exception handling for invalid inputs and missing records.

Documentation Must Contain

- Objective of library management
- Book record structure
- Flowchart for Add → Search → Update operations
- Module descriptions
- File handling mechanism
- Screenshots
- Enhancements (book category management, issue status tracking)

Project – 3: Contact Management System**Implementation Mainly Focuses On**

- Adding, viewing, updating, and deleting contact details.
- Storing contact information using dictionaries.
- Phone number and email validation using regular expressions.
- File handling for persistent storage of contacts.
- Searching contacts by name or phone number.
- Exception handling for duplicate and invalid entries.

Documentation Must Contain

- Objective
- Contact record structure
- Validation rules using regular expressions
- Module explanation: Add, Search, Update, Delete
- File storage mechanism
- Screenshots of sample execution
- Enhancements (contact grouping, search filters)

Project – 4: Employee Payroll Analyzer**Implementation Mainly Focuses On**

- Employee details management.
- Salary calculation based on basic pay, allowances, and deductions.
- Generation of employee salary reports.
- Use of functions for payroll processing.
- Data analysis using Pandas for salary statistics.
- Exception handling for invalid salary inputs.

Documentation Must Contain

- Objective
- Employee record structure
- Salary calculation logic

- Flowchart for payroll processing
- Module descriptions
- Sample salary report screenshots
- Enhancements (department-wise salary analysis)

Project – 5: Online Quiz and Result Evaluator

Implementation Mainly Focuses On

- Creating and managing multiple-choice questions.
- Conducting quizzes using menu-driven programming.
- Automatic score calculation and result generation.
- Use of lists, dictionaries, and functions.
- Performance analysis of quiz results.
- File handling for storing questions and scores.

Documentation Must Contain

- Objective
- Question bank structure
- Flowchart for quiz execution
- Module explanations
- Score calculation logic
- Screenshots of quiz execution
- Enhancements (random question selection, result summary)

Project – 6: Text Pattern Analyzer using Regular Expressions

Implementation Mainly Focuses On

- Reading text data from files.
- Identifying patterns using regular expressions.
- Extraction and validation of email addresses, URLs, and phone numbers.
- String manipulation and text processing.
- Generating summary statistics of extracted patterns.
- Exception handling for file operations.

Documentation Must Contain

- Objective
- Pattern matching requirements
- Regular expressions used
- Flowchart for text processing
- Module explanations
- Sample input/output screenshots
- Enhancements (custom pattern search, report generation)

Textbooks:

1. Core Python Programming, Wesley J. Chun, Second Edition, Pearson. ISBN-13, 978-0132269933.
2. Think Python How to Think Like a Computer Scientist, Allen B. Downey 2nd Edition, Version 2.4.0. ISBN-13 978-1491939369.

Reference Books:

1. An Introduction to Python Tutorial Guido van Rossum and the Python development team, version 3.2.1, Python Software Foundation. ISBN-13: 978-9355425218
2. A Practical Introduction to Python Programming, Brian Heinold, Generic Publication ISBN-13: 9798848271577.
3. Python For Everybody: Python Programming Made Easy Kindle Edition, ISBN 13: 9781329427426.

Web Links:

1. <https://www.w3schools.com/python/default.asp>
2. <http://www.cse.iitm.ac.in/~rupesh/books/python>

Note:

- The cornerstone project team size shall be four (4) students.
- Students may select any one of the above prescribed projects or a project of their own choice with the prior permission of the Course instructor.
- The evaluation of the Cornerstone Project shall be based on the project implementation, oral presentation, 10-minute video presentation, report and viva voce.
- The video presentation should consist of the working procedure of the project along with contribution of each student for a minimum of 2 minutes.
- Finally, the Source code of the cornerstone Project has to be pushed into the Students GitHub repository (if applicable).

ADVANCED DATA STRUCTURES AND ALGORITHMS

Course Code: 2504MC07

L	T	P	C
2	0	1	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Understand arrays and linked lists and their applications.
- CO2:** Implement stacks and queues for solving basic computational problems.
- CO3:** Work with tree structures and perform various tree operations.
- CO4:** Apply graph algorithms for traversal and optimization tasks.
- CO5:** Use searching and sorting algorithms and analyze their performance.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	2	-	2	1	-	-	1
CO2	3	-	2	2	-	-	-	2
CO3	3	3	2	2	1	-	-	2
CO4	3	2	1	3	1	-	-	-
CO5	3	-	2	3	-	-	-	2

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT – I

Fundamentals Data Structures and Linked Lists: Overview of Arrays, Classification of Data Structures, Storage structure of linear arrays, Storage representation of matrices, Sparse matrices, Singly Linked Lists, Doubly Linked Lists, Circular Linked Lists, Linked List Applications

Practice:

1. Write a program to implement a singly linked list with insertion and deletion operations.
2. Write a program to implement a doubly linked list and display the list in forward and reverse order.

UNIT – II

Stacks, Queues and Circular Queues: Stack ADT, Stack Implementation using Arrays and Linked Lists, Queue ADT, Implementation of Queues, Circular Queues, Deques, Priority Queues, Evaluation of Postfix Expression, Infix to Postfix Conversion, Recursion

Practice:

1. Write a program to implement a stack using arrays and perform push and pop operations.
2. Write a program to convert an infix expression into postfix expression and evaluate it.

UNIT – III

Trees and Balanced Trees: Representation of Binary Trees using Arrays and Linked Lists, Binary Tree Traversals (Inorder, Preorder, Postorder), Binary Search Trees, AVL Trees, 2-3 Trees, Heaps, Heap Sort

Practice:

1. Write a program to construct a binary search tree and perform inorder, preorder, and postorder traversals.
2. Write a program to implement an AVL tree with insertion and rotation operations.

UNIT – IV

Graphs and Optimization Techniques: Graph Representations (Adjacency Matrix, Adjacency List), Graph Traversals: Depth First Search (DFS) and Breadth First Search (BFS), Spanning Tree, Minimal Spanning Tree, Kruskal's Algorithm, Prim's Algorithm, Topological Sorting, Transitive Closure, Dijkstra's Algorithm, All-Pairs Shortest Path (Floyd-Warshall), Binomial Coefficient.

Practice:

1. Write a program to perform BFS and DFS traversal of a graph.
2. Write a program to find the shortest path using Dijkstra's algorithm.

UNIT – V

Searching, Sorting and Algorithm Analysis: Notion of Algorithm, Asymptotic Notations (Big-O, Omega, Theta), Linear Search, Selection Sort, Bubble Sort, Insertion Sort, Factorial, Tower of Hanoi, Fibonacci, Binary Search, Merge Sort, Quick Sort

Practice:

1. Write a program to implement binary search on a sorted array.
2. Write a program to sort a list of elements using quick sort.

Textbooks:

1. "Data Structures", Lipschutz Seymour ,Revised First Edition, Schaum's Outline Series, McGraw Hill Education India, ISBN-10: 0070605188
2. "Introduction to the Design and Analysis of Algorithms", Anany Levitin, Third Edition, Pearson Education. ISBN-10: 0132316817

Reference Books:

1. Data structures-A pseudocode approach with C, Richard F. Gilberg and Behrouz A. Fourouzan, 2nd Edition, 2005,Cengage Learning, ISBN-10: 0619216490
2. An Introduction to Data Structures With Application, Jean-Paul Tremblay, Paul G. Sorenson, 2nd Edition, Mcgraw Hill , ISBN-10: 0072943790
3. "Introduction to Algorithms", Thomas H.Cormen, Charles E.Leiserson, Ronald L. Rivest and Clifford Stein, Third Edition, PHI Learning, ISBN-10: 0262033844

Web Links:

1. <http://freevideolectures.com/Course/2519/C-Programming-and-Data-Structures>
2. <http://www.nptelvideos.in/2012/11/data-structures-and-algorithms.html>
3. https://onlinecourses.nptel.ac.in/noc16_cs06/preview
4. <http://nptel.ac.in/courses/106101060>
5. <https://nptel.ac.in/courses/106/106/106106145>

COMPUTER NETWORKS

Course Code: 2504MC08

L	T	P	C
2	0	1	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Understand protocol layers, network components, and their roles in Internet communication.
- CO2:** Apply principles of network applications like HTTP, FTP, email, and DNS in practical scenarios.
- CO3:** Demonstrate understanding of TCP and UDP functionalities in reliable data transfer.
- CO4:** Analyze IP addressing, routing algorithms (Link State, Distance-Vector), and their impact on network performance.
- CO5:** Implement error detection, multiple access protocols, and understand their role in network connectivity.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	2	1	2	1	-	-	1
CO2	1	3	2	1	2	-	-	2
CO3	3	-	2	2	1	-	-	-
CO4	3	3	2	2	2	-	-	1
CO5	2	1	1	2	2	-	1	2

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT – I

Introduction to Computer Networks, Protocol Layers: Computer Networks and the Internet, What is Internet? The network Edge, The Network Core, Delay, Loss, and Throughput in Packet-Switched Networks, Protocol Layers and their Service Models.

Practice:

1. Write a program to calculate transmission delay, propagation delay, and throughput in a packet-switched network.
2. Write a program to demonstrate the working of protocol layers in a computer network using a layered approach.

UNIT – II

Application Layer: Application Layer: Principles of Network Applications, The Web and HTTP, File Transfer: FTP, Electronic Mail in the Internet, DNS-The Internet's Directory Service: Services provided by DNS, overview of how DNS works.

Practice:

1. Write a program to simulate client-server communication using application layer concepts.
2. Write a program to demonstrate DNS name resolution in the Internet.

UNIT-III

Data Link Layer: Error Detection and Correction – Block Coding, Cyclic Redundancy Check, Checksum, Data Link Control – Framing, Flow and Error Control, Multiple Access – Random Access, ALOHA, CSMA, CSMA/CD, CSMA/CA, Channelization.

Practice:

1. Write a program to implement Cyclic Redundancy Check (CRC) for error detection.
2. Write a program to simulate the CSMA/CD protocol for multiple access communication.

UNIT – IV

The Network Layer: Overview of Network Layer, What's inside a Router? The Internet Protocol (IP), Routing Algorithms: Link State Routing Algorithm, Distance-Vector Routing Algorithm

Practice:

1. Write a program to implement the Distance Vector Routing algorithm.
2. Write a program to implement the Link State Routing algorithm.

UNIT – V

Transport Layer: Relationship Between Transport and Network Layers-Transport Services, Elements of Transport Protocols-Overview of the Transport Layer in the Internet-Multiplexing and Demultiplexing - Connection management, TCP and UDP protocols.

Practice:

1. Write a program to simulate TCP connection establishment and termination.
2. Write a program to demonstrate multiplexing and demultiplexing in the transport layer.

Textbooks:

1. “Computer Networking”: A Top-Down Approach, James F Kurose and Keith W Ross, 8th Edition, Pearson Publication, ISBN-10: 1292405465, ISBN-13: 978-1292405469
2. Data communications and networking: Behrouz Forouzan, Edition 5, McGraw Hill, ISBN- 10: 0073376221, ISBN-13: 978-0073376226

Reference Books:

1. “Computer Networks”, Andrew S. Tanenbaum and David J. Wetherill, 5th edition, Prentice Hall, ISBN-10: 0132126958, ISBN-13: 978-0132126953
2. "Computer Networks": A Systems Approach, Larry L Peterson and Bruce S. Davie, 6th Edition, Morgan Kaufmann, ISBN-10: 0128182006, ISBN-13: 978-0128182000

Web Links:

1. <https://nptel.ac.in/courses/106105081>
2. https://gaia.cs.umass.edu/kurose_ross/lectures.php
3. <https://archive.nptel.ac.in/courses/106/105/106105183/>
4. <https://www.coursera.org/learn/fundamentals-network-communications>

SOFTWARE PROJECT AND TESTING MANAGEMENT

Course Code: 2504MC09

L	T	P	C
2	0	1	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Understand software process models and estimation techniques.
- CO2:** Identify, assess, and plan for project risks.
- CO3:** Design test cases from requirements and design.
- CO4:** Apply black box testing techniques for software validation.
- CO5:** Conduct static and regression testing to ensure software quality.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	3	2	2	-	3	2	2
CO2	2	3	2	2	2	1	-	-
CO3	2	2	-	2	-	-	1	-
CO4	2	2	2	-	3	2	-	-
CO5	2	1	2	3	2	2	1	-

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT – I

Introduction to Project Planning: Introduction to Project Management, Importance of Software Project Management. Categorization of Software Projects, Setting objectives.

Project Life Cycle and Effort Estimation: Software Process Models, Rapid Application development, Agile methods, Extreme Programming, Reliability Growth Models, Load Testing Model, Managing, Basics of Software estimation, Effort and Cost estimation techniques, COSMIC Full function points, COCOMO.

Practice:

1. Using COCOMO model estimate effort.
2. Perform Estimation of effort using FP Estimation for chosen system.

UNIT – II

Activity Planning and Risk Management: Objectives of Activity planning, Project schedules, Activities, Sequencing and scheduling, Network Planning models, Formulating Network Model, Forward Pass & Backward Pass techniques, Critical path (CRM) method, Risk identification, Assessment, Risk Planning, Risk Management, PERT technique, Project Management and Control.

Practice:

1. Analyze the Risk related to the project and prepare RMMM plan.
2. Develop Time-line chart and project table using PERT or CPM project scheduling methods.

UNIT – III

Project Management Tools: JIRA, Asana, Trello, Basecamp, Smartsheet.

Software Testing: Introduction, Software testing principles, The tester's role in a software development organization.

Verification and Validation: Verification & Validation Activities, Verification, Verification of Requirements, verifying code, Validation.

Practice:

1. Design of Test cases based on requirements and design.
2. Prepare FTR.

UNIT – IV

Black Box Testing Techniques: Requirement based testing, Boundary Value Analysis, Equivalence class Testing, State Table based testing, Decision table-based testing, Cause-Effect Graphing based testing, Error guessing

White-Box Testing: Need, Logic Coverage criteria, Basis Path testing, Graph matrices, Loop testing, data flow testing, mutation testing.

Practice:

1. Design test cases based on the decision table.
2. Design test cases to cover different data flow scenarios.

UNIT – V

Levels of Testing: Need for Levels of Testing, unit testing, Integration testing, system testing, Regression testing, Acceptance testing.

Static and Regression Testing: Inspections, Structured Walkthroughs, Technical Reviews, Progressives Vs regressive testing, Regression testing ability, Objectives of regression testing, Regression testing types, Regression testing techniques.

Practice:

1. Conduct a walkthrough meeting with stakeholders and Gather feedback and identify issues.
2. Design and execute regression test cases.

Textbooks:

1. "Software Project Management" – Bob Hughes, Mike Cotterell and Rajib Mall: Fifth Edition, McGraw Hill, New Delhi, ISBN - 9780077122799.
2. "Software Quality Assurance, Testing and Metrics" by Anirban Basu, 1st Edition, PHI Learning, ISBN: 978-8120339988.

Reference Books:

1. "Software Testing: A Craftsman's Approach" by Paul C. Jorgensen, 4th Edition, CRC Press, ISBN - 978-1466560680.
2. Foundations of Software testing, Aditya P Mathur, 2nd ed, Pearson. ISBN - 978-0137035150
3. "Managing Global Software Projects" by Gopalaswamy Ramesh, Fourteenth Reprint edition published by McGraw Hill Education, ISBN - 978-0070613224.
4. Software Testing, Principles and Practices, Naresh Chauhan, 2nd Edition, Oxford, ISBN-9780199465873.

Web Links:

1. <https://www.coursera.org/specializations/product-management>
2. https://onlinecourses.nptel.ac.in/noc18_mg08/preview.
3. <https://www.scribd.com/doc/7102316/Software-Project-Management>
4. https://onlinecourses.nptel.ac.in/noc23_cs38/preview

MACHINE LEARNING WITH PYTHON

Course Code: 2504MC10

L	T	P	C
2	0	2	4

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Understand the basics of machine learning, including data representation, classification challenges, and validation techniques.
- CO2:** Apply linear regression and regularization techniques to build more accurate predictive models.
- CO3:** Use decision trees, Bayesian classification, and k-Nearest Neighbor (k-NN) for pattern recognition and prediction tasks.
- CO4:** Apply ensemble techniques like bagging, boosting, stacking, and SVMs for advanced classification tasks.
- CO5:** Apply clustering and dimensionality reduction techniques, and understand neural network architectures including ANN, CNN, and RNN.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	2	1	2	-	-	-	2
CO2	1	2	2	1	-	-	-	1
CO3	2	-	2	2	-	-	-	1
CO4	2	3	2	3	-	-	-	-
CO5	3	3	3	3	3	-	-	2

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1	3	1
CO2	1	3
CO3	-	3
CO4	3	3
CO5	3	1

UNIT – I

Introduction to Machine Learning: Problems, data, and tools. Learning systems, goals, challenges, and applications of machine learning systems. Aspects of developing a system, training data, testing data, classification errors, Model validation techniques.

Practice:

1. Implement a Python program to load a dataset (e.g., CSV file), preprocess it (e.g., handle missing values), and split it into training and testing sets.
2. Implement a Python program to visualize the dataset using scatter plot, bar plot, histogram, distribution plot, box plot.

UNIT – II

Regression and Regularization: Linear regression, gradient descent, Support Vector Regression, bias and variance estimation, overfitting and underfitting, regularization, ridge and lasso regression.

Practice:

1. Develop a Python program to implement simple linear regression using gradient descent.
2. Develop a Python program to implement simple linear regression using ridge or lasso regression.

UNIT – III

Classification Methods: Basic Concepts: Decision tree induction: Attribute Selection Measures Tree Pruning, Bayes Classification Methods, Logistic Regression, Rule- Based Classification, k-Nearest Neighbors method. Model Evaluation and Selection.

Practice:

1. Develop a Python program to build a decision tree classifier using a dataset.
2. Implement the k-Nearest Neighbor algorithm in Python for classification tasks.
3. Implement logistic regression in Python to model binary classification problems.

UNIT – IV

Ensemble Learning Methods: Introduction. Voting classifiers, Bagging and pasting, boosting, stacking, random forest and Support Vector machines, Types of SVM's Classification- linear SVM and non-linear SVM.

Practice:

1. Implement SVM for a classification task with a non-linear decision boundary.
2. Implement a random forest classifier and compare it with a decision tree.

UNIT – V

Clustering Techniques: Clustering – Introduction – K-Means Clustering, Limitation of DBSCAN – Gaussian Mixtures, Dimensionality reduction, Data compression, PCA.

Neural Networks: Introduction to artificial neural network(ANN), CNN and RNN.

Practice:

1. Implement K-Means clustering in Python to group a dataset into clusters based on similarity.
2. Write a Python program to perform hierarchical clustering on a dataset.
3. Implement PCA to reduce the dimensionality of the dataset and visualize the principal components

Additional Practice:

1. Implement Density-Based Spatial Clustering of Applications with Noise(DBSCAN).
2. Implement k-fold cross-validation to evaluate the performance of a machinelearning model.

3. Develop a program on Boosting optimization techniques on any given Dataset.
4. Implement Support Vector Machine algorithm on any dataset.

Textbooks:

1. Machine Learning, Tom Mitchell, McGraw Hill, ISBN-10: 0070428077, ISBN-13: 978-0070428072
2. “Pattern Classification” - Duda, Richard, Peter Hart, and David Stork, Second Edition, New York, NY: Wiley-Inter science, ISBN-10: 0471056693, ISBN-13 978-0471056690

Reference Books:

1. Data Mining Concepts and Techniques - Jiawei Han, Micheline Kamber, Jian Pie, Third Edition, Morgan and Kaufmann Publisher, ISBN-10: 9780123814791, ISBN-13: 978-9380931913
2. The Elements of Statistical Learning: Data Mining, Inference and Prediction- T. Hastie, R. Tibshirani and J. Friedman, 2nd Edition, Springer publication, ISBN-10: 0387848576, ISBN-13: 978-0387848570

Web Links:

1. <https://data-flair.training/blogs/data-mining-tutorial/>
2. <https://machinelearningmastery.com/>
3. <https://towardsdatascience.com/>
4. <https://scikit-learn.org/stable/>
5. <https://nptel.ac.in/courses/106106139>

MACHINE LEARNING WITH PYTHON

Course Code: 2604MC04

L	T	P	C
2	0	2	4

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Understand the basics of machine learning, including data representation, classification challenges, and validation techniques.
- CO2:** Apply linear regression and regularization techniques to build more accurate predictive models.
- CO3:** Use decision trees, Bayesian classification, and k-Nearest Neighbor (k-NN) for pattern recognition and prediction tasks.
- CO4:** Apply ensemble techniques like bagging, boosting, stacking, and SVMs for advanced classification tasks.
- CO5:** Apply clustering and dimensionality reduction techniques, and understand neural network architectures including ANN, CNN, and RNN.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	2	1	2	-	-	-	2
CO2	1	2	2	1	-	-	-	1
CO3	2	-	2	2	-	-	-	1
CO4	2	3	2	3	-	-	-	-
CO5	3	3	3	3	3	-	-	2

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1	3	1
CO2	1	3
CO3	-	3
CO4	3	3
CO5	3	1

UNIT – I

Introduction to Machine Learning: Problems, data, and tools. Learning systems, goals, challenges, and applications of machine learning systems. Aspects of developing a system, training data, testing data, classification errors, Model validation techniques.

Practice:

1. Implement a Python program to load a dataset (e.g., CSV file), preprocess it (e.g., handle missing values), and split it into training and testing sets.
2. Implement a Python program to visualize the dataset using scatter plot, bar plot, histogram, distribution plot, box plot.

UNIT – II

Regression and Regularization: Linear regression, gradient descent, Support Vector Regression, bias and variance estimation, overfitting and underfitting, regularization, ridge and lasso regression.

Practice:

1. Develop a Python program to implement simple linear regression using gradient descent.
2. Develop a Python program to implement simple linear regression using ridge or lasso regression.

UNIT – III

Classification Methods: Basic Concepts: Decision tree induction: Attribute Selection Measures Tree Pruning, Bayes Classification Methods, Logistic Regression, Rule- Based Classification, k-Nearest Neighbors method. Model Evaluation and Selection.

Practice:

1. Develop a Python program to build a decision tree classifier using a dataset.
2. Implement the k-Nearest Neighbor algorithm in Python for classification tasks.
3. Implement logistic regression in Python to model binary classification problems.

UNIT – IV

Ensemble Learning Methods: Introduction. Voting classifiers, Bagging and pasting, boosting, stacking, random forest and Support Vector machines, Types of SVM's Classification- linear SVM and non-linear SVM.

Practice:

1. Implement SVM for a classification task with a non-linear decision boundary.
2. Implement a random forest classifier and compare it with a decision tree.

UNIT – V

Clustering Techniques: Clustering – Introduction – K-Means Clustering, Limitation of DBSCAN – Gaussian Mixtures, Dimensionality reduction, Data compression, PCA.

Neural Networks: Introduction to artificial neural network(ANN), CNN and RNN.

Practice:

1. Implement K-Means clustering in Python to group a dataset into clusters based on similarity.
2. Write a Python program to perform hierarchical clustering on a dataset.
3. Implement PCA to reduce the dimensionality of the dataset and visualize the principal components

Additional Practice:

1. Implement Density-Based Spatial Clustering of Applications with Noise(DBSCAN).
2. Implement k-fold cross-validation to evaluate the performance of a machinelearning model.
3. Develop a program on Boosting optimization techniques on any given Dataset.

Cornerstone Project:

Project – 1: Student Result Analysis System

Implementation Mainly Focuses On

- Student result data preprocessing.
- Performance analysis using Machine Learning.
- Grade classification and visualization.
- Identification of performance trends.
- Model evaluation.
- Report generation.

Documentation Must Contain

- Objective
- Dataset Description
- Data Preprocessing
- Flowchart
- Model Development
- Performance Evaluation
- Sample Outputs and Screenshots
- Conclusion and Future Enhancements

Project – 2: Disease Prediction System

Implementation Mainly Focuses On

- **Disease** dataset collection and preprocessing.
- Feature selection and data analysis.
- **Disease** prediction using Logistic Regression, Decision Tree, Random Forest, and Support Vector Machine (SVM).
- Model training, testing, and performance comparison.
- Prediction of **Disease** risk based on patient health parameters.
- Result visualization and analysis.

Documentation Must Contain

- Objective
- Dataset Description
- Data Preprocessing
- Flowchart
- Model Development
- Performance Evaluation and Algorithm Comparison
- Sample Outputs and Screenshots
- Conclusion and Future Enhancements

Project – 3: Email Spam Detection System

Implementation Mainly Focuses On

- Text preprocessing.
- Feature extraction using TF-IDF.
- Spam classification using Naïve Bayes and Support Vector Machine.
- Model evaluation using Precision, Recall and F1-Score.
- Comparison of classifiers.

Documentation Must Contain

- Objective
- Dataset Description
- Text Preprocessing
- Flowchart
- Model Building
- Evaluation Metrics
- Sample Outputs
- Future Scope

Project – 4: Product Recommendation System

Implementation Mainly Focuses On

- Product and customer dataset collection and preprocessing.
- Feature selection and data analysis.
- Product recommendation using K-Means Clustering, k-Nearest Neighbors (k-NN), and Association Rule Mining.
- Model training, testing, and performance evaluation.
- Personalized product recommendation based on customer preferences.
- Result visualization and analysis.

Documentation Must Contain

- Objective
- Dataset Description
- Data Preprocessing
- Flowchart
- Model Development
- Performance Evaluation and Algorithm Comparison
- Sample Outputs and Screenshots
- Conclusion and Future Enhancements

Project – 5: Image Classification using Convolutional Neural Networks (CNN)

Implementation Mainly Focuses On

- Image dataset preprocessing.
- Image augmentation.
- CNN model development.
- Model training and testing.
- Accuracy and loss visualization.
- Prediction on unseen images.

Documentation Must Contain

- Objective
- Dataset Description
- CNN Architecture
- Flowchart
- Training Process
- Performance Analysis
- Prediction Screenshots
- Future Enhancements

Project – 6: Online Shopping Purchase Prediction

Implementation Mainly Focuses On

- Customer data collection and preprocessing.
- Feature selection and data analysis.
- Purchase prediction using classification algorithms.
- Model training and performance evaluation.
- Prediction of customer purchase behaviour.
- Result visualization and comparison.

Documentation Must Contain

- Objective
- Dataset Description
- Data Preprocessing
- Flowchart
- Model Development
- Performance Evaluation
- Sample Outputs and Screenshots
- Conclusion and Future Enhancements

Textbooks:

1. Machine Learning, Tom Mitchell, McGraw Hill, ISBN-10: 0070428077, ISBN-13: 978-0070428072
2. “Pattern Classification” - Duda, Richard, Peter Hart, and David Stork, Second Edition, New York, NY: Wiley-Inter science, ISBN-10: 0471056693, ISBN-13 978-0471056690

Reference Books:

1. Data Mining Concepts and Techniques - Jiawei Han, Micheline Kamber, Jian Pie, Third Edition, Morgan and Kaufmann Publisher, ISBN-10: 9780123814791, ISBN-13: 978-9380931913
2. The Elements of Statistical Learning: Data Mining, Inference and Prediction- T. Hastie, R. Tibshirani and J. Friedman, 2nd Edition, Springer publication, ISBN-10: 0387848576, ISBN-13: 978-0387848570

Web Links:

1. <https://data-flair.training/blogs/data-mining-tutorial/>
2. <https://machinelearningmastery.com/>
3. <https://towardsdatascience.com/>
4. <https://scikit-learn.org/stable/>
5. <https://nptel.ac.in/courses/106106139>

Note:

- The cornerstone project team size shall be four (4) students.
- Students may select any one of the above prescribed projects or a project of their own choice with the prior permission of the Course instructor.
- The evaluation of the Cornerstone Project shall be based on the project implementation, oral presentation, 10-minute video presentation, report and viva voce.
- The video presentation should consist of the working procedure of the project along with contribution of each student for a minimum of 2 minutes.
- Finally, the Source code of the cornerstone Project has to be pushed into the Students GitHub repository (if applicable).

MERN STACK TECHNOLOGIES

Course Code: 2504MC11

L	T	P	C
1	0	2	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Make use of HTML elements and their attributes for designing static web pages.
- CO2:** Build a web page by applying appropriate CSS styles to HTML elements.
- CO3:** Experiment with JavaScript to develop dynamic web pages and validate forms.
- CO4:** Build a basic web server using Node.js and working with Node Package Manager.
- CO5:** Make use of Typescript to optimize JavaScript code using strict type checking.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	1	2	1	-	1	1	-	2
CO2	2	-	1	3	-	2	1	-
CO3	1	3	1	1	2	1	1	1
CO4	2	-	2	1	-	3	1	2
CO5	-	1	2	3	1	-	-	1

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT – I

Introduction to HTML: Introduction to HTML tags, HTML Attributes, Lists, Links and Images, HTML Table Tags, Forms and Frames.

Practice:

1. Write a HTML program, to explain the working of lists.
 Note: It should have an ordered list, unordered list, nested list and ordered list in an unordered list and definition lists.
2. Write a HTML program, to explain the working of hyperlinks using <a> tag and href, target Attributes.
 Note: Use text to link <https://www.aec.edu.in/>
 Use image to link <https://www.aec.edu.in/?p=Gallery>
3. Create a HTML document that has your image and your friend's image with a specific height and width. Also, when clicked on the images it should navigate to their respective profiles.
4. Write a HTML program, to explain the working of tables. (Use tags: <table>, <tr>, <th>, <td> and attributes: border, rowspan, colspan)

5. Write a HTML program, to explain the working of tables by preparing a timetable. (Note: Use <caption> tag to set the caption to the table & also use cell spacing, cell padding, border, rowspan, colspan etc.).
6. Write a HTML program, to explain the working of forms by designing Registration form. (Note: Include text field, password field, number field, date of birth field, checkboxes, radio buttons, list boxes using <select> & <option> tags, <text area> and two buttons i.e. submit and reset. Use tables to provide a better view).
7. Write a HTML program, to explain the working of frames, such that page is to be divided into 3 parts on either direction. (Note: First frame image, Second frame paragraph, Third frame hyperlink. And make sure of using “no frame” attribute such frames to be fixed.)

UNIT – II

Introduction to CSS: HTML5 and Cascading Style Sheets, Types of CSS, Selector forms, CSS with Color, Background, Font, Text and CSS Box Model

Practice:

1. Write a HTML program, that makes use of <article>, <aside>, <figure>, <figcaption>, <footer>, <header>, <main>, <nav>, <section>, <div>, tags.
2. Write a HTML program, to embed audio and video into HTML web page.
3. Write a program to apply different types (or levels of styles or style specification formats) - inline, internal, external styles to HTML elements. (identify selector, property and value).
4. Write a program to apply different types of selector forms
 - i. Simple selector (element, id, class, group, universal)
 - ii. Combinator selector (descendant, child, adjacent sibling, general sibling)
 - iii. Pseudo-class selector
 - iv. Pseudo-element selector
 - v. Attribute selector
5. Write a program to demonstrate the various ways you can reference a color in CSS.
6. Write a CSS rule that places a background image halfway down the page, tilting it horizontally. The image should remain in place when the user scrolls up or down.
7. Write a program using the following terms related to CSS font and text:
 - i. font-size ii. font-weight iii. font-style
 - iv. text-decoration v. text-transformation vi. text-alignment
8. Write a program, to explain the importance of CSS Box model using
 - i. Content ii. Border iii. Margin iv. padding

UNIT – III

Javascript: History of javascript, Applying javascript - internal and external, I/O, Type Conversion, Datatypes, Arrays, variables, Pre-defined and User-defined Objects, Conditional Statements and Looping Statements, Functions and Events.

Practice:

1. Write a program to embed internal and external JavaScript on a web page.
2. Write a program to explain the different ways for displaying output.
3. Write a program to explain the different ways of taking input.
4. Create a webpage which uses prompt dialogue box to ask a voter for his name and age. Display the information in table format along with either the voter can vote or not.
5. Write a program using document and window object properties and methods.
6. Write a program using array object properties and methods.
7. Write a program using math object properties and methods.
8. Write a program using string object properties and methods.
9. Write a program using date object properties and methods.
10. Write a program to explain user-defined object by using properties, methods, accessors, constructors and display.
11. Write a program which asks the user to enter three integers, obtains the numbers from the user and outputs HTML text that displays the larger number followed by the words "LARGER NUMBER" in an information message dialog. If the numbers are equal, output HTML text as "EQUAL NUMBERS".
12. Design a HTML having a text box and four buttons named Factorial, Fibonacci, When a button is pressed an appropriate function should be called to display
 - i. Factorial of that number
 - ii. Fibonacci series up to that number
13. Write a program to validate the following fields in a registration page
 - i. Name (start with alphabet and followed by alphanumeric and the length should not be less than 6 characters)
 - ii. Mobile (only numbers and length 10 digits)
 - iii. E-mail (should contain format like xxxxxxxx@xxxxxx.xxx)

UNIT – IV

NODE.JS: Installation of Node.js, data transfer over Http protocol, Why and What Node.js, How to use Node.js, Create a web server in Node.js, Node Package Manager, Modular programming in Node.js, Restarting Node Application, File Operations.

Practice:

1. Write a program to show the workflow of JavaScript code executable by creating web server in Node.js.
2. Write a program to transfer data over http protocol using http module.
3. Create a text file src.txt and add the following content to it. (HTML, CSS, JavaScript, Typescript, MongoDB, Express.js, React.js, Node.js)
4. Write a program to parse an URL using URL module.
5. Write a program to create a user-defined module and show the workflow of Modularization of application using Node.js

UNIT – V

Typescript: Installing typescript, Basics of Typescript, Function, Parameter types and Return Types, Arrow Function, Function Types, Optional and Default Parameters, Rest Parameter, Function Types, Extending Interfaces, Classes, Constructor, Access Modifiers, Properties and Methods.

Practice:

1. Write a program to understand simple and special types.
2. Write a program to understand function parameters and return types.
3. Write a program to show the importance of Arrow function. Use optional, default and REST parameters.
4. Write a program to understand the working of typescript with class, constructor, properties, methods and access specifiers.

Additional Practice:

1. Write a CSS program, to apply 2D and 3D transformations in a web page
2. Design a web page with new features of HTML5 and CSS3.

Case Studies:

1. Student Course Registration and Report Generator
2. Feedback Collection and Analysis System
3. Online Portfolio Builder
4. Real-Time Chat Application
5. E-commerce Product Listing and Cart System

Textbooks:

1. “Programming the World Wide Web”, Robert W Sebesta ,7th Edition, Pearson Education, ISBN:978-0132665810.
2. “Ultimate Full-Stack Web Development with MERN”, Nabendu Biswas, 1st Edition orange Education, ISBN: 978-8119416424.
3. “Node.js Web Development: Server-Side Web Development Made Easy with Node 14”, David Heron,4th Edition, Packt Publishing, ISBN: 978-1801071046.

Reference Books:

1. “Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node”, Vasan Subramanian, “O’Reilly, 2nd edition, APress, ISBN:978-84243909.
2. “MERN Stack Front To Back: Full Stack React”, Redux and Node.js by Brad raversy, 1st Edition, ISBN: 978-1789343120.

Web Links:

1. <https://www.w3schools.com/html>
2. <https://www.w3schools.com/css>
3. <https://www.w3schools.com/js/>

MERN STACK TECHNOLOGIES

Course Code: 2604MC05

L	T	P	C
2	0	1	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Make use of HTML elements and their attributes for designing static web pages.
- CO2:** Build a web page by applying appropriate CSS styles to HTML elements.
- CO3:** Experiment with JavaScript to develop dynamic web pages and validate forms.
- CO4:** Build a basic web server using Node.js and working with Node Package Manager.
- CO5:** Make use of Typescript to optimize JavaScript code using strict type checking.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	1	2	1	-	1	1	-	2
CO2	2	-	1	3	-	2	1	-
CO3	1	3	1	1	2	1	1	1
CO4	2	-	2	1	-	3	1	2
CO5	-	1	2	3	1	-	-	1

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT – I

Introduction to HTML: Introduction to Web Technologies, Structure of HTML Document, HTML Elements and Attributes, Lists, Hyperlinks and Images, Tables, Forms and Form Controls, Frames and IFrames, Semantic HTML5

Practice:

1. Write a HTML program, to explain the working of tables. (Use tags: <table>, <tr>, <th>, <td> and attributes: border, rowspan, colspan)
2. Write a HTML program, to explain the working of forms by designing Registration form. (Note: Include text field, password field, number field, date of birth field, checkboxes, radio buttons, list boxes using <select> & <option> tags, <text area> and two buttons i.e. submit and reset. Use tables to provide a better view).
3. Write a HTML program, to explain the working of frames, such that page is to be divided into 3 parts on either direction. (Note: First frame image, Second frame paragraph, Third frame hyperlink. And make sure of using “no frame” attribute such frames to be fixed.)

UNIT – II

Introduction to CSS: Types of CSS, CSS Selectors, Colors and Backgrounds, Fonts and Text Formatting, CSS Box Model, Display Properties, CSS3 Features

Practice:

1. Write a program to apply different types (or levels of styles or style specification formats) - inline, internal, external styles to HTML elements. (identify selector, property and value).
2. Write a program to apply different types of selector forms
 - a. Simple selector (element, id, class, group, universal)
 - b. Combinator selector (descendant, child, adjacent sibling, general sibling)
 - c. Pseudo-class selector
 - d. Pseudo-element selector
 - e. Attribute selector

UNIT – III

Javascript: Introduction to JavaScript, Variables and Data Types, Operators, Input and Output, Arrays, Objects, Functions, Date Object, String Object, Conditional Statements, Loops, Form Validation.

Practice:

1. Write a program to embed internal and external JavaScript on a web page.
2. Write a program using document and window object properties and methods.
3. Design a HTML having a text box and four buttons named Factorial, Fibonacci, When a button is pressed an appropriate function should be called to display
 - a. Factorial of that number
 - b. Fibonacci series up to that number
4. Write a program to validate the following fields in a registration page
 - a. Name (start with alphabet and followed by alphanumeric and the length shouldnot be less than 6 characters)
 - b. Mobile (only numbers and length 10 digits)
 - c. E-mail (should contain format like xxxxxxxx@xxxxxxx.xxx)

UNIT – IV

NODE.JS: Introduction to Node.js, Installing Node.js, Architecture of Node.js, HTTP Module, Creating Web Servers, Node Package Manager (NPM), Modules in Node.js File System Module, URL Module

Practice:

1. Write a program to transfer data over http protocol using http module.
2. Write a program to parse an URL using URL module.
3. Write a program to create a user-defined module and show the workflow of Modularization of application using Node.js

UNIT – V

Typescript: Installing typescript, Basics of Typescript, Function, Parameter types and Return

Types, Arrow Function, Function Types, Optional and Default Parameters, Rest Parameter, Function Types, Extending Interfaces, Classes, Constructor, Access Modifiers, Properties and Methods.

Practice:

1. Write a program to show the importance of Arrow function. Use optional, default and REST parameters.
2. Write a program to understand the working of typescript with class, constructor, properties, methods and access specifiers.

Additional Practice:

1. Write a CSS program, to apply 2D and 3D transformations in a web page
2. Design a web page with new features of HTML5 and CSS3.

Textbooks:

1. “Programming the World Wide Web”, Robert W Sebesta ,7th Edition, Pearson Education, ISBN:978-0132665810.
2. “Ultimate Full-Stack Web Development with MERN”, Nabendu Biswas, 1st Edition orange Education, ISBN: 978-8119416424.
3. “Node.js Web Development: Server-Side Web Development Made Easy with Node 14”, David Heron, 4th Edition, Packt Publishing, ISBN: 978-1801071046.

Reference Books:

1. “Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node”, Vasan Subramanian, “O’Reilly, 2nd edition, APress, ISBN:978-84243909.
2. “MERN Stack Front To Back: Full Stack React”, Redux and Node.js by Brad raversy, 1st Edition, ISBN: 978-1789343120.

Web Links:

1. <https://www.w3schools.com/html>
2. <https://www.w3schools.com/css>
3. <https://www.w3schools.com/js/>

FUNDAMENTALS OF DATA SCIENCE

Course Code: 2504MC21

L	T	P	C
0	0	2	2

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Apply principles of NumPy and Pandas to the analysis of data.
- CO2:** Make use of various file formats in loading and storage of data
- CO3:** Identify and apply the need and importance of preprocessing techniques
- CO4:** Show the results and present them in a pictorial format
- CO5:** Facilitate new solutions for visualization of datasets by using different plotting techniques.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	2	1	3	-	-	-	2
CO2	3	1	2	-	1	-	-	-
CO3	2	3	2	-	-	1	-	3
CO4	2	2	3	3	2	-	-	-
CO5	3	2	-	-	1	1	-	-

Mapping of Course Outcomes with Program Specific Outcomes:

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

Practice:

1. Create a NumPy array of random integers. Demonstrate how Boolean indexing and fancy indexing can be used to select specific elements from this array.
2. Generate a NumPy array and illustrate how conditional logic can be expressed using array operations (e.g., setting values greater than a threshold to a specific number).
3. Create an array with repeated integer values. Demonstrate methods to sort this array and extract unique elements from it.
4. Given a DataFrame with employee details (name, department, salary), demonstrate sorting by salary in ascending and descending order, and then rank employees by salary.
5. Using Pandas, create a DataFrame with random numbers. Compute and display descriptive statistics including mean, median, standard deviation, minimum, and maximum values.
6. Create a DataFrame with some missing values. Demonstrate two methods to handle these missing values: filtering out missing data and filling them with appropriate values.
7. Write a Python script using Pandas to load data from a JSON file, convert it to a DataFrame, and save the processed data back into JSON format.
8. Show how to read data from a Microsoft Excel file using Pandas, perform basic data

cleaning operations, and then save the cleaned data into an HDF5 file format.

9. Create multiple small DataFrames and demonstrate how to concatenate them along rows (axis=0) and columns (axis=1).
10. Demonstrate reshaping a DataFrame using pivoting. Given a dataset with categorical variables, reshape it from long format to wide format and vice versa.
11. Illustrate data transformation operations by creating a DataFrame with duplicate entries and missing values. Demonstrate how to remove duplicates, replace specific values, and handle missing data.
12. Create a bar plot from a given dataset using Pandas and Matplotlib. Demonstrate customizing bar colors, labeling axes, and saving the final plot as an image file.

Additional Practice:

1. Using BeautifulSoup and Pandas, write code to perform web scraping on an HTML webpage and extract table data into a Pandas DataFrame.
2. Generate a dataset of random numbers and plot both a histogram and a density plot. Clearly explain the difference between these two visualizations.
3. Use a dataset to create a scatter plot with Pandas or Matplotlib. Demonstrate the customization of markers, colors, axis labels, and titles, and include a trendline on the plot.

Case Studies:

1. Student Performance Analysis: Load student data from a CSV file. Handle missing values, calculate basic statistics (mean, median, standard deviation), rank students by total score, and visualize the results using bar plots, histograms, and scatter plots.
2. Employee Salary Analysis: Load employee and department data from CSV files. Merge the datasets, sort and rank employees by salary, calculate average salary per department, and visualize the data using bar and line plots.
3. COVID-19 Web Data Analysis: Scrape COVID-19 data from an HTML webpage using BeautifulSoup. Convert it into a DataFrame, clean the data, calculate daily statistics, and visualize trends using line and bar plots with annotations.
4. Sales Data Analysis: Load monthly sales data from a CSV file. Perform data cleaning and compute total sales, average monthly sales, and identify the highest and lowest sales months. Visualize the trends using line plots and bar charts.
5. Movie Ratings Analysis: Load movie data from a CSV or JSON file containing titles, genres, and ratings. Filter movies by genre, calculate average ratings, find top-rated movies, and visualize the results using bar plots and pie charts.

Textbooks:

1. “Python for Data Analysis”, Wes McKinney, 1st edition, O’REILLY publication, ISBN: 9781449319793.
2. “Doing Data Science”, Rachel Schutt & O’neil, 1st edition, O’REILLY publication, ISBN: 9781449358655.

Reference Books:

1. “Data Science from Scratch :First Principles with Python”, JoelGrus, O’ReillyMedia,2nd edition, ISBN:978-1492041139.
2. “Learning the Pandas Library: Python Tools for Data Munging, Analysis, and Visualization” Matt Harrison, 1st edition, O’Reilly publication, ISBN:978-1533598240.

Web Links:

1. <https://www.geeksforgeeks.org/howtobecomeadataanalystcompleteroadmap/>
2. https://www.mygreatlearning.com/wiki/Data_Science_and_Predictive_Analytic

FUNDAMENTALS OF DATA SCIENCE

Course Code: 2604MC06

L	T	P	C
0	0	2	2

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Apply principles of NumPy and Pandas to the analysis of data.
- CO2:** Make use of various file formats in loading and storage of data
- CO3:** Identify and apply the need and importance of preprocessing techniques
- CO4:** Show the results and present them in a pictorial format
- CO5:** Facilitate new solutions for visualization of datasets by using different plotting techniques.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	2	1	3	-	-	-	2
CO2	3	1	2	-	1	-	-	-
CO3	2	3	2	-	-	1	-	3
CO4	2	2	3	3	2	-	-	-
CO5	3	2	-	-	1	1	-	-

Mapping of Course Outcomes with Program Specific Outcomes:

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

Practice:

1. Create a NumPy array of random integers. Demonstrate how Boolean indexing and fancy indexing can be used to select specific elements from this array.
2. Generate a NumPy array and illustrate how conditional logic can be expressed using array operations (e.g., setting values greater than a threshold to a specific number).
3. Create an array with repeated integer values. Demonstrate methods to sort this array and extract unique elements from it.
4. Given a DataFrame with employee details (name, department, salary), demonstrate sorting by salary in ascending and descending order, and then rank employees by salary.
5. Using Pandas, create a DataFrame with random numbers. Compute and display descriptive statistics including mean, median, standard deviation, minimum, and maximum values.
6. Create a DataFrame with some missing values. Demonstrate two methods to handle these missing values: filtering out missing data and filling them with appropriate values.
7. Write a Python script using Pandas to load data from a JSON file, convert it to a

DataFrame, and save the processed data back into JSON format.

8. Show how to read data from a Microsoft Excel file using Pandas, perform basic data cleaning operations, and then save the cleaned data into an HDF5 file format.
9. Create multiple small DataFrames and demonstrate how to concatenate them along rows (axis=0) and columns (axis=1).
10. Demonstrate reshaping a DataFrame using pivoting. Given a dataset with categorical variables, reshape it from long format to wide format and vice versa.
11. Illustrate data transformation operations by creating a DataFrame with duplicate entries and missing values. Demonstrate how to remove duplicates, replace specific values, and handle missing data.
12. Create a bar plot from a given dataset using Pandas and Matplotlib. Demonstrate customizing bar colors, labeling axes, and saving the final plot as an image file.

Additional Practice:

1. Using BeautifulSoup and Pandas, write code to perform web scraping on an HTML webpage and extract table data into a Pandas DataFrame.
2. Generate a dataset of random numbers and plot both a histogram and a density plot. Clearly explain the difference between these two visualizations.
3. Use a dataset to create a scatter plot with Pandas or Matplotlib. Demonstrate the customization of markers, colors, axis labels, and titles, and include a trendline on the plot.

Cornerstone Project:

Project-1: Student Performance Analytics Dashboard

Implementation Mainly Focuses On

- Import student data from CSV/Excel using Pandas.
- Clean missing and duplicate records.
- Compute descriptive statistics (mean, median, standard deviation).
- Rank students based on total marks.
- Visualize subject-wise and overall performance using bar charts, histograms, scatter plots, and pie charts.
- Export processed data to JSON and Excel formats.

Documentation Must Contain

- Problem statement
- Dataset description
- Data preprocessing steps
- Statistical analysis
- Visualization results
- Observations
- Screenshots of outputs

- Source code with comments
- Conclusion and future enhancements

Project-2: Employee Salary Analysis and Department Insights

Implementation Mainly Focuses On

- Load employee and department datasets.
- Merge multiple DataFrames.
- Handle missing salary values.
- Perform sorting, ranking, grouping, and aggregation.
- Calculate average, minimum, and maximum salaries by department.
- Visualize salary distribution using line, bar, and box plots.
- Save processed results into HDF5 and JSON formats.

Documentation Must Contain

- Objective
- Database structure
- Data cleaning methodology
- Group-by operations
- Graphical analysis
- Interpretation of departmental trends
- Code documentation
- References

Project-3: Sales Data Intelligence System

Implementation Mainly Focuses On

- Read monthly sales data from CSV/Excel.
- Perform preprocessing and missing-value handling.
- Compute monthly, quarterly, and yearly summaries.
- Identify highest- and lowest-performing products.
- Generate dashboards using Matplotlib.
- Export reports in Excel and JSON formats.

Documentation Must Contain

- Business objective
- Dataset description
- Data transformation
- Sales trend analysis
- Visualization screenshots
- Results and business recommendations
- User manual

Project-4: COVID-19 Web Data Collection and Visualization

Implementation Mainly Focuses On

- Scrape COVID-19 data from HTML pages using BeautifulSoup.
- Convert scraped data into Pandas DataFrames.
- Remove inconsistencies and missing values.
- Calculate daily statistics.
- Create line graphs, bar charts, moving averages, and trend visualizations.
- Save cleaned datasets for further analysis.

Documentation Must Contain

- Web source details
- Web scraping methodology
- Data cleaning techniques
- Statistical analysis
- Visualization outputs
- Limitations
- Ethical considerations
- Source code

Project-5: Movie Recommendation and Rating Analytics**Implementation Mainly Focuses On**

- Load movie data from CSV/JSON.
- Filter movies by genre and rating.
- Calculate average ratings and popularity metrics.
- Identify top-rated movies.
- Visualize genre-wise distribution, ratings, and popularity using bar, histogram, and pie charts.
- Export reports in JSON format.

Documentation Must Contain

- Introduction
Dataset information
- Data preprocessing
- Statistical findings
- Visualization analysis
- Recommendation logic
- Testing results
- References

Project-6: Comprehensive Data Cleaning and Visualization Toolkit**Implementation Mainly Focuses On**

- Develop a Python application that accepts CSV, Excel, and JSON files.

- Automatically detect missing values, duplicates, and outliers.
- Perform preprocessing using NumPy and Pandas.
- Generate descriptive statistics.
- Produce bar charts, histograms, scatter plots, density plots, and correlation visualizations.
- Export cleaned datasets and graphical reports automatically.

Documentation Must Contain

- System architecture
- Software requirements
- Functional modules
- Flowchart
- Algorithm
- Implementation details
- Testing methodology
- Sample inputs and outputs
- User guide
- Future scope

Textbooks:

1. “Python for Data Analysis”, Wes McKinney, 1st edition, O’REILLY publication, ISBN: 9781449319793.
2. “Doing Data Science”, Rachel Schutt & O’neil, 1st edition, O’REILLY publication, ISBN: 9781449358655.

Reference Books:

1. “Data Science from Scratch :First Principles with Python”, JoelGrus, O’ReillyMedia,2nd edition, ISBN:978-1492041139.
2. “Learning the Pandas Library: Python Tools for Data Munging, Analysis, and Visualization” Matt Harrison, 1st edition, O’Reilly publication, ISBN:978-1533598240.

Web Links:

1. <https://www.geeksforgeeks.org/howtobecomeadataanalystcompleteroadmap/>
2. https://www.mygreatlearning.com/wiki/Data_Science_and_Predictive_Analytic

Note:

- The cornerstone project team size shall be four (4) students.
- Students may select any one of the above prescribed projects or a project of their own choice with the prior permission of the Course instructor.
- The evaluation of the Cornerstone Project shall be based on the project implementation, oral presentation, 10-minute video presentation, report and viva voce.
- The video presentation should consist of the working procedure of the project along with contribution of each student for a minimum of 2 minutes.
- Finally, the Source code of the cornerstone Project has to be pushed into the Students GitHub repository (if applicable).

INTERNET OF THINGS (IoT)

Course Code: 2504MC12

L	T	P	C
0	0	2	2

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Choose the sensors and actuators for an IoT application.
- CO2:** Select protocols for a specific IoT application.
- CO3:** Utilize the cloud platform and APIs for IoT application.
- CO4:** Experiment with embedded boards for creating IoT prototypes.
- CO5:** Design a solution for a given IoT application.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	-	2	-	2	1	-	-
CO2	2	-	2	-	2	1	-	-
CO3	2	-	3	1	2	1	-	-
CO4	2	1	3	1	2	1	-	-
CO5	2	1	3	1	2	1	2	-

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1:	2	-
CO2:	2	-
CO3:	2	-
CO4:	1	-
CO5:	1	1

Practice:

1. Familiarization with Arduino/Raspberry Pi and perform necessary software installation.
2. To interface LED/Buzzer with Arduino/Raspberry Pi and write a program to turn ON LED for 1 sec after every 2 seconds.
3. To interface Push button/Digital sensor (IR/LDR) with Arduino/Raspberry Pi and write a program to turn ON LED when push button is pressed or at sensor detection.
4. To interface DHT11 sensor with Arduino/Raspberry Pi and write a program to print temperature and humidity readings.
5. To interface motor using relay with Arduino/Raspberry Pi and write a program to turn ON motor when push button is pressed.
6. To interface OLED with Arduino/Raspberry Pi and write a program to print temperature and humidity readings on it.
7. To interface Bluetooth with Arduino/Raspberry Pi and write a program to send sensor data to smart phone using Bluetooth.

8. To interface Bluetooth with Arduino/Raspberry Pi and write a program to turn LED ON/OFF when '1'/'0' is received from smart phone using Bluetooth.
9. Write a program on Arduino/Raspberry Pi to upload temperature and humidity data to thing speak cloud.
10. Write a program on Arduino/Raspberry Pi to retrieve temperature and humidity data from thing speak cloud.
11. How can an IoT-based system be designed to continuously monitor and send real-time alerts when temperature and humidity levels cross predefined thresholds?
12. Write a program on Arduino/Raspberry Pi to publish temperature data to MQTT broker.

Additional Practice:

1. Write a program on Arduino/Raspberry Pi to subscribe to MQTT broker for temperature data and print it.
2. Write a program to create UDP server on Arduino/Raspberry Pi and respond with humidity data to UDP client when requested.
3. Write a program to create TCP server on Arduino/Raspberry Pi and respond with humidity data to TCP client when requested.

Cornerstone Projects:

S. No.	Project Title	Short Summary
1	Smart Healthcare Patient Monitoring System	Develop an IoT healthcare platform that continuously monitors patient vital signs such as heart rate, temperature, and oxygen level, sending emergency notifications to doctors and caregivers.
2	IoT-Based Smart Agriculture Using Environmental Sensors	Build a precision farming system that monitors soil moisture, humidity, temperature, and weather conditions to automate irrigation and improve crop productivity.
3	Smart Home Automation with Mobile Application	Create a smart home system that controls lighting, fans, security cameras, and appliances remotely using smartphones and cloud-based IoT platforms.
4	Industrial Equipment Health Monitoring (Predictive Maintenance)	Implement IoT sensors on industrial machines to monitor vibration, temperature, and motor current, enabling predictive maintenance and reducing equipment failures.
5	Smart Waste Management System	Design an intelligent garbage bin monitoring system that tracks fill levels and optimizes waste collection routes using IoT sensors and GPS technology.

Textbooks:

1. Internet Of Things (IOT): Architecture and Design Principles, 2nd Edition, 2022, Raj Kamal Publisher McGraw Hill, ISBN-13: 978-9390727384.
2. Internet of Things: A Hands-On Approach, 2nd Edition, 2023, Vijay Madiseti and Arshdeep Bahga, Publisher Techno Series, ISBN-13: 978-0996025515.

Reference Books:

1. IoT Fundamentals, Networking Technologies, Protocols and Use Cases for the Internet of Things, 2nd Edition, 2021. David Hanes, Gonzalo Salgueiro, Patrick Grossetette, Rob Barton, Jerome Henry, CISCO, Pearson. ISBN-13: 978-1587144110
2. Designing the Internet of Things, 1st Edition, Adrian McEwen and Hakim Cassimally, Wiley. ISBN-13: 978-1118430625.
3. An Introduction to Internet of Things, Connecting devices, Edge Gateway and Cloud with Applications, 1st Edition, 2019. Rahul Dubey, Cengage, Adrian McEwen, Wiley Publishers. ISBN-13: 978-0359844174.

Web Links:

1. <https://iotify.io/iot-virtual-lab/>
2. <https://www.coursera.org/specializations/iot>
3. <https://www.tinkercad.com/>

Note:

- The cornerstone project team size shall be four (4) students.
- Students may select any one of the above prescribed projects or a project of their own choice with the prior permission of the Course instructor.
- The evaluation of the Cornerstone Project shall be based on the project implementation, oral presentation, 10-minute video presentation, report and viva voce.
- The video presentation should consist of the working procedure of the project along with contribution of each student for a minimum of 2 minutes.
- Finally, the Source code of the cornerstone Project has to be pushed into the Students GitHub repository (if applicable).

AGILE METHODOLOGIES

Course Code: 2504MC13

L	T	P	C
1	0	2	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Understand software lifecycle, agility, DevOps, CI/CD.
- CO2:** Practice Scrum, and Kanban for automation.
- CO3:** Simulate Scrum roles and ceremonies.
- CO4:** Apply customer collaboration, bottleneck resolution, and Git for CI/CD.
- CO5:** Implement Scrum for team formation and scaling, track project metrics.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	-	-	2	-	-	-	-
CO2	2	2	2	-	-	-	-	-
CO3	2	2	-	2	-	-	-	-
CO4	2	2	2	-	-	2	-	-
CO5	2	1	2	-	2	2	-	-

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT – I

Agile Software Development Ecosystems: Agile development model, DevOps, and ITIL, DevOps process and Continuous Delivery, Release management, Scrum, Kanban, DevOps delivery pipeline, bottlenecks, DevOps eco system.

Practice:

1. Get an understanding of the stages in software development lifecycle, the process Models.
2. Get an understanding of the values and principles of agility and the need for agile software development.

UNIT – II

Fundamentals of DevOps: DevOps Architecture, Deployments, Orchestration, Need, Instance of applications, DevOps delivery pipeline, DevOps eco system.

Practice:

1. It is important to comprehend the need to automate the software development lifecycle stages through DevOps.
2. Gain an understanding of the capabilities required to implement DevOps, continuous integration and continuous delivery practices.

UNIT – III

Backdrop-The Science of Scrum: The Skelton and Heart of Scrum, Scrum Roles, Scrum Flow, Scrum Artifacts New Management Responsibilities: The Scrum Master at Meta Eco, The Product Owner at Mega Energy, The team at Service 1st. The Scrum Master: The Untrained Scrum Master at Trey Research, The Untrained Scrum Master at Lit Ware.

Practice:

1. Set up a Kanban board and demonstrate the workflow.
2. Create a Scrum board using tools like Jira or Trello.
3. Discuss and document the responsibilities of Scrum roles in different case scenarios.

UNIT – IV

The Product Owner and Introduction to Ci/Cd: Customer and team Collaboration, **Planning A Scrum Project:** Managing Cash at Mega Bank, Certified Scrum Masters Take on Return on Investment. Introduction to Continuous Integration, Continuous Delivery and Deployment, Benefits of CI/CD, Metrics to track CICD practices.

Practice:

1. Configure the web application and Version control using Git using Git commands and version control operations.
2. Configure a static code analyzer which will perform static analysis of the web application code and identify the coding practices that are not appropriate.
3. Configure the profiles and dashboard of the static code analysis tool.

UNIT – V

The Team: Team Formation at Service 1ST, Giving the team a Chance at Web New Site. Scaling the Project with Scrum: Scaling at Mega Fund, Scrum Scaling, Scaling at Medcin soft. Rules: Sprint Planning Meeting, Daily Scrum Meeting, Sprint, Sprint Review Meeting, Sprint Retrospective Meeting.

Practice:

1. Create and manage Scrum artifacts (Product Backlog, Sprint Backlog, Burndown Chart) for a project.
2. Simulate Sprint Planning, Daily Stand-ups, Sprint Review, and Retrospective meetings.

Additional Practice:

1. Diagnose your team with the Scrum Team Survey.
2. Configure the Jenkins tool with the required paths, path variables, users and pipeline views.
3. Write a build script to build the application using a build automation tool like Maven. Create a folder structure that will run the build script and invoke the various software development build stages. This script should invoke the static analysis tool and unit testcases & deploy the application to a web application server like Tomcat.

Textbooks:

1. Agile Project Management with Scrum, Ken Schwaber, 1st Edition, Microsoft Press, ISBN: 978-0735619937.
2. The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations, Gene Kim, John Willis, Patrick Debois, Jez Humble, 1st Edition, O'Reilly publications, ISBN: 978-1942788003.

Reference Books:

1. Agile Software Engineering, Series: Undergraduate Topics in Computer Science, Hazza and Dubinsky, Springer, ISBN: 978-3319251319.
2. What is Devops? Infrastructure as code, 1st Edition, Mike Loukides, O'Reilly Publications, ISBN: 978-1491924358.
3. Agile Software Development, Cockburn, Highsmith, Alistair Cockburn Series, Addison-Wesley, ISBN: 978-0321249166.

Web Links:

1. <https://www.edx.org/course/agile-software-development>
2. <https://www.class-central.com/course/coursera-agile-software-development-9513>
3. <https://www.cprime.com/resources/what-is-agile-what-is-scrum/>
4. <https://www.atlassian.com/agile/kanban>
5. https://file.scirp.org/pdf/JCC_2017033115471602.pdf

CLOUD COMPUTING

Course Code: 2504MC14

L	T	P	C
3	0	0	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Explore the various service and deployment models in cloud computing.
- CO2:** Demonstrate the basic concepts of virtualization and implementation levels of Virtualization.
- CO3:** Illustrate the architecture of cloud computing.
- CO4:** Apply the cloud programming and software environments on any real cloud service.
- CO5:** Analyse the Cloud Security risks and Mechanisms.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	2	-	1	3	-	-	-	-
CO2	3	2	-	1	-	-	-	-
CO3	1	-	2	-	-	-	-	3
CO4	-	-	3	2	1	-	-	-
CO5	-	-	-	-	-	-	3	2

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT – I

Introduction to Cloud Computing: Cloud models, Evolution of cloud computing, system models for distributed and cloud computing, Cloud computing properties and characteristics, business drivers for adopting cloud computing, Software environments for distributed systems and clouds, performance, security and energy efficiency.

UNIT – II

Virtual machines and Virtualization of clusters and data centers: Implementation levels of virtualization, Virtualization structures / tools and mechanisms, Virtualization of CPU, memory and I/O devices, Virtual clusters and resource management, Virtualization for data center automation, Technology examples like Xen: Paravirtualization, VMware: Full Virtualization, Microsoft Hyper-V.

UNIT – III

Cloud platform architecture: Cloud computing and service models, Architectural design of compute and storage clouds, public cloud platforms, Inter cloud resource management, Cloud security and trust management. Service oriented architecture, message oriented middleware.

UNIT – IV

Programming model: Parallel and Distributed programming paradigms, Map reduce, Twister and iterative MaP reduce, Hadoop library from Apache, Mapping applications, programming support.

Security in the cloud: Security overview, Cloud security challenges, Access control mechanisms, Security governance, Risk management, Security monitoring, Security architecture design, Virtual machine security.

UNIT – V

Cloud Resource Management and Scheduling: Policies and mechanisms for resource management applications of control theory to task scheduling on a cloud, stability of a two-level resource allocation architecture, Feedback control based on dynamic thresholds. Coordination of specialized autonomic performance managers, Resource bundling, Scheduling algorithms for computing clouds, Fair queuing, Start time fair queuing, Borrowed virtual time, Cloud scheduling subject to deadlines, Scheduling MapReduce applications subject to deadlines.

Textbooks:

1. Cloud Computing: Principles, Systems and Applications- Nick Antonopoulos, Lee Gillam; Springer Publication; ISBN: 978-1-84882-889-1
2. Cloud Computing: Concepts, Technology & Architecture - Thomas Erl, Zaigham Mahmood, Ricardo Puttini; Prentice Hall Publication; ISBN: 978-0-13-338752-0

Reference Books:

1. Distributed and Cloud Computing: From Parallel Processing to the Internet of Things - Kai Hwang, Geoffrey C. Fox, Jack G. Dongarra; 1st Edition; Morgan Kaufmann Publishers; ISBN: 978-0-12-385880-8
2. Cloud Computing: Theory and Practice - Dan C. Marinescu; 1st Edition; Publication- Elsevier; ISBN: 978-0-12-404627-4

Web Links:

1. <http://nptel.ac.in/courses/106106129/28>
2. <https://aws.amazon.com/getting-started/tutorials/>
3. <https://www.coursera.org/learn/cloud-computing/lecture/VOIHP/introductionto-cloud-computing-concepts-part-1>
4. <https://www.lynda.com/Cloud-Computing-training-tutorials/1385-0.html>

OPERATING SYSTEMS

Course Code: 2504MC15

L	T	P	C
2	1	0	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Explain the fundamental concepts, structure, and functionalities of operating systems for efficient resource management.
- CO2:** Analyze process management, scheduling, synchronization, and deadlock handling techniques for system performance optimization.
- CO3:** Apply memory management strategies and virtualization techniques for effective utilization of computing resources.
- CO4:** Design file systems, storage management solutions, and evaluate distributed file systems in modern computing environments.
- CO5:** Demonstrate the use of security mechanisms, cloud-based OS architectures, and containerization technologies in real-world scenarios.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	2	1	-	-	-	-	-
CO2	3	3	2	1	-	-	-	-
CO3	3	2	3	2	-	-	-	-
CO4	3	3	3	2	-	1	-	-
CO5	3	2	2	2	-	-	3	2

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1	3	-
CO2	3	-
CO3	3	2
CO4	3	2
CO5	2	3

UNIT – I

Introduction to Operating Systems: Types (Mainframe, Desktop, Real-Time, Embedded, Cloud, Mobile OS). OS Structures: Kernel, Shell, System Calls. CPU Scheduling: Concepts and Algorithms (FCFS, SJF, RR, Priority).

UNIT – II

Process & Thread Management: Process States, PCB, Context Switching. IPC, Synchronization (Semaphores, Monitors). Deadlock Handling: Prevention, Avoidance, Detection & Recovery

UNIT – III

Memory Management: Background – Swapping - Contiguous Memory Allocation - Paging - Segmentation – Segmentation with paging - Virtual Memory: Demand paging - Page Replacement - Thrashing. Buddy Systems – Storage Compaction

UNIT – IV

File Systems & Storage: File Concepts, Access Methods, Directory Structures, File Protection. Disk Scheduling (FCFS, SSTF, SCAN). Introduction to Distributed File Systems (HDFS, S3).

UNIT – V

Distributed & Modern Operating Systems: Transparency, Coordination, Consensus (Raft, Paxos). Cloud OS (AWS, Azure). OS Security: Sandboxing, SE-Linux. Case Studies: Linux, Android OS, Kubernetes.

Textbooks:

1. Operating System Concepts - Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, 10th Edition, Wiley, ISBN: 978-1-119-32491-7
2. Modern Operating Systems - Andrew S. Tanenbaum, Herbert Bos, 4th Edition, Pearson, ISBN: 978-0-13-359162-0

Reference Books:

1. Operating Systems: Internals and Design Principles - William Stallings, 9th Edition, Pearson, ISBN: 978-0-13-467095-9
2. Operating Systems: Three Easy Pieces - Remzi H. Arpaci-Dusseau, Andrea C. Arpaci-Dusseau, Latest Online Edition, ISBN: 978-1-7332691-0-3
3. Kubernetes: Up & Running - Kelsey Hightower, Brendan Burns, Joe Beda, 3rd Edition, O'Reilly Media, ISBN: 978-1-492-05475-3

Web Links:

1. <https://ocw.mit.edu/courses/6-828-operating-system-engineering-fall-2012/>
2. <https://nptel.ac.in/courses/106105214>
3. <https://www.udacity.com/course/linux-command-line-basics--ud595>
4. <https://training.linuxfoundation.org/training/introduction-to-linux/>

INFORMATION RETRIEVAL SYSTEMS

Course Code: 2504MC16

L	T	P	C
3	0	0	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Ability to apply IR principles to locate relevant information in large collections of data.
- CO2:** Ability to design different document clustering algorithms.
- CO3:** Implement retrieval systems for web search tasks.
- CO4:** Design an Information Retrieval System for web search tasks.
- CO5:** Understand principles of text search techniques and Multimedia Information Retrieval.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	2	-	-	-	-	-	-	-
CO2	-	2	-	-	-	-	-	-
CO3	-	-	2	-	-	-	-	-
CO4	-	-	-	2	-	-	-	-
CO5	-	-	-	2	-	-	-	-

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1:	2	-
CO2:	-	2
CO3:	-	-
CO4:	-	-
CO5:	-	-

UNIT – I

Introduction to Information Retrieval Systems: Definition of Information Retrieval System, Objectives of Information Retrieval Systems, Functional Overview, Relationship to Database Management Systems, Digital Libraries and Data Warehouses

Information Retrieval System Capabilities: Search Capabilities, Browse Capabilities, Miscellaneous Capabilities

UNIT – II

Cataloging and Indexing: History and Objectives of Indexing, Indexing Process, Automatic Indexing, Information Extraction

Data Structure: Introduction to Data Structure, Stemming Algorithms, Inverted File Structure, N-Gram Data Structures, PAT Data Structure, Signature File Structure, Hypertext and XML Data Structures, Hidden Markov Models

UNIT – III

Automatic Indexing: Classes of Automatic Indexing, Statistical Indexing, Natural Language, Concept Indexing, Hypertext Linkages

Document and Term Clustering: Introduction to Clustering, Thesaurus Generation, Item Clustering, Hierarchy of Clusters

UNIT – IV

User Search Techniques: Search Statements and Binding, Similarity Measures and Ranking, Relevance Feedback, Selective Dissemination of Information Search, Weighted Searches of Boolean Systems, Searching the INTERNET and Hypertext

Information Visualization: Introduction to Information Visualization, Cognition and Perception, Information Visualization Technologies

UNIT – V

Text Search Algorithms: Introduction to Text Search Techniques, Software Text Search Algorithms, Hardware Text Search Systems

Multimedia Information Retrieval: Spoken Language Audio Retrieval, Non-Speech Audio Retrieval, Graph Retrieval, Imagery Retrieval, Video Retrieval

Textbooks:

1. “Multimedia Information Retrieval”: Content-Based Information Retrieval from Large Text and Audio Databases” 1st Edition, by Peter Schäuble. ISBN-13: 978-0262023485.
2. “Information Storage and Retrieval Systems” – Theory and Implementation, 2nd Edition, Gerald J. Kowalski, Mark T. Maybury, Springer. ISBN-13: 978-0387406038.

Reference Books:

1. Information Retrieval Data Structures and Algorithms, 1st Edition, Prentice Hall. Frakes, W.B., Ricardo Baeza-Yates: ISBN-13: 978-0131661999
2. Information Storage & Retrieval, 1st Edition, By Robert Korfhage – John Wiley & Sons. ISBN-13: 978-0471048688.
3. “Intelligent Multimedia Information Retrieval”, 1st Edition, by Giorgio Giunchiglia, Fausto Giunchiglia, and Enrico Blanzieri. ISBN-13: 978-3540186813.

Web Links:

1. <https://www-nlp.stanford.edu/IR-book/>
2. <https://www.geeksforgeeks.org/what-is-information-retrieval/>
3. https://www.tutorialspoint.com/natural_language_processing/natural_language_processing_information_retrieval.htm

RECOMMENDER SYSTEMS

Course Code: 2504MC17

L	T	P	C
3	0	0	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** To learn the goal and applicability of recommendation system in various fields.
- CO2:** To acquire knowledge of basics of content-based recommendation system.
- CO3:** To learn various types of memory based collaborative recommendation techniques.
- CO4:** To develop skills of ensemble and hybrid recommendation systems.
- CO5:** To develop skills of ensemble and hybrid recommendation systems.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	2	2	-	-	-	-	-	-
CO2	-	3	3	-	-	-	-	-
CO3	-	3	2	2	-	-	-	-
CO4	-	2	2	-	-	-	3	3
CO5	-	3	3	-	-	-	-	-

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1	3	-
CO2	2	3
CO3	-	2
CO4	-	3
CO5	3	-

UNIT – I

Introduction to Recommender Systems: Recommender Systems Function, Recommendation Techniques, Recommender Systems as a Multi-Disciplinary Field, Challenges.

UNIT – II

Content-Based Recommender Systems: Basic Components of Content-Based Systems, Preprocessing and Feature Extraction, Learning User Profiles and Filtering, Nearest Neighbor Classification.

UNIT – III

Collaborative Filtering Approaches: User-Based collaborative filtering, Similarity Function Variants, Variants of the Prediction Function, Item-Based Collaborative filtering, Comparing User-Based and Item-Based Methods, Strengths and Weaknesses of Neighborhood-Based Methods.

UNIT – IV

Advanced Collaborative Filtering Techniques: Rule-Based Collaborative Filtering, Association Rules, Naive Bayes Collaborative Filtering, Neural Network, Singular Value Decomposition, Stochastic Gradient Descent, Regularization.

UNIT – V

Hybrid Recommender Systems: Weighted Hybrids, Switching Hybrids, Cascade Hybrids, Feature Augmentation Hybrids, Meta-Level Hybrids, Feature Combination Hybrids.

Textbooks:

1. Recommender Systems Handbook: 1st Edition, Francesco Ricci, Lior Rokach, Bracha Shapira, Paul B. Kantor, Publisher: Springer. ISBN: 978-1461422070.
2. Recommender Systems: The Textbook. 1st Edition, Charu C. Aggarwal, Springer. ISBN: 978-3319260070.

Reference Books:

1. Recommender Systems: An Introduction, 1st Edition, by Dietmar Jannach, Markus Zanker, Alexander Felfernig, Gerhard Friedrich, Publisher: Cambridge University Press. ISBN: 978-0521493369.
2. Introduction to Information Retrieval, 1st Edition, Authors: Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, Publisher: Cambridge University Press. ISBN: 978-0521865715
3. Collaborative Filtering Recommender Systems, 1st Edition, Authors: Michael D. Ekstrand, John T. Riedl, Joseph A. Konstan. ISBN: 978-1601986276.
4. Recommender Systems Handbook. 2nd Edition, Francesco Ricci, Lior Rokach, Bracha Shapira, Paul B. Kantor, Springer. ISBN-13: 978-1493938491.

Web Links:

1. <https://www.coursera.org/learn/recommender-systems>
2. <https://towardsdatascience.com/the-netflix-prize-and-recommender-systems-439fcd33f1e9>
3. <https://medium.com/syncedreview/recommender-systems-in-practice-e-commerce-movies-and-music-64d51e88937>
4. <https://www.datacamp.com/community/tutorials/recommender-systems-python>
5. <https://towardsdatascience.com/evaluation-metrics-for-recommender-systems-df56c6611093>

NEURAL NETWORKS

Course Code: 2504MC18

L	T	P	C
2	1	0	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Understand Neural Network Fundamentals.
- CO2:** Master Learning Processes in Neural Networks.
- CO3:** Apply Perceptron Models.
- CO4:** Implement Multilayer Perceptron's and Backpropagation.
- CO5:** Explore Advanced Neural Network Concepts.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	1	2	2	-	-	-	-	-
CO2	2	3	3	-	-	-	-	-
CO3	3	3	3	-	-	-	-	-
CO4	3	3	3	-	-	-	-	-
CO5	3	3	3	-	-	-	-	-

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT – I

Introduction: What is a neural network? Models of a Neuron, Neural networks viewed as Directed Graphs, Network Architectures, Knowledge Representation, Artificial Intelligence and Neural Networks.

Learning Process 1: Error Correction learning, Memory based learning.

UNIT – II

Learning Process 2: Competitive, Boltzmann learning, Credit Assignment Problem, Memory, Adaption, Statistical nature of the learning process.

Single Layer Perceptrons - Adaptive filtering problem, Linear least square filters, least mean square algorithm, learning curves, Learning rate annealing techniques, perception -convergence theorem, Relation between perception and Bayes classifier for a Gaussian Environment.

UNIT – III

Multilayer Perceptron: Back propagation algorithm XOR problem, Heuristics, Output representation and decision rule, feature detection,

Back Propagation: back propagation and differentiation, Generalization, Cross validation, Network pruning Techniques, Virtues and limitations of back propagation learning, Accelerated convergence, supervised learning.

UNIT – IV

Self Organization Maps: Two basic feature mapping models, Self-organization map, SOM algorithm, properties of feature map, computer simulations, learning vector quantization, Adaptive pattern classification.

UNIT – V

Neurodynamics: Dynamical systems, stability of equilibrium states, attractors.

Hopfield Models -Hopfield models, computer experiment I.

Textbooks:

1. Neural networks a comprehensive foundation, Simon Haykin, Pearson Education 2nd Edition. ISBN-13: 978-0131471399.
2. Artificial neural networks, 1st Edition, B.Vegnanarayana Prentice Hall of India P Ltd., ISBN-13: 978-0131531518.

Reference Books:

1. Neural networks in Computer intelligence, 1st Edition, Li Min Fu TMH., ISBN-13:978-0070260067.
2. Neural networks, 1st Edition, Pearson Education, James A Freeman David M S Kapura ISBN-13: 978-0201544350.
3. Bishop, C. M. Neural Networks for Pattern Recognition. 1st Edition, Oxford University Press. ISBN-13: 978-0198538646.

Web Links:

1. https://onlinecourses.nptel.ac.in/noc19_ee53/preview
2. <http://neuralnetworksanddeeplearning.com/>
3. <https://webreference.com/ai/neural-networks/>

DEEP LEARNING

Course Code: 2504MC19

L	T	P	C
2	0	1	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Understand the basics of neural networks and their use in deep learning.
- CO2:** Learn to implement and use convolutional neural networks for image analysis.
- CO3:** Gain skills in implementing recurrent neural networks for sequential data tasks.
- CO4:** Learn about autoencoders, variational autoencoders, and generative adversarial networks for unsupervised learning.
- CO5:** Explore restricted Boltzmann machines and their applications in pattern recognition and data representation.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	2	-	1	-	-	-	3
CO2	2	1	2	3	-	-	-	-
CO3	2	3	-	1	-	-	-	1
CO4	2	2	-	3	-	1	-	-
CO5	2	2	-	3	-	-	-	-

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT – I

Fundamentals of Deep Networks: Neural networks, training neural networks, defining deep learning, Common architectural principles of deep networks: parameters, Layers, Activation functions, Loss functions, Hyper parameters, Building blocks of Deep Networks-RBMs, and Autoencoders.

Practice:

1. Implement a simple feedforward neural network using a library like TensorFlow or PyTorch.
2. Experiment with different activation functions (e.g., sigmoid, tanh, ReLU) and observe their impact on training.
3. Explore the impact of hyperparameters on the performance of a deep neural network.

UNIT – II

Convolutional Neural Networks: Introduction - components of CNN architecture - Rectified Linear Unit (ReLU) Layer -Exponential Linear Unit (ELU, or SELU) - Unique properties of CNN -Architectures and Applications of CNN.

Practice:

1. Implement basic CNN architecture using TensorFlow or Keras.
2. Visualize the filters learned by different convolutional layers in a trained CNN.

UNIT – III

Recurrent Neural Network: Introduction- The Architecture of Recurrent Neural Network- The Challenges of Training Recurrent Networks- Echo-State Networks- Long Short-Term Memory (LSTM); Applications of RNN.

Practice:

1. Implement a basic RNN model using TensorFlow or PyTorch for text generation.
2. Use an RNN (e.g., LSTM) to predict future values in a time series dataset.

UNIT – IV

Deep Unsupervised Learning: Autoencoders (standard, sparse, denoising, contractive, etc), variational Auto encoders, denoising encoders, Adversarial generative networks.

Practice:

1. Implement basic GAN architecture using TensorFlow or PyTorch.

UNIT – V

Deep Belief Networks: Energy-Based Models, Restricted Boltzmann Machines, Sampling in RBM. Applications of deep neural networks in handwritten character recognition, face recognition, semantic web, social networks.

Practice:

1. Implement RBM using libraries like TensorFlow or PyTorch.
2. Use a pre-trained deep neural network (e.g., CNN or RNN) to recognize handwritten characters from the MNIST dataset.

Textbooks:

1. Deep Learning - Ian Goodfellow, Yoshua Bengio, Aaron Courville, 1st Edition, The MIT Press publication, ISBN-10: 0262035618, ISBN-13: 978-0262035613
2. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems - Aurélien Géron, Edition 3, O'Reilly Media Publication on 2022-11-08, ISBN-13: 9781098125974, ISBN-10: 1098125975

Reference Books:

1. Dive into Deep Learning - Aston Zhang, Zachary C. Lipton, Mu Li, Alexander J. Smola, 1st Edition, Cambridge University Press on 2023, ISBN-13: 9781009389433
2. Deep Learning: Foundations and Concepts - K. Taylor, 1st Edition, Publisher: Springer, Publication Date: 2023, ISBN-13: 9783031454677, ISBN-10: 3031454677

Web Links:

1. <http://deeplearning.stanford.edu/tutorial/>
2. <https://nptel.ac.in/courses/106/106/106106184/>
3. <https://www.coursera.org/specializations/deep-learning>
4. <https://www.deeplearningbook.org>

NATURAL LANGUAGE PROCESSING

Course Code: 2504MC20

L	T	P	C
2	1	0	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Understand the fundamentals and applications of NLP using NLTK and context-free grammar.
- CO2:** Perform syntactic analysis and ambiguity resolution using pre-processing techniques.
- CO3:** Implement feature engineering techniques and NLP algorithms for parsers.
- CO4:** Apply TFIDF, Vectorization, and advanced feature engineering for NLP tasks.
- CO5:** Explain text summarization and sentiment analysis, and explore unsupervised methods.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	2	1	3	1	-	-	1
CO2	1	3	2	3	-	1	1	2
CO3	3	2	3	2	-	-	1	2
CO4	2	-	3	2	1	1	-	-
CO5	1	3	2	3	-	1	1	2

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT – I

Overview of Natural Language Processing and Python: Understanding natural language processing and applications, NLTK, Corpus and Dataset, understanding structure of sentence and Defining context-free grammar, Morphological Analysis NLP Applications - NLP hard Ambiguity- Algorithms and models, Knowledge Bottlenecks in NLP- Introduction to NLTK, Case study.

UNIT – II

Syntactic Analysis and Semantic Analysis: Ambiguity resolution, Discourse integration, Preprocessing, tokenization, stemming, lemmatization, and Word tokenization.

Parsing and Syntax: Word Level Analysis: Regular Expressions, Text Normalization, Edit Distance, Parsing and Syntax-Spelling, Error Detection and correction- Words and Word Classes-Part-of speech Tagging, Naive Bayes and Sentiment Classification: Case study.

UNIT – III

Feature engineering: NLP algorithms parsers, context free grammar, different types of parsers, POS tagging and different types of POS parsers.

Semantic Analysis and Discourse Processing: Meaning Representation-Lexical Semantics- Ambiguity-Word Sense Disambiguation. Discourse Processing: cohesion-reference resolution- discourse coherence and structure.

UNIT – IV

Basic Statistical Features of NLP: TF-IDF, Vectorization encoders and decoders, Normalization, Advanced feature engineering and NLP algorithms Basics of Word2Vec

Natural Language Generation: Architecture of NLG Systems, Applications, Machine Translation: Problems in Machine Translation- Machine Translation Approaches- Evaluation of machine translation systems. Case study: Characteristics of Indian Languages.

UNIT – V

Rule Based system for NLP: Machine Learning for NLP problems, Applications of NLP, Text Summarization, Sentiment Analysis.

Methods in NLP: Graphical Models for Sequence Labelling in NLP.

Textbooks:

1. “Python Natural Language Processing”, Thanaki J, 1st edition, Packt Publishing Ltd, ISBN:978-1787126786.
2. “Speech and Language Processing”, Daniel Jurafsky and James H., 2nd Edition, Martin Prentice Hall, ISBN:978-0131873216.

Reference Books:

1. “Foundations of Statistical Natural Language Processing”, Cambridge, Manning and Hinrich Schutze, 1st edition, MIT Press, ISBN:978-0262133609.
2. “Speech & language processing” Jurafsky D, 1st edition, Pearson Education India, ISBN:978-0130950697.
3. “Foundations of Statistical Natural Language Processing” Christopher D. Manning and Hinrich Schütze, 1st edition, MIT Press, ISBN:978-0262133609.

Web Links:

1. <https://www.nltk.org/data.html>
2. <https://www.nlp-fundamentals.com/>

FOUNDATIONS OF DATA ENGINEERING

Course Code: 2504MC49

L	T	P	C
2	0	1	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Describe the fundamental concepts of data engineering, data lifecycle, and the role of data engineers in modern data ecosystems.
- CO2:** Apply techniques to ingest structured and unstructured data from various sources using batch and streaming methods.
- CO3:** Design and implement data storage solutions using RDBMS, NoSQL, data lakes, and modern file formats.
- CO4:** Develop data pipelines using tools such as Apache Spark, Kafka, and Airflow for batch and real-time processing.
- CO5:** Evaluate the performance, scalability, and quality of data workflows and pipelines based on modern data engineering practices.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	2	1	1	1	-	-	-
CO2	2	3	3	3	3	-	-	-
CO3	2	3	3	3	3	-	-	-
CO4	2	2	3	2	3	-	-	-
CO5	2	3	3	3	3	-	-	-

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT – I

Introduction to Data Engineering: What is Data Engineering, The Data Lifecycle, Data Engineering vs. Data Science, Types of Data (Structured, Semi-structured, Unstructured), Roles and Responsibilities of a Data Engineer, Introduction to ETL (Extract, Transform, Load).

Practice:

1. Explain the role of a Data Engineer in a modern data-driven organization. How is it different from a Data Scientist?
2. Describe the stages of the data lifecycle. Illustrate with an example from an e-commerce platform.

UNIT – II

Data Collection and Ingestion: Data Sources: APIs, Logs, Files, Databases, Data Ingestion Techniques: Batch vs. Stream, Tools: Apache Kafka, Apache Flume, Sqoop, Real-Time vs. Near-Real-Time Data Ingestion.

Practice:

1. Compare batch data ingestion and real-time data ingestion. Provide one example use-case for each.
2. Write a Python script to fetch data from a public API (e.g., weather or COVID-19 API) and store it in a local file.

UNIT – III

Data Storage and Management: Relational Databases (RDBMS): MySQL, PostgreSQL, NoSQL Databases: MongoDB, Cassandra, HBase, Data Warehousing: Snowflake, Amazon Redshift, Google BigQuery, Data Lakes and Lakehouses, File Formats: CSV, JSON, Avro, Parquet, ORC.

Practice:

1. Differentiate between RDBMS and NoSQL databases with examples. When would you prefer one over the other?
2. Explain the advantages of using Parquet or Avro over CSV for big data processing.

UNIT – IV

Data Processing and Transformation: Batch Processing: Apache Hadoop, Apache Spark, Stream Processing: Apache Flink, Apache Storm, Data Cleaning and Validation, Data Transformation with Spark, Pandas, Workflow Orchestration: Apache Airflow, Luigi.

Practice:

1. Write a PySpark program to read a CSV file, remove null entries, and perform a basic group-by operation.
2. What are the major differences between batch processing and stream processing? Give examples using Apache Spark and Apache Flink.

UNIT – V

Data Pipelines and Deployment: Designing Scalable Data Pipelines, Monitoring and Logging, CI/CD for Data Pipelines, Data Governance and Quality, Case Studies: End-to-End Data Pipeline Projects.

Practice:

1. Design a simple ETL pipeline for a movie recommendation system. List the steps and tools you would use.
2. Explain how Apache Airflow can be used for scheduling and monitoring a data pipeline. Describe the concept of DAG with an example.

Textbooks:

1. Designing Data-Intensive Applications: Martin Kleppmann, March 2017, Publisher(s): O'Reilly Media, Inc., ISBN-13: 9781491903100.
2. Fundamentals of Data Engineering: Joe Reis and Matt Housley, June 2022, Publisher(s): O'Reilly Media, Inc., ISBN-13: 9781098108304.

Reference Books:

1. Streaming Systems: Tyler Akidau, Slava Chernyak, and Reuven Lax, July 2018. Publisher(s): O'Reilly Media, Inc. ISBN-13: 9781491983874.
2. Big Data: Principles and Best Practices of Scalable Real-Time Data Systems: Nathan Marz, Publisher: Dreamtech Press (1 January 2015), ISBN-13: 978-9351198062.

Web Links:

1. <https://cloud.google.com/learn/learning-paths/data-engineer>
2. <https://github.com/DataTalksClub/data-engineering-zoomcamp>
3. <https://aws.amazon.com/blogs/big-data/>

NoSQL DATABASES

Course Code: 2504MC22

L	T	P	C
2	0	1	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Understand NoSQL databases, perform CRUD operations in NoSQL environments.
- CO2:** Utilize MongoDB for document storage, CRUD operations, indexing, and aggregation.
- CO3:** Implement key-value stores (Memcached, Redis) for data storage and retrieval.
- CO4:** Compare SQL and MongoDB query features, access data in column-oriented databases (HBase), and query Redis data stores.
- CO5:** Implement indexing in MongoDB, CouchDB, and Apache Cassandra.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	1	-	-	-	-	-	-	-
CO2	1	2	3	-	-	-	-	-
CO3	-	-	-	3	-	-	-	-
CO4	-	2	2	-	-	3	-	-
CO5	2	3	-	-	3	-	-	-

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT – I

Introduction to NoSQL: Definition and introduction, Traditional relational databases vs NoSQL databases, types of NoSQL databases, storing and accessing data in NoSQL databases, Querying in NoSQL databases, evolution and motivation behind NoSQL databases, characteristics, advantages, and challenges of NoSQL databases Performing crud operations, creating records, Accessing data, updating and deleting Data.

Practice:

- Software Installation
- Write a program to connect to a MongoDB database and perform basic CRUD operations (Create, Read, Update, Delete) on a collection

UNIT – II

Document Stores: Introduction to MongoDB, Importance of NoSQL databases, Data types, documents, nested Documents, CRUD Operations, Basic cursor methods: map, to Array, pretty, for Each, limit, count, sort, Columnar Databases, Indexing and Aggregation, MongoDB Node JS Drivers and CAP theorem, spatial and temporal queries in MongoDB.

Practice:

1. Develop a program to utilize MongoDB's Aggregation Framework.
2. Write spatial and temporal queries in mongoDB

UNIT – III

Key Value Databases: What Is a KeyValue Store, KeyValue Store Features, Suitable Use Cases, Understanding Key/Value Stores in Memcached and Redis, Eventually Consistent Non-Relational Databases., When Not to Use.

Document Databases: What Is a Document Database, Features, Suitable Use Cases, When Not to Use NoSQL Data Bases.

Practice:

1. Write a program to create a simple keyvalue store that allows users to store and retrieve data using keys.
2. Implement functions to insert keyvalue pairs, retrieve values by key, update values, and delete keyvalue pairs.

UNIT – IV

NoSQL Stores: Similarities between Sql and Mongoddb query features, accessing data from column-oriented databases like Hbase, querying Redis data stores, changing document databases, schema evolution in column-oriented databases, Hbase data import and export, data evolution in key/Value stores.

Practice:

1. Design and create a column-oriented table schema in HBase.
2. Store and retrieve key/value pairs in Redis.
3. Perform basic data operations (SET, GET, DEL) and implement more complex

UNIT – V

Indexing and Ordering Data Sets: Essential concepts behind a database index, indexing and ordering in Mongoddb, creating and using indexes in Mongoddb, indexing and ordering in Couchdb, indexing in Apache Cassandra.

Practice:

1. Create a collection and insert sample documents.
2. Create indexes on single fields, compound indexes (multiple fields), and multikey indexes (arrays).

Textbooks:

1. "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence" by Pramod J. Sadalage and Martin Fowler, Addison-Wesley, ISBN: 978-0321826626
2. Making Sense of NoSQL, Dan McCreary and Ann Kelly, 1st Edition Manning Publications, ISBN: 978-1617290651.

Reference Books:

1. Professional NoSQL, Shashank Tiwari, 1st Edition, Wrox Press, Wiley, ISBN: 978-0-470-94224
2. Getting Started with NoSQL, Gaurav Vaish, 1st Edition, Packt Publishing, ISBN: 978-1782161315.

Web Links:

1. <https://www.coursera.org/courses?query=nosql>
2. <https://www.edx.org/learn/nosql>
3. <https://www.udemy.com/topic/nosql/>

BIG DATA ANALYTICS

Course Code: 2504MC23

L	T	P	C
3	0	0	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Understand the fundamentals, characteristics, and challenges of big data and the Hadoop Ecosystem.
- CO2:** Use various techniques for mining data stream.
- CO3:** Demonstrate Building blocks of Hadoop.
- CO4:** Make use of Pig and Hive to structure and work with big Data.
- CO5:** Evaluate and implement NoSQL databases and visualization techniques.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	-	-	2	-	-	-	-
CO2	-	2	-	3	-	-	-	-
CO3	2	-	-	3	-	-	-	-
CO4	-	-	2	3	-	-	-	-
CO5	-	-	2	3	-	-	-	2

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT – I

Introduction to Big Data: What is big data, why big data, Evolution of Big data, Best Practices for Big data Analytics, Big data characteristics, Validating, The Promotion of the Value of Big Data, Big Data Use Cases, Characteristics of Big Data Applications, Understanding Big Data Storage, A General Overview of High, Performance Architecture, HDFS, MapReduce and YARN, Map Reduce Programming Model. Features of big data, big data challenges, Hadoop and its features

UNIT-II

Stream Processing: Introduction to Streams Concepts, Stream Data Model and Architecture, Stream Computing, Sampling Data in a Stream, Filtering Streams, Counting Distinct Elements in a Stream, Estimating Moments, Counting Oneness in a Window, Decaying Window, Real time Analytics Platform (RTAP) Applications.

UNIT – III

Introduction to Hadoop: History of Hadoop, the Hadoop Distributed File System, Components of Hadoop Analysing the Data with Hadoop, Scaling Out, Hadoop Streaming, Design of HDFS, Java interfaces to HDFS Basics, developing a Map Reduce Application, How Map Reduce Works, Anatomy of a Map Reduce Job run, Failures, Job Scheduling, Shuffle and Sort, Task execution, Map Reduce Types and Formats, Map Reduce Features Hadoop environment.

UNIT – IV

Pig: Hadoop Programming Made Easier: Admiring the Pig Architecture, going with the Pig Latin Application Flow, working through the ABCs of Pig Latin, Checking out the Pig Script Interfaces, Scripting with Pig Latin. Working with Hive Data Types, Creating and Managing Databases and Tables, Seeing How the Hive Data Manipulation Language Works with examples, Querying and Analyzing Data.

UNIT – V

SPARK: Installing Spark, Spark applications, Jobs, stages and Tasks, Resilient Distributed data sets, Shared Variables, Anatomy of a Spark job run.

NoSQL Data Management for Big Data and Visualization, NoSQL Databases:

Schema-less Models, Increasing Flexibility for Data Manipulation, Key Value Stores, Document Store, Tabular Stores, Object Data Stores, Graph Databases Hive, Sharding, HBase, Analysing big data with twitter, Big data for E- Commerce Big data for blogs.

Textbooks:

1. Hadoop: The Definitive Guide, Tom White, O'Reilly, 4th Edition, ISBN: 978-1491901632
2. Hadoop for Dummies, Dirk deRoos, Paul C.Zikopoulos, Roman B.Melnyk, Bruce Brown, Rafael Coss, John Wiley & Sons, ISBN: 978-1118727128.

Reference Books:

1. Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics, Bill Franks, John Wiley & sons, ISBN: 978-1118224912.
2. Harness the Power of Big Data: The IBM Big Data Platform, Paul Zikopoulos, DirkdeRoos, Krishnan Parasuraman, Thomas Deutsch, James Giles, David Corrigan, Tata McGraw Hill Publications, ISBN: 9780071808187.
3. Learning Spark: Lightning-Fast Big Data Analysis Paperback, Holden Karau, ISBN: 978-1449358624.

Web Links:

1. <https://www.coursera.org/courses?query=big%20data%20analytics&page=1>
2. <https://www.edx.org/learn/big-data>
3. <https://www.tutorialspoint.com/hadoop/index.htm>

DATA VISUALIZATION

Course Code: 2504MC24

L	T	P	C
2	0	1	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Understand the various types of data, apply and evaluate the principles of data visualization
- CO2:** Acquire skills to apply visualization techniques to a problem and its associated dataset
- CO3:** Apply structured approach to create effective visualizations
- CO4:** Learn how to bring valuable insight from the massive dataset using visualization.
- CO5:** Learn how to build visualization dashboard to support decision making

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	2	3	-	3	-	-	-	2
CO2	-	3	-	3	-	-	-	-
CO3	-	3	-	3	-	-	-	-
CO4	-	3	-	3	-	-	-	3
CO5	-	3	-	3	-	2	-	3

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT – I

Introduction to Visualization: Introduction to Data Visualizations and Perception: Introduction to visualization and visual perception, visual representation of data, Gestalt principles, Information overload.

Practice:

1. Introduction to various Data Visualization tools
2. Basic Visualization in Python

UNIT – II

Visual Representations: Creating visual representations, visualization reference model, visual mapping, visual analytics, Design of visualization applications.

Practice:

1. Basic Visualization in R
2. Introduction to Tableau and Installation

UNIT – III

Classification of Visualization Systems: Classification of visualization systems, Interaction and visualization techniques misleading, Visualization of one, two and multidimensional data, text and text documents.

Practice:

1. Connecting to Data and preparing data for visualization in Tableau
2. Data Aggregation and Statistical functions in Tableau

UNIT – IV

Visualization of Groups: Visualization of groups, trees, graphs, clusters, networks, software, Metaphorical visualization. Various visualization techniques, data structures used in data visualization.

Practice:

1. Data Visualizations in Tableau

UNIT – V

Visualization of Volumetric Data and Evaluation of Visualizations: Visualization of volumetric data, vector fields, processes and simulations, Visualization of maps, geographic information, GIS systems, collaborative visualizations, evaluating visualizations

Practice:

1. Basic Dashboards in Tableau

Textbooks:

1. Interactive Data Visualization: Foundations, Techniques, and Applications. Natick, Ward, Grinstein, Keim, 2nd edition, A K Peters, Ltd. ISBN-10: 1482257995
2. Modern Data Visualization with R , Robert kabacoff, 1st Edition, O'Reilly Media, 2020 ISBN-10: 1492071143

Reference Books:

1. Visualization Analysis & Design, Tamara Munzner, 1st edition, AK Peters Visualization Series, ISBN-10: 1466508190
2. Interactive Data Visualization for the Web, Scott Murray, 2nd Edition, MGH, ISBN-10: 1491921288.

Web Links:

1. <https://elearn.nptel.ac.in/shop/iitworkshops/completed/datavisualizationwithr/?v=c86ee0d9d7ed>
2. <https://efaidnbmnnnibpcajpcgclefindmkaj/https://www.cs.put.poznan.pl/jstefanowski/sed/DM14-visualisation.pdf>
3. https://www.tutorialspoint.com/business_writing_skills/data_visualization.htmhttps://www.geeksforgeeks.org/datavisualizationanditsimportance/

GEN AI & PROMPT ENGINEERING

Course Code: 2504MC25

L	T	P	C
2	0	1	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Explain the importance of prompt engineering and differentiate between explicit, implicit, and creative prompts.
- CO2:** Apply prompt engineering techniques to improve NLP tasks, enhance creativity, and diversify AI-generated content.
- CO3:** Evaluate fine-tuning methods for language models, managing prompt complexity for optimal output.
- CO4:** Analyze data preprocessing techniques for effective language model training in generative AI.
- CO5:** Evaluate the ethical considerations and societal impacts of generative AI technologies.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	2	-	1	-	-	-	-
CO2	-	3	2	2	-	-	1	-
CO3	-	-	3	3	-	-	-	-
CO4	2	-	2	3	-	-	-	1
CO5	-	-	-	2	-	-	3	2

Mapping Course Outcomes with Program Specific Outcomes:

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

UNIT – I

Introduction to Prompt Engineering: What is prompt engineering and why it matters, Prompt types: explicit, implicit, and creative prompts, Best Practices for Crafting Effective Prompts

Practice:

1. Write a program for Prompt Clarity Evaluation.
2. Write a program for prompt Relevance Assessment
3. Write a program for Prompt Diversity Experiment.

UNIT – II

Practical Applications of Prompt Engineering: Improving NLP Tasks with Custom Prompts, Enhancing Creativity and Diversity in AIGenerated Content, Addressing AI Ethics and Bias through Thoughtful Prompt Engineering, Personalization and Adaptability in AIGenerated Content

Practice:

1. Write a program for Custom Prompt for NLP Task
2. Write a program for Enhancing Creativity in AI Generated Content
3. Write a program for Addressing Bias in AI Generated Content

UNIT – III

Advanced Prompt Engineering Techniques: Customising Prompts for Specific Use Cases, Finetuning Language Models for Optimal Output Managing Prompt Complexity and Length, Enhancing Control and Diversity in Generated Text

Practice:

1. write a program for Customizing Prompts for Specific Use Cases
2. Finetuning Language Models for Optimal Output.

UNIT – IV

Foundations of Generative AI: Neural Networks and Their Relevance in AI Language Generation, Deep Learning Architectures for Text Generation and its Implications for Prompt Engineering, Data Preprocessing Techniques for Language Models

Practice:

1. Write a program for Training a Simple Text Generation Model using LSTM
2. Data Preprocessing Techniques for Language Models.

UNIT – V

Advanced Topics in Generative AI: Adversarial attacks and defences in generative models, Ethical considerations and societal impact of generative AI, LangChain Framework.

Practice:

1. Detecting and Mitigating Adversarial Attacks.

Textbooks:

1. Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play – David Foster, 2nd Edition, O'Reilly Media, 2022, ISBN: 978-1098103242
2. Deep Learning for Natural Language Processing – Palash Goyal, Sumit Pandey, Karan Jain, 2nd Edition, Apress, 2018, ISBN: 978-1484237328

Reference Books:

1. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow” – Aurélien Géron, 2nd Edition, O'Reilly Media, 2019, ISBN: 978-1492032649
2. Transformers for Natural Language Processing – Denis Rothman, 2nd Edition, Packt Publishing, 2022, ISBN: 978-1801814973

Web Links:

1. https://onlinecourses.nptel.ac.in/noc25_cs137/preview
2. https://onlinecourses.swayam2.ac.in/imb24_mg116/preview

ETHICAL HACKING

Course Code: 2504MC26

L T P C
2 1 0 3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Understand the fundamentals of ethical hacking and networking.
- CO2:** Perform foot printing, scanning, and enumeration techniques.
- CO3:** Identify and exploit system vulnerabilities and understand malware threats.
- CO4:** Assess web application and wireless network security.
- CO5:** Utilize cryptographic techniques, ethical hacking tools, and document findings effectively.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	3	2	1	2	-	-	-
CO2	2	3	3	2	2	-	-	-
CO3	2	3	2	2	2	-	-	-
CO4	2	-	2	3	1	-	-	-
CO5	2	3	3	2	3	-	-	3

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1	2	2
CO2	2	2
CO3	2	2
CO4	2	2
CO5	2	2

UNIT – I

Introduction to Hacking: Hacking, Types and phases of hacking, Introduction to Ports & Protocols: Ports, Protocols, Primary Network Types, Virtualization & Introduction to Kali Linux: Virtualization, Virtualization software, supported platforms, Introduction to **Penetration Testing:** Penetration test, Categories and Types of Penetration tests, Structure of Penetration Test Report.

UNIT – II

Footprinting: Footprinting, Types, using ping and ns Lookup commands in Windows command line, **Scanning:** Scanning, Basics of Scanning, Basic Techniques of Scanning, Enumerating DNS using dns enum, Performing flag scan using hping3.

UNIT – III

Hacking into System: System Hacking, Password Cracking, Default password databases, Manual and Automated Password Cracking, Process of System Hacking, Using Keyloggers, Trojans & Backdoors: Trojans, Working of Trojan, Infection Techniques, Attack, Lifecycle and Classification of Virus.

UNIT – IV

Sniffing, Packet Analysis & Session Hijacking: Sniffing, Packet Analysis, Types of Sniffing, Active and Passive Sniffing Techniques, Session Hijacking, Social Engineering: Social Engineering, Process, Identity Theft, Human and Computer Based Social Engineering Techniques, Phishing Process, Types of Phishing Attacks, Social Engineering Toolkit (SET)

UNIT – V

Cryptography: Cryptography, Digital Signature, Hash Functions, Steganography: Steganography Process, watermarking, Steganography Methods and Attacks, Steganography tools, Vulnerability Assessment: Vulnerability, The Open Web Application Security Project (OWASP), Prevention, Damn Vulnerable Web Application (DVWA), installation and testing of DVWA

Textbooks:

1. Hacking: Be a Hacker with Ethics, Harsh Bothra, Khanna Publications 1st Edition ISBN: 9386173050.
2. Ethical Hacking and Penetration Testing Guide, Rafay Baloch, ISBN : 9781138436824

Reference Books:

1. Kali Linux Wireless Penetration Testing Beginner's Guide, Vivek Ramachandran, Cameron Buchanan, Packt Publishing ISBN-13. 978-1783280414.
2. SQL Injection Attacks and Defense, 1st Edition, Justin Clarke-Salt, Syngress Publication ISBN-13 :978-1597499637.
3. Mastering Modern Web Penetration Testing, Prakhar Prasad, Packt Publishing ISBN-13: 978-1785284588.

Web Links:

1. <http://www.exploit-db.com/>
2. www.geeksforgeeks.org/introduction-to-ethical-hacking
3. www.geeksforgeeks.org/ethical-hacking-tutorial

CYBER SECURITY

Course Code: 2504MC27

L	T	P	C
3	0	0	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Illustrate the basics of cybercrime, its types, legal perspectives, and global context.
- CO2:** Identify how cybercrimes occur and are planned using technology, and the basics of cyber forensics in investigating these crimes.
- CO3:** Analyse how cyber criminals use mobile and wireless devices, the security challenges they cause, and how organizations can protect these devices to prevent issues.
- CO4:** Explore how tools and methods like phishing, viruses, and attacks are used in cybercrimes, how they harm systems, and how to secure against them.
- CO5:** Explore cyber security laws, the Indian IT Act, challenges in handling cybercrimes, Digital Signature, security policies, and the need for training in information security.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	2	1	-	-	-	-	-	1
CO2	1	-	-	2	-	-	-	2
CO3	1	-	2	-	-	-	2	-
CO4	-	-	-	-	-	-	2	2
CO5	1	-	-	-	2	-	-	1

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT – I

Introduction to Cybercrime: Definition & Origin of the Word, Cybercrime and Cyber Security, who are Cybercriminals, Classification of cybercrimes, Cybercrime – Legal and Indian perspective, Cyber Crime and Indian ITA 2000, Global Perspective on cyber-crimes.

UNIT – II

Cyber Offences: Introduction, How criminals plan the attacks, Social engineering, Cyberstalking, Cybercafé and Cybercrimes, Botnets.

Cyber Forensics: Introduction, Digital Spectrum, Cyber Forensics Principles, Goals
Evolution Digital Forensics: Life cycle, Computer Forensics & Steganography, Relevance of OSI 7 Layer model to Computer Forensics, challenges

UNIT – III

Cybercrime: Mobile & Wireless Devices - Introduction, Proliferation of Mobile and Wireless Devices, Trends in Mobility, Credit Card Frauds in Mobile and Wireless Computing Era, Security Challenges Posed by Mobile Devices, Registry Settings for Mobile Devices, Authentication Service Security, Attacks on Mobile/Cell Phones.

Mobile Devices: Security Implications for Organisations, Organisational Measures for Handling Mobile Devices-Related Security Issues, Organizational Security Policies and Measures in Mobile Computing Era, Laptops

UNIT – IV

Tools and Methods used in Cyber Crime: Phishing, Password Cracking, Keyloggers, and Spywares, Virus and Worms, Trojan Horses & Backdoors, Steganography, DOS & DDOS Attacks, SQL Injection, Buffer Overflow, Attacks on Wireless Networks.

UNIT – V

Cyberspace & Cyber security Laws: Why Do We Need Cyber laws: The Indian Context, The Indian IT Act, Challenges to Indian Law and Cybercrime Scenario in India, Consequences of Not Addressing the Weakness in Information Technology Act, Digital Signatures and the Indian IT Act, Information Security Planning and Governance, Information Security Policy Standards and Practices, The information Security Blueprint, Security education, Training and awareness program.

Textbooks:

1. Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives - Nina Godbole, Sunit Belpure; 1st Edition; Wiley Publications; ISBN: 978-8126537081
2. Cyber Security and Cyber Laws - Nilakshi Jain , Ramesh Menon, 1st Edition; Wiley Publications; ISBN: 978-93-90395-75-0

Reference Books:

1. Cyber Security Essentials - James Graham, Richard Howard and Ryan Otson; CRC Press; ISBN: 978-0367335356
2. Introduction to Cyber Security - Chwan-Hwa(john) Wu,J. David Irwin; CRC PressT&F Group; ISBN: 978-0367337077

Web Links:

1. <https://nptel.ac.in/courses/106106248>
2. <https://www.coursera.org/specializations/intro-cyber-security>
3. <https://www.coursera.org/professional-certificates/google-cybersecurity>

NETWORK SECURITY AND CRYPTOGRAPHY

Course Code: 2504MC28

L	T	P	C
2	1	0	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Understand the security fundamentals and compare various symmetric encryption techniques.
- CO2:** Summarize the concepts of asymmetric encryption techniques.
- CO3:** Discuss the functioning of digital signature standards and data integrity.
- CO4:** Utilize the services provided by the PGP, S/ MIME & SSL, TLS.
- CO5:** Describe the concept of IP Security.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	2	2	1	-	-	-	-
CO2	3	2	-	1	-	-	-	-
CO3	3	2	-	1	-	-	2	-
CO4	2	-	3	2	-	-	2	-
CO5	2	-	-	-	-	-	2	2

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT – I

Basic Principles: Security Goals, Cryptographic Attacks, Services and Mechanisms, Mathematics of Cryptography.

Symmetric Encryption: Mathematics of Symmetric Key Cryptography, Introduction to Modern Symmetric Key Ciphers, Data Encryption Standard, Advanced Encryption Standard.

UNIT – II

Asymmetric Encryption: Mathematics of Asymmetric Key Cryptography- Primes, Primality Testing, Factorization, Asymmetric Key Cryptography-RSA Cryptosystem, Rabin Cryptosystem, ElGamal Cryptosystem, Elliptic Curve Cryptosystem.

UNIT – III

Cryptographic Hash Functions: Applications of Cryptographic Hash Functions, Two Simple Hash Functions Requirements and Security Hash Functions Based on Cipher Block Chaining, Secure Hash Algorithm (SHA), SHA-3.

Digital Signatures: Elgamal Digital Signature Scheme, Schnorr Digital Signature, NIST Digital Signature Algorithm.

UNIT – IV

Key Management and Distribution: Symmetric Key Distribution Using Symmetric Encryption, Symmetric Key Distribution Using Asymmetric Encryption, Distribution of Public Keys, X.509 Certificates.

User Authentication: User Authentication, Remote User-Authentication Principle, Remote User-Authentication Using Symmetric Encryption, Kerberos, Remote User-Authentication Using Asymmetric Encryption.

UNIT – V

Electronic Mail Security: Internet Mail Architecture, Email Formats, Email Threats and Comprehensive Email Security, S/MIME.

Ip Security: IP Security Policy, Encapsulating Security Payload, Combining Security Associations Internet Key Exchange.

Textbooks:

1. Cryptography and Network Security -Behrouz A Forouzan, Debdeep Mukhopadhyay ; 3rd Edition; McGraw Hill; ISBN: 978-0070702080
2. Cryptography and Network Security- William Stallings; Global Edition; 7e Pearson; ISBN: 978-0134444284

Reference Books:

1. Network Security and Cryptography - Bernard Meneges, Cengage Learning; First Edition; ISBN: 978-8131514631
2. Cryptography and Network Security – AtulKahate; Second Edition ;Tata McGraw-Hill; ISBN: 978-0070635974

Web Links:

1. <https://nptel.ac.in/courses/106105031>
2. <https://nptel.ac.in/courses/106105162>
3. <https://www.coursera.org/browse/computer-science>
4. <https://www.sciencedirect.com/science/article/pii/S1877050915019882>

BLOCK CHAIN TECHNOLOGIES

Course Code: 2504MC29

L	T	P	C
3	0	0	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Demonstrate the foundation of the Block Chain technology and understand the processes in payment and funding.
- CO2:** Choose the present landscape of Block chain implementations and understand Crypto currency markets.
- CO3:** Review of legal implications using smart contracts.
- CO4:** Understanding the Ethereum Virtual Machine (EVM).
- CO5:** Analyze security, privacy, and efficiency of a given block chain system.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	2	1	-	-	-	-	-	2
CO2	1	1	-	-	-	-	-	2
CO3	1	-	-	-	-	-	-	2
CO4	2	-	-	-	-	-	-	2
CO5	2	-	-	-	-	-	-	2

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT – I

Introduction: What is Blockchain (BC), public ledgers, BC as public ledgers; BC history - Bitcoin and Cryptocurrency, BC 2.0, Smart contracts; BC architecture – Blocks in BC, transactions and distributed consensus; BC conceptualization - The Chain and the Longest Chain, Cryptocurrency to Blockchain 2.0, Permissioned Model of Blockchain.

UNIT – II

Cryptocurrency: Cryptographic basics for cryptocurrency, A short overview of Hashing, signature schemes, Public Key Cryptography, A basic cryptocurrency.

UNIT – III

Bitcoin: Bitcoin Wallet, Blocks, Merkle Tree, hardness of mining, transaction verifiability, anonymity, forks, double spending, mathematical analysis of properties of Bitcoin. Blockchain Consensus Algorithms – Proof-of-Work, Proof-of-Stake, Delegated POS

UNIT – IV

Ethereum: Ethereum Virtual Machine (EVM), Wallets for Ethereum, Solidity, Smart Contracts, some attacks on smart contracts. Scaling of Blockchains, Future of Blockchains

UNIT – V

Trends and Topics: Zero Knowledge proofs and protocols in Blockchain, Succinct non-interactive argument for Knowledge (SNARK), pairing on Elliptic curves, Zcash.

Textbooks:

1. Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction”, Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder, 1st Edition, Princeton University Press, ISBN: 978-0691171692.
2. Blockchain: Blueprint for a New Economy, Melanie Swan, O'Reilly, ISBN: 1491920475, 9781491920473.

Reference Books:

1. Research Perspectives and Challenges for Bitcoin and Cryptocurrency, Joseph Bonneau, Andrew Miller, Jeremy Clark, Arvind Narayanan, Joshua A. Kroll, Edward W. Felten, IEEE Symposium on Security and Privacy.
2. The Bitcoin Backbone Protocol: Analysis and Applications, Juan A. Garay, Aggelos Kiayias, Nikos Leonardos, EUROCRYPT 2015, Part II, Lecture Notes in Computer Science (LNCS), Volume 9057, Springer, ISBN: 978-3662469607
3. Mastering Ethereum: Building Smart Contracts and DApps by Andreas M. Antonopoulos, ISBN-13: 978-1491971949.

Web Links:

1. <https://builtin.com/blockchain>
2. <https://www.computerworld.com/article/3191077/what-is-blockchain-the-complete-guide.html>
3. <https://blockgeeks.com/guides/what-is-blockchain-technology/>

SECURE CODING TECHNIQUES

Course Code: 2504MC30

L	T	P	C
3	0	0	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Differentiate the objectives of information security.
- CO2:** Summarize the trend, reasons and impact of the recent Cyber-attacks.
- CO3:** Describe OWASP design principles while designing a web application.
- CO4:** Explain threat modelling and importance of security in all phases of SDLC.
- CO5:** Write secure coding using some of the practices in C/C++/Java and Python programming languages.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	2	1	2	-	-	-	-	-
CO2	2	3	2	-	-	-	-	-
CO3	2	2	1	1	-	-	-	-
CO4	2	2	3	2	-	-	-	-
CO5	2	2	2	3	-	-	-	-

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT – I

Network and Information Security Fundamentals: Network Basics, Network Components, Network Types, Network Communication Types, Introduction to Networking Models, Cyber Security Objectives and Services, Other Terms of Cyber Security, Myths Around Cyber Security, Myths Around Cyber Security, Recent Cyber Attacks, Generic Conclusion about Attacks, Why and What is Cyber Security, Categories of Attack.

UNIT – II

Introduction to Cyber Security: Introduction to OWASP Top 10, A1 Injection, A1 Injection Risks Root Causes and its Mitigation, A1 Injection, A2 Broken Authentication and Session Management, A7 Cross Site Scripting XSS, A3 Sensitive Data Exposure, A5 Broken Access Control, A4 XML External Entity (XEE), A6 Security Misconfiguration, A7 Missing Function Level Access Control, A8 Cross Site Request Forgery CSRF, A8 Insecure Deserialization, A9 Using Components With Known Vulnerabilities, A10 Unvalidated Redirects and Forwards, A10 Insufficient Logging and Monitoring, Secure Coding Practices, Secure Design Principles, Threat Modeling, Microsoft SDL Tool.

UNIT – III

Secure Coding Practices and Owasp Top 10: Declarative Security, Programmatic Security, Concurrency, Configuration, Cryptography, Input and Output Sanitization, Error Handling, Input Validation, Logging and auditing, Session Management, Exception Management, Safe APIs, Type Safety, Memory Management, Tokenizing, Sandboxing, Static and dynamic testing, vulnerability scanning and penetration testing.

UNIT – IV

Secure Coding Practices in C/C++ and Java: Potential Software Risks in C/C++, Defensive coding, Preventative Planning, Clean Code, Iterative Design, Assertions, Pre Post Conditions, Low level design inspections, Unit Tests. Java- Managing Denial of Service, Securing Information, Data Integrity, Accessibility and Extensibility, Securing Objects, Serialization Security.

UNIT – V

Secure Coding in Python: Interactive Python Scripting, Python Variables, Conditionals, Loops, Functions, External Modules, File operations, Web requests.

Textbooks:

1. Networking Fundamentals - Gordon Davies, 2019 edition, Packt Publication, ISBN: 978-1789800937
2. Principles of Information Security - Michael E. Whitman and Herbert J. Mattord, 6th Edition, Course Technology, Cengage Learning publication, ISBN: 978-1337102063

Reference Books:

1. Cisslp secure software lifecycle professional all-in-one exam guide, Wm. Arthur Conklin, Daniel Paul Shoemaker, Third Edition, Released February 2022, McGraw-Hill publication, ISBN: 9781264258215
2. OCP Oracle Certified Professional Java SE 11 Programmer II Study Guide: Exam 1Z0-816 and Exam 1Z0-817 - Scott Selikoff, Jeanne Boyarsky, Sybex publication on August 6, 2020, ISBN: 978-1119588214

Web Links:

1. https://infyspringboard.onwingspan.com/en/app/toc/lex_auth_01268375129606553_63_54_shared/conte
2. https://infyspringboard.onwingspan.com/en/app/toc/lex_auth_01268375129606553_63_54_shared/conte
3. https://infyspringboard.onwingspan.com/en/viewer/html/lex_auth_0135015696571_59_6809160
4. https://infyspringboard.onwingspan.com/en/viewer/html/lex_auth_0135015689927_55_7129660
5. https://infyspringboard.onwingspan.com/en/viewer/html/lex_auth_0135015930409_77_92013093

APPLIED MODERN TECHNOLOGIES

Course Code: 2504MC31

L	T	P	C
3	0	0	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Acquire conceptual knowledge about various Information technology-related domains Along with their pros and cons.
- CO2:** Analyze numerous case studies based on current IT tools and technologies.
- CO3:** Interpret current and future career perspectives of the existing IT domain.
- CO4:** Summarize cyber security threats that organizations need to be aware of and learn about Various defense mechanisms.
- CO5:** Evaluate the innovation that emerging technologies like Blockchains can bring about.

Mapping of Course Outcomes with Program Outcomes:

CO/ PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	2	2	-	-	1	-	2
CO2	2	3	2	2	1	-	-	-
CO3	2	2	-	-	2	-	-	3
CO4	1	-	3	-	-	1	3	-
CO5	2	2	3	2	-	1	-	-

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT – I

Artificial Intelligence and Big Data: Introduction of AI & its applications, History of AI, Understanding Data, Application Domains of AI, Workflow of AI Projects, Technical Tools for AI, Evolution & Future of AICareer & Jobs in AI.

Machine Learning- Introduction to ML, Supervised & Unsupervised Learning, Overview of some ML Algorithms.

Deep Learning-Types of Neural Networks, Techniques & Applications of NLP, Overview of NLP Algorithms, Introduction to CV, Deep Learning Revolution.

Big Data-5v's of Big Data, How Big Data Works, Applications of Big Data, Introduction to Data Engineering, Difference b/w Data Analyst, Data Engineer & Data Scientist Responsibilities of Data Analyst, Data Engineer, and Data Scientist.

Big Data Frameworks-Introduction to Hadoop, Introduction to Spark, Hadoop vs. Spark

UNIT – II

Cloud Computing Fundamentals: Introduction to Cloud Computing, Definition & Characteristics, Overview of Cloud Infrastructure, I & Cloud Computing on Cloud, Cloud Computing models (IaaS, PaaS, SaaS)

Components of Cloud Computing -Virtualization & Virtual Machine, Containers, Cloud Storage, Content Delivery Networks, Hybrid Multicloud, Microservices, Serverless Computing

Cloud Security & Monitoring-What is cloud security? Identity & Access Management, Cloud Encryption, Cloud Monitoring

UNIT – III

Internet of Things: Internet of Things (IoT), Introduction of IoT, Features of IoT, Advantages and Disadvantages of IoT, Application in Business & Industry Domain, Future Scope of IoT

IoT Hardware and Software-IoT Devices, Sensors, Microcontrollers, Raspberry Pi, IoT Languages, IoT Framework

UNIT – IV

Block chain: Block chain – Introduction, what is Blockchain, Blockchain Structure, Features of Blockchain, Types of Block chain

Ethereum Block Chain-What is different about Ethereum Blockchain? Smart Contracts Ethereum Structure, Cryptocurrency

UNIT – V

Cyber security and Cloud services: Cyber Security, Types of Cyber Attack, Cyber Attack Counter Measures, Applications of Cyber Security

Cryptography-Cryptography & its Techniques, Types of Cryptography, Cryptography Algorithms Cloud Services Platforms-AWS, Google Cloud Platform (GCP)

Textbooks:

1. Artificial Intelligence with Python by Prateek Joshi, 2nd Edition , Packet Publishing, ISBN-13: 978-1839219535
2. Cyber security for Beginners by Raef Meeuwisse, 4th Edition, IT Governance Publishing, ISBN-13: 978-1780173405

Reference Books:

1. Cloud Computing: Concepts, Technology & Architecture" by Thomas Erl, Zaigham Mahmood, and Ricardo Puttini., 1st Edition, Prentice Hall, ISBN-13: 978-0133387520.
2. Block chain For Dummies by Tiana Laurence and Don Tapscott, 2nd Edition, For Dummies, ISBN-13: 978-1119555018

3. Internet of Things for Architects: Architecting IoT solutions by implementing sensors, communication infrastructure, edge computing, analytics, and security" by Perry Lea, 1st Edition, Packet Publishing, ISBN-13: 978-1788470599.
4. Big Data: A Beginner's Guide To Using Data Science For Business by David Feldspar, 1st Edition , Independently published, ISBN-13: 978-1520719925.

Web Links:

1. <https://archive.nptel.ac.in/courses/106/104/106104189/>
2. https://onlinecourses.nptel.ac.in/noc22_cs53/preview

WEB DESIGN AND DEVELOPMENT

Course Code: 2504MC32

L	T	P	C
2	0	1	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Demonstrate a foundational understanding of web development principles
- CO2:** Master core web development tools and resources and apply best practices for user-friendly web design
- CO3:** Craft accessible & responsive websites using HTML & CSS
- CO4:** Leverage JavaScript for basic interactivity
- CO5:** Collaborate effectively and manage code versions

Mapping of Course Outcomes with Program Outcomes:

CO/ PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	-	-	2	-	-	-	2
CO2	2	2	1	3	-	-	-	2
CO3	2	2	-	3	-	-	-	-
CO4	1	2	-	3	-	-	-	2
CO5	-	-	-	-	3	2	1	2

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT – I

Introduction to Web Design: Introduction to Web and Web Development, Evolution of the web, Different types of websites and their purposes, What is web designing?, Careers in web development, Impact of the web on society and communication, Client-side vs. server-side development, Web development process: planning, coding, testing, and deployment

Web Designing Tools and Resources-Popular web browsers and their developer tools, Online code editors and development environments, Web design resources and tutorials, Identifying website goals and target audience

Needs Analysis and UI/UX Considerations-User interface (UI) and user experience (UX) considerations, Sketching basic website layouts and wireframes

Practice:

1. Create a sitemap outlining the website's structure and form the content for each page (e.g., About Me, Skills, Projects, Contact), Create a GitHub repository for your project and ensure continuous commit for this project

2. Sketch a basic wireframe for each page, visualizing the layout and placement of content elements.
3. Install tools like a code editor (e.g., VS Code) and browser developer tools (e.g., Chrome DevTools) and create a Basic HTML structure (website's main structure using HTML elements like <header>, <nav>, <section>, And<footer>)

UNIT – II

Core Concepts of HTML: Introduction to HTML-What is HTML and its role in web development, Basic HTML structure: elements, attributes, tags, Importance of semantic HTML for website meaning and accessibility

Common HTML Elements-I-Headings, paragraphs, and lists for structuring content, Images and multimedia elements for visual appeal, Links and anchors for navigation within and between web pages, Tables for data presentation (with a note on responsible use of tables for layout), Forms for user input and data collection, Basic form elements: text fields, radio buttons, checkboxes

Creating Semantic HTML Documents-Importance of using proper HTML tags to convey meaning, Semantic elements for headings, navigation, sections, articles, aside, and more, Building a well-structured and accessible website foundation

Organizing Content with HTML-Nesting of HTML elements to create complex website structures, Using HTML attributes to provide additional information and functionality, creating reusable HTML components for efficient development

Introduction to HTML Validation-Importance of validating HTML code for errors and best practices, using online validators and developer tools for code-checking, maintaining clean and well-formatted HTML

Building Interactive Forms-Form validation techniques to ensure user input accuracy, sending form data using different methods (GET vs. POST)), Introduction to form handling on the server side

Practice:

1. Adding Content with Semantic HTML: Focus on using semantic HTML elements like <h1> for headings, <p> for paragraphs, and for lists to improve website meaning and accessibility.
2. Integrating Images and Links: Incorporate relevant images and links to other web pages or sections within the Website.

UNIT – III

Styling with CSS: Introduction to CSS: What is CSS and its role in web design, Selectors in CSS: targeting elements for styling, Basic CSS properties and values for formatting text and elements

CSS Selectors-I&II-Targeting elements with IDs, classes, and element types, combining selectors for precise control over styles, Pseudo-classes and pseudo-elements for dynamic styling, Descendant selectors, child selectors, and universal selectors, grouping selectors and nesting for efficient CSS code, Specificity in CSS and resolving conflicts

Formatting Text with CSS-Font properties for controlling font family, size, color, and

style, Text alignment, justification, and decoration for visual appeal, Line height and letter spacing for readability

Box Model and Layout in CSS-The CSS box model: understanding content, padding, border, and margin, Layout techniques using CSS: float, clear, and positioning

Backgrounds and Images in CSS-Setting background colors and images for web pages, Background properties: position, size, repeat, and attachment, Using images effectively: optimization, accessibility considerations

Borders and Outlines in CSS-Styling borders with width, style, and color properties, creating decorative borders and outlines for elements, Using borders for visual separation and grouping

Accessibility Considerations-Importance of building accessible websites using semantic HTML and ARIA attributes, Implementing alternative text for images using the 'alt' attribute., Color contrast and proper heading structure for screen readers.

Practice:

1. Building Forms for Contact: Create a basic contact form with input fields like name, email, and message, Ensuring proper form structure.
2. Design an HTML Learning CSS Selectors: Use CSS selectors to target specific elements within their HTML Document for styling.

UNIT – IV

Advanced CSS: CSS Animations and Transitions: Exploring properties for text shadows, gradients, etc., CSS animations and transitions, creating simple animations for hover effects, element appearances, or transitions, Exploring animation properties like 'animation-name', 'duration', and 'timing-function'

CSS Frameworks -CSS frameworks like Bootstrap and Foundation, Advantages of using CSS frameworks, Exploring the basic structure and components of a chosen framework (e.g., grid system, navigation bars, buttons), When to consider using a framework vs. writing custom CSS

Advanced CSS Layouts-Introduction to Grid Layout for a more structured approach to layout, Basic Grid concepts: rows, columns, gaps, and grid placement, Introduction to Flex box for flexible layout and responsive design, exploring basic Flex box properties (display: flex, justify-content, align-items)

Organizing CSS Styles -Internal vs. external style sheets for managing CSS code, Best practices for writing clean, maintainable, and reusable CSS, Using CSS pre-processors like SASS or LESS

Responsive Web Design Fundamentals-Responsive design considerations for different screen sizes, using media queries to adapt layouts for different screen sizes, Techniques for creating responsive navigation, images, and content, Advanced Media Query features: device targeting (e.g., (max-width: 768px)), Media query orientation queries (e.g., (orientation: portrait)) and nesting for complex layouts.

Building a Basic Website Layout with CSS-Applying CSS concepts to create a simple website layout, combining elements and styles for a visually appealing design, Understanding the separation of concerns between HTML and CSS

Practice:

1. Formatting Text and Layouts: Using CS, control text styles (font-family, size, color), margins, padding, and basic layouts using floats.
2. Adding Backgrounds and Images: Use CSS to set background colors and images, style images, and understand Responsive design considerations.
3. Refining the Design: Customize the website's visual appeal by applying color schemes, typography choices, and basic CSS animations

UNIT – V

Introduction to JavaScript, Version Control, and Testing: What is JavaScript and Why Use It? Introduction to JavaScript as a scripting language for web interactivity, Adding dynamic behavior to web pages with JavaScript

Core JavaScript Concepts-Basic JavaScript syntax: variables, data types, operators, Control flow statements: if-else, loops (for, while), Introduction to functions for code reusability **Event Handling with JavaScript**-Responding to user interactions like clicks, scrolls, and form submissions, Event listeners and event handlers for dynamic behavior, Simple examples of event handling in practice

DOM Manipulation with JavaScript-Introduction to the Document Object Model (DOM), Accessing and modifying HTML elements using JavaScript, Adding, removing, and modifying content dynamically.

Introduction to Version Control (Git)-Importance of Git version control for collaborative

development, Basic Git concepts: repositories, commits, branches, and merging, Utilizing a Git client or online platform to manage code versions.

Testing and Debugging Techniques-Importance of testing websites across different browsers and devices for compatibility., Using browser developer tools to inspect and debug web pages., Common CSS debugging techniques for layout and style issues

Practice:

1. Ensuring Responsiveness: Introduce basic media queries in CSS to adapt the websitelayout for different screen sizes (mobile-friendly design).
2. Validation and Testing: Use online validators to check their HTML and CSS code for errors and test the website functionality across different browsers.
3. Deployment: Explore hosting the website online using free platforms such as GitHub Pages

Textbooks:

1. Head First HTML & CSS by Elisabeth Robson & Eric Freeman; 2nd Edition, Shroff/O'Reilly, ISBN-13: 978-9350238890
2. Web Design for Developers: A Programmer's Guide to Design Tools and Techniques by Brian Hogan; 1st Edition, Publisher: O'Reilly, ISBN-13: 978-1934356135.

Reference Books:

1. Responsive Web Design by Ethan Marcotte; 2nd Edition, Publisher: A Book Apart, ISBN-13: 978-1937557188
2. CSS–The Definitive Guide by Eric Meyer, Estelle Weyl; 4th Edition, O'Reilly, ISBN-13: 978-1449393199.
3. The Web Developer's Guide to Completing Projects by Ashley Nolan & Lewis Powell. 1st Edition, Site Point, ISBN-13: 978-0995466317.
4. Web Design with HTML, CSS, JavaScript, and jQuery set, by Jon Duckett; Packt Edition, Wiley, ISBN-13: 978-1118907757.

Web Links:

1. <https://www.shiksha.com/online-courses/web-development-courses-certification-training-by-nptel-st644>
2. <https://www.upwork.com/resources/web-design-vs-web-development>

ADVANCED JAVASCRIPT

Course Code: 2504MC33

L	T	P	C
2	0	1	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Analyze the fundamental concepts and syntax of JavaScript.
- CO2:** Apply JavaScript programming techniques to solve real-world problems.
- CO3:** Evaluate different JavaScript data types and their appropriate usage.
- CO4:** Create interactive web applications using JavaScript and DOM manipulation.
- CO5:** Synthesize advanced JavaScript concepts and techniques to develop robust applications.

Mapping of Course Outcomes with Program Outcomes:

CO/ PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	-	1	2	-	-	-	-
CO2	2	3	2	2	1	-	-	-
CO3	3	1	-	2	-	-	-	-
CO4	1	2	3	3	2	-	-	-
CO5	-	2	3	2	2	-	-	3

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT – I

JavaScript Fundamentals: Introduction to JavaScript-Brief history and overview of JavaScript, JavaScript in web development, Basic syntax and structure of a JavaScript program. **Lexical Structure**-Character set and Unicode in JavaScript, Tokens and keywords, Identifiers and variable.

Types, Values, and Variables-Primitive types: number, string, Boolean, null, undefined, symbol, Object types: object, function, array, date, regex, error, Type coercion and conversion

Expressions and Operators-Arithmetic, comparison, logical, bitwise, and assignment operators, Operator precedence and associativity, Expressions and their evaluation.

Statements-Control flow statements: if/else, switch, while, do/while, for, break, continue, Exception handling with try/catch/finally.

Practice:

1. Set up project structure and files.
2. Design the basic HTML structure for the weather application.

UNIT – II

JS Objects, Arrays, Classes and DOM: Functions-Defining and calling functions, Function parameters and arguments, Function scope and closures

Objects-Object literals and properties, Creating objects with constructors and classes, Object methods and this keyword

Arrays-Creating and accessing arrays, Array methods for manipulation and iteration, Arrays and objects

Classes and Modules-Static and instance members, Inheritance and polymorphism, ES6 modules and import/export statements

Regular Expressions-Syntax and basic patterns, Metacharacters and quantifiers, Capturing and non-capturing groups

JavaScript Subsets and Extensions-ECMAScript and its versions, Browser-specific extensions, Server-side JavaScript and Node.js

The Document Object Model-The structure and hierarchy of the DOM, Accessing and manipulating DOM elements, Events and event handling

Practice:

1. Style the HTML elements using CSS for a visually appealing layout.
2. Create an input field where users can enter the name of a city.
3. Implement a JavaScript function to fetch current weather data for the entered city.
4. Style the current weather display section for better presentation.

UNIT – III

JS Scripting & npm: Scripting Web Pages: Embedding JavaScript code in HTML pages, Loading external scripts, best practices for web page scripting. **Handling Events**-Event types and propagation, Event listeners and handlers, using event objects and event delegation **Scripting CSS**-Dynamic CSS manipulation with JavaScript, CSS selectors and rules **Handling XML**-The structure and syntax of XML, Parsing and generating XML documents, XML and JSON comparison.

Scripting HTTP-Overview of the HTTP protocol and its components, HTTP requests and responses, Using the fetch API to make HTTP requests

Node and npm-Installing and using Node.js, Running JavaScript code outside the browser, Node.js modules and packages

Practice:

1. Design an HTML section for displaying the 3-day weather forecast.
2. Implement JavaScript function to generate simulated 3-day weather forecast data for entered the city.
3. Implement geolocation functionality to detect the user's current location (using simulated data).

UNIT – IV

Server-Side JS: Server: Side JavaScript with Node-Building a web server with Node.js, Handling HTTP requests and responses, Node.js frameworks for web development **Modules and Packages**-The module system in Node.js and ES6, Importing and exporting modules, Creating and publishing packages with npm

File system I/O-Reading and writing files with Node.js, Using the fs module for file system operations, Asynchronous vs. synchronous I/O

Networking-Network protocols and communication, TCP and UDP sockets, WebSocket and real-time communication **Child Processes**-Spawning and managing child processes with Node.js, Inter-process communication and messaging, Forking and clustering for parallel processing. **Tools and Techniques**-Debugging and profiling JavaScript code, Build tools and automation with Grunt and Gulp, Code quality and linting with ESLint

Practice:

1. Fetch weather data based on the user's geolocation (using simulated data) and display the simulated weather Information for the user's current location.
2. Implement basic error handling for invalid city inputs and geolocation errors.
3. Implement a feature to display weather icons corresponding to the current weather conditions (e.g., sunny, Cloudy, rainy).

UNIT – V

Client Side JS and Advanced JS: Client-Side JavaScript Frameworks-Overview of popular client-side frameworks, React, Angular, and Vue.js, Single-page applications and routing

The jQuery Library-Introduction to jQuery and its features, Selectors and DOM manipulation with jQuery, Event handling and animation with jQuery

Ajax and JSON-Asynchronous requests with XMLHttpRequest, JSON data format and parsing, Cross-origin resource sharing (CORS), Scripting Mobile Apps

Web Assembly and WebGL-Introduction to Web Assembly and its benefits, Running C/C++ and other compiled code in the browser, WebGL for high-performance graphics and visualization

Advanced Topics-Promises and async/await, Generators and iterators, Web workers and offloading tasks

Practice:

1. Add additional features such as a toggle switch for temperature units (Celsius / Fahrenheit) or a button to refresh weather data.
2. Integrate a feature to allow users to save their favorite cities and quickly access weather information for those cities without re-entering the names.
3. Ensure basic responsiveness for different screen sizes using CSS media queries.

Textbooks:

1. JavaScript: The Definitive Guide by David Flanagan, 7th Edition, O'Reilly Media, ISBN-13: 978-0596000487
2. Eloquent JavaScript: A Modern Introduction to Programming by Marijn Haverbeke, 3rd Edition, No Starch Press, ISBN-13: 978-1593279509.

Reference Books:

1. JavaScript: The Good Parts by Douglas Crockford, 1st Edition, O'Reilly Media, ISBN-13: 978-0596517748
2. You Don't Know JS: Up & Going by Kyle Simpson, 1st Edition, O'Reilly Media, ISBN-13: 978-1491924464
3. JavaScript: The Complete Reference by Thomas A. Powell, 3rd Edition, McGraw-Hill Education, ISBN-13: 978-1260454832
4. Head First JavaScript Programming by Eric Freeman and Elisabeth Robson, 1st Edition, O'Reilly Media, ISBN-13: 978-1491904886.

Web Links:

1. <https://www.geeksforgeeks.org/advanced-javascript-backend-basics/>
2. <https://www.w3schools.com/js/>

CLIENT-SIDE SCRIPTING

Course Code: 2504MC34

L	T	P	C
2	0	1	3

Course Outcomes: At the end of the Course, Student will be able to:

- CO1:** Create accessible and semantically structured web pages using HTML and ARIA attributes.
- CO2:** Implement responsive layouts and design using CSS, including flexbox and positioning techniques.
- CO3:** Develop interactive user interfaces and manage state using React components, props, and state.
- CO4:** Validate and test React applications, including form validation, unit testing, and end-to-end testing.
- CO5:** Apply advanced concepts in React, such as server-side rendering (SSR) and state management with Redux.

Mapping of Course Outcomes with Program Outcomes:

CO/ PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	2	1	-	3	-	-	-	2
CO2	3	2	2	3	-	-	-	-
CO3	2	3	2	3	2	-	-	1
CO4	1	2	3	3	1	-	2	-
CO5	3	2	3	2	-	2	-	3

Mapping of Course Outcomes with Program Specific Outcomes:

CO/ PSO	PSO1	PSO2
CO1	3	-
CO2	3	-
CO3	3	2
CO4	-	3
CO5	2	3

UNIT – I

Lexical Analysis: HTML Accessibility, Semantic Markup for accessibility, Alternative Text for images, ARIA (Accessible Rich Internet Applications) Attributes, Keyboard Accessibility and Focus Elements

HTML Multimedia-Embedding images, videos, and audio, Video and audio controls, HTML5 Canvas for drawing and animations, Working with SVG (Scalable Vector Graphics) **HTML APIs:** Geolocation API, Local storage and session storage, Web workers for multithreading, Drag and drop API

CSS Flexbox: Flex container and flex items, Flex directions and alignment, Flexbox properties: flex-grow, flex-shrink, flex-basis, Flexbox ordering and wrapping, Nested flexbox layouts, Responsive design with Flexbox. **Lecture 3a:** CSS Layout, Box model, Positioning elements with CSS, Display properties, CSS units of measurement

CSS Performance Optimization-CSS minification and compression techniques, CSS sprites and image optimization, Critical path CSS optimization, CSS caching and

performance tools, CSS performance profiling and debugging

JavaScript Asynchronous Programming-Understanding asynchronous programming, Callback functions and their usage, Error handling with callbacks (e.g.try/catch),Callback hell and how to mitigate it, Promises and their syntax, Chaining promises for asynchronous operations.

Practice:

1. Enhance the accessibility of a movie app webpage, Embed movie trailers or clips.
2. Use the Geolocation API to fetch the user's location and display nearby movie theaters or Cinemas on a movie app webpage, Design a movie app search results page that display Movie cards in a flexbox layout with details such as movie poster, title, and rating.
3. Design the layout of the movie list & booking form, Optimize the performance of a movie app.

UNIT – II

React: Introduction to React, React Components, JSX Syntax, Props and State, React Lifecycle Methods

React Hooks-Use State Hook, use Effect Hook, use Context Hook, use Reducer Hook,Custom Hooks.

Practice:

1. Fetch Movie Data from API.
2. Create a React component for displaying movie cards.
3. Replace class components with functional components and implement use State to Manage movie data and selected movie.

UNIT – III

React Validation & Testing: React Forms and Validation, Controlled Components, Uncontrolled Components, Form Validation, Formik Library, Yup Validation

React Testing-Unit Testing with Jest, Testing React Components, Mocking APIs, Testing Redux, End-to-End Testing with Cypress.

Practice:

1. Create a form for adding new movies to the movie app, including fields for title, poster URL, release date, and genre.
2. Write unit tests for a movie app component.
3. Implement Redux store and actions for managing movie data in a movie app.

UNIT – IV

Rendering with React + Angular Basics: State Management with Redux, Redux Basics, Redux Middleware, Redux Saga, Reselect Library

Server-side Rendering (SSR) with React-SSR Basics, Next.js Framework, Server-side Data Fetching, SEO Optimization

Introduction to angular-Setting up an Angular development environment, Components, directives, and templates in Angular, Data binding in Angular, Angular

modules and services **Angular Forms**-Form controls and validation, Form submission and handling, Form validation using built-in and custom validators, Dynamic forms and form arrays

Practice:

1. Implement server-side rendering using Next.js in a movie app.
2. Creating Angular Components for Movie App.
3. Creating Movie Search Form.

UNIT – V

Advance Angular concepts: Angular Routing and Navigation, Angular routing concepts, Configuring routes in Angular, Navigation and route guards, Lazy loading and dynamic routing.

Angular Components and Directives-Creating custom Angular components, Using Angular built-in components, Component communication and lifecycle hooks, Creating custom Angular directives

Angular Pipes and Filters-Built-in pipes in Angular, Creating custom pipes, Using filters to format data

Angular HTTP and API Integration-Making HTTP requests in Angular, Handling HTTP responses and errors, Interceptors and request caching, Integration with RESTful APIs.

Practice:

1. Implementing Movie Details Page.
2. Creating Movie Card Component, Implementing Movie Search Filter.
3. Adding Movie CRUD Functionality.

Textbooks:

1. Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics by Jennifer Niederst Robbins, 5th Edition, O'Reilly Media, ISBN-13: 978-1491960201.
2. React: Up & Running: Building Web Applications by Stoyan Stefanov and Kiran Prasad, 2nd Edition, O'Reilly Media, ISBN-13: 978-1492051721.

Reference Books:

1. JavaScript: The Definitive Guide by David Flanagan, 7th Edition, O'Reilly Media, ISBN-13: 978-0596000487.
2. CSS Mastery: Advanced Web Standards Solutions by Andy Budd, 3rd Edition, A Book Apart, ISBN-13: 978-1937557423.

Web Links:

1. <https://docs.servicenow.com/bundle/washingtondc-application-development/page/script/client-scripts/concept/client-side-scripting-overview.html>
2. <https://developer.mozilla.org/en-US/docs/Learn/JavaScript>

SERVER-SIDE SCRIPTING

Course Code: 2504MC35

L	T	P	C
2	0	1	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Apply Node.js fundamentals to create basic web applications and APIs.
- CO2:** Design, develop and deploy scalable backend systems using Node.js
- CO3:** Analyze and implement advanced Node.js concepts such as real-time applications, testing and debugging, deployment, and security
- CO4:** Evaluate the benefits of microservices architecture and apply them to build and manage microservices in Node.js
- CO5:** Integrate Node.js with other server-side technologies to develop efficient and scalable web applications

Mapping of Course Outcomes with Program Outcomes:

CO/ PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	2	1	-	3	-	-	2	-
CO2	3	2	3	2	1	2	-	-
CO3	2	3	3	3	-	-	2	3
CO4	1	2	2	2	3	3	-	1
CO5	3	2	3	3	2	-	-	-

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT – I

Node.js Fundamentals: Introduction to Node.js, Understanding the basics of Node.js, Installing Node.js and npm, Building a Hello World application in Node.js

Node.js Core Modules-Exploring built-in modules in Node.js, Using fs module for file system operations, working with http module for creating a web server, Utilizing path module for working with file paths

Asynchronous Programming in Node.js-Understanding the concept of asynchronous programming, Using callbacks, promises, and async/await in Node.js, Handling errors in asynchronous code, Utilizing the event loop and understanding event-driven architecture in Node.js.

Practice:

1. Build a Hello World application that greets the user with their name.
2. Create a module that fetches movie data from an API and returns it to the application.
3. Fetch Movie data and display it on the console.

UNIT – II

Express.js & Passport.js: Express.js, Introduction to Express.js, a popular Node.js web framework, Building Restful APIs with Express.js, Working with middleware in Express.js for request handling, Implementing routing and handling HTTP requests in Express.js

Data Persistence and Database Integration-Connecting Node.js to databases like MongoDB, MySQL, or PostgreSQL-Using popular Node.js database libraries like Mongoose, Sequelize, or Knex-Performing CRUD (Create, Read, Update, Delete) operations with databases in Node.js

Authentication and Authorization-Implementing authentication and authorization in Node.js applications, Using popular authentication frameworks like Passport or JWT, Implementing authentication strategies like OAuth or OAuth2, Securing routes and resources in Node.js applications.

Practice:

1. Create a RESTful API that returns a list of movies.
2. Create a schema for movie data and interact with the database.
3. Create a login and registration page for users to access the app.

UNIT – III

Advance Node.js Concepts: Real time Applications with Socket.io, Introduction to Socket.io, a popular library for real-time communication, Building real-time applications like chat applications using Socket.io, Error handling with callbacks (e.g. try/catch), Implementing bi-directional communication between server and clients, Handling events and managing connections in real-time applications

Testing and Debugging in Node.js-Writing unit tests for Node.js applications using frameworks like Mocha or Jest, Using debuggers and logging for troubleshooting in Node.js, Applying best practices for error handling and debugging in Node.js applications

Deployment and Performance Optimization-Deploying Node.js applications to production environments, Optimizing performance and scalability of Node.js applications, Using performance monitoring tools and techniques for Node.js applications, Implementing caching, load balancing, and other performance optimization strategies in Node.js applications

Best Practices and Security in Node.js-Following best practices for writing clean and maintainable Node.js code, Understanding security vulnerabilities and best practices in Node.js applications, Implementing security measures like input validation, output encoding, and authentication, Protecting against common security threats like cross-site scripting (XSS) or cross-site request forgery (CSRF).

Practice:

1. Create a real-time chat application in the movie app.
2. Write unit tests for the movie app.
3. Deploy the movie app to a cloud service like Heroku or AWS.

UNIT – IV

Micro services and Error Handling: Building Micro services with Node.js-Understanding Micro services architecture and its benefits, Building micro services using Node.js and frameworks like Express, Hapi, or Fastify, Implementing communication and coordination between microservices in Node.js, Deploying and managing microservices with Docker and Kubernetes. **Error Monitoring and Logging in Node.js Applications**-Implementing error monitoring and logging with tools like Sentry, Loggly, or ELK stack, Centralized logging and error tracking for Node.js applications, Analyzing and troubleshooting errors in Node.js applications using log data, Best practices for error monitoring and logging in Node.js applications

Continuous Integration and Deployment (CI/CD) in Node.js-Understanding the importance of CI/CD in modern software development, Setting up automated build and deployment pipelines for Node.js applications, Configuring popular CI/CD tools (e.g., Jenkins, Travis CI, CircleCI) for Node.js projects, Best practices for continuous integration, automated testing, and deployment in Node.js applications.

Practice:

1. Implement security measures like input validation, output encoding, and authentication.
2. Building a Microservices Architecture with Node.js.
3. Implementing Error Monitoring and Logging in a Node.js Application.

UNIT – V

Basics of Node.js Server Integration: Building Scalable APIs with GraphQL and Node.js, Advanced GraphQL concepts like subscriptions, directives, and fragments, Building high-performance APIs with GraphQL and Node.js, Implementing pagination, batching, and caching in GraphQL APIs, Securing and testing GraphQL APIs in Node.js applications.

Serverless Computing with Node.js- Introduction to serverless computing and its advantages, Building server less applications with Node.js using serverless frameworks like AWS Lambda, Azure Functions, or Google Cloud Functions, Implementing serverless event-driven architectures with Node.js, Serverless deployment and monitoring best practices with Node.js.

Practice:

1. Implementing CI/CD for a Node.js Application.
2. Creating Angular Components for Movie App.
3. Creating Movie Search Form.

Textbooks:

1. Node.js Design Patterns by Mario Casciaro and Luciano Mammino, 2nd Edition, Packet Publishing, ISBN-13: 978-1801077447.
2. Hands-On Microservices with Node.js by Diogo Resende, 1st Edition, Packet Publishing, ISBN-13: 978-1801070806.

Reference Books:

1. Node.js in Action by Mike Cantelon, Marc Harter, T.J. Holowaychuk, and Nathan Rajlich, 2nd Edition, Manning Publications, ISBN-13: 978-1617294579. .
2. Mastering Node.js by Sandro Pasquali, 1st Edition, Packt Publishing, ISBN-13: 978-1782165238 .
3. Express.js Guide: The Comprehensive Book on Express.js by Azat Mardan, 1st Edition, Azat Mardan, ISBN-13: 978-1505778282.

Web Links:

1. <https://www.upwork.com/resources/server-side-scripting-back-end-web-development-technology>
2. https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Introduction

CLOUD ESSENTIALS

Course Code: 2504MC36

L	T	P	C
2	0	1	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Explain the fundamentals of Cloud Storage, Cloud Architecture, and Cloud Computing.
- CO2:** Explain Cloud Computing technologies concerning platforms, services, networks, security, and applications.
- CO3:** Build Cloud platforms using Google, Amazon, and Microsoft services.
- CO4:** Examine Cloud services and applications using Webmail, Media, and Streaming.
- CO5:** Experiment with Cloud-based solutions for individuals and enterprises using Google and Microsoft cloud offerings.

Mapping of Course Outcomes with Program Outcomes:

CO/ PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	2	1	-	-	-	-	1
CO2	2	3	2	-	-	-	1	1
CO3	3	2	3	2	-	1	-	-
CO4	2	3	1	2	-	-	2	-
CO5	1	-	3	3	1	2	-	-

Mapping of Course Outcomes with Program Specific Outcomes:

CO/ PSO	PSO1	PSO2
CO1	3	-
CO2	3	-
CO3	3	2
CO4	-	3
CO5	2	2

UNIT – I

Cloud Computing Fundamentals: Introduction to Cloud Computing – I, History of cloud computing, Defining clouds, Cloud providers. **Introduction to Cloud Computing II-**Consuming cloud services, Cloud architecture, Cloud storage. **Why Cloud Computing?** -Why cloud computing matters, Advantages and disadvantages of cloud computing.

Practice:

1. Configure and manage cloud infrastructure using a popular cloud provider such as Amazon Web Services (AWS), Microsoft Azure, or Google Cloud Platform.
2. Learn to deploy and configure cloud services such as virtual machines, databases, and Web applications using a cloud provider such as AWS or Microsoft Azure to create a Scalable and efficient cloud infrastructure.
3. Analyze the benefits of cloud computing, using real-life examples such as the adoption of cloud computing by Netflix and Uber.

UNIT – II

Cloud Types and Models: Types of Cloud, Types of Cloud, Architecture of Cloud, Cloud services. **Cloud Models-Iaas, Paas, SaaS.**

Practice:

1. Use real-life examples such as the use of public cloud by Airbnb and private cloud by Walmart, students will learn about the different types of cloud environments including Public, private, and hybrid.
2. Use real-life use cases such as the adoption of SaaS by Salesforce and PaaS by Google App Engine, students will learn about the different cloud models including IaaS, PaaS, and SaaS.
3. Learn to use technical interfaces such as APIs and SDKs to interact with cloud Services. Use a cloud provider such as AWS or Google Cloud Platform to develop and deploy applications.

UNIT – III

Inside the Cloud: Inside the Cloud-Administering cloud services, Technical interface, Cloudservices. **Web-based Application-Web-based Application, Pros and cons of cloud servicedevelopment. Hardware and Infrastructure:** Clients, Security, Network, Services.

Practice:

1. Using a real-life use case such as the development of web-based applications by Instagram, students will learn to deploy and configure web-based applications in a Cloud environment.
2. This lab will focus on the hardware and infrastructure required to support cloud Computing, using real-life examples such as the use of containerization by Docker and Kubernetes. Students will learn to deploy and manage hardware and infrastructure in a Cloud environment.
3. Using a real-life use case such as the integration of web APIs by Uber, students will Learn to use web APIs to integrate cloud services and develop applications.

UNIT – IV

Accessing Cloud and Managing Data: Accessing the Cloud, Platforms, Web applications, Web API, Cloud storage.

Managing Data on Clouds-"Nature of cloud, Traditional data center, Cost of cloud datacenter", scaling computer systems, Cloud workload, public, private and hybrid clouds.

Practice:

1. Using real-life examples such as the use of cloud storage by Drop box and the use of Cloud databases by Airbnb, students will learn to manage data on the cloud including Storage, backup, and recovery. They will use a cloud provider such as AWS or Microsoft Azure to configure and manage data on the cloud.
2. Learn the fundamentals of Service-Oriented Architecture (SOA) and its principles. Develop and deploy web services using popular tools such as Apache Axis2 or Apache CXF and integrate them into a service-oriented architecture.

3. Learn to configure and manage cloud-based storage solutions such as Amazon S3, Microsoft Azure Blob Storage, or Google Cloud Storage. Use these cloud services to store, backup, and recover data in a cloud environment

UNIT – V

Using Cloud: Using Cloud Platforms – I, Understanding service-oriented architecture, Moving applications to the cloud. **Using Cloud Platforms II**-Working with cloud-based storage, working with productive software

Cloud Elements: Infrastructure as a service, Platform as a service, Software as a service

Thin Clients-Google app engine, Microsoft Windows Azure

Migration-Virtualizing your organization, Server solutions.

Practice:

1. Deploy a web application on a PaaS provider like Heroku or Google App Engine
2. Deploy and configure thin client devices using a cloud provider such as AWS Work Spaces or Citrix Virtual Apps and Desktops. Learn how to manage user access, performance, and Data security in a thin client environment.
3. Use popular migration tools such as AWS Migration Hub or Microsoft Azure Migrate to Migrate on-premises applications and data to the cloud.

Textbooks:

1. Anthony Velte, Toby Velte, and Robert Elsenpeter, “Cloud Computing – A Practical Approach”, 1st Edition, McGraw Hill, ISBN-13: 978-0071626949.
2. Rajkumar Buyya and Vecchiola, Selvi, “Mastering Cloud Computing”, 1st Edition, McGraw Hill Education, ISBN-13: 978-0073376229.

Reference Books:

1. Barrie Sosinsky, “Cloud Computing Bible”, 1st Edition, John Wiley & Sons, ISBN-13: 978-0470903568.
2. Massimo Cafaro and Giovanni Aloisio, “Grids, Clouds and Virtualization”, 1st Edition Springer, ISBN-13: 978-3642177556.
3. Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, “Cloud Computing: Principles and Paradigms”, 1st Edition, Wiley Publications, ISBN-13: 978-0470887990.

Web Links:

1. https://onlinecourses.nptel.ac.in/noc21_cs14/preview
2. <https://www.shiksha.com/online-courses/cloud-computing-basics-by-nptel-course-nptel25>

CLOUD ARCHITECTURE, SERVICES AND STORAGE

Course Code: 2504MC37

L	T	P	C
2	0	1	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Develop an understanding of cloud infrastructure and storage fundamentals, including virtualization, deployment models and service models.
- CO2:** Develop knowledge of cloud services, including platform and middleware, software, and applications, as well as cloud management and security.
- CO3:** Gain familiarity with different cloud storage options and their benefits, including encryption, backup, and recovery.
- CO4:** Analyze the potential advantages and challenges of adopting cloud computing and storage for businesses and organizations.
- CO5:** Design and implement cloud solutions using platform and middleware services, software, and applications.

Mapping of Course Outcomes with Program Outcomes:

CO/ PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	-	2	2	-	-	-	1
CO2	2	2	3	-	-	-	1	2
CO3	2	3	2	2	-	-	-	1
CO4	1	2	3	1	3	-	-	-
CO5	3	1	2	3	-	2	-	-

Mapping of Course Outcomes with Program Specific Outcomes:

CO/ PSO	PSO1	PSO2
CO1	3	-
CO2	3	-
CO3	-	3
CO4	-	3
CO5	2	2

UNIT – I

Cloud Computing and Architecture: Introduction to Cloud Computing, Definition of cloud computing, Benefits and drawbacks of cloud computing, Evolution of cloud computing. **Cloud Architecture**-Overview of cloud architecture, Types of cloud services (SaaS, PaaS, IaaS) Virtualization and containerization.

Practice:

1. Research and present the history and evolution of cloud computing, starting from the Early days of mainframe computing to modern cloud technologies.
2. Design and implement cloud architecture for a hypothetical company, including Computing, storage, networking, and security components.
3. Create and deploy applications using each of the three main cloud service models (IaaS, PaaS, SaaS), comparing and contrasting their strengths and limitations.

UNIT – II

Cloud Service and Deployment Models: Cloud Service Models, Explanation of Infrastructure as a Service (IaaS), Explanation of Platform as a Service (PaaS), Explanation of Software as a Service (SaaS), Comparison of IaaS, PaaS, and SaaS. **Cloud Deployment Models**-Overview of cloud deployment models, Advantages and disadvantages of each model, Hybrid cloud solutions.

Practice:

1. Deploy and manage cloud resources using the four main cloud deployment models (Public, private, hybrid, multi-cloud), evaluating the pros and cons of each.
2. Compare and contrast the offerings and pricing of different cloud providers and Develop a strategy for avoiding vendor lock-in when using cloud services.
3. Practice securing cloud resources and services using techniques like encryption, access Controls, and network segmentation, based on real-life use cases of cloud security Breaches.

UNIT – III

Cloud Providers, Security and Network: Cloud Providers and Vendor Lock-In, Overview of cloud providers, Pros and cons of using a single vendor, avoiding vendor lock-in

Cloud Security-Importance of cloud security, Cloud security challenges, Best practices for securing cloud infrastructure

Cloud Networking-Overview of cloud networking, Virtual Private Cloud (VPC), NetworkAddress Translation (NAT)

Practice:

1. Learn how to design and configure cloud networks, including virtual private clouds, Subnets, and load balancers, using real-life examples of cloud networking challenges and solutions.
2. Experiment with different cloud storage services and learn how to manage data in the Cloud, including backup, replication, and archiving, based on real-life use cases of Cloud storage.
3. Compare and contrast different cloud storage services like Amazon S3, Google Cloud Storage, and Azure Blob Storage, and learn how to use them for specific use cases like object storage, block storage, and file storage.

UNIT – IV

Cloud Storage and Services: Cloud Storage, Overview of cloud storage, Types of cloudstorage (Object, Block, File),Amazon S3 and other popular cloud storage services

Cloud Storage Services-Overview of cloud storage services, Database as a Service (DBaaS),Backup as a Service (BaaS), Archive as a Service (AaaS)

Practice:

1. Calculate the costs and benefits of using cloud computing versus on-premises Infrastructure for a hypothetical workload, using real-life cost models and business Case examples.
2. Learn how to ensure cloud resources and services comply with industry regulations and Organizational policies, using real-life examples of cloud governance and compliance Challenges.
3. Practice migrating a legacy application to the cloud, including assessing readiness, Choosing the right migration approach, and managing the migration process, based on Real-life migration projects.

UNIT – V

More on Cloud Computing: Cloud Computing Economics, Overview of cloud economics, Pay-as-you-go pricing model, Cost analysis and optimization

Cloud Governance and Compliance-Overview of cloud governance, Compliance and regulatory considerations, Data privacy and security laws

Cloud Migration-Importance of cloud migration, Strategies for cloud migration, Tools and techniques for cloud migration

Cloud Performance and Monitoring-Overview of cloud performance, Monitoring and logging tools, Performance optimization techniques

Cloud Scalability and Availability-Overview of cloud scalability, Techniques for scaling cloud infrastructure, High availability and disaster recovery in the cloud

Future of Cloud Computing-Emerging trends in cloud computing, New technologies and services, Future prospects for cloud computing.

Practice:

1. Learn how to monitor and optimize the performance of cloud resources and services, Including identifying bottlenecks, configuring monitoring tools, and implementing Automated scaling, using real-life use cases of cloud performance issues.
2. Design and configure cloud resources and services for high scalability and availability, Including strategies like auto-scaling, multi-region deployment, and fault-tolerant Architectures, based on real-life examples of cloud scalability and availability Challenges.
3. Research and present emerging cloud technologies and trends, including server less Computing, edge computing, and hybrid cloud, and explore how they might impact the Future of cloud computing.

Textbooks:

1. Cloud Computing: Principles and Paradigms by Rajkumar Buyya, James Broberg, and Andrzej Goscinski - Wiley-IEEE Press. 1st Edition. ISBN-13: 978-0470887995.
2. Cloud Native Architectures: Design high-availability and cost-effective applications for the cloud by Tom Laszewski and Kamal Arora - O'Reilly Media 1st edition. ISBN-13: 978-1492056244.

Reference Books:

1. Cloud Computing: A Practical Approach by Anthony T. Velte, Toby J. Velte, and Robert Elsenpeter - McGraw-Hill Education, 1st edition. ISBN-13: 978-0071626958.
2. Cloud Architecture Patterns: Using Microsoft Azure by Bill Wilder - Microsoft Press, 1st Edition. ISBN-13:978-0735684186.
3. Cloud Computing for Science and Engineering by Ian Foster and Dennis B. Gannon - MIT Press, 1st edition. ISBN-13:978-0262036120.
4. Cloud Computing: Concepts, Technology & Architecture by Thomas Erl, Ricardo Puttini, and Zaigham Mahmood - Prentice Hall, 1st Edition, ISBN-13:978-0133387520.

Web Links:

1. <https://www.geeksforgeeks.org/architecture-of-cloud-computing/>
2. <https://corporatefinanceinstitute.com/resources/data-science/cloud-services/>

CLOUD AND DEVOPS

Course Code: 2504MC38

L	T	P	C
2	0	1	3

Course Outcomes: At the end of the Course, Student will be able to:

- CO1:** Explain the principles of DevOps and its importance in modern software development.
- CO2:** Identify and analyze different DevOps tools and frameworks for continuous integration, delivery, and deployment.
- CO3:** Apply infrastructure as code and configuration management using appropriate tools and best practices.
- CO4:** Design and implement a DevOps pipeline for cloud-native applications or big data projects.
- CO5:** Demonstrate effective communication and collaboration skills required for building a DevOps culture and team.

Mapping of Course Outcomes with Program Outcomes:

CO/ PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	2	1	2	-	-	-	3	-
CO2	2	3	-	3	-	-	-	1
CO3	3	2	3	3	-	-	-	-
CO4	1	-	3	3	2	-	-	2
CO5	-	2	1	-	3	2	-	3

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT – I

DevOps Fundamentals: Introduction to DevOps, Definition and History of DevOps, Importance of DevOps, Key Principles of DevOps, Agile Development Methodology Overview, Scrum and Kanban Frameworks. **Agile Development Methodology** -Agile Methodology Overview. Scrum vs. Kanban, Agile vs. Waterfall, CI Tools and Frameworks **Continuous Delivery (CD)**-Continuous Delivery (CD) Definition and Benefits, CD Pipeline Components, CD Tools and Frameworks

Continuous Deployment (CD)-Continuous Deployment (CD) Definition and Benefits, CD vs. Continuous Deployment, CD Pipeline Components

Continuous Integration (CI) -Continuous Integration (CI) Definition and Benefits, IaC Tools and Frameworks, IaC Best Practices, Wrap-up and Review

Infrastructure as Code (IaC) -Infrastructure as Code (IaC) Definition and Benefits, IaC Tools and Frameworks, IaC Best Practices.

Practice:

1. Introduction to DevOps and Agile Methodology: Set up a small Agile project using Scrum or Kanban frameworks. Track the progress and discuss the differences between Agile and Waterfall methodologies.
2. Continuous Integration (CI): Implement a CI pipeline using Jenkins. Create a simple project, configure Jenkins to run automated tests, and generate reports
3. Continuous Delivery (CD): Create a CD pipeline using tools like GitLab CI or CircleCI. Deploy a web application to a staging environment automatically after passing CI tests.

UNIT – II

DevOps Tools: Introduction to Configuration Management, Definition and benefits of configuration management, Configuration management tools and frameworks overview, Configuration management best practices

Git Essentials-Introduction to Git, Basic Git commands and workflows, it branches and merging strategies.

Jenkins Fundamentals-Introduction to Jenkins, Overview of Jenkins plugins and integrations, Jenkins best practices and common use cases, setting up Jenkins environment

Containerization with Docker-Introduction to Docker, Understanding Docker architecture and components, Docker best practices and security considerations, Docker hands-on demonstration

Orchestration with Kubernetes-Introduction to Kubernetes, Overview of Kubernetes architecture and components, Kubernetes best practices for scalability and resilience, Deploying applications on Kubernetes

Automating with Ansible-Introduction to Ansible, Creating Ansible playbooks and using modules, Ansible best practices for configuration management, Ansible hands-on lab

Practice:

1. Infrastructure as Code (IaC): Write Terraform scripts to provision and manage infrastructure on a cloud Platform (e.g., AWS, Azure, GCP). Demonstrate version control and rollback features.
2. Configuration Management with Ansible: Create and run Ansible playbooks to automate the setup of a Web server (e.g., Apache, Nginx) on multiple virtual machines.
3. Automated Testing with CI: Set up a simple project with a basic test suite (e.g., unit tests). Configure a CI tool like Travis CI or GitHub Actions to automatically run the tests on every commit.

UNIT – III

DevOps Advance Tools: Introduction to Terraform-Overview of DevOps Tools, Introduction to Infrastructure as Code (IaC), Introduction to Terraform, Terraform Providers and Resources

Terraform Best Practices-Understanding Infrastructure as Code (IaC) principles, Terraform Configuration Management, Terraform State Management, Terraform

Lifecycle Management, Terraform Security Best Practices

Introduction to Configuration Management with Chef-Introduction to Configuration Management, Overview of Chef, Chef Cookbooks and Recipes, Chef Architecture, Chef Infrastructure Setup. **Chef Best Practices**-Understanding Configuration Management Best Practices, Chef Environment Management, Testing Chef Cookbooks, Chef Code Deployment Best Practices, Chef Security Best Practices. Introduction to Puppet, Introduction to Puppet, Puppet Manifests and Modules, Puppet Architecture, Puppet Infrastructure Setup, Puppet Code Deployment

Monitoring, Alerting, Logging, and Analysis-Overview of Monitoring, Alerting, Logging, and Analysis, Importance of Monitoring and Alerting in DevOps, Monitoring Tools and Frameworks, Alerting Tools and Frameworks, Logging Tools and Frameworks, Analysis Tools and Frameworks

Practice:

1. Containerization with Docker: Build and run a Docker container for a simple web application. Explore Docker Compose for managing multi-container applications
2. Orchestration with Kubernetes: Deploy a multi-tier application on a Kubernetes cluster. Implement scaling And load balancing features and monitor the application's performance.
3. Terraform for IaC: Write advanced Terraform scripts to manage infrastructure as code, including state Management and provisioning of complex cloud resources.

UNIT – IV

DevOps Security and Compliance + Cloud Computing: High Availability and Load Balancing-Overview of High Availability (HA) and Load Balancing, Concepts and Techniques of HA and Load Balancing, A and Load Balancing Tools and Frameworks, Case Studies: Implementing HA and Load Balancing in DevOps Environments, Best Practices for HA and Load Balancing

DevOps Security and Compliance-Introduction to DevOps Security and Compliance, Understanding Security and Compliance in DevOps Environments-Best Practices for Ensuring Security and Compliance, Tools and Frameworks for Security and Compliance Automation, Case Studies: Implementing Security and Compliance in DevOps Pipe

Cloud Computing Fundamentals-Overview of Cloud Computing, Cloud Service Models (IaaS, PaaS, SaaS) and Deployment Models (Public, Private, Hybrid, Multi-cloud), Key Considerations for Cloud Adoption, Best Practices for Cloud Computing in DevOps, Case Studies: Implementing Cloud Computing in DevOps Workflows

Virtualization in DevOps-Introduction to Virtualization, Types and Benefits of Virtualization (Server, Network, Storage), Virtualization Tools and Frameworks, Integration of Virtualization with DevOps Practices, Best Practices for Virtualization in DevOps Environments

Service-Oriented Architecture (SOA)-Introduction to Service-Oriented Architecture (SOA), Contrasting SOA with Micro services Architecture, Best Practices for Designing and Implementing SOA, Case Studies: Implementing SOA in Enterprise DevOps Environments, Integration of SOA with DevOps Practices and Tools

Practice:

1. **Monitoring and Logging:** Set up monitoring and alerting for a web application using Prometheus and Grafana. Implement centralized logging with the ELK stack (Elasticsearch, Logstash, Kibana).
2. **Configuration Management with Chef:** Develop Chef Cookbooks and recipes to automate the Configuration of an application stack. Test and deploy these configurations using Chef Workstation.
3. **High Availability and Load Balancing:** Configure load balancers (e.g., HAProxy, Nginx) and set up high Availability for a web application using redundant instances and failover techniques.

UNIT – V

DevOps Data Analysis: Testing in DevOps-Introduction to Testing in DevOps, Overview of Testing Processes, Types of Testing in DevOps, Testing Frameworks and Tools. Testing BestPractices

DevOps for Mobile Applications-Introduction to DevOps for Mobile Applications, Mobile App Development Lifecycle, Mobile App Testing and Deployment, Mobile App Analytics and Monitoring, Mobile App Security and Compliance

DevOps for Big Data-Introduction to DevOps for Big Data, Big Data Infrastructure and Architecture, Big Data Storage and Processing Technologies, DevOps Tools and Frameworks for Big Data, Big Data Analytics and Visualization

DevOps for Cloud-Native Applications-Introduction to DevOps for Cloud-Native Applications, Cloud-Native Architecture and Design Patterns, Containerization and Orchestration Tools, Cloud-Native DevOps Tools and Frameworks, Best Practices for DevOps in Cloud-Native Application Development

DevOps Culture and Team Dynamics-Introduction to DevOps Culture, Agile and Lean Principles in DevOps Culture, Building a DevOps Team, DevOps Communication and Collaboration Tools, Best Practices for Creating a DevOps Culture

Future of DevOps-Emerging Trends and Technologies in DevOps, Predictions for the Future of DevOps, Impact of DevOps on Business and IT, Challenges and Opportunities in DevOps, Best Practices for Staying Up-to-Date with DevOps Trends and Developments

Practice:

1. **Security and Compliance:** Implement security best practices in a CI/CD pipeline. Use tools like SonarQube For static code analysis and Vault for secrets management.
2. **Cloud Computing Fundamentals:** Deploy a multi-cloud architecture using services from different cloud Providers (AWS, Azure, GCP). Explore cloud service models (IaaS, PaaS, SaaS) in practical scenarios.
3. **Virtualization in DevOps:** Create and manage virtual machines using a hypervisor (e.g., VMware, Virtual Box). Integrate these VMs into a DevOps workflow for testing and development purposes.

Textbooks:

1. "The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations" by Gene Kim, Jez Humble, Patrick Debois, and John Willis. Published by IT Revolution Press, 1st Edition, ISBN-13: 978-1942788003
2. "DevOps: A Software Architect's Perspective" by Len Bass, Ingo Weber, and Liming Zhu. Published by Addison-Wesley Professional, 1st Edition, ISBN-13: 978-1942788003.

Reference Books:

1. "Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation" by Jez Humble and David Farley. Published by Addison-Wesley Professional, 1st Edition, ISBN-13: 978-0321601919.
2. "Infrastructure as Code: Managing Servers in the Cloud" by Kief Morris. Published by O'Reilly Media, 2nd edition, ISBN-13: 978-1491924358.
3. "Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale" by Jennifer Davis and Katherine Daniels. Published by O'Reilly Media, 1st edition, ISBN-13 978-1491924358

Web Links:

1. https://onlinecourses.nptel.ac.in/noc19_cs64/preview
2. <https://elearn.nptel.ac.in/shop/iit-workshops/completed/cicd-devops-automation-and-devsecops-automation/?v=c86ee0d9d7ed>

CLOUD ENGINEERING

Course Code: 2504MC39

L	T	P	C
2	0	1	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Explain the benefits and drawbacks of different cloud computing platforms and select the most appropriate platform for a given application.
- CO2:** Design and deploy applications using containerization technologies such as Docker and container orchestration platforms such as Kubernetes.
- CO3:** Apply DevOps practices and tools for continuous integration, testing, and deployment of cloud-based applications.
- CO4:** Design and implement security measures to protect cloud-based applications from potential threats and attacks.
- CO5:** Develop and implement disaster recovery plans to ensure high availability and data recovery in case of failures.

Mapping of Course Outcomes with Program Outcomes:

CO/ PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	2	3	1	2	-	-	-	-
CO2	1	2	3	3	-	-	-	1
CO3	2	3	2	3	2	-	-	-
CO4	2	-	2	3	-	-	3	2
CO5	1	1	3	-	1	3	-	3

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT – I

Introduction to Cloud Platforms: Cloud computing platforms, AWS, Azure, GCP.

Containerization-Docker, Kubernetes, StatefulSets.

DevOps practices-CI/CD, Version control, Automated testing, IaC.

Practice:

1. Setting up a Virtual Machine on AWS EC2 Instance.
2. Docker Container Deployment and Management.
3. Implementing Continuous Integration and Deployment with Jenkins.

UNIT – II

Introduction to Cryptography + Threats and Vulnerabilities: Web servers and deployment tools, Apache, Nginx, Deployment tools

Security and scalability-Security best practices, Scalability, Permissioned vs permissionless block chains, Encryption. **Serverless computing**-AWS Lambda, Azure Functions, Google Cloud Functions, Event-driven computing, Serverless architecture.

Practice:

1. Deploying a Web Application using Nginx and Deployment Tools.
2. Securing and Scaling a Web Application.
3. Building and Deploying Serverless Functions with AWS Lambda.

UNIT – III

Block chain Interpretability & Tokenomics: Monitoring and logging, Monitoring, Logging, Testing and debugging smart contracts, Google Cloud Monitoring

Performance optimization-Performance tuning, Database optimization, Content Delivery Networks. **Disaster recovery planning**-Backup and recovery strategies, Replication, Failover, Disaster recovery tools

Networking and load balancing-Virtual networks, Subnets, Load balancers, DNS/CDN.

Practice:

1. Monitoring and Logging for a Web Application.
2. Optimizing Sorting Algorithms for Large Data Sets.
3. Disaster Recovery Planning and Testing for IT Systems.

UNIT – IV

Advance concepts of Blockchain: Security in deployment and hosting, Network security, Firewall, Intrusion detection and prevention, Security groups, SSL/TLS

Deployment strategies-Blue-green deployment, Canary deployment, Rolling deployment, A/B testing. **Performance monitoring and profiling**-Performance monitoring tools, Profiling tools, Performance metrics, Performance analysis.

Practice:

1. Networking and Load Balancing for High Availability.
2. Secure Deployment and Hosting for Web Applications.
3. Deploying and Managing a Database Server with MySQL.

UNIT – V

Block chain Use Cases + Tools & Frameworks: Deployment automation, Scripting, Configuration management tools, Continuous deployment, Infrastructure as Code (IaC)

Compliance and governance-Compliance requirements, Auditing, Security policies, Risk management.

Practice:

1. Performance Monitoring and Profiling for Web Applications.
2. Deployment Automation for Web Applications.
3. Compliance and Governance in IT Operations.

Textbooks:

1. "Cloud Computing: From Beginning to End" by Ray Jezek and Joe Khoury, 1st Edition, Wiley Publishing, ISBN Number: 978-0470615207
2. "Cloud Computing Principles and Paradigms" by Rajkumar Buyya, James Broberg, and Andrzej Goscinski, 1st Edition, Wiley Publishing, ISBN Number: 978-0470887998

Reference Books:

1. Kubernetes in Action, Marko Luksa, 1st Edition, Manning Publications, ISBN Number: 978-1617293726
2. "Docker Deep Dive" by Nigel Poulton, 4th Edition Independently published, ISBN: 978-1736049118
3. The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations, Gene Kim, Jez Humble, Patrick Debois, and John Willi, 1st Edition, Publisher: IT Revolution Press, ISBN: 978-1942788003

Web Links:

1. <https://www.coursera.org/in/articles/what-is-a-cloud-engineer>

FUNDAMENTALS OF PRODUCT MANAGEMENT

Course Code: 2504MC40

L	T	P	C
2	0	1	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Learn the core principles and processes of product management, market research, competitive analysis, product strategy etc.
- CO2:** Develop skills and knowledge in product management tools and techniques, including product road mapping, user persona development, and agile methodologies.
- CO3:** Analyze market trends, customer needs, and competitive landscapes to identify potential opportunities and threats for new and existing products.
- CO4:** Apply product management concepts and techniques to real-world scenarios and problems and develop comprehensive product strategies.
- CO5:** Learn best practices and ethical considerations in product management and develop product leadership skills that include stakeholder management, collaboration, and career development.

Mapping of Course Outcomes with Program Outcomes:

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	2	3	-	-	-	2	-	1
CO2	3	2	-	3	2	1	-	-
CO3	-	3	2	2	-	-	-	2
CO4	-	-	3	-	-	2	-	3
CO5	-	-	-	-	3	3	3	-

Mapping of Course Outcomes with Program Specific Outcomes:

	PSO1	PSO2
CO1	3	-
CO2	3	2
CO3	-	3
CO4	-	2
CO5	-	3

UNIT – I

Introduction to Product Management: What is Product Management? Definition and Importance of Product Management, Role of Product Managers in the Company, Product Management vs. Project Management.

Product Life Cycle and Market Research-Different stages of Product Life Cycle, Importance of Product Life Cycle, Challenges faced by Product Managers in Product Life Cycle.

Practice:

1. Provide students with wire framing and prototyping tools and have them create Prototypes for a product idea. This experiment helps students learn how to visually Communicate product concepts.

2. Assign students to gather requirements for a product from hypothetical stakeholders. This experiment helps students practice eliciting and documenting product requirements.

UNIT – II

Market Research for Product Management: Importance of Market Research, Types of Market Research, Techniques for conducting Market Research.

Competitive Analysis and Product Strategy-Introduction to Competitive Analysis, Types of Competitive Analysis, How to conduct a Competitive Analysis.

Practice:

1. Provide students with examples of technical debt in software products and discuss its Impact. This experiment helps students understand the trade-offs involved in Technical decisions.
2. Divide students into groups and assign them to research and present on either Agile or waterfall development methodologies. This experiment helps students understand the differences between these approaches.

UNIT – III

Product Strategy: Understanding Product Strategy, Developing Product Vision and Mission Statement, Setting Product Goals and Objectives, Group Activity

Product Development and Launch-Introduction to Product Road mapping, Types of Product Roadmaps, Building a Product Roadmap, Group Activity.

Practice:

1. Have students create a comprehensive plan for launching a new product, including Marketing, sales, and distribution strategies. This experiment helps students understand the complexities of product launches.

UNIT – IV

Minimum Viable Product (MVP) and User Research: Definition and Importance of MVP, How to build an MVP, User Research and Personas, Group Activity.

Marketing, Growth, and Leadership-Introduction to Product Launch, Marketing Channels and Promotion, Digital Marketing, Group Activity.

Practice:

1. Task students with creating a digital marketing campaign for a specific product, including strategies for SEO, SEM, and social media marketing. This experiment helps students apply digital marketing concepts in a practical context.

UNIT – V

Product Growth and Maintenance: Product Metrics and KPIs, Product Analytics, Customer Success Management, Group Activity, Product Leadership, Stakeholder Management, Product Management Tools and Techniques.

Practice:

1. Provide students with data on product metrics and key performance indicators (KPIs) and ask them to analyze and interpret the data to make strategic recommendations. This Experiment helps students understand how to measure product success and Performance.

Textbooks:

1. Product Management: A Multi-Disciplinary Perspective, R. Jayakumar, 1st Edition, PHI Learning Pvt. Ltd., ISBN Number: 978-9389347609
2. The Lean Product Playbook: How to Innovate with Minimum Viable Products and Rapid Customer Feedback by Dan Olsen, Wiley India Pvt. Ltd, ISBN Number: 978-1118960875
3. Product Management in India by Amanjot Malhotra, 1st Edition, Notion Press, 2021, ISBN Number: 978-1685236714

Reference Books:

1. Product Management: An Integrated Approach by Gopalakrishnan R & Suresh P, 1st Edition, Vikas Publishing, 2019, ISBN Number: 978-9325984824
2. Cracking the PM Interview: How to Land a Product Manager Job in Technology by Gayle McDowell & Jackie Bavaro, 1st Edition, Publication- CareerCup, ISBN Number: 978-0984782807.
3. The Art of Product Management: Lessons from a Silicon Valley Innovator by Rich Mironov, BookBaby Publisher in 2020, ISBN Number: 978-1642011167
4. Product Leadership: How Top Product Managers Launch Awesome Products and Build Successful Teams, Richard Banfield, Martin Eriksson, and Nate Walkingshaw, 1st Edition, Publisher: O'Reilly Media, Inc, ISBN Number: 978-1491950357

Web Links:

1. <https://www.mindtheproduct.com/what-are-the-key-fundamentals-of-product-management/>

SOFTWARE QUALITY ASSURANCE

Course Code: 2504MC41

L	T	P	C
2	0	1	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Evaluate different quality attributes as they relate to a specific software application and design scenarios to test them.
- CO2:** Draft an outline for an SQAP that outlines quality goals, processes, and resources for a small-scale project.
- CO3:** Describe one quality standard (e.g., ISO 9001, CMMI) and its relevance to SQA practices.
- CO4:** Employ error handling or input validation techniques to increase application robustness.
- CO5:** Analyze quality data and propose solutions for improvement (e.g., enhanced code review processes, and additional testing).

Mapping of Course Outcomes with Program Outcomes:

CO/ PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	2	1	-	-	-	-	-
CO2	2	-	3	-	1	2	-	-
CO3	-	2	-	3	-	-	1	-
CO4	1	-	-	2	-	-	3	1
CO5	-	3	2	-	-	-	-	2

Mapping of Course Outcomes with Program Specific Outcomes:

CO/ PSO	PSO1	PSO2
CO1	2	-
CO2	2	3
CO3	-	-
CO4	-	2
CO5	3	-

UNIT – I

Software Quality Assurance Framework and Standards: Software Quality Attributes, Software Quality Assurance, Components of Software Quality Assurance, Software Quality Assurance Plan, Quality Standards: ISO 9000 and Companion ISO Standards, CMM, CMMI, PCMM, Malcom.

Practice:

1. Evaluating Software Quality Attributes, Select a simple application (e.g., Calculator), Identify key quality attributes (KQAs) relevant to the application (e.g., Reliability, usability, performance), Design scenarios to test each KQA, Analyze results and document how the application meets or fails to meet KQAs.

2. Building a Software Quality Assurance Plan (SQAP), Choose a small-scale project With defined requirements. Develop an SQAP outlining quality goals, processes, and resources., Include sections on defect prevention, testing strategies, reporting, and risk Management.
3. Demystifying Quality Standards (ISO, CMMI, etc.), Research a specific quality Standard (e.g., ISO 9001, CMMI), Identify key principles and how they relate to SQA Practices. Compare and contrast different standards, highlighting strengths and Weaknesses.

UNIT – II

Quality Assurance: Quality Assurance as dealing with defects, Defects Prevention Techniques, Defect reduction: Inspection –Direct fault detection and removal, testing., Defect Containment: Software fault tolerance and safety assurance and failure containment

Practice:

1. Defect Prevention Techniques in Action, Simulate code development for a smallFunction (e.g., string manipulation)., Apply defect prevention techniques like code Reviews, pair programming, and static analysis tools. Track the number and types of defects identified at each stage.
2. Defect Containment Techniques in Practice, Design a test suite for the simulated code from Experiment 2.1, Implement fault tolerance mechanisms (e.g., error handling, input validation)., Simulate failure scenarios and evaluate containment effectiveness.

UNIT – III

Software Quality Assurance Metrics and Measurement: Software Quality Metrics, Product Quality metrics. Software Quality metrics methodology, Establish quality requirements, Identify Software quality metrics., Implement the software quality metrics, analyze software metrics results, and validate the software quality metrics.

Practice:

1. Defining and Measuring Software Quality Metrics, Select a real-world or simulated software project. Identify relevant product and process quality metrics (e.g., code Coverage, defect density, complexity measures)., Collect data during the development process and analyze trends., Use these metrics to assess project health and make Informed decisions.
2. Exploring Functional vs. Structural Testing, Develop test cases for a simple function (e.g., login functionality) based on requirements. Implement both functional (black box) and structural (white box) testing approaches., Compare the effectiveness of each technique in uncovering defects.

UNIT – IV

Software Testing: Functional vs, Structural testing, Test planning and preparation, Test executions, Result Checking and Measurement, Test Automation., Testing techniques: Adaptation, specialization, and Integration, Case Study: Hierarchical web Testing., Process Improvement: Process Classification, Process Measurement, Process Analysis, and Modelling

Practice:

1. Building, Executing, and Evaluating Test Plans, Design a test plan for a real-world application or a simulated scenario. Specify test objectives, scope, resources, and testing levels (unit, integration, system)., Execute the test plan and record results., Analyze results, identify defect types, and suggest improvements to the test plan.
2. Introduction to Test Automation, Choose a repetitive testing task (e.g., regression testing for a core module)., Learn and experiment with a basic test automation tool (e.g., Selenium)., Record scripts for the chosen task and evaluate the benefits of Automation in terms of efficiency and accuracy.

UNIT – V

Quantifiable Quality Improvement: QA monitoring and measurement, Analysis, and follow-up actions, Implementations, Integration, and tool support. and product-specific models., Risk Identification for quantifiable quality improvement: Traditional, statistical analysis techniques., New techniques for risk identification.

Practice:

1. Monitoring QA Metrics and Taking Corrective Actions, Monitor software quality metrics throughout a development cycle (e.g., code coverage, defect density)., Use control charts or other statistical tools to identify trends and outliers., Propose corrective actions based on the data analysis (e.g., additional code reviews, targeted testing).
2. Risk Identification for Quality Improvement, Conduct a brainstorming session to identify potential risks related to software quality within a project (e.g., schedule slippage, requirement changes)., Analyze risks using a risk matrix (likelihood vs. impact) and prioritize mitigation efforts.

Textbooks:

1. Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement, Jeff Tian, 1st Edition, Wiley India Pvt. Ltd, ISBN Number: 978-8126511667
2. Software Testing and Quality Assurance by Dr. Rajiv Chopra; 1st Edition , Publication by S.K. Kataria & Sons, ISBN Number: 978-8188453358

Reference Books:

1. Software Engineering: A Practitioner Approach by Roger Pressman, Roger S. Pressman and Bruce R. Maxim, 10th Edition, Tata McGraw-Hill (TMH), ISBN Number: 978-1259029988
2. Software Quality Assurance: From Theory to Implementation by Daniel Galin, 1st edition Pearson India, ISBN Number: 978-8131709382
3. Software Reliability Engineering by John D. Musa, 1st Edition TMH Pub, ISBN: 978-0070582261
4. Software Quality: State of the Art in Management, Testing, and Tools, Martin Wieczorek and Dirk Meyerhoff, 1st Edition, Publisher: Springer, ISBN Number: 978-3662495816

Web Links:

1. <https://www.geeksforgeeks.org/software-engineering-software-quality-assurance/>
2. <https://www.softwaretestinghelp.com/software-quality-assurance/>

SOFTWARE ARCHITECTURE

Course Code: 2504MC42

L	T	P	C
2	0	1	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Demonstrate proficiency in software architecture fundamentals
- CO2:** Design high-quality software architecture
- CO3:** Apply software quality assurance practices
- CO4:** Utilize formal models and tools for architecture design
- CO5:** Evaluate software quality metrics and testing strategies

Mapping of Course Outcomes with Program Outcomes:

CO/ PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	-	2	2	-	-	-	1
CO2	2	3	3	-	2	-	-	-
CO3	-	2	-	3	2	-	1	2
CO4	1	-	-	3	-	2	-	3
CO5	-	-	1	-	-	3	3	2

Mapping of Course Outcomes with Program Specific Outcomes:

CO/ PSO	PSO1	PSO2
CO1	2	-
CO2	3	2
CO3	-	3
CO4	-	-
CO5	-	-

UNIT – I

Software Architecture Fundamentals and Designing: Software Architecture Terms, Components, Relationships, Views, Architectural Styles, Types of IT Architectur, Frameworks, Patterns, Methodologies, Processes, Functional and Non-functional Properties. **Overview of Software Architecture**-Understanding the importance of quality architecture, Introduction to architectural styles and trends, Exploring frameworks and patterns in software architecture. **Quality Architecture**-Factors contributing to quality architecture, Architectural lifestyle and its significance. **Service-Oriented Architecture (SOA)**-Principles of SOA, Service-Oriented Analysis and Design (SOAD), Trends and Enterprise-Wide SOA, Application of SOA in real-world scenarios.

Practice:

1. Designing simple software architecture: Students could design the architecture for a basic software system, focusing on components, connectors, and their interactions.
2. Implementing different architectural styles: Students could implement simple applications using various architectural styles such as client-server, peer-to-peer, and layered architectures.

3. Conducting architectural reviews: Students could peer review software architecture designs, focusing on identifying potential risks and suggesting improvements.

UNIT – II

Architectural Styles and Formal Models: Introduction to Architectural Styles, Pipes and Filters, Data Abstraction, Object-Orientation, Event-Based, Implicit Invocation, Layered Systems, Repositories, Interpreters, Process Control, Heterogeneous Architectures

Formalizing Software Architecture-Formalizing a specific system architecture, Formalizing architectural styles and design space, Requirements for Architecture Description Languages, Tools for architectural design: Universal connector language, automated support.

Practice:

1. Analyzing software quality standards: Students could analyze and compare different software quality standards such as ISO 9001 or CMMI, assessing their impact on development processes and outcomes.
2. Implementing defect prevention techniques: Students could implement defect prevention techniques such as code reviews, static analysis, and unit testing to reduce the number and severity of defects.
3. Experimenting with fault tolerance mechanisms: Students could experiment with fault tolerance mechanisms such as redundancy, replication, and diversity to improve system reliability.

UNIT – III

Software Quality Framework and Standards: Concept of Software Quality, Understanding software quality attributes and their significance.

Software Quality Assurance (SQA)-Components of SQA: Planning, implementation, assessment, and improvement. Steps to Develop a SQA Plan: Defining objectives, establishing processes, implementing procedures, and continuous improvement.

Quality Standards-ISO 9000 and Companion ISO Standards, CMM, CMMI, PCMM, Malcolm Baldrige, 3 Sigma, 6 Sigma.

Practice:

1. Analyzing software quality metrics: Students could analyze software quality metrics such as defect density, code coverage, and cyclomatic complexity to measure software quality and identify areas for improvement.
2. Conducting root cause analysis: Students could conduct root cause analysis of software defects to identify underlying process or design issues contributing to the defects.
3. Implementing continuous integration and delivery: Students could set up and configure continuous integration and delivery pipelines to automate the software build, test, and deployment process.

UNIT – IV

Defect Prevention and Containment: Quality Assurance Strategies, Defects Prevention Techniques: Education & training, formal methods., Defect Reduction: Inspection, testing.

Defect Containment-Software Fault Tolerance and Safety Assurance: Implementing mechanisms to contain failures and ensure system reliability.

Practice:

1. Experimenting with different testing methodologies: Students could experiment with different testing methodologies such as black-box testing, white-box testing, and exploratory testing to assess their effectiveness in identifying defects.
2. Analyzing the impact of software quality assurance on customer satisfaction: Students could analyze the relationship between software quality assurance activities and customer satisfaction, using surveys or feedback mechanisms to gather data.
3. Implementing software fault injection testing: Students could simulate and analyze the impact of software faults on system behavior using fault injection techniques.

UNIT – V

Software Quality Assurance Metrics, Measurement, and Testing: Software Quality Metrics, Product Quality Metrics, Process Quality Metrics, Metrics for Software Maintenance
Metrics Methodology-Establishing quality requirements, Identifying software quality metrics, Implementing and analyzing software metrics results, Software quality indicators, Fundamentals in Measurement theory

Software Testing-Functional vs. Structural Testing, Test Planning and Preparation, Test Execution and Result Checking, Test Automation.

Practice:

1. Conducting failure mode and effects analysis (FMEA): Students could conduct FMEA to identify potential failure modes and their impact on system reliability, using this Analysis to improve system design and implementation.
2. Experimenting with architectural patterns for security: Students could experiment with architectural patterns for security such as zero-trust architecture or defense-in-depth, evaluating their effectiveness in mitigating security threats.
3. Analyzing the impact of software defects and failures on user experience: Students could analyze the impact of software defects and failures on user experience, using metrics such as user satisfaction scores or user retention rates.

Textbooks:

1. Service Oriented Architecture (SOA): A Practical Guide for Understanding and Implementing Service-Oriented Architecture, Shankar Kambhampaty, 2nd Edition, Wiley India Pvt. Ltd, ISBN-10: 8126528838, ISBN-13: 978-8126528834
2. Pattern-Oriented Software Architecture, Volume 1: A System of Patterns, Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal, 1st Edition, Wiley India Pvt. Ltd, ISBN-10: 9814126730, ISBN-13: 978-9814126734

Reference Books:

1. Software Architecture in Practice, Len Bass, Paul Clements, Rick Kazman, 3rd Edition, Pearson Edu, ISBN-10: 9332518668, ISBN-13: 978-9332518667
2. Practical Software Testing: A Process-Oriented Approach, Ilene Burnstein, Springer International Edition, Publication: Springer, ISBN-10: 8181280734
3. Software Testing in the Real World: Improving the Process, Edward Kit, 1st Edition, Pearson Education, ISBN-10: 817758355X, ISBN-13: 978-8177583551

Web Links:

1. <https://sarrahpaliya.medium.com/understanding-software-architecture-a-complete-guide-cb8f0590060>
2. https://www.tutorialspoint.com/software_architecture_design/introduction.htm

SOFTWARE VERIFICATION, VALIDATION AND TESTING

Course Code: 2504MC43

L	T	P	C
2	0	1	3

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Demonstrate understanding of software testing fundamentals and principles.
- CO2:** Apply diverse testing approaches and techniques for effective software testing.
- CO3:** Analyze software reliability models and develop testing management strategies.
- CO4:** Conduct various types of software testing, considering usability, scalability, and security.
- CO5:** Utilize system testing methodologies and tools for comprehensive software evaluation.

Mapping of Course Outcomes with Program Outcomes:

CO/ PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	2	-	1	-	-	-	-
CO2	2	3	-	-	-	-	-	-
CO3	-	2	3	-	-	1	-	-
CO4	1	-	2	3	-	-	2	-
CO5	-	-	1	-	3	2	-	1

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT – I

Software Testing Fundamentals: Software Testing Fundamentals, Testing Objectives, TestCases, General Principles of Testing.

Agile Methodology Impact-Verification and Validation, Failure, Error, Fault, DefectAnalysis

Testing Lifecycles-Human Error in Testing, Testing and Debugging, Test Metrics.

Practice:

- Unit Testing with Mock Objects: Conduct experiments on unit testing using mock Objects to isolate and test individual components of a software system.
- Automated Regression Testing: Experiment with automated regression testing tools to Validate that recent code changes haven't introduced new bugs or issues.
- Validation of Input Data: Perform experiments to validate input data, ensuring that the Software handles various types of input correctly (e.g., boundary values, invalid inputs)

UNIT – II

Testing Approaches and Techniques: Testing Approaches, Static Testing, Dynamic Testing, Black Box Testing **Structured Group Examination-**Equivalence Class Partitioning, Boundary Value Analysis, Cause-Effect Graphing **Advanced Testing Techniques-**Gray Box Testing, Experience-Based Testing, Control Flow and Data Flow Testing.

Practice:

1. Exploratory Testing: Conduct experiments on exploratory testing techniques to uncover hidden defects and issues in the software by simulating real-world usage scenarios.
2. Performance Testing: Experiment with performance testing tools to evaluate the responsiveness, scalability, and stability of the software under different loads and stress Conditions.
3. Security Testing: Perform experiments on security testing methodologies to identify Vulnerabilities and weaknesses in the software's security measures

UNIT – III

Design Concepts and Principles: Software Reliability, Reliability Models, Reliability Measures, Verification and Validation Planning.

Top-Down vs. Bottom-Up Testing-Mutation Testing, Test Planning and Management, Testing Process. **Functional vs. Structured Testing-**Maturity Models.

Practice:

1. Usability Testing: Conduct experiments on usability testing to assess the user Friendliness and intuitiveness of the software interface.
2. Ad Hoc Testing: Experiment with ad hoc testing techniques to perform spontaneous and Unplanned tests on the software to identify defects quickly.
3. Static Analysis: Perform experiments on static code analysis tools to identify potential code Defects, security vulnerabilities, and coding standards violations.

UNIT – IV

Software Testing Techniques & Strategies: Types of Testing, Unit Testing, Integration Testing, System Testing

Acceptance Testing-Alpha & Beta Testing, Installation Testing, Usability Testing

Performance Testing-Load Testing, Stress Testing, Security Testing.

Practice:

1. Mutation Testing: Conduct experiments on mutation testing to evaluate the effectiveness of test suites by introducing artificial faults into the codebase and assessing how well the tests detect them.
2. Integration Testing: Experiment with integration testing techniques to verify the interactions between different modules or components of the software system.
3. Model-Based Testing: Perform experiments on model-based testing approaches to Generate test cases from software models and verify the software behavior against these Models.

UNIT – V

Software Technical Metrics: System Tests, Functionality Tests, Robustness Tests, Interoperability Tests.

Scalability Tests-Documentation Tests, Gorilla Testing, Syntax-Based Testing

Testing Tools-Automation of Test Execution, Requirement Tracker, Test Process and Plans.

Practice:

1. **Fault Injection Testing:** Conduct experiments on fault injection techniques to Intentionally introduce faults into the system and observe how the software handles Them.
2. **Compliance Testing:** Perform experiments on compliance testing to ensure that the Software complies with industry standards, regulations, and specifications.
3. **User Acceptance Testing (UAT):** Conduct experiments on UAT methodologies to Involve end-users in testing the software to ensure it meets their requirements and Expectations before deployment.

Textbooks:

1. Software Testing and Quality Assurance, Naik, Kshirasagar; Tripathy, Priyadarshi, 1st Edition, Wiley India pub, ISBN-10: 8126528836, ISBN-13: 978-8126528834
2. Software Testing, Gopalaswamy Ramesh Limaye, 1st Edition, Tata McGraw-Hill Education (TMH), ISBN-10: 0070139903, ISBN-13: 978-0070139909

Reference Books:

1. Introduction To Software Testing, Paul Ammann, Jeff Offutt, 2nd Edition, Cambridge Univ Press, ISBN-10: 1107172012, ISBN-13: 978-1107172012
2. Software testing concepts Tools, K.V.K.K. Prasad, 1st Edition, Dreamtech press, ISBN-10: 8177225324, ISBN-13: 978-8177225321

Web Links:

1. https://onlinecourses.nptel.ac.in/noc22_cs61/preview
2. <https://www.geeksforgeeks.org/software-engineering-verification-and-validation>

DESIGN THINKING

Course Code: 2504MC44

L	T	P	C
0	0	1	1

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Appreciate various design process procedure
- CO2:** Generate and develop design ideas through different technique
- CO3:** Identify the significance of reverse Engineering to Understand products
- CO4:** Draw technical drawing for design ideas
- CO5:** Illustrate design teams to create feasible and user-focused solutions to complex problems in design.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	1	2	2	1	-	-	-	-
CO2	2	3	3	2	1	-	-	-
CO3	1	2	-	2	1	-	-	-
CO4	-	-	2	3	-	-	-	-
CO5	-	1	2	2	3	-	-	-

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT – I

Process of Design Understanding Design thinking: Shared model in team-based design – Theory and practice in Design thinking – Explore presentation signers across globe – MVP or Prototyping.

UNIT – II

Tools for Design Thinking: Real-Time design interaction capture and analysis – Enabling efficient collaboration in digital space – Empathy for design – Collaboration in distributed Design

UNIT – III

Design Thinking in IT: Design Thinking to Business Process modelling – Agile in Virtual collaboration environment – Scenario based Prototyping

UNIT – IV

DT For strategic innovations: Growth – Story telling representation – Strategic Foresight
- Change – Sense Making - Maintenance Relevance – Value redefinition - Extreme
Competition – experience design.

UNIT – V

Design thinking workshop: Design Thinking Workshop Empathize, Design, Ideate,
Prototype and Test.

Textbooks:

1. “Engineering Design”- John.R.Karsnitz, Stephen O’Brien and John P. Hutchinson
Second Edition, Cengage learning (International edition), ISBN Number: 978-1-305-
49650-1
2. "The Design of Business: Why Design Thinking is the Next Competitive
Advantage" - Roger Martin, First Edition, Harvard Business Press, ISBN Number:
978-1422138565

Reference Books:

1. “Engineering Design Process” - Yousef Haik and Tamer M.Shahin, Second Edition,
Cengage Learning, ISBN Number: 978-1-305-20680-4
2. Solving Problems with Design Thinking - Ten Stories of What Works - Hardcover
by Jeanne Liedtka, Andrew King, Kevin Bennett, Columbia Business School
Publishing, ISBN Number: 978-0231156366.

Web Links:

1. www.tutor2u.net/business/presentations/
2. www./Product lifecycle/default.html
3. https://docs.oracle.com/cd/E11108_02/otn/pdf/

RESEARCH METHODOLOGY

Course Code: 2504AC01

L	T	P	C
2	0	0	0

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Understand what research is, why it's done, and its importance in different fields.
- CO2:** Learn how to clearly define research questions and review existing literature to improve research focus.
- CO3:** Gain skills in planning research studies and using literature effectively to support research methods.
- CO4:** Understand how data is collected, measured (quantitative and qualitative), and prepared for analysis in research.
- CO5:** Develop the ability to interpret research findings, write clear reports, and present research results effectively.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	2	3	-	-	-	-	-	1
CO2	-	3	-	-	-	-	-	2
CO3	-	2	1	-	-	-	-	2
CO4	-	2	3	-	-	-	-	1
CO5	-	2	3	-	2	-	-	2

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT – I

Research Methodology: Introduction, Meaning of Research, Objectives of Research, Motivation in Research, Types of Research, Research Approaches, Significance of Research, Research Methods versus Methodology, Research and Scientific Method, Importance of Knowing How Research is Done, Research Process, Criteria of Good Research, and Problems Encountered by Researchers in India.

UNIT – II

Defining the Research Problem: Research Problem, Selecting the Problem, Necessity of Defining the Problem, Technique Involved in Defining a Problem, An Illustration. Reviewing the literature: Place of the literature review in research, bringing clarity and focus to your research problem, improving research methodology, broadening knowledge base in research area, enabling contextual findings, How to review the literature, searching the existing

literature, reviewing the selected literature, developing a theoretical framework, developing a conceptual framework, writing about the literature reviewed.

UNIT – III

Research Design: Meaning of Research Design, Need for Research Design, Features of a Good Design, Important Concepts Relating to Research Design, Different Research Designs, Basic Principles of Experimental Designs, Principles of experimental design- Working with Literature: Importance- finding literature- Using your resources- Managing the literature- Keep track of references- Using the literature- Literature review- On-line Searching: Database-SCI Finder- Scopus- Science Direct-Searching research articles- Citation Index - Impact Factor -H-index.

UNIT – IV

Research Data: Measurement of Scaling: Quantitative-Qualitative, -Classification of Measure scales- Data Collection- Data Preparation.

UNIT – V

Report Writing: Interpretation and Report Writing: Meaning of Interpretation, Technique of Interpretation, Precaution in Interpretation, Significance of Report Writing, Different Steps in Writing Report, Layout. Types of Reports, Oral Presentation, Mechanics of Writing a Research Report, Precautions for Writing Research Reports.

Textbooks:

1. C. R. Kothari, Research Methodology Methods and Techniques, 4th Edition, New Age International Publishers, 2019, ISBN: 978-9386649225
2. The Essential Guide of Doing Research - Zina O'Leary, 3rd Edition, SAGE Publications Ltd, ISBN: 978-1473952072

Reference Books:

1. Research Design: Qualitative, Quantitative, and Mixed Methods Approaches, J. W. Creswell, 4th Edition, SAGE Publications, ISBN: 978-1483346355
2. Research Methodology: A Step-by-Step Guide for Beginners, Kumar, 4th Edition, SAGE Publications Ltd, ISBN: 978-1526446715
3. Research Methods: The concise knowledge base - Trochim, Atomic Dog Publishing, ISBN: 978-1592603946
4. Conducting Research Literature Reviews: From the Internet to Paper - Fink A, Edition: 5th Edition, Sage Publications, ISBN: 978-1506350240
5. Research Methodology is a step-by- step guide for beginners. (For the topic Reviewing the literature (Unit – II)) Ranjit Kumar, 3rd Edition, SAGE Publications Ltd Study Material, ISBN: 978-1849203228

Web Links:

1. <https://paperpal.com/blog/academic-writing-guides/what-is-research-methodology>
2. <https://gradcoach.com/what-is-research-methodology/>
3. <https://www.oxbridgeessays.com/blog/different-types-of-research-methodology-in-research/>
4. https://onlinecourses.nptel.ac.in/noc24_ge41/preview
5. <https://www.euacademic.org/BookUpload/9.pdf>

EMPLOYABILITY SKILLS-I

Course Code: 2504AC02

L	T	P	C
0	0	3	0

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Apply fundamental arithmetic concepts such as number systems, LCM & HCF, ratio & proportion, and averages to solve quantitative problems efficiently.
- CO2:** Analyze number and letter series to identify logical patterns and predict subsequent elements.
- CO3:** Select the option by evaluating relationships in analogy-based problems and detecting inconsistencies in logical sequences.
- CO4:** Demonstrate effective verbal communication skills by using appropriate parts of speech and tone in various interpersonal and professional contexts.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	3	-	-	-	-	-	-
CO2	2	3	-	-	-	-	-	-
CO3	2	3	-	-	-	-	-	-
CO4	-	-	-	-	-	-	-	3

Aptitude:

Number System, LCM & HCF, Ratio and Proportion, Averages

Reasoning:

Number Series, Letter Series, Number Analogy, Letter Analogy, Odd Man Out, Logical Sequence of Words.

Verbal:

Introduction to soft skills, how to improve communication? Parts of Speech, Mind your language towards better English, Vocabulary Expansion

Textbooks:

- Quantitative Aptitude for Competitive Examinations - Dr. R. S. Aggarwal, S. Chand Publishing, ISBN: 978-9352534029
- A Modern Approach to Verbal and Non-Verbal Reasoning - Dr. R. S. Aggarwal, Publication: S. Chand Publishing, ISBN: 978-9352832163

Reference Books:

- Quick Learning Objective General English - Dr. R. S. Aggarwal, Vikas Aggarwal, S. Chand Publishing, ISBN: 978-9352837564
- Quantitative Aptitude for Competitive Examinations, Abhijit Guha, McGraw Hill Education, ISBN: 978-9353160180
- Analytical Reasoning - M. K. Pandey, Arihant Publications, ISBN: 978-9350947982

Web Links:

1. www.indiabix.com
2. www.bankersadda.com

EMPLOYABILITY SKILLS-II

Course Code: 2504AC03

L	T	P	C
0	0	3	0

Course Outcomes:

At the end of the Course, Student will be able to:

- CO1:** Apply percentage, profit & loss, and partnership concepts to solve real-world quantitative problems.
- CO2:** Analyze age-related problems to develop logical problem-solving strategies.
- CO3:** Evaluate logical reasoning questions including ranking, directions, and alphabet tests for accurate decision-making.
- CO4:** Solve pattern-based problems in coding and decoding to enhance analytical skills.
- CO5:** Demonstrate effective written communication using correct grammar and structured presentation techniques

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	3	3	-	-	-	-	-	-
CO2	2	3	-	-	-	-	-	-
CO3	2	3	-	-	-	-	-	-
CO4	2	3	-	-	1	-	-	-
CO5	-	-	-	-	-	-	-	3

Aptitude:

Problems on Ages, Partnership, Percentages, Profit and Loss

Reasoning:

Coding and Decoding, Ranking Test, Alphabet Test, Direction Test

Verbal:

Written communication skill practice, Grammatical use, Concept of 4 step method for presentation, Present Tense

Textbooks:

- Quantitative Aptitude for Competitive Examinations - Dr. R. S. Aggarwal, S. Chand Publishing, ISBN: 978-9352534029
- A Modern Approach to Verbal and Non-Verbal Reasoning - Dr. R. S. Aggarwal, Publication: S. Chand Publishing, ISBN: 978-9352832163

Reference Books:

- Quick Learning Objective General English - Dr. R. S. Aggarwal, Vikas Aggarwal, S. Chand Publishing, ISBN: 978-9352837564
- Quantitative Aptitude for Competitive Examinations, Abhijit Guha, McGraw Hill Education, ISBN: 978-9353160180
- Analytical Reasoning - M. K. Pandey, Arihant Publications, ISBN: 978-9350947982

Web Links:

1. www.indiabix.com
2. www.bankersadda.com