Bachelor of Computer Applications (BCA)

Semester III and IV
*With effect from the academic year **2026-2027***

Under R25 Academic Regulations
(Applicable to Students Admitted
from the Academic Year 2025–2026)

Based on AICTE Model Curriculum for
Bachelor's Degree in Computer Applications



Faculty of Informatics
Osmania University
2026-2027

Table of Contents

GeneralCourseStructure.....3
Schemes of Semesters III and IV for AY 2026-27.....4
Syllabus of Semester III for AY 2026-27.....6
Syllabus of Semester IV for AY 2026-27.....34
Model Question Paper Format for CIE.....60
Model Question Paper Format for SEE.....61

P. V. Sridhar

Dean,
Faculty of Informatics
Osmania University

General Course Structure

A. Definition of Credit:

1 Hr. Lecture (L) per week	1 Credit
1 Hr. Tutorial (T) per week	1 Credit
1 Hr. Practical (P) per week	0.5 Credit
2 Hours Practical (P) per week	1 Credit

B. Course Code and Definition:

Abbreviations	Definitions
PCC	Professional Core Course
PEC	Professional Elective Course
MGC	Management Course
HSC	Humanities Course
LCC	Laboratory Core Course
Proj	Project Work
CIE	Continuous Internal Evaluation
SEE	Semester End Evaluation
HPW	Hours Per Week
L	Lecture Hours/Week
T	Tutorial Hours/Week
P	Practical Hours/Week
Cr	Credits

C. Credits Table

Semester	Credits
I	21
II	21
III	24
IV	23
V	20
VI	17
Total	**126**

P. V. Sudha

Dean,
Faculty of Informatics
Osmania University

Schemes of Semesters III and IV for AY 2026-27

SEMESTER-III

S. No.	Course Code	Course Title	Hours/ Week		No of Credits	Scheme of Examination			
						Max Marks		Duration (hrs)	
THEORY			**L**	**P**	**Cr**	**SEE**	**CIE**	**SEE**	**CIE**
1	PCC301	Database Systems	3	-	3	70	30	3	1
2	PCC302	Software Engineering	3	-	3	70	30	3	1
3	PCC303	Applied Mathematics	3	-	3	70	30	3	1
4	PCC304	Python Programming	3	-	3	70	30	3	1
5	HSC305	Environmental Science	3	-	3	70	30	3	1
6	**Professional Elective - I**		3	-	3	70	30	3	1
	PEC306	Basics of Data Analytics using Spreadsheet							
	PEC307	Feature Engineering							
	PEC308	Web Programming							
PRACTICAL									
7	LCC301	Database Systems Lab	-	2*2	2	50	25	3	2
8	LCC302	Software Engineering Lab	-	2	1	50	25	3	2
9	LCC304	Python Programming Lab	-	2*2	2	50	25	3	2
10	**Professional Elective – I Lab**		-	2	1	50	25	3	2
	LEC306	Basics of Data Analytics using Spreadsheet Lab							
	LEC307	Feature Engineering Lab							
	LEC308	Web Programming Lab							
		Total	**18**	**8**	**24**	**620**	**280**	-	-

P. V. Sridhar

Dean,
Faculty of Informatics
Osmania University

SEMESTER-IV

SNo	Course Code	Course Title	Hours/ Week		No of Credits	Scheme of Examination			
						Max Marks		Duration (hrs)	
THEORY			**L**	**P**	**Cr**	**SEE**	**CIE**	**SEE**	**CIE**
1	PCC401	Cloud Computing	3	-	3	70	30	3	1
2	PCC402	Computer Networks	3	-	3	70	30	3	1
3	PCC403	Design and Analysis of Algorithm	3	-	3	70	30	3	1
4	PCC404	Artificial Intelligence	3	-	3	70	30	3	1
5	PCC405	Introduction to Machine Learning	3		3	70	30	3	1
6	**Professional Elective - II**		3	-	3	70	30	3	1
	PEC406	Data Visualization							
	PEC407	Advanced Web Programming							
PRACTICALS									
7	LCC402	Computer Networks Lab	-	2	1	50	25	3	2
8	LCC404	Artificial Intelligence Lab	-	2	1	50	25	3	2
9	LCC 405	Introduction to Machine Learning Lab	-	2*2	2	50	25	3	2
10	**Professional Elective – II Lab**		-	2	1	50	25	3	2
	LEC406	Data Visualization Lab							
	LEC407	Advanced Web Programming Lab							
		Total	**18**	**10**	**23**	**620**	**280**	-	-

P. V. Sridhar

Dean,
Faculty of Informatics
Osmania University

Syllabus of Semester III for AY 2026-27

BACHELOR OF COMPUTER APPLICATIONS (BCA)

PCC301 Database Systems

3L - 3 Credits

Course Objectives

After completing this course, students will be able to:

1. Understand the fundamentals of databases, DBMS architecture, data models, and relational database concepts.
2. Apply database design principles and develop Entity-Relationship (ER) models using keys and integrity constraints.
3. Develop SQL and Advanced SQL queries for data definition, manipulation, retrieval, and database programming.
4. Analyse normalization techniques, functional dependencies, and transaction management mechanisms for efficient database design.
5. Explore NoSQL databases, Big Data technologies, database security, and data warehousing concepts for modern data management systems.

Course Outcomes

Upon successful completion of the course, students will be able to:

1. Explain database concepts, DBMS components, data models, and relational schema design principles.
2. Design database schemas and ER diagrams by applying keys, constraints, and database design methodologies.
3. Write SQL and Advanced SQL programs using joins, aggregate functions, analytical queries, stored procedures, functions, and triggers.
4. Apply normalization techniques, functional dependency analysis, concurrency control, and transaction management concepts to database systems.
5. Demonstrate understanding of NoSQL databases, Big Data technologies, database security mechanisms, data warehousing, OLAP, and data mining applications.

CO–PO Articulation Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	3	2	1	2	1	1	1	1	2
CO2	3	3	2	1	1	1	1	1	2
CO3	3	3	3	1	1	1	1	1	2
CO4	3	3	2	1	1	1	1	1	3
CO5	3	2	3	2	3	2	1	2	3

Correlation Levels: **1** – Low Correlation **2** – Medium Correlation **3** – High Correlation

Course Contents

UNIT I: Introduction to Databases and Data Models

Introduction to Databases: Definition of Data, Database, DBMS, Overview of Database Applications, Advantages and Disadvantages of DBMS, Roles of Database Users and Administrators, Introduction to Data Models, Types of Data Models (Hierarchical, Network, Relational, Object-Oriented), Importance of Data Models in DBMS, Introduction to the Relational Model and Relational Schema

P.V. Sudha

UNIT II: Database Design and ER Modelling
Database Design, Keys: Primary Key, Candidate Key, Super Key, Foreign Key, Composite Key, Alternate Key, Unique Key, Surrogate Key, Constraints in a Table: Primary Key, Foreign Key, Unique Key, NOT NULL, CHECK, Entity-Relationship (ER) Model, Entities and Entity Sets, Attributes and Relationships, ER Diagrams, Key Constraints and Weak Entity Sets, Extended ER Features

UNIT III: SQL and Advanced SQL
SQL Basics: DDL and DML, Aggregate Functions (Min(), Max(), Sum(), Avg(), Count()), Logical Operators (AND, OR, NOT), Predicates (LIKE, BETWEEN, ALIAS, DISTINCT), Clauses (GROUP BY, HAVING, ORDER BY, TOP/LIMIT), Inner Join, Natural Join, Full Outer Join, Left Outer Join, Right Outer Join, Equi Join, Advanced SQL: Analytical Queries, Hierarchical Queries, Recursive Queries, Stored Procedures and Functions, Triggers

UNIT IV: Normalization and Transaction Management
Functional Dependencies: Types (Trivial, Non-Trivial, Partial and Full Functional Dependency), Closure of Functional Dependencies, Normal Forms (1NF, 2NF, 3NF, BCNF), Denormalization, ACID Properties, Transactions and Schedules, Concurrent Execution of Transactions, Lock-Based Concurrency Control, Performance of Locking, Introduction to Crash Recovery, 2PL, Serializability, Recoverability, Introduction to Lock Management, Dealing with Deadlocks, Data on External Storage.

UNIT V: NoSQL Databases, Big Data, and Database Security
Introduction to NoSQL: Uses and Features of NoSQL Document Databases, Data Models: Document, Key-Value, Column Family, Graph, CAP Theorem, CRUD Operations, **Overview of Big Data Technologies**: Hadoop, MongoDB Overview, MongoDB Operators, Cassandra, Introduction to Data warehousing, **Introduction to Database Security:** Access Control, Discretionary Access Control.

Suggested Readings

1. Ramakrishnan, Raghu, and Johannes Gehrke. Database Management Systems. 3rd ed., McGraw-Hill, 2018.
2. Mahapatra, R. P., and GovindVerma. Database Management Systems. 1st ed., Khanna Publishing House, 2025.
3. Rosenzweig, Benjamin, and Elena Rakhimov. Oracle PL/SQL by Example. 5th ed., Prentice Hall, 2015.
4. Dayley, Brad. NoSQL with MongoDB in 24 Hours. 1st ed., Sams Publishing, 2024.
5. Korth, Henry F., Abraham Silberschatz, and S. Sudarshan. Database System Concepts. 7th ed., McGraw-Hill, 2019.

Web Resources

1. https://oracle-base.com/articles
2. https://asktom.oracle.com/ords/f?p=100:1:0

P. V. Sudha

PCC 302 Software Engineering

3L - 3 Credits

Course Objectives

After completing this course, students will be able to:

1. Understand software engineering principles, software process models, and Agile development methodologies used in modern software development.
2. Apply requirements engineering techniques and project planning methods for effective software project management.
3. Analyse and design software systems using software architecture concepts and UML modeling techniques.
4. Understand software testing strategies, debugging methods, and software quality metrics for evaluating software products.
5. Explore software quality assurance, release management, maintenance, and product sustenance practices throughout the software lifecycle.

Course Outcomes

Upon successful completion of the course, students will be able to:

1. Explain software engineering concepts, software process models, Agile practices, and software development methodologies.
2. Prepare software requirements specifications and apply project planning, scheduling, and estimation techniques for software projects.
3. Design software systems using architectural models and UML diagrams such as class, sequence, collaboration, use case, and component diagrams.
4. Apply software testing techniques, debugging strategies, and software metrics to assess and improve software quality.
5. Demonstrate understanding of software quality assurance, release planning, software maintenance, product sustenance, and migration strategies in real-world software systems.

CO–PO Articulation Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	3	2	2	1	1	1	1	1	2
CO2	3	3	2	2	1	1	2	3	2
CO3	3	3	3	2	1	1	2	2	2
CO4	3	3	3	1	2	1	1	1	2
CO5	2	2	2	2	3	2	2	2	3

Correlation Levels: **1** – Low Correlation **2** – Medium Correlation **3** – High Correlation

Course Contents

UNIT I: Introduction to Software Engineering and Agile Development

The Evolving Role of Software, Changing Nature of Software, Layered Technology, Process Framework, Process Models: Waterfall Model, Incremental Process Models, Evolutionary Process Models, Unified Process, Agile Software Development, Agility Principles, Agile Methods, Plan-Driven and Agile Development, Extreme Programming (XP), Scrum, Tool Set for the Agile Process

UNIT II: Requirements Engineering and Project Planning
Software Requirements Engineering, Functional and Non-Functional Requirements, Software Requirements Document (SRS), Requirements Specification, Requirements Engineering Processes, Requirements Elicitation and Analysis, Requirements Validation, Requirements Management, Project Planning, Software Pricing, Plan-Driven Development, Project Scheduling, Agile Planning, Estimation Techniques.

UNIT III: Software Design and UML Modelling
Design Process and Design Quality, Design Concepts, Design Model, Software Architecture, Data Design, Architectural Design, Basic Structural Modelling, UML Diagrams: Class Diagrams, Sequence Diagrams, Collaboration Diagrams, Use Case Diagrams, Component Diagrams.

UNIT IV: Software Testing and Product Metrics
Testing Strategies, Strategic Approach to Software Testing, Test Strategies for Conventional Software, Black-Box Testing, White-Box Testing, Validation Testing, System Testing, Art of Debugging, Product Metrics, Software Quality, Metrics for Analysis Model, Metrics for Design Model, Metrics for Source Code, Metrics for Testing, Metrics for Maintenance

UNIT V: Quality Management, Release Management, and Product Sustenance
Quality Concepts, Software Quality Assurance (SQA), Software Reviews, Formal Technical Reviews, Statistical Software Quality Assurance, Software Reliability, Release Management, Release Planning, Development and Build Plans, Release Strategies, Post-Deployment Monitoring, Product Sustenance, Software Maintenance, Updates, End of Life (EOL), Migration Strategies

Suggested Readings :

1. Gill, N. S. Software Engineering. 1st ed., Khanna Publishing House, 2025. AICTE Recommended Textbook.
2. Pressman, Roger S., and Bruce R. Maxim. Software Engineering: A Practitioner's Approach. 9th ed., McGraw Hill Education, 2020.
3. Somerville, Ian. Software Engineering. 9th ed., Pearson Education, 2011.

P.V. Sridhar

PCC303 Applied Mathematics

3L - 3 Credits

Course Objectives

After completing this course, students will be able to:

1. Understand the concepts of partial differentiation and functions of several variables.
2. Apply total differentiation and optimization techniques for solving problems involving maxima and minima.
3. Analyze systems of linear equations and determine linear dependence and independence of vectors and matrices.
4. Understand vector spaces, subspaces, bases, eigenvalues, and eigenvectors in linear algebra.
5. Apply diagonalization techniques and use eigenvalue methods in solving differential equations and related applications.

Course Outcomes

Upon successful completion of the course, students will be able to:

1. Compute partial derivatives and analyze continuity and limits of functions of two variables.
2. Solve problems involving total differentials, implicit functions, and constrained optimization using Lagrange multipliers.
3. Solve systems of linear equations and determine the linear independence or dependence of vectors and matrices.
4. Demonstrate knowledge of vector spaces, bases, eigenvalues, eigenvectors, and characteristic equations.
5. Diagonalize matrices and apply eigenvalue techniques to solve differential equations and engineering applications.

CO–PO Articulation Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	3	3	1	1	1	1	1	1	2
CO2	3	3	1	1	1	1	1	1	2
CO3	3	3	2	1	1	1	1	1	2
CO4	3	2	2	1	1	1	1	1	3
CO5	3	3	2	1	1	2	1	1	3

Correlation Levels: **1** – Low Correlation **2** – Medium Correlation **3** – High Correlation

Course Contents

UNIT – I: Partial Differentiation

Introduction, Functions of Two Variables, Neighborhood of a Point ((a, b)), Continuity of a Function of Two Variables, Continuity at a Point, Limit of a Function of Two Variables, Partial Derivatives, Homogeneous Functions

UNIT– II: Total Differentiation and Optimization

Theorem on Total Differentials, Composite Functions, Differentiation of Composite Functions, Implicit Functions, Maxima and Minima of Functions of Two Variables, Lagrange's Method of Undetermined Multipliers

P.V. Sridhar

UNIT– III: Linear Equations in Linear Algebra
Systems of Linear Equations, Consistent and Inconsistent Systems, Solution Sets of Linear Systems, Trivial and Non-Trivial Solutions, Linear Independence, Linear Independence of Matrix Columns, Characterization of Linearly Dependent Sets

UNIT– IV: Vector Spaces and Eigenvalues
Vector Spaces and Subspaces, Linearly Independent Sets, Bases, Eigenvalues and Eigenvectors, The Characteristic Equation

UNIT– V: Diagonalization and Applications
Diagonalization - Diagonalizing Matrices with Distinct Eigenvalues, Diagonalizing Matrices with Non-Distinct Eigenvalues, Applications to Differential Equations.

Suggested Readings

1. Lay, David C. Linear Algebra and Its Applications. 6th ed., Pearson Education Limited, 2022.
2. Lang, Serge. Introduction to Linear Algebra. 3rd ed., Springer, 2000.
3. Strang, Gilbert. Linear Algebra and Its Applications. 4th ed., Cengage Learning, 2005.
4. Stewart, James. Calculus. 8th ed., Cengage Learning, 2015.
5. Lay, David C., Steven R. Lay, and Judi J. McDonald. Linear Algebra and Its Applications. 6th ed., Pearson, 2021.

P.V. Sridhar

PCC304 Python Programming

3L - 3 Credits

Course Objectives

After completing this course, students will be able to:

1. Understand the fundamentals of Python programming, syntax, operators, data types, and input/output operations.
2. Apply string manipulation techniques and control flow statements for solving programming problems.
3. Develop modular programs using functions, lambda expressions, and user-defined modules.
4. Utilize Python data structures and numerical computing libraries such as NumPy for efficient data processing.
5. Implement file handling, exception handling

Course Outcomes

Upon successful completion of the course, students will be able to:

1. Explain Python programming concepts, program structure, operators, data types, and expressions.
2. Develop Python programs using strings, conditional statements, loops, and nested control structures.
3. Create modular and reusable programs using functions, parameters, lambda functions, and modules.
4. Apply lists, tuples, dictionaries, and NumPy arrays for data manipulation and numerical computations.
5. Design Python applications involving file operations, exception handling

CO–PO Articulation Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	3	2	2	1	1	1	1	1	2
CO2	3	3	2	1	1	1	1	1	2
CO3	3	3	3	1	1	1	1	1	2
CO4	3	3	3	1	1	1	1	1	3
CO5	3	3	3	1	2	1	1	1	3

Correlation Levels: 1 – Low Correlation **2** – Medium Correlation **3** – High Correlation

Course Contents

UNITI: Introduction to Python Programming

Python and its features, History and Application, Areas of Python, Structure of Python Program, Python interpreter and IDE's, Comments, Lines and Indentation, Identifiers and Keywords, Operators and Precedence, Basic Data Types and Type Conversion, Statements and Expressions, Input/Output Statements.

UNIT II: Strings and Control Flow Statements

Creating and Storing Strings, Built-in Functions for Strings, String Operators, String Slicing and Joining, Formatting Strings, Conditional Flow Statements, Loop Control Statements, Nested Control Flow, Continue and Break Statements, Pass and Exit Statements.

P. V. Sridhar

UNIT III: Functions and Modules

Built-in Functions, Function Definition and its composition, Function Call, Fruitful function vs Void function, Scope and Lifetime of Variables, Default Parameters, Command Line Arguments, Lambda Functions, Assert Statement, Importing User-Defined Modules.

UNIT IV: Data Structures and Numerical Computing

Mutable and Immutable Objects, Lists, Tuples and Dictionaries, Tuple Assignment, Commonly Used Functions on Lists, Tuples and Dictionaries, Passing Lists, Tuples and Dictionaries as Arguments to Functions, List Comprehension, Nested List, Using Math Module, Using NumPy Module for Lists of Integers and Arrays.

UNIT V: Files, Exception Handling

Types of Files, Creating, Reading and Writing Text Files, Creating, Reading and Writing Binary Files, Pickle Module, Exception Handling, Try-Except-Else-Finally Block, Raise Statement, Hierarchy of Exceptions, Adding Exceptions.

Suggested Readings

1. Venkatesh, and Nagaraju Y. Introduction to Python Programming. 1st ed., Khanna Publishing House, 2024.
2. Jose, Jeeva. Introduction to Computing & Problem Solving with Python. 1st ed., Khanna Publishing House, 2023.
3. Taneja, Sheetal, and Naveen Kumar. Python Programming: A Modular Approach with Graphics, Database, Mobile and Web Applications. 1st ed., Pearson, 2023.

Web Resources

1. https://www.learnpython.org/
2. https://www.w3schools.com/python/default.asp

P.V-Sridhar

HSC305 Environmental Science

3L –3Credits

Course Objectives

After completing this course, students will be able to:

1. Understand the scope, importance, and significance of environmental studies and natural resource management.
2. Analyze the utilization, conservation, and environmental impact of water, land, and energy resources.
3. Understand ecosystem structure, biodiversity, ecological relationships, and biodiversity conservation methods.
4. Examine various forms of environmental pollution, waste management practices, and environmental protection laws.
5. Explore social and environmental issues, climate change, and disaster management strategies for sustainable development.

Course Outcomes

Upon successful completion of the course, students will be able to:

1. Explain environmental concepts, natural resources, and issues related to water, land, energy resources, and modern agriculture.
2. Describe ecosystem components, ecological interactions, aquatic ecosystems, and biodiversity conservation practices.
3. Identify causes, effects, and control measures of air, water, soil, noise, and thermal pollution along with waste management methods.
4. Demonstrate understanding of environmental protection acts, environmental ethics, climate change, global warming, and resource conservation strategies.
5. Analyze disaster management principles, disaster mitigation techniques, and the impact of disasters on the environment and infrastructure.

CO–PO Articulation Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	2	1	1	2	2	3	1	1	2
CO2	2	2	1	2	2	3	1	1	2
CO3	2	3	1	2	2	3	1	1	2
CO4	2	2	1	2	3	3	1	1	3
CO5	2	3	1	2	2	3	2	2	3

Correlation Levels: **1** – Low Correlation **2** – Medium Correlation **3** – High Correlation

Course Contents

UNIT – I: Introduction to Environmental Studies and Natural Resources

Environmental Studies: Definition, Scope, Importance, Need for Public Awareness. Natural Resources: Concept and Classification. Water Resources: Use and Over-utilization of Surface and Ground Water, Floods and Droughts, Conflicts over Water, Dams – Benefits and Problems. Land Resources: Land as a Resource, Land Degradation, Soil Erosion, Desertification

UNIT – II: Energy Resources and Sustainable Agriculture

Energy Resources: Growing Energy Needs, Renewable and Non-Renewable Energy Sources. Modern Agriculture and Environment: Fertilizer and Pesticide Problems, Water Logging and Salinity. Sustainable Resource Utilization: Conservation and Efficient Use of Resources

P. V. Sudha

UNIT – III: Ecosystems and Ecological Balance
Ecosystems: Concept, Structure, and Functions of Ecosystems. Components of Ecosystems: Producers, Consumers, and Decomposers. Energy Flow: Food Chains, Food Webs, Ecological Pyramids. Aquatic Ecosystems: Ponds, Streams, Lakes, Rivers, Oceans, and Estuaries

UNIT – IV: Biodiversity and Environmental Conservation
Biodiversity: Genetic, Species, and Ecosystem Diversity. Value of Biodiversity: Ecological, Economic, and Social Importance. Threats to Biodiversity: Habitat Loss, Overexploitation, Invasive Species. Conservation of Biodiversity: Endangered and Endemic Species, In-situ and Ex-situ Conservation Methods

UNIT – V: Environmental Pollution, Protection, and Disaster Management
Environmental Pollution: Causes, Effects, and Control Measures of Air, Water, Soil, Noise, andThermal Pollution. Waste Management: Solid Waste Management and Liquid Waste Management. Environmental Protection: Air Act, Water Act, Forest Act, Wildlife Act, Enforcement of Environmental Legislation. Social Issues and Environment: Water Conservation, Watershed Management, Environmental Ethics, Climate Change, Global Warming, Acid Rain, Ozone Layer Depletion. Environmental Disaster Management: Types of Disasters, Environmental and Infrastructure Impacts, Principles of Disaster Mitigation, Disaster Management Methodology, Disaster Management Cycle, Disaster Management in India.

Suggested Readings:

1. De, A. K. Environmental Chemistry. 9th ed., New Age International (P) Ltd., 2019.
2. Odum, Eugene P., and Gary W. Barrett. Fundamentals of Ecology. 5th ed., Brooks/Cole, Cengage Learning, 2005.
3. Joseph, Benny. Environmental Studies. 3rd ed., McGraw Hill Education (India) Pvt. Ltd., 2017.
4. Sharma, V. K. Disaster Management. 1st ed., National Centre for Disaster Management, Indian Institute of Public Administration, 1999.

P. V. Sridhar

PEC306 Basics of Data Analytics using Spreadsheet

3L -3 Credit

Course Objectives

After completing this course, students will be able to:

1. Understand the fundamentals of data analytics, data types, and the role of analytics in decision-making across various domains.
2. Apply data collection, data preparation, and ethical practices for managing and analyzing data effectively.
3. Utilize spreadsheet tools and functions for organizing, managing, and processing data.
4. Perform data analysis using spreadsheet-based techniques such as descriptive statistics, pivot tables, and trend analysis.
5. Create effective visualizations, dashboards, and analytical reports to communicate insights and support decision-making.

Course Outcomes

Upon successful completion of the course, students will be able to:

1. Explain the concepts of data analytics, classify different types of data, and identify applications of analytics in business, healthcare, education, and government sectors.
2. Apply data collection, cleaning, transformation, and ethical data management practices to prepare datasets for analysis.
3. Use spreadsheet software to organize data, perform calculations, and apply mathematical, statistical, logical, and text functions.
4. Analyze datasets using descriptive statistics, pivot tables, conditional formatting, data validation, correlation analysis, forecasting, and what-if analysis techniques.
5. Design data visualizations, interactive dashboards, and analytical reports to effectively communicate insights and support data-driven decision making.

CO–PO Articulation Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	3	2	2	1	1	1	1	1	2
CO2	3	2	3	1	1	1	1	1	2
CO3	3	3	3	1	2	1	1	1	2
CO4	2	2	1	2	3	2	1	1	2
CO5	3	3	2	2	2	2	1	2	3

Correlation Levels: 1 – Low Correlation **2** – Medium Correlation **3** – High Correlation

Course Contents

UNIT I: Introduction to Data Analytics

Understanding Data: Structured Data, Unstructured Data, Semi-structured Data. Data Analytics Fundamentals: introduction to Data Analytics, Types of Data Analytics, Importance of Data Analytics Applications of Data Analytics: Business Applications, Healthcare Applications, Education Applications, Government Applications

P. V. Sudha

UNIT II: Data, Ethics, and Industry Case Studies
Data Collection: Data Collection Methods, Data Sources, Data Formats. Data Preparation: Data Cleaning, Data Transformation, Handling Missing Data, Handling Outliers. Data Ethics and Applications: Ethical Considerations in Data Analytics, Industry-specific Applications, Case Studies

UNIT III: Spreadsheet Fundamentals for Data Analytics
Spreadsheet Basics: Workbook, Worksheet, Rows, Columns, Cells, Data Types. Data Management in Spreadsheets: Data Entry, Formatting, Sorting, Filtering. Spreadsheet Functions: Mathematical Functions, Statistical Functions, Logical Functions, Text Functions.

UNIT IV: Data Analysis using Spreadsheets
Data Summarization: Descriptive Statistics, Measures of Central Tendency, Measures of Dispersion. Data Analysis Tools: Pivot Tables, Conditional Formatting, Data Validation. Trend Analysis: Correlation Analysis, Forecasting Basics, What-if Analysis

UNIT V: Data Visualization and Reporting using Spreadsheets
Data Visualization: Charts, Graphs, Sparklines, Visualization Best Practices. Dashboard Creation: Key Performance Indicators (KPIs), Interactive Dashboards, Report Design. Data Communication: Data Storytelling, Report Generation, Presentation of Analytical Findings

Suggested Readings

1. Jose, Jeeva. Beginner's Guide for Data Analysis Using R Programming. 1st ed., Khanna Publishing House, 2024.
2. Jain, V. K. Data Analytics. 1st ed., Khanna Book Publishing Company, 2024.
3. WebTech Solutions. Mastering Excel. 1st ed., Khanna Publishing House, 2024.
4. Trivedi, M. C. Data Science and Data Analytics Using Python. Khanna Publishing House, 2025.

P. V. Sudha

PEC307 Feature Engineering

3L - 3 Credit

Course Objectives

After completing this course, students will be able to:

1. Understand the importance of features, data types, and pre-processing techniques in machine learning workflows.
2. Apply feature engineering techniques for numerical and categorical data, including feature selection and dimensionality reduction methods.
3. Develop advanced feature extraction and feature construction strategies to improve model performance.
4. Perform feature engineering for specialized data types such as text, time-series, and image data.
5. Evaluate feature quality, design feature engineering pipelines, and apply feature engineering techniques to real-world machine learning applications.

Course Outcomes

Upon successful completion of the course, students will be able to:

1. Explain feature engineering concepts, identify different data types, and apply pre-processing techniques such as data cleaning, normalization, scaling, and transformation.
2. Implement numerical and categorical feature engineering techniques including binning, polynomial features, interaction features, encoding methods, feature selection approaches, and PCA.
3. Apply advanced feature extraction, dimensionality reduction, and feature construction techniques to create informative features for machine learning models.
4. Develop feature engineering solutions for text, time-series, and image datasets using appropriate extraction and transformation methods.
5. Evaluate feature importance, build automated feature engineering pipelines, and apply feature engineering techniques effectively for classification, regression, clustering, and industry-specific machine learning applications while considering ethical implications.

CO–PO Articulation Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	3	2	2	1	1	1	1	1	2
CO2	3	3	3	1	1	1	1	1	2
CO3	3	3	3	1	1	1	1	1	2
CO4	3	3	3	1	2	1	1	1	3
CO5	3	3	3	1	1	1	1	1	3

Correlation Levels: 1 – Low Correlation **2** – Medium Correlation **3** – High Correlation

Course Contents

UNIT I: Introduction to Feature Engineering

Introduction to Data and Features: Importance of Features in Machine Learning, Data Types and Features (Numerical, Categorical, Ordinal, Discrete, Continuous, Interval, Ratio). Basic Feature Pre-processing: Handling Missing Data, Data Cleaning, Feature Scaling, Normalization, Data Transformation

P. V. Sridhar

UNIT II: Feature Engineering Techniques
Numerical Feature Engineering: Binning and Discretization, Polynomial Features, Interaction Features Categorical Feature Engineering: One Hot Encoding, Label Encoding. Feature Selection and Reduction: Feature Extraction vs. Feature Selection, Steps in Feature Selection, Filter Methods, Wrapper Methods, Hybrid Methods, Introduction to Principal Component Analysis (PCA)

UNIT III: Advanced Feature Extraction Techniques
Feature Extraction Methods: Statistical Feature Extraction, Domain-Based Feature Extraction, Automated Feature Extraction. Dimensionality Reduction Techniques: Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Feature Space Transformation. Feature Construction: Creating New Features, Aggregation Features, Derived Features

UNIT IV: Feature Engineering for Specialized Data
Text Feature Engineering: Bag of Words, TF-IDF, N-Grams, Word Embeddings. Time Series Feature Engineering: Lag Features, Rolling Statistics, Date-Time Features, Trend and Seasonality Features Image Feature Engineering: Color Features, Texture Features, Shape Features, Introduction to Deep Feature Extraction

UNIT V: Feature Evaluation and Applications
Feature Importance and Evaluation: Correlation Analysis, Feature Importance Scores, Permutation Importance. Feature Engineering Pipelines: Workflow Design, Automation of Feature Engineering, Best Practices. Applications of Feature Engineering: Feature Engineering for Classification, Regression, Clustering, Real-World Case Studies and Industry Applications, Ethical Considerations in Feature Engineering.

Suggested Readings

1. Trivedi, M. C. Data Science and Data Analytics Using Python Programming. Khanna Publishing House, 2024.
2. Bhaskar, N., and Vasundhara. Machine Learning. Khanna Publishing House, 2024.
3. Jose, Jeeva. Machine Learning. Khanna Publishing House, 2024.
4. Kalita, J. K., D. K. Bhattacharyya, and S. Roy. Fundamentals of Data Science: Theory and Practice. Elsevier, 2023.
5. Trivedi, M. C. Deep Learning and Neural Networks. Khanna Publishing House, 2024.

P. V. Sridhar

PEC308 Web Programming

3L – 3 Credits

Course Objectives

After completing this course, students will be able to:

1. Understand advanced JavaScript concepts including objects, prototypes, functions, closures, and object-oriented programming principles.
2. Apply DOM manipulation, event handling, and form validation techniques to create interactive web applications.
3. Utilize regular expressions, debugging techniques, and development tools to improve code quality and application reliability.
4. Implement JSON processing and jQuery-based solutions for efficient client-side web development.
5. Develop asynchronous web applications using AJAX and Axios for communication with RESTful web services and APIs.

Course Outcomes

Upon successful completion of the course, students will be able to:

1. Develop JavaScript applications using objects, prototypes, closures, constructors, classes, inheritance, abstraction, and polymorphism.
2. Manipulate web page content dynamically using DOM APIs, event handling mechanisms, and client-side form validation techniques.
3. Apply regular expressions, exception handling, debugging methods, and browser development tools to develop robust web applications.
4. Process and exchange data using JSON and implement interactive web interfaces using jQuery selectors, events, animations, and plugins.
5. Build asynchronous web applications using AJAX, Axios, and REST API integration for efficient client-server communication.

CO–PO Articulation Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	3	3	3	1	1	1	1	1	2
CO2	3	3	3	1	1	1	1	1	2
CO3	3	3	3	1	1	1	1	1	3
CO4	3	3	3	1	1	1	1	1	3
CO5	3	3	3	2	1	1	1	1	3

Correlation Levels: 1 – Low Correlation **2** – Medium Correlation **3** – High Correlation

Course Contents

UNIT – I: Advanced JavaScript and OOP

Objects and Prototypes: Object creation, object properties, object methods, prototypes, prototype chaining.Functions and Closures: Function declaration, parameters, invocation, return values, anonymous functions, closures, lexical scope.Object-Oriented Programming in JavaScript: Constructors, classes, encapsulation, inheritance, abstraction, polymorphism.

P. V. Sridhar

UNIT – II: DOM Manipulation and Form Handling
Document Object Model (DOM): DOM hierarchy, HTML DOM, DOM elements, DOM methods, DOM manipulation. Event Handling: Event types, event listeners, event propagation, browser events. Forms and Validation: Forms API, form controls, client-side validation, dynamic form handling.

UNIT – III: Regular Expressions and Debugging
Regular Expressions: Pattern matching, validation patterns, RegEx syntax, search and replace operations. Error Handling: JavaScript errors, exception handling, debugging techniques. Development Tools: Browser Developer Tools, code inspection, performance monitoring, JSLint and code quality practices.

UNIT – IV: JSON and jQuery
JSON: JSON syntax, JSON objects, JSON arrays, JSON files, parsing and serialization. jQuery Fundamentals: Selectors, filters, events, DOM traversal, DOM manipulation. Advanced jQuery: Animation effects, data attributes, templates, utility functions, plugins.

UNIT – V: AJAX and Axios
AJAX: AJAX architecture, XMLHttpRequest, asynchronous communication, AJAX with jQuery, web services integration.Axios HTTP Client: Axios configuration, request handling, response handling, error management, REST API consumption.

Suggested Reading

1. Haverbeke, Marijn. Eloquent JavaScript. 4th ed., No Starch Press, Inc., 2024.
2. Flanagan, David. JavaScript: The Definitive Guide. 7th ed., O'Reilly Media, Inc., 2020.
3. Nixon, Robin. Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5. 6th ed., O'Reilly Media, Inc., 2021.

P. V. Sudha

LCC301 Database Systems Lab

4P - 2 Credit

Course Objectives

After completing this laboratory course, students will be able to:

1. Understand and apply SQL Data Definition Language (DDL) and Data Manipulation Language (DML) commands for database creation and management.
2. Develop SQL queries using predicates, sorting techniques, aggregate functions, and grouping operations for data retrieval and analysis.
3. Perform relational database operations using different types of joins to retrieve and integrate data from multiple tables.
4. Design and implement database programming constructs such as stored procedures and functions for automation of database tasks.
5. Develop and manage database triggers to enforce business rules and maintain database integrity.

Course Outcomes

Upon successful completion of the laboratory course, students will be able to:

1. Create, modify, and manage database tables using DDL commands and perform record manipulation using DML commands.
2. Write SQL queries using predicates, aggregate functions, GROUP BY, HAVING, and ORDER BY clauses to retrieve and analyze data effectively.
3. Implement Inner Join, Natural Join, Equi Join, and Full Outer Join operations for relational database applications.
4. Develop stored procedures and functions to automate database operations and perform customized data processing.
5. Create and manage database triggers for auditing, integrity enforcement, and prevention of unauthorized database modifications.

CO–PO Articulation Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	3	3	3	1	1	1	1	1	2
CO2	3	3	3	1	1	1	1	1	2
CO3	3	3	3	1	1	1	1	1	2
CO4	3	3	3	1	1	1	1	1	3
CO5	3	3	3	1	2	1	1	1	3

Correlation Levels: 1 – Low Correlation **2** – Medium Correlation **3** – High Correlation

Lab Experiments

1. Create a database table using DDL commands with appropriate constraints such as PRIMARY KEY and NOT NULL.
2. Write SQL queries to insert, update, and delete records using DML commands.
3. Demonstrate the use of ALTER, DROP, and TRUNCATE commands on database tables.
4. Retrieve records using predicates such as `LIKE`, `BETWEEN`, `DISTINCT`, and column aliases.
5. Write SQL queries to sort records using the `ORDER BY` clause in ascending and descending order.
6. Write SQL queries using aggregate functions `MIN()`, `MAX()`, `SUM()`, `AVG()`, and `COUNT()`.

P. V. Subba

7. Retrieve grouped data using `GROUP BY` and filter grouped records using the `HAVING` clause.
8. Perform an Inner Join between two related tables and display matching records.
9. Perform a Natural Join operation on two tables with common attributes.
10. Perform an Equi Join on two related tables using equality conditions.
11. Perform a Full Outer Join to retrieve all matching and non-matching records from two tables.
12. Develop a stored procedure to insert and update employee details.
13. Write a stored function to calculate total salary with bonus.
14. Write a trigger to automatically update audit information after record insertion.
15. Develop a trigger to prevent deletion of records from a table.
16. Develop a mini project with creation of complete Databases for – Students enrolment system, Employee Salary management system, Library books storage and distribution system, an perform operations on them.

P. V. Sudheer

LCC302 Software Engineering Lab

2P -1 Credit

Course Objectives

After completing this laboratory course, students will be able to:

1. Understand software engineering concepts, software process models, and software development life cycle methodologies through practical case studies and documentation.
2. Apply requirements engineering techniques to identify, analyze, and document software requirements using industry-standard practices.
3. Design software systems using UML modelling techniques, software architecture, and data design principles.
4. Evaluate software quality through testing techniques, design quality assessment, and software quality assurance practices.
5. Develop professional software engineering artifacts such as SRS documents, test plans, review reports, and quality management documents.

Course Outcomes

Upon successful completion of the laboratory course, students will be able to:

1. Analyze software engineering processes and prepare reports on software evolution, process models, and software development methodologies.
2. Develop Software Requirements Specification (SRS) documents and classify functional and non-functional requirements for software systems.
3. Design UML diagrams including Use Case, Class, Sequence, Collaboration, and Component Diagrams for real-world applications.
4. Apply software testing techniques such as Black-Box Testing, White-Box Testing, Validation Testing, and System Testing to evaluate software quality.
5. Prepare Software Quality Assurance (SQA) plans, conduct Formal Technical Reviews (FTRs), and assess software designs using quality principles and standards.

CO–PO Articulation Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	3	2	2	3	1	1	1	1	2
CO2	3	3	2	3	1	1	1	2	2
CO3	3	3	3	2	1	1	2	2	2
CO4	3	3	3	2	2	1	1	1	2
CO5	2	3	2	2	3	1	2	2	3

Correlation Levels: 1 – Low Correlation **2** – Medium Correlation **3** – High Correlation

Lab Experiments

1. Identify and prepare IEEE format template of SRS and SDS
2. Design a software development lifecycle template using the Unified Process Model.
3. Prepare a Software Requirements Specification (SRS) document for an Online Shopping System.
4. Identify and classify functional and non-functional requirements for a Library Management System.
5. Draw UML Use Case and Class Diagrams for a Student Information System.
6. Create Sequence and Collaboration Diagrams for an ATM Banking System.
7. Design Component Diagrams for an Online Examination System.

P.V. Sudha

8. Develop a data design and architectural design for a Payroll Management System.
9. Analyse the quality of a software design using design quality principles.
10. Prepare Black-Box and White-Box test cases for a Login Authentication Module.
11. Perform Validation Testing and System Testing for a Railway Reservation System.
12. Prepare a Software Quality Assurance (SQA) plan for an Inventory Management System.
13. Conduct a Formal Technical Review (FTR) for a software design document.

P. V. Sridhar

LCC304 Python Programming Lab

4P -2 Credit

Course Objectives

After completing this laboratory course, students will be able to:

1. Understand Python programming fundamentals including identifiers, keywords, operators, input/output operations, and program structure.
2. Apply string manipulation techniques, conditional statements, and loop control structures to solve programming problems.
3. Develop modular Python programs using functions, parameters, and user-defined modules.
4. Utilize Python data structures and NumPy arrays for efficient data storage, manipulation, and computation.
5. Implement exception handling mechanisms to develop robust and error-resistant Python applications.

Course Outcomes

Upon successful completion of the laboratory course, students will be able to:

1. Write Python programs demonstrating identifiers, keywords, operators, user input handling, and formatted output techniques.
2. Develop programs using strings, conditional statements, loops, and control statements such as break, continue, pass, and exit.
3. Create modular applications using user-defined functions, default parameters, keyword arguments, and custom modules.
4. Apply lists, tuples, dictionaries, and NumPy arrays for data manipulation and computational problem solving.
5. Develop Python applications with effective exception handling using try, except, else, finally, and raise statements.

CO–PO Articulation Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	3	2	3	1	1	1	1	1	2
CO2	3	3	3	1	1	1	1	1	2
CO3	3	3	3	1	1	1	1	1	3
CO4	3	3	3	1	1	1	1	1	3
CO5	3	3	3	1	2	1	1	1	3

Correlation Levels: 1 – Low Correlation **2** – Medium Correlation **3** – High Correlation

List of Programming Exercises

1. Write a Python program to perform arithmetic, relational, logical, bitwise, and assignment operations on user-provided data and demonstrate operator precedence and type conversion.
2. Write a Python program to generate a formatted student grade report using input/output statements, formatted printing, and conditional expressions.
3. Write a Python program to perform various string operations including slicing, searching, replacing, counting, reversing, palindrome checking, and formatting.
4. Write a Python program to implement menu-driven applications using conditional statements and nested control structures (e.g., simple calculator or ATM simulation).

P.V. Sridhar

5. Write a Python program to generate mathematical series and number patterns using loops (Prime numbers, Fibonacci series, Armstrong numbers, Perfect numbers, Pascal's Triangle, etc.).
6. Write a Python program to solve real-world problems using nested loops, break, continue, pass, and exit statements.
7. Write a Python program to implement user-defined functions for solving mathematical problems and demonstrate parameter passing, return values, and variable scope.
8. Write a Python program using recursive and lambda functions to solve suitable computational problems and compare their implementation.
9. Write a Python program to create and use user-defined modules for performing common mathematical and utility operations.
10. Write a Python program to demonstrate the use of built-in functions, command-line arguments, assert statements, and exception-safe function execution.
11. Write a Python program to perform various operations on Lists including insertion, deletion, searching, sorting, merging, and list comprehensions.
12. Write a Python program to demonstrate operations on Tuples and Dictionaries for storing, searching, updating, and processing structured information such as student or employee records.
13. Write a Python program to pass Lists, Tuples, and Dictionaries as function arguments and perform suitable data processing operations.
14. Write a Python program using nested lists and list comprehensions to represent and process matrices and tabular data.
15. Write a Python program using the Math module to solve engineering and scientific computation problems.
16. Write a Python program using NumPy to perform array creation, indexing, slicing, reshaping, statistical analysis, and matrix operations.
17. Write a Python program to create, read, write, append, and process text files for maintaining student or employee information.
18. Write a Python program to store and retrieve Python objects using binary files and the Pickle module.
19. Write a Python program to demonstrate comprehensive exception handling using try, except, else, finally, raise statements, and user-defined exceptions.
20. Develop an integrated Python application that combines functions, modules, data structures, file handling, NumPy, and exception handling to solve a real-world problem (e.g., Student Management System, Library Management System, Inventory Management System, Banking System, or Expense Tracker).

P. V. Sridhar

LEC306 Basics of Data Analytics using Spreadsheet Lab

2P -1 Credit

Course Objectives

After completing this laboratory course, students will be able to:

1. Understand the fundamentals of data analytics and identify the role of data-driven decision-making across various application domains.
2. Apply data collection, cleaning, transformation, and ethical data management practices using spreadsheet tools.
3. Utilize spreadsheet functionalities and built-in functions for efficient data organization, processing, and analysis.
4. Perform statistical analysis, trend analysis, and data summarization using spreadsheet-based analytical techniques.
5. Design interactive dashboards, visualizations, and analytical reports to communicate insights effectively.

Course Outcomes

Upon successful completion of the laboratory course, students will be able to:

1. Classify different types of data and analyze real-world problems using descriptive, diagnostic, predictive, and prescriptive analytics approaches.
2. Collect, clean, transform, and prepare datasets while addressing data quality issues and ethical considerations in analytics.
3. Apply spreadsheet functions, sorting, filtering, conditional formatting, and data management techniques to analyze datasets effectively.
4. Perform descriptive statistics, pivot table analysis, correlation analysis, forecasting, and What-If Analysis to derive actionable insights from data.
5. Develop professional dashboards, visualizations, and analytical reports that integrate data storytelling, KPI monitoring, and business recommendations for decision support.

CO–PO Articulation Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	2	2	3	1	1	1	1	1	2
CO2	3	3	3	1	1	1	1	1	2
CO3	3	3	2	1	1	1	1	1	2
CO4	3	3	3	1	1	1	1	1	2
CO5	3	3	3	2	1	1	1	1	3

Correlation Levels:1 – Low Correlation **2** – Medium Correlation **3** – High Correlation

Lab Experiments

1. Identify and classify a given dataset into structured, semi-structured, and unstructured data categories and document the observations.
2. Analyse a real-world dataset and categorize the analytics tasks into descriptive, diagnostic, predictive, and prescriptive analytics.
3. Prepare a short case study report highlighting the application of data analytics in business, healthcare, education, or government sectors.
4. Collect data from multiple sources (CSV files, web data, surveys, etc.) and document the data collection process.

P.V. Sridhar

5. Perform data cleaning operations in a spreadsheet by handling missing values, duplicate records, inconsistent data, and outliers.
6. Analyze an industry-specific dataset and prepare a report discussing ethical considerations, privacy concerns, and business insights.
7. Create and manage a workbook containing multiple worksheets and demonstrate various data formatting techniques.
8. Perform sorting, filtering, and conditional formatting operations on a dataset to extract meaningful information.
9. Use spreadsheet functions to analyse a dataset such as:
    a. Mathematical Functions (SUM, AVERAGE, COUNT)
    b. Statistical Functions (MEDIAN, STDEV)
    c. Logical Functions (IF, AND, OR)
    d. Text Functions (LEFT, RIGHT, CONCAT, LEN)
10. Compute descriptive statistics including mean, median, mode, variance, and standard deviation for a given dataset.
11. Create Pivot Tables and Pivot Charts to summarize and analyze large datasets.
12. Perform correlation analysis, forecasting, and What-If Analysis using spreadsheet tools and interpret the results.
13. Create various charts and graphs (Bar, Line, Pie, Scatter, Histogram) and apply visualization best practices.
14. Design an interactive dashboard using spreadsheet tools to display Key Performance Indicators (KPIs) and business metrics.
15. Prepare a professional analytical report incorporating visualizations, dashboards, and data storytelling techniques to communicate insights.

P.V. Sridhar

LEC307 Feature Engineering Lab

2P - 1 Credit

Course Objectives

After completing this laboratory course, students will be able to:

1. Apply data pre-processing and feature preparation techniques to improve the quality of machine learning datasets.
2. Implement feature engineering methods for numerical and categorical data using appropriate transformation and selection techniques.
3. Develop advanced feature extraction and feature construction solutions to create informative representations of data.
4. Perform feature engineering for specialized data types such as text, time-series, and image datasets.
5. Evaluate feature importance, build automated feature engineering pipelines, and assess the impact of engineered features on machine learning model performance.

Course Outcomes

Upon successful completion of the laboratory course, students will be able to:

1. Identify feature types and perform data pre-processing operations including handling missing values, cleaning data, normalization, standardization, and transformation.
2. Apply numerical and categorical feature engineering techniques such as binning, discretization, polynomial features, interaction features, label encoding, one-hot encoding, feature selection, and PCA.
3. Extract, construct, and transform features using statistical measures, dimensionality reduction techniques, aggregation methods, and domain-specific feature generation approaches.
4. Implement feature engineering methods for text, time-series, and image data using techniques such as TF-IDF, N-Grams, lag features, rolling statistics, and image feature extraction.
5. Evaluate feature importance, develop automated feature engineering pipelines using machine learning frameworks, and demonstrate performance improvements in classification and regression models through effective feature engineering.

CO–PO Articulation Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	3	3	3	1	1	1	1	1	2
CO2	3	3	3	1	1	1	1	1	2
CO3	3	3	3	1	1	1	1	1	2
CO4	3	2	3	2	1	1	1	1	2
CO5	3	3	3	2	2	1	2	2	3

Correlation Levels: **1** – Low Correlation **2** – Medium Correlation **3** – High Correlation

Lab Experiments

1. Analyse a real-world dataset and identify different feature types such as numerical, categorical, ordinal, discrete, and continuous features.
2. Perform data pre-processing by handling missing values, removing duplicates, and cleaning inconsistent data.
3. Apply feature scaling techniques including normalization, standardization, and data transformation, and compare their effects on the dataset.

P.V. Subba

4. Implement numerical feature engineering techniques such as binning, discretization, polynomial feature generation, and interaction features.
5. Apply categorical feature encoding methods including Label Encoding and One-Hot Encoding on a dataset and compare their outputs.
6. Perform feature selection using Filter Methods (Correlation, Chi-Square), Wrapper Methods (Recursive Feature Elimination), and apply PCA for dimensionality reduction.
7. Extract statistical features (mean, median, variance, skewness, kurtosis) from a dataset and analyse their significance.
8. Apply dimensionality reduction techniques such as PCA and Linear Discriminant Analysis (LDA) and compare the transformed feature spaces.
9. Create new features using feature construction techniques including aggregation features, derived features, and domain-specific feature generation.
10. Perform text feature engineering using Bag of Words (BoW), TF-IDF, and N-Grams on a text dataset.
11. Generate time-series features including lag features, rolling statistics, date-time features, trend indicators, and seasonality features.
12. Extract image features such as color histograms, texture descriptors, and shape-based features from image datasets and compare their effectiveness.
13. Evaluate feature importance using correlation analysis, feature importance scores from tree-based models, and permutation importance techniques.
14. Design and implement an automated feature engineering pipeline using Scikit-Learn Pipelines for pre-processing and feature transformation.
15. Develop a machine learning model for classification or regression and compare model performance before and after feature engineering.

P. V. Sudhakar

LEC308 Web Programming Lab

2P –1 Credit

Course Objectives

After completing this laboratory course, students will be able to:

1. Apply advanced JavaScript concepts including objects, prototypes, functions, closures, and object-oriented programming techniques to develop interactive web applications.
2. Design dynamic web interfaces using DOM manipulation, event handling, and client-side form validation mechanisms.
3. Utilize regular expressions, debugging tools, and exception handling techniques to improve the reliability and quality of web applications.
4. Implement JSON processing and jQuery-based programming for efficient client-side development and user interaction.
5. Develop asynchronous web applications using AJAX, Axios, and REST APIs to build modern responsive web solutions.

Course Outcomes

Upon successful completion of the laboratory course, students will be able to:

1. Develop JavaScript programs using objects, prototypes, closures, classes, inheritance, encapsulation, abstraction, and polymorphism.
2. Create interactive web pages through DOM manipulation, event-driven programming, and client-side form validation techniques.
3. Apply regular expressions, exception handling, and browser developer tools for debugging, validation, and performance analysis of web applications.
4. Process and exchange data using JSON and develop interactive user interfaces using jQuery selectors, events, animations, effects, and plugins.
5. Build responsive web applications that integrate AJAX communication, Axios-based REST API consumption, and modern client-side web technologies through real-world mini projects.

CO–PO Articulation Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	3	3	3	1	1	1	1	1	2
CO2	3	3	3	1	1	1	1	1	2
CO3	3	3	3	1	2	1	1	1	3
CO4	3	3	3	1	1	1	1	1	3
CO5	3	3	3	2	1	1	2	2	3

Correlation Levels:1 – Low Correlation **2** – Medium Correlation **3** – High Correlation

Lab Experiments

1. Create JavaScript objects using object literals and constructors. Demonstrate object properties, methods, prototypes, and prototype chaining.
2. Develop JavaScript programs using functions, anonymous functions, arrow functions, and closures to demonstrate lexical scope and function behavior.
3. Implement Object-Oriented Programming concepts in JavaScript including classes, constructors, encapsulation, inheritance, abstraction, and polymorphism.
4. Create a web page that dynamically modifies HTML elements using DOM methods such as getElementById(), querySelector(), and createElement().

5. Develop a web application demonstrating event handling using click, mouse, keyboard, and form events with event listeners.
6. Design a student registration form with client-side validation using JavaScript and dynamic form handling techniques.
7. Develop JavaScript programs using Regular Expressions to validate email addresses, phone numbers, passwords, and URLs.
8. Create a program that demonstrates exception handling using try, catch, finally, and custom error objects.
9. Use Browser Developer Tools to inspect elements, debug JavaScript code, analyze console outputs, and monitor webpage performance.
10. Create and manipulate JSON objects and arrays. Perform JSON parsing and serialization using JSON.parse() and JSON.stringify().
11. Develop web pages using jQuery selectors, filters, event handling, and DOM traversal techniques.
12. Implement advanced jQuery features including animations, effects, templates, plugins, and data attribute manipulation.
13. Create an AJAX application using XMLHttpRequest to retrieve and display data asynchronously from a server or public API.
14. Develop a web application using jQuery AJAX methods ($.get(), $.post(), $.ajax()) to interact with web services.
15. Build a REST API client using Axios to perform GET, POST, PUT, and DELETE operations with proper error handling and response processing.

P. V. Sridhar

Syllabus of Semester IV for AY 2026-27

PCC401 Cloud Computing

3L - 3 Credits

Course Objectives

After completing this course, students will be able to:

1. Understand the evolution of distributed computing and the fundamental concepts of cluster, grid, and cloud computing systems.
2. Analyze cloud computing architectures, service models, deployment models, virtualization technologies, and cloud resources.
3. Explore cloud service delivery models including IaaS, PaaS, SaaS, and XaaS along with their practical applications.
4. Develop an understanding of cloud programming environments and cloud platforms such as AWS, Microsoft Azure, and Google App Engine.
5. Evaluate cloud security, reliability, interoperability, governance, and compliance issues in modern cloud environments.

Course Outcomes

Upon successful completion of the course, students will be able to:

1. Explain the concepts, architectures, and software environments of distributed, cluster, grid, and cloud computing systems.
2. Analyze cloud characteristics, service models, deployment models, virtualization technologies, hypervisors, and cloud resource management techniques.
3. Differentiate and evaluate IaaS, PaaS, SaaS, and XaaS models and their implementations through real-world cloud computing case studies.
4. Apply cloud programming concepts and utilize cloud platforms such as AWS, Microsoft Azure, and Google App Engine for application deployment and management.
5. Assess cloud security, authentication, authorization, fault tolerance, privacy, compliance, interoperability, and cloud federation mechanisms for secure and reliable cloud services.

CO–PO Articulation Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	2	2	1	2	2	1	1	3	2
CO2	2	3	1	2	1	2	1	3	2
CO3	2	3	2	2	1	1	3	3	2
CO4	1	2	1	2	3	1	1	3	2
CO5	2	3	1	2	2	3	2	3	3

Correlation Levels: **1** – Low Correlation **2** – Medium Correlation **3** – High Correlation

Course Contents

UNIT I

History of Centralized and Distributed Computing - Overview of Distributed Computing, Cluster computing, Grid computing. Technologies for Network based systems- System models for Distributed and cloud computing- Software environments for distributed systems and clouds.

P.V. Sridevi

UNIT II

Introduction to Cloud Computing- Cloud issues and challenges - Properties - Characteristics - Service models, Deployment models. Cloud resources: Network and API - Virtual and Physical computational resources - Data-storage. Virtualization concepts - Types of Virtualizations- Introduction to Various Hypervisors - High Availability (HA)/Disaster Recovery (DR) using Virtualization, Moving VMs.

UNIT III

Service models - Infrastructure as a Service (IaaS) - Resource Virtualization: Server, Storage, Network - Case studies. Platform as a Service (PaaS) - Cloud platform & Management: Computation, Storage - Case studies. Software as a Service (SaaS) - Web services - Web 2.0 - Web OS - Case studies – Anything as a service (XaaS).

UNIT IV

Cloud Programming and Software Environments – Parallel and Distributed Programming Paradigms – Programming on Amazon AWS and Microsoft Azure – Containerization using Docker – Container Orchestration using Kubernetes – Emerging Cloud Software Environments.

UNIT V

Cloud Access: authentication, authorization and accounting - Cloud Provenance and meta-data - Cloud Reliability and fault-tolerance - Cloud Security, privacy, policy and compliance- Cloud federation, interoperability and standards.

Suggested Readings

1. RajkumarBuyya et al., Mastering Cloud Computing, 2nd Edition, Morgan Kaufmann (Elsevier), 2026.
2. Erl, Thomas, Monroy, Eric Barceló, et al. Cloud Computing: Concepts, Technology, Security& Architecture. 2nd Edition, Pearson, 2023.
3. Poulton, Nigel. Docker Deep Dive: Zero to Docker in a Single Book. 2025 ed., Nigel Poulton Ltd., 2025.
4. Poulton, Nigel, and PushkarJoglekar. The Kubernetes Book. 3rd ed., Packt Publishing, 2025.

P.V. Sridhar

PCC402 Computer Networks

3L - 3 Credits

Course Objectives

After successful completion of this course, students will be able to:

1. Understand the fundamentals of computer networks, network architectures, transmission media, and the functions of the physical and data link layers.
2. Explain the principles of network layer design, routing algorithms, IP addressing, internetworking, and Internet protocols.
3. Describe the services and protocols of the transport layer, including TCP, UDP, congestion control, and network performance measurement.
4. Understand the architecture and services of the application layer, including DNS, email systems, web technologies, and content delivery.
5. Explain the fundamental concepts of network security, including cryptography, firewalls, intrusion detection systems, and secure communication mechanisms.

Course Outcomes

Upon successful completion of this course, students will be able to:

1. Explain the architecture of computer networks, compare the OSI and TCP/IP reference models, and describe the operation of physical and data link layer technologies.
2. Analyze network layer functions, routing algorithms, IP protocols, and internetworking techniques used in modern computer networks.
3. Demonstrate the operation of transport layer protocols, evaluate congestion control mechanisms, and assess network performance using appropriate metrics.
4. Illustrate the functioning of application layer protocols and services, including DNS, email, HTTP/HTTPS, and web applications.
5. Apply fundamental network security concepts to identify network threats and recommend appropriate mechanisms for secure communication and data protection.

CO–PO Articulation Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	3	2	2	1	1	1	1	1	2
CO2	3	3	3	1	1	1	1	1	2
CO3	3	3	3	1	2	1	1	1	3
CO4	3	2	2	2	1	1	1	1	2
CO5	3	3	3	1	2	2	1	2	3

Correlation Levels: 1 – Low Correlation **2** – Medium Correlation **3** – High Correlation

Course Contents

UNIT-I: Introduction to computer Networks and The Physical Layer, data link layer

Definition and objectives of computer networks, types of computer networks, network technology, network protocols, reference models-OSI and TCP/IP.

Guided transmission media, wireless transmission, using spectrum for transmission, multiplexing, switching. Cellular networks-5G.

Data link layer design issues, Error Detection and correction, elementary data link protocol-ARQ, sliding window.

P. V. Sudha

UNIT-II: The Network layer

Network layer design issues, Routing algorithms in a single network-shortest path algorithm, flooding, distance vector routing, link state routing, broadcast routing, multicast routing.

Traffic management at the network layer-congestion, internetworking-connecting heterogeneous networks, connecting endpoints across heterogeneous networks, packet fragmentation.

The network layer in the internet- IP version 4 protocol, IP addresses, IP version 6, Internet control protocols-ICMP,ARP,DHCP, OSPF, BGP, Internet multicasting.

UNIT-III: The Transport layer

The transport service, elements of transport protocols, congestion control, the internet transport protocols-UDP, TCP. Performance issues-problems in computer networks, network performance measurement, measuring access network throughput.

UNIT-IV: The Application layer

The Domain name system-DNS name space and hierarchy, DNS queries and responses, name resolution, hands on with DNS,DNS privacy. Electronic mail-architectural and services, the user agent, message format, message transfer, final delivery. The world wide web-architectural overview, static web objects, Dynamic web pages and web application, HTTP and HTTPS, web privacy, content delivery.

UNIT-V: Network Security

Fundamentals of network security, the core ingredients of an attack-Overview, firewalls and intrusion detection systems, cryptography, communication security, email security, web security, social issues.

Suggested Readings

1. Tanenbaum, Andrew S., Nick Feamster, and David Wetherall. Computer Networks. 6th ed., Pearson Education, 2021. (Unit-I to Unit-V)
2. Browning, Paul, and Farai Tafa. *Networking Basics*. 2nd ed., Reality Press, 2022. (Unit-I Basics)
3. Kurose, James F., and Keith W. Ross. Computer Networking: A Top-Down Approach. 9th ed., Pearson Education, 2025.
4. Forouzan, Behrouz A. Data Communications and Networking with TCP/IP Protocol Suite. 6th ed., McGraw Hill Education, 2022.
5. Sidhu, Bhavneet. An Integrated Approach to Computer Networks. Khanna Publishing House, 2023.
6. Mastering PC Hardware & Networking. Khanna Publishing House, 2024. (No author specified.)

Web Resources

1. Cisco Networking Academy - Online Courses and Resources
2. NetworkLessons.com - Tutorials on Various Networking Topics

P.V. Sridhar

PCC403 Design and Analysis of Algorithms

3L - 3 Credits

Course Objectives

After completing this course, students will be able to:

1. Understand algorithm design principles, complexity analysis, asymptotic notations, and fundamental data structures.
2. Apply divide and conquer strategies for designing and analyzing efficient algorithms.
3. Analyze and implement greedy algorithms for optimization and graph-related problems.
4. Apply dynamic programming techniques to solve complex optimization and shortest-path problems.
5. Understand and implement backtracking and branch-and-bound techniques for combinatorial optimization problems.

Course Outcomes

Upon successful completion of the course, students will be able to:

1. Analyze algorithm efficiency using time and space complexity measures, asymptotic notations, and fundamental data structures such as stacks, queues, linked lists, and trees.
2. Design and implement divide and conquer algorithms including Binary Search, Merge Sort, Quick Sort, and Strassen's Matrix Multiplication.
3. Apply greedy methods to solve optimization problems such as Knapsack, Minimum Spanning Tree, and Single Source Shortest Path problems.
4. Implement dynamic programming algorithms including Floyd-Warshall, Bellman-Ford, and 0/1 Knapsack for efficient problem solving.
5. Solve computational problems using backtracking and branch-and-bound techniques for applications such as N-Queen, Knapsack, and Travelling Salesman Problems.

CO–PO Articulation Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	3	3	2	1	1	1	1	1	2
CO2	3	3	3	1	1	1	1	1	2
CO3	3	3	3	1	1	1	1	1	2
CO4	3	3	3	1	1	1	1	1	3
CO5	3	3	3	1	1	1	1	1	3

Correlation Levels: 1 – Low Correlation **2** – Medium Correlation **3** – High Correlation

Course Contents

UNIT I: Introduction to Algorithms and Complexity Analysis

Definition of Algorithms, Design and Performance Analysis of Algorithms, Time Complexity, Space Complexity, Asymptotic Notations: Big O (O), Omega (Ω), Theta (Θ), Growth of Functions, Complexity Analysis of Algorithms, Sequential Search, Bubble Sort, Selection Sort, Insertion Sort.

Elementary Data Structures: Stacks, Queues, Linked List, Tree terminology, representation and types of Trees.

P.V. Sridhar

UNIT II: Divide and Conquer
Divide and Conquer Technique: General Concept, Binary Search, Finding Maximum and Minimum, Merge Sort, Quick Sort, Best Case and Worst-Case Analysis, Strassen's Matrix Multiplication, Lower Bound for Comparison-Based Sorting.

UNIT III: Greedy Techniques
General Concept, Knapsack Problem, Minimum Cost Spanning Trees, Prim's Algorithm, Kruskal's Algorithm, Dijkstra's Algorithm for Single Source Shortest Path Problem.

UNIT IV: Dynamic Programming
General Concept, Floyd-Warshall Algorithm for All Pairs Shortest Paths, Bellman-Ford Algorithm, 0/1 Knapsack Problem

UNIT V:Backtracking and Branch and Bound
Backtracking: State Space tree, Branch and Bound condition, Backtracking, N-Queen Problem, 0/1 Knapsack Problem,
Branch and Bound: 0/1 Knapsack Problem, Travelling Salesman Problem.

Suggested Readings

1. Levitin, Anany. Introduction to the Design and Analysis of Algorithms. 4th ed., Pearson Education, 2025.
2. Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to Algorithms. 4th ed., MIT Press, 2022.
3. Sharma, Gajendra. Design & Analysis of Algorithms. 4th ed., Khanna Publishing House, 2019. AICTE Recommended Textbook.
4. Horowitz, Ellis, SartajSahni, and SanguthevarRajasekaran. Fundamentals of Computer Algorithms. 2nd ed., Universities Press (India) Pvt. Ltd., 2008.
5. Kleinberg, Jon, and ÉvaTardos. Algorithm Design. 1st ed., Pearson Education, 2005.

Web Resource

1. https://nptel.ac.in/courses/106101060

P. V. Sridhar

PCC404 Artificial Intelligence

3L - 3 Credits

Course Objectives

After completing this course, students will be able to:

1. Understand the fundamentals of Artificial Intelligence, intelligent agents, rational behavior, and problem-solving approaches.
2. Apply search techniques, game-playing strategies, and constraint satisfaction methods for solving AI problems.
3. Analyze logical reasoning, inference mechanisms, planning approaches, and uncertainty handling in intelligent systems.
4. Explore major AI domains such as machine learning, computer vision, robotics, natural language processing, and expert systems.
5. Examine emerging AI applications, explainable AI, governance frameworks, and future trends in Artificial Intelligence.

Course Outcomes

Upon successful completion of the course, students will be able to:

1. Explain the concepts of Artificial Intelligence, intelligent agents, knowledge-based systems, and problem-solving environments.
2. Implement and analyze uninformed and informed search algorithms, adversarial search techniques, and constraint satisfaction methods.
3. Apply logical reasoning, inference techniques, planning methods, probabilistic reasoning, and fuzzy logic in AI-based problem solving.
4. Demonstrate understanding of AI domains including machine learning, computer vision, robotics, NLP, deep neural networks, and expert systems.
5. Evaluate emerging AI technologies, Explainable AI (XAI), human-AI interaction, governance policies, and the challenges and opportunities of future AI systems.

CO–PO Articulation Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	3	2	2	1	1	1	1	1	2
CO2	3	3	3	1	1	1	1	1	2
CO3	3	3	3	1	1	1	1	1	3
CO4	3	2	3	1	1	1	1	1	3
CO5	2	2	2	2	3	2	1	2	3

Correlation Levels: 1 – Low Correlation **2** – Medium Correlation **3** – High Correlation

Course Contents

UNIT I: Introduction to Artificial Intelligence and Intelligent Agents

Introduction to Artificial Intelligence (AI), Definition and Scope of AI, Intelligent Agents, Agents and Environment, Concept of Rationality, Nature of Environment, Structure of Agents, Knowledge-Based Agents, Introduction to Knowledge-Based Agents, Problem Solving, Problem-Solving Agents.

UNIT II: Search Techniques and Constraint Satisfaction

Uninformed Search Techniques: Depth First Search (DFS), Breadth First Search (BFS), Iterative Deepening Search, Informed Search Techniques: Best First Search, A* Search, AO* Search, Adversarial Search and Games, Two-Player Zero-Sum Games, Minimax Search, Alpha-Beta Pruning, Constraint Satisfaction Problems (CSPs)

P.V. Sridhar

UNIT III: Logical Reasoning, Inference, and Uncertainty
Propositional Logic, First-Order Predicate Logic, Propositional vs First-Order Inference, Unification and Lifting, Inference Techniques, Forward Chaining, Backward Chaining, Resolution, Truth Maintenance Systems, Introduction to Planning, Handling Uncertainty, Non-Monotonic Reasoning, Probabilistic Reasoning, Introduction to Fuzzy Set Theory

UNIT IV: Domains and Applications of Artificial Intelligence
Introduction to Machine Learning, Computer Vision, Robotics, Natural Language Processing (NLP), Deep Neural Networks, Applications of AI Domains, Expert Systems, Architecture of Expert Systems, Role of Expert Systems

UNIT V: Emerging Trends and Future Directions in AI
AI in Healthcare, AI in Education, AI in Cyber Security, AI in Smart Systems and Automation, Explainable Artificial Intelligence (XAI), Human-AI Interaction, AI Governance and Policies, Future Trends in Artificial Intelligence, Challenges and Opportunities in AI Development

Suggested Readings

1. Trivedi, M. C. A Classical Approach to Artificial Intelligence. 1st ed., Khanna Book Publishing Company, 2024.
2. Chopra, Rajiv. Data Science with Artificial Intelligence, Machine Learning and Deep Learning. 1st ed., Khanna Book Publishing Company, 2024.
3. Russell, Stuart J., and Peter Norvig. Artificial Intelligence: A Modern Approach. 4th ed., Pearson Education, 2021.

P. V. Sridhar

PCC405 Introduction to Machine Learning

3L – 3 Credit

Course Objectives

After completing this course, students will be able to:

1. Understand the fundamentals, history, types, and applications of machine learning.
2. Analyze supervised, unsupervised, semi-supervised, and reinforcement learning paradigms and their datasets.
3. Apply machine learning algorithms for regression, classification, and clustering problems.
4. Evaluate machine learning models using performance metrics such as accuracy, precision, recall, F1-score, and AUC.
5. Explore neural networks, support vector machines, ethical considerations, and real-world applications of machine learning.

Course Outcomes

Upon successful completion of the course, students will be able to:

1. Explain machine learning concepts, learning paradigms, datasets, and the machine learning workflow.
2. Apply regression and classification algorithms such as Linear Regression, Logistic Regression, Naïve Bayes, KNN, and Decision Trees to solve prediction problems.
3. Demonstrate understanding of neural networks, perceptron learning, and support vector machines for classification tasks.
4. Implement clustering algorithms such as K-Means, Hierarchical Clustering, and DBSCAN and evaluate clustering performance.
5. Analyze machine learning applications, evaluate model performance using standard metrics, and discuss ethical considerations in machine learning systems.

CO–PO Articulation Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	3	2	2	1	1	1	1	1	2
CO2	3	3	3	1	1	1	1	1	3
CO3	3	3	3	1	1	1	1	1	3
CO4	3	3	3	1	1	1	1	1	3
CO5	3	3	2	2	3	2	1	1	3

Correlation Levels: **1** – Low Correlation **2** – Medium Correlation **3** – High Correlation

Course Contents

UNIT I: Introduction to Machine Learning

Introduction: Definition, History and Applications of Machine Learning, Types of Machine Learning: Supervised, Unsupervised, Semi-Supervised and Reinforcement Learning. Datasets and Learning Framework: Labeled and Unlabeled Datasets, Training, Validation and Testing of ML Models. Performance Evaluation: Confusion Matrix, Accuracy, Precision, Recall, F1 Score, AUC

UNIT II: Supervised Learning and Unsupervised Learning

Regression: Linear Regression, Non-linear Regression, Logistic Regression. Classification: Naïve Bayes, K-Nearest Neighbors, Decision Trees. Linear Models: Artificial Neural Networks, Perceptron Learning Algorithm, Single Layer Perceptron, Support Vector Machines for Linearly Separable Data. Clustering: K-Means, Hierarchical Clustering, DBSCAN, Clustering Validation Measures. ML. Applications: Ethical Considerations, Case Studies and Real-world Applications

P. V. Sudha

UNIT III: Feature Engineering and Dimensionality Reduction

Data Pre-processing: Data Cleaning, Handling Missing Values, Handling Outliers, Data Normalization and Standardization. Feature Engineering: Feature Selection, Feature Extraction, Encoding Categorical Variables. Dimensionality Reduction: Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Applications of Dimensionality Reduction

UNIT IV: Ensemble Learning and Advanced Machine Learning Techniques

Ensemble Methods: Bagging, Boosting, Voting Classifiers, Random Forests. Advanced Support Vector Machines: Kernel Functions, Non-linear Classification using SVM. Model Optimization: Hyper parameter Tuning, Cross Validation, Bias-Variance Tradeoff.

UNIT V: Machine Learning Applications and Model Deployment

Machine Learning Workflow: Model Building, Model Evaluation, Model Selection. ML Applications: Natural Language Processing, Computer Vision, Recommendation Systems, Predictive Analytics. Model Deployment: Introduction to Model Deployment, ML Pipelines, Monitoring and Maintenance of ML Models, Responsible AI and Future Trends in Machine Learning.

Suggested Readings

1. Chopra, Rajiv. Machine Learning and Machine Intelligence. 1st ed., Khanna Publishing House, 2024.
2. Jose, Jeeva. Introduction to Machine Learning. Khanna Publishing House, 2023.
3. Bhaskar, N. Machine Learning. 1st ed., Khanna Publishing House, 2024.

P. V. Sridhar

PEC406 Data Visualization

3L - 3 Credit

Course Objectives

After completing this course, students will be able to:

1. Understand the fundamentals, processes, and significance of data visualization in data-driven decision making.
2. Apply visualization design principles and data storytelling techniques to communicate insights effectively.
3. Create and interpret a variety of charts, graphs, and exploratory visualizations for data analysis.
4. Develop interactive dashboards and business intelligence reports using modern visualization tools.
5. Explore advanced visualization techniques and evaluate their applications in real-world decision-making scenarios.

Course Outcomes

Upon successful completion of the course, students will be able to:

1. Explain the concepts, processes, tools, and applications of data visualization and distinguish among different types of data and visualization approaches.
2. Design effective visualizations by applying principles of visual design, color usage, data modelling, and storytelling techniques.
3. Create and analyze charts, graphs, and exploratory visualizations to identify trends, patterns, relationships, and outliers in datasets.
4. Develop interactive dashboards and business intelligence solutions using tools such as Excel, Tableau, and Power BI to support organizational decision making.
5. Apply advanced visualization techniques, evaluate visualization solutions across domains, and assess emerging trends, ethical considerations, and AI-assisted visualization practices.

CO–PO Articulation Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	2	2	2	2	1	1	1	1	2
CO2	3	3	2	2	1	1	1	1	2
CO3	2	2	3	1	1	1	1	1	2
CO4	3	3	3	2	1	1	1	1	2
CO5	3	3	3	3	1	1	1	2	3

Correlation Levels: 1 – Low Correlation **2** – Medium Correlation **3** – High Correlation

Course Contents

UNIT I: Introduction to Data Visualization and Visualization Tools

Data Visualization Fundamentals: Definition and Importance of Data Visualization, Role of Data Visualization in Decision Making, Types of Data (Numerical, Categorical, Temporal, Geographical) Data Visualization Process: Data Collection, Data Exploration, Data Analysis, Data Visualization, Data Interpretation. Visualization Tools: Challenges and Limitations of Data Visualization, Overview of Visualization Tools (Excel, Tableau, Power BI, Python), Comparing and Contrasting Features and Use Cases among Visualization Tools

P. V, Sudha

UNIT II: Designing Effective Visualizations and Data Storytelling
Visualization Design Principles: Principles of Good Visualization Design, Understanding and Using Color in Visualizations, Importance of Data Modelling in Visualization. Data Storytelling: Narrative and Context, Communicating Insights through Visualizations. Dashboard Design: Best Practices for Dashboard Layout, Interactivity and User Experience

UNIT III: Charts, Graphs, and Exploratory Visualization
Basic Visualizations: Bar Charts, Line Charts, Pie Charts, Area Charts, Histograms. Comparative and Distribution Visualizations: Scatter Plots, Box Plots, Heat Maps, Bubble Charts. Exploratory Data Visualization: Identifying Trends, Patterns, Relationships, and Outliers.

UNIT IV: Dashboard Development and Business Intelligence
Dashboard Components: KPIs, Scorecards, Filters, Slicers, Interactive Elements Business Intelligence Visualization: Reporting Techniques, Executive Dashboards, Operational Dashboards. Visualization using Tools: Creating Dashboards in Excel, Tableau, and Power BI

UNIT V: Advanced Visualization and Real-World Applications
Advanced Visualization Techniques: Geographical Maps, Time-Series Visualizations, Network Visualizations, Infographics. Visualization for Decision Making: Industry Applications in Finance, Marketing, Healthcare, and Operations. Emerging Trends: Visual Analytics, AI-Assisted Visualization, Ethical Considerations in Data Visualization, Case Studies and Real-World Applications.

Suggested Readings

1. Knaflic, Cole Nussbaumer. Storytelling with Data: A Data Visualization Guide for Business Professionals. 10th Anniversary ed., John Wiley & Sons, Inc., 2025.
2. Trivedi, M. C. Data Science and Data Analytics Using Python. 1st ed., Khanna Publishing House, 2022.
3. Ferrari, Alberto, and Marco Russo. Analyzing Data with Power BI and Power Pivot for Excel. 1st ed., Microsoft Press, 2017.
4. Knight, Devin, Brian Knight, Mitchell Pearson, and Manuel Quintana. Microsoft Power BI Complete Reference. 1st ed., Packt Publishing, 2018.

Web Resources

1. https://learn.microsoft.com/en-us/power-bi/
2. https://www.storytellingwithdata.com/

P. V. Sridhar

PEC407 Advanced Web Programming

3L – 3 Credits

Course Objectives

After completing this course, students will be able to:

1. Understand Node.js architecture, modern JavaScript (ES6) features, and server-side JavaScript programming concepts.
2. Apply asynchronous programming techniques, module management, and package management tools for scalable application development.
3. Develop server-side applications using Node.js core modules, HTTP services, and the Express.js framework.
4. Design interactive user interfaces using React components, props, state management, and lifecycle methods.
5. Build modern web applications using advanced React concepts and Redux for efficient application state management.

Course Outcomes

Upon successful completion of the course, students will be able to:

1. Develop Node.js applications using ES6 features, modules, and command-line programming techniques.
2. Implement asynchronous programming patterns using callbacks, promises, async/await, and effectively manage project dependencies using npm.
3. Create server-side web applications using the File System module, HTTP module, and Express.js framework with routing and middleware support.
4. Design and develop React applications using components, props, state management, lifecycle methods, event handling, and Context API.
5. Build scalable front-end applications using advanced React features, forms, controlled components, Redux architecture, and centralized state management techniques.

CO–PO Articulation Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	3	3	3	1	1	1	1	1	2
CO2	3	3	3	1	1	1	1	1	2
CO3	3	3	3	1	1	1	1	1	3
CO4	3	3	3	1	1	1	1	1	3
CO5	3	3	3	2	1	1	2	1	3

Correlation Levels: **1** – Low Correlation **2** – Medium Correlation **3** – High Correlation

Course Contents

UNIT – I: Node.js Fundamentals

Introduction to Node.js: Browser JavaScript vs Node.js, Node.js architecture, REPL environment. Modern JavaScript (ES6): Let and const, arrow functions, template literals, destructuring, modules. Node.js Programming: Writing Node.js applications, execution environment, command-line applications.

P. V. Sridhar

UNIT – II: Asynchronous Programming and Modules
Asynchronous Programming: Callbacks, promises, async and await, event loop, timers. Node.js Modules: Module system, exports, require, custom modules. Package Management: npm, package.json, package-lock.json, local and global packages, dependency management.

UNIT – III: File System, HTTP and Express.js
File System Module: File creation, reading, writing, synchronous and asynchronous operations. HTTP Module: HTTP server creation, request handling, response generation. Express.js Framework: Express architecture, routing, middleware, request and response objects, template engines, view rendering.

UNIT – IV: React Fundamentals
Introduction to React: React architecture, React elements, virtual DOM. Components and Props: Functional components, class components, component composition, props. State and Lifecycle: State management, lifecycle methods, event handling, conditional rendering, Context API.

UNIT – V: Advanced React and Redux
React Forms and Components: Lists and keys, forms, controlled components, refs, lifting state up. Advanced React Concepts: Error boundaries, composition vs inheritance, thinking in React. Redux: Actions, reducers, store, React-Redux integration, application state management.

Suggested Readings

1. Banks, Alex, and Eve Porcello. Learning React: Modern Patterns for Developing React Apps. 2nd ed., O'Reilly Media, Inc., 2020.
2. Boduch, Adam. React and React Native. 4th ed., Packt Publishing Ltd., 2020.
3. Brown, Ethan. Web Development with Node and Express: Leveraging the JavaScript Stack. 2nd ed., O'Reilly Media, Inc., 2019.
4. Wandschneider, Marc. Learning Node.js. 2nd ed., Addison-Wesley Professional, 2017.

Online References

1. Node.js Documentation
2. Express.js Documentation
3. React Documentation
4. Redux Documentation
5. npm Documentation

P.V. Sridhar

LCC402 Computer Networks Lab

2P - 1 Credit

Course Objectives

Upon successful completion of this laboratory course, students will be able to:

1. Understand and implement fundamental data communication techniques, including error detection and reliable data transmission protocols.
2. Develop network programming applications using TCP and UDP socket programming for client–server communication.
3. Apply IP addressing concepts and networking utilities to perform subnet calculations, domain name resolution, and network performance analysis.
4. Design and implement applications that interact with Internet services using standard protocols such as HTTP and DNS.
5. Develop secure and diagnostic networking applications by implementing port scanning and secure communication techniques.

Course Outcomes

Upon successful completion of this laboratory course, students will be able to:

1. Implement error detection and reliable data transmission protocols to ensure accurate communication over computer networks.
2. Develop and test TCP and UDP client–server applications using Python socket programming.
3. Apply networking concepts to perform IPv4 subnetting, domain name resolution, and evaluate network performance metrics.
4. Build applications that communicate with web servers using HTTP requests and other Internet protocols.
5. Design basic network security and diagnostic tools, including port scanners and secure communication applications, while following standard networking practices.

CO–PO Articulation Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	3	2	2	1	1	1	1	1	2
CO2	3	3	3	1	1	1	1	1	2
CO3	3	3	3	1	1	1	1	1	2
CO4	3	2	3	1	1	1	1	1	2
CO5	3	3	3	2	2	1	2	2	3

Correlation Levels: 1 – Low Correlation **2** – Medium Correlation **3** – High Correlation

Lab Experiments

(Implement the following program using any of the programming language of previous semesters)

1. Write a program to implement Cyclic Redundancy Check (CRC) for error detection in data transmission.
2. Write a program to simulate Stop-and-Wait ARQ and Sliding Window Protocol for reliable data transmission.
3. Write a program to calculate the Network Address, Broadcast Address, First Host, Last Host, and Number of Hosts for a given IPv4 address and subnet mask.

P. V. Sridhar

4. Write programs to implement TCP Client-Server communication using socket programming.
5. Write programs to implement UDP Client-Server communication using socket programming.
6. Write a program to measure network response time and throughput.
7. Write a program to resolve domain names into IP addresses.
8. Write a program to send HTTP GET and POST requests and display the server response using the requests library.
9. Write a program to implement a basic TCP port scanner for identifying open ports on a host.
10. Write a program to implement a secure communication technique

P. V. Sridhar

LCC404 Artificial Intelligence Lab

2P – 1 Credit

Course Objectives

After completing this laboratory course, students will be able to:

1. Understand and implement intelligent agents, knowledge-based systems, and problem-solving approaches in Artificial Intelligence.
2. Apply uninformed, informed, and adversarial search algorithms to solve AI-based search and game problems.
3. Develop knowledge representation and reasoning systems using logic and inference techniques.
4. Implement uncertainty handling methods including probabilistic reasoning and fuzzy logic for intelligent decision making.
5. Build and evaluate basic machine learning models for classification and prediction applications.

Course Outcomes

Upon successful completion of the laboratory course, students will be able to:

1. Design and simulate intelligent agents, analyze different environments, and develop rule-based knowledge systems.
2. Implement and compare search algorithms such as BFS, DFS, Iterative Deepening Search, Best First Search, A*, Minimax, and Alpha-Beta Pruning.
3. Represent knowledge using propositional and predicate logic and apply forward chaining, backward chaining, unification, and resolution techniques.
4. Develop AI solutions for uncertain environments using probabilistic reasoning and fuzzy inference mechanisms.
5. Build and evaluate machine learning models for classification and prediction tasks using appropriate AI techniques and tools.

CO–PO Articulation Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	3	2	2	1	1	1	1	1	2
CO2	3	3	3	1	1	1	1	1	3
CO3	3	3	2	1	1	1	1	1	3
CO4	3	3	3	1	1	1	1	1	3
CO5	3	3	3	1	2	1	1	1	3

Correlation Levels: 1 – Low Correlation **2** – Medium Correlation **3** – High Correlation

Lab Experiments

1. Design and simulate an intelligent agent for a simple environment.
2. Compare different types of environments based on observability, determinism, and dynamics.
3. Develop a simple knowledge-based agent using rule-based representation.
4. Design a problem-solving agent for a real-world search problem.
5. Implement Breadth First Search (BFS) and Depth First Search (DFS) algorithms.

P. V. Sridhar

6. Develop a program for Iterative Deepening Search and compare its performance with BFS and DFS.
7. Implement Best First Search and A* Search algorithms for path finding problems.
8. Develop a Minimax algorithm for a two-player zero-sum game.
9. Implement Alpha-Beta Pruning to optimize adversarial search.
10. Represent knowledge using propositional logic and predicate logic statements.
11. Implement forward chaining and backward chaining inference techniques.
12. Develop a program for unification and resolution in First-Order Logic.
13. Implement probabilistic reasoning for uncertain decision-making problems.
14. Demonstrate fuzzy set operations and fuzzy inference mechanisms.
15. Develop a simple machine learning model for classification or prediction tasks.

P. V. Sridhar

LEC405 Introduction to Machine Learning Lab

4P – 2 Credits

Course Objectives

After completing this laboratory course, students will be able to:

1. Apply machine learning workflows including data exploration, dataset preparation, model training, and performance evaluation.
2. Develop and compare supervised and unsupervised learning models for classification, regression, and clustering tasks.
3. Perform data pre-processing, feature engineering, and dimensionality reduction to improve model effectiveness.
4. Implement ensemble learning methods, support vector machines, and model optimization techniques for enhanced predictive performance.
5. Build, evaluate, and deploy machine learning applications for real-world problem-solving using modern ML tools and frameworks.

Course Outcomes

Upon successful completion of the laboratory course, students will be able to:

1. Perform exploratory data analysis, prepare datasets, and evaluate machine learning models using appropriate performance metrics such as Accuracy, Precision, Recall, F1-Score, and ROC-AUC.
2. Implement and compare supervised learning algorithms including Linear Regression, Logistic Regression, Naïve Bayes, KNN, and Decision Trees, as well as clustering techniques such as K-Means, Hierarchical Clustering, and DBSCAN.
3. Apply data pre-processing, feature engineering, feature selection, feature extraction, PCA, and LDA techniques to improve machine learning model performance.
4. Develop optimized machine learning solutions using ensemble learning methods, SVMs with different kernels, hyper parameter tuning, cross-validation, and bias-variance analysis.
5. Design end-to-end machine learning pipelines, build domain-specific ML applications, and deploy trained models using Streamlit for real-time prediction and decision support.

CO–PO Articulation Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	3	3	3	1	1	1	1	1	3
CO2	3	3	3	1	1	1	1	1	3
CO3	3	3	3	1	1	1	1	1	3
CO4	3	3	3	1	1	1	1	1	3
CO5	3	3	3	2	2	1	1	1	3

Correlation Levels: **1** – Low Correlation **2** – Medium Correlation **3** – High Correlation

Lab Experiments

1. Perform Exploratory Data Analysis (EDA) on a real-world dataset and identify features, target variables, and data characteristics.
2. Split a dataset into training, validation, and testing sets and compare model performance under different data split ratios.
3. Train a classification model and evaluate its performance using Confusion Matrix, Accuracy, Precision, Recall, F1-Score, and ROC-AUC metrics.
4. Implement Linear Regression and Logistic Regression on suitable datasets and analyze

P.V. Sridhar

prediction accuracy.

5. Develop classification models using Naïve Bayes, K-Nearest Neighbors (KNN), and Decision Trees and compare their performance.
6. Apply K-Means, Hierarchical Clustering, and DBSCAN clustering algorithms on a dataset and evaluate cluster quality using clustering validation measures.
7. Perform data preprocessing by handling missing values, outliers, normalization, and standardization on a real-world dataset.
8. Implement feature engineering techniques including feature selection, feature extraction, and categorical variable encoding.
9. Apply Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) for dimensionality reduction and compare model performance before and after reduction.
10. Implement ensemble learning methods such as Bagging, Random Forest, and Voting Classifiers and compare their accuracy with individual classifiers.
11. Develop a Support Vector Machine (SVM) classifier using different kernel functions (Linear, Polynomial, RBF) and analyze the results.
12. Perform Hyper parameter Tuning using Grid Search or Random Search along with Cross Validation and study the Bias-Variance Tradeoff.
13. Develop an end-to-end machine learning pipeline including preprocessing, training, evaluation, and prediction.
14. Build a machine learning application in one of the following domains:
    a. Natural Language Processing (Sentiment Analysis)
    b. Computer Vision (Image Classification)
    c. Recommendation Systems
    d. Predictive Analytics
15. Deploy a trained machine learning model using Streamlit and create a user interface for real-time predictions.

P. V. Sridhar

LEC406 Data Visualization Lab

2P - 1 Credit

Course Objectives

After completing this laboratory course, students will be able to:

1. Apply data exploration and visualization techniques using modern visualization tools to analyze real-world datasets.
2. Design effective visualizations and data storytelling solutions that communicate insights clearly and accurately.
3. Create and interpret a variety of charts, graphs, and exploratory visualizations for pattern discovery and data analysis.
4. Develop interactive dashboards and business intelligence solutions using industry-standard visualization tools.
5. Implement advanced visualization techniques to support decision-making across diverse application domains.

Course Outcomes

Upon successful completion of the laboratory course, students will be able to:

1. Collect, explore, analyze, and visualize datasets using tools such as Excel, Tableau, Power BI, and Python, while selecting appropriate visualization techniques for different data types.
2. Design effective visualizations, apply visualization best practices, and develop data storytelling presentations that communicate analytical insights to diverse audiences.
3. Create and interpret basic, comparative, and exploratory visualizations to identify trends, patterns, correlations, relationships, and outliers in datasets.
4. Develop interactive dashboards incorporating KPIs, scorecards, filters, slicers, drill-down features, and business intelligence reporting capabilities.
5. Apply advanced visualization methods including geographical mapping, time-series analysis, forecasting dashboards, and domain-specific visual analytics to support informed decision-making in real-world scenarios.

CO–PO Articulation Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	2	2	3	1	1	1	1	1	2
CO2	3	3	3	1	1	1	1	1	2
CO3	3	3	3	1	1	1	1	1	3
CO4	3	2	3	2	1	1	1	1	2
CO5	3	3	3	2	1	1	1	2	3

Correlation Levels: **1** – Low Correlation **2** – Medium Correlation **3** – High Correlation

Lab Experiments

2. Collect a real-world dataset and classify its attributes as numerical, categorical, temporal, and geographical data. Prepare a report on suitable visualization techniques for each type.
3. Perform data exploration and analysis using Excel or Python and create basic visualizations to summarize the dataset.
4. Compare visualization tools (Excel, Tableau, Power BI, and Python) by creating the same chart in each tool and documenting their features, advantages, and limitations.
5. Design a set of visualizations demonstrating the principles of effective visualization, including appropriate use of colors, labels, scales, and chart types.

P. V. Sudha

6. Create a data storytelling presentation using visualizations to communicate insights from a business, healthcare, or education dataset.
7. Design an interactive dashboard layout with user-friendly navigation, filters, and visual elements based on a selected dataset.
8. Create and compare various basic visualizations including Bar Charts, Line Charts, Pie Charts, Area Charts, and Histograms for a given dataset.
9. Develop comparative and distribution visualizations such as Scatter Plots, Box Plots, Heat Maps, and Bubble Charts and interpret the results.
10. Perform exploratory data visualization to identify trends, patterns, relationships, correlations, and outliers in a real-world dataset.
11. Create KPI indicators, scorecards, slicers, and filters for a business dataset using Excel, Tableau, or Power BI.
12. Design and develop an Executive Dashboard and an Operational Dashboard for the same dataset and compare their objectives and visual components.
13. Build an interactive business intelligence dashboard using Tableau or Power BI with drill-down and filtering capabilities.
14. Create geographical visualizations using map charts to analyze regional or location-based data.
15. Develop time-series visualizations and forecasting dashboards to analyze trends and future predictions.
16. Analyse a real-world case study from finance, healthcare, marketing, or operations and create advanced visualizations to support decision-making.

P.V. Sridhar

LEC407 Advanced Web Programming Lab

2P - 1 Credit

Course Objectives

After completing this laboratory course, students will be able to:

1. Develop server-side applications using Node.js fundamentals, modern JavaScript features, modular programming, and package management tools.
2. Apply asynchronous programming concepts and Node.js modules to build efficient and scalable applications.
3. Design and implement web servers and backend applications using Node.js core modules and the Express.js framework.
4. Create interactive front-end applications using React components, state management, lifecycle methods, and Context API.
5. Build full-stack web applications using React, Redux, Express.js, and API integration techniques for modern web development.

Course Outcomes

Upon successful completion of the laboratory course, students will be able to:

1. Develop Node.js applications utilizing ES6 features, modular programming techniques, command-line processing, and npm package management.
2. Implement asynchronous programming using callbacks, promises, async/await, event loops, timers, and custom modules.
3. Create backend applications using the File System module, HTTP module, and Express.js framework with routing, middleware, and template rendering capabilities.
4. Design React applications using functional and class components, props, state management, event handling, lifecycle methods (or Hooks), and Context API.
5. Build complete full-stack applications incorporating Redux-based state management, REST API integration, CRUD operations, Axios communication, and responsive user interfaces.

CO–PO Articulation Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	3	3	3	1	1	1	1	1	2
CO2	3	3	3	1	1	1	1	1	2
CO3	3	3	3	1	1	1	1	1	3
CO4	3	3	3	1	1	1	1	1	3
CO5	3	3	3	2	1	1	2	2	3

Correlation Levels: **1** – Low Correlation **2** – Medium Correlation **3** – High Correlation

Lab Experiments

1. Write a Node.js program demonstrating the use of ES6 features such as let, const, arrow functions, template literals, destructuring, and modules.
2. Develop a command-line Node.js application that accepts user input and performs arithmetic and string operations.
3. Create a modular Node.js application using separate files and demonstrate importing/exporting functionality using ES6 modules or Common JS modules.
4. Implement asynchronous programming using callbacks, promises, and async/await to perform sequential and parallel tasks.

P. N. Sudha

5. Develop a Node.js application that demonstrates the Event Loop, timers (set Timeout, set Interval), and asynchronous execution behavior.
6. Create a custom Node.js module and manage dependencies using npm, including package installation and package configuration using package.json.
7. Develop a Node.js application using the File System module to create, read, update, and delete files using both synchronous and asynchronous methods.
8. Create an HTTP server using Node.js that handles different request URLs and returns appropriate responses.
9. Build a web application using Express.js that demonstrates:
   - Routing
   - Middleware
   - Request and Response handling
   - Template rendering using EJS/Pug/Handlebars
10. Create a React application using functional and class components and demonstrate component composition using props.
11. Develop a React application that manages user input using state, event handling, and conditional rendering.
12. Build a React application demonstrating lifecycle methods (or Hooks), Context API, and dynamic data sharing between components.
13. Create a React form application using controlled components, validation, refs, and lifting state up between components.
14. Develop a React application that displays dynamic lists, handles errors using Error Boundaries, and demonstrates component composition.
15. Build a Redux-based React application that implements:
    - Actions
    - Reducers
    - Store creation
    - React-Redux integration
    - Centralized state management

P. V. Sridhar

LCC408 Design Thinking and Innovation Lab

2P - 1 Credit

Course Objectives

After completing this laboratory/activity-based course, students will be able to:

1. Apply Design Thinking principles to identify, analyze, and define real-world problems from a user-centric perspective.
2. Develop empathy and customer understanding through user research, interviews, and experience mapping techniques.
3. Generate innovative solutions using ideation, brainstorming, systems thinking, and creative problem-solving methods.
4. Design, test, evaluate, and refine prototypes based on user feedback and usability testing.
5. Develop implementation and business strategies for innovative products, services, and solutions using design and innovation frameworks.

Course Outcomes

Upon successful completion of the course, students will be able to:

1. Identify and define real-world problems using Design Thinking methodologies and user-centered problem analysis techniques.
2. Conduct empathy studies, customer journey mapping, and stakeholder analysis to understand user needs, expectations, and challenges.
3. Generate, evaluate, and prioritize innovative solutions using brainstorming, mind mapping, systems thinking, and value proposition tools.
4. Develop, test, and refine prototypes through usability testing, feedback analysis, and iterative design improvement processes.
5. Present feasible and innovative solutions supported by implementation plans, business model canvases, deployment strategies, and impact assessments.

CO–PO Articulation Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	2	3	2	2	1	1	2	2	3
CO2	2	3	2	2	1	1	2	2	3
CO3	2	3	3	2	1	1	2	2	3
CO4	2	3	2	2	1	1	2	2	3
CO5	2	3	2	3	1	1	2	2	3

Correlation Levels: **1** – Low Correlation **2** – Medium Correlation **3** – High Correlation

Lab Experiments

1. Identify a real-world problem faced by students, customers, or society and document the problem using Design Thinking principles.
2. Conduct an empathy study by interviewing at least five users, recording their challenges, needs, and expectations, and preparing an Empathy Map.
3. Analyse a global success story (Airbnb, Apple, IDEO, or Netflix) and present how Design Thinking contributed to its innovation and success.
4. Use brainstorming and mind-mapping techniques to generate multiple innovative solutions for an identified problem.

P. V. Sridevi

5. Create a Customer Journey Map (Customer Experience Map) for a selected product or service and identify pain points and improvement opportunities.
6. Develop a low-fidelity prototype (paper prototype, sketch, storyboard, or wireframe) for the proposed solution and present it to peers.
7. Apply the "How Might We" technique to redefine a wicked problem and generate alternative solution statements.
8. Use systems thinking tools to analyse stakeholders, dependencies, and impacts associated with a selected societal or business problem.
9. Prepare a Value Proposition Canvas showing how the proposed solution addresses user needs, gains, and pain points.
10. Conduct usability testing of the developed prototype with a group of users and collect structured feedback.
11. Analyse user feedback and identify design improvements using affinity diagrams and prioritization techniques.
12. Refine and redesign the prototype based on feedback and prepare a comparative report showing the improvements made.
13. Develop an implementation plan for the proposed solution, including resources, stakeholders, risks, and deployment strategy.
14. Create a business model canvas or innovation canvas for the proposed product/service and evaluate its feasibility.
15. Present the final innovative solution, prototype, implementation strategy, and expected impact before a review panel or class.

P. V. Sridhar

Model Question Paper Format for CIE

BACHELOR OF COMPUTER APPLICATIONS (BCA)

Osmania University

B.C.A.-- Semester Examination, -------

Subject: -------

Time: 1 Hours **Max. Marks: 20**

NOTE: i)Answer all questions from Part – A, & any five questions from Part – B, choosing one question from each unit.

ii) BT-Bloom's Taxonomy and CO-Course Outcome

PART – A (5Q x 2M = 10 Marks)

S No		Question	Marks	BT	CO
1.	a	Theory/Problem			
	b	Theory/Problem			
	c	Theory/Problem			
	d	Theory/Problem			
	e	Theory/Problem			

PART – B (2Q x 5M = 10 Marks)

S No		Question	Marks	BT	CO
2.	a	Theory			
	b	Theory/Problem			
OR					
3.	a	Theory			
	b	Theory/Problem			
4.	a	Theory			
	b	Theory/Problem			
OR					
5.	a	Theory			
	b	Theory/Problem			

P. V. Sridhar

Model Question Paper Format for SEE

BACHELOR OF COMPUTER APPLICATIONS (BCA)

Code No:

Osmania University

B.C.A.-- Semester Examination, -------

Subject: -------

Time: 3 Hours **Max. Marks: 70**

NOTE: i)Answer all questions from Part – A, & any five questions from Part – B, choosing one question from each unit.

ii) BT-Bloom's Taxonomy and CO-Course Outcome

PART – A (10Q x 2M = 20 Marks)

SNo	Question	Marks	BT	CO
1. a	Unit- I			
b	Unit- I			
c	Unit- II			
d	Unit- II			
e	Unit- III			
f	Unit- III			
g	Unit- IV			
h	Unit- IV			
i	Unit- V			
j	Unit- V			

PART – B (5Q x 10M = 50 Marks)

SNo	Question	Marks	BT	CO
Unit-I				
2a	Theory			
2b	Theory/Problem			
	OR			
3a	Theory			
3b	Theory/Problem			
Unit-II				
4a	Theory			
4b	Theory/Problem			
	OR			
5a	Theory			
5b	Theory/Problem			
Unit-III				
6a	Theory			
6b	Theory/Problem			
	OR			
7a	Theory			
7b	Theory/Problem			

P. V. Sudheer

Unit-IV				
8a	Theory			
8b	Theory/Problem			
	OR			
9a	Theory			
9b	Theory/Problem			
Unit-V				
10a	Theory			
10b	Theory/Problem			
	OR			
11a	Theory			
11b	Theory/Problem			

P. V. Sridevi