Bachelor of Computer Applications

(Cloud Technology and Information Security)

Semester–III

Course Code	Course Title	L	T	P	Credits
BCA-201A	**Software Engineering** (Pre-requisite: None)	3	0	0	3
		CIE	**SEE**		**Total**
		40	60		100

Course Outcomes (COs): At the end of this course, students will be able to	
BCA-201A.1	Describe the phases of software development life cycle for designing an efficient software.
BCA-201A.2	Analyze requirement techniques like Data flow diagram, Entity relationship diagram, Object diagram
BCA-201A.3	Identification of user requirements using various requirements elicitation techniques
BCA-201A.4	Describe the basics of software design using various techniques.

Course Outcomes (CO) to Program Outcomes (PO) mapping (scale 1: Low, 2: Medium, 3: High)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
BCA-201A.1	3	-	-	-	-	-	-	-
BCA-201A.2	3	3	-	2	-	-	-	2
BCA-201A.3	3	-	-	2	-	-	-	2
BCA-201A.4	3	2	3	2	-	-	2	2

Instructions for Paper Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

UNIT-I **Contact Hours: 10**

Introduction to Software Engineering

Introduction: Software, Program, Software Crisis and Solutions, Software Evolution, Software Development Paradigm, Need of Software Engineering, Software Processes, Software Characteristics, Attributes of Software Product, Software Myths, Software basic terminologies, Characteristics of good software.

UNIT-II **Contact Hours: 13**

Software life cycle, Requirements Analysis and Specifications

Software life cycle models: Build and Fix, Waterfall, Prototype, Iterative Enhancement, Evolutionary Development Model, Prototyping, and Spiral Model.

Software Requirements Analysis and Specifications: Types of Requirements. Requirement Elicitation: Interview, Brainstorming, Quality Functional Deployment, Use Case Approach. Problem Analysis – DFD, Data dictionaries, ER diagrams, object diagrams, approaches to problems analysis, SRS, specifying behavioural

& non-behavioural requirements.

UNIT-III **Contact Hours: 12**

Software Design

Software Design: Design framework, Conceptual and Technical Design, Trade-off between modularity and software cost, Cohesion and Coupling, Types of cohesion and coupling, Object oriented Design, Function Oriented Design.

UNIT-IV **Contact Hours: 10**

Software Metrics and Testing

Software Metrics: Definition, Categories of metrics, Token Count, Data Structure Metrics.

Software Testing: Introduction, difference between Bug, Mistake, error, Fault and Failure, Alpha, Beta and Acceptance Testing, Levels of Testing..

Text Books:

1. K. K. Aggarwal and Yogesh Singh, *Software Engineering*, New Age International Private Limited; Fourth edition.
2. R. S. Pressman, *Software Engineering – A practitioner's approach*, , McGraw Hill

Reference Books:

1. Richard Fairley, *Software Engineering Concepts*, McGraw Hill Education.
2. Pankaj Jalote, *An Integrated Approach to Software Engineering*, Narosa Publications

Course Code	Course Title	L	T	P	Credits
BCA-211A	**Linux and Shell Programming** (Pre-requisite: None)	3	0	0	3
		CIE	**SEE**		**Total**
		40	60		100

Course Outcomes (COs): At the end of this course, students will be able to	
BCA-211A.1	Use various Linux commands that are used to manipulate system operations
BCA-211A.2	Understand File management and process manipulation in Linux.
BCA-211A.3	Write Shell Programming using Linux commands.
BCA-211A.4	Understand the file operations and user management at administrator level.

Course Outcomes (CO) to Program Outcomes (PO) mapping (scale 1: Low, 2: Medium, 3: High)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
BCA-211A.1	3	-	-	2	-	-	-	-
BCA-211A.2	3	2	2	2	-	2	-	2
BCA-211A.3	3	-	2	2	-	-	-	2
BCA-211A.4	3	2	3	2	-	3	2	2

Instructions for Paper Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

UNIT-I Contact Hours: 10

Linux operating system and basic commands

Introduction to Linux and Linux Utilities: A brief history of LINUX, architecture of LINUX, features of LINUX, Introduction to vi editor.

Linux Commands: - General Purpose commands (PATH, man, echo, printf, script, passwd, uname, who, date, stty etc.), File Oriented Commands (cat, tail, head, join, tee, pg, comm, cmp, diff, tr, more, wc, lp, od, ls, cp, mv, rm etc.), Directory Oriented Commands (pwd, cd, mkdir, rmdir, ls)

UNIT-II Contact Hours: 12

File permissions, process, and disk related commands

File access permissions: chmod, umask, chgrp, groups i.e checking permissions, changing ownership, groups and permissions of files.

Manipulating Processes and Signals: Basics, process states and transitions, zombie and orphan processes, process-oriented commands. Handling foreground and background jobs. Process scheduling using cron, crontab, at, batch. Changing priority. Signal generation and Handling.

Disk related commands: dd, du, df, dfspace, fdisk, compressing and uncompressing files.

UNIT-III **Contact Hours: 13**

Shell Scripting, filters and pipes

Shell Scripting: Introduction of Bash shell, Bash Features, Command Line, Command Line Expansion, and Editing, Scripting Basics, Creating Shell Scripts, Handling Input/ Output, Control Structures, Conditional Execution, File and string Tests, continue and break, Using positional parameters, Scripting at the command line, Shell Script debugging.

Filters: Filters and Pipes, Concatenating files, Display Beginning and End of files, Cut and Paste, Sorting, Translating Characters, Files with Duplicate Lines, Count Characters, Words or Lines, Comparing Files.

UNIT-IV **Contact Hours: 10**

File system and user account management

File System: Introduction to files, Organization, Assessing File systems, Structure - boot block, super block, inode block, data block. Creating, mounting and unmounting file systems. Shutting down of system, init command.
Managing User accounts: creating, modifying & deleting user accounts and groups

Text Books:

1. Harwani B.M., *Unix and Shell Programming*, Oxford University Press.
2. Goerzen John, *Linux Programming Bible*, IDG Books, New Delhi.
3. Das Sumitabha, *You UNIX – The Ultimate Guide*, Tata McGraw Hill

Reference Books:

1. Matthew Neil, Stones Richard, *Beginning Linux Programming*, Wiley India Pvt. Ltd.
2. Christopher Negus, *Linux Bible*, Wiley India Pvt. Ltd.
3. Richard Peterson, *Linux – The Complete Reference*, Tata McGraw Hill

Course Code	Course Title	L	T	P	Credits
BCA-281A	**Linux and Shell Programming Lab** (Pre-requisite: Linux)	0	0	4	2
		CIE		**SEE**	**Total**
		50		50	100

Course Outcomes (COs): At the end of this course, students will be able to

BCA-281A.1	Understand various Linux System Configuration.
BCA-281A.2	Understand various File Oriented and Directory Oriented Commands.
BCA-281A.3	Understand various control structures of shell scripting.
BCA-281A.4	Work with file manipulation and its permissions.

Course Outcomes (CO) to Program Outcomes (PO) mapping (scale 1: Low, 2: Medium, 3: High)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
BCA-281A.1	3	-	-	2	-	-	-	-
BCA-281A.2	3	2	2	2	-	-	-	2
BCA-281A.3	3	-	2	2	-	-	-	2
BCA-281A.4	3	2	2	2	2	2	-	2

List of Experiments

No.	Experiment Detail
1.	Write shell script to show various system configurations like (i) Currently logged user and his long name (ii) Current shell (iii) Your home directory (iv) Your operating system type (v) Your current path setting (vi) Your current working directory (vii) Show all available shells
2.	Write a shell script that accept a file name starting and ending line numbers as arguments and display all the lines between given line no.
3.	Write a shell script to count the number of words in a file without using in-built commands.
4.	Write a shell script to copy a file to another directory with using relative paths.
5.	Write a shell script to find all files with write permissions at user level, add execute permission also to such files.
6.	Write a shell script to make a file execute at particular time using at command.
7.	Write a script to perform following basic math operations as (i) Addition (ii) Subtraction (iii) Multiplication (iv) Division Note: i) Take input from keyboard. ii) Take input from command line arguments

8.	Write a shell script to find factorial of a number.
9.	Write a shell script to find Fibonacci series.
10.	Write a shell script to read a file name from user, check whether the file exist in directory or not. If yes, then append the content into file. If not then, create file and append the content into the file.
11.	Write a shell script to convert upper case characters in a file to lower case and vice versa.
12.	Write a shell script to compare two files, sort and find duplicate records in the file.

Text Books:

1. Harwani B.M., *Unix and Shell Programming*, Oxford University Press.
2. Goerzen John, *Linux Programming Bible*, IDG Books, New Delhi.
3. Das Sumitabha, *You UNIX – The Ultimate Guide*, Tata McGraw Hill

Reference Books:

1. Matthew Neil, Stones Richard, *Beginning Linux Programming*, Wiley India Pvt. Ltd.
2. Christopher Negus, *Linux Bible*, Wiley India Pvt. Ltd.
3. Richard Peterson, *Linux – The Complete Reference*, Tata McGraw Hill

<table>
<tr><th rowspan="1">Course Code</th><th>Course Title</th><th>L</th><th>T</th><th>P</th><th>Credits</th></tr>
<tr><td rowspan="3"><b>BCA-203A</b></td><td rowspan="3"><b>Database Management System</b>
(Pre-requisite: None)</td><td>3</td><td>0</td><td>0</td><td>3</td></tr>
<tr><td colspan="1"><b>CIE</b></td><td colspan="2"><b>SEE</b></td><td><b>Total</b></td></tr>
<tr><td>40</td><td colspan="2">60</td><td>100</td></tr>
</table>

Course Outcomes (COs): At the end of this course, students will be able to	
BCA-203A.1	Understand the fundamentals of DBMS, data models, schema architecture, and ER/EER design.
BCA-203A.2	Apply SQL and PL/SQL concepts to create, query, and manage relational databases.
BCA-203A.3	Normalize database schemas using functional dependencies and various normal forms.
BCA-203A.4	Analyze transaction processing, concurrency control, and data warehousing concepts.

Course Outcomes (CO) to Program Outcomes (PO) mapping (scale 1: Low, 2: Medium, 3: High)

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
BCA-203A.1	3	-	-	-	-	-	-	-
BCA-203A.2	3	2	2	2	-	-	-	-
BCA-203A.3	3	2	2	2	-	-	-	2
BCA-203A.4	3	3	-	2	-	2	2	3

Instructions for Paper Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

UNIT-I **Contact Hours: 11**

Database and Database Management System

Introduction and need of Database and Database Management System (DBMS). Characteristics of DBMS, Database Users, Data Models, Schema, and Instances. Three-Schema architecture and data Independence.

Database Design using ER model: Entity types, Entity set, attributes and Keys. Relationship types. Expressing M:N relation. Enhanced Entity Relationship Model (EER): Specialization, Generalization, Attribute inheritance and Aggregation

UNIT-II **Contact Hours: 12**

Relational model and Relational Algebra

Relational model concepts. Introduction to SQL, Types of SQL commands (DDL, DML, and DCL etc),SQL operators and their procedure, views and indexes. Queries and sub queries, Group by and Having clause, Aggregate functions and Constraints. PL/SQL: Architecture of PL/SQL, Basic Elements of PL/SQL.

Relational Algebra: Introduction of Relational Algebra, Selection and projection. Renaming, Joins, Unions, Intersection, Cartesian product and Division.

UNIT-III **Contact Hours: 10**

Normalization

Need for Normalization, anomalies (insert, delete and update), Functional Dependencies, Minimal set of

Functional Dependencies, Normal Forms: 1NF, 2NF, 3NF. Higher Level Normal Forms: Boyce/ Codd Normal Form, multi-valued dependency (MVD), Fourth Normal Form, Join dependencies and Fifth Normal Form.

UNIT-IV **Contact Hours: 12**

Transaction processing, Overview of data mining and warehousing

Transaction Processing Concepts: Transaction and Schedules, transaction properties, concurrent execution of transaction, Conflict and View serializability, testing for serializability, concepts in Recoverable and Cascadeless schedules.Concurrency Control Techniques: Lock based protocols, Two phase locking technique, time stamp based protocols, validation based protocols.

Overview of data mining technology, Introduction of data warehousing, Characteristics of data warehouses.

Text Books:

1. Ramez Elmasri and Shamkant B. Navathe, *Fundamentals of Database Systems*, Pearson
2. Abraham Silberschatz, Henry F. Korth, and S. Sudarshan, *Database System Concepts,* McGraw Hill
3. Ivan Bayross, *SQL, PL/SQL the Programming Language of Oracle*, BPB

Reference Books:

1. Raghu Ramakrishnan and Johannes Gehrke, *Database Management Systems,* McGraw Hill
2. Markus Winand, *SQL Performance Explained*, Self Publishing

<table>
<tr><th rowspan="1">Course Code</th><th>Course Title</th><th>L</th><th colspan="2">T</th><th>P</th><th>Credits</th></tr>
<tr><td rowspan="3"><b>BCA-273A</b></td><td rowspan="3"><b>Database Management System Lab</b>
(Pre-requisite: DBMS, PL/SQL)</td><td>0</td><td colspan="2">0</td><td>4</td><td>2</td></tr>
<tr><td colspan="2"><b>CIE</b></td><td colspan="2"><b>SEE</b></td><td><b>Total</b></td></tr>
<tr><td colspan="2">50</td><td colspan="2">50</td><td>100</td></tr>
</table>

Course Outcomes (COs): At the end of this course, students will be able to

BCA-273A.1	Understand the fundamentals ER design.
BCA-273A.2	Apply SQL and PL/SQL concepts to create, query, and manage relational databases.
BCA-273A.3	Perform pattern matching and other operations
BCA-273A.4	Write PL/SQL programs using selection, loop, exception and Normalization.

Course Outcomes (CO) to Program Outcomes (PO) mapping (scale 1: Low, 2: Medium, 3: High)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
BCA-273A.1	3	-	-	-	-	-	-	-
BCA-273A.2	3	2	3	2	-	2	2	-
BCA-273A.3	3	-	-	-	-	-	-	-
BCA-273A.4	3	2	3	2	-	-	2	2

List of Experiments

No.	Experiment Detail
1.	Design an ER diagram for a college database with students, departments, and courses.
2.	Convert the ER diagram into relational schema and identify primary, foreign keys.
3.	Create tables in a relational DBMS using SQL based on the ER schema.
4.	Execute basic SQL commands: DDL, DML, DCL on a sample database
5.	Demonstrate the use of Data Constraints.
6.	Write SQL queries using WHERE, BETWEEN, LIKE, IN, and ORDER BY.
7.	Use aggregate functions with GROUP BY and HAVING clauses.
8.	Demonstrate pattern matching and range searching functions.
9.	Demonstrate the use of ORACLE operators • Arithmetic operators • Comparison operators • Logical operators
10.	Demonstrate the use of DUAL table
11.	Demonstrate the use of INDEX in DBMS
12.	Perform different types of joins (INNER, OUTER).
13.	Write PL/SQL programs using variables, IF-THEN, loops, and exceptions.
14.	Identify anomalies and perform step-by-step normalization (1NF to 3NF) on a sample unnormalized relation.

15.	Demonstrate the concept of locking using a small transaction simulation.

Text Books:

1. Ramez Elmasri and Shamkant B. Navathe, *Fundamentals of Database Systems*, Pearson
2. Abraham Silberschatz, Henry F. Korth, and S. Sudarshan, *Database System Concepts,* McGraw Hill
3. Ivan Bayross, *SQL, PL/SQL the Programming Language of Oracle*, BPB

Reference Books:

1. Raghu Ramakrishnan and Johannes Gehrke, *Database Management Systems,* McGraw Hill
2. Markus Winand, *SQL Performance Explained*, Self Publishing

<table>
<tr><th rowspan="1">Course Code</th><th>Course Title</th><th>L</th><th>T</th><th>P</th><th>Credits</th></tr>
<tr><td rowspan="3"><b>BCA-213A</b></td><td rowspan="3"><b>Introduction to Information Security</b>
(Pre-requisite: None)</td><td>2</td><td>0</td><td>0</td><td>2</td></tr>
<tr><td colspan="2"><b>CIE</b></td><td><b>SEE</b></td><td><b>Total</b></td></tr>
<tr><td colspan="2">40</td><td>60</td><td>100</td></tr>
</table>

Course Outcomes (COs): At the end of this course, students will be able to

BCA-213A.1	Understand the fundamental principles of information security.
BCA-213A.2	Learn cryptographic techniques and authentication mechanisms to ensure data confidentiality, integrity, and access control in secure systems.
BCA-213A.3	Analyze network security threats and implement protective technologies such as firewalls, IDS/IPS
BCA-213A.4	understand the concepts of operating systems and web securities

Course Outcomes (CO) to Program Outcomes (PO) mapping (scale 1: Low, 2: Medium, 3: High)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
BCA-213A.1	3	-	-	-	-	-	-	-
BCA-213A.2	3	2	2	2	-	-	2	2
BCA-213A.3	3	3	3	2	-	-	2	2
BCA-213A.4	3	2	-	2	-	-	2	2

Instructions for Paper Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

UNIT-I Contact Hours: 8

Information Security Concepts

Information Security Concepts: Information security issues, goals of information security (CIA triad: Confidentiality, Integrity, Availability), architecture, Attacks, Security Services and Mechanisms.
Overview of Security threats and Vulnerability: Types of attacks on Confidentiality, Integrity and Availability. Vulnerability and Threats..

UNIT-II Contact Hours: 7

Cryptography, Authentication and Access Control

Introduction to Cryptography: Network security model, Cryptographic systems, Cryptanalysis, Steganography. Types of Cryptography: Symmetric key and Asymmetric Key Cryptography, Encryption and Decryption Techniques. Authentication and Access Control: Authentication methods (passwords, biometrics, multi-factor), and access control models (DAC, MAC, RBAC).

UNIT-III Contact Hours: 8

Network and Software Security

Network Security Model, Network Security Threats, Firewalls: Overview, Types, Features, User Management, Intrusion Detection System , Intrusion Prevention System,
Secure Software Development: Secure coding practices, common vulnerabilities (SQL injection, XSS), and secure software development lifecycle (SDLC).

UNIT-IV **Contact Hours: 7**

Operating System and Web Security

Operating System Security: Designing Secure Operating Systems, OS Security Vulnerabilities, patch management, and system hardening.
Email security: PGP and SMIME,
Web Security: Web security threats, Web authentication, Injection Flaws, SQL Injection, secure web application development, and common web security tools, Web Browser Security.

Text Books:

1. Michael E. Whitman, Herbert J. Mattrod , *Principles of Information Securtiy*, Mindtap.
2. Sanil Nandkarni , *Fundamentals of Information Securtiy*, BPB Publications.

Reference Books:

1. Glen Sagers, Bryan Hosack, *Information Technology Security Fundamentals*, Business Expert Press
2. David Kim, Michael G. Solomon, *Fundamentals Of Information Systems Security,* Jones and Bartlett Publishers, Inc.

Course Code	Course Title	L	T	P	Credits
BCA-215A	**Full Stack Development-1 (MERN)** (Pre-requisite: Web Technology (HTML, CSS, JavaScript))	3	0	0	3
		CIE		**SEE**	**Total**
		40		60	100

Course Outcomes (COs): At the end of this course, students will be able to

BCA-215A.1	Understand the core concepts of the MERN stack, including React.js, Express.js, Node.js, and MongoDB.
BCA-215A.2	Develop front-end applications using React.js, including state management and component-based architecture.
BCA-215A.3	Build RESTful APIs with Express.js and integrate MongoDB for backend data storage and retrieval.
BCA-215A.4	Implement authentication, middleware, and deployment techniques for full-stack applications.

Course Outcomes (CO) to Program Outcomes (PO) mapping (scale 1: Low, 2: Medium, 3: High)

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
BCA-215A.1	3	-	-	-	-	-	-	2
BCA-215A.2	3	-	3	3	-	2	-	2
BCA-215A.3	3	2	3	3	-	-	-	2
BCA-215A.4	3	2	3	3	-	2	2	2

Instructions for Paper Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

UNIT-I **Contact Hours: 12**

Introduction to MERN Stack & Node.js

Overview of Full Stack Development and MERN Stack.Node.js: Introduction, history, features, and its role in MERN. Comparison between JavaScript Client-side vs. Server-side programming. Installation of Node.js and npm, writing and executing basic scripts. Understanding JavaScript fundamentals: Variables, functions, loops, and arrays. Event-driven architecture and the Node.js runtime environment.

UNIT-II **Contact Hours: 11**

Frontend Development with React.js basics

React.js Introduction: Why React? Virtual DOM, React Setup (Create React App). JSX & Components: Functional vs. Class Components, Props, State. React Hooks: useState, useEffect, useContext, handling side effects. Event Handling & Forms: Controlled vs. Uncontrolled components. React Router: Navigation, dynamic routing, route parameters. State Management in React: Context API, Prop Drilling, Lifting State Up.

UNIT-III **Contact Hours: 12**

Backend Development with Node.js & Express.js

Node.js Modules: Core modules, global modules, and user-defined modules.Express.js: Introduction, routing, request handling, middleware. Template Engines (EJS) for rendering dynamic web pages. Middleware: Introduction to Express.js middleware and error handling. Database Integration: Introduction to MongoDB, performing CRUD operations with Mongoose. Using Postman for API testing.

UNIT-IV **Contact Hours: 10**

Advanced Concepts and Deployment

Asynchronous Programming: Callback, Promises, Async/Await, Event loop in Node.js. Mongoose ORM: Schema and model creation, advanced queries. Authentication & Authorization: JSON Web Tokens (JWT), bcrypt, Role-based authentication. State Management in React: Context API, Redux basics. File Handling & APIs: Using `multer` for file uploads, building Search APIs. Deployment: Hosting MERN applications using services like Vercel, Netlify, and AWS.

Text Books:

1. Basarat Ali Syed, *Beginning Node.js*, Apress.
2. Adam Boduch, Roy Derks, *React and React Native*, Packt Publishing.
3. Ethan Brown, *Learning JavaScript Design Patterns*, O'Reilly.
4. Vasan Subramanian, *Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node,* Apress, 2019.

Reference Books:

1. Boronczyk, Naramore, *Beginning PHP, Apache, MySQL Web Development*, Wiley India Pvt.Ltd.
2. Kyle Simpson, *You Don't Know JS: Up and Going*, O'Reilly.
3. David Flanagan, *JavaScript: The Definitive Guide*, O'Reilly Media.
4. Simon Holmes, Clive Harber, *Getting MEAN with Mongo, Express, Angular, and Node*, Manning Publications.

Course Code	Course Title	L	T	P	Credits
BCA-283A	**Full Stack Development-1 (MERN) Lab** (Pre-requisite: MERN Stack)	0	0	4	2
		CIE	**SEE**		**Total**
		50	50		100

Course Outcomes (COs): At the end of this course, students will be able to

BCA-283A.1	Implement Node.js applications
BCA-283A.2	Write programs implementing Express.js
BCA-283A.3	Write programs implementing React.js
BCA-283A.4	Deploy a Full stack development application

Course Outcomes (CO) to Program Outcomes (PO) mapping (scale 1: Low, 2: Medium, 3: High)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
BCA-283A.1	3	2	2	2	-	-	-	2
BCA-283A.2	3	2	2	2	-	-	-	2
BCA-283A.3	3	3	2	2	-	-	-	2
BCA-283A.4	3	3	3	2	-	2	2	2

List of Experiments

No.	Experiment Detail
1.	Write a simple 'Hello World' program in Node.js.
2.	Create a Node.js script that reads and writes files using the `fs` module.
3.	Build a basic Express.js server with different routes.
4.	Implement middleware in Express.js for logging request details.
5.	Create a RESTful API in Express.js to perform CRUD operations on a MongoDB database.
6.	Implement authentication using JWT in an Express.js application.
7.	Upload and retrieve files/images using `multer` in Express.js.
8.	Set up a basic React application and create a functional component.
9.	Implement React state and props in a simple To-Do List app.
10.	Build a multi-page React app using React Router.
11.	Fetch data from an API and display it using React (Axios or Fetch API).
12.	Implement global state management using the Context API in React.
13.	Implement form validation and handle user input in React.
14.	Connect a React frontend with a Node.js/Express backend using Axios.
15.	Deploy a full-stack MERN application (frontend + backend) using Vercel and Netlify.

Text Books:

1. Basarat Ali Syed, *Beginning Node.js*, Apress.
2. Adam Boduch, Roy Derks, *React and React Native*, Packt Publishing.
3. Ethan Brown, *Learning JavaScript Design Patterns,* O'Reilly.
4. Vasan Subramanian, *Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node,* Apress, 2019.

Reference Books:

1. Boronczyk, Naramore*, Beginning PHP, Apache, MySQL Web Development*, Wiley India Pvt.Ltd.
2. Kyle Simpson*, You Don't Know JS: Up and Going*, O'Reilly.
3. David Flanagan, *JavaScript: The Definitive Guide*, O'Reilly Media.
4. Simon Holmes, Clive Harber*, Getting MEAN with Mongo, Express, Angular, and Node*, Manning Publications.