PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

Bangalore Trunk Road, Varadharajapuram,
Poonamallee, Chennai – 600123

Minor Degree

FULL STACK WEB DEVELPMENT

Curriculum & Syllabus

DEPARTMENT OF
INFORMATION TECHNOLOGY

REGULATION 2023

PANIMALAR ENGINEERING COLLEGE

DEPARTMENT OF INFORMATION TECHNOLOGY

Minor Degree - Full Stack Web Development

S. No	COURSE CODE	COURSE TITLE	Category	L/T/P	Contact Hours	Credit	Ext / Int Weightage
1.	**23IT4001**	**Web Technologies**	PE	3/0/0	3	3	60/40
2.	**23IT4002**	**Front End Frameworks**	PE	3/0/0	3	3	60/40
3.	**23IT4003**	**Back End Development**	PE	3/0/0	3	3	60/40
4.	**23IT4004**	**Database and Deployment**	PE	3/0/0	3	3	60/40
5.	**23IT4005**	**Advanced JavaScript**	PE	3/0/0	3	3	60/40
6.	**23IT4006**	**Software Engineering for Web Applications**	PE	3/0/0	3	3	60/40
7.	**23IT4007**	**DevOps Deployment**	PE	3/0/0	3	3	60/40
8.	**23IT4008**	**UI / UX Design Principles and Tools**	PE	3/0/0	3	3	60/40

<table>
<tr><td rowspan="2">23IT4001</td><td rowspan="2">WEB TECHNOLOGIES</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td>3</td><td>0</td><td>0</td><td>3</td></tr>
</table>

COURSE OBJECTIVE:

- Introduce the core concepts of the World Wide Web and web technologies.
- Design structured and styled web pages using HTML5 and CSS based on semantic principles and best practices.
- Familiarize students with interactive behavior using JavaScript to enhance user experience.
- Teach basic principles of responsive and accessible web design.
- Introduce the process of web hosting and domain fundamentals using modern platforms.
- Encourage practical problem-solving and hands-on webpage creation.

UNIT I **Introduction to Web & HTML5** **9**

Internet vs. Web-Working of the Web – Client, Server, DNS, HTTP/HTTPS-Introduction to HTML5-HTML Elements: Tags, Attributes, Lists, Tables, Forms, Media Embeds-Semantic HTML and Best Practices.

UNIT II **Styling with CSS3** **9**

Introduction to CSS: Inline, Internal, External-CSS Selectors and Properties-Colors, Fonts, Backgrounds, Borders, Box Model-Flexbox and Grid Layout-CSS Media Queries for Responsiveness.

UNIT III **JavaScript for Web Interactivity** **9**

Basics of JavaScript: Syntax, Variables, Operators, Data Types-Conditional Statements and Loops- Functions and Events-DOM Manipulation (getElementById, innerHTML)-Form Validation Basics

UNIT IV **Responsive and Accessible Web Design** **9**

Introduction to Responsive Design-Mobile-First Approach-Viewport and Breakpoints-Accessibility Guidelines (WCAG), Alt Text, ARIA Roles-Introduction to Bootstrap Framework

UNIT V **Web Publishing & Tools** **9**

Introduction to Web Hosting and Domain Registration-Git & GitHub Basics-Hosting on GitHub Pages or Netlify-Introduction to Browser Developer Tools-Web Design Tools: Canva, Figma (Basic overview).

TOTAL :45 PERIODS

COURSE OUTCOME

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

CO1	Describe the functioning of the web, client-server models, and browsers.
CO2	Develop structured web pages using HTML5.
CO3	Apply CSS to design visually appealing and responsive layouts.
CO4	Demonstrate interactivity using basic JavaScript functions and events.
CO5	Create accessible web content compatible with different devices.
CO6	Deploy a static website using hosting platforms like GitHub Pages or Netlify.

TEXT BOOKS:

1. Jon Duckett, “HTML and CSS: Design and Build Websites”, Wiley, 2021.
2. Jennifer Robbins, “Learning Web Design: A Beginner's Guide” , 5th Edition, O'Reilly Media, 2022.
3. Terry Felke-Morris, “Web Development with HTML5, CSS, JavaScript” , Pearson, 2021.
4. Ben Frain, “Responsive Web Design with HTML5 and CSS” , 4th Edition, Packt Publishing, 2023
5. David Flanagan, “JavaScript: The Definitive Guide” , 7th Edition, O’Reilly Media, 2020

REFERENCE BOOKS:

1. Zak Ruvalcaba and Anne Boehm, Mike Murach & Associates, “Murach’s HTML5 and CSS3” , 2021
2. Cay S. Horstmann, “Modern JavaScript for the Impatient”, Addison-Wesley, 20213. Jonathan Fielding, “Beginning Responsive Web Design with HTML5 and CSS3”, Apress, 2020.

WEB REFERENCES:

1. https://developer.mozilla.org/ (MDN Web Docs)
2. https://www.w3schools.com/
3. https://css-tricks.com/
4. https://www.freecodecamp.org/
5. https://web.dev/ (by Google)

CO-PO-PSO MAPPING

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2			2					2	2
CO2	3	2	3		3					2	2
CO3	3	2	3		3					2	2
CO4	3	2	3		3					2	2
CO5	3	2	2		2	2	2			2	2
CO6	3	2	3	2	3					2	3

Internal Assessment				End Semester Examinations
Assessment I (100 Marks)		Assessment II (100 Marks)		
Individual Assignment / Case Study / Seminar / Mini Project	Written Test	Individual Assignment / Case Study / Seminar / Mini Project	Written Test	Written Examinations
40	60	40	60	100
40%				60 %

23IT4002	FRONT END FRAMEWORKS	L	T	P	C
		3	0	0	3

COURSE OBJECTIVE:

- Introduce component-based front-end architecture using JavaScript frameworks.
- Build dynamic and interactive user interfaces with state-driven logic.
- Apply state management and event handling for smooth UI updates.
- Demonstrate routing, form handling, and lifecycle methods in web apps.
- Follow best practices for responsive and modular UI design.
- Develop single-page applications (SPAs) using front-end frameworks.

UNIT I Introduction to JavaScript Frameworks 9

Need for front-end frameworks-Overview of popular frameworks: React, Angular, Vue (comparison) - Introduction to React.js: Setup using Vite/Create React App-JSX and rendering elements-Functional vs class components.

UNIT II Components and Props 9

Component creation and composition-Props and data flow-List rendering and keys-Event handling and conditional rendering-CSS styling in React (inline, modules, styled-components).

UNIT III State and Lifecycle 9

useState, useEffect hooks-Component lifecycle in functional components-Lifting state up-Controlled vs uncontrolled components-React Developer Tools.

UNIT IV Routing and Forms 9

React Router DOM: Navigation, Route, Link, useParams-Dynamic routing-Building and handling forms-Form validation using React Hook Form or Formik-Error handling and user feedback

UNIT V Advanced Concepts and Deployment 9

useContext and global state (intro to Redux or Context API)-Fetching data from REST APIs using fetch/axios-Handling promises and async/await- Environment variables and build optimization- Hosting SPAs on Netlify/Vercel/GitHub Pages

TOTAL :45 PERIODS

COURSE OUTCOME

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

CO1 Describe the principles of front-end frameworks and their advantages.

CO2 Apply concepts to build and reuse UI components using a JavaScript framework like React

CO3 Analyze routing strategies and implement state management in front-end applications.

CO4 Evaluate user input and form validation dynamically.

CO5 Integrate REST APIs and handle asynchronous operations using modern techniques.

CO6 Design and deploy a complete single-page applications (SPA) using industry-standard tools.

TEXT BOOKS:

1. Alex Banks & Eve Porcello, “Learning React: Modern Patterns for Developing React Apps” , 3rd Ed., O’Reilly, 2023
2. Stoyan Stefanov, “React Up and Running: Building Web Applications” , 2nd Ed., O'Reilly, 2022
3. Adam Freeman, “Pro React 16” , Apress, 2021
4. Roy Derks, “React Projects” Packt Publishing, 2021
5. Accomazzo, Murray, Lerner, “Fullstack React: The Complete Guide to ReactJS and Friends” , Fullstack.io, 2022

REFERENCE BOOKS:

1. Dave Ceddia, “Pure React” , 2021
2. Michele Bertoli, “React Design Patterns and Best Practices” Packt, 2021
3. Robin Wieruch, “The Road to React” , 2023

WEB REFERENCES:

1. https://reactjs.org/ – Official React documentation
2. https://javascript.info/ – JavaScript essentials
3. https://www.freecodecamp.org/news/tag/react/ – Free tutorials and guides
4. https://www.w3schools.com/react/ – Beginner-friendly tutorials
5. https://beta.reactjs.org/ – New React docs (2023+)

CO-PO-PSO MAPPING

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	2	1	3	2					2	3	2	3
CO2	3	3	3	2	3	2					2	3	3	3
CO3	3	3	3	3	3					2	3	3	3	3
CO4	3	3	3	3	3	3				3	3	3	3	3
CO5	3	3	3	2	3	3				3	3	3	3	3
CO6	3	3	3	3	3	3				3	3	3	3	3

<table>
<tr><th colspan="4">Internal Assessment</th><th rowspan="2">End Semester Examinations</th></tr>
<tr><th colspan="2">Assessment I (100 Marks)</th><th colspan="2">Assessment II (100 Marks)</th></tr>
<tr><th>Individual Assignment / Case Study / Seminar / Mini Project</th><th>Written Test</th><th>Individual Assignment / Case Study / Seminar / Mini Project</th><th>Written Test</th><th>Written Examinations</th></tr>
<tr><td>40</td><td>60</td><td>40</td><td>60</td><td>100</td></tr>
<tr><td colspan="4"><b>40%</b></td><td><b>60 %</b></td></tr>
</table>

23IT4003	BACKEND DEVELOPMENT	L	T	P	C
		3	0	0	3

COURSE OBJECTIVE:

- Introduce the fundamentals of server-side programming using Node.js.
- Teach how to build and manage web servers using Express.js
- Enable creation and handling of RESTful APIs for structured client-server interaction.
- Provide knowledge of middleware, routing, and templating engines to support dynamic back-end logic.
- Apply essential security techniques, including authentication and session handling, in back-end systems.
- Demonstrate database connectivity and full-stack integration

UNIT I **Introduction to Node.js** **9**

Node.js and its use- Setting up Node.js environment-npm packages and modules- Writing your first Node.js server-File system, events, and asynchronous programming.

UNIT II **Working with Express.js** **9**

Express.js - Creating routes and handling requests-Express middleware functions-Serving static files-Using templating engines (EJS or Pug)

UNIT III **RESTful API Development** **9**

Understanding REST architecture-Creating GET, POST, PUT, DELETE endpoints-JSON and request/response structure-Handling query params and route parameters-Using tools like Postman for testing

UNIT IV **Database Connectivity** **9**

Introduction to MongoDB-Connecting Node.js with MongoDB using Mongoose-Performing CRUD operations-Data modeling and schemas-Error handling and validations

UNIT V **Authentication, Security & Deployment** **9**

Introduction to JWT (JSON Web Token) authentication-Securing routes and user sessions-Using dotenv and environment variables-Hosting apps on platforms like Render, Railway, or Heroku- Debugging and logging with tools like Morgan.

TOTAL :45 PERIODS

COURSE OUTCOME

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

CO1 Explain how to build and run web servers using Node.js.
CO2 Apply Express.js to develop robust back-end applications
CO3 Construct RESTful APIs to interact with front-end systems.
CO4 Implement routing, middleware, and error handling in Express.
CO5 Integrate MongoDB databases using Mongoose and perform CRUD operations.
CO6 Deploy and secure a complete back-end application on cloud platforms using environment-based configurations.

TEXT BOOKS:

1. Andrew Mead, “Learning Node.js Development” , Packt, 2nd Edition, 2022
2. Evan Hahn, “Express in Action” , Manning Publications, 2021
3. Colin J. Ihrig, “Pro Node.js for Developers” , Apress, 2nd Edition, 2022
4. Manuel Kiessling, “Node.js: The Complete Guide” , Leanpub, 2023
5. David Herron, “Node.js Web Development” , Packt, 6th Edition, 2023

REFERENCE BOOKS:

1. Sandro Pasquali, Mastering Node.js, Packt, 2021
2. Fernando Doglio, REST API Development with Node.js, Apress, 2022
3. Adam Bretz & Colin J. Ihrig, Full-Stack Web Development with MongoDB and Express, 2021

WEB REFERENCES:

1. https://nodejs.org/en/docs/ – Official Node.js Documentation
2. https://expressjs.com/ – Express.js Guide
3. https://mongoosejs.com/docs/ – MongoDB and Mongoose Docs
4. https://www.freecodecamp.org/news/tag/node/ – Free learning articles
5. https://developer.mozilla.org/ – MDN Web Docs for JS/HTTP

CO-PO-PSO MAPPING

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	**3**	**2**			**3**						**2**
CO2	**3**	**2**	**3**	**2**	**3**					**2**	**2**
CO3	**3**	**3**	**3**	**2**	**3**					**2**	**2**
CO4	**3**	**2**	**3**	**2**	**3**					**2**	**2**
CO5	**3**	**3**	**3**	**2**	**3**					**2**	**2**
CO6	**3**	**2**	**3**	**2**	**3**				**2**	**2**	**3**

Internal Assessment				**End Semester Examinations**
Assessment I (100 Marks)		**Assessment II (100 Marks)**		
Individual Assignment / Case Study / Seminar / Mini Project	**Written Test**	**Individual Assignment / Case Study / Seminar / Mini Project**	**Written Test**	**Written Examinations**
40	60	40	60	100
40%				**60 %**

23IT4004	DATABASE AND DEPLOYMENT	L	T	P	C
		3	0	0	3

COURSE OBJECTIVE:

- Explain the core concepts of relational and non-relational database systems.
- Teach ER modeling and normalization for designing efficient databases.
- Develop SQL queries for data definition, manipulation, retrieval, and aggregation.
- Enable integration of databases with backend technologies for full-stack applications.
- Introduce NoSQL databases and apply document-based design using MongoDB.
- Provide hands-on exposure to database connectivity and APIs.

UNIT I **Database Basics and ER Modeling** **9**

Introduction to databases and types (Relational vs NoSQL)-DBMS vs RDBMS-ER Model: Entities, Attributes, Relationships-Keys: Primary, Foreign, Composite-Mapping ER diagrams to relational schema

UNIT II **Relational Database Design** **9**

Relational model basics-Schema design principles-Functional dependencies-Normalization (1NF to 3NF, BCNF)-Integrity constraints and referential integrity

UNIT III **Structured Query Language (SQL)** **9**

Introduction to SQL: DDL, DML, DCL, TCL-Creating and modifying tables-SELECT, INSERT, UPDATE, DELETE-Joins (INNER, OUTER, SELF), GROUP BY, HAVING-Subqueries and views

UNIT IV **Database Integration with Applications** **9**

Introduction to backend integration (Node.js + Express) - Connecting to MySQL/ PostgreSQL using drivers-Performing CRUD operations through web APIs-Query parameterization and avoiding SQL injection-Connecting front-end forms with backend databases

UNIT V **Introduction to NoSQL and MongoDB** **9**

Overview of NoSQL: key-value, document, column, graph-MongoDB basics: Collections, Documents-CRUD with MongoDB using Mongoose-Data modeling in MongoDB-Comparing SQL and NoSQL – use cases

TOTAL :45 PERIODS

COURSE OUTCOME

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

CO1	Describe relational database schemas using ER models
CO2	Apply normalization to remove data redundancy.
CO3	Develop SQL queries for CRUD operations and joins
CO4	Integrate databases with Node.js/Express backend applications
CO5	Implement MongoDB for NoSQL-based applications
CO6	Design and evaluate full-stack components involving both SQL and NoSQL database interactions.

TEXT BOOKS:

1. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, “Database System Concepts” , 7th Ed., McGraw-Hill, 2022
2. Ben Forta, “SQL in 10 Minutes, Sams Teach Yourself” , 6th Ed., Pearson, 2023
3. Alan Beaulieu, “Learning SQL” , 3rd Ed., O'Reilly, 2021
4. Vasan Subramanian, “Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node” , Apress, 2nd Ed., 2021

REFERENCE BOOKS:

1. Ramez Elmasri, Shamkant B. Navathe, Fundamentals of Database Systems ,7th Ed., Pearson, 2022
2. Anthony Molinaro, 2nd Ed., SQL Cookbook ,O’Reilly, 2020
3. Martin Kleppmann, O’Reilly, Designing Data-Intensive Applications- 2022

WEB REFERENCES:

1. https://www.w3schools.com/sql/ – SQL tutorials and examples
2. https://sqlzoo.net/ – Interactive SQL learning
3. https://www.mongodb.com/docs/ – MongoDB Official Docs
4. https://dev.mysql.com/doc/ – MySQL Documentation
5. https://sequelize.org/ – ORM for SQL DBs in Node.js

CO-PO-PSO MAPPING

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2			2						1
CO2	3	3	2	2	2						2
CO3	3	2	2	2	3						2
CO4	3	2	3	2	3					2	2
CO5	3	2	3	2	3					2	2
CO6	3	3	3	2	3				2	2	3

Internal Assessment				**End Semester Examinations**
Assessment I (100 Marks)		**Assessment II (100 Marks)**		
Individual Assignment / Case Study / Seminar / Mini Project	**Written Test**	**Individual Assignment / Case Study / Seminar / Mini Project**	**Written Test**	**Written Examinations**
40	60	40	60	100
40%				**60 %**

23IT4005	ADVANCED JAVASCRIPT	L	T	P	C
		3	0	0	3

COURSE OBJECTIVE:

- Explain the core features of JavaScript, including ES6+ enhancements.
- Develop advanced client-side logic using closures, prototypes, and functional techniques.
- Utilize asynchronous programming constructs like promises, async/await, and AJAX.
- Apply modular JavaScript practices using tools like Babel, Webpack, and NPM.
- Integrate JavaScript with browser APIs and event-driven architectures.
- Build optimized, maintainable, and scalable web applications using JavaScript.

UNIT I JavaScript Internals and Execution Contexts 9

Execution context and call stack-Scope, lexical environment, and closures - Variable hoisting (var, let, const) - The this keyword and its binding - Function declarations vs expressions - Prototype-based inheritance and prototype chain - Memory management and garbage collection - Event loop and asynchronous behavior overview

UNIT II ES6+ Features and Functional Programming 9

Introduction to ES6 and JavaScript evolution - Arrow functions and lexical -Template literals, default parameters - Destructuring (arrays and objects) - Spread and rest operators - Object shorthand and enhancements - Functional programming principles - Pure functions, immutability, higher-order functions - Array methods: map, filter, reduce, and forEach

UNIT III Asynchronous Programming and API 9

Asynchronous JavaScript: Why and how- Callbacks and callback hell - Promises: creation, chaining, error handling - async and await – syntax and flow - AJAX vs Fetch API - Making API calls with fetch() - Handling API responses and JSON - Error handling in asynchronous code - Real-time use case: chaining API calls

UNIT IV Modules, Tooling, and Project Structuring 9

ES6 modules: import, export- CommonJS vs ESModules - Introduction to Babel and transpilation - Introduction to Webpack and bundling - Working with package.json and NPM -Creating and managing NPM scripts - JavaScript project folder structure -Linting and code formatting (ESLint, Prettier) -Using .env and environment-based configuration

UNIT V DOM, Browser APIs, and Application Development 9

DOM traversal and manipulation (getElementById, querySelector)- DOM events and event delegation-Browser APIs: localStorage, sessionStorage, Geolocation - Client-side form validation using JavaScript - Animations with JavaScript and CSS transitions - Error handling in browser context -Building interactive components (e.g., carousel, modal)

Mini project: Build a dynamic single-page interface using vanilla JavaScript

TOTAL :45 PERIODS

COURSE OUTCOME

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

CO1 Explain JavaScript concepts like scope, closures, hoisting, and prototypes.
CO2 Apply ES6+ features like arrow functions, destructuring, and modules in real code.
CO3 Develop asynchronous code using callbacks, promises, and async/await.
CO4 Use tools like Babel, Webpack, and NPM to organize modular JavaScript code.

CO5 Integrate JavaScript with the DOM, browser APIs, and event-driven code.
CO6 Create clean, optimized, and maintainable web apps using advanced JavaScript patterns.

TEXT BOOKS:

1. D. Flanagan, JavaScript: The Definitive Guide, 7th ed. Sebastopol, CA: O'Reilly Media, 2020.
2. N. C. Zakas, Understanding ECMAScript 6: The Definitive Guide for JavaScript Developers, 1st ed. San Francisco, CA: No Starch Press, 2016.

REFERENCE BOOKS:

1. K. Simpson, You Don't Know JS Yet: Scope and Closures, 2nd ed. Sebastopol, CA: O'Reilly Media, 2020.
2. Banks and E. Porcello, Learning React: Functional Web Development with React and Redux, 2nd ed. Sebastopol, CA: O'Reilly Media, 2020.
3. Mozilla Developer Network (MDN), JavaScript Documentation. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/JavaScript
4. Babel, Babel Handbook. [Online]. Available: https://babeljs.io/docs/
5. Webpack Contributors, Webpack Documentation. [Online]. Available: https://webpack.js.org/

WEB REFERENCES:

1. https://expressjs.com/ – Express.js Official Docs
2. https://swagger.io/docs/ – Swagger Documentation
3. https://www.postman.com/ – Postman API Platform
4. https://jwt.io/ – JWT Resources
5. https://rapidapi.com/ – API Marketplace & Testing

CO-PO-PSO MAPPING

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2	2	2	2						1
CO2	3	2	3		3					1	2
CO3	3	3	3	2	3					2	2
CO4	3	2	3	2	3				1	2	3
CO5	3	2	2	2	3	1				1	2
CO6	3	2	3	2	3				1	2	3

Internal Assessment				**End Semester Examinations**
Assessment I (100 Marks)		**Assessment II (100 Marks)**		
Individual Assignment / Case Study / Seminar / Mini Project	**Written Test**	**Individual Assignment / Case Study / Seminar / Mini Project**	**Written Test**	**Written Examinations**
40	60	40	60	100
40%				**60 %**

23IT4006	SOFTWARE ENGINEERING FOR WEB APPLICATIONS	L	T	P	C
		3	0	0	3

COURSE OBJECTIVE:

- To learn the basic concepts and phases of software development for web applications.
- To understand how to collect and write software requirements clearly.
- To design user interfaces and architecture for web-based systems.
- To know how to test web applications and ensure quality.
- To learn how to use tools for version control and deployment.
- To build and manage a complete web application project using best practices.

UNIT I **Introduction to Software Engineering and SDLC** **9**

Overview of Software Engineering - Characteristics of Web Applications - Software Development Life Cycle (SDLC) Models – Waterfall, Incremental, Spiral, V-Model - Agile Development: Scrum and Kanban - Web-specific engineering considerations

UNIT II **Requirements Engineering and Project Planning** **9**

Requirement Elicitation Techniques (Interviews, Surveys, Use Cases) - Functional and Non-functional Requirements - Software Requirement Specification (SRS) for Web Apps - Estimation Techniques – Function Point, Use Case Points - Project Scheduling – Gantt Charts, Work Breakdown Structure (WBS)

UNIT III **Design and Architecture for Web Applications** **9**

Introduction to Software Design Principles (DRY, KISS, SOLID) - Architecture Patterns: MVC, MVVM for Web Development - UI/UX Design Principles for Web - Low-Fidelity and High-Fidelity Prototyping - Wireframing Tools and Design Mockups

UNIT IV **Web Application Testing and Quality Assurance** **9**

Levels of Testing – Unit, Integration, System, Acceptance - Testing Techniques – White Box, Black Box - Test Case Design and Management - Tools for Web Testing (Selenium, Postman, Jest) - Basics of Quality Assurance and Metrics

UNIT V **Deployment, Maintenance, and DevOps** **9**

Deployment Strategies for Web Applications - Version Control Systems (Git, GitHub) - CI/CD Pipelines – Concepts and Tools - Maintenance Types – Corrective, Adaptive, Preventive - Software Documentation and Sunset Phase

TOTAL :45 PERIODS

COURSE OUTCOME

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

CO1	Explain software engineering principles and lifecycle models for web application projects.
CO2	Apply requirements engineering techniques to create structured SRS documents
CO3	Design user interfaces and system architectures for scalable web applications
CO4	Analyze software testing techniques to ensure quality and reliability in web applications
CO5	Evaluate different deployment strategies and version control tools for web software.

CO6 Construct a complete web application with proper integration, deployment, and maintenance

TEXT BOOKS:

1. Roger S. Pressman, Software Engineering: A Practitioner’s Approach, 8th Ed., McGraw-Hill, 2020
2. Ian Sommerville, Software Engineering, 10th Ed., Pearson Education, 2022
3. Alan Dennis, Barbara Wixom, Roberta Roth, Systems Analysis and Design, Wiley, 2021

REFERENCE BOOKS:

1. Richard Murch, Web Engineering: The Discipline of Systematic Development of Web Applications, Springer, 2021
2. Thomas Erl, Modern Web Development and DevOps, Pearson, 2023
3. Addison Wesley, Agile Web Development with Rails, 6th Ed., 2020

WEB REFERENCES:

1. https://agilemanifesto.org
2. https://scrumguides.org
3. https://developer.mozilla.org
4. https://www.softwaretestinghelp.com

CO-PO MAPPING

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2		1							1
CO2	3	2	2	2					2	2	1
CO3	3	2	3	2	2			1	2	2	1
CO4	3	3	3	2	2	1			1	1	1
CO5	2	1	2	2	3				2	2	2
CO6	3	2	3	2	3	1	1	1	3	3	3

<table>
<tr><th colspan="4">Internal Assessment</th><th rowspan="2">End Semester Examinations</th></tr>
<tr><th colspan="2">Assessment I (100 Marks)</th><th colspan="2">Assessment II (100 Marks)</th></tr>
<tr><th>Individual Assignment / Case Study / Seminar / Mini Project</th><th>Written Test</th><th>Individual Assignment / Case Study / Seminar / Mini Project</th><th>Written Test</th><th>Written Examinations</th></tr>
<tr><td>40</td><td>60</td><td>40</td><td>60</td><td>100</td></tr>
<tr><td colspan="4"><b>40%</b></td><td><b>60 %</b></td></tr>
</table>

23IT4007	DEVOPS DEPLOYMENT	L	T	P	C
		3	0	0	3

COURSE OBJECTIVE:

- Introduce the principles of DevOps and continuous integration/deployment (CI/CD).
- **I**mplement continuous integration and continuous deployment (CI/CD) pipelines using industry-standard tools.
- Teach the use of containers (Docker) and orchestration (Kubernetes)
- Implement cloud-based deployment practices (AWS, Azure, or GCP)
- Develop skills in monitoring, logging, and incident response
- Apply version control, scripting, and configuration management tools for maintaining DevOps workflows.

UNIT I Introduction to DevOps and Version Control 9

DevOps philosophy: culture, collaboration, automation-DevOps lifecycle and toolchains-Git basics and GitHub workflows-Git branching strategies and version control best practices-Intro to CI/CD concepts.

UNIT II CI/CD Pipeline Implementation 9

Setting up CI pipelines (GitHub Actions / GitLab CI / Jenkins)-Automated build and test flows- Continuous delivery and rollback strategies-YAML files and configuration for pipelines-Integration testing with CI tools

UNIT III Containerization with Docker 9

Docker fundamentals: images, containers, Dockerfile-Creating Docker images for web apps-Docker Compose for multi-container applications-Docker Hub and image registries-Volume, networking, and environment configuration

UNIT IV Orchestration and Cloud Deployment 9

Introduction to Kubernetes: pods, services, deployments-Setting up clusters and namespaces-Helm basics for Kubernetes deployment-Deploying apps on AWS (EC2, S3, Elastic Beanstalk), GCP, or Azure-DevOps as a Service (Render, Vercel, Railway)

UNIT V Monitoring, Logging, and Security 9

Application logging and log aggregation (ELK stack, Prometheus)-Error tracking (Sentry, New Relic)-System and app monitoring tools-Security in CI/CD pipelines (secrets management, code scanning)-Backup, scaling, and disaster recovery basics

TOTAL :45 PERIODS

COURSE OUTCOME

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

CO1 Explain DevOps concepts, tools, and cultural changes in software delivery.
CO2 Apply Git, GitHub Actions, or Jenkins to build basic CI/CD pipelines.
CO3 Implement and deploy web apps using Docker and Docker Compose.
CO4 Analyze Kubernetes features like pods, services, and deployments.
CO5 Deploy full-stack apps on cloud platforms like AWS, Azure, or Render.
CO6 Evaluate app reliability using monitoring, logging, and recovery tools..

TEXT BOOKS:

1. Mitch Thomas, “DevOps Bootcamp: Web Applications Deployment Guide” ,.Packt, 2023
2. Gene Kim et al., “The DevOps Handbook” (Updated Edition) , 2021
3. Mikael Krief, “Learning DevOps: Continuously Deliver Better Software” , 2022, Packt
4. Richard Bullington-McGuire, “Docker for Developers” , 2023, O’Reilly
5. Brendan Burns, “Kubernetes: Up and Running” (3rd Ed.),O'Reilly, 2022

REFERENCE BOOKS:

1. Kief Morris, Infrastructure as Code ,O’Reilly, 2021
2. Google SRE team, Site Reliability Engineering ,O’Reilly
3. Shivakumar Gopalakrishnan, Hands-On Kubernetes on Azure, Packt, 2022

WEB REFERENCES:

1. https://docs.github.com/actions – GitHub Actions Docs
2. https://docs.docker.com/ – Docker Docs
3. https://kubernetes.io/docs/ – Kubernetes Official Docs
4. https://learn.microsoft.com/en-us/azure/devops/ – Azure DevOps Docs
5. https://www.jenkins.io/doc/ – Jenkins Documentation

CO-PO-PSO MAPPING

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2	2	2	2	2	2	2		2	3
CO2	3	2	3	2	3			2	1	3	3
CO3	3	2	3	2	3			2		2	2
CO4	3	2	3	2	3					2	2
CO5	3	2	3		3	2				2	2
CO6	3	2	3	3	3	2	2			3	3

Internal Assessment				End Semester Examinations
Assessment I (100 Marks)		Assessment II (100 Marks)		
Individual Assignment / Case Study / Seminar / Mini Project	Written Test	Individual Assignment / Case Study / Seminar / Mini Project	Written Test	Written Examinations
40	60	40	60	100
40%				**60 %**

23IT4008	UI / UX DESIGN PRINCIPLES AND TOOLS	L	T	P	C
		3	0	0	3

COURSE OBJECTIVE:

- Understand the basic concepts and importance of UI and UX design.
- Learn the process of design thinking and its stages in product design.
- Apply user research methods to identify real-world design problems.
- Use tools like Figma or Adobe XD to create wireframes and prototypes.
- Design interfaces that are visually appealing, accessible, and brand-aligned.
- Conduct usability testing to refine and improve design solutions.

UNIT - I Foundations of design 9

Difference between UI and UX Design-Core stages of Design Thinking-Divergent vs Convergent Thinking-Brainstorming and Gamestorming techniques -Observational empathy techniques

UNIT - II Foundations of UI Design 9

Visual and UI Principles - UI elements and patterns (e.g., buttons, toggles, modals) - Interaction behaviors (hover, transitions, feedback)- Branding and its impact on UI- Style guides and design systems

UNIT - III Foundations of UX Design 9

Importance and process of UX design - UX methodology and frameworks - Research in UX design: user interviews, surveys, contextual inquiry-Tools for UX research (e.g., Dovetail, Maze) -Identifying user needs and business goals

UNIT - IV Wireframing, Prototyping and Testing 9

Sketching for design ideation -Red routes and critical path flows -Responsive design concepts-Wireframes and wireflows - High-fidelity mockups (Figma, Adobe XD)- Building interactive prototypes - Usability testing methods and synthesizing test results -Iteration techniques based on feedback

UNIT - V Research, Designing, Ideating, & Information Architecture 9

Problem statement framing - Research methods and persona creation -Solution ideation and storyboarding- Creating user stories and flow diagrams - Mapping task flows and user journeys - Information architecture: hierarchy, navigation, content organization Scenarios - Flow Diagrams - Flow Mapping - Information Architecture

TOTAL : 45 PERIODS

COURSE OUTCOME(S):

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

CO1 Explain the difference between UI and UX with key design principles..

CO2 Use design thinking and research to understand user needs..

CO3 Create wireframes and prototypes using design tools.
CO4 Design user-friendly and visually consistent interfaces.
CO5 Test designs with users and improve them based on feedback.
CO6 Build user flows, personas, and organize content effectively.

TEXT BOOKS:

1. Cooper, R. Reimann, D. Cronin, and C. Noessel, About Face: The Essentials of Interaction Design, 4th ed. Hoboken, NJ: Wiley, 2014.
2. J. J. Garrett, The Elements of User Experience: User-Centered Design for the Web and Beyond, 2nd ed. Berkeley, CA: New Riders, 2010.
3. D. A. Norman, The Design of Everyday Things, Rev. ed. New York, NY: Basic Books, 2013

REFERENCE BOOKS:

1. Steve Krug, Don't Make Me Think: A Common Sense Approach to Web Usability, New Riders, 2023.
2. Jeff Gothelf, Josh Seiden, Lean UX: Designing Great Products with Agile Teams, O'Reilly Media, 2023.
3. Frank Spillers, UX Design and Usability Mentor Book, CRC Press, 2023.
4. Scott Hurff, Designing Products People Love: How Great Designers Create Successful Products, O'Reilly Media, 2023.
5. Will Grant, UX Storytellers: Connecting the Dots in User Experience, UX Book Club, 2023.

CO-PO-PSO MAPPING

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	2	2	2			2	2		2		2
CO2	3	3	2	2	2	2		2	2	1	3
CO3	3		3		3			2	2		2
CO4	3		3		3	2			2		2
CO5	2	2	3	3	2				2	2	3
CO6	3	2	2	2	2	2			2		3

<table>
<tr><th colspan="4">Internal Assessment</th><th rowspan="2">End Semester Examinations</th></tr>
<tr><th colspan="2">Assessment I (100 Marks)</th><th colspan="2">Assessment II (100 Marks)</th></tr>
<tr><th>Individual Assignment / Case Study / Seminar / Mini Project</th><th>Written Test</th><th>Individual Assignment / Case Study / Seminar / Mini Project</th><th>Written Test</th><th>Written Examinations</th></tr>
<tr><td>40</td><td>60</td><td>40</td><td>60</td><td>100</td></tr>
<tr><td colspan="4"><b>40%</b></td><td><b>60 %</b></td></tr>
</table>