

Front-End Development

1. Introduction to Front-End Development

- A. What is Front-End Development?
- B. Client vs Server vs Full-Stack Roles
- C. Modern Web Architecture & SPA (Single-Page Applications)
- D. Tools & Environment Setup (VS Code, Git, Browser DevTools)
- E. Package Management (NPM/Yarn)

2. Core Web Technologies

- A. HTML5: Structure, Semantics, Forms, Media
- B. CSS3: Syntax, Selectors, Box Model, Typography
- C. Responsive Design: Flexbox & CSS Grid
- D. CSS Frameworks: Bootstrap / Tailwind CSS
- E. Accessibility (WCAG Best Practices)
- F. Browser Compatibility & Debugging

3. JavaScript for Front-End

- A. Fundamentals: Variables, Types, Functions, Scope
- B. ES6+ Modern Features: Let/Const, Arrow Functions, Modules
- C. DOM Manipulation & Events
- D. Async JavaScript: Promises, Async/Await, Fetch API
- E. Error Handling & Debugging Techniques
- F. Local Storage & Browser APIs

4. UI/UX Design Principles

- A. UI vs UX: Concepts & Best Practices
- B. Color Theory, Typography & Layout
- C. Wireframes & Prototyping
- D. Tools: Figma / Adobe XD / Sketch
- E. Usability & Accessibility Standards
- F. Design Thinking & User Research
- G. Responsive UI Patterns

5. Introduction to React

- A. Why React? History & Ecosystem
- B. JSX & Component Architecture
- C. Props, State & Component Lifecycle
- D. Handling Lists, Keys & Forms

- E. Styling Components (CSS Modules, Styled Components)
- F. Working with React DevTools

6. Advanced React Concepts

- A. React Hooks (useState, useEffect, useContext)
- B. Context API for State Management
- C. Routing with React Router
- D. API Integration (Axios / Fetch with REST APIs)
- E. Performance Optimization (Memoization, Lazy loading)
- F. Error Boundaries & Best Practices

7. State Management & Modern Frameworks

- A. Overview of Global State Management Solutions
- B. Redux Fundamentals & Toolkit
- C. Alternative Patterns: Zustand / Recoil
- D. Next.js Basics for SSR/SSG
- E. TypeScript Integration Basics
- F. Component Libraries (Material-UI, Ant Design)

8. Version Control & Collaboration

- A. Git Essentials & Branching Workflow
- B. GitHub for Teams & Code Reviews
- C. Pull Requests & Merging Strategies
- D. CI/CD Basics • Deploying to Vercel / Netlify

9. Testing & Quality Assurance

- A. Unit Testing with Jest
- B. Component Testing with React Testing Library
- C. End-to-End Testing (Cypress / Playwright)
- D. Linting and Formatting (ESLint, Prettier)
- E. Performance & Accessibility Audits (Lighthouse)

Mini Projects:

- Responsive Portfolio Website
- React E-Commerce Interface with API Integration
- UI/UX Prototype to Production Workflow
- Dashboard with State Management & Routing
- Performance & Accessibility Optimization Report