

# FULL STACK WEB DEVELOPMENT COURSE

## Introduction to Web Technologies

- What is Web Development?
- Role of HTML, CSS, and JavaScript

## HTML Basics

- Understanding HTML
- Structure of an HTML Document
- HTML Tags and Elements
- Text Formatting and Semantics
- Lists and Tables
- Links, Images
- Iframes, videos
- Forms

## CSS Fundamentals

- Introduction to CSS
- CSS Syntax and Selectors
- Working with Colors and Backgrounds
- Box Model and Layout
- Typography and Fonts
- Position, z-Index, Links
- Lists, Box Model, display

## Intermediate CSS

- CSS Flexbox
- CSS Grid
- Responsive Design with Media Queries
- Transform
- Keyframes
- Transitions, animations
- ToolTips

## JavaScript Fundamentals

- Introduction to JavaScript
- Variables and Data Types
- Operators and Expressions
- Control Structures (if statements, loops)
- Functions and Scopes
- Arrays and Objects

## **Intermediate JavaScript**

- ES6+ Features (Arrow Functions, Promises, etc.)
- Working with Events
- Asynchronous Programming Basics
- Classes
- Js Objects
- Debugging

## **Project: Building a Simple Webpage**

- Combine HTML, CSS, and JavaScript
- Create a Basic Interactive Webpage

## **React**

- What is ReactJS?
- Why use ReactJS?
- ReactJS ecosystem
- Setting up the development environment

## **Components**

- What are components? Creating components
- Component props, Component state, Component lifecycle methods

## **JSX**

- What is JSX?
- JSX syntax, Using JSX in React components
- Differences between JSX and HTML

## **Rendering**

- Rendering elements, Rendering components
- Conditional rendering, List rendering

## **Events**

- Handling events, Event binding
- Event object, Synthetic events

## **Forms**

- Controlled vs uncontrolled components, Handling form data, Form validation

## **Hooks**

- What are hooks?
- useState hook, useEffect hook, useContext hook
- Creating custom hooks

## **Routing**

- React Router, Setting up routes, Route parameters, Navigation

## **Redux**

- What is Redux?
- Redux principles, Redux store, Redux actions, Redux reducers

## **Server-Side Rendering**

- What is server-side rendering?
- Setting up server-side rendering with React
- Benefits of server-side rendering

## **Testing**

- Testing React components
- Unit testing, Integration testing, End-to-end testing

## **Best practices**

- Component organization, Performance optimization, Code splitting, Error handling

## **Advanced topics**

- Higher-order components
- Render props, Context API, React Native

## **Conclusion**

- Recap of ReactJS concepts, Next steps for learning ReactJS

## **Introduction to Node.js and Backend Development**

- Introduction to server-side programming and Node.js
- Installing Node.js and NPM (Node Package Manager)
- Understanding the event-driven architecture of Node.js
- Building and running your first Node.js application
- Working with modules and the CommonJS pattern

## **Node.js Core Modules and RESTful APIs**

- Exploring core modules: fs (File System), http, https, path, util, etc.
- Creating an HTTP server using the http module
- Handling routing and requests using core modules
- Introduction to RESTful APIs
- Building RESTful APIs with Express.js
- Using Postman for API testing

## **NPM, Package Management, and Frontend Basics**

- Using NPM to manage dependencies and packages
- Initializing a Node.js project with package.json
- Installing and using third-party packages
- Semantic versioning and managing package versions
- Introduction to HTML, CSS, and JavaScript for frontend development
- Setting up a basic HTML/CSS/JS frontend

## **Templating Engines and Frontend-Backend Integration**

- Templating engines (e.g., EJS) for dynamic content
- Integrating EJS templates with Express.js
- Sending data from backend to frontend using templates
- Client-side JavaScript and handling DOM manipulation
- AJAX and Fetch API for making asynchronous requests to the server

## **Working with Databases and RESTful API Integration**

- Connecting to databases (e.g., MongoDB) using libraries like Mongoose
- Handling CRUD operations (Create, Read, Update, Delete) with MongoDB and Express.js
- Consuming RESTful APIs from the frontend using Fetch or Axios
- Displaying data from the backend in the frontend

## **Authentication, Security and Frontend Frameworks**

- Introduction to user authentication and authorization
- Implementing user registration and login functionality
- Using packages like Passport.js for authentication strategies
- Securing Express.js applications: best practices and middleware
- Introduction to frontend frameworks (e.g., React, Vue.js)

## **Advanced Frontend-Backend Interaction**

- Building interactive frontend components with React/Vue.js
- State management in frontend frameworks
- Real-time communication between frontend and backend using WebSockets
- Implementing features like live chat or notifications

## **Deployment, Final Project and Wrap-up**

- Deploying Node.js applications and frontend assets to production servers
- Students work on a final full-stack project integrating concepts from the course
- Presenting and showcasing final projects
- Review of key topics, Q&A session, and discussions about future learning paths
- Sharing resources for further learning and exploration in both frontend and backend development