

React JS App Development with Vibe Coding

Course Duration: 32 Hours

Course code: R-JS-ADVC

1. Course Overview

This course provides a hands-on, project-driven introduction to React JS using the Vibe Coding approach. Learners will not only understand React fundamentals but also practice building applications step by step in an engaging, collaborative style. The course emphasizes component-driven development, hooks, state management, routing, and API integration. By the end of the course, participants will have the skills to design, develop, and deploy modern React applications with confidence.

2. What you'll learn?

By the end of this course, you will be able to:

- Understand the React ecosystem and architecture.
- Build functional and class-based components.
- Use JSX, props, and state effectively.
- Manage side effects using React Hooks (useState, useEffect, useContext, useReducer).
- Implement routing and navigation with React Router.
- Consume and manage REST APIs within React apps.
- Apply state management techniques with Context API and Redux.
- Style components using CSS Modules, Styled Components, and Tailwind.
- Optimize React apps with performance best practices.
- Deploy React applications to Vercel, Netlify, or AWS Amplify.

3. Target Audience

- Beginners transitioning into React development.
- JavaScript developers seeking to master frontend frameworks.

- Students and professionals preparing for React developer roles.
- Teams adopting the Vibe Coding learning method for real-world projects.

4. Pre-Requisites

Familiarity with:

- HTML5, CSS3, and JavaScript (ES6+).
- Basics of Git and version control.
- Understanding of modern web development concepts.

5. Course content

Day 1: Vibe Coding Foundations + Frontend UI with React

- Intro to Vibe Coding (Karpathy's concept + Cursor AI)
- Setup: Cursor, Render, Netlify
- Create React App via Prompt
- Components, Props, State (prompting style)
- Tailwind CSS Styling by Prompt
- useState, useEffect with live generation

Labs

- Lab 1: Layout (Header, Footer, Navbar, Sidebar)
- Lab 2: Components: Card, Button, Table, Modal
- Lab 3: Prompting Forms with Real-Time Validation

Day 2: Routing, Backend API, Database (Fullstack Integration)

- React Router (Pages, Links, Params)
- Prompt Express Server & CRUD API
- MongoDB/MySQL integration (via prompt)
- Axios setup for frontend ↔ backend connection
- RESTful architecture & API testing

Labs

- Lab 4: Create Product Manager (CRUD)
- Lab 5: Dynamic Route (Product Details)
- Lab 6: Filter, Search & Paginate Items

Day 3: Auth, Role-based Access, Uploads, Charts, Payments

- JWT Auth via Prompt (Register/Login)
- Role-Based Access (User vs Admin)
- File Upload (image/pdf via Multer or Firebase)
- Add Recharts or Chart.js dashboard
- Stripe Payment Integration via Prompt
- Add Search, Filter & Category system

Labs

- Lab 7: Auth Flow + Dashboard Guard
- Lab 8: Role-based Route Protection
- Lab 9: File Upload + Preview & Storage
- Lab 10: Analytics Dashboard with Bar/Pie Charts
- Lab 11: Stripe Checkout for Simple Product Flow

Day 4: Testing, PDF Export, Deployment, Final App

- Unit Testing with Prompts
- Describe tests → AI writes .test.jsx
- Prompt test files for: Form input, API response, Protected route

Labs

- Lab 12: Final Testing, Deployment, and Export via AI & jsPDF
- Lab 13 (Capstone): Build & deploy: Login + Dashboard, Role Guard + CRUD, Charts + File Upload, PDF Export + Stripe Checkout