



React Development Course

Course Overview

Transform your career with our intensive React development course. Learn to build modern, dynamic web applications using React, the most popular frontend library in the world. This course combines theoretical knowledge with extensive hands-on practice to ensure you're job-ready in just 8 weeks.

Why Choose This Course?

- **Industry-Driven Curriculum:** Learn what companies actually use
- **Project-Based Learning:** Build real-world applications
- **Modern Stack:** Latest React features and best practices
- **Job-Ready Skills:** Graduate with a portfolio of projects
- **Practical Approach:** 60% hands-on coding

What You'll Learn

- Modern JavaScript (ES6+)
- React fundamentals and advanced concepts
- State management solutions
- API integration
- Testing and deployment

- Industry best practices

Course Highlights

- **Comprehensive Learning:** From basics to advanced React
- **Real Projects:** Build portfolio-worthy applications
- **Interactive Sessions:** Daily live coding and Q&A
- **Small Batch Size:** Personalized attention
- **Career Support:** Interview preparation and guidance

Tools & Technologies

- React 18+
- Redux Toolkit
- React Router
- Modern JavaScript
- REST APIs
- Git & GitHub
- VS Code
- Testing tools

Who Should Take This Course?

- Web developers wanting to learn React
- JavaScript developers
- College graduates interested in frontend development
- Professionals looking to switch careers
- Backend developers wanting frontend skills

Prerequisites

- Basic HTML/CSS knowledge
- JavaScript fundamentals
- ES6 concepts understanding

- Basic command line familiarity

Our Support

- Dedicated mentorship
- Code reviews
- Interview preparation
- Resume building
- Project guidance
- Career counseling

What Makes This Course Special

- Industry-aligned curriculum
- Practical approach
- Real-world projects
- Portfolio development
- Career guidance
- Networking opportunities

By The End You Will

- Build production-ready React applications
- Implement complex state management
- Create reusable component libraries
- Handle API integration effectively
- Deploy and optimize React apps
- Write clean, maintainable code
- Have a strong project portfolio

Projects You'll Build

1. **Mini Projects**

- Interactive UI components

- Todo application
- Weather dashboard
- Shopping cart
- Movie database

2. Final Project Options

- E-commerce platform
- Social media dashboard
- Project management tool
- Content management system

Course Duration: 8 Weeks

Course Overview

- Duration: 8 weeks
- Daily Schedule:
 - Theory (1.5 hours)
 - Live coding demonstration (1 hour)
 - Hands-on practice (2.5 hours)
 - Q&A and troubleshooting (1 hour)

Month 1: JavaScript & React Fundamentals

Week 1: Modern JavaScript & React Basics

- **Day 1-2: Modern JavaScript Essentials**
 - ES6+ features
 - Arrow functions
 - Destructuring
 - Array methods (map, filter, reduce)
 - Promises and async/await
 - Practice: Modern JS exercises

- **Day 3-4: React Introduction**
 - What is React?
 - Virtual DOM
 - Create React App
 - JSX syntax
 - Components basics
 - Practice: First React app
- **Day 5: Components & Props**
 - Functional components
 - Working with props
 - Component composition
 - Practice: Profile card component

Week 2: React Core Concepts

- **Day 1-2: State & Events**
 - useState hook
 - Event handling
 - Forms basics
 - Input handling
 - Practice: Interactive form
- **Day 3-4: Effects & Lifecycle**
 - useEffect hook
 - Dependencies
 - Cleanup
 - API calls
 - Practice: Data fetching app
- **Day 5: Mini Project**
 - Todo application

- Combining states
- Local storage
- CRUD operations

Week 3: React Ecosystem

- **Day 1-2: React Router**
 - Router setup
 - Routes and navigation
 - Dynamic routes
 - Protected routes
 - Practice: Multi-page app
- **Day 3-4: API Integration**
 - Fetch API
 - Axios
 - Error handling
 - Loading states
 - Practice: Weather dashboard
- **Day 5: Mini Project**
 - Movies browser
 - API integration
 - Search functionality
 - Responsive design

Week 4: State Management & Styling

- **Day 1-2: Context API**
 - Global state
 - Context providers
 - useContext hook
 - Practice: Theme switcher

- **Day 3-4: Styling in React**
 - CSS Modules
 - Styled-components
 - Tailwind CSS basics
 - Practice: Styled components
- **Day 5: Mini Project**
 - Shopping cart
 - Context implementation
 - Cart functionality
 - Styling practice

Month 2: Advanced Concepts & Project Development

Week 5: Advanced React & Performance

- **Day 1-2: Advanced Hooks**
 - useRef
 - useMemo
 - useCallback
 - Custom hooks
 - Practice: Custom hooks
- **Day 3-4: Performance**
 - React.memo
 - Code splitting
 - Lazy loading
 - Practice: Optimization
- **Day 5: Redux Toolkit Basics**
 - Redux concepts
 - Store setup

- Reducers
- Practice: Counter with Redux

Week 6: Forms & Validation

- **Day 1-2: Form Handling**
 - Form events
 - Controlled components
 - Form submission
 - Practice: Registration form
- **Day 3-4: Form Validation**
 - Input validation
 - Error handling
 - Success states
 - Practice: Validated form
- **Day 5: Mini Project**
 - Multi-step form
 - Validation logic
 - Progress tracking
 - Form submission

Week 7: Final Project Week 1

- **Day 1: Project Planning**
 - Requirements gathering
 - Component structure
 - State management plan
- **Day 2-3: Core Implementation**
 - Basic components
 - Routing setup
 - State management

- **Day 4-5: Feature Development**

- Main features
- API integration
- Authentication

Week 8: Final Project Week 2

- **Day 1-2: Feature Completion**

- Remaining features
- Error handling
- Loading states

- **Day 3-4: Styling & Polish**

- Responsive design
- UI improvements
- Performance optimization

- **Day 5: Deployment**

- Build preparation
- Deployment (Netlify/Vercel)
- Project presentation

Projects Overview

Mini Projects

1. Profile Card Component
2. Todo Application
3. Weather Dashboard
4. Movie Browser
5. Shopping Cart
6. Multi-step Form

Final Project Options

1. E-commerce Store
 - Product listing
 - Cart functionality
 - Checkout process
2. Social Media Dashboard
 - Feed display
 - Post creation
 - User interactions
3. Task Management System
 - Task CRUD
 - Categories
 - Progress tracking

Course Deliverables

1. Course materials and recordings
2. Project source code
3. Certificate of completion
4. Portfolio of projects
5. Interview preparation guide

Technical Requirements

- Laptop with minimum 8GB RAM
- Node.js installed
- VS Code or preferred IDE
- GitHub account
- Good internet connection

Assessment Methods

1. Weekly coding assignments (30%)

2. Mini-projects (30%)
3. Final project (40%)

Learning Outcomes

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

1. Build React applications from scratch
2. Implement state management solutions
3. Create reusable components
4. Handle API integration
5. Deploy production-ready applications
6. Follow React best practices

Key Skills Covered

- Modern JavaScript (ES6+)
- React fundamentals
- Hooks and state management
- Routing and navigation
- API integration
- Styling solutions
- Deployment basics

Prerequisites

- Basic HTML/CSS knowledge
- JavaScript fundamentals
- Familiarity with ES6
- Basic command line knowledge