React & JavaScript Training Syllabus

CorporateSpace Web Development Academy

Hands-On Training for Building Modern Web Applications

Last Updated: August 2025

https://corporatespace.in/react-javascript-training

Contents

1 Course Overview

This React & JavaScript Training program equips participants with the skills to build dynamic, scalable, and interactive web applications using modern JavaScript (ES6+) and the React framework. Through hands-on coding, real-world projects, and expert-led instruction, learners will master JavaScript fundamentals, React's component-based architecture, state management with Redux, and deployment on platforms like Vercel or Netlify. The curriculum is designed for beginners and professionals, aligning with real-world use cases in e-commerce, fintech, and SaaS industries.

1.1 Course Objectives

- Master core JavaScript concepts, including ES6+ features and asynchronous programming.
- Build interactive user interfaces with React components, hooks, and Context API.
- Implement state management using Redux for complex applications.
- Develop and integrate REST APIs with Node.js and Express.
- Create and deploy a functional e-commerce dashboard through a real-time project.

1.2 Course Duration

- Total: 24–30 hours
- Format: Live sessions (onsite, online via Zoom/Google Meet, or hybrid)
- Schedule: Flexible, including weekend batches for corporate teams

1.3 Target Audience

- Software Engineers seeking frontend development expertise
- Professionals transitioning to modern JavaScript frameworks
- College Students (BCA, MCA, BE) targeting web development roles
- Startups and Founders building React-based web applications

1.4 Prerequisites

- Basic knowledge of HTML and CSS is recommended.
- No prior JavaScript experience required, as the course covers JavaScript from the ground up.

2 Module 1: Core JavaScript (ES6+)

2.1 JavaScript Fundamentals

- Overview of JavaScript: Role in web development
- Variables, data types, and operators
- Control structures: Loops, conditionals, and functions

2.2 Modern JavaScript (ES6+)

- Arrow functions, destructuring, and template literals
- Modules: Import and export syntax
- Spread/rest operators and default parameters

2.3 Asynchronous JavaScript

- Promises and async/await syntax
- Handling asynchronous operations with fetch API
- Error handling in asynchronous code

2.4 Lab: JavaScript Basics

- Writing ES6+ functions for data manipulation
- Creating an async function to fetch data from a public API

3 Module 2: React Fundamentals

3.1 Introduction to React

- Overview of React: Features and component-based architecture
- Setting up a React project with Vite or Create React App
- Understanding JSX syntax and rendering

3.2 Components and Props

- Creating functional and class components
- Passing and managing props
- Component lifecycle and rendering behavior

3.3 Lab: Building Components

- Creating a reusable component (e.g., product card)
- Passing props to display dynamic data

4 Module 3: React Hooks and Context API

4.1 React Hooks

- Introduction to hooks: useState, useEffect, useReducer
- Managing component state and side effects
- Custom hooks for reusable logic

4.2 Context API

- Managing global state with Context API
- Creating and consuming context for app-wide data
- Alternatives to Context for state management

4.3 Lab: Hooks and Context

- Building a form with useState and useEffect
- Implementing a theme toggle using Context API

5 Module 4: State Management with Redux

5.1 Redux Fundamentals

- Introduction to Redux: Store, actions, and reducers
- Setting up Redux Toolkit for simplified state management
- Connecting Redux to React components

5.2 Advanced Redux

- Handling asynchronous actions with Redux Thunk
- Organizing state with slices
- Debugging with Redux DevTools

5.3 Lab: Redux Integration

- Setting up a Redux store for a shopping cart
- Dispatching actions to update cart state

6 Module 5: Building REST APIs with Node.js and Express

6.1 Node.js and Express Basics

- Setting up a Node.js project with Express
- Creating RESTful routes for CRUD operations
- Handling requests and responses

6.2 Integrating APIs with React

- Fetching data from a REST API in React
- Handling API errors and loading states
- Testing APIs with Postman

6.3 Lab: API Development

- Building a simple Express API for product data
- Connecting the API to a React frontend

7 Module 6: Performance Optimization Techniques

7.1 Optimizing React Applications

- Using memoization: React.memo, useMemo, and useCallback
- Lazy loading components with React.lazy and Suspense
- Optimizing re-renders with shouldComponentUpdate

7.2 Best Practices

- Code splitting for faster load times
- Minimizing bundle size with tree shaking
- Improving accessibility in React apps

7.3 Lab: Performance Optimization

- Implementing lazy loading for a component
- Optimizing a list component with React.memo

8 Module 7: Deploying React Applications

8.1 Cloud Deployment

- Preparing a React app for production
- Deploying to Vercel: Environment setup and configuration
- Deploying to Netlify: Continuous deployment setup

8.2 Version Control with GitHub

- Setting up a Git repository for React projects
- Managing code versions and collaboration
- Automating deployments with GitHub Actions

8.3 Lab: Deployment

- Deploying a React app to Vercel
- Pushing project code to a GitHub repository

9 Module 8: Real-Time Project: E-Commerce Dashboard

9.1 Project Overview

- Designing an e-commerce dashboard with React
- Defining requirements: Product listing, cart, and user management

9.2 Implementation

- Building reusable React components for the dashboard
- Implementing state management with Redux
- Integrating with a Node.js/Express REST API
- Adding authentication and role-based access

9.3 Project Deliverables

- Functional e-commerce dashboard application
- Documentation of components and API endpoints
- Project codebase hosted on GitHub

10 Module 9: Career Development and Certification

10.1 Building a Web Development Portfolio

- Crafting a resume with React project highlights
- Showcasing the project codebase on GitHub
- Preparing for frontend developer roles

10.2 Certification and Support

- Earning a CorporateSpace React & JavaScript Certificate
- 30 days of post-training doubt-solving and code support
- Job and internship references (on request)

11 Course Materials and Tools

11.1 Provided Resources

- Comprehensive course slides and cheat sheets
- Access to recorded sessions for review
- Sample React project code and configurations
- Templates for API documentation

11.2 Tools Covered

- Visual Studio Code for coding
- GitHub for version control and project hosting
- Postman for API testing
- Vercel and Netlify for cloud deployment
- Node.js and Express for backend development

12 Program Delivery Options

12.1 Training Formats

- Onsite: At your company premises
- Online: Live sessions via Zoom/Google Meet
- Hybrid: Weekend batches for flexible learning

12.2 Customization for Corporate Teams

- Tailored modules based on company project needs
- Flexible scheduling for team availability
- Integration with company-specific tech stacks

13 Instructor Profile

- Assistant Professor and Full-Stack Developer with over 6 years of experience
- Delivered training to 15+ corporate teams globally
- Expertise in JavaScript, React, and full-stack web development