

MERN React Full Stack Web Developer

Duration: 15 days

Tools: MongoDB (NoSQL), MySQL (SQL), Redux Toolkit, React Query, RxJS, Context API

Prerequisites: Knowledge of Web Development and Any Programming language

Day 1: Web Fundamentals & HTML5 Deep Dive

Topics:

- Overview of Full Stack, MERN architecture
- Setting up local development environment (Node.js, npm, Git, VSCode)
- HTML5 semantic structure, best practices
- Forms and attributes (required, pattern, input types)
- SEO fundamentals using HTML
- Accessibility principles (ARIA, semantic tags)

Labs:

- Create a semantic HTML5 resume page
 - Form with validations and accessibility
 - Create structure for a single-page layout with placeholders
-

Day 2: CSS3, Flexbox, Grid & Responsive UI

Topics:

- CSS Selectors, Specificity & Inheritance
- Box Model, Positioning, Display Properties
- Flexbox and Grid systems
- Media Queries, Responsive Units
- CSS Variables, Transitions, and Animations

Labs:

- Build responsive card-based layout
 - Product grid with Flexbox and responsive styles
 - CSS animation banner
-

Day 3: JavaScript Core & DOM Manipulation

Topics:

- Primitive & Reference Types, Functions & Scopes
- Object & Array destructuring, Spread/Rest
- Event handling, bubbling, delegation
- DOM Manipulation: create, append, remove, traverse
- LocalStorage, SessionStorage

Labs:

- Dynamic ToDo App using Vanilla JS
 - Simple calculator UI with event delegation
 - Form auto-save with LocalStorage
-

Day 4: Modern JavaScript & Asynchronous Code

Topics:

- ES6+ (Arrow Functions, Template Strings, Modules)
- Promises, Async/Await, Fetch API
- Error handling (try/catch)
- Functional Array Methods: map, filter, reduce
- Module Bundling (ESM vs CommonJS)

Labs:

- API-driven joke fetcher using async/await
 - Table search/filter using reduce
 - Convert classic JS code to modern ES6+
-

Day 5: React.js Basics & Component Architecture

Topics:

- React introduction & ecosystem
- Functional vs Class components
- JSX, Virtual DOM, Component Tree
- useState, props, event handling
- Component composition & reusable components

Labs:

- Counter with multiple buttons
 - User Profile card with props
 - Build a modular "Blog Card" component
-

Day 6: React Routing & Lifecycle

Topics:

- React Router DOM (v6+): Route, Link, Navigate, useParams
- useEffect Hook & Cleanup
- Conditional rendering & fragment usage
- Nested routes and layout components

Labs:

- Multi-page blog with dynamic user route
- useEffect to fetch post data

- Navigation bar with active route styling
-

Day 7: Forms, Formik & Custom Hooks

Topics:

- Controlled vs Uncontrolled Inputs
- Form Validation with Formik + Yup
- Reusable Custom Hooks (useInput, useForm)
- Handling complex form inputs (checkboxes, selects)

Labs:

- Multi-step registration form
 - Reusable useForm() custom hook
 - Profile update form with validation
-

Day 8: Redux Toolkit & Context API

Topics:

- State management overview
- Redux Toolkit (createSlice, configureStore, asyncThunk)
- Middleware & DevTools
- React Context API for global UI state (theme, language)
- State comparison: Redux vs Context vs Props drilling

Labs:

- Redux store for authentication
 - Theme toggler using Context API
 - API integration with asyncThunk
-

Day 9: React Query + RxJS Introduction

Topics:

- React Query: useQuery, useMutation
- Query caching, stale time, pagination
- Error & loading states
- RxJS Introduction: Observables, Subjects, Operators (map, debounceTime)
- Integration RxJS with React using custom hooks

Labs:

- Product list with React Query (with loading/error states)
 - Search bar using RxJS observable and debounce
 - Mutation example (POST new item with feedback)
-

Day 10: Node.js Fundamentals & Express APIs

Topics:

- Intro to Node.js & Event Loop
- Core Modules: fs, path, http
- Express.js routing, middleware
- REST API design patterns (GET, POST, PUT, DELETE)
- Express error handling & logging

Labs:

- Build RESTful API for users
 - Create middleware for logging requests
 - User CRUD operations with Postman testing
-

Day 11: MongoDB & Mongoose

Topics:

- MongoDB basics (collections, documents)
- Mongoose: schema design, model creation
- MongoDB CRUD Operations with Mongoose
- Data validation and middleware (pre/post hooks)
- Relationships: references vs embedding

Labs:

- Build product & category schema
 - Create routes for CRUD with Mongoose
 - Relate user → orders → products
-

Day 12: MySQL & Sequelize ORM

Topics:

- MySQL Overview, ER Diagrams, Relationships
- Sequelize Setup, Model Definition
- Querying & Joins
- Sequelize Associations (1:1, 1:N, N:M)
- Using Sequelize with Express

Labs:

- Create user, order, and product MySQL models
 - Implement relational API endpoints
 - Filter & join order data with Sequelize
-

Day 13: Authentication, Authorization & Security

Topics:

- bcrypt for password hashing
- JWT Authentication flow
- Protecting routes with middleware
- Role-based access control
- CORS, Helmet, Rate Limiting

Labs:

- Secure login & register APIs with JWT
 - Middleware for protecting admin routes
 - Token refresh strategy (optional)
-

Day 14: Full Stack Integration & Deployment**Topics:**

- Connect React with Express (Axios, Proxy)
- Environment variables & .env files
- CORS setup and error boundaries
- MongoDB Atlas, MySQL Cloud (PlanetScale or Railway)
- Deploy:
 - React (Vercel/Netlify)
 - Backend (Render/Fly.io)
 - Database config for production

Labs:

- Connect frontend to backend APIs
 - Configure environments & deploy to cloud
 - Health check + logs + testing
-

Day 15: Final Capstone Project**Project Title: Inventory & Order Management System****Features:**

- User registration/login (JWT)
- Role-based access (Admin/User)
- CRUD for products (MongoDB)
- Order management (MySQL)
- React Dashboard (Admin & User)
- State: Redux Toolkit + React Query
- Data Streams: RxJS autocomplete or order notifier