



# **REACTJS**

## **Introduction to ReactJS:**

- What is ReactJS?
  - Features of ReactJS
  - Advantages of ReactJS
  - ReactJS vs Traditional JS Frameworks
  - ReactJS Architecture Overview
  - Setting Up ReactJS Environment
  - Hello World Example in React
- React JS Basics:**
- JSX Syntax & Rules
  - Components (Functional & Class Components)
  - Props & Prop Types
  - State and setState.
  - Event Handling in React

- Conditional Rendering
- Lists & Keys in React

## **React Component Lifecycle:**

- Lifecycle Overview
- Mounting Phase
  1. Constructor()
  2. Static getDerivedStateProps()
  3. Render()
  4. componentDidMount()
- Updating Phase
  1. Static getDerivedStateFromProps()
  2. shouldComponentUpdate()
  3. render()
  4. getSnapshotBeforeUpdate()
  5. componentDidUpdate()
- Unmounting Phase
  1. componentWillUnmount()

## **React Hooks (For Functional Components)**

- introduction to Hooks
- useState()
- useEffect()

- useContext()
- useReducer()
- useRef()
- useMemo() and useCallback()
- Custom Hooks

### **Handling Forms in React:**

- Controlled vs. Uncontrolled Components
- Handling Input Fields (Text, Checkbox, Radio, Select)
- Form Submission
- Form Validation (Manual & Using Libraries like Formik, Yup)
- Error Handling in Forms

### **React Router**

- Introduction to React Router
- Setting Up React Router
- Route, Switch & Link Components
- Nested Routing
- URL Parameters & Query Params
- Redirects and Navigation Programmatically
- 404 Pages and Fallback Routes

### **State Management**

- Local State (useState, useReducer)

- Global State (Context API)
- Third-Party State Management (Redux, MobX)
- Redux Basics: Actions, Reducers, Store
- Middleware (Redux Thunk, Redux Saga)
- Best Practices in State Management

## **API Calls in React**

- Fetch API and Axios Basics
- Making GET, POST, PUT, DELETE Requests
- Handling Responses & Errors
- useEffect with API Calls
- Loading and Error States
- Custom Hook for API Requests

## **Styling in React**

- Inline Styling
- CSS Stylesheets
- CSS Modules
- Styled Components
- Emotion CSS-in-JS
- Using UI Libraries (Bootstrap, Material-UI, Ant Design)
- Responsive Design in React

## **Advanced Concepts**

- Higher-Order Components (HOCs)

- Render Props
- Context API (Advanced Use Cases)
- Portals
- Error Boundaries
- Code Splitting and Lazy Loading (React.lazy & Suspense)
- Server-Side Rendering (SSR) with Next.js
- Progressive Web Apps (PWA)

## **Unit Testing in React**

- Importance of Testing
- Tools and Frameworks (Jest, React Testing Library, Enzyme)
- Writing Unit Tests for Components
- Mocking Props and Functions
- Snapshot Testing
- Testing Async Code and API Calls
- Code Coverage Reports

## **Deployment of React Applications**

- Building React App for Production
- Deployment to GitHub Pages
- Deployment to Netlify, Vercel, and Firebase
- Deployment to AWS S3 / EC2
- Environment Variables Setup
- SEO Best Practices for Deployment

- Performance Optimization Before Deployment.

