

## ReactJS Training Curriculum

For: New Employees / Junior Developers

Duration: 2 Weeks

### Objective:

To introduce ReactJS for building modern, component-based user interfaces with a focus on practical application development.

## Week 1: React Fundamentals

### Day 1: Introduction to React

- What is React?
- Benefits of using React
- Understanding SPA (Single Page Applications)
- Setting up a React project using Create React App (CRA)
- Project structure overview

### Day 2: JSX and Components

- What is JSX?
- Creating functional components
- Component structure and naming
- Props and prop types
- Reusability of components

### Day 3: State and Events

- useState hook
- Event handling (onClick, onChange, etc.)

- Conditional rendering
- Form handling

#### Day 4: Lists and Keys

- Rendering lists with map()
- Using keys in lists
- Dynamic list rendering
- Filtering and updating lists

#### Day 5: Component Communication

- Passing data between components (Parent to Child, Child to Parent)
- Lifting state up
- useEffect basics for side effects

#### Week 2: Intermediate to Advanced Concepts

##### Day 6: React Router

- Introduction to React Router
- Setting up routes and navigation
- Route parameters and useParams
- Nested routes

##### Day 7: useEffect and Lifecycle

- useEffect in depth
- Dependency array
- Cleanup functions
- Simulating lifecycle methods

## Day 8: Forms and Validation

- Controlled vs Uncontrolled components
- Form input handling
- Basic validation
- Using libraries like Formik or React Hook Form (optional)

## Day 9: Project: React Application

Build a mini React app (e.g., Task Manager or Notes App)

- Functional components
- Routing
- Forms and validation
- Component reusability

## Day 10: State Management and Final Touches

- Intro to Context API
- Global state management example
- Styling React apps (CSS Modules / Tailwind / Styled Components)
- Deployment basics (Vercel, Netlify)

## Assessment Task:

Build a fully functional ReactJS project including:

- At least 3 routed pages
- Interactive components (forms, lists, modals)
- State management with useState and useContext
- Aesthetic layout and responsiveness

## Resources:

- React Official Docs: <https://reactjs.org/>
- React Router Docs: <https://reactrouter.com/>
- FreeCodeCamp React: <https://www.freecodecamp.org/learn/>
- CodeSandbox: <https://codesandbox.io/>

Aalions-Infotech