

REACT.JS – COMPLETE CONTENT

1. INTRODUCTION TO REACT

- What is React?
 - Why React?
 - SPA vs MPA
 - React features
 - Virtual DOM
 - React vs Angular vs Vue
 - Setting up React environment
 - Node.js & npm
 - Create React App
 - Vite
-

2. JAVASCRIPT FOR REACT (PREREQUISITES)

- let, const, var
 - Arrow functions
 - Template literals
 - Destructuring
 - Spread & Rest operators
 - Array methods (map, filter, reduce)
 - Promises
 - Async / Await
 - Modules (import / export)
-

3. JSX (JavaScript XML)

- What is JSX
 - JSX syntax rules
 - Expressions in JSX
 - Conditional rendering
 - JSX vs HTML
-

4. COMPONENTS ★

4.1 Functional Components

- Creating components
- Reusability
- Component composition

4.2 Class Components (Legacy)

- Component lifecycle methods
 - state & props (class)
-

5. PROPS & STATE ★

- Props (passing data)
 - Props validation (PropTypes)
 - State
 - State immutability
 - Lifting state up
-

6. EVENT HANDLING

- Synthetic events
 - Handling form events
 - Passing arguments to handlers
-

7. CONDITIONAL RENDERING

- if/else
 - Ternary operator
 - Logical && operator
 - Switch cases
-

8. LISTS & KEYS

- Rendering lists
 - Keys and importance
 - Index as key (pros & cons)
-

9. FORMS & CONTROLLED COMPONENTS

- Controlled vs Uncontrolled components
 - Handling input fields
 - Form submission
 - Validation basics
-

10. HOOKS ★ ★ ★

10.1 Built-in Hooks

- useState
- useEffect
- useContext
- useRef
- useMemo
- useCallback
- useReducer
- useEffect

10.2 Custom Hooks

- Creating custom hooks
 - Reusability logic
-

11. CONTEXT API

- Prop drilling problem
 - Context creation
 - Provider & Consumer
 - Global state management
-

12. REACT ROUTER ★

- SPA routing
- BrowserRouter
- Routes & Route
- Link & NavLink
- useParams
- useNavigate
- useLocation
- Protected routes
- Lazy loading routes

13. API INTEGRATION

- Fetch API
- Axios
- GET, POST, PUT, DELETE
- Handling loading & errors
- Environment variables

14. STATE MANAGEMENT

14.1 Redux ★ ★

- Redux principles
- Store, Action, Reducer
- Redux Toolkit
- Middleware
- Async actions (Thunk)

14.2 Alternatives

- Zustand
- Recoil
- MobX

15. PERFORMANCE OPTIMIZATION

- React.memo
- useMemo
- useCallback
- Code splitting
- Lazy & Suspense
- Avoid unnecessary re-renders

16. STYLING IN REACT

- CSS & SCSS
- Inline styles
- CSS Modules
- Styled Components

- Tailwind CSS
 - Material UI (MUI)
 - Ant Design
-

17. ERROR HANDLING

- Error boundaries
 - try/catch in async code
 - Fallback UI
-

18. TESTING IN REACT

- Jest
 - React Testing Library
 - Unit testing
 - Integration testing
 - Mocking APIs
-

19. BUILD & DEPLOYMENT

- Production build
 - Environment configurations
 - Deployment options
 - Netlify
 - Vercel
 - AWS S3
 - Firebase
 - CI/CD overview
-

20. ADVANCED REACT ★ ★ ★

- Higher Order Components (HOC)
 - Render props
 - Portals
 - Reconciliation
 - Concurrent Rendering (overview)
 - Server Side Rendering (SSR)
 - Next.js overview
-

21. REAL-TIME PROJECT TOPICS

- Authentication & Authorization
 - JWT handling
 - Role-based UI
 - Admin vs User modules
 - Lazy loading
 - Dashboard design
 - API security
 - Form validation (Formik, Yup)
-

22. INTERVIEW & REAL-TIME QUESTIONS ★

- Props vs State
- Controlled vs Uncontrolled
- Hooks vs Class components
- useEffect lifecycle
- Virtual DOM
- Key usage
- Redux vs Context API
- Performance optimization techniques