ReactJS Mastery Roadmap (Beginner to Expert)

# 🟢 Stage 1: Basics & Fundamentals (React Core)

* 🔧 Concepts to Learn:
* • JSX
* • Components (Functional vs Class)
* • Props
* • useState
* • Event handling
* • Conditional rendering
* • Lists & keys
* 🧩 Components to Build:
* • Button with click event
* • Counter App (useState)
* • Greeting Component (accepts `name` as prop)
* • Toggle visibility component
* • Todo list (add/remove items)
* • Accordion
* • Tabs
* • Basic Form (text input + submit)
* • Modal (open/close on button click)
* • Tooltip

# 🟡 Stage 2: Intermediate React (Hooks, Lifting State, Context)

* 🔧 Concepts to Learn:
* • useEffect
* • useRef
* • useContext
* • Lifting state up
* • Controlled components (inputs)
* • Basic form validation
* • Custom hooks
* 🧩 Components to Build:
* • Live Search Bar (filters a list)
* • Image Carousel
* • Form with validation (name, email)
* • Multi-step form
* • Toast notification component
* • Theme toggle (light/dark)
* • Sidebar with toggle + animation
* • Dropdown menu
* • useDebounce Hook
* • useLocalStorage Hook
* • useClickOutside Hook

# 🟠 Stage 3: Advanced React (Routing, State Management, Async)

* 🔧 Concepts to Learn:
* • React Router v6
* • Code splitting / lazy loading
* • useReducer
* • Error boundaries
* • Redux / Zustand / Recoil (choose one)
* • Asynchronous data (fetch, axios, try/catch)
* • useCallback / useMemo / React.memo
* 🧩 Components to Build:
* • Multi-page app (Home, About, Contact using React Router)
* • Pagination Component
* • Auto-complete / Typeahead Input
* • Drag-and-drop list (react-beautiful-dnd)
* • File Uploader (progress bar)
* • Table with search/sort/pagination
* • Notification center (with read/unread)
* • Dark mode with Context + persistent localStorage
* • Real-time Dashboard with Charts (Recharts)

# 🔴 Stage 4: Expert React (Architecture, Optimization, Patterns)

* 🔧 Concepts to Learn:
* • Advanced patterns: Compound components, Render props, HOCs
* • Global state management (Redux Toolkit)
* • Authentication (JWT)
* • Authorization (RBAC)
* • Optimizations (React.memo, useMemo, lazy, Suspense)
* • Testing: Jest, React Testing Library
* • Accessibility (ARIA roles, keyboard support)
* 🧩 Components to Build:
* • Login / Signup / Logout (JWT-based)
* • Role-based access (admin/user views)
* • Dynamic Routing (/profile/:id)
* • Admin Panel (CRUD with fetch)
* • Kanban Board with drag & drop
* • Chat UI (WebSocket or polling)
* • Realtime Collaboration UI (e.g., shared task board)
* • Virtualized List (like react-window)
* • Custom form builder (drag-and-drop input elements)
* • Fully responsive e-commerce UI

# 🧠 Expert Tools & Ecosystem

* 🔧 Concepts to Learn:
* • React Router v6
* • Redux Toolkit / Zustand / Recoil
* • React Query / SWR
* • React Hook Form / Formik + Yup
* • Tailwind CSS / Styled Components
* • Framer Motion
* • Axios / Fetch API
* • Jest + React Testing Library
* • ESLint + Prettier
* • Storybook
* • Git + GitHub

# 🚀 Bonus Topics (When You’re Near Expert Level)

* 🔧 Concepts to Learn:
* • Server-side rendering (Next.js)
* • Static site generation
* • Micro frontends
* • Monorepo setup (Turborepo or Nx)
* • Custom webpack config
* • Integrating with backend APIs (GraphQL, REST)
* • CI/CD for React apps (GitHub Actions, Vercel, Netlify)
* • Component Libraries (Material UI, Chakra, Ant Design)
* • Accessibility testing
* • Internationalization (i18n)