---

## ⚛️ React — Complete Structured Topic Roadmap

---

### 🟢 1. React Basics & Core Concepts

These build your foundation.

- What is **React** and why it's used
- What is **one-way data binding**
- What causes **re-rendering** in React
- **Normal variable vs state variable**
- **React.createElement**
- **JSX & transpiler (Babel)** — what a transpiler does
- **Dynamic imports**
- **Default export vs named exports**
- **Import aliasing**
- **React.Fragment** — usage and benefits
- **Controlled vs uncontrolled components**
- **Passing data parent → child (props)**
- How to **pass props in React class components**
- What are **render props**
- **Create props proxy for HOC**
- **Benefits of HOC (Higher Order Component)**

---

### 🟠 2. React Components, Lifecycle & Rendering

Covers component lifecycle, rendering logic, and reusability.

- What is **conditional rendering**
- What is a **pure component**
- **React.PureComponent vs React.memo**
- **React.Memo** — how and when to use it
- What is the **Diffing Algorithm**
- What is **Reconciliation**
- What is **React Fiber** (core of React 16+)

- **Referential equality** — and its effect on re-renders
- **React Profiler** — measuring performance
- **React.Suspense** — lazy loading and fallback UI
- **React.lazy() implementation**
- **Error boundaries** — concept and example code
- **Render proper (render props)**
- **ForwardRef**
- **useRef()** — advantages & use cases
- **useLayoutEffect vs useEffect**
- **Cleanup function** in useEffect (and why it’s needed)
- **React lifecycle (functional)** — mount, update, unmount

---

### 🟡 3. React Hooks (Core & Custom)

Everything about hooks, from rules to custom hook patterns.

- **Rules of hooks & custom hooks**
- **useState**, **useEffect**, **useLayoutEffect** — differences
- **useCallback** and **useMemo** — optimizing re-renders
- **useRef** — storing mutable values across renders
- **useLocation**, **useNavigate**, and **MemoryRouter**
- **useNavigate vs useHistory** (React Router v6 change)
- **Custom hooks** — when & how to build
- Example: *Custom hook scenario (form handling, fetch wrapper, etc.)*

---

### 🟣 4. React Router

Routing concepts used in single-page apps.

- **Types of routers** — BrowserRouter, MemoryRouter, HashRouter
- **useNavigate() parameters**
- **Link props**
- **Dynamic routing**
- **Nested routing**
- **Passing data with routes**

- **React Router v6 improvements**

---

## 🔵 5. React Performance Optimization

Covers key tools and techniques to make React faster.

- **PureComponent, React.memo, useCallback**
- **Debouncing vs Throttling**
- **Axios interceptors**
- **Axios cancel tokens**
- **Dead code elimination** (Webpack, tree-shaking)
- **Code splitting & Lazy Loading code**
- **React Profiler** — identify performance bottlenecks
- **Referential equality** & re-render prevention
- **Shadow DOM** — conceptual difference with React’s virtual DOM
- **Event pooling & Synthetic events** — performance handling
- **Cleanup functions** — avoiding memory leaks
- **Polyfills** — browser compatibility
- **Webpack** — why code bundlers are needed
- **Why do we need code bundler?**

---

## 🧩 6. Event System

Understand how React handles events.

- **Synthetic events vs Native events**
- **Event pooling** (how React reuses events)
- **HTML sanitization** (preventing XSS)
- **Core principles of Redux (Event-driven state management)**

---

## 🔴 7. Redux & State Management

Understand global state and middleware handling.

- **Core principles of Redux**
  - Single source of truth
  - State is read-only

    - Changes via pure functions
- **Redux middleware** — purpose and examples
- **Axios interceptors** in Redux thunk
- **JWT & refresh token logic** — token renewal flow
- **Access token & refresh token handling**
- **Redux Toolkit** (modern Redux pattern)

---

## 🧠 8. Advanced & Internal React Concepts

These go deeper into React internals & ecosystem.

- **React Fiber architecture**
- **Reconciliation & Diffing algorithm**
- **React Suspense** for async loading
- **React Profiler** and performance debugging
- **ForwardRef & createRef**
- **Portals** — rendering outside the main DOM tree
- **Shadow DOM vs Virtual DOM**
- **MemoryRouter** (for non-browser environments)
- **Dead code elimination, Polyfills, Transpiling**
- **SEO in React (CSR vs SSR)**
    - *How CSR affects SEO*
    - *Strategies to improve SEO in SPA*

---

## ⚙️ 9. Async, Network & External Integrations

Connects React with APIs & async behavior.

- **Axios interceptors**
- **Axios cancel tokens**
- **Debouncing & Throttling in React**
- **Promises & async handling in React**
- **useEffect async patterns**
- **Fetching new access token using refresh token**
- **JWT verification & structure**

- **HTML sanitization** — handling user inputs securely

---

### 🧩 10. Miscellaneous (Ecosystem & Dev Tools)

Additional real-world concepts you'll use in production React apps.

- **React Profiler** (React DevTools)
- **Webpack** & **Babel**
- **Polyfills**
- **Transpiler** (why we need it)
- **SEO optimization in React apps**
- **CSR vs SSR**
- **Dynamic imports / Lazy loading**
- **Event pooling, Synthetic events**
- **npm vs npx**

---

### ✅ Summary Learning Path (Recommended Order)

1️⃣ React Basics (components, props, rendering)
2️⃣ Lifecycle & Hooks
3️⃣ Routing (React Router)
4️⃣ Performance Optimization
5️⃣ Redux / State Management
6️⃣ Advanced React (Fiber, Suspense, Portals)
7️⃣ Async Patterns & Axios
8️⃣ Build Tools (Webpack, Babel, Polyfills, SEO)

---