**✅ Phase 1: React Core Concepts (Day 1 - Day 5)**

🎯 *Goal: Build UI using React components with JSX.*

**📅 Day 1: JSX & Functional Components**

* What is JSX?
* Embedding expressions
* Functional Components vs Class Components
* Return single parent element

**📅 Day 2: Props and Reusability**

* Props basics (passing data to components)
* Prop drilling
* Component reusability techniques

**📅 Day 3: useState Hook**

* State management in function components
* Initialize and update state
* Handle multiple state variables

**📅 Day 4: Event Handling & Conditional Rendering**

* onClick, onChange handlers
* if, ? :, && conditions in JSX
* Toggle visibility

**📅 Day 5: Lists and .map()**

* Rendering arrays in JSX
* Key prop importance
* Build reusable list items

🛠 Mini Projects:

* Counter App
* To-do List App
* Simple Calculator

**✅ Phase 2: React Intermediate (Day 6 - Day 10)**

🎯 *Goal: Handle side effects and routing.*

**📅 Day 6: useEffect Basics**

* useEffect for side effects
* Cleanup function
* Dependency array explained

**📅 Day 7: Component Lifecycle with useEffect**

* Simulating mount, update, unmount
* Timers, event listeners in useEffect

**📅 Day 8: React Router Setup**

* Install react-router-dom
* BrowserRouter, Route, Link, Navigate
* Dynamic routes with useParams

**📅 Day 9: Form Handling**

* Controlled vs uncontrolled inputs
* Form submission, validations

**📅 Day 10: Lifting State Up**

* Pass state from child to parent
* Share state among components

🛠 Mini Projects:

* Weather App using API
* Blog with Multi-pages
* Form Validator

**✅ Phase 3: React Advanced (Day 11 - Day 15)**

🎯 *Goal: Optimize, manage logic, and reuse.*

**📅 Day 11: useRef, useMemo, useCallback**

* useRef for DOM references
* useMemo for expensive computation
* useCallback to memoize functions

**📅 Day 12: useContext API**

* Create and provide context
* Use context across components
* Avoid prop drilling

**📅 Day 13: useReducer Hook**

* Reducer function basics
* Dispatch actions
* Compare with useState

**📅 Day 14: Custom Hooks**

* Extract and reuse logic
* Naming conventions
* Return values from hooks

**📅 Day 15: Code Splitting & Error Boundaries**

* Lazy loading components
* React.lazy, Suspense
* ErrorBoundary component

🛠 Projects:

* Theme Switcher (useContext)
* Shopping Cart (useReducer)
* Performance Dashboard (useMemo)

**✅ Phase 4: State Management (Global) (Day 16 - Day 18)**

🎯 *Goal: Complex state sharing across app.*

**📅 Day 16: Context API Deep Dive**

* Structure of global provider
* When to use vs when not to use

**📅 Day 17: Redux Basics**

* Actions, Reducers, Store
* Redux Toolkit setup
* useSelector, useDispatch

**📅 Day 18: Integration & Patterns**

* Combine context or Redux with routing
* Debugging tools like Redux DevTools

🛠 Projects:

* E-Commerce Cart with Redux
* Task Manager with Context

**✅ Phase 5: React with APIs (Day 19 - Day 22)**

🎯 *Goal: Work with real-time data and API integration.*

**📅 Day 19: Fetch & Axios**

* Fetch basics
* Axios configuration

**📅 Day 20: API with useEffect**

* Loading state
* Error handling
* Async/await inside useEffect

**📅 Day 21: Pagination & Filtering**

* Client-side vs server-side pagination
* Filtering logic using search

**📅 Day 22: Integration Patterns**

* Common structure for API-based apps
* Services or helpers folder

🛠 Projects:

* GitHub Profile Viewer
* Movie App with OMDB API
* Search/Filter UI

**✅ Phase 6: Styling in React (Day 23 - Day 25)**

🎯 *Goal: Build beautiful, responsive UIs.*

**📅 Day 23: CSS Modules**

* Scoped styling
* Naming conventions

**📅 Day 24: Tailwind CSS**

* Setup and usage
* Responsive utilities

**📅 Day 25: Styled Components (optional)**

* Dynamic styling with props
* Theme setup

🛠 Tasks:

* Responsive Blog
* Figma to React Page

**✅ Phase 7: Fullstack React (Day 26 - Day 30)**

🎯 *Goal: Connect frontend to backend (Node.js / Firebase).*

**📅 Day 26: Backend Basics**

* Express server setup
* MongoDB/Mongoose basics

**📅 Day 27: Auth & Tokens**

* JWT tokens
* Login/Register flow

**📅 Day 28: Protected Routes**

* Save auth in localStorage/context
* Redirect logic

**📅 Day 29: Fullstack CRUD**

* Axios with backend
* Create, read, update, delete

**📅 Day 30: Project Wrap-up**

* Deployment (Netlify + Render)
* Env setup (.env)
* Folder structure best practices

🛠 Final Projects:

* MERN Blog
* Auth System
* Task Manager with Backend

🎓 **Cheatsheet Summary**:

* JSX = HTML + JS ➝ Use {} for logic
* useState = Track local state
* useEffect = Run code after render
* props = Pass data
* .map() = Render lists
* useContext = Global state
* useReducer = State + actions
* React Router = Multiple pages
* Tailwind = Utility-first styling
* Redux = Centralized state mgmt
* fetch/axios = Talk to APIs
* Node.js + Express = Backend
* MongoDB = Database
* JWT = Auth token for security

📦 **Tools**: React DevTools, Vite/CRA, Postman, VS Code, GitHub