

# React Learning Roadmap with Practical Experience

## 1. Prerequisites (Before React)

### Learn:

- HTML basics (tags, attributes, semantic HTML)
- CSS basics (selectors, box model, flexbox, grid)
- JavaScript fundamentals:
  - Variables, data types, operators
  - Functions, arrays, objects
  - DOM manipulation
  - `map`, `filter`, `reduce`
  - ES6+ features (arrow functions, destructuring, spread/rest, template literals)

### Practical:

- Build a static portfolio page with HTML/CSS/JS.
  - Create a simple "To-Do List" app in vanilla JavaScript.
-

## 2. React Basics

### Learn:

- What is React, SPA concept
- `create-react-app` or Vite setup
- JSX syntax
- Components (function components)
- Props
- State (`useState`)
- Event handling
- Conditional rendering
- Lists & keys

### Practical:

- Build a "Counter" app.
  - Create a "User Card" list from an array of user objects.
  - Build a "Toggle Light/Dark" button.
-

## 3. Styling in React

### Learn:

- CSS Modules
- Inline styles
- Styled-components / Tailwind CSS (optional)

### Practical:

- Create a responsive "Profile Page" with styled-components.
  - Rebuild the "To-Do List" with Tailwind CSS.
- 

## 4. React Hooks (Core)

### Learn:

- `useState`
- `useEffect`
- `useRef`
- `useReducer`
- `useContext`

### Practical:

- Build a "Countdown Timer" using `useEffect`.
- Create a "Theme Switcher" with `useContext`.
- Build a "Form with Validation" using `useReducer`.

---

## 5. Forms & Controlled Components

### Learn:

- Controlled vs uncontrolled components
- Handling form inputs
- Validation (manual & with libraries like `Formik` or `react-hook-form`)

### Practical:

- Build a "Sign-up Form" with live validation.
  - Create a "Search Filter" that updates a list as you type.
- 

## 6. Fetching Data

### Learn:

- Fetch API / `axios`
- Using `useEffect` for API calls
- Loading & error states

### Practical:

- Build a "Weather App" fetching data from OpenWeather API.
  - Create a "GitHub User Search" app.
-

## 7. Routing

### Learn:

- React Router (`react-router-dom`)
- `BrowserRouter`, `Routes`, `Route`
- Dynamic routes
- Navigation (`useNavigate`)

### Practical:

- Build a "Blog App" with home, post list, and single post pages.
  - Create a "Product Catalog" with dynamic product detail pages.
- 

## 8. State Management

### Learn:

- Context API
- Redux Toolkit
- Zustand or Jotai

### Practical:

- Rebuild the "To-Do App" with Context API.
  - Build a "Shopping Cart" with Redux Toolkit.
-

## 9. Advanced Hooks

### Learn:

- `useMemo`
- `useCallback`
- Custom hooks
- Performance optimization

### Practical:

- Build a "Debounced Search" input using a custom hook.
  - Create a "Window Size Tracker" hook.
- 

## 10. Working with External Libraries

### Learn:

- UI libraries (Material UI, Ant Design)
- Chart libraries (Recharts, Chart.js)
- Date pickers, modals, sliders

### Practical:

- Build a "Dashboard" with Material UI and charts.
  - Create a "Task Calendar" using a date picker.
-

## 11. Testing

### Learn:

- Jest basics
- React Testing Library

### Practical:

- Write tests for a Counter component.
  - Test API call rendering in a data table.
- 

## 12. Deployment

### Learn:

- Netlify / Vercel / GitHub Pages
- Environment variables

### Practical:

- Deploy the "Weather App" to Netlify with `.env` for API key.
- 

## 13. Extra Skills

### Learn:

- Error boundaries
- Code splitting & lazy loading
- React 18 concurrent features

- Server Components (Next.js)

**Practical:**

- Convert your project to Next.js for SSR/SSG.
  - Add lazy-loaded components to improve performance.
- 

## **Final Capstone Project Ideas**

- E-Commerce App (Products, Cart, Checkout)
- Social Media Dashboard
- Recipe Finder with Favorites
- Expense Tracker
- Chat Application (Socket.io)