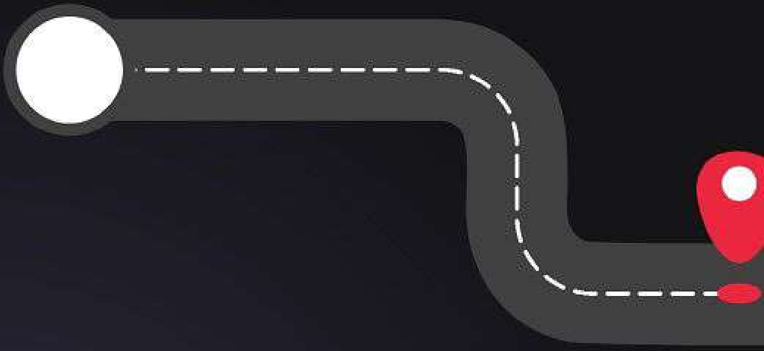
Education
@topdev_media

# React.js ROADMAP

Save it for later !



# 1. WEB FUNDAMENTALS

- **HTML**, **CSS**, and **Responsive** Design.
- **JavaScript Basics**: ES6+ features (arrow functions, destructuring, modules, etc.).
- **DOM** Manipulation.
- **Fetch** API and **Promises**.
- **Git/GitHub** for version control.

## 2. CORE REACT

- **React Basics:** Components, Props, and State.
- **JSX** and Component **Lifecycle**.
- **Event** Handling.
- **Conditional** Rendering.
- **Lists** and **Keys**.

# 3. ADVANCED REACT

- **React Hooks:** useState, useEffect, useContext.

- **Context** API for state management.

- React **Router** for navigation.

- **Optimizations**: React.memo, useMemo, useCallback.

- Error **Boundaries**.

## 4. STATE MANAGEMENT

- **Local State** with React.
- **Global State** with Context API.
- **Advanced**: Redux or Zustand.

# 5. PERFORMANCE OPTIMIZATION

- **Lazy Loading** with React.lazy and Suspense.
- Code **Splitting**.
- **Virtualization** for large lists (e.g., react-window).
- React **Profiler**.

## 6. TESTING

- Unit Testing with **Jest**.
- **Component Testing** with React Testing Library.
- End-to-End Testing with **Cypress**.

# 7. STYLING

- **CSS Modules** or Inline Styles.
- Utility Frameworks: **Tailwind CSS.**
- **Styled Components** or Emotion for CSS-in-JS.

## 8. TYPESCRIPT WITH REACT

- Type **annotations** for props and state.
- Advanced **types** (Generics, Utility Types).

## 9. ECOSYSTEM TOOLS

- **Next.js** for server-side rendering and routing.
- **React Query** for data fetching and caching.
- **Form Libraries:** React Hook Form or Formik.

## 10. REAL-WORLD PRACTICES

- Building **reusable** components.
- Folder **structure** and component **architecture**.
- Deployment with tools like **Vercel** or **Netlify**.