ChatGPT

React & React Native Roadmap (Beginner to Advanced)

Phase 1: Absolute Beginner

Prerequisites:

- Basic HTML, CSS, and JavaScript
- Understanding of ES6+ features (let/const, arrow functions, destructuring, template literals, spread/rest operators, promises)

React Basics:

- What is React and why use it?
- Creating a React app (with Vite or CRA)
- Components (Functional & Class-based)
- JSX syntax
- Props and State
- Event handling
- Conditional rendering
- Lists and keys
- Basic forms with controlled components

React Native Basics:

- Setting up React Native CLI or Expo
- Views, Text, Image, ScrollView, TextInput, Button
- Styling with StyleSheet
- Handling user input

Phase 2: Intermediate

Core Concepts:

- Lifting state up
- Component composition
- Reusable components
- `useState`, `useEffect` hooks
- React Router (for React web)
- Basic navigation in React Native (React Navigation)

APIs:

- Fetching data with `fetch` and `axios`
- Handling loading and error states

Forms & Validation:

- Controlled vs uncontrolled inputs
- Validation with simple logic

Local Storage:

- Web: `localStorage`, `sessionStorage`
- React Native: `AsyncStorage`

Phase 3: Advanced

Advanced Hooks:

- `useReducer`
- `useCallback`, `useMemo`, `useRef`
- Custom Hooks
- Debouncing/throttling with hooks

Performance Optimization:

- React.memo
- useMemo and useCallback to prevent re-renders
- Lazy loading with `React.lazy` and `Suspense`

State Management:

- Context API + useContext
- Redux Toolkit
- Alternatives: Zustand, Jotai, Recoil

Form Libraries:

- `react-hook-form`
- `formik`
- Validation with `yup` or `zod`

React Native Specific:

- Navigation: Stack, Tab, Drawer
- Push Notifications (FCM)
- Location and Maps

- Animations with `react-native-reanimated`
- Permissions handling
- Native modules and platform-specific code

Phase 4: Professional Developer Skills

TypeScript:

- Types, interfaces, type assertions
- Typing props, state, functions

Testing:

- Unit Testing with Jest
- Component Testing with React Testing Library
- End-to-End Testing (Cypress, Detox)

DevOps & Tooling:

- ESLint, Prettier setup
- Git branching strategy (feature, develop, release)
- CI/CD: GitHub Actions, Bitrise

Project Structure & Patterns:

- Atomic design methodology
- Feature-based folder structure
- Reusable hooks & components
- Code splitting

Final Phase: Real-World Projects

Project Ideas:

- E-commerce App (React or React Native + Stripe)
- Chat App (Firebase)
- Task Manager with Drag & Drop
- Food Delivery App
- Travel Booking App
- Admin Dashboard (Charts, tables, filters)

🛠 Bonus Tools & Libraries:

- React Query (data fetching & caching)
- Framer Motion (animations)

- Tailwind CSS (web styling)
- Native Base / UI Kitten (React Native styling)

Tips to Succeed:

- Build projects, not just tutorials
- Join communities (Reddit, Discord, Twitter)
- Read docs regularly
- Push code to GitHub daily
- Pair program or contribute to open source

> "Consistency beats intensity — a little code every day goes a long way!"

Start building. Start shipping. Become a React Pro!