

React Guideline

1. Prerequisites & environment

- Javascript (es6+) (arrow functions, destructuring, spread/rest operators, modules, async/await, promises)
 - Node.js, npm / yarn / pnpm
 - Html & css (basics, responsive design, and layout)
 - Build tools (vite, create react app, next.js basics)
-

2. Core react fundamentals

- What is react & why use it
 - Spa (single page applications): concept & benefits
 - Jsx syntax & rules
 - Components: functional vs class (functional + hooks preferred)
 - Props & children, proptypes, destructuring props
 - State: this.state / setstate (class), usestate (functional)
 - Conditional rendering
 - Rendering lists & keys
 - Handling events
 - React.fragment / <> shorthand
 - Rendering and reconciliation: how react updates the dom
 - Virtual dom concept: basic understanding
-

3. Modern react with hooks

- Basic hooks: usestate, useeffect and dependency array
- Other hooks: useref, usememo, usecallback
- Custom hooks: creation & usage
- Rules of hooks
- Complex state: usereducer
- Context api: usecontext

- Advanced hooks: `useLayoutEffect`, `useId` (react 18)
-

4. Component architecture & patterns

- Smart (container) vs dumb (presentational) components
 - Component composition vs inheritance
 - Controlled vs uncontrolled components
 - Lifting state up
 - Props drilling & alternatives (context, redux)
 - Reusable component design
 - Error boundaries
 - Code organization & folder structure best practices
-

5. Routing & navigation

- React router v6+ (setup & usage)
 - Route, link, NavLink, navigate
 - Nested routes
 - Dynamic routes & url parameters
 - Query strings
 - Protected routes & authentication flows
 - Redirects & navigation guards
-

6. Styling in react

- CSS basics, sass
 - Inline styles, CSS modules, styled components, emotion
 - Utility-first frameworks (tailwind CSS)
 - Material UI basics
 - Dynamic class names & responsive design
-

7. Forms & validation

- Controlled vs uncontrolled forms
- Handling multiple inputs & form state

- React hook form, formik basics
 - Schema validation (zod / yup)
-

8. State management

- Local component state
 - Context api for global state
 - Redux / redux toolkit
 - Middleware in redux
 - Lightweight alternatives (zustand, jotai)
-

9. Data fetching & apis

- Fetch api, axios
 - Useeffect with async calls
 - Abort controllers & cleanup
 - Handling loading and error states
 - React query (tanstack query)
 - GraphQL & websocket
-

10. Performance optimization

- React.memo & purecomponent
 - Usememo, usecallback deep dive
 - Lazy loading & suspense
 - Code splitting & dynamic imports
 - Virtualization (react-window, react-virtualized)
 - Profiling & performance debugging
 - Bundle optimization & devtools profiler
 - Avoiding unnecessary re-renders
-

11. Testing in react

- Unit testing (jest, vitest, react testing library)
- Integration & e2e testing (cypress, playwright)

- Mocking api calls
-

12. Advanced topics & ecosystem

- Typescript in react
 - Server-side rendering (ssr) & static site generation (next.js)
 - React 18 concurrent features: suspense, useTransition
 - Accessibility (aria roles, keyboard navigation)
 - Security best practices
 - Internationalization (i18n): react-i18next
 - Animations: framer motion, react spring
 - Design systems / storybook introduction
 - React native basics for mobile
-

13. Project architecture & best practices

- Folder / file structure (feature-based vs domain-based)
 - Component & code organization
 - Code splitting & lazy loading strategies
 - Error boundaries & debugging techniques
 - Deployment & CI/CD basics
 - Monitoring & performance in production
-

14. Soft & workflow skills

- Git & github workflows (branching, prs, code reviews)
- Debugging react apps (browser devtools, breakpoints)
- Writing clean, maintainable code
- Collaboration tools (storybook, figma handoff)