

## **1. What is React?**

React is a JavaScript library developed by Facebook for building user interfaces, especially for single-page applications.

## **2. Explain the concept of Virtual DOM. How does it work in React?**

The Virtual DOM is a lightweight, in-memory representation of the real DOM. React updates the Virtual DOM first, then compares it with the previous version (diffing), and efficiently updates only the changed parts in the real DOM.

## **3. What is JSX? Why is it used in React?**

JSX (JavaScript XML) is a syntax extension that allows writing HTML-like code inside JavaScript. It's used in React to describe UI components in a readable and declarative way.

## **4. What are components in React? Differentiate between functional and class components.**

Components are reusable pieces of UI in React.

- **Functional Components:** Are stateless or use hooks; simple JavaScript functions.
- **Class Components:** Use ES6 classes; support lifecycle methods and state with this.state.

## **5. What is the difference between controlled and uncontrolled components?**

- **Controlled Components:** Form elements whose values are controlled via React state.
- **Uncontrolled Components:** Use refs to access form values directly from the DOM.

## **6. Explain the lifecycle methods of React class components.**

Key lifecycle methods:

- componentDidMount(): Runs after component mounts.
- componentDidUpdate(): Runs after updates.
- componentWillUnmount(): Runs before component unmounts.

## **7. What is the purpose of the useState hook? Provide an example.**

useState allows you to add state to functional components.

**Example:**

```
const [count, setCount] = useState(0);
```

## **8. How does the useEffect hook work? Give a real-world scenario where it is used.**

useEffect handles side effects like fetching data or subscribing to events.

**Example:** Fetching data when a component loads:

## **9. Explain the concept of props in React. How are they different from state?**

Props are inputs passed from parent to child components. Unlike state, props are read-only and cannot be changed by the receiving component.

## **10. What is prop drilling and how can it be avoided?**

Prop drilling is passing data through multiple nested components. It can be avoided using **Context API** or **state management libraries** like Redux.

## **11. What are React Hooks? List some commonly used hooks.**

Hooks are functions that let you use React features in functional components. Common hooks: useState, useEffect, useContext, useRef, useMemo, useReducer.

## **12. How do you lift state up in React?**

Lifting state means moving state to a common parent component so it can be shared among child components via props.

## **13. What is context in React and when should you use the Context API?**

Context provides a way to share values like theme or auth data across components without prop drilling. Use it for global or app-wide state.

## **14. How does React handle forms?**

React uses **controlled components** where form values are managed via state, enabling real-time validation and input handling.

## **15. How do you perform conditional rendering in React?**

You can use:

- if statements
- Ternary operators: condition ? <A /> : <B />
- Logical AND (&&): condition && <Component />

## **16. What are keys in React and why are they important in lists?**

Keys are unique identifiers for list items, helping React track and optimize changes during rendering.

## **17. Explain the reconciliation process in React.**

Reconciliation is React's process of comparing the current Virtual DOM with the previous one and updating only the changed elements in the real DOM.

## **18. What is Redux?**

Redux is a state management library that maintains application state in a single store.

## **19. how does Redux work with React?**

It works with React via actions, reducers, and the connect or useSelector/useDispatch hooks.

## **20. How do you connect a React application to Redux?**

Use Provider to wrap the app and connect HOC or useSelector/useDispatch hooks to access and manipulate the Redux store..

## **21.What is a component?**

A self-contained, reusable piece of UI defined in React.

## **22. What is useCallback?**

Memoizes functions to prevent unnecessary re-creations.

## **23. What is useMemo?**

Memoizes expensive calculations to avoid recomputation.

## **24. What is useRef?**

Provides a reference to a DOM node or stores mutable values.

**25.What is useReducer?**

A Hook for managing complex state using a reducer function.

**26. What is code splitting?**

Splitting app code into chunks and loading parts as needed using React.lazy.

**27. What is React.memo?**

A HOC that memoizes functional components to avoid unnecessary renders.

**28. What is Error Boundary?**

A class component using ErrorBoundary to catch errors in its child component tree.

**29. What is server-side rendering (SSR)?**

Rendering components on a server to send HTML to the client, improving performance & SEO.

**30.what is static site generation (SSG)**

**static site generation (SSG)** a web development technique where a website's HTML pages are pre-built during the build process, rather than being dynamically generated on demand for each user request.

**31.What's the difference between global state and local state?**

Local affects only one component; global is shared across the app.

**32.When would you use React.lazy and Suspense?**

To lazily load components and show fallback while loading.

**33. What are props?**

Read-only inputs passed from parent to child components.

**34. What is state?**

Internal data that determines a component's behavior and rendering.

**35.Props vs. State?**

Props are external and immutable; state is internal and mutable.

**36. What is React Router?**

A routing library for managing navigation in React apps.

### **37. what is custom hooks in react?**

Custom Hooks in React are JavaScript functions designed to extract and reuse stateful logic from functional components.

### **38. what is navigate in react?**

useNavigate Hook: This hook provides a function that allows programmatic navigation.

### **39. what are react main features?**

Its main features include component-based architecture, Virtual DOM, JSX, unidirectional data flow, and support for hooks.

### **40. What is actions in Redux?**

**Actions:** Describe events with a type and optional payload.

### **41.What is Reducers in Redux?**

**Reducers** Pure functions that update state based on actions.

### **42.What is Store in Redux?**

**Store:** Holds the entire app's state and allows state access and dispatching actions.

### **43. What is <Route>?**

Defines a path and the component to render.

### **44. What is <Link>?**

Provides client-side navigation without reloading the page.

### **45. what is HOC in react?**

A HOC is essentially a function that accepts a React component as an argument and returns a new,

### **46.What is StrictMode?**

StrictMode is a tool for highlighting potential problems in an application.  
Like Fragment

## **47. What is Portals?**

React Portals provide a way to render children components into a DOM node that exists outside the DOM hierarchy of the parent component

## **48. What is Fragments?**

Fragments provide a way to group a list of children elements without adding extra nodes

## **49. What is Optimizing Performance In react?**

Optimizing performance in React applications involves several techniques aimed at reducing unnecessary re-renders, improving initial load times, and enhancing overall responsiveness.

## **50.What are Key Optimization Techniques?**

Memoization

Code Splitting and Lazy Loading

Proper Key Usage in Lists

Server-Side Rendering (SSR) and Pre-rendering (PR)

