React JS Interview Questions

Basic Level

1. What is React?

2. What are the features of React?

3. What is JSX?

4. What is the difference between a class component and a functional component?

5. What is a component in React?

6. What is the purpose of the render() method?

7. What are props in React?

8. What is state in React?

9. What is the Virtual DOM and how does it work?

10. What is the difference between state and props?

Intermediate Level

1. Explain React Hooks.

2. How does useState work?

3. How does useEffect work?

4. What is prop drilling and how can it be avoided?

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

6. What is conditional rendering in React?

7. What are keys in React and why are they important?

8. How does form handling work in React?

9. How does React handle events?

10. Explain component lifecycle methods in class components.

Advanced Level

React JS Interview Questions

1. What is the Context API and when would you use it?

2. What is useContext?

3. What is useRef and when do you use it?

4. Difference between useMemo and useCallback.

5. What are Higher-Order Components (HOC)?

6. What is React.memo and how does it help in performance?

7. How do you optimize performance in React apps?

8. Explain lazy loading and code splitting in React.

9. What are React Fragments and why are they used?

10. What is Reconciliation in React?

Routing and State Management

1. How does routing work in React?

2. Explain the purpose of BrowserRouter, Route, and Link.

3. What is useNavigate, useParams, and useLocation in React Router?

4. How does Redux work?

5. What is the difference between Redux and Context API?

Testing React Apps

1. What tools are used to test React applications?

2. How do you test components using React Testing Library?

3. How do you test events like onClick?

4. What is snapshot testing?

5. What are mock functions in Jest?

React JS Interview Questions

Best Practices

1. How do you manage reusable components?

2. What are the best practices for folder structure in React projects?

3. How do you secure a React application?

4. How do you handle API requests in React?

5. How do you implement error boundaries in React?