

React Evaluation Questions

Section A – Basic Level

1. What is JSX, and why do we use it in React?
2. Differentiate between functional components and class components.
3. What is the difference between props and state in React?
4. Why must React components return a single parent element?
5. In the below snippet, what mistake can you spot?

```
function Welcome() {  
  return (  
    <h1>Hello</h1>  
    <p>World</p>  
  );  
}
```

5. Explain why React elements are immutable.
 6. What is the significance of the key attribute in lists?
 7. How does React handle events differently than plain HTML/JS?
 8. What is the role of ReactDOM.render() (before React 18) or createRoot() (after React 18)?
 9. What is the Virtual DOM, and how does React use it?
-

Section B – Intermediate Level

1. What are React Hooks, and why were they introduced?
2. Differentiate between useState and useReducer. In what scenario would you prefer useReducer?
3. Explain useEffect. What problems can arise if dependencies are not set correctly?
4. What is the difference between useMemo and useCallback?
5. Why must Hooks always be called at the top level of a function component?
6. In the snippet below, identify the issue:

```
function Counter() {  
  if (true) {  
    const [count, setCount] = useState(0);  
  }  
}
```

```
    return <p>{count}</p>;  
  }  
}
```

6. Explain the Context API. How does it solve the problem of prop drilling?
 7. What are controlled vs uncontrolled components in React forms?
 8. Differentiate between conditional rendering using `&&`, ternary operators, and early returns.
 9. Why do React keys need to be stable, predictable, and unique?
-

Section C – Miscellaneous (Advanced)

1. What are Error Boundaries in React? Why don't they catch errors in event handlers?
 2. How does React's reconciliation algorithm (diffing algorithm) work in simple terms?
 3. Explain what Suspense is in React. Give an example scenario where it is useful.
 4. What are custom hooks, and when would you create one?
 5. What are higher-order components (HOCs)? How are they different from custom hooks?
 6. What is the difference between `React.lazy` and dynamic imports?
 7. How does React 18's concurrent rendering improve user experience?
 8. What is server-side rendering (SSR), and how does it differ from client-side rendering (CSR)?
 9. What are hydration and rehydration in React apps?
 10. Explain React's `StrictMode`. What does it help developers identify?
-

Miscellaneous Questions (Extra)

1. What is a React Fragment and why would you use it instead of a `<div>`?
2. Can Fragments have keys, and in what scenario is this useful?
3. What is `React.memo` and how does it help with performance optimization?
4. When should you use `useMemo` and `useCallback` hooks in React?
5. What is React's reconciliation process and how does it affect performance?
6. How does code-splitting or lazy loading improve React app performance?
7. What is the difference between local state and global state in React?
8. When would you use Context API instead of Redux for state management?
9. What are the main building blocks of Redux (store, reducers, actions)?
10. What is a common drawback of using Context API for deep state sharing?