JavaScript Fundamentals:

- What are closures in JavaScript, and how are they useful?
- Explain the differences between var, let, and const in JavaScript.
- How does prototypal inheritance work in JavaScript?
- What is event bubbling and event delegation in JavaScript?

React Basics:

- What are the key features of React?
- Explain the concept of Virtual DOM in React and its significance.
- Differentiate between state and props in React components.
- How does JSX differ from regular JavaScript?

React Component Lifecycle:

- Describe the lifecycle methods of a React component and their purposes.
- What are the differences between componentDidMount() and componentWillMount()?
- How do you handle side effects in React components?

State Management in React:

- What is the state in React, and how is it different from props?
- Explain the significance of lifting state up in React applications.
- How do you manage application-level state in React?

React Hooks:

- What are React Hooks, and why were they introduced?
- Discuss the differences between useState and useEffect hooks.
- How do you create custom hooks in React?

Handling Events in React:

- Explain event handling in React.
- Discuss the differences between using inline event handlers and event listeners.
- How do you prevent default behavior in React event handlers?

Component Composition and Reusability:

- What is component composition in React, and why is it important?
- Discuss strategies for creating reusable React components.

- How do you pass data between parent and child components in React?

Error Handling and Debugging in React:

- How do you handle errors in React applications?
- Explain the usage of Error Boundaries in React.
- What tools and techniques do you use for debugging React applications?

Performance Optimization in React:

- Discuss techniques for optimizing performance in React applications.
- What are the potential performance bottlenecks in React applications, and how do you address them?
- Explain the significance of using memoization and pure components in React.

Working with External APIs in React:

- How do you fetch data from external APIs in React?
- Discuss strategies for handling asynchronous data fetching in React applications.
- How do you manage API requests and responses in React components?

JavaScript Promises and Async/Await:

- Explain the purpose of Promises in JavaScript and how they help with asynchronous programming.
- What are the differences between Promises and callbacks?
- How do you handle errors in Promises and async/await functions?

React Router:

- What is React Router, and how does it help with client-side routing in React applications?
- Discuss the differences between BrowserRouter and HashRouter in React Router.
- How do you implement nested routing in React applications?

Redux and State Management:

- What is Redux, and why would you use it in a React application?
- Describe the core principles of Redux, including actions, reducers, and the store.
- How do you connect Redux with React components?

React Hooks Advanced Concepts:

- Discuss the usage of useRef, useMemo, and useContext hooks in React.
- Explain the purpose of useCallback hook and when it's useful.
- How do you handle side effects with the useEffect hook?

Testing in React:

- What tools and libraries do you use for testing React applications?
- Discuss the differences between unit testing, integration testing, and end-to-end testing in React.
- How do you mock external dependencies and simulate user interactions in React tests?

React Performance Optimization Techniques:

- Discuss the significance of code splitting and lazy loading in React performance optimization.
- How do you optimize rendering performance in React components?
- Explain the importance of using keys in React lists and the potential performance implications.

React Context API:

- What is the React Context API, and when would you use it?
- Discuss the advantages and limitations of using React Context for state management.
- How do you create and consume context providers and consumers in React applications?

Security Considerations in React Applications:

- What are some common security vulnerabilities in React applications, and how do you mitigate them?
- Discuss best practices for handling user authentication and authorization in React.
- How do you prevent Cross-Site Scripting (XSS) attacks in React applications?

Hooks :

- useState
- useEffect
- useContext
- useReduce
- useMemo
- useCallback
- useRef

Higher Order Components :

- What are higher order components?
- Why do we use higher order components?
- When do we use higher order components?
- How do we use higher order components?

Life Cycle method of component

- **Phese**
- Mounting
- Unmounting
- Updating

State Management:

- State and Props
- Props Drilling
- React Context

Redux:

- What is Redux?
- Why do we use Redux?
- When do we use Redux?
- How do we use Redux?

Custom Hooks:

- When do we use Custom Hooks?
- **Code**

Lazy Loading:

Virtual DOM:

- Reconciliation
- React Fiber
- Diff Algo
- Render

SSR (Server Side Rendering) vs CSR (Client Side Rendering):

- SEO
- Performance

Routing:

- React Router
- Protected Route
- Dynamic Routing

ASYNC Task:

- API Calling
- Event Handling
- Promises
- setTimeout
- setInterval