**Explain the need for Unit Testing in React**  
Unit testing in React is essential to ensure that individual components and functions work as expected. It helps detect bugs early in the development process, improves code quality, and allows for easier refactoring with confidence. By testing components in isolation, developers can verify that each unit performs its logic correctly, handles props and state properly, and responds to user interactions as intended. Unit tests also serve as documentation for the component's expected behavior and promote long-term maintainability of the codebase.

**Working with Jest and Enzyme in React**  
Jest is a popular JavaScript testing framework used with React for writing and running tests. It offers features like test assertions, mocking, and snapshot testing. Enzyme is a testing utility developed by Airbnb that works alongside Jest to allow traversal, manipulation, and simulation of component behavior in React. Using Enzyme's shallow, mount, or render methods, developers can test components at various levels of depth. Typically, a test is written using Jest’s test or it function, combined with Enzyme’s methods to verify the output or behavior of a React component under specific conditions.

**List the types of Router Components**  
React Router provides different types of router components to manage navigation in React applications:

1. **BrowserRouter** – Uses the HTML5 history API to create clean URLs without the hash (#), suitable for modern web applications.
2. **HashRouter** – Uses the hash portion of the URL (#) to manage routing, often used in static file servers that do not support dynamic routes.
3. **MemoryRouter** – Keeps the history of the URL in memory (does not modify the browser URL), mainly used for testing or non-browser environments.
4. **NativeRouter** – Used for routing in React Native applications.  
   Each router provides context and enables the use of routing components like Route, Link, and Switch.