React Router FAQ

Question 1: What is React Router? How does it handle routing in single-page applications?

React Router is a standard library for routing in React applications. It enables navigation between different views or pages without requiring a full-page reload. This makes React applications feel faster and more responsive.

React Router handles routing in single-page applications (SPAs) by using:
1. Client-side Routing: It updates the URL without reloading the page.
2. Dynamic Component Rendering: It renders components based on the current URL path.
3. Route Matching: It maps specific URL paths to corresponding React components.
4. History API: It manages browser history, enabling forward and backward navigation.
5. Nested Routing: It allows hierarchical navigation structures within an application.

Question 2: Explain the difference between BrowserRouter, Route, Link, and Switch components in React Router.

1. BrowserRouter:
   - It is the main component that enables routing in a React application.
   - Uses the HTML5 history API to keep the UI in sync with the URL.

2. Route:
   - Defines the mapping between a specific URL path and a React component.
   - Example: <Route path='/about' element={<About />} /> renders the About component when the URL matches /about.

3. Link:
   - It is used to create navigation links without causing a full-page reload.
   - Example: <Link to='/home'>Home</Link> navigates to the /home route.

4. Switch (Replaced by Routes in React Router v6):
   - It ensures that only the first matching route is rendered.
   - In newer versions, Routes is used instead of Switch for better route handling.