# Routing in React (React Router)

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

**ANS:-** React Router is a standard routing library for React.js that enables navigation between different views or components in a Single Page Application (SPA). In traditional multi-page websites, each new page loads from the server. However, in SPAs, the page doesn't reload completely; instead, JavaScript dynamically updates the content. React Router manages this by enabling client-side routing, which means routing is handled within the browser without refreshing the page.

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

**ANS:-**

**BrowserRouter**

This is a top-level router component that uses the HTML5 history API (pushState, replaceState, and the popstate event) to keep your UI in sync with the URL. It enables navigation without full page reloads and is the most common router for web applications.

**Route**

This component defines a specific route within your application. It takes a path prop, which is a URL pattern, and typically an element (or component in older versions) prop, which specifies the React component to render when the URL matches the given path.

**Link**

This component creates navigation links within your application. It functions similarly to an HTML <a> tag but leverages React Router's functionality to navigate between different routes without triggering a full page reload. It takes a to prop, which specifies the destination path.

**Switch**

This component ensures that only one Route is rendered at a time among its children. It iterates through its child Route components and renders the first one whose path matches the current URL. This is crucial for preventing multiple route

components from rendering simultaneously when multiple paths might partially match the current URL. In React Router v6 and later, the Routes component replaces Switch and serves the same purpose of exclusive route rendering.