**Module-16) React -Advance React- Styling , Routing**

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

Ans - **What is React Router?**

**React Router** is a **popular library** used in **React** applications to manage and handle **routing**. It enables navigation between different **views** or **components** without requiring a **page reload**, making it perfect for **Single-Page Applications (SPAs)**.

In a SPA, instead of loading a **new HTML page** from the server, **React Router** dynamically renders components based on the **URL**, improving both performance and user experience.

**How Does React Router Work in SPAs?**

React Router uses **Client-Side Routing** instead of traditional **Server-Side Routing**. Here’s how it works:

1. **URL Management:**
   * React Router monitors the **URL** in the browser and displays the corresponding **React component**.
2. **No Full Page Reload:**
   * Changes in the route do not reload the entire page, making navigation **faster**.
3. **Component Mapping:**
   * Each route is mapped to a **specific component** that renders when the **URL** matches.
4. **History API:**
   * React Router uses the **Browser’s History API** to update the URL and track navigation (e.g., pushState and replaceState).
5. **Dynamic Routing:**
   * You can pass **parameters** (like /user/:id) to render specific content dynamically.

**Basic Setup of React Router**

**Step 1: Install React Router**

In a React project, run this command:

bash

CopyEdit

npm install react-router-dom

**Step 2: Create Basic Routes**

Here is a simple **React Router** setup:

javascript

CopyEdit

// App.js

import React from 'react';

import { BrowserRouter as Router, Route, Routes, Link } from 'react-router-dom';

const Home = () => <h1>Home Page</h1>;

const About = () => <h1>About Page</h1>;

const NotFound = () => <h1>404 - Not Found</h1>;

const App = () => {

return (

<Router>

<nav>

<Link to="/">Home</Link> | <Link to="/about">About</Link>

</nav>

<Routes>

<Route path="/" element={<Home />} />

<Route path="/about" element={<About />} />

<Route path="\*" element={<NotFound />} />

</Routes>

</Router>

);

};

export default App;

**Explanation of the Code:**

1. **BrowserRouter (Router)**
   * Wraps your app and enables **client-side routing**.
2. **Route**
   * Maps a **URL path** to a **component**:
     + / → Home Page
     + /about → About Page
3. **Routes**
   * Renders the **first matching route** (introduced in **React Router v6**).
4. **Link**
   * Provides **in-app navigation** without reloading the page.
5. **404 Page (Not Found)**
   * The path="\*" route catches all **unknown URLs** and displays a **404 error**.

**Key Features of React Router:**

1. **Dynamic Routing:**
   * Pass **parameters** like /user/:id.
2. **Nested Routes:**
   * Create **routes inside other routes**.
3. **Redirects and Navigation:**
   * Use **useNavigate()** for programmatic navigation.
4. **Protected Routes:**
   * Implement **authentication** and restrict access.
5. **Lazy Loading:**
   * Optimize performance using **React.lazy()** with routes.

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

Ans - Table

| **Component** | **Purpose** | **Key Feature** | **Version Support** |
| --- | --- | --- | --- |
| **BrowserRouter** | Enables client-side routing | Uses the **HTML5 History API** | v5 & v6 |
| **Route** | Matches URL to a component | Supports **dynamic and nested routes** | v5 & v6 |
| **Link** | Navigates between routes | No **page reload**, supports dynamic paths | v5 & v6 |
| **Switch** | Renders the **first matching** route | **Deprecated** – Replaced by Routes | v5 (not in v6) |