**React Router – Objectives**

**1. Explain the Need and Benefits of React Router**

**Need:**

In a React application, routing is not built-in. By default, React renders only a single component. To enable navigation between multiple views or components without reloading the entire page (as in Single Page Applications), React Router is required.

**Benefits:**

* Enables Single Page Application (SPA) behavior.
* Allows navigation between different components via URLs.
* Provides dynamic routing using declarative syntax.
* Supports nested routes and code splitting.
* Maintains application state without full page reloads.
* Enables parameterized routes and route protection.

**2. Identify the Components in React Router**

React Router offers several key components used for routing:

| **Component** | **Purpose** |
| --- | --- |
| BrowserRouter | Wraps the application; uses HTML5 history API to sync UI with the URL. |
| Routes | Container for route definitions. |
| Route | Maps a path to a specific React component. |
| Link | Provides navigation between routes without page reload (like <a>). |
| NavLink | Similar to Link but adds styling for active links. |
| useParams | Hook to access route parameters (e.g., :id). |
| useNavigate | Hook for programmatic navigation. |

**3. List the Types of Router Components**

| **Router Type** | **Description** |
| --- | --- |
| BrowserRouter | Uses the browser history API for clean URLs (recommended for web apps). |
| HashRouter | Uses the hash part of the URL (e.g., #/home), useful for static hosting. |
| MemoryRouter | Keeps history in memory (not in the address bar); useful for testing. |
| NativeRouter | Used in React Native mobile applications. |
| StaticRouter | Used in server-side rendering scenarios (e.g., with frameworks like Next.js). |

**4. Parameter Passing via URL**

React Router allows parameters to be passed through the URL, enabling dynamic rendering based on the route.

**a) Define a Route with Parameter:**

jsx

CopyEdit

<Route path="/trainer/:id" element={<TrainerDetails />} />

**b) Navigate to the Route with a Specific Parameter:**

jsx

CopyEdit

<Link to="/trainer/T101">Ravi Kumar</Link>

**c) Access the Parameter inside the Component:**

jsx

CopyEdit

import { useParams } from "react-router-dom";

function TrainerDetails() {

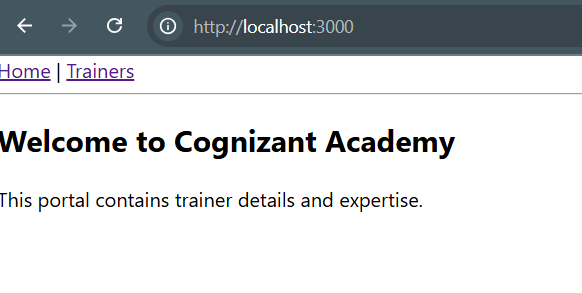
const { id } = useParams();

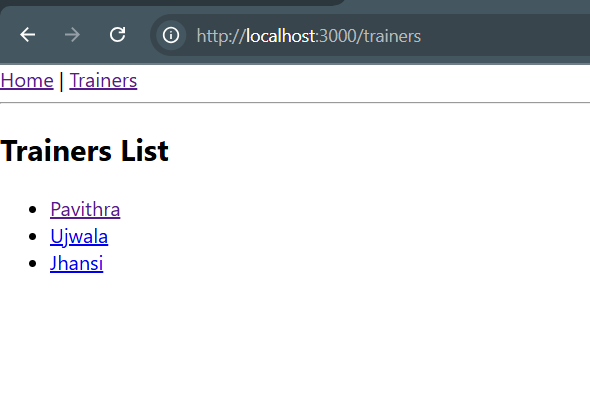
return <h2>Trainer ID is {id}</h2>;

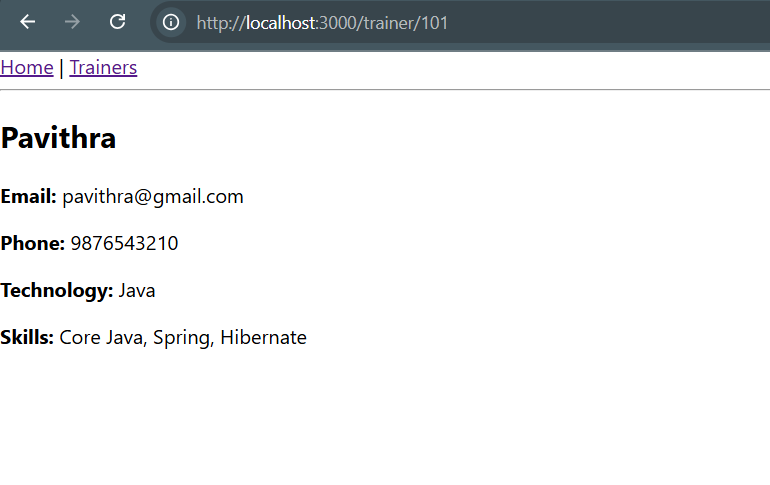
}

This mechanism helps build dynamic pages, such as user profiles, product details, etc., based on the URL.

**Output:**

****

****

****