**6.ReactJs HOL**

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

**Need:**

In a traditional website, clicking a link loads a new HTML page from the server. In a Single Page Application (SPA) like React, we need routing to navigate between views/components without reloading the page.

**Benefits of React Router:**

* **Enables SPA behavior** – routes without full page reload.
* **Dynamic URL handling** – URL reflects app state (e.g., /user/123).
* **Component-based navigation** – easily map components to routes.
* **Browser history integration** – allows back/forward navigation.
* **Code-splitting** – load components lazily based on route.

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

|  |  |
| --- | --- |
| BrowserRouter | * Wraps the entire app to enable routing using the browser's URL |

|  |  |
| --- | --- |
| Routes | * Container for all defined Route components |

|  |  |
| --- | --- |
| Route | * Defines a path and the component to render |

|  |  |
| --- | --- |
| Link | * Navigates between routes (like anchor tag <a>) |

|  |  |
| --- | --- |
| UseParams | * Hook to read dynamic parameters from the URL |

|  |  |
| --- | --- |
| UseNavigate - | Hook to programmatically navigate to a route |

|  |  |
| --- | --- |
| Navigate | * Component for redirecting |

|  |  |
| --- | --- |
| Outlet | * Used in nested routing to render child routes |
|  |  |

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

| **Type of Router** | **Description** |
| --- | --- |
| **BrowserRouter** | Uses HTML5 History API (clean URLs like /home) – most common |
| **HashRouter** | Uses URL hash (e.g., /#/home) – good for static file servers |
| **MemoryRouter** | Stores history in memory (not in the URL bar) – used in testing or non-browser environments |
| **StaticRouter** | Used for server-side rendering (SSR), like in Next.js apps |

**4.** **Parameter Passing via URL**

**Step 1: Define a Route with a Param**

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

**Step 2: Use useParams to Read It**

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

const TrainerDetail = () => {

const { id } = useParams(); // gets 'id' from URL

// Use 'id' to fetch or display trainer data

}

**Step 3: Pass the ID in Link**

<Link to={`/trainer/${trainer.TrainerId}`}>{trainer.Name}</Link>

This enables dynamic rendering of components based on the route parameter.

**Cognizant Academy teams want to maintain a list of trainers along with their expertise in a SPA using React as the technology. You are assigned the task of creating this React app.**

**React App : trainersapp**

**Trainer.js**

class Trainer {

constructor(TrainerId, Name, Email, Phone, Technology, Skills) {

this.TrainerId = TrainerId;

this.Name = Name;

this.Email = Email;

this.Phone = Phone;

this.Technology = Technology;

this.Skills = Skills;

}

}

export default Trainer;

**TrainersMock.js**

import Trainer from './Trainer';

const trainers = [

new Trainer(1, 'Poorani', 'poorani@mail.com', '1234567890', 'React', ['Hooks', 'Routing']),

new Trainer(2, 'John', 'john@mail.com', '9876543210', 'Java', ['Spring', 'Hibernate']),

new Trainer(3, 'Aisha', 'aisha@mail.com', '1122334455', 'Python', ['Flask', 'Django']),

];

export default trainers;

**TrainerList.js**

import React from 'react';

import { Link } from 'react-router-dom';

const TrainersList = ({ trainers }) => {

return (

<div>

<h2>Trainer List</h2>

<ul>

{trainers.map(trainer => (

<li key={trainer.TrainerId}>

<Link to={`/trainer/${trainer.TrainerId}`}>{trainer.Name}</Link>

</li>

))}

</ul>

</div>

);

};

export default TrainersList;

**Home.js**

import React from 'react';

const Home = () => {

return (

<div>

<h2>Welcome to the Trainers App</h2>

<p>This app displays a list of trainers and their details.</p>

</div>

);

};

export default Home;

**App.js**

import React from 'react';

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

import Home from './Home';

import TrainersList from './TrainerList';

import TrainerDetail from './TrainerDetail';

import trainers from './TrainersMock';

function App() {

return (

<BrowserRouter>

<div>

<nav>

<Link to="/">Home</Link> | <Link to="/trainers">Trainers</Link>

</nav>

<Routes>

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

<Route path="/trainers" element={<TrainersList trainers={trainers} />} />

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

</Routes>

</div>

</BrowserRouter>

);

}

export default App;

**TrainerDetail.js**

import React from 'react';

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

import trainers from './TrainersMock';

const TrainerDetail = () => {

const { id } = useParams();

const trainer = trainers.find(t => t.TrainerId === parseInt(id));

if (!trainer) return <p>Trainer not found.</p>;

return (

<div>

<h2>{trainer.Name}'s Details</h2>

<p><strong>Email:</strong> {trainer.Email}</p>

<p><strong>Phone:</strong> {trainer.Phone}</p>

<p><strong>Technology:</strong> {trainer.Technology}</p>

<p><strong>Skills:</strong> {trainer.Skills.join(', ')}</p>

</div>

);

};

export default TrainerDetail;

**Final Output:**









