ReactJS Hands-On Lab Report – Trainers SPA using React Router

# Objectives

- Explain the need and benefits of React Router  
- Identify the Components in React Router  
- List the types of Router Components  
- Parameter passing via URL

# In this hands-on lab, you will learn how to:

- Implement a Simple Navigation Menu  
- Add Basic Routes (install, configure)  
- Use Routes in React Applications

# Prerequisites

- Node.js  
- NPM (Node Package Manager)  
- Visual Studio Code

# Step-by-Step Instructions

1. 1. Create a new React app using: npx create-react-app trainersapp
2. 2. Navigate to the project folder: cd trainersapp
3. 3. Open the project in Visual Studio Code: code .
4. 4. Create a new file 'Trainer.js' in the 'src' folder and define a Trainer class.
5. 5. Create a 'TrainersMock.js' file in 'src' folder to define mock trainer data.
6. 6. Install React Router DOM using: npm install react-router-dom
7. 7. Create 'TrainerList.js' component to list trainer names as clickable links.
8. 8. Create 'Home.js' component for the home page message.
9. 9. Create 'TrainerDetail.js' component to show trainer details using useParams().
10. 10. Modify 'App.js' to include navigation and define routes using BrowserRouter.
11. 11. Run the application using: npm start and open http://localhost:3000

## Code: 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;

## Code: TrainersMock.js

import Trainer from './Trainer';  
  
const trainers = [  
 new Trainer(1, "Aditi Sharma", "aditi@cognizant.com", "9876543210", "React", ["JSX", "Hooks", "Routing"]),  
 new Trainer(2, "Rohan Mehta", "rohan@cognizant.com", "9876543211", "Java", ["Spring", "JPA", "REST"]),  
 new Trainer(3, "Priya Verma", "priya@cognizant.com", "9876543212", "Python", ["Flask", "Django", "Pandas"]),  
];  
export default trainers;

## Code: TrainerList.js

import React from 'react';  
import { Link } from 'react-router-dom';  
  
function TrainerList({ trainers }) {  
 return (  
 <div>  
 <h2>Trainers List</h2>  
 <ul>  
 {trainers.map(trainer => (  
 <li key={trainer.trainerId}>  
 <Link to={`/trainers/${trainer.trainerId}`}>{trainer.name}</Link>  
 </li>  
 ))}  
 </ul>  
 </div>  
 );  
}  
  
export default TrainerList;

## Code: Home.js

import React from 'react';  
  
function Home() {  
 return (  
 <div>  
 <h2>Welcome to Cognizant Academy Trainer Portal</h2>  
 <p>This app helps manage trainer profiles and expertise.</p>  
 </div>  
 );  
}  
  
export default Home;

## Code: TrainerDetail.js

import React from 'react';  
import { useParams } from 'react-router-dom';  
import trainers from './TrainersMock';  
  
function TrainerDetail() {  
 const { id } = useParams();  
 const trainer = trainers.find(t => t.trainerId === parseInt(id));  
  
 if (!trainer) return <h2>Trainer Not Found</h2>;  
  
 return (  
 <div>  
 <h2>{trainer.name}'s Details</h2>  
 <p><strong>ID:</strong> {trainer.trainerId}</p>  
 <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;

## Code: App.js

import React from 'react';  
import { BrowserRouter as Router, Routes, Route, Link } from 'react-router-dom';  
import Home from './Home';  
import TrainerList from './TrainerList';  
import TrainerDetail from './TrainerDetail';  
import trainers from './TrainersMock';  
  
function App() {  
 return (  
 <Router>  
 <div>  
 <nav>  
 <Link to="/">Home</Link> |{" "}  
 <Link to="/trainers">Trainers</Link>  
 </nav>  
  
 <Routes>  
 <Route path="/" element={<Home />} />  
 <Route path="/trainers" element={<TrainerList trainers={trainers} />} />  
 <Route path="/trainers/:id" element={<TrainerDetail />} />  
 </Routes>  
 </div>  
 </Router>  
 );  
}  
  
export default App;





