React-6

**Project Title: TrainersApp – Single Page Application using React Router**

**Objectives**

* Explain the necessity and advantages of using React Router.
* Identify the core components involved in React Router.
* List the various types of router components.
* Demonstrate parameter passing through the URL.

**Learning Outcomes**

By the end of this hands-on lab, the learner will be able to:

* Implement a simple navigation menu.
* Add and configure basic routes.
* Utilize routing effectively within a React application.

**Prerequisites**

To complete this hands-on lab, the following tools must be installed:

* Node.js
* NPM
* Visual Studio Code

**Estimated Time**

* Approximately 60 minutes.

**Use Case**

The goal is to develop a React-based SPA (Single Page Application) to manage a list of trainers, including their details and areas of expertise.

**Implementation Steps**

1. **Project Setup**
   * Create a new React application named TrainersApp using create-react-app.
   * Open the application in Visual Studio Code.
2. **Trainer Class**
   * Create a file trainer.js inside the src directory.
   * Define a class Trainer with the following properties:
     + TrainerId
     + Name
     + Email
     + Phone
     + Technology
     + Skills
3. **Mock Data**
   * Create TrainersMock.js to store mock data for trainers.
4. **React Router Installation**
   * Install React Router DOM using the command:
   * npm install react-router-dom
5. **Trainers List Component**
   * Create Trainerlist.js.
   * Define a TrainersList component that:
     + Accepts trainers' data as a prop.
     + Displays each trainer's name as a clickable hyperlink.
6. **Home Component**
   * Create Home.js.
   * Define a Home component responsible for rendering the home page layout.
7. **App Component and Routing**
   * Modify App.js to:
     + Include routing using BrowserRouter, Routes, Route, and Link components.
     + Define navigation links to / (Home) and /trainers (Trainers List).
8. **Trainer Detail Component**
   * Create TrainerDetails.js.
   * Define a TrainerDetail component that:
     + Retrieves the trainer ID from the URL using useParams.
     + Searches the trainer in mock data and displays detailed information.
9. **Link Integration**
   * Update the TrainersList component to:
     + Use Link to route to the TrainerDetail page, passing the trainer ID.
   * Add a corresponding route in the App component.

**Expected Layouts**

* **Home Page** (Figure 9)
* **Trainers List Page** (Figure 10)
* **Trainer Detail Page** (Figure 11)