**Week-6 Hands-On(React)**

6. 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.

The following trainers’ data application will deal.

1. T-ID
2. Name
3. Phone
4. Email
5. Stream
6. Skills

Create a new React app using *create-react-app* tool with the as “TrainersApp”

Open the application using the VS Code

Add a new file called *trainer.js* inside the **src folder** and define a class named as “Trainer” with the following properties

* 1. TrainerId
  2. Name
  3. Email
  4. Phone
  5. Technology
  6. Skills

**Objective:**

**Explain the need and benefits of React Router**  
React Router is needed to enable navigation between different views or components in a single-page React application, making the user experience seamless without full page reloads. Its benefits include improved navigation, bookmarkable URLs, and better organization of app structure.

**Identify the Components in React Router**  
Key components include:

* + <BrowserRouter>
  + <Routes>
  + <Route>
  + <Link>
  + <Navigate>
  + <Outlet>

**Parameter passing via URL**  
Parameters can be passed through the URL using route paths, such as /trainer/:id. The parameter value can then be accessed in the component using React Router hooks like useParams().

**List the types of Router Components**  
Common types are:

* + Browser Router
  + Hash Router
  + Memory Router

**Trainer.js:**

class Trainer {

constructor(trainerId, name, phone, email, technology, skills) {

this.trainerId = trainerId;

this.name = name;

this.phone = phone;

this.email = email;

this.technology = technology;

this.skills = skills;

}

}

export default Trainer;

**TrainerMock.js:**

import Trainer from './Trainer';

const trainers = [

new Trainer(1, 'John Doe', '9876543210', 'john@example.com', 'React', ['JS', 'React', 'Redux']),

new Trainer(2, 'Jane Smith', '9876543211', 'jane@example.com', 'Node', ['JS', 'Node', 'Express']),

new Trainer(3, 'Mark Taylor', '9876543212', 'mark@example.com', 'Python', ['Python', 'Django']),

];

export default trainers;

Home.js:

import React from 'react';

function Home() {

return (

<div>

<h2>Welcome to Cognizant Academy Trainers App</h2>

</div>

);

}

export default Home;

TrainerList.js:

import React from 'react';

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

function TrainerList({ 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 TrainerList;

**TrainerDetails.js:**

import React from 'react';

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

import trainers from './TrainersMock';

function TrainerDetails() {

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}</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 TrainerDetails;

**App.js:**

import React from 'react';

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

import Home from './Home';

import TrainerList from './TrainerList';

import TrainerDetails from './TrainerDetails';

import trainers from './TrainersMock';

function App() {

return (

<BrowserRouter>

<div>

<nav style={{ marginBottom: '20px' }}>

<Link to="/">Home</Link> |{" "}

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

</nav>

<Routes>

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

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

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

</Routes>

</div>

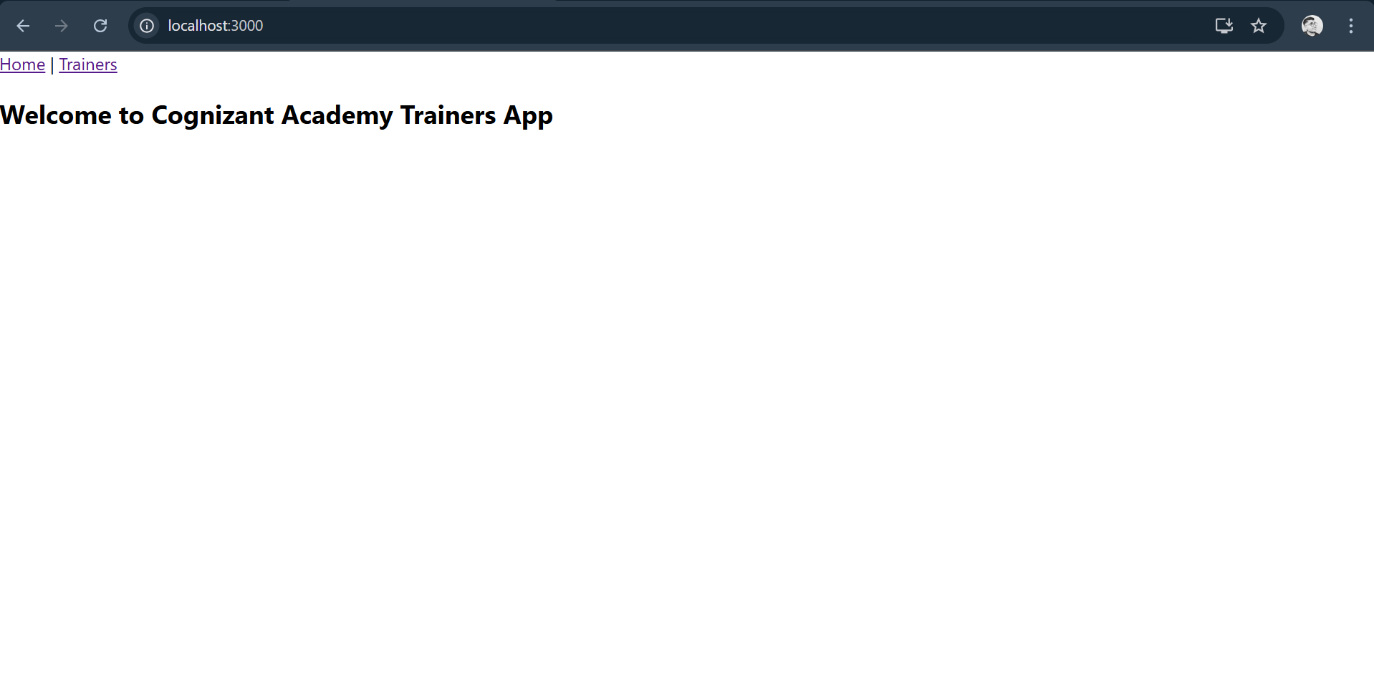
</BrowserRouter>

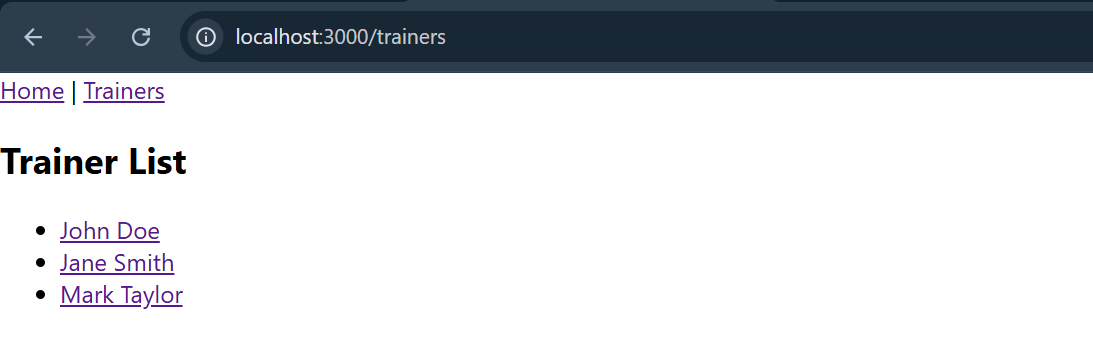
);

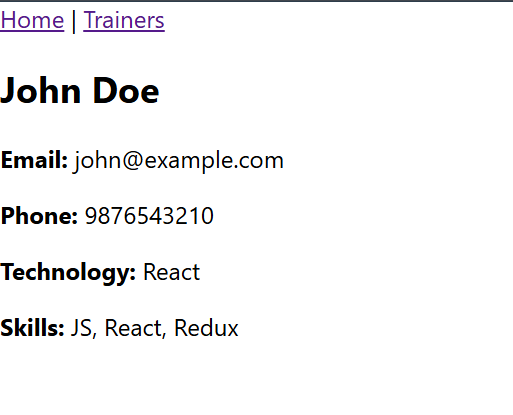
}

export default App;

**Output:**







7. Create a React Application named “shoppingapp” with a class component named “OnlineShopping” and “Cart”.

**Objectives:**

**Define Props:**  
Props (short for "properties") are read-only inputs passed from a parent component to a child component in React. They allow data and event handlers to flow from one component to another, helping components communicate and remain reusable.

**Identify the differences between State and Props:**

| **Aspect** | **Props** | **State** |
| --- | --- | --- |
| Mutability | Read-only, cannot be changed by child component | Read/write, managed within the component |
| Who sets it? | Set by the parent component | Set and managed by the component itself |
| Purpose | Pass data or functions to children | Store and manage local, dynamic data |

**Cart.js:**

import React, { Component } from 'react';

class Cart extends Component {

render() {

const { itemname, price } = this.props;

return (

<div style={{ borderBottom: '1px solid #ccc', padding: '10px' }}>

<p><strong>Item:</strong> {itemname}</p>

<p><strong>Price:</strong> ₹{price}</p>

</div>

);

}

}

export default Cart;

**OnlineShopping.js:**

import React, { Component } from 'react';

import Cart from './Cart';

class OnlineShopping extends Component {

constructor(props) {

super(props);

this.state = {

cartItems: [

{ itemname: 'Shoes', price: 2500 },

{ itemname: 'Shirt', price: 1200 },

{ itemname: 'Watch', price: 3500 },

{ itemname: 'Bag', price: 1800 },

{ itemname: 'Sunglasses', price: 900 }

]

};

}

render() {

return (

<div>

<h2>Shopping Cart</h2>

{this.state.cartItems.map((item, index) => (

<Cart key={index} itemname={item.itemname} price={item.price} />

))}

</div>

);

}

}

export default OnlineShopping;

**App.js:**

import React from 'react';

import OnlineShopping from './OnlineShopping';

function App() {

return (

<div className="App">

<OnlineShopping />

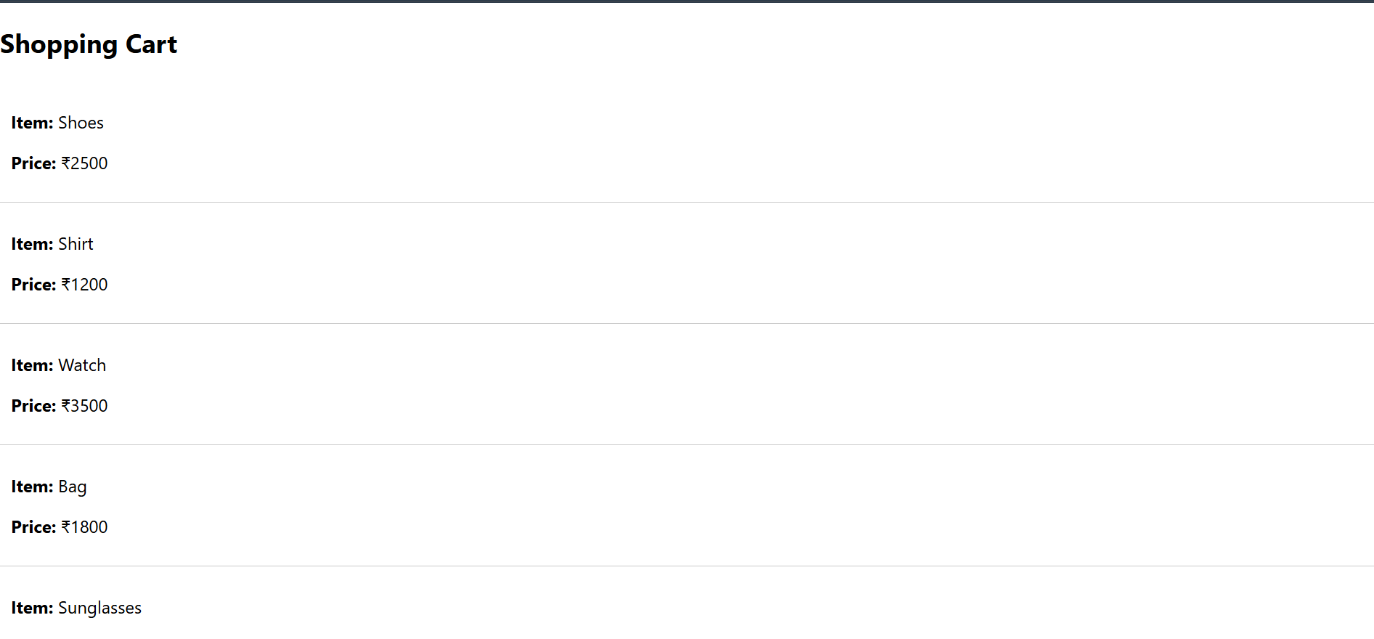
</div>

);

}

export default App;

Output:



8. Create a React App “counterapp” which will have a component named “CountPeople” which will have 2 methods.

UpdateEntry() à which will display the number of people who entered the mall.

UpdateExit() à which will display the number of people who exited the mall.

Use Constructor and state to Store the entrycount and exitcount.

The component has 2 buttons

1. Login à when clicked, the entrycount should get incremented by 1
2. Exit à when clicked, the exitcount should get incremented by 1

**Objective:**

**Define Props**

Props (short for "properties") are inputs passed from a parent component to a child component in React. They are **read-only**, meaning the child component cannot modify them. Props allow components to be reusable and dynamic by customizing their content or behavior based on the data passed down.

**Identify the differences between State and Props**

| **Aspect** | **Props** | **State** |
| --- | --- | --- |
| Mutability | Read-only; passed from parent to child component | Mutable; managed within the component |
| Source | Set and controlled by the parent component | Managed internally by the component itself |
| Purpose | Pass data and event handlers to child components | Store dynamic data that affects rendering |
| Can be changed by Child? | No | Yes, via setState or useState |
| Updating causes | Re-render when parent changes props | Re-render when state changes |

**CountPeople.js:**

import React, { Component } from 'react';

class CountPeople extends Component {

constructor(props) {

super(props);

this.state = {

entryCount: 0,

exitCount: 0

};

}

updateEntry = () => {

this.setState(prevState => ({

entryCount: prevState.entryCount + 1

}));

};

updateExit = () => {

this.setState(prevState => ({

exitCount: prevState.exitCount + 1

}));

};

render() {

return (

<div style={{ padding: '20px', textAlign: 'center' }}>

<h2>Mall Entry Counter</h2>

<p><strong>People Entered:</strong> {this.state.entryCount}</p>

<p><strong>People Exited:</strong> {this.state.exitCount}</p>

<button onClick={this.updateEntry} style={{ marginRight: '10px' }}>Login</button>

<button onClick={this.updateExit}>Exit</button>

</div>

);

}

}

export default CountPeople;

**App.js:**

import React from 'react';

import CountPeople from './CountPeople';

function App() {

return (

<div className="App">

<CountPeople />

</div>

);

}

export default App;

**Output:**

