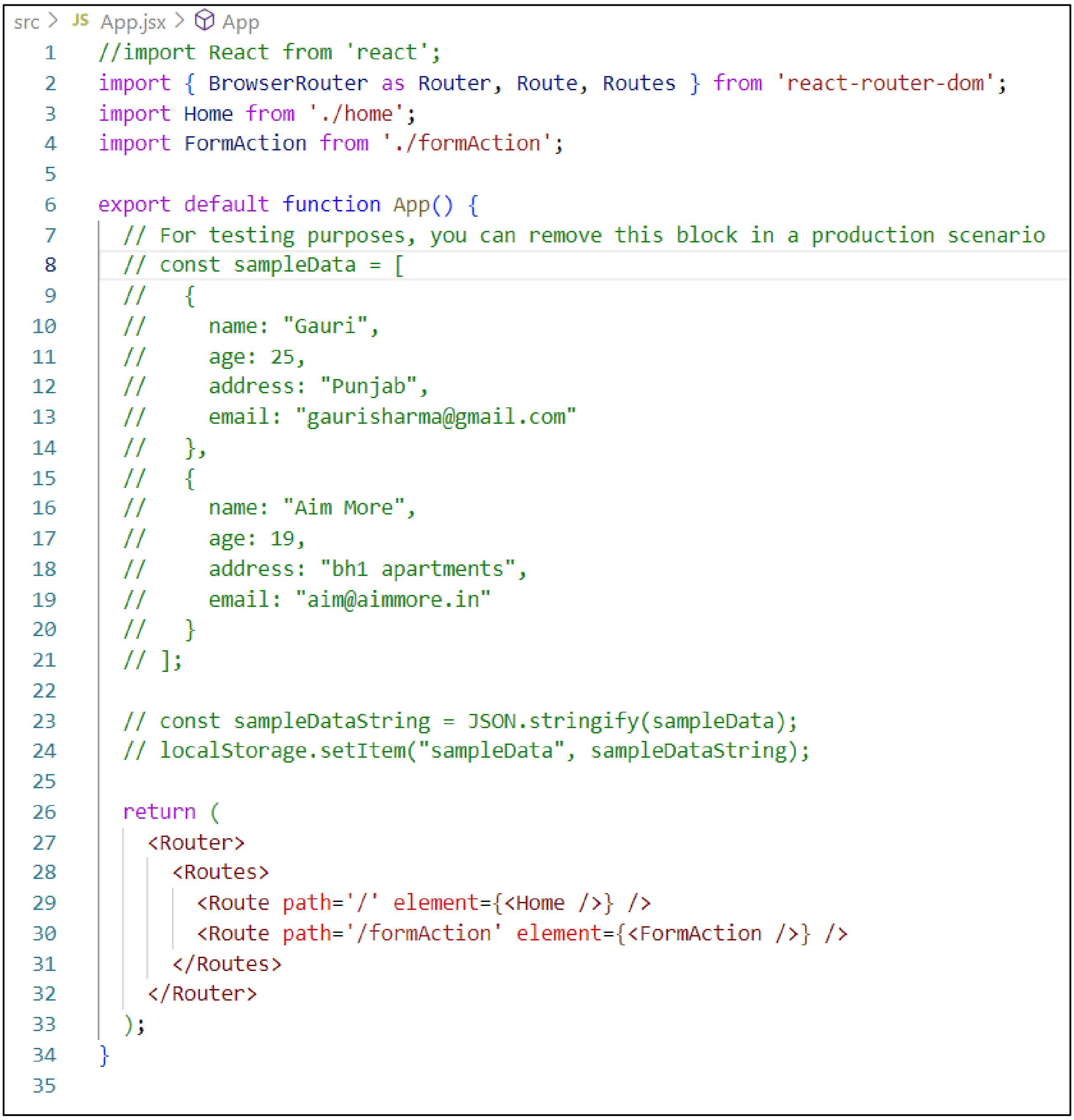
React JS Winter Pep Project

React App

This React application has two components (Home and Navbar) and it implements routing to navigate between these components. Additionally, it also has a FormAction component to gather user input.

Code Images:

app.jsx



```jsx
//import React from 'react';
import { BrowserRouter as Router, Route, Routes } from 'react-router-dom';
import Home from './home';
import FormAction from './formAction';

export default function App() {
  // For testing purposes, you can remove this block in a production scenario
  // const sampleData = [
  //   {
  //     name: "Gauri",
  //     age: 25,
  //     address: "Punjab",
  //     email: "gaurisharma@gmail.com"
  //   },
  //   {
  //     name: "Aim More",
  //     age: 19,
  //     address: "bh1 apartments",
  //     email: "aim@aimmore.in"
  //   }
  // ];

  // const sampleDataString = JSON.stringify(sampleData);
  // localStorage.setItem("sampleData", sampleDataString);

  return (
    <Router>
      <Routes>
        <Route path='/' element={<Home />} />
        <Route path='/formAction' element={<FormAction />} />
      </Routes>
    </Router>
  );
}

```

This code is a React application that uses the React Router library for navigation. It defines two components, Home and FormAction, representing different pages. The App component serves as the main component where routing is configured.

Import statements imports necessary components and functions from the 'react' and 'react-router-dom' libraries.

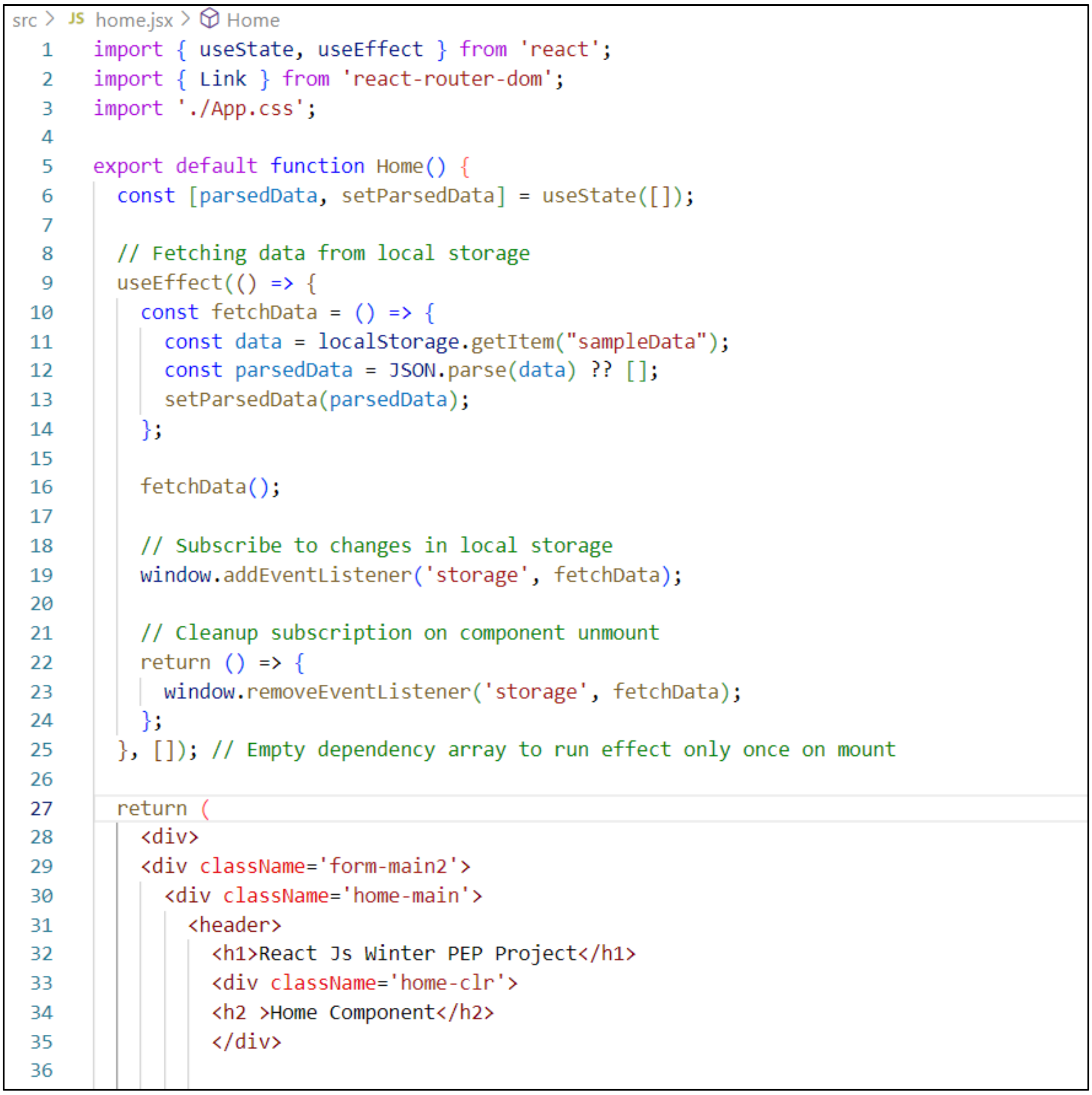
Routes Configuration uses the <Routes> component to define different routes in the application. There are two routes defined:

1. Route for the home page ('/') rendering the Home component.
2. Route for the 'formAction' page rendering the FormAction component.

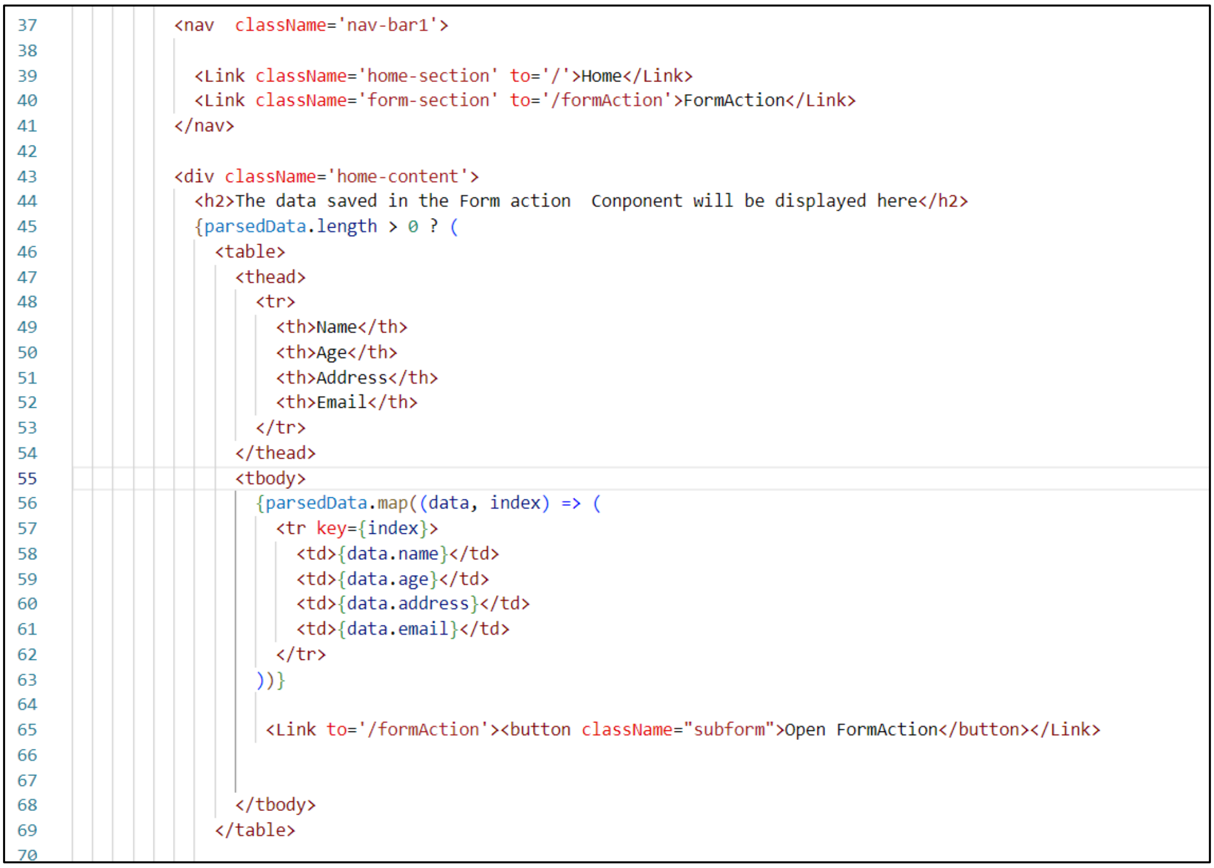
   The app.jsx code sets up a basic React application with two pages and navigation between them using React Router. It also includes a commented-out section for testing purposes with sample data stored in local storage.

Components of App:

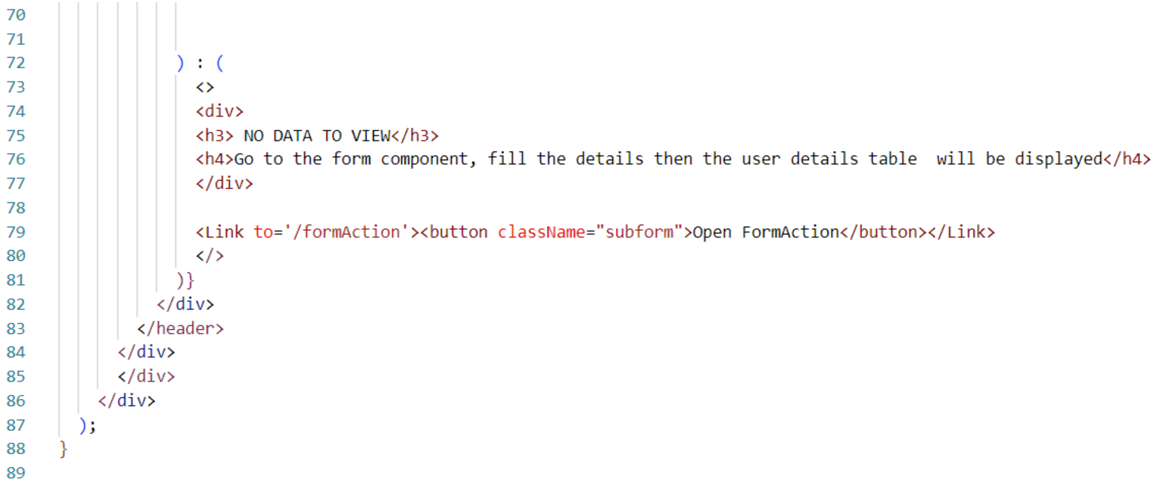
home.jsx



```jsx
import { useState, useEffect } from 'react';
import { Link } from 'react-router-dom';
import './App.css';

export default function Home() {
  const [parsedData, setParsedData] = useState([]);

  // Fetching data from local storage
  useEffect(() => {
    const fetchData = () => {
      const data = localStorage.getItem("sampleData");
      const parsedData = JSON.parse(data) ?? [];
      setParsedData(parsedData);
    };

    fetchData();

    // Subscribe to changes in local storage
    window.addEventListener('storage', fetchData);

    // Cleanup subscription on component unmount
    return () => {
      window.removeEventListener('storage', fetchData);
    };
  }, []); // Empty dependency array to run effect only once on mount

  return (
    <div>
    <div className='form-main2'>
      <div className='home-main'>
        <header>
          <h1>React Js Winter PEP Project</h1>
          <div className='home-clr'>
          <h2 >Home Component</h2>
          </div>

```



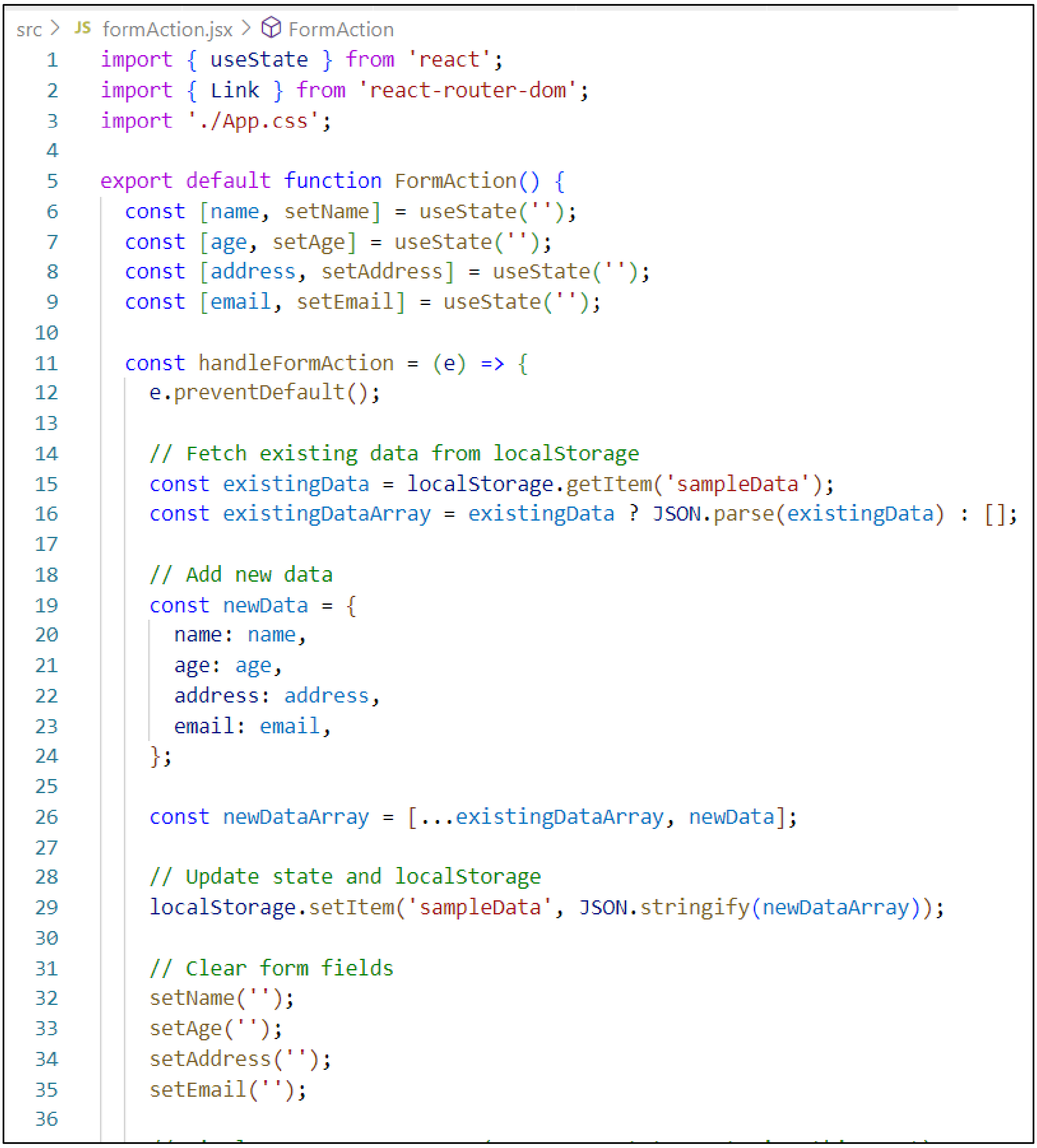
```
          <nav  className='nav-bar1'>

            <Link className='home-section'  to='/'>Home</Link>
            <Link className='form-section'  to='/formAction'>FormAction</Link>
          </nav>

          <div className='home-content'>
            <h2>The data saved in the Form action  Conponent will be displayed here</h2>
            {parsedData.length > 0 ? (
              <table>
                <thead>
                  <tr>
                    <th>Name</th>
                    <th>Age</th>
                    <th>Address</th>
                    <th>Email</th>
                  </tr>
                </thead>
                <tbody>
                  {parsedData.map((data, index) => (
                    <tr key={index}>
                      <td>{data.name}</td>
                      <td>{data.age}</td>
                      <td>{data.address}</td>
                      <td>{data.email}</td>
                    </tr>
                  ))}

                   <Link to='/formAction'><button className="subform">Open FormAction</button></Link>


                </tbody>
              </table>
```



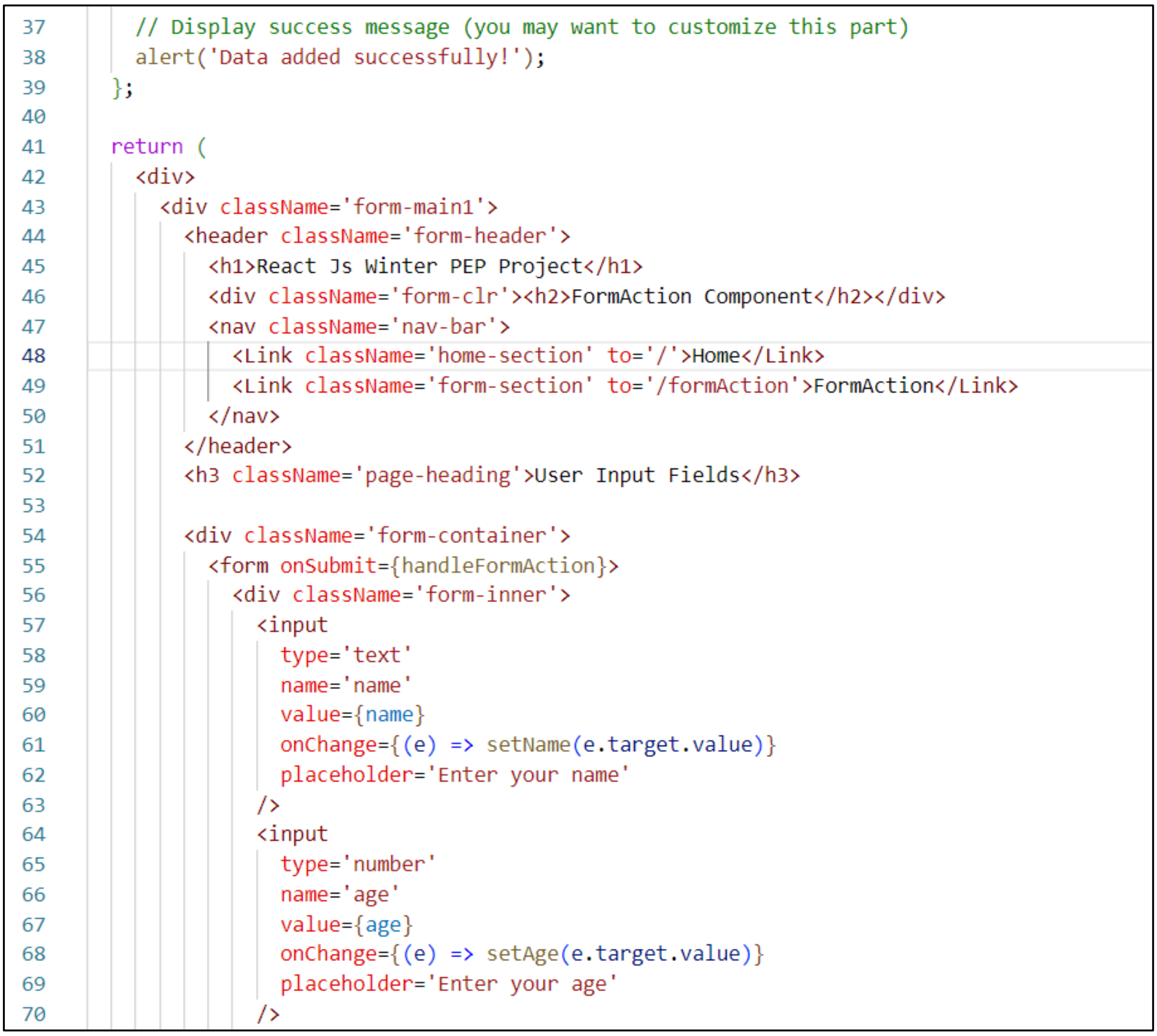
```


            ) : (
              <>
              <div>
              <h3> NO DATA TO VIEW</h3>
              <h4>Go to the form component, fill the details then the user details table  will be displayed</h4>
              </div>

              <Link to='/formAction'><button className="subform">Open FormAction</button></Link>
              </>
            )}
          </div>
        </header>
      </div>
      </div>
    </div>
  );
}

```

This piece of code defines a component in a React application called 'Home.' In simple terms, its job is to show information on the webpage. It keeps track of data using a special memory space called 'parsedData,' which is initially an empty list. When the webpage loads, it checks if there is any stored data in the browser's memory (local storage) and updates itself. If you've added new data or edited something, it notices and updates itself too. The webpage is designed with a clear structure: it has a title at the top, some links to move around the app, and a main content section where data is displayed. If there's data, it shows a neat table with details like Name, Age, Address, and Email. If there's no data, it kindly tells you so and suggests going to another part of the app ('FormAction') to add some. Lastly, there's a button to easily jump to the 'FormAction' section

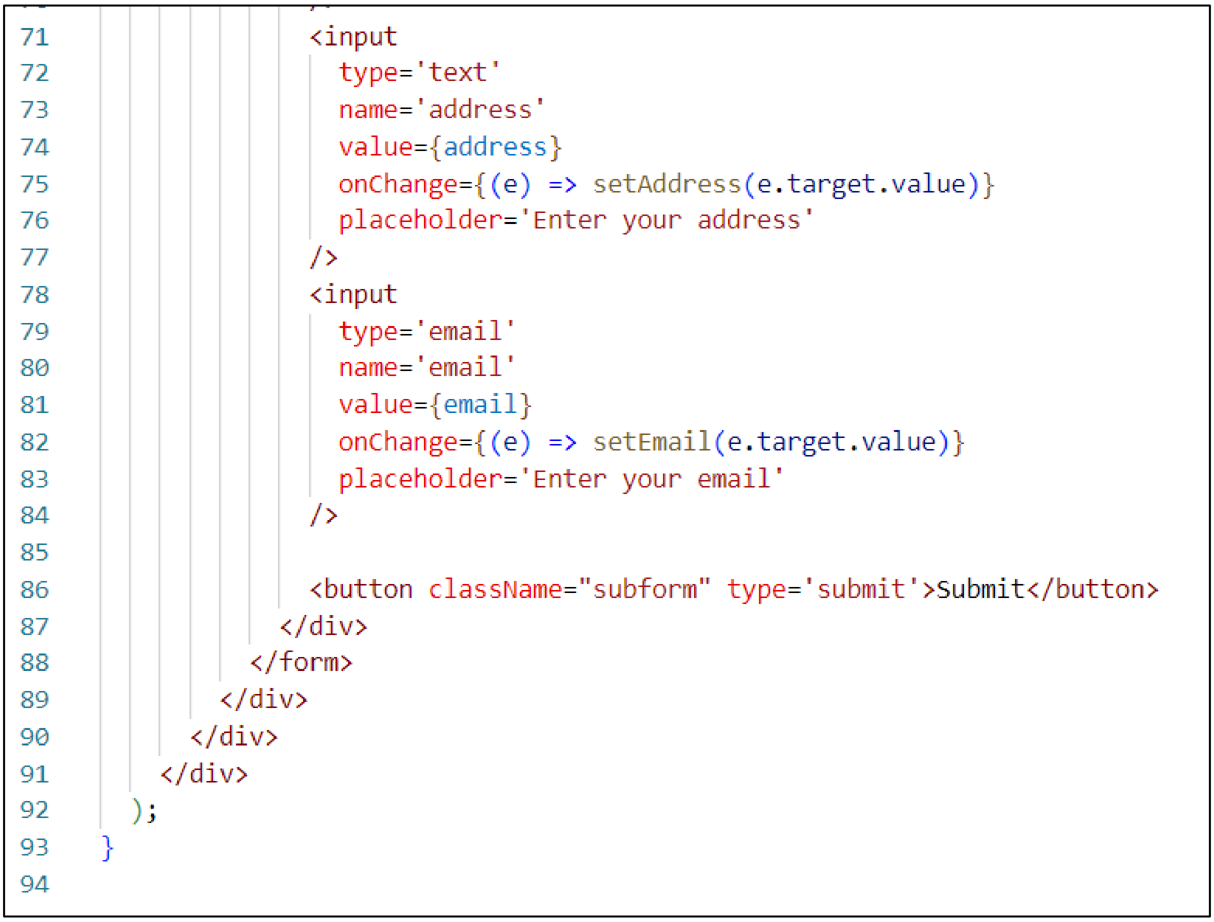
formAction.jsx:



```jsx
import { useState } from 'react';
import { Link } from 'react-router-dom';
import './App.css';

export default function FormAction() {
  const [name, setName] = useState('');
  const [age, setAge] = useState('');
  const [address, setAddress] = useState('');
  const [email, setEmail] = useState('');

  const handleFormAction = (e) => {
    e.preventDefault();

    // Fetch existing data from localStorage
    const existingData = localStorage.getItem('sampleData');
    const existingDataArray = existingData ? JSON.parse(existingData) : [];

    // Add new data
    const newData = {
      name: name,
      age: age,
      address: address,
      email: email,
    };

    const newDataArray = [...existingDataArray, newData];

    // Update state and localStorage
    localStorage.setItem('sampleData', JSON.stringify(newDataArray));

    // Clear form fields
    setName('');
    setAge('');
    setAddress('');
    setEmail('');

```



```
    // Display success message (you may want to customize this part)
    alert('Data added successfully!');
  };

  return (
    <div>
      <div className='form-main1'>
        <header className='form-header'>
          <h1>React Js Winter PEP Project</h1>
          <div className='form-clr'><h2>FormAction Component</h2></div>
          <nav className='nav-bar'>
            <Link className='home-section' to='/'>Home</Link>
            <Link className='form-section' to='/formAction'>FormAction</Link>
          </nav>
        </header>
        <h3 className='page-heading'>User Input Fields</h3>

        <div className='form-container'>
          <form onSubmit={handleFormAction}>
            <div className='form-inner'>
              <input
                type='text'
                name='name'
                value={name}
                onChange={(e) => setName(e.target.value)}
                placeholder='Enter your name'
              />
              <input
                type='number'
                name='age'
                value={age}
                onChange={(e) => setAge(e.target.value)}
                placeholder='Enter your age'
              />
```



```
              <input
                type='text'
                name='address'
                value={address}
                onChange={(e) => setAddress(e.target.value)}
                placeholder='Enter your address'
              />
              <input
                type='email'
                name='email'
                value={email}
                onChange={(e) => setEmail(e.target.value)}
                placeholder='Enter your email'
              />

              <button className="subform" type='submit'>Submit</button>
            </div>
          </form>
        </div>
      </div>
    </div>
  );
}
```

This code defines a React functional component named FormAction, which represents a form for user input.
In this code we utilized the “useState” hook to manage state for four input fields: name, age, address, and email.

Form Submission Handling: Defines a “handleFormAction” function that is triggered when the form is submitted.Prevents the default form submission behavior. This Retrieves existing data from local storage and parses it into an array (existingDataArray). Then (newData) constructs a new data object using the input values. After that the existing data is combined with the new data, which updates local storage, and resets form fields.

After the submit button is pressed a simple alert for successful data addition is diplayed.

The code creates the layout for a form wrapped in a container (a box on the webpage). Inside this box, there's a navigation bar containing links that direct users to the 'Home' and 'FormAction' sections of the application. These links are created using a special component from the react-router-dom library, making it easy for users to navigate between different parts of the webpage.

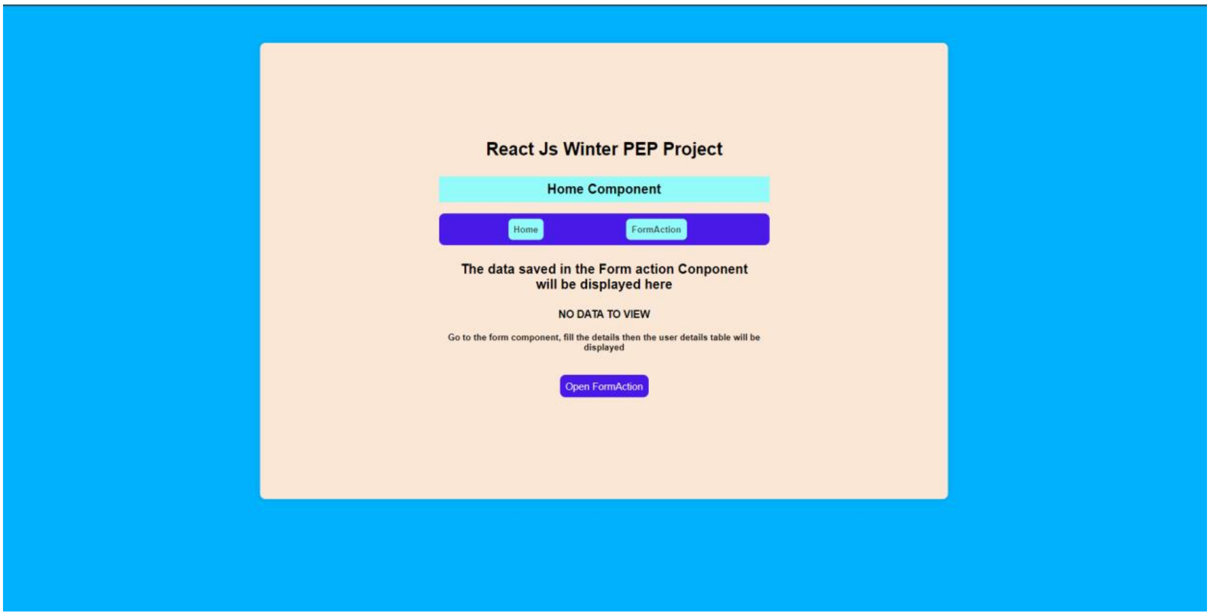
Form Inputs: Employs input fields for name, age, address, and email, each linked to their respective state variables. It utilizes the “onChange” event to update state as users input data.

The styles for this page are imported from an external CSS file => ('./App.css').

The submit button is put under the input fields which when clicked triggers the “handleFormAction” function. After successful submission an alert is shown.

Web Images:

1. HOMEPAGE



React Js Winter PEP Project

Home Component

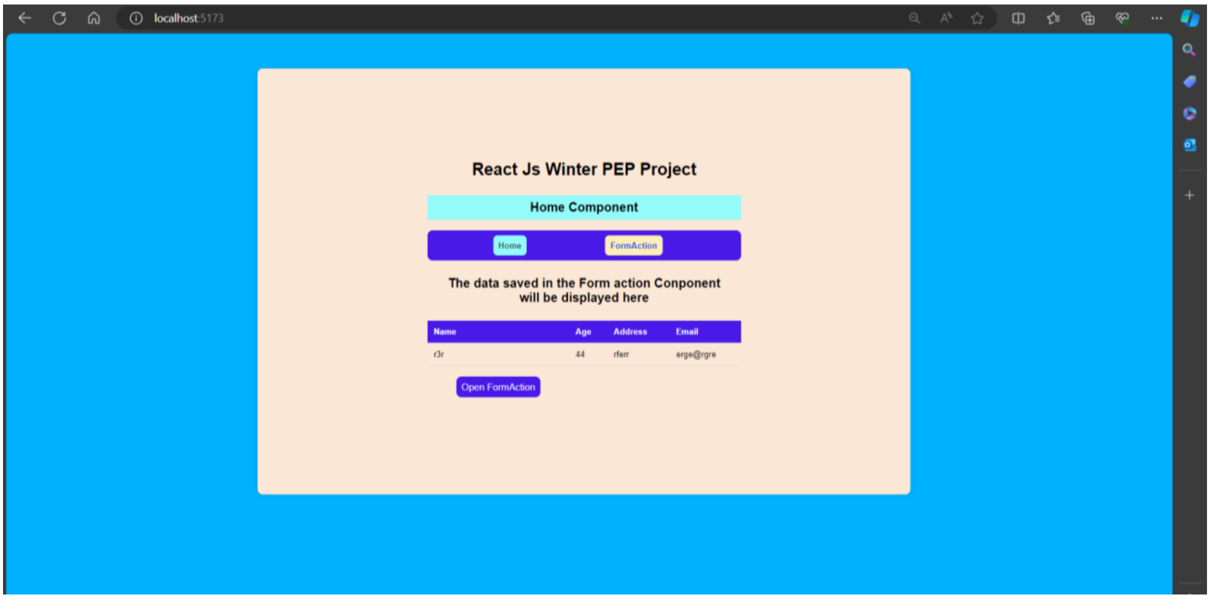
The data saved in the Form action Conponent will be displayed here

NO DATA TO VIEW

Go to the form component, fill the details then the user details table will be displayed

This home page has two components nav and home page body.

HOMEPAGE (After filling details in the Form in the Form Action Component):



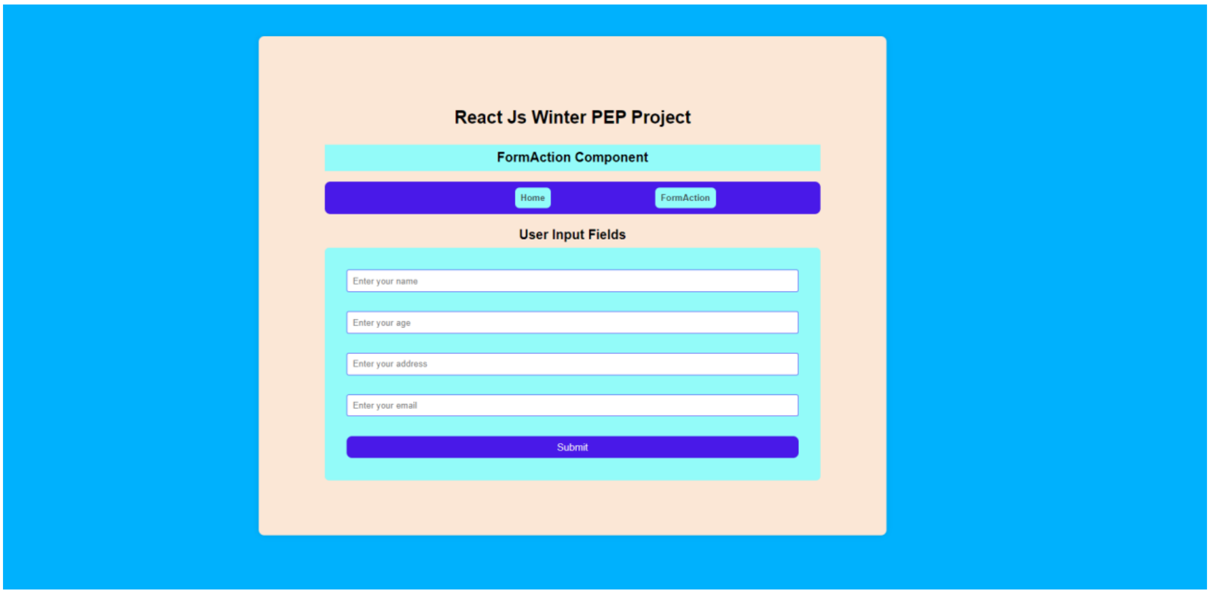
React Js Winter PEP Project

Home Component

The data saved in the Form action Conponent will be displayed here

Name	Age	Address	Email
r3r	44	rferr	erge@rgre

2. FORM ACTION

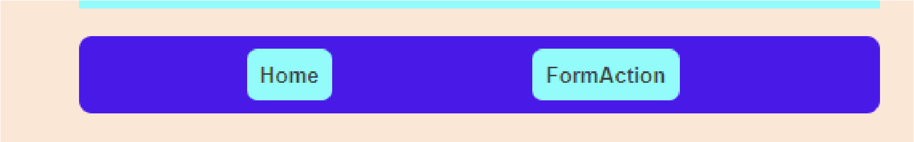


React Js Winter PEP Project

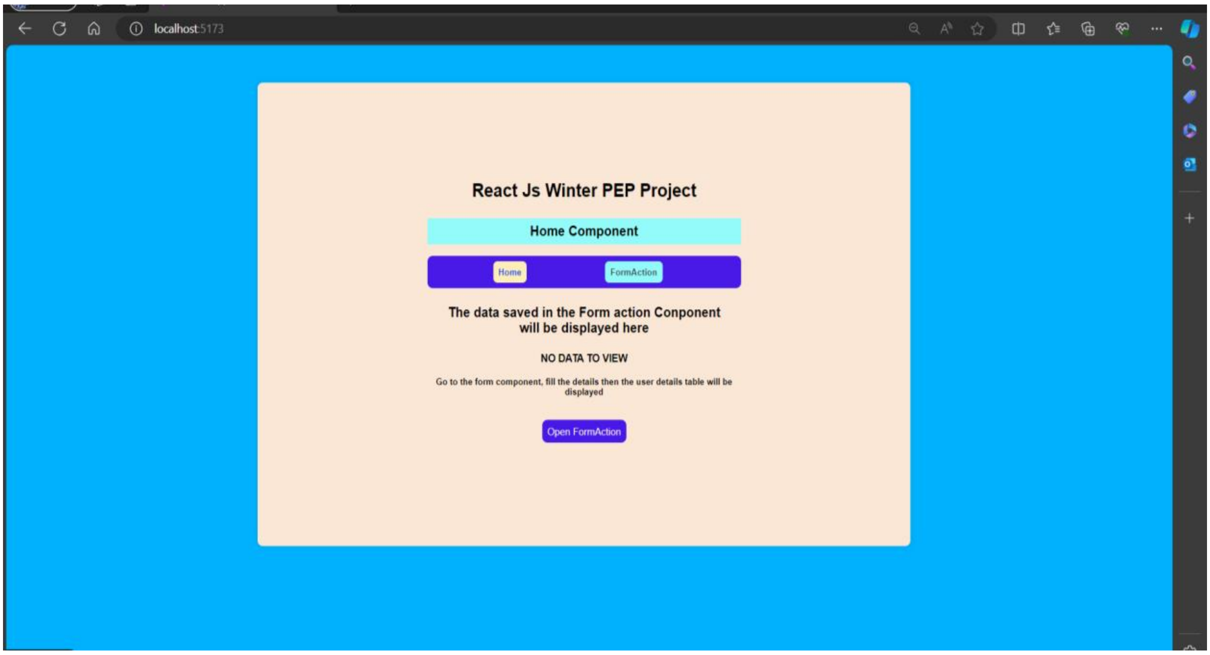
FormAction Component

User Input Fields

4. NAVBAR (before hovering)



5. NAVBAR (after hovering)



React Js Winter PEP Project

Home Component

The data saved in the Form action Conponent will be displayed here

NO DATA TO VIEW

Go to the form component, fill the details then the user details table will be displayed