# Lesson 04 Demo 03

# Implementing Route Parameters in a React Application

**Objective:** To demonstrate the implementation of route parameters in a React application using React Router and components for enhanced navigation and dynamic content

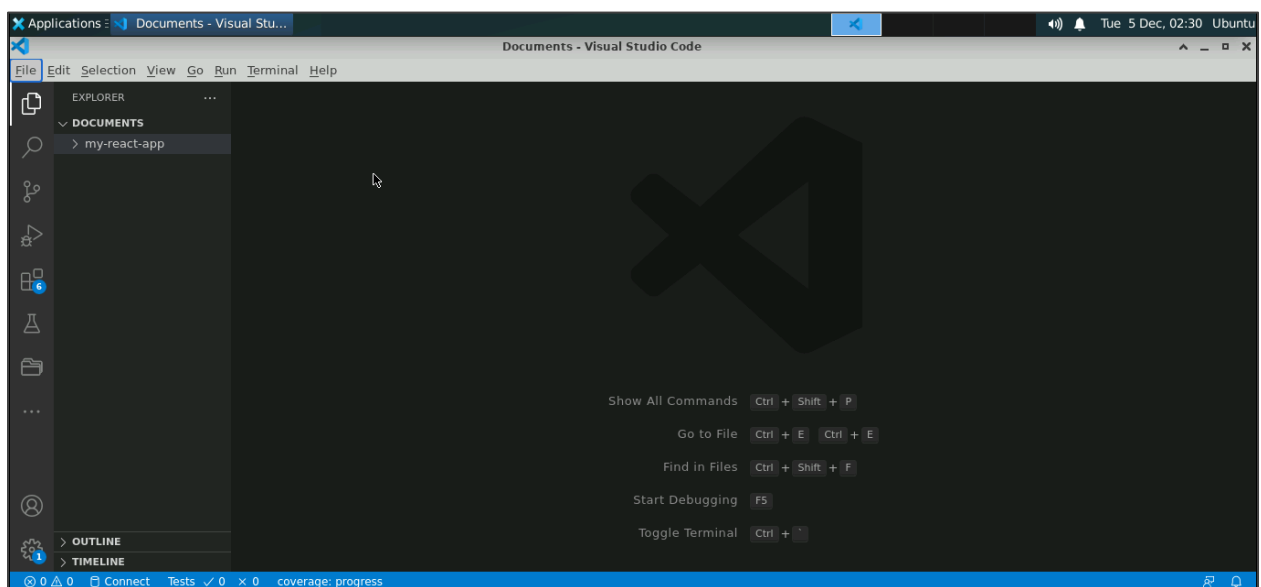**Tools required:** Visual Studio Code and Node.js

**Prerequisites:** A basic understanding of routes parameters in React
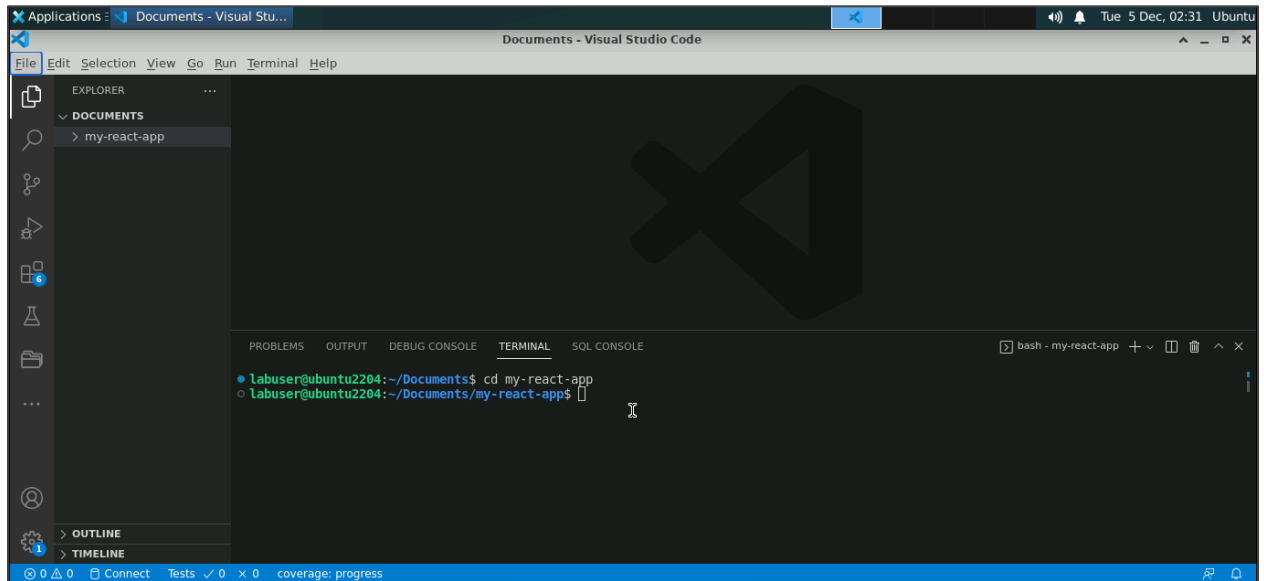
Steps to be followed:
1. Set up a new React app
2. Install React Router
3. Create component files
4. Create routes with parameters
5. Run the application

## Step 1: Set up a new React app

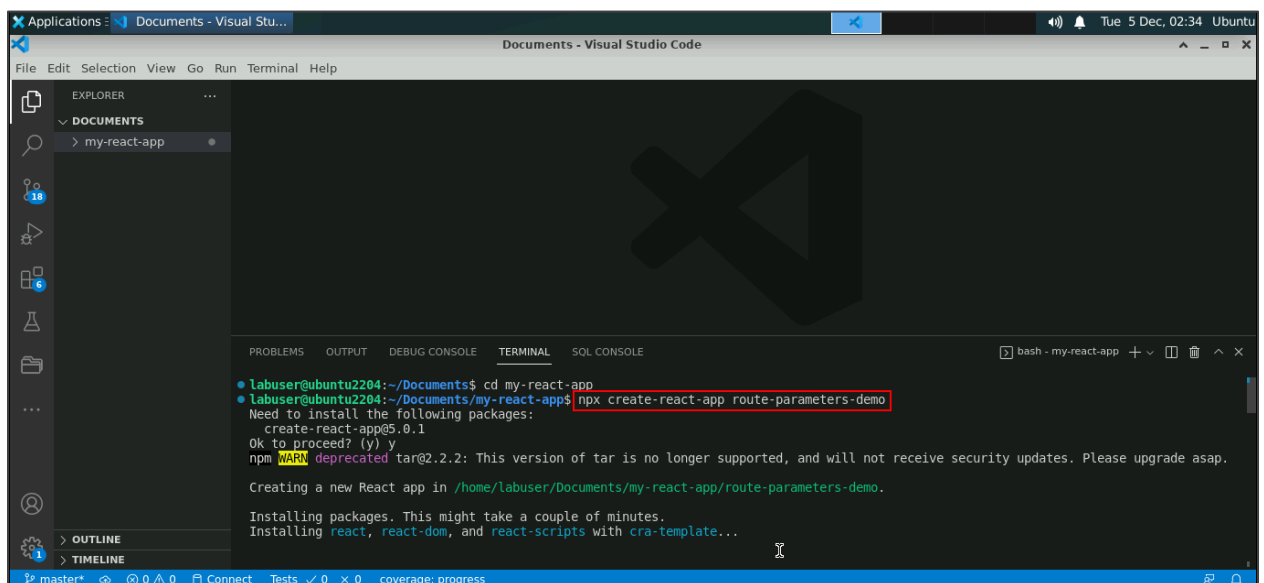1.1 Open **Visual Studio Code** and create a new folder named **my-react-app**

1.2 Open the terminal and navigate to the **my-react-app** folder using the following command:
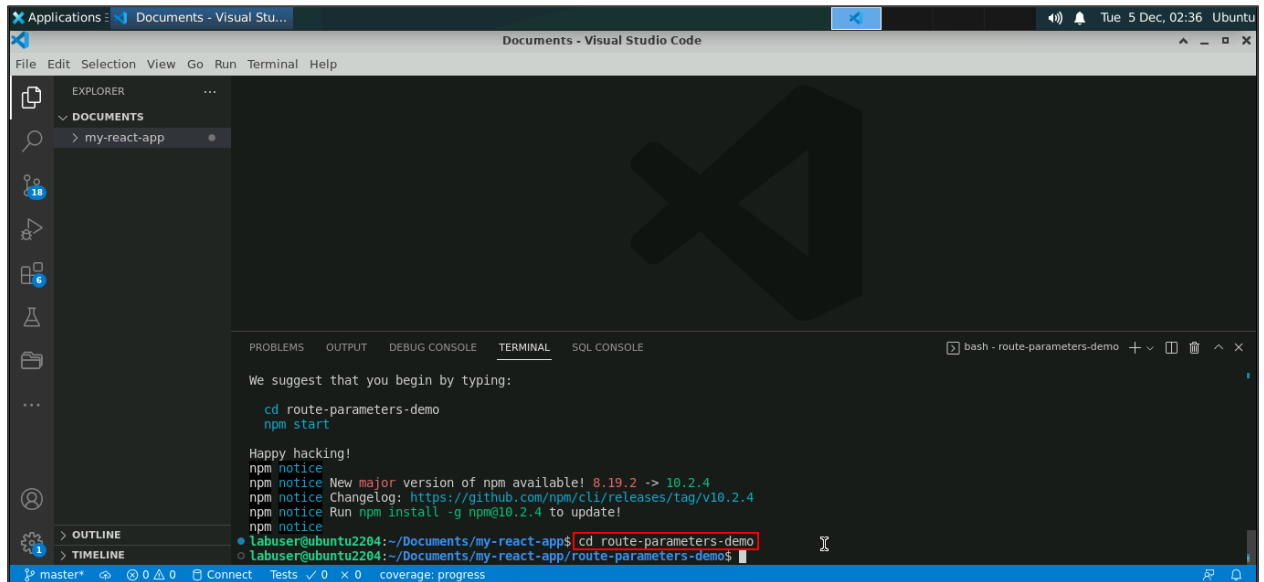**cd my-react-app**



1.3 Initialize a new React app using the following command:
**npx create-react-app route-parameters-demo**

1.4 Navigate to the **route-parameters-demo** folder using the following command:
**cd route-parameters-demo**



## Step 2: Install React Router

2.1 Install the React router using the following command:
**npm install react-router-dom**

## Step 3: Create component files

3.1 In the **src** folder, create two new folders named **components** and **pages**
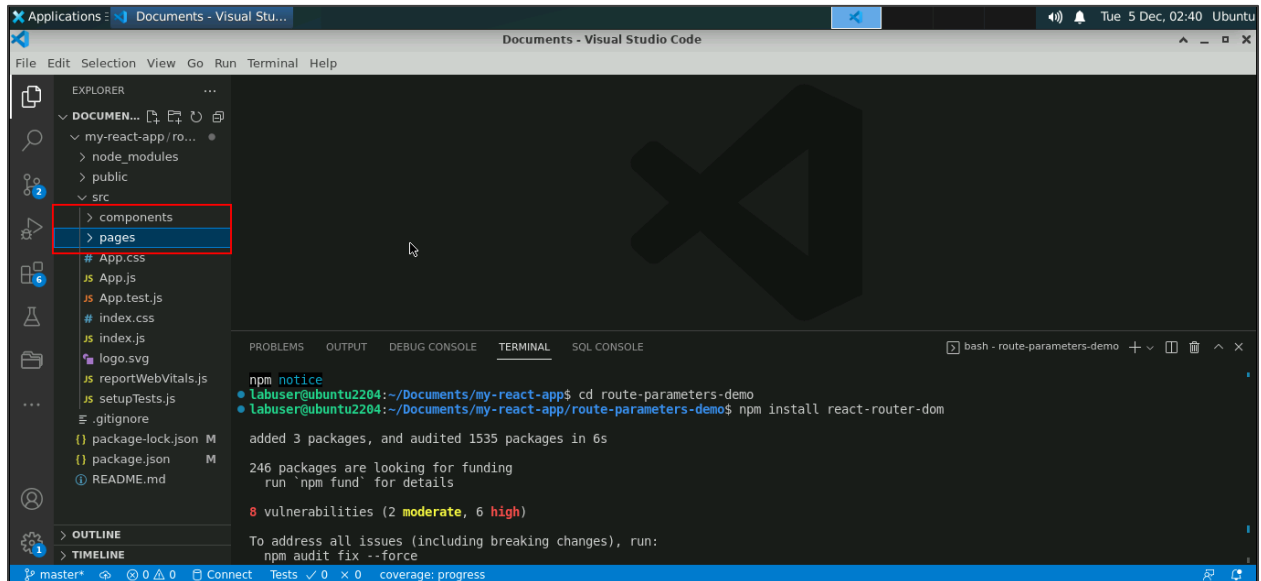


3.2 Inside the **components** folder, create a new JavaScript file named **Header.js**
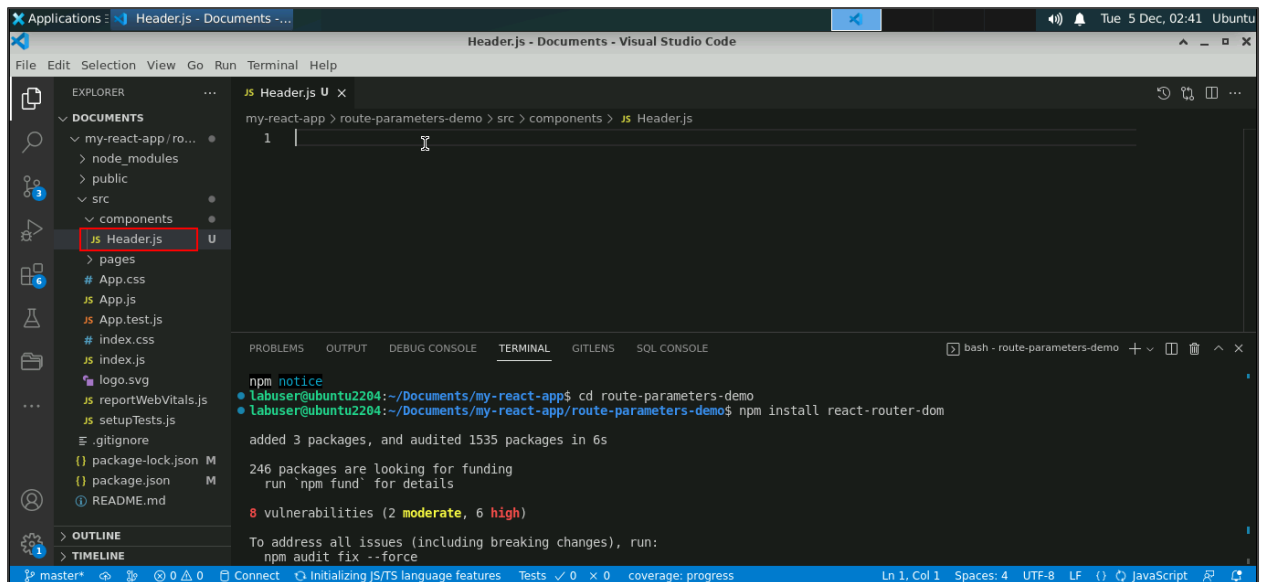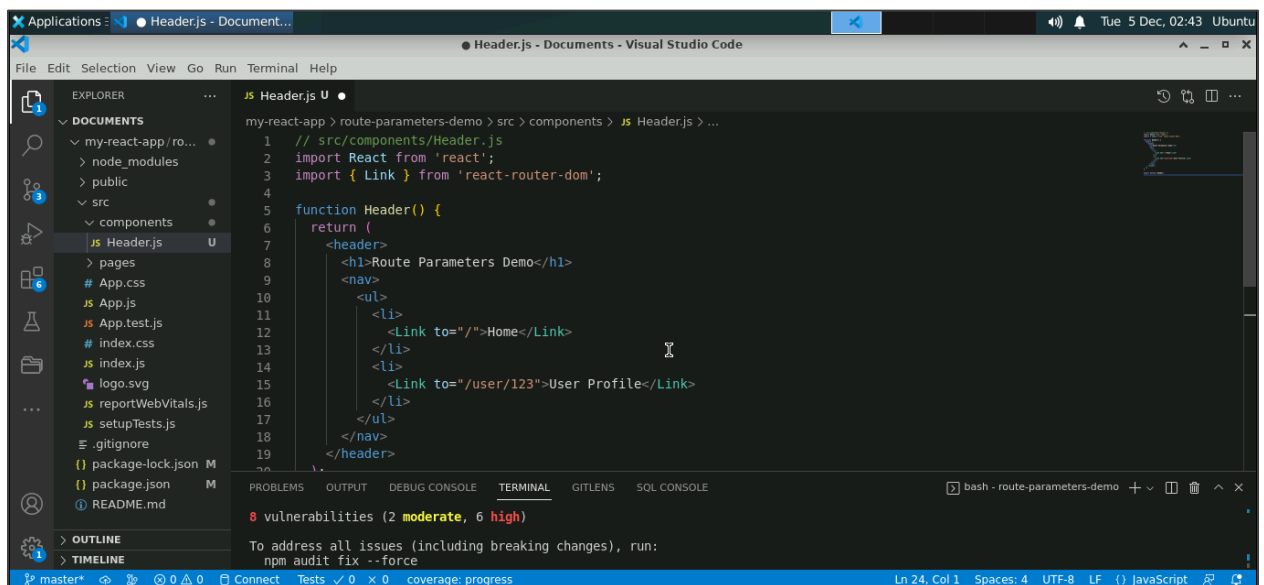
3.3 Add the following code to the **Header.js** file:

```
// src/components/Header.js
import React from 'react';
import { Link } from 'react-router-dom';

function Header() {
  return (
    <header>
      <h1>Route Parameters Demo</h1>
      <nav>
       <ul>
        <li>
          <Link to="/">Home</Link>
        </li>
        <li>
          <Link to="/user/123">User Profile</Link>
        </li>
       </ul>
      </nav>
    </header>
  );
}

export default Header;
```
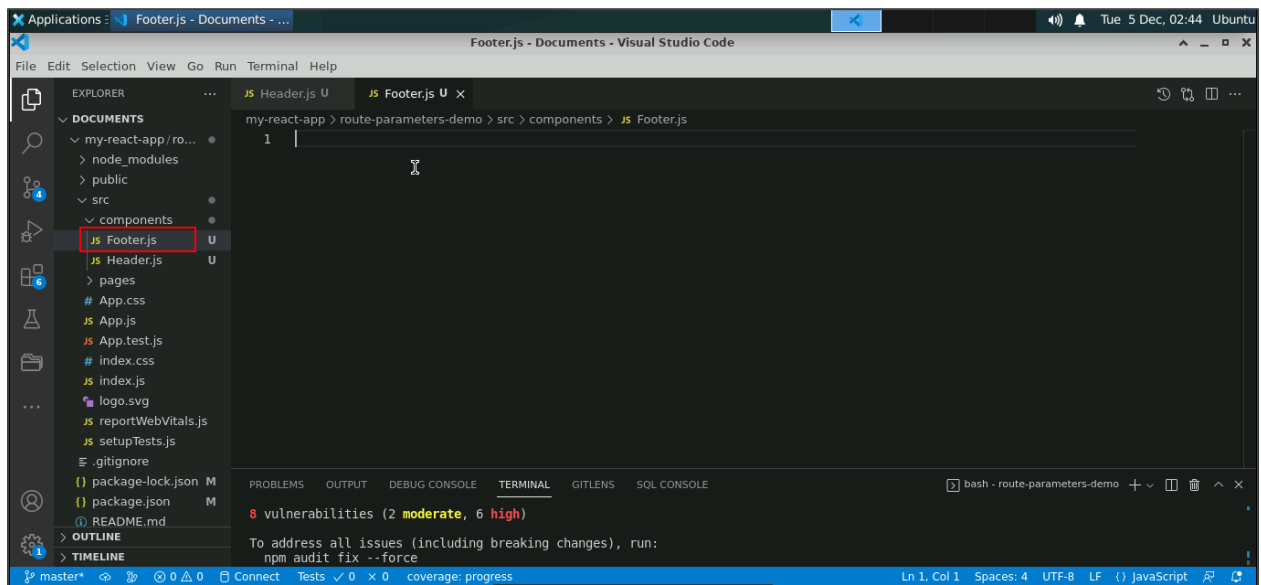


**Note:** Save the file.

**3.4** Inside the **components** folder, create a new JavaScript file named **Footer.js**



**3.5** Add the following code to the **Footer.js** file:

**// src/components/Footer.js**
**import React from 'react';**

**function Footer() {**
  **return (**
    **<footer>**
      **<p>© 2023 Route Parameters Demo</p>**
    **</footer>**
  **);**
**}**

**export default Footer;**

3.6 Inside the **pages** folder, create a new JavaScript file named **Home.js**

3.7 Add the following code to the **Home.js** file:

```
// src/pages/Home.js
import React from 'react';

function Home() {
  return (
    <div>
      <h2>Welcome to the Home Page!</h2>
    </div>

  );
}

export default Home;
```
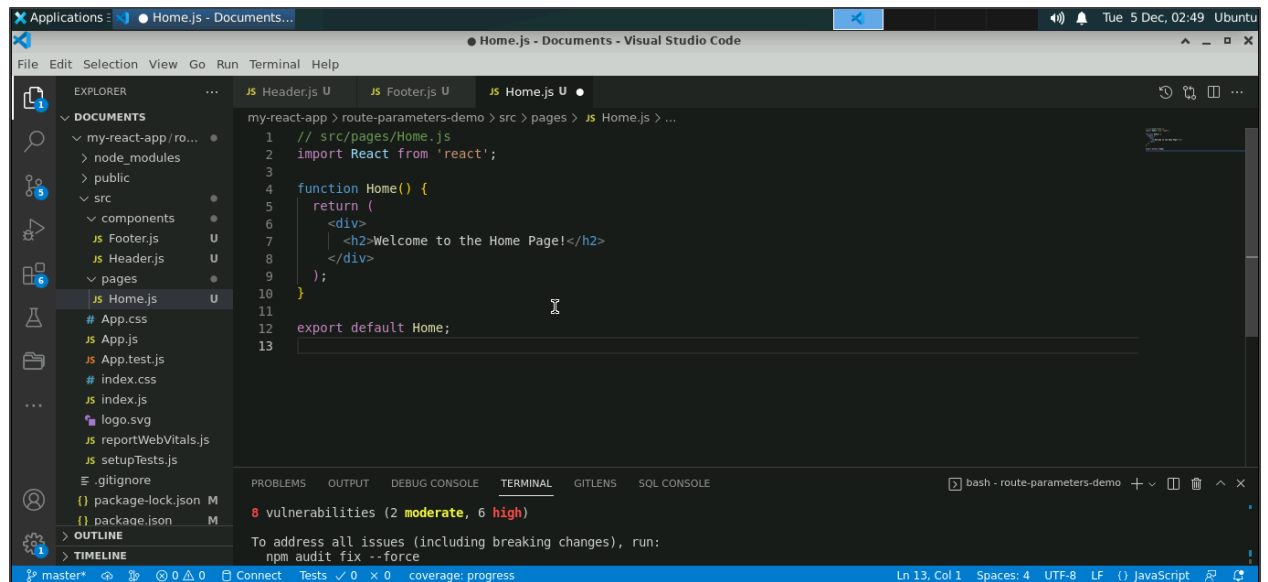
3.8 Inside the **pages** folder, create a new JavaScript file named **UserProfile.js**
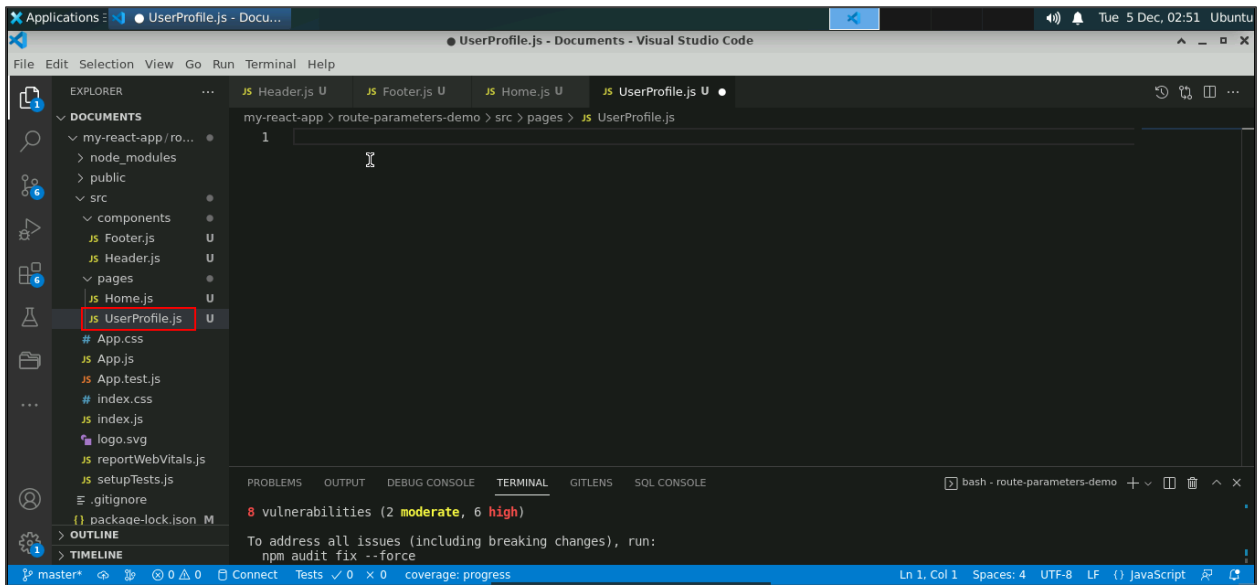


3.9 Add the following code to the **UserProfile.js** file:

**// src/pages/UserProfile.js**
**import React from 'react';**
**import { useParams } from 'react-router-dom';**

**function UserProfile() {**
  **// Extracting user ID from the route parameters**
  **const { userId } = useParams();**

  **return (**
   **<div>**
    **<h2>User Profile</h2>**
    **<p>User ID: {userId}</p>**
   **</div>**
  **);**
**}**

**export default UserProfile;**
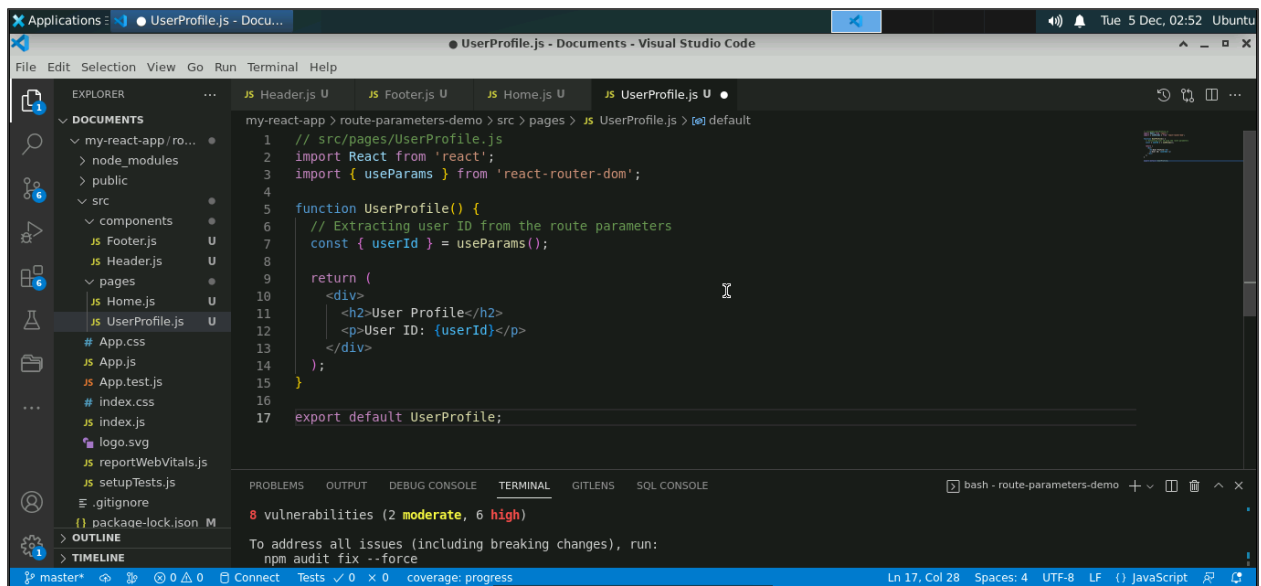
## Step 4: Create routes with parameters

4.1 Replace the existing code in the **App.js** file with the following code:

```
// src/App.js
import React from 'react';
import { Routes, Route, Link } from 'react-router-dom';
import Header from './components/Header';
import Footer from './components/Footer';
import Home from './pages/Home';
import UserProfile from './pages/UserProfile';


function App() {
  return (
    <div>
      <Header />

      <Routes>
        <Route path="/" exact element={<Home/>} />
        <Route path="/user/:userId" element={<UserProfile/>} />
      </Routes>

      <Footer />
    </div>
  );
}
```

**export default App;**
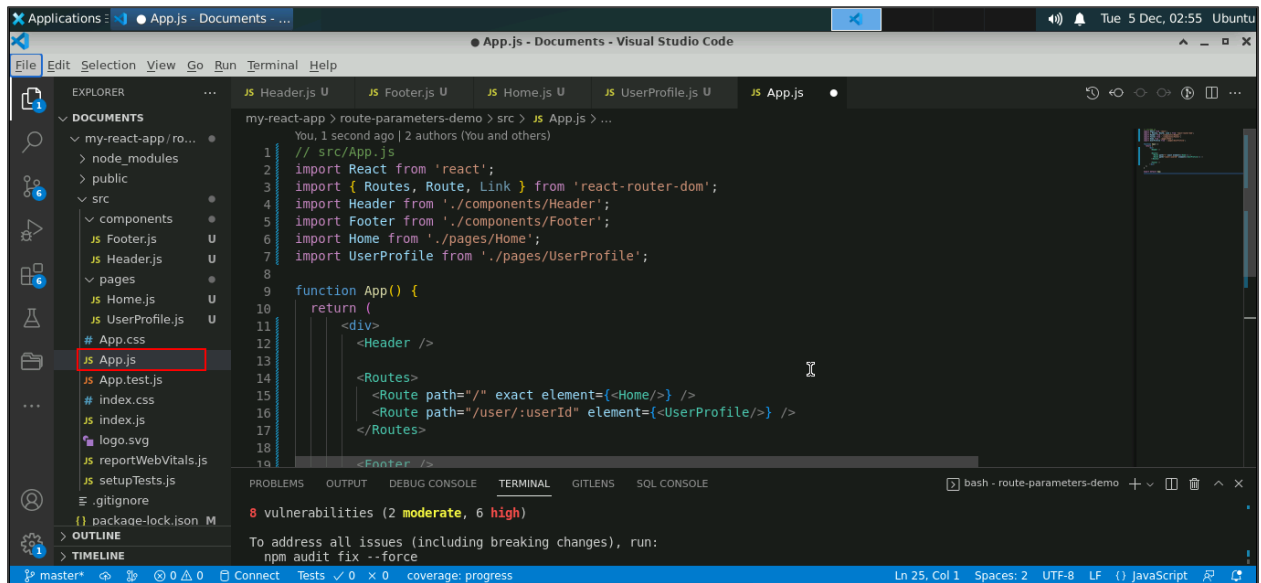


4.2 Replace the existing code in the **index.js** file with the following code:
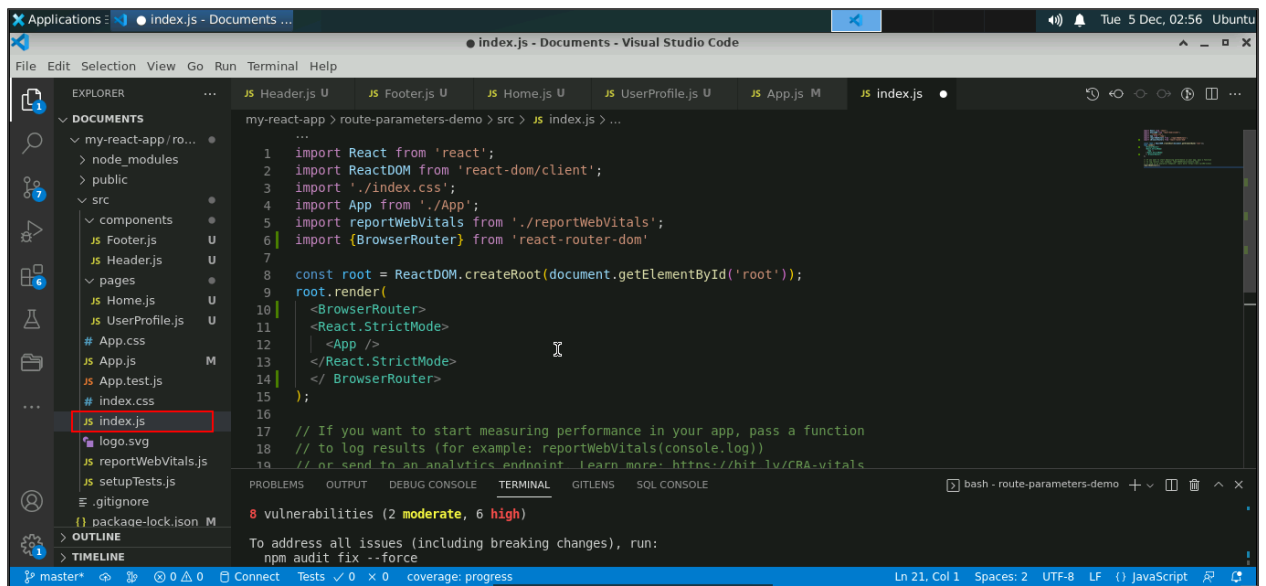
**import React from 'react';**
**import ReactDOM from 'react-dom/client';**
**import './index.css';**
**import App from './App';**
**import reportWebVitals from './reportWebVitals';**
**import {BrowserRouter} from 'react-router-dom'**

**const root = ReactDOM.createRoot(document.getElementById('root'));**
**root.render(**

  **<BrowserRouter>**
  **<React.StrictMode>**
   **<App />**
  **</React.StrictMode>**
  **</ BrowserRouter>**

**);**

**// If you want to start measuring performance in your app, pass a function**
**// to log results (for example: reportWebVitals(console.log))**
**// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals**
**reportWebVitals();**

**Note:** Save all the files.

## Step 5: Run the application

5.1 Start the React development server using the following command:
**npm start**

By following these steps, you have successfully set up a React application with route parameters to facilitate dynamic content and improved navigation using React Router and components.