

## Lesson 04 Demo 05

### Implementing Protected Routes in a React Application

**Objective:** To demonstrate the react routing with protected routing features to enhance security and control navigation based on user sign-in status

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

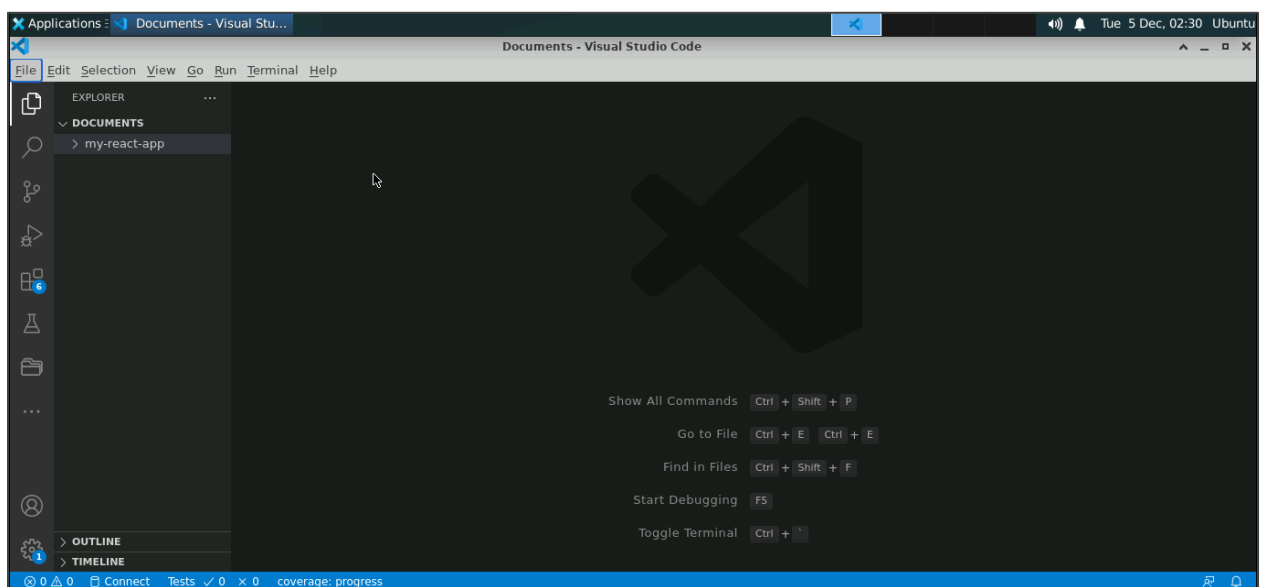
**Prerequisites:** A basic understanding of Protected Routes in React Router

Steps to be followed:

1. Set up a new React application
2. Install React Router
3. Create component files
4. Implement Protected Routes
5. Run the application

#### Step 1: Set up a new React application

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**

The screenshot shows the Visual Studio Code interface. The Explorer panel on the left shows the file structure of the 'my-react-app' folder. The Terminal panel at the bottom shows the command prompt with the following commands and output:

```
labuser@ubuntu2204:~/Documents$ cd my-react-app
labuser@ubuntu2204:~/Documents/my-react-app$
```

- 1.3 Initialize a new React app using the following command:  
**npx create-react-app protected-routes**

The screenshot shows the Visual Studio Code interface. The Explorer panel on the left shows the file structure of the 'my-react-app' folder, including 'node\_modules', 'public', 'src', '.gitignore', 'package-lock.json', 'package.json', and 'README.md'. The Terminal panel at the bottom shows the command prompt with the following commands and output:

```
labuser@ubuntu2204:~/Documents$ cd my-react-app
labuser@ubuntu2204:~/Documents/my-react-app$ npx create-react-app protected-routes
Need to install the following packages:
  create-react-app@5.0.1
Ok to proceed? (y) y
npm WARN deprecated tar@2.2.2: This version of tar is no longer supported, and will not receive security updates. Please upgrade asap.
Creating a new React app in /home/labuser/Documents/my-react-app/protected-routes.
Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...
```

1.4 Navigate to the **protected-routes** folder using the following command:  
**cd protected-routes**

The screenshot shows the Visual Studio Code interface. The Explorer panel on the left shows the file structure of a project named 'my-react-app'. The file 'package-lock.json' is highlighted. The Terminal panel at the bottom shows the following output:

```

We suggest that you begin by typing:

  cd protected-routes
  npm start

Happy hacking!
npm notice
npm notice New major version of npm available! 8.19.2 -> 10.2.5
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.2.5
npm notice Run npm install -g npm@10.2.5 to update!
npm notice
labuser@ubuntu2204:~/Documents/my-react-app$ cd protected-routes
labuser@ubuntu2204:~/Documents/my-react-app/protected-routes$

```

## Step 2: Install React Router

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

The screenshot shows the Visual Studio Code interface. The Explorer panel on the left shows the file structure of a project named 'my-react-app'. The file 'package-lock.json' is highlighted. The Terminal panel at the bottom shows the following output:

```

npm notice
labuser@ubuntu2204:~/Documents/my-react-app$ cd protected-routes
labuser@ubuntu2204:~/Documents/my-react-app/protected-routes$ npm install react-router-dom

added 3 packages, and audited 1535 packages in 4s

246 packages are looking for funding
run 'npm fund' for details

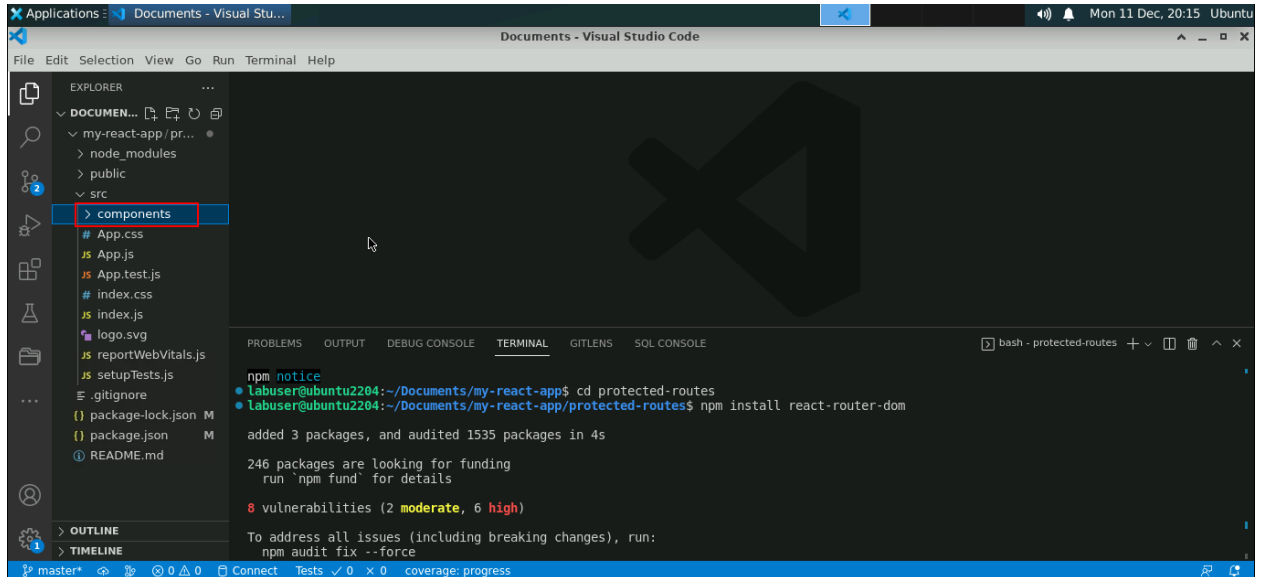
8 vulnerabilities (2 moderate, 6 high)

To address all issues (including breaking changes), run:
  npm audit fix --force

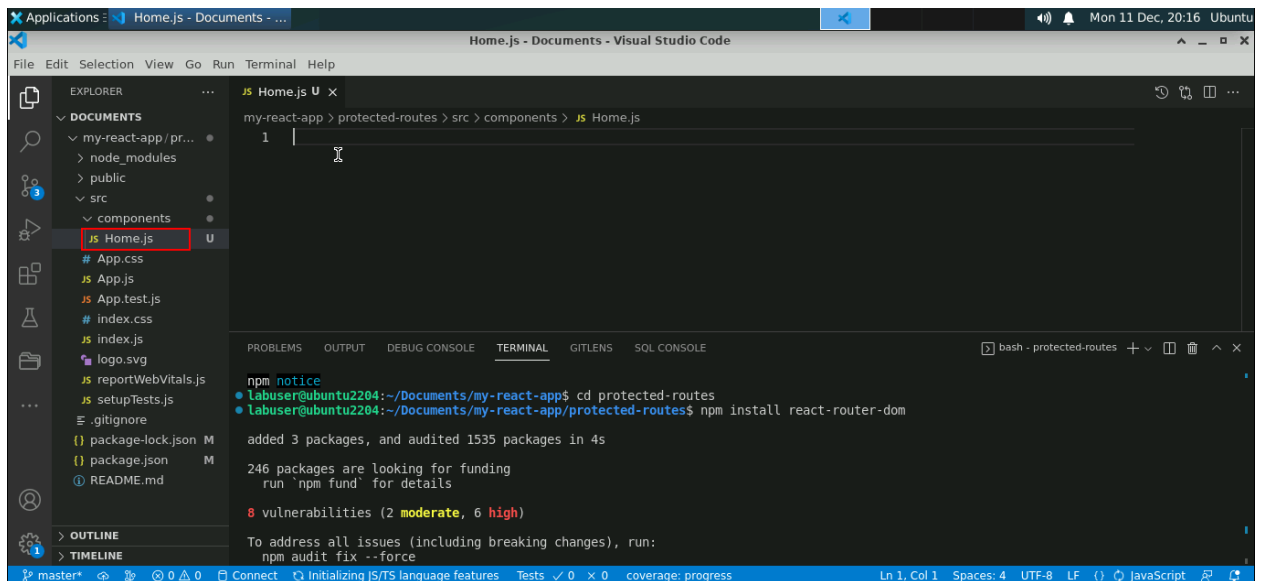
```

## Step 3: Create component files

### 3.1 In the **src** folder, create a new folder named **components**



### 3.2 Inside the **components** folder, create a new JavaScript file named **Home.js**



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

```
import React from 'react'
function Home() {
  return (
    <div className="card mt-5 text-center">
      <div className="card-body">Home</div>
    </div>
  )
}
export default Home
```

The screenshot shows the Visual Studio Code editor with the **Home.js** file open. The file content matches the code provided in the previous block. The Explorer sidebar shows the project structure with **components** > **Home.js** selected. The terminal at the bottom shows the command `npm install react-router-dom` being executed, with output indicating that 3 packages were added and 246 packages are looking for funding.

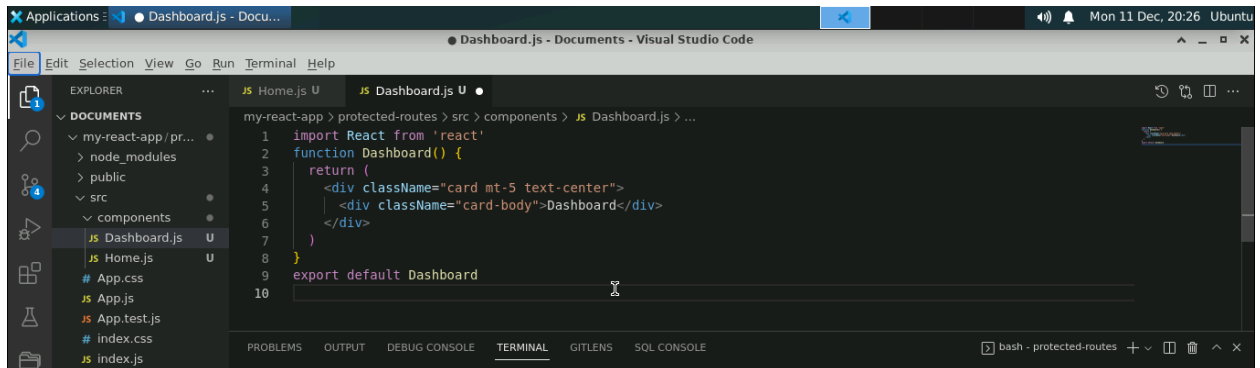
**Note:** Save the file

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

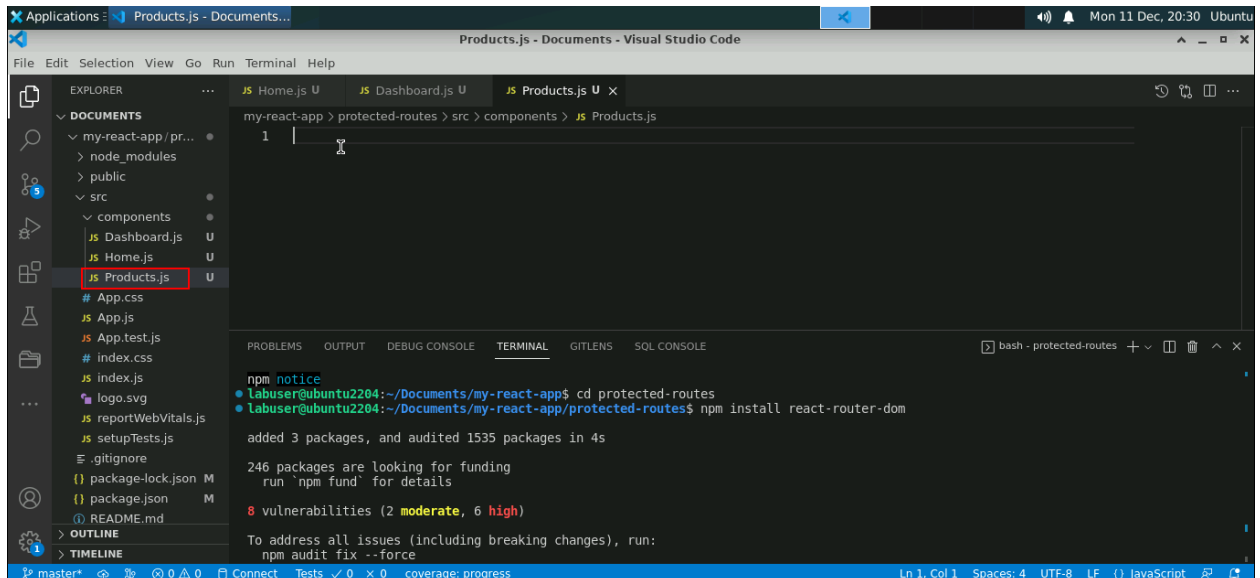
The screenshot shows the Visual Studio Code editor with the **Dashboard.js** file open. The file is currently empty. The Explorer sidebar shows the project structure with **components** > **Dashboard.js** selected. The terminal at the bottom shows the command `npm install react-router-dom` being executed, with output indicating that 3 packages were added and 246 packages are looking for funding.

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

```
import React from 'react'
function Dashboard() {
  return (
    <div className="card mt-5 text-center">
      <div className="card-body">Dashboard</div>
    </div>
  )
}
export default Dashboard
```



3.6 Inside the **components** folder, create a new JavaScript file named **Products.js**



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

```
import React from 'react'
function Products() {
```

```

return (
  <div className="card mt-5 text-center">
    <div className="card-body">Products</div>
  </div>
)
}
export default Products

```

The screenshot shows the Visual Studio Code editor with the 'Products.js' file open. The file content is as follows:

```

1 import React from 'react'
2 function Products() {
3   return (
4     <div className="card mt-5 text-center">
5       <div className="card-body">Products</div>
6     </div>
7   )
8 }
9 export default Products
10

```

The Explorer sidebar on the left shows the project structure, with 'Products.js' highlighted under the 'components' folder. The terminal at the bottom shows the command to install 'react-router-dom' and the resulting npm notice, including the number of packages audited and the presence of vulnerabilities.

3.8 Inside the **components** folder, create a new JavaScript file named **Protected.js**

The screenshot shows the Visual Studio Code editor with the 'Protected.js' file open. The file content is as follows:

```

1

```

The Explorer sidebar on the left shows the project structure, with 'Protected.js' highlighted under the 'components' folder. The terminal at the bottom shows the command to install 'react-router-dom' and the resulting npm notice, including the number of packages audited and the presence of vulnerabilities.

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

```
import React from 'react'
import { Navigate } from 'react-router-dom'
function Protected({ isSignedIn, children }) {
  if (!isSignedIn) {
    return <Navigate to="/" replace />
  }
  return children
}
export default Protected
```

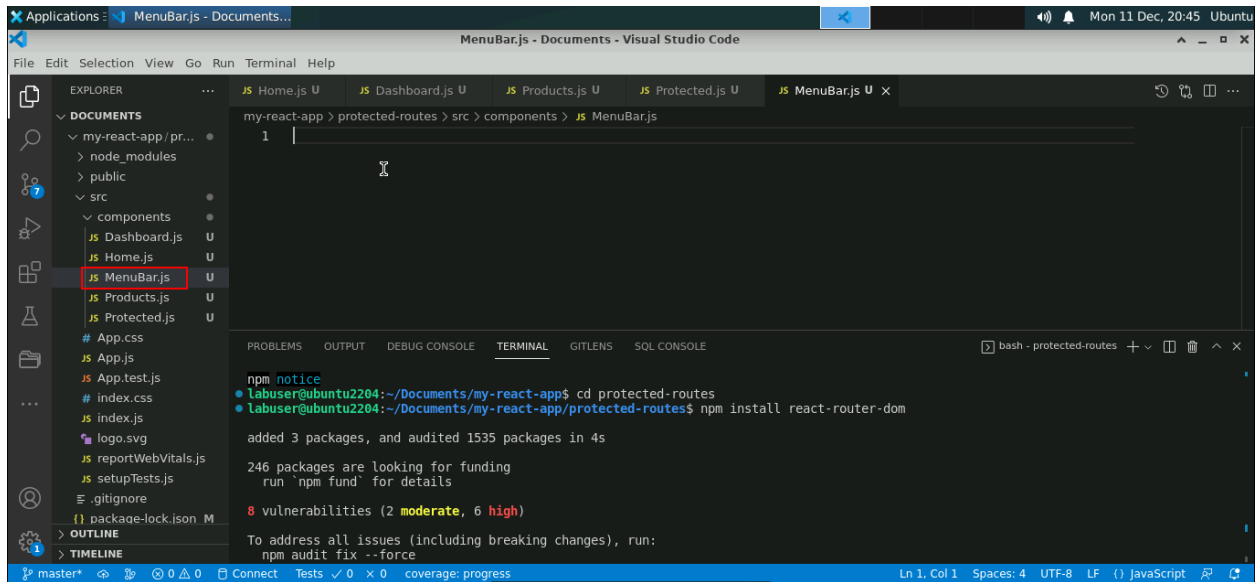
The screenshot shows the Visual Studio Code editor with the **Protected.js** file open. The file content matches the code provided in the previous block. The Explorer sidebar on the left shows the project structure, including **components** and **Protected.js**. The Terminal panel at the bottom shows the following output:

```
npm notice
labuser@ubuntu2204:~/Documents/my-react-app$ cd protected-routes
labuser@ubuntu2204:~/Documents/my-react-app/protected-routes$ npm install react-router-dom
added 3 packages, and audited 1535 packages in 4s
246 packages are looking for funding
run 'npm fund' for details
8 vulnerabilities (2 moderate, 6 high)
To address all issues (including breaking changes), run:
npm audit fix --force
```

**Note:** This file contains rules specifying that, if signed, allow movement of other components; otherwise, prohibit movement. This logic utilizes a Boolean value, **isSignedIn**.



### 3.10 Inside the **components** folder, create a new JavaScript file named **MenuBar.js**



### 3.11 Add the following code to the **MenuBar.js** file:

```
import { Link } from 'react-router-dom'
const MenuBar = () => {
  return (
    <nav className="d-flex justify-content-center">
      <div className="p-2">
        <Link to="/">Home</Link>
      </div>
      <div className="p-2">
        <Link to="/products">Products</Link>
      </div>
      <div className="p-2">
        <Link to="/dashboard">Dashboard</Link>
      </div>
    </nav>
  )
}
export default MenuBar
```

The screenshot shows the Visual Studio Code editor with the `MenuBar.js` file open. The file contains the following code:

```

1 import { Link } from 'react-router-dom'
2 const MenuBar = () => {
3   return (
4     <nav className="d-flex justify-content-center">
5       <div className="p-2">
6         <Link to="/">Home</Link>
7       </div>
8       <div className="p-2">
9         <Link to="/products">Products</Link>
10      </div>
11      <div className="p-2">
12        <Link to="/dashboard">Dashboard</Link>
13      </div>
14    </nav>
15  )
16 }
17 export default MenuBar

```

The terminal window at the bottom shows the following output:

```

npm notice
labuser@ubuntu2204:~/Documents/my-react-app$ cd protected-routes
labuser@ubuntu2204:~/Documents/my-react-app/protected-routes$ npm install react-router-dom
added 3 packages, and audited 1535 packages in 4s
246 packages are looking for funding
run `npm fund` for details
8 vulnerabilities (2 moderate, 6 high)
To address all issues (including breaking changes), run:
npm audit fix --force

```

This file provides routing links for all components, such as **Home.js**, **Product.js**, and **Dashboard.js**.

## Step 4: Implement Protected Routes

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

```

import './App.css'
import { useState } from 'react'
import { Routes, Route } from 'react-router-dom'
import MenuBar from './components/MenuBar'

import Home from './components/Home'
import Dashboard from './components/Dashboard'
import Products from './components/Products'
import Protected from './components/Protected'
export default function App() {
  const [isSignedIn, setIsSignedIn] = useState(null)
  const signin = () => {
    setIsSignedIn(true)
  }
  const signout = () => {
    setIsSignedIn(false)
  }
  return (
    <div className="container mt-2">
      <h2 className="mb-5 text-center">React Protected Routes Example</h2>

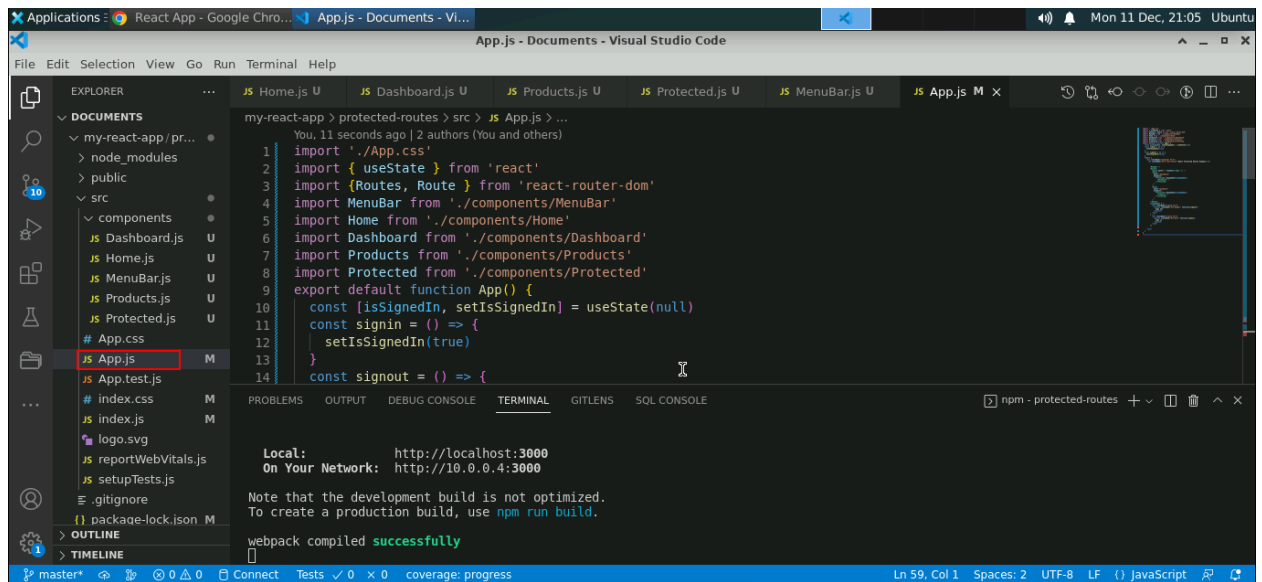
```

```

<MenuBar />
<Routes>
  <Route path="/" element={<Home />} />
  <Route
    path="/dashboard"
    element={
      <Protected isSignedIn={isSignedIn}>
        <Dashboard />
      </Protected>
    }
  />
  <Route
    path="/products"
    element={
      <Protected isSignedIn={isSignedIn}>
        <Products />
      </Protected>
    }
  />
</Routes>
{isSignedIn ? (
  <div className="d-grid mt-2">
    <button className="btn-danger" onClick={signout}>
      Sign out
    </button>
  </div>
) : (
  <div className="d-grid mt-2">
    <button className="btn-dark" onClick={signin}>
      Sign in
    </button>
  </div>
)}

</div>
)
}

```



**Note:** This file contains route rules for all components, with the **NavBar** component hosting links to each component, serving as hyperlinks for navigation. According to the rules, access to the **Dashboard** and **Product** components is permitted only when the **isSignedIn** Boolean value is **true**.

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(
  <React.StrictMode>
    <BrowserRouter>
      <App />
    </BrowserRouter>
  </React.StrictMode>
);

```

```

// 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();

```

```

1 import React from 'react';
2 import ReactDOM from 'react-dom/client';
3 import './index.css';
4 import App from './App';
5 import reportWebVitals from './reportWebVitals';
6 import {BrowserRouter} from 'react-router-dom'
7 const root = ReactDOM.createRoot(document.getElementById('root'));
8 root.render(
9   <React.StrictMode>
10     <BrowserRouter>
11       <App />
12     </BrowserRouter>
13   </React.StrictMode>
14 );
15
16 // If you want to start measuring performance in your app, pass a function
17 // to log results (for example: reportWebVitals(console.log))
18 // or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
19 reportWebVitals();

```

4.3 Add the following bootstrap CDN link in index.css page:

**@import**

**url('https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css');**

```

1 @import url('https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css');
2
3 body {
4   margin: 0;
5   font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',
6     'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
7     sans-serif;
8   -webkit-font-smoothing: antialiased;
9   -moz-osx-font-smoothing: grayscale;
10 }
11
12 code {
13   font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',
14     monospace;
15 }
16

```

**Note:** Save all the JavaScript files

## Step 5: Run the application

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

The screenshot shows the Visual Studio Code editor with the `index.css` file open. The file contains the following CSS code:

```

1 | @import url('https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css');
2 |
3 | body {
4 |   margin: 0;
5 |   font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',
6 |     'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
7 |     sans-serif;
8 |   -webkit-font-smoothing: antialiased;
9 |   -moz-osx-font-smoothing: grayscale;
10 | }
11 |
12 | code {
13 |   font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',
14 |     monospace;
15 | }
16 |

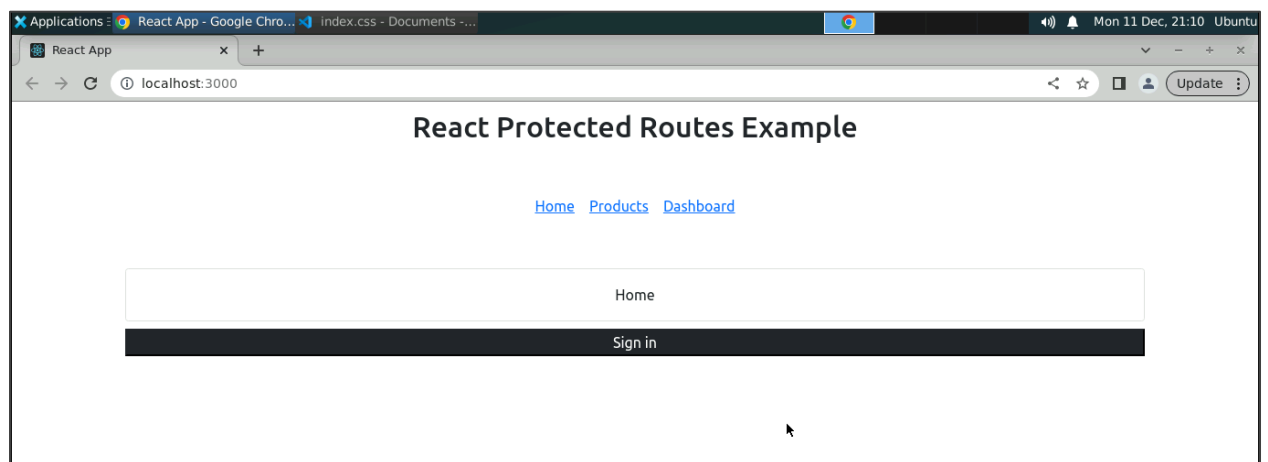
```

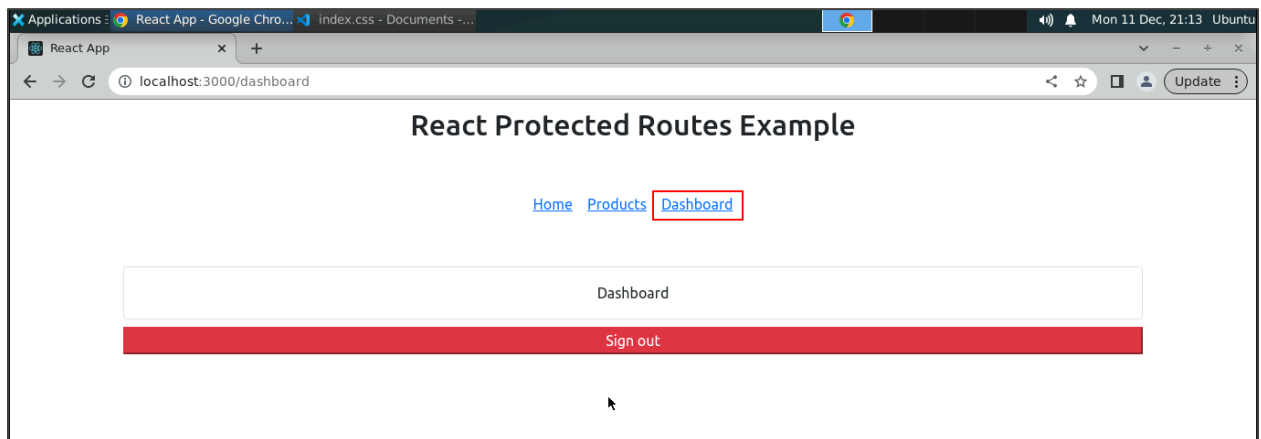
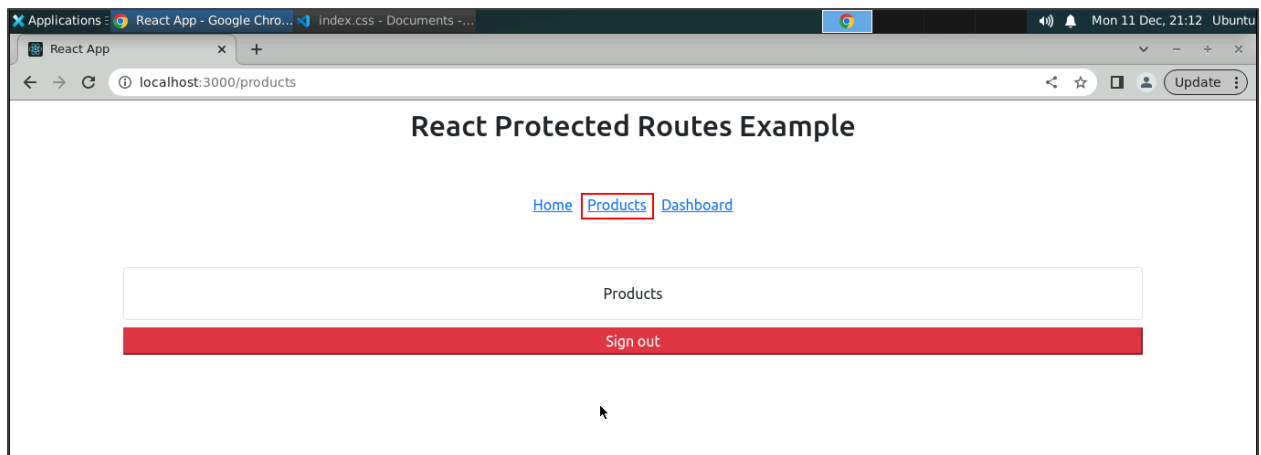
The terminal at the bottom shows the command `npm start` being executed, resulting in the output:

```

labuser@ubuntu2204:~/Documents/my-react-app/protected-routes$ npm start
> protected-routes@0.1.0 start
> react-scripts start

```





On clicking **Products** and **Dashboard**, navigation is restricted. However, after clicking the **SignIn** button, the **isSignIn** Boolean value is enabled, allowing the navigation to the **Products** or **Dashboard** components.

By following these steps, you have successfully implemented protected routes in a React application to enhance security and control navigation based on user sign-in status.