**Lesson 02 Demo 03**

**Protected router concept**

**Objective:** To demonstrate the react routing to with protected routing features. You can’t route other component without authentication.

**Tools required:** Node JS and React JS

**Prerequisites:** HTML, CSS, JavaScript ES5/ES6, Basic React Concept

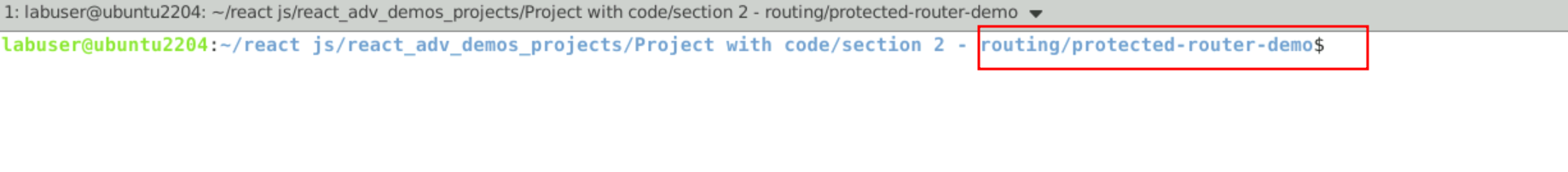
**Note** : All react js project already created with version 18.x with Sample App.js file

**Steps to be followed:**

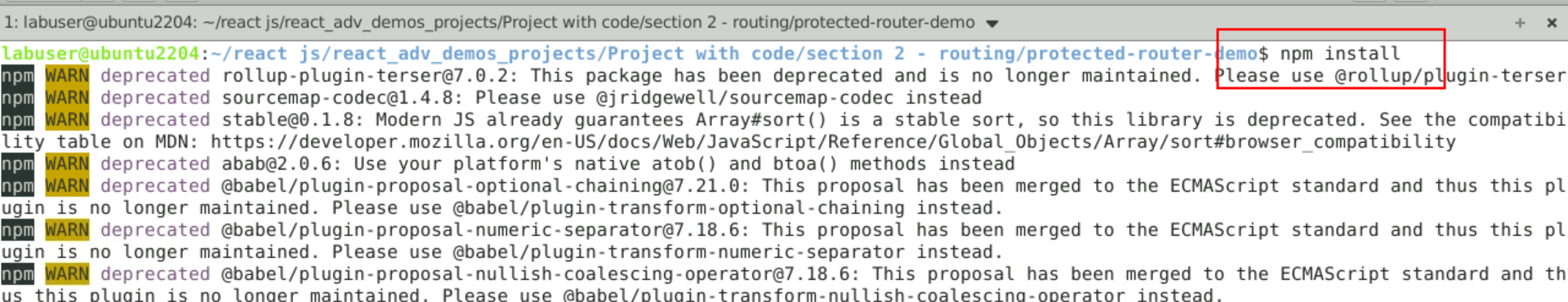
1. Set up for react js project
2. Using npm command install routing dependencies.
3. Enable BrowserRouter features in index.js file
4. Now you need to create Home.js, Dashboard.js, MenuBar.js, Products.js, Protected.js component files.
5. Now create MenuBar.js component file which contains Route link rules.
6. Add the bootstrap features to project
7. Import all component in app.js file and write the rules in App.js file
8. Now test the application

**Step 1: Set up for react js project**

1. Open a terminal window inside a React JS pre-created project ie **protected-router-demo**

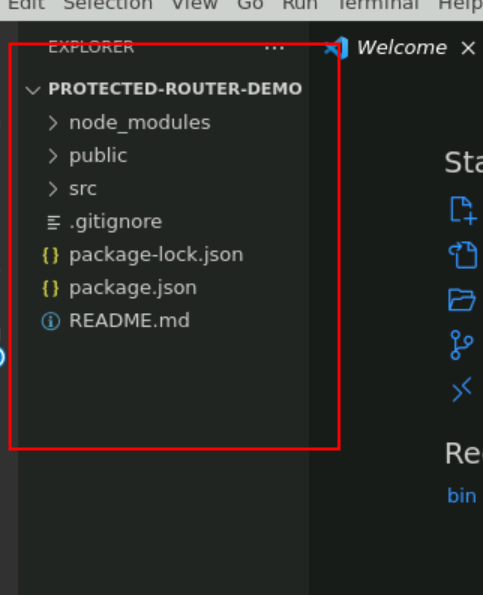


1. Now you need to run the command as **npm install.** This command helps us to installed all required dependencies mention in package.json file in local machine in the form of node\_module folder.



1. Now open **protected-router-demo** folder in VS Code Editor

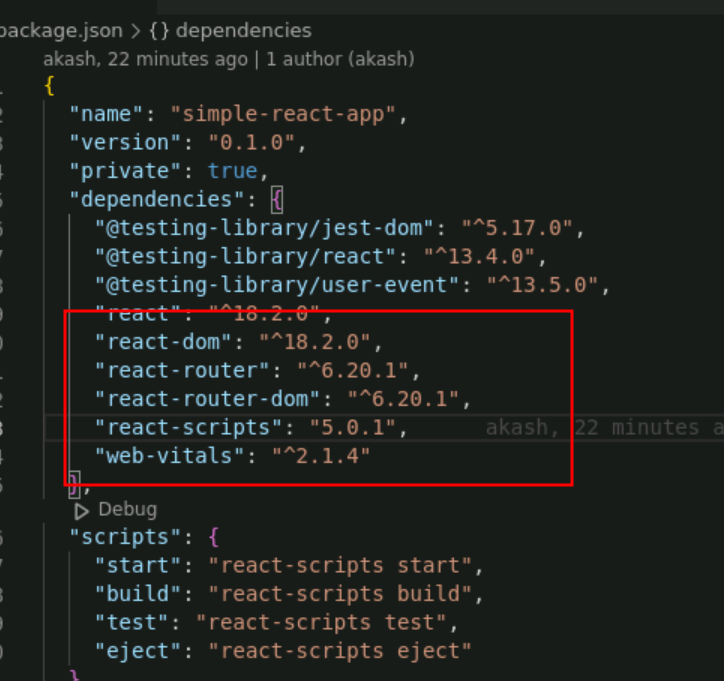
Note: short cut to open write **code .**



**Step 2:** Using npm command install routing dependencies.

1. **npm install react-router-dom**

2.2. View the react-router-dom dependencies in package.json file



**Step 3:** Enable BrowserRouter features in index.js file

3.1 Now open the index.js file and add the BrowserRouter tag which help to enable to routing features to whole application. Ie Root tag.

Index.js 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>

);

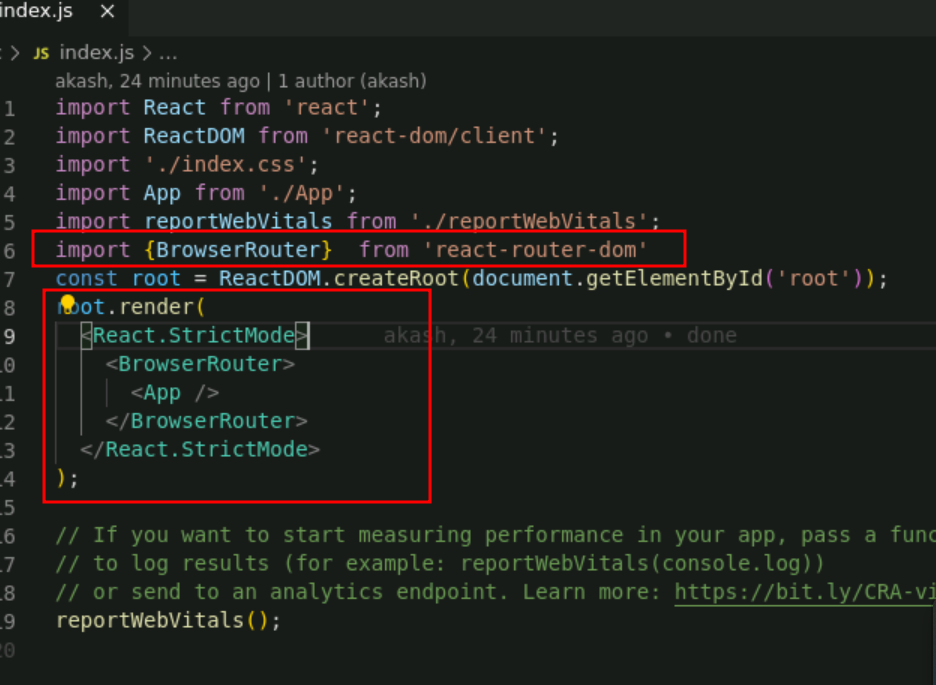
// If you want to start measuring performance in your app, pass a function

// to log results (for example: reportWebVitals(console.log))

// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals

reportWebVitals();

3.2 Screen short for the changes done in App.js file



**Step 4: Now you need to create Home, About Us and Contact Us component**

4.1 Create the Home.js component file. This file contains static contents.

import React from 'react'

function Home() {

return (

<div className="card mt-5 text-center">

<div className="card-body">Home</div>

</div>

)

}

export default Home

4.2 Create Dashboard.js component file. This file contains static contents.

import React from 'react'

function Dashboard() {

return (

<div className="card mt-5 text-center">

<div className="card-body">Dashboard</div>

</div>

)

}

export default Dashboard

4.3. Create Products.js component file. This file contains static contents.

import React from 'react'

function Products() {

return (

<div className="card mt-5 text-center">

<div className="card-body">Products</div>

</div>

)

}

export default Products

4.4. Now create Producted.js file. This file contains rules if you do signed then only allow us to move other component else not. We are doing this logic using Boolean value as isSignedIn.

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

**Step 5: Now create MenuBar.js component file which contains Route link rules**

5.1 Now create MenuBar.js. This file provide routing link for all component ie Home.js, Product.js Dashboard.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>

)

}

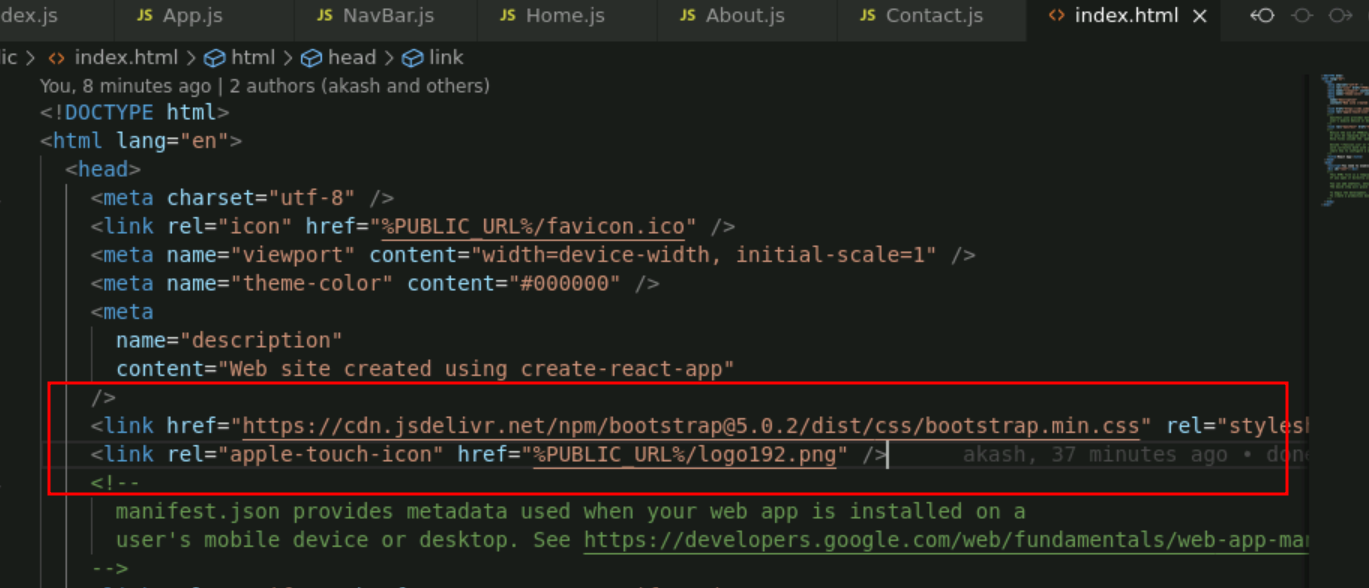
export default MenuBar

**Step 6: Add the bootstrap features to project**

6.1 Add the bootstrap CDN link in index.html page

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

6.2 Screen short of index.html page code after added link of bootstrap



**Step 7 :** Import all component in app.js file and write the rules in App.js file

7.1 App.js file : inside this file contains all component Route rules and NavBar component contains all component link ie hyperlink which help to navigate. Here rules to move Dashboard component and product component if you isSignedIn Boolean value true.

import './App.css'

import { useState } from 'react'

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

import MenuBar from './MenuBar'

import Home from './Home'

import Dashboard from './Dashboard'

import Products from './Products'

import Protected from './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>

)

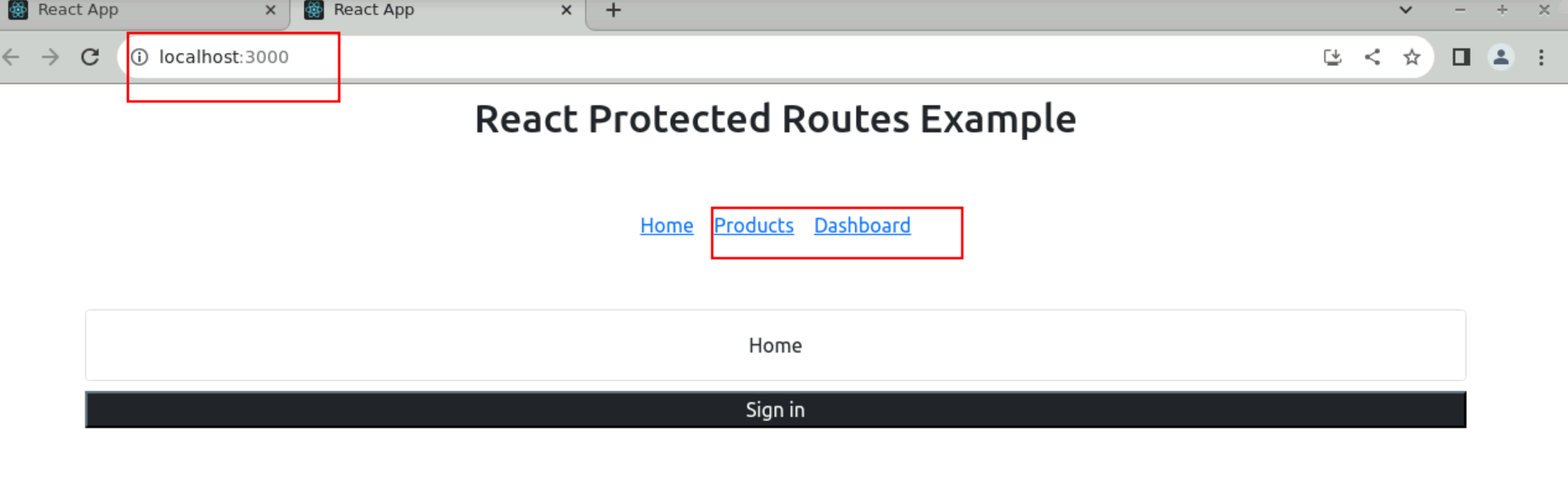
}

**Step 8. Now test the application**

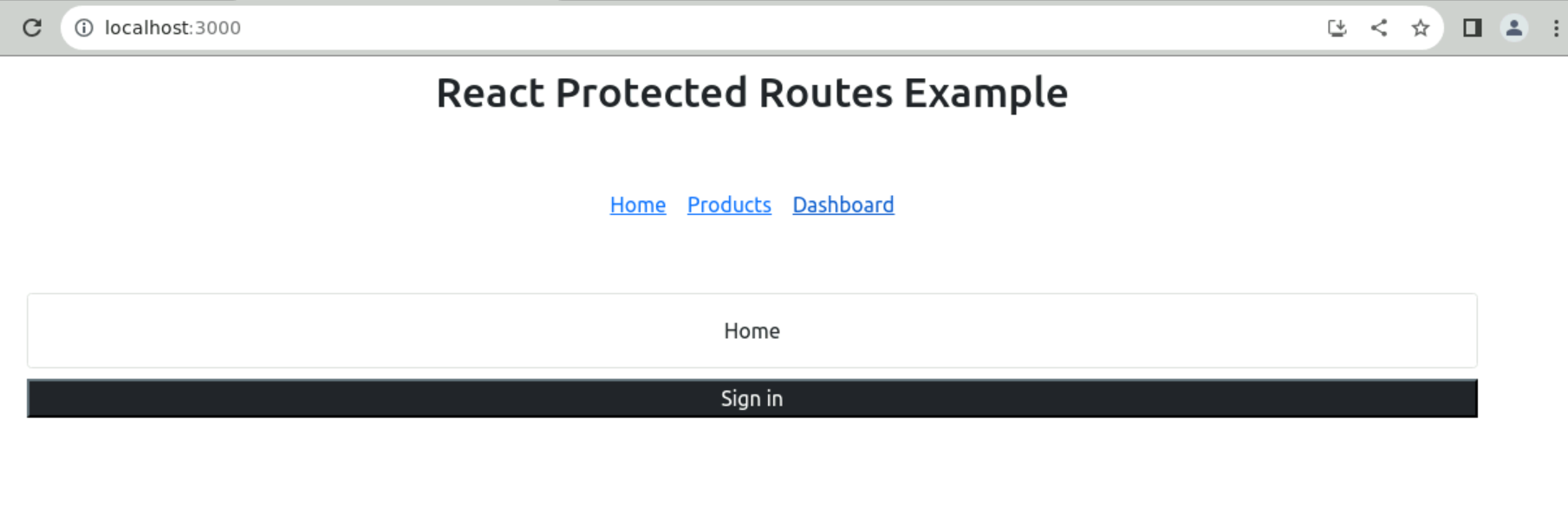
**8.1 npm start**



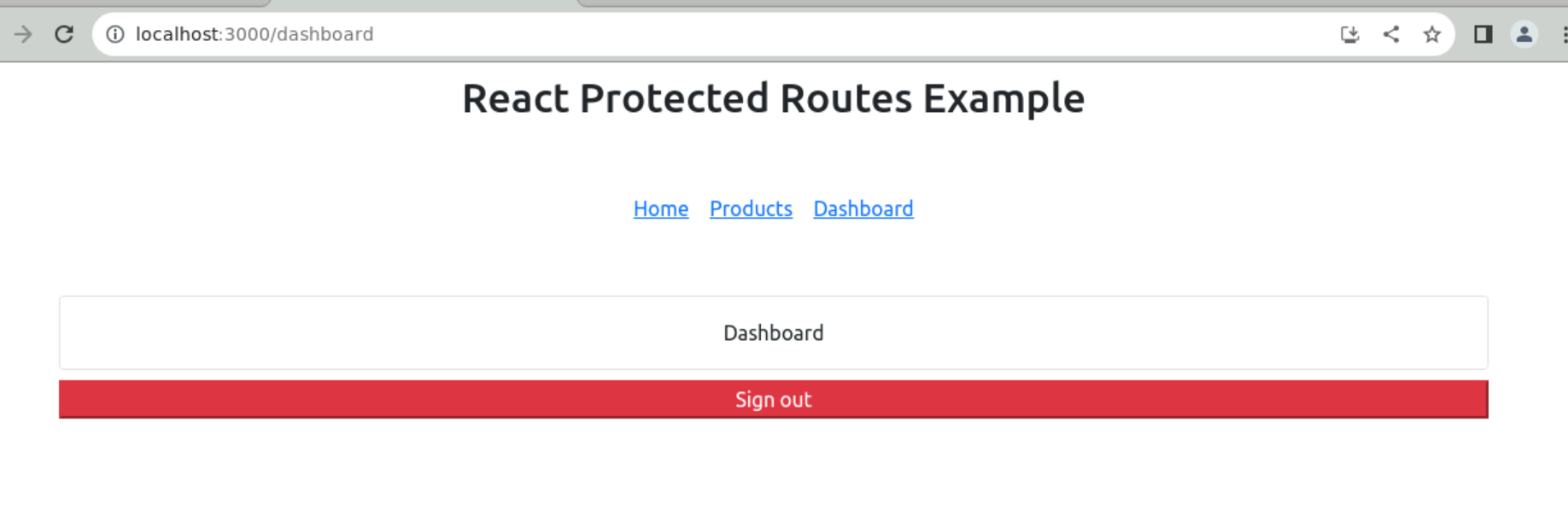
**8.2 See the output on browser**



**8.3 Now you can click on Products and Dashboad. You can’t navigate.**



**8.4 But after click on SignIn button you can enable isSignIn Boolean value. Then only you can navigate to Products or Dashboard components.**



**8.5 After Sign out you can’t move.**

