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

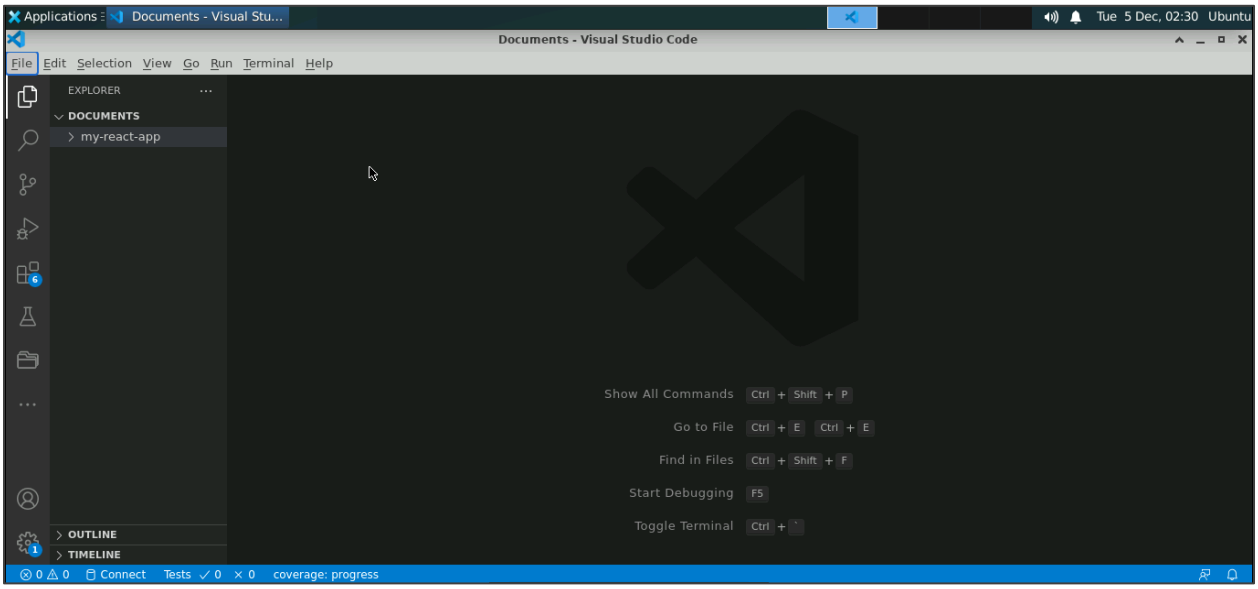
Prerequisites: A basic understanding of Protected Routes in React Router

Steps to be followed:

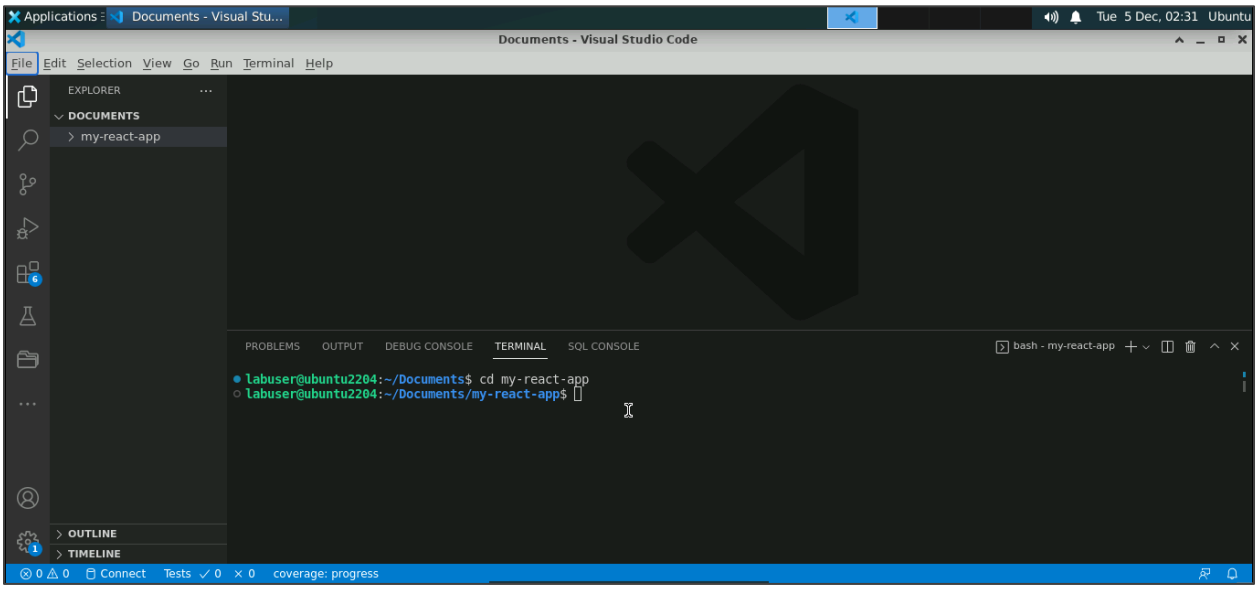
1. Set up a new React application
2. Install React Router
3. Create component files
4. Implement Protected Routes
5. Run the application

Step 1: Set up a new React application

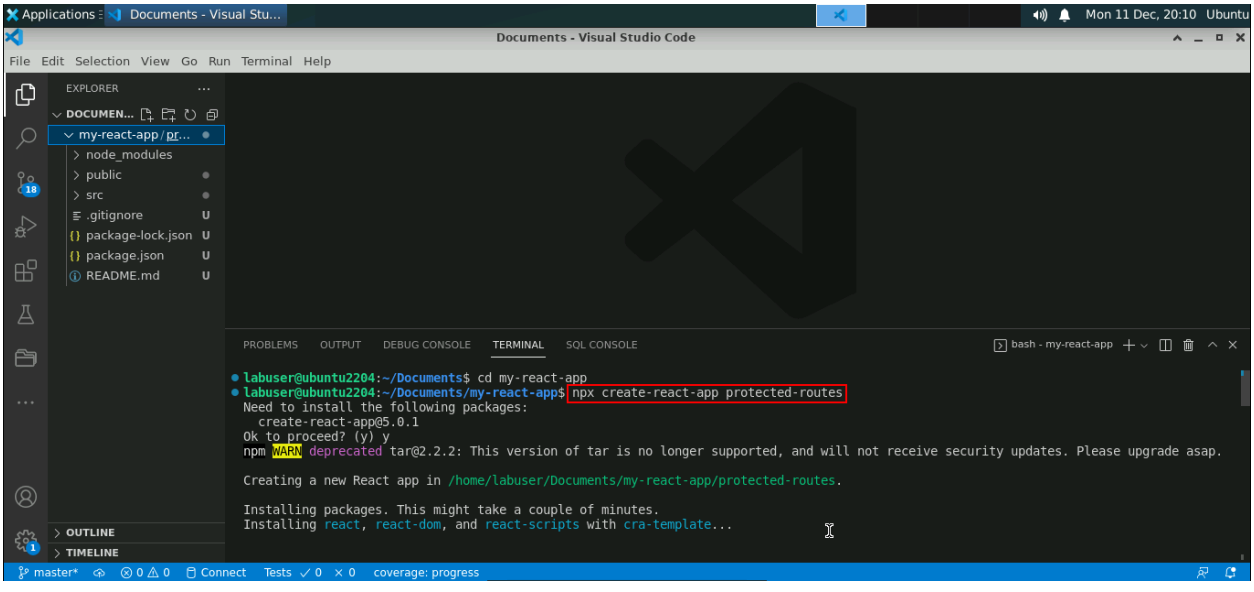
1.1 Open **Visual Studio Code** and create a new folder named **my-react-app**



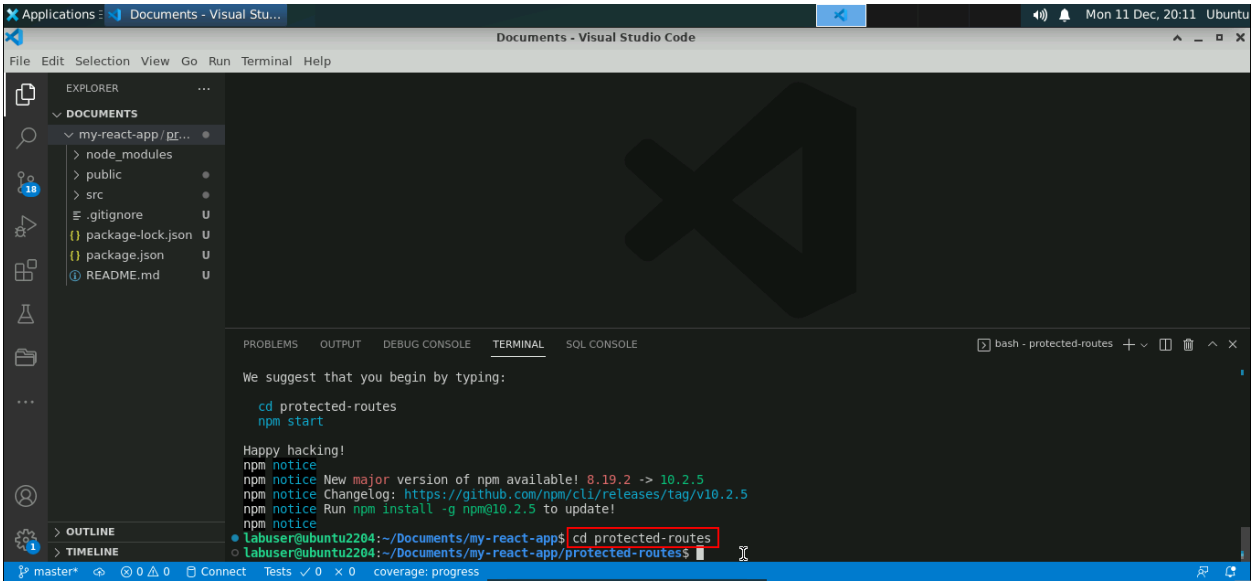
1.2 Open the terminal and navigate to the **my-react-app** folder using the following command:
cd my-react-app



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

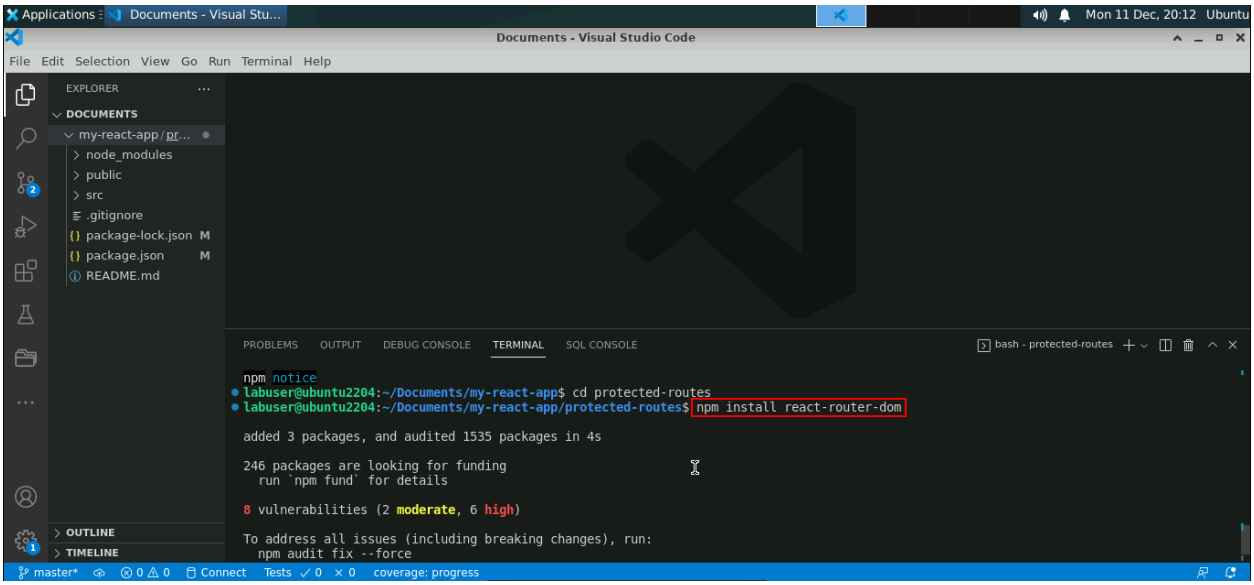


1.4 Navigate to the **protected-routes** folder using the following command:
cd protected-routes



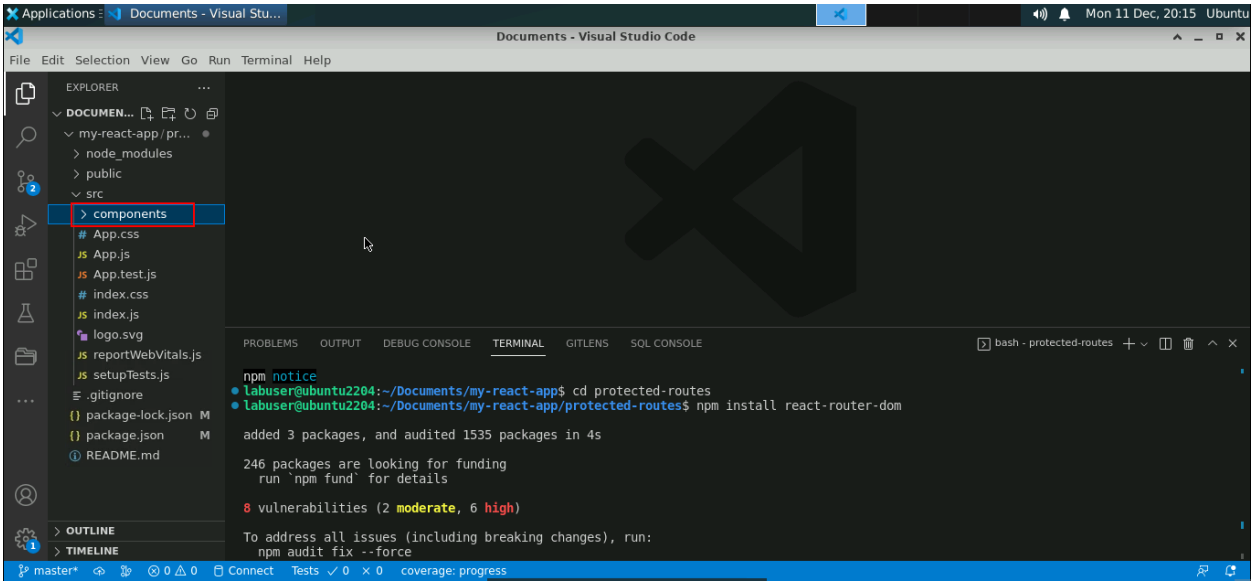
Step 2: Install React Router

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

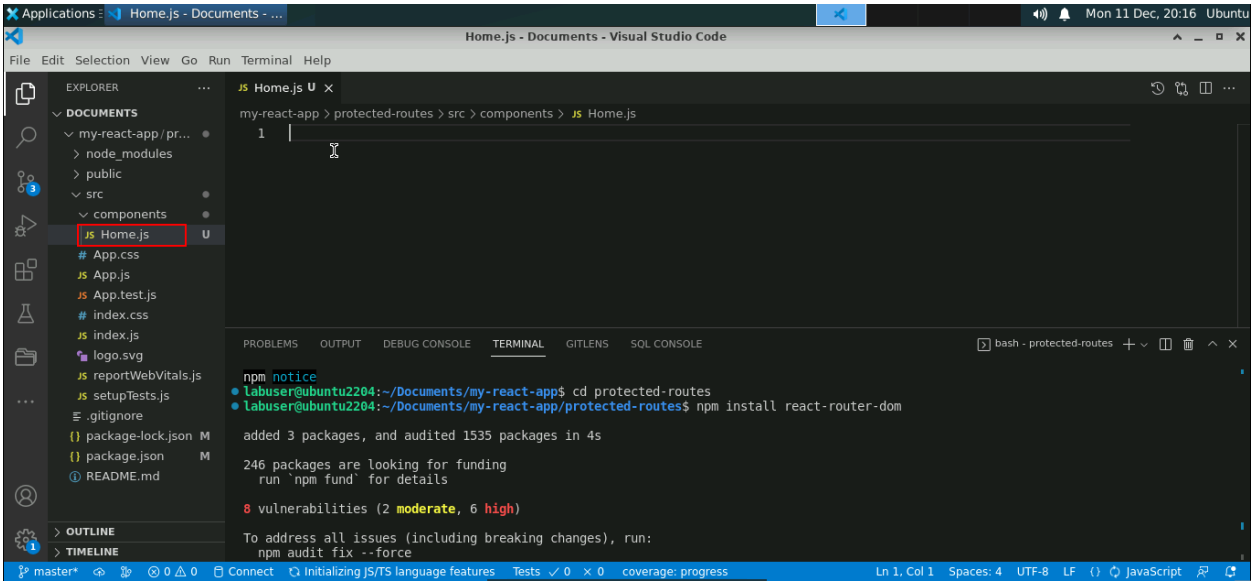


Step 3: Create component files

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

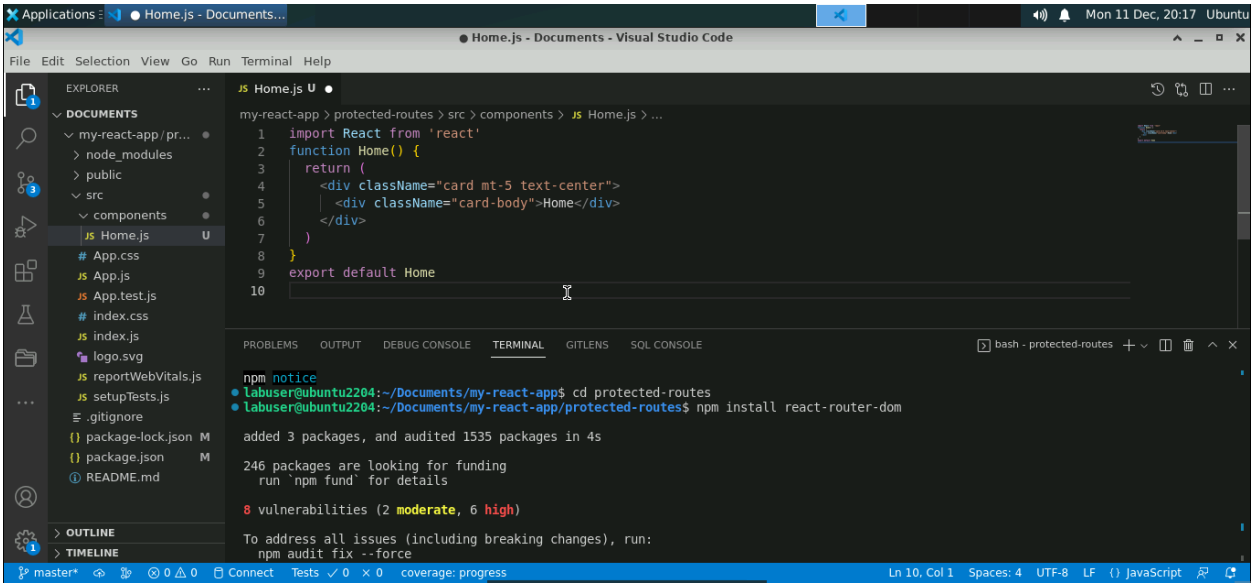


3.2 Inside the **components** folder, create a new JavaScript file named **Home.js**



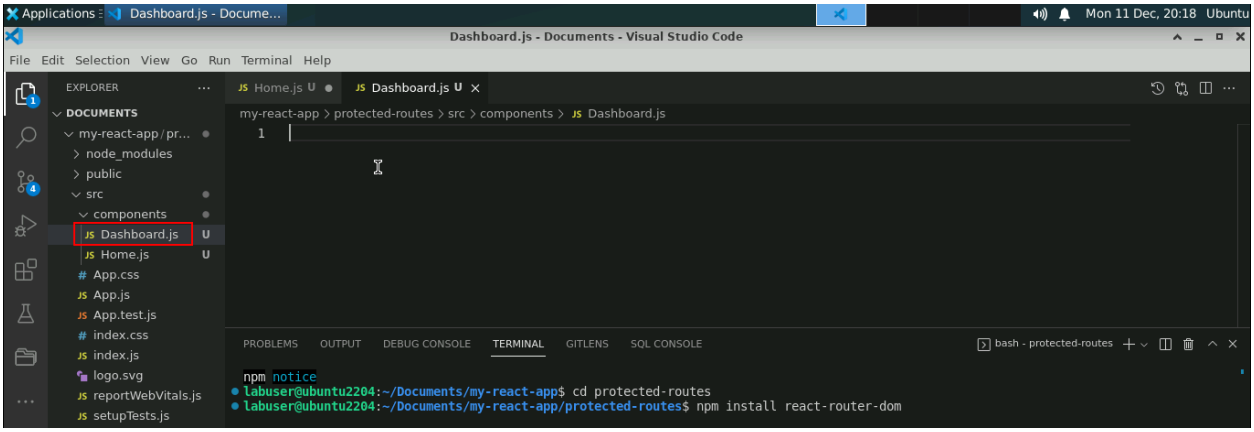
3.3 Add the following code to the **Home.js** file:

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



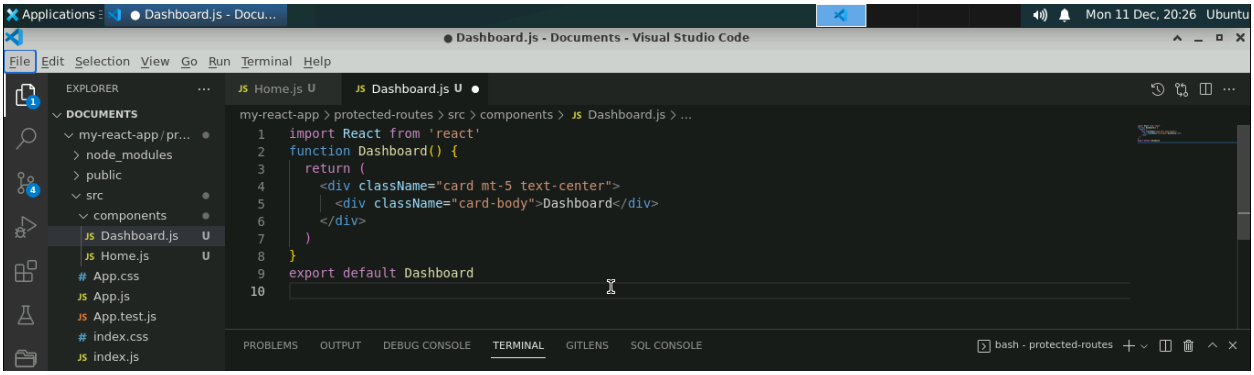
Note: Save the file

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

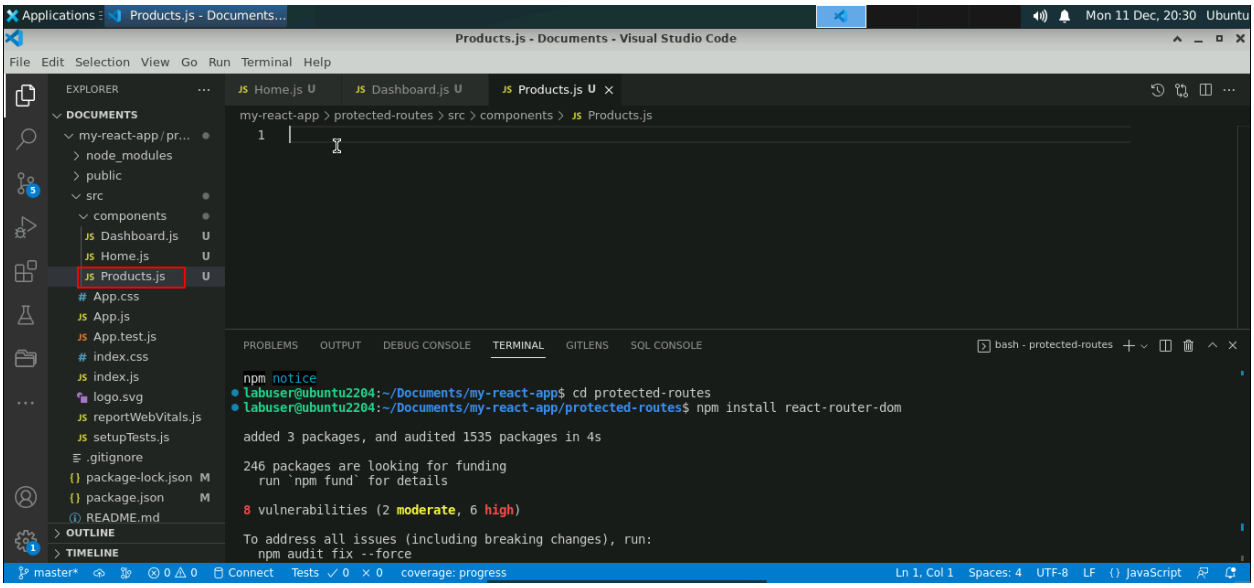


3.5 Add the following code to the **Dashboard.js** file:

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



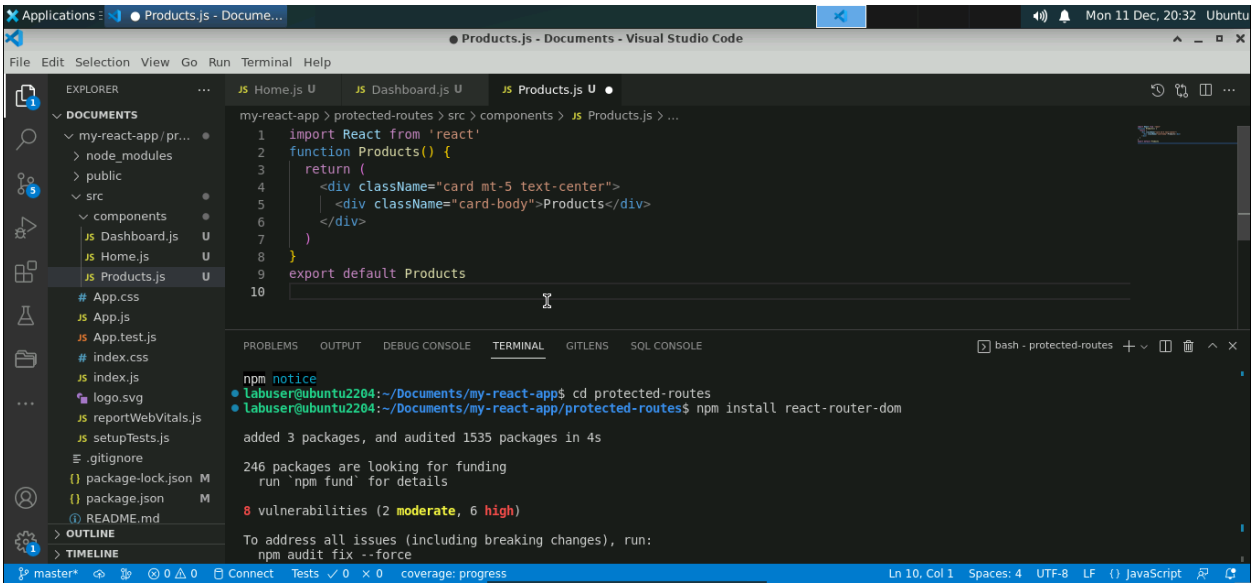
3.6 Inside the **components** folder, create a new JavaScript file named **Products.js**



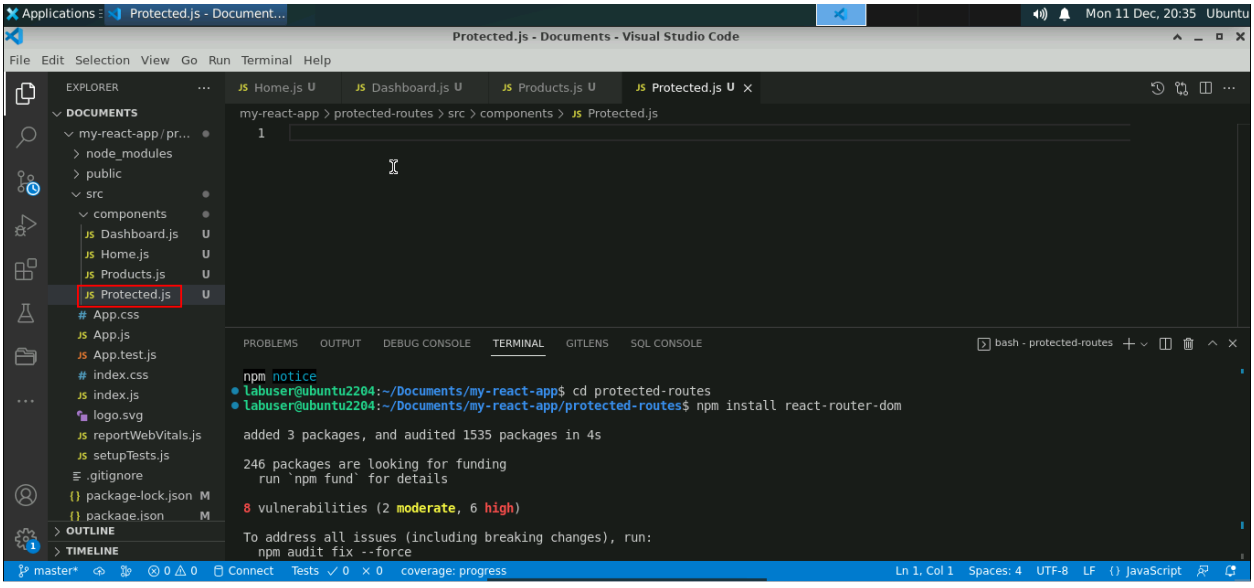
3.7 Add the following code to the **Products.js** file:

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

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

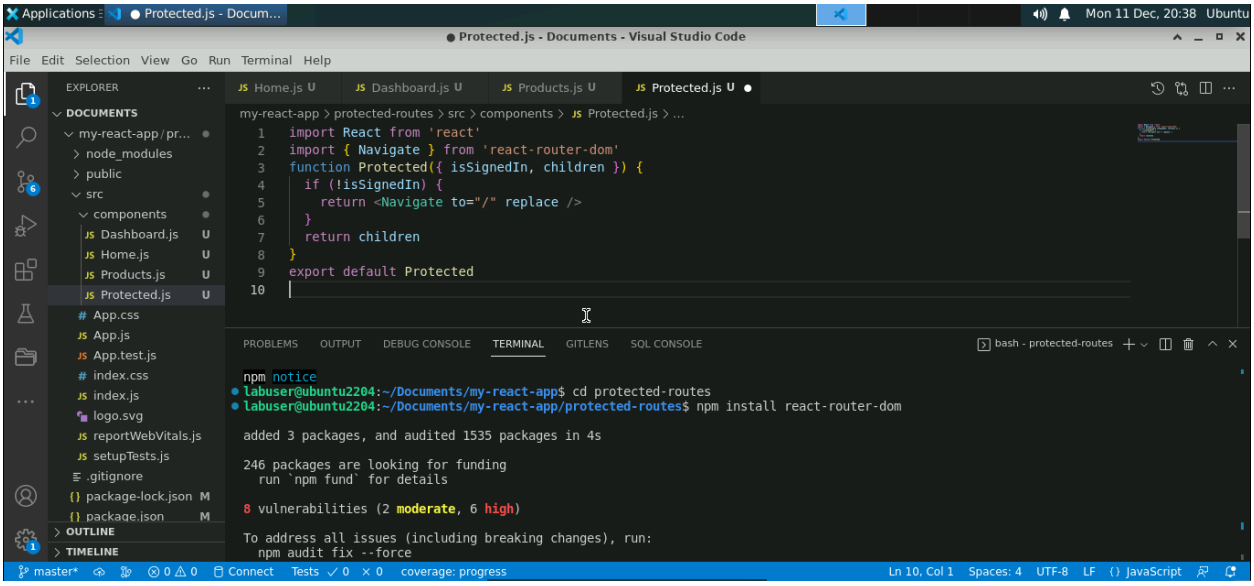


3.8 Inside the **components** folder, create a new JavaScript file named **Protected.js**



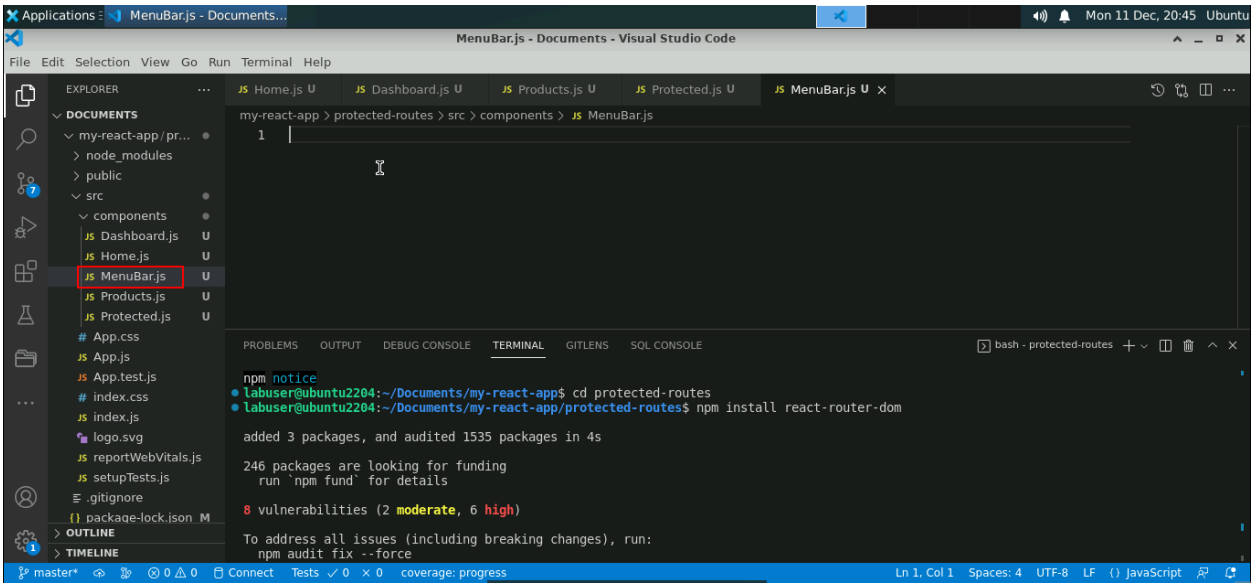
3.9 Add the following code to the **Protected.js** file:

```
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
```



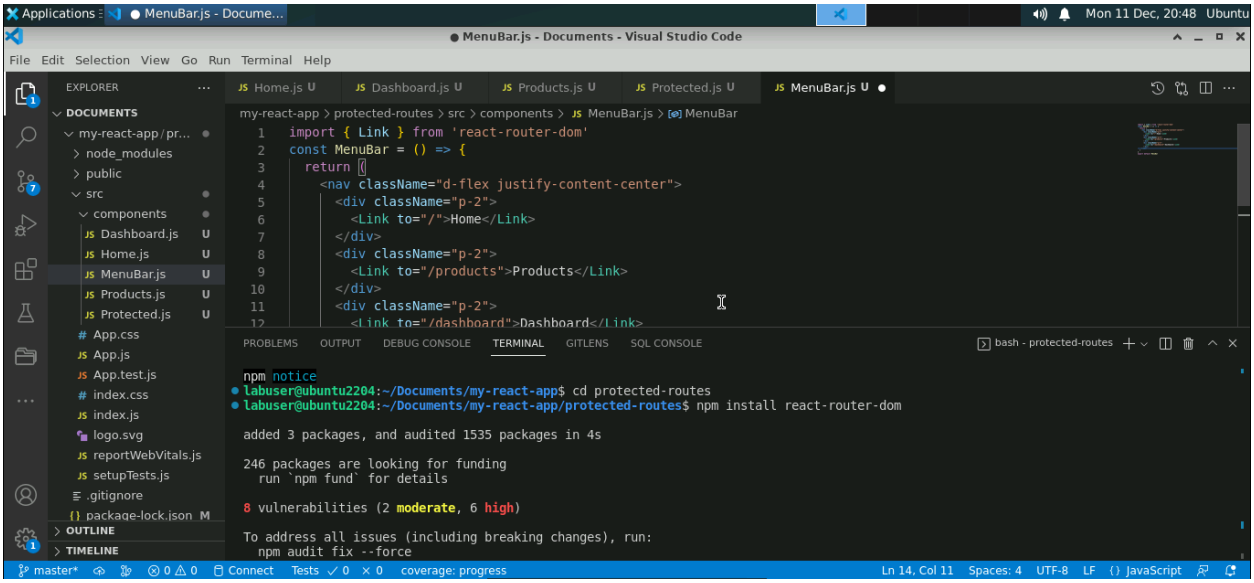
Note: This file contains rules specifying that, if signed, allow movement of other components; otherwise, prohibit movement. This logic utilizes a Boolean value, **isSignedIn**.

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
```



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:

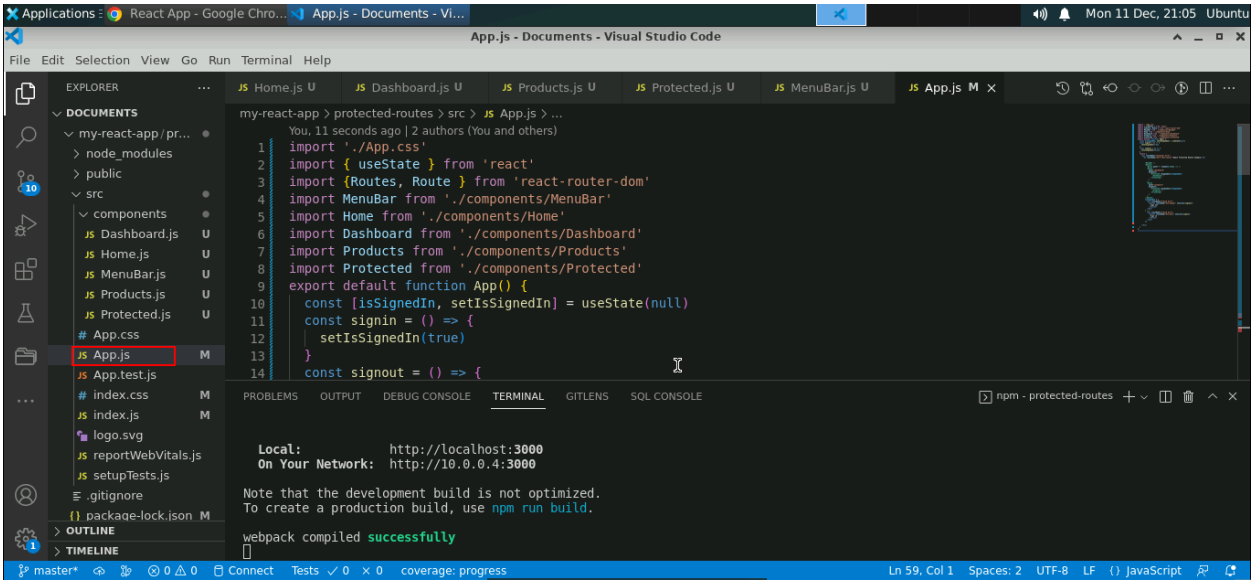
```
import './App.css'
import { useState } from 'react'
import {Routes, Route } from 'react-router-dom'
import MenuBar from './components/MenuBar'

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>
  )
}
```

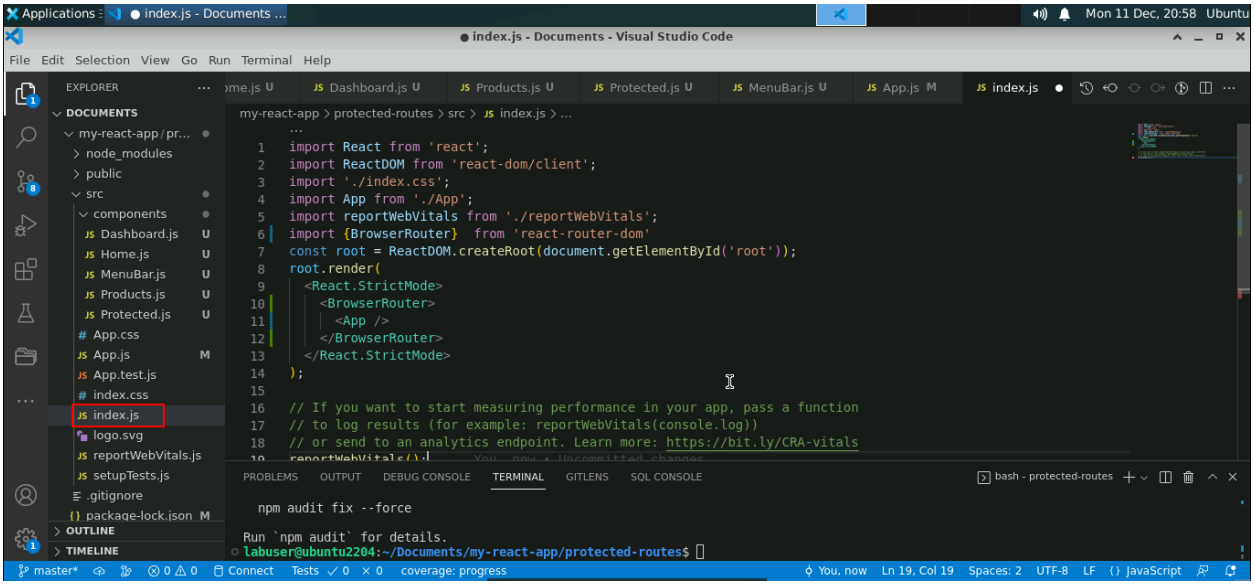


Note: This file contains route rules for all components, with the **NavBar** component hosting links to each component, serving as hyperlinks for navigation. According to the rules, access to the **Dashboard** and **Product** components is permitted only when the **isSignedIn** Boolean value is **true**.

4.2 Replace the existing code in the **index.js** file with the following code:

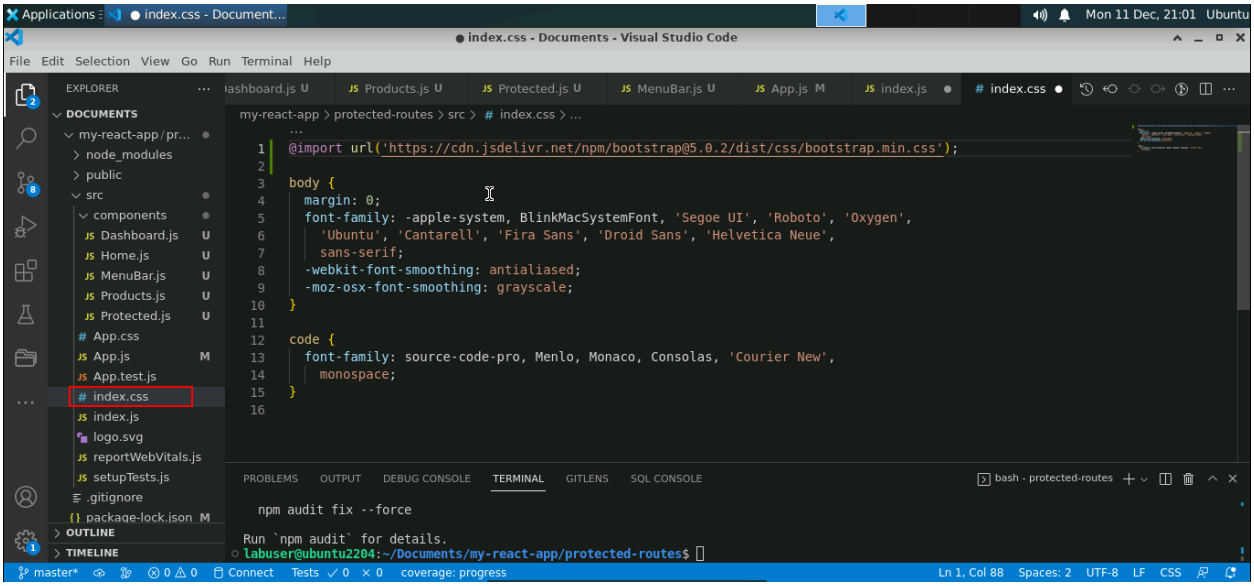
```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';
import {BrowserRouter}  from 'react-router-dom'
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
 <React.StrictMode>
  <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();
```



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');

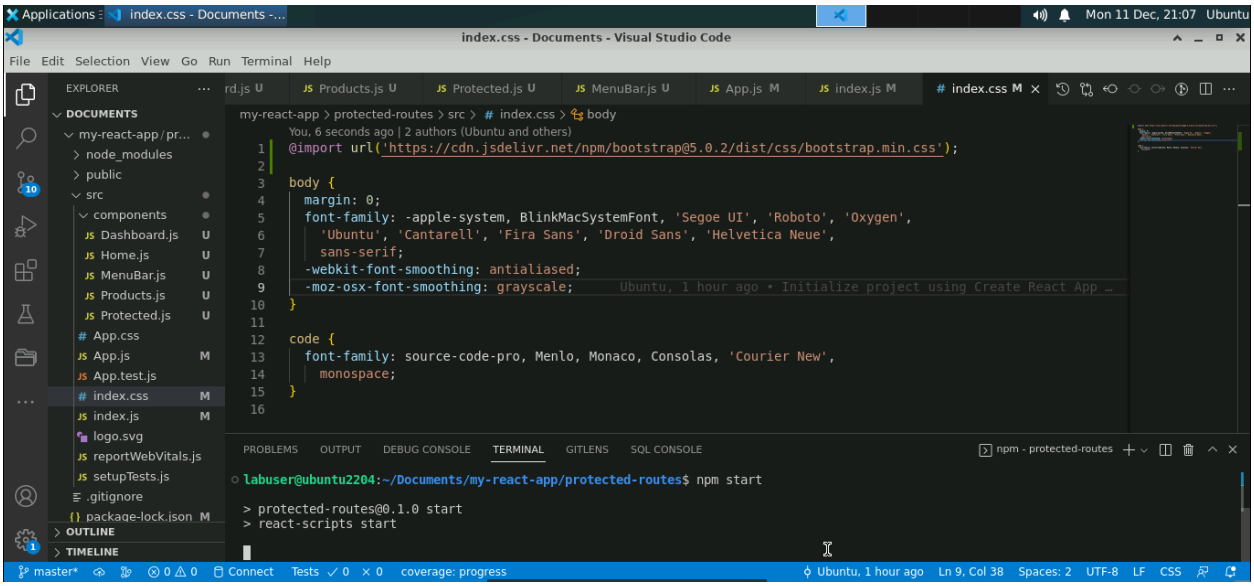


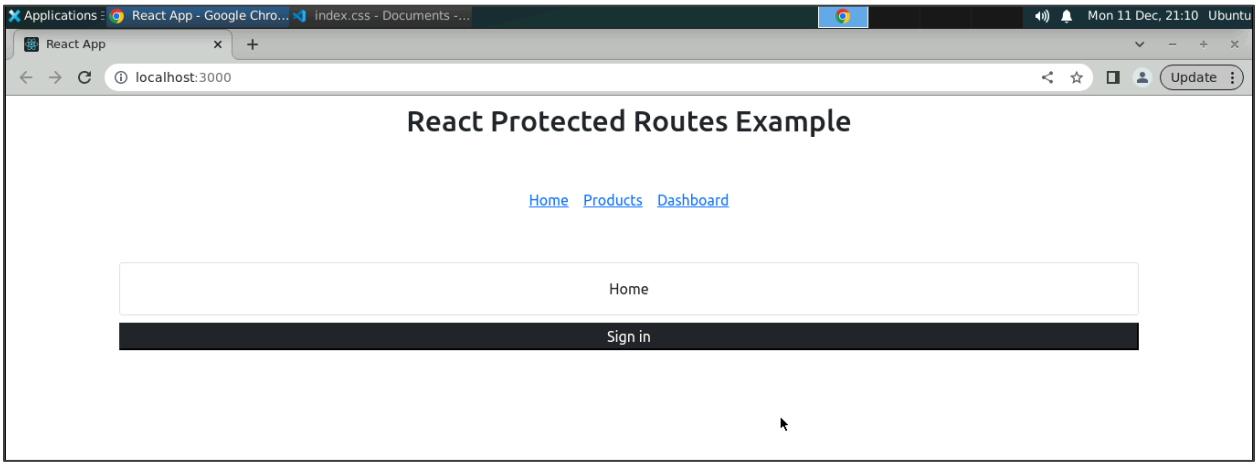
Note: Save all the JavaScript files

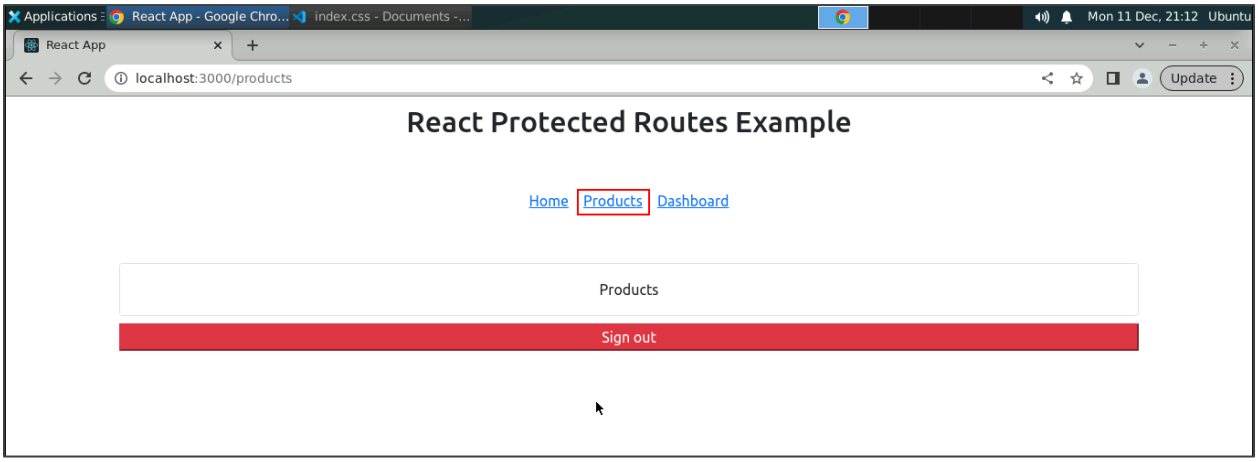
Step 5: Run the application

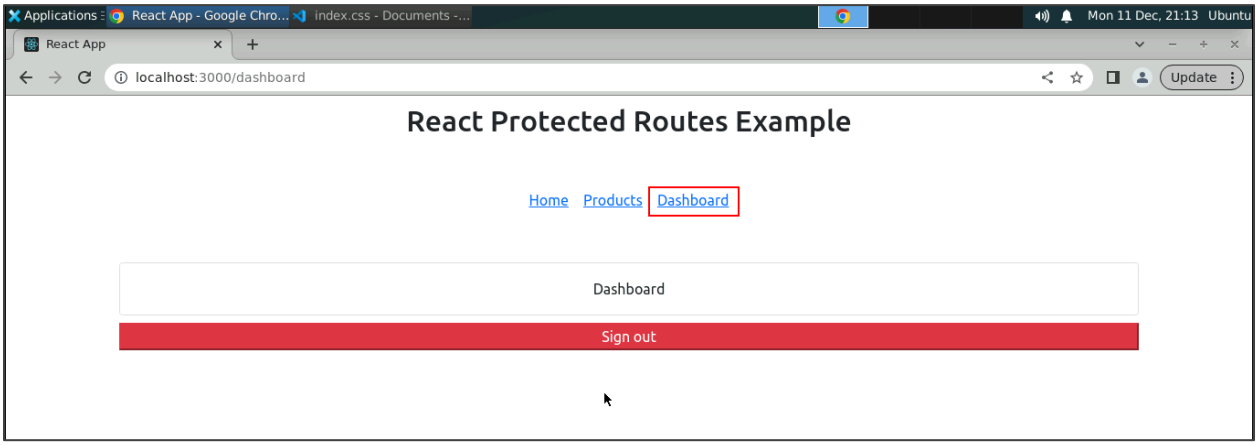
5.1 Start the React development server using the following command:

npm 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.