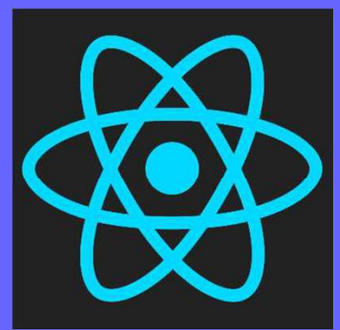
1

React Router

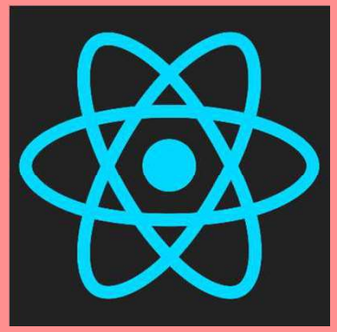


Introduction to React Routing

- It is a fully featured client and server side routing library for react, the library helps create and navigate between different urls that make up your web application.
- It also provides unique urls for different components in the app and makes the ui easily shareable with other users
- If you're building a medium too large scale react app react router is a must have package

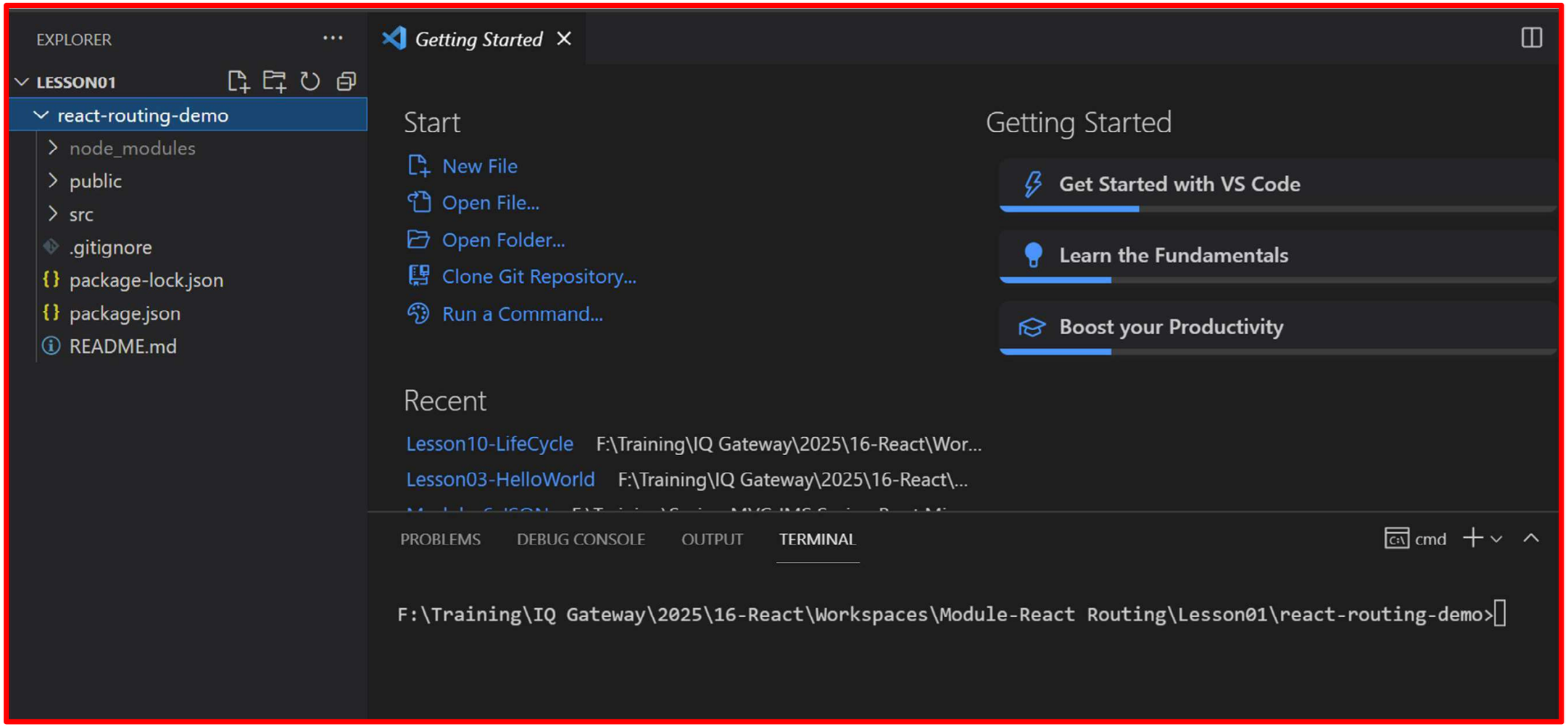
Focus on

1. Configuring Routes
2. Navigating on Button Click
3. Navigating Programmatically
4. Dynamic Routes
5. Nested Routes
6. Route Parameters

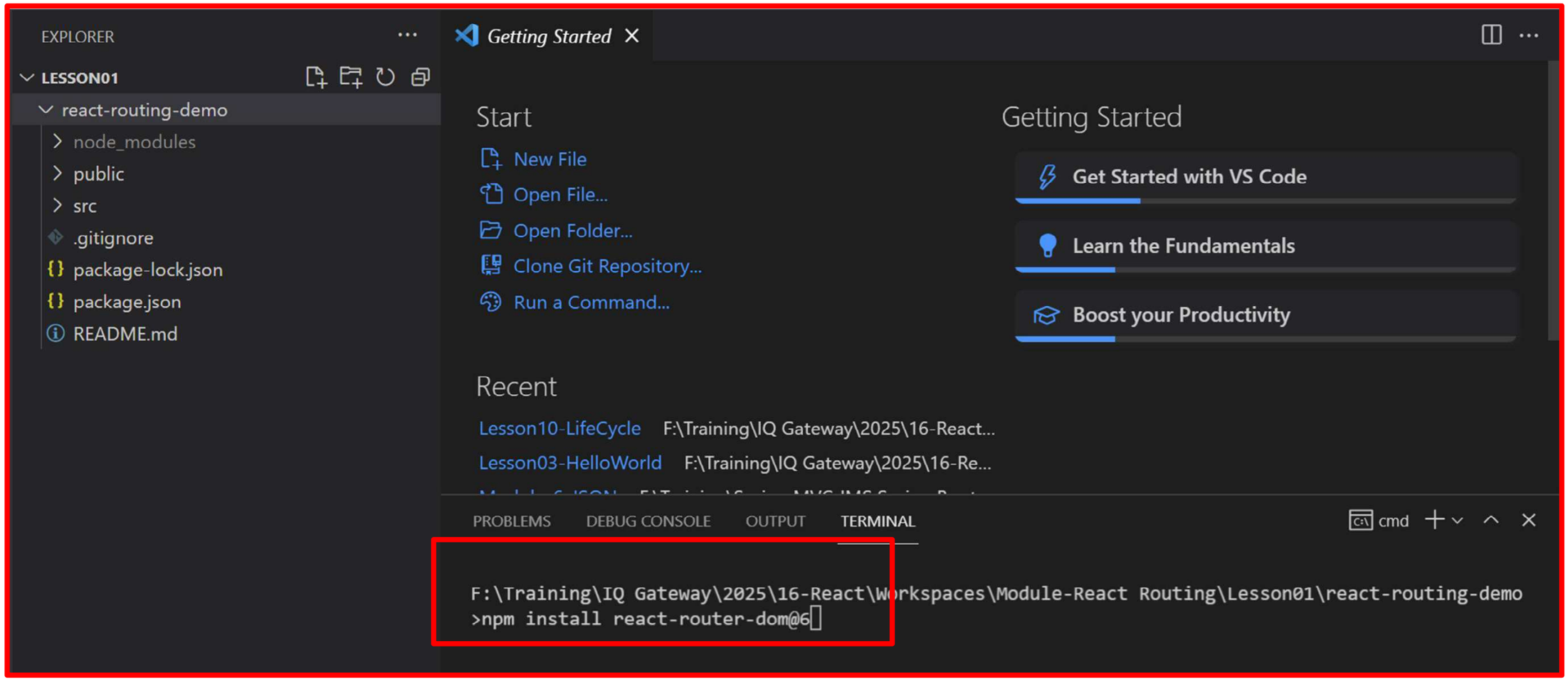


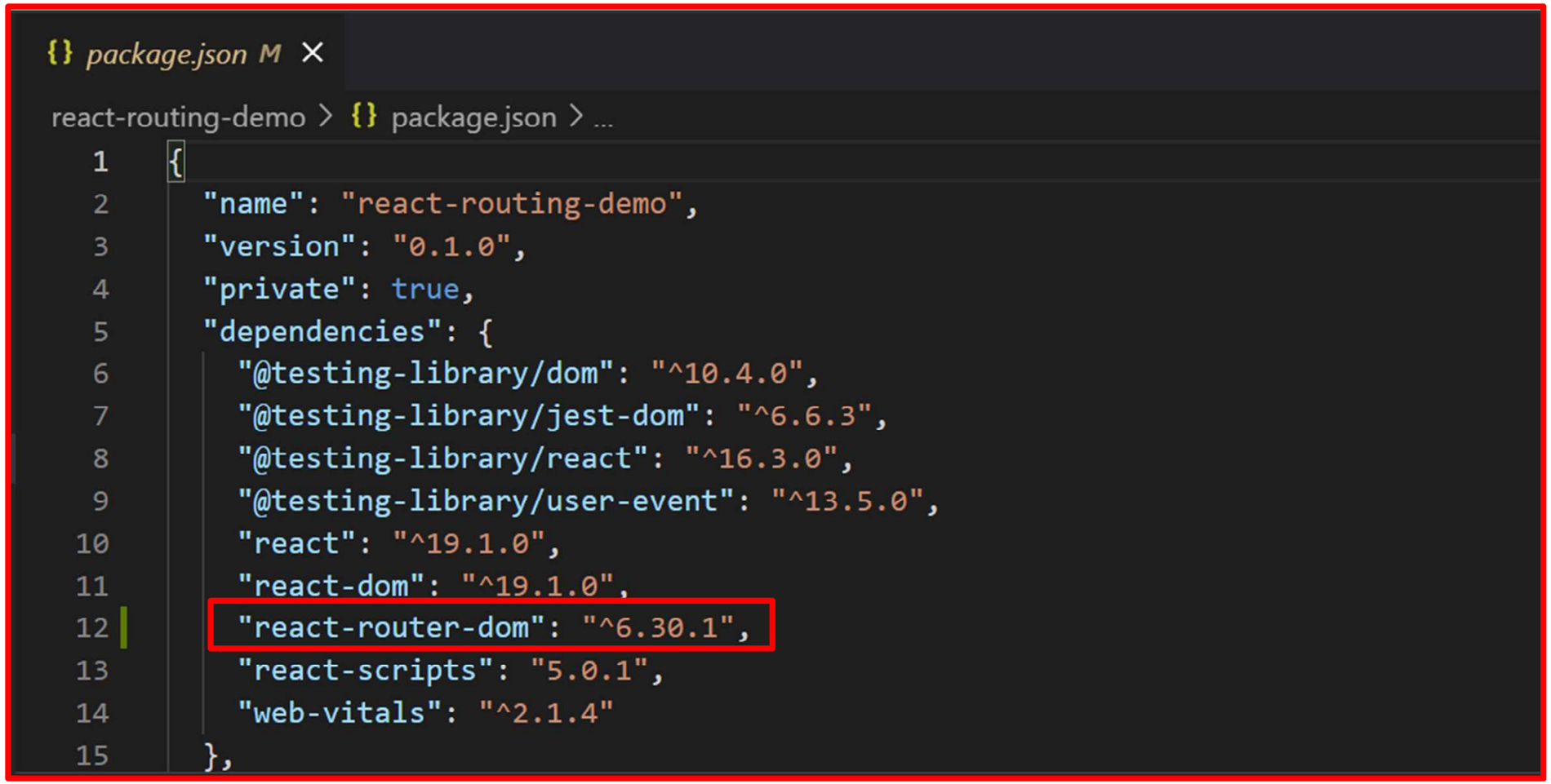
Installation of React Routing

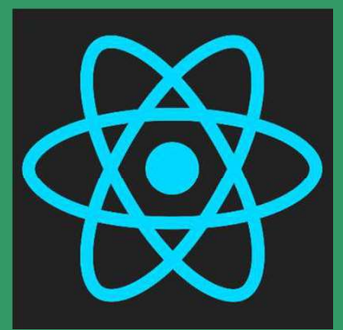
Step1 : Create React App



Step 2 : Install React Router Package




{} package.json M
react-routing-demo > {} package.json > ...
{
"name": "react-routing-demo",
"version": "0.1.0",
"private": true,
"dependencies": {
"@testing-library/dom": "^10.4.0",
"@testing-library/jest-dom": "^6.6.3",
"@testing-library/react": "^16.3.0",
"@testing-library/user-event": "^13.5.0",
"react": "^19.1.0",
"react-dom": "^19.1.0",
"react-router-dom": "^6.30.1",
"react-scripts": "5.0.1",
"web-vitals": "^2.1.4"
},

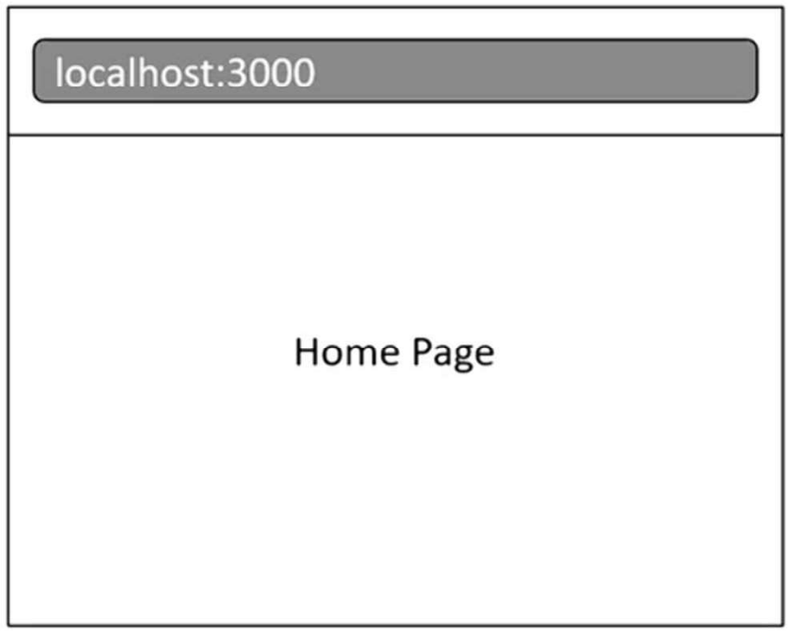


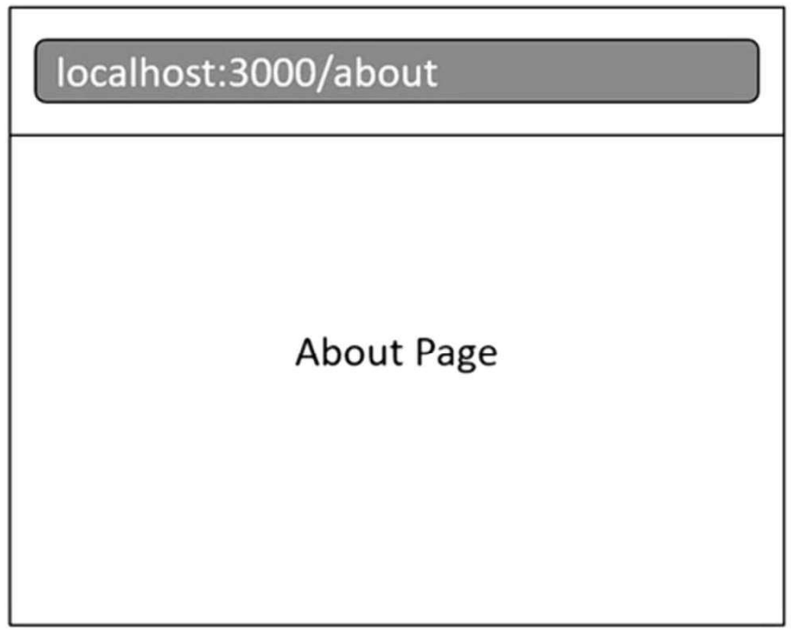
Configuring Routes

Configuring Routes

Scenario : We're going to set up two routes the first route is the home route

- If the user navigates to **localhost:3000** they should be able to see the **home page**
- If the visit **localhost:3000/about** they should be able to see the **about page**





We render the app component to the DOM

index.js

react-routing-demo > src > index.js > ...

```
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
7   const root = ReactDOM.createRoot(document.getElementById('root'));
8   root.render(
9     <React.StrictMode>
10      <App />
11    </React.StrictMode>
12  );
13
14  // If you want to start measuring performance in your app, pass a function
15  // to log results (for example: reportWebVitals(console.log))
16  // or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
17  reportWebVitals();
18
```

App Component Renders

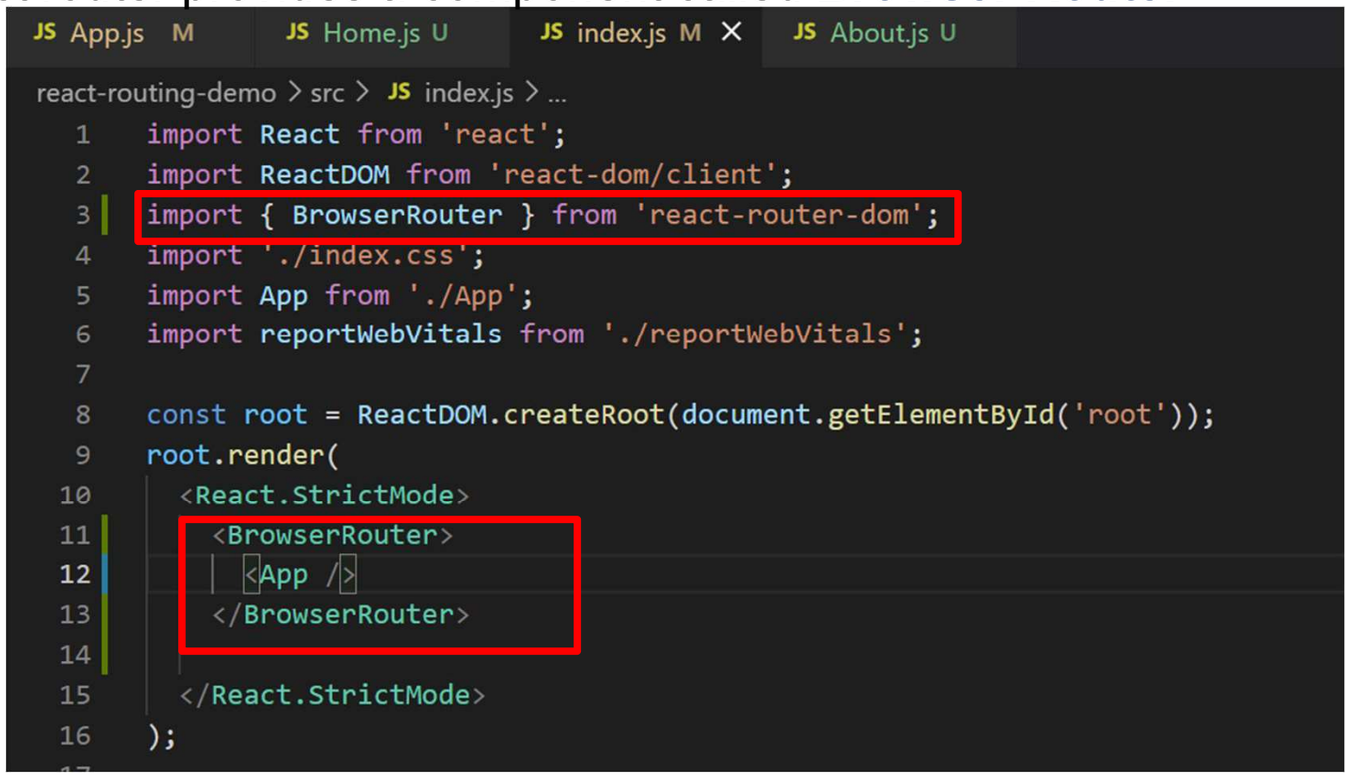
App.js

react-routing-demo > src > App.js > ...

```
1  import logo from './logo.svg';
2  import './App.css';
3
4  function App() {
5    return (
6      <div className="App">
7        <header className="App-header">
8          <img src={logo} className="App-logo" alt="logo" />
9          <p>
10           Edit <code>src/App.js</code> and save to reload.
11         </p>
12         <a
13           className="App-link"
14           href="https://reactjs.org"
15           target="_blank"
16           rel="noopener noreferrer"
17         >
18           Learn React
19         </a>
20       </header>
21     </div>
22   );
23 }
24
25 export default App;
```

Configuring

- The first step to configuring routes with react router is to connect the url in the browser with our react application
- For that react router provides a component called **Browser Router**



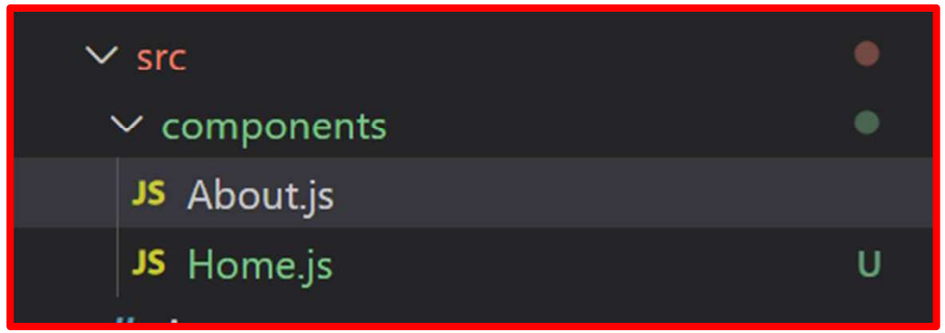
- This allows us to do is use all the features from react router within the app component tree

Let's Configure the routes in App.js

App.js 1, M

react-routing-demo > src > App.js > ...

```
1
2   function App() {
3     return (
4
5     );
6   }
7
8   export default App;
9
```



App.js 1, M | Home.js U | About.js U

react-routing-demo > src > components > Home.js > ...

```
1   import React from 'react'
2
3   export const Home = () => {
4       return (
5           <div>
6               Home Page
7           </div>
8       )
9   }
10
```

App.js 1, M | Home.js U | About.js U

react-routing-demo > src > components > About.js > About

```
1   import React from 'react'
2
3   export const About = () => {
4       return (
5           <div>
6               About Page
7           </div>
8       )
9   }
```

Cont …

- For the route configuration we need two components from react router

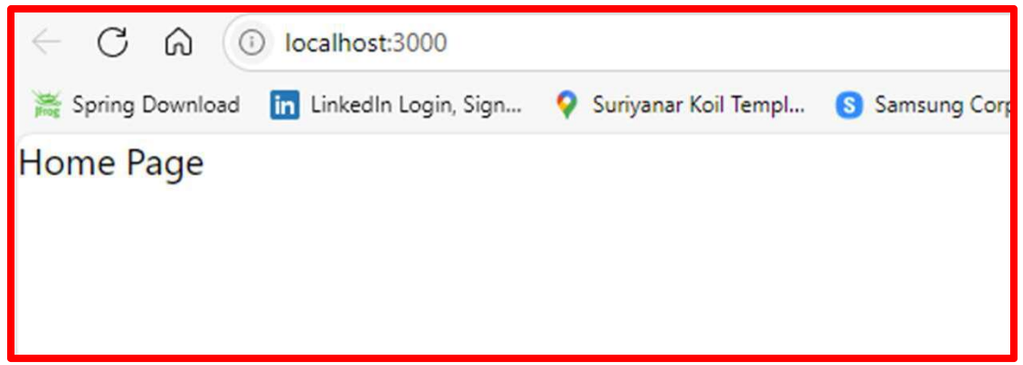
App.js > ...

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

App.js M | Home.js U | About.js U

react-routing-demo > src > App.js > App

```
1  import { Routes, Route } from 'react-router-dom';
2  import { Home } from './components/Home'
3
4  function App() {
5    return (
6      <Routes>
7        <Route path="/" element={<Home />}></Route>
8      </Routes>
9    );
10 }
11
12 export default App;
13
```

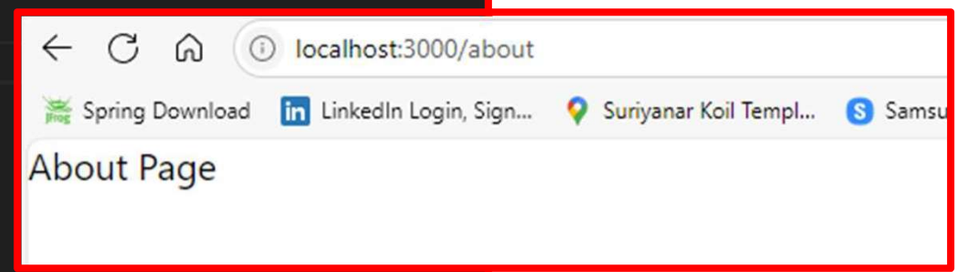


Second Route

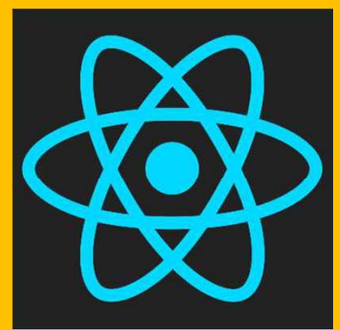
JS App.js M × JS Home.js U JS index.js M JS About.js U

react-routing-demo > src > JS App.js > ...

```
1  import { Routes, Route } from 'react-router-dom';
2  import { Home } from './components/Home'
3  import { About } from './components/About'
4
5  function App() {
6    return (
7      <Routes>
8        <Route path="/" element={<Home />}></Route>
9
10       <Route path="about" element={<About />}></Route>
11
12     </Routes>
13   );
14 }
15
16 export default App;
17
```



- This is pretty much how you configure Routes with React Router

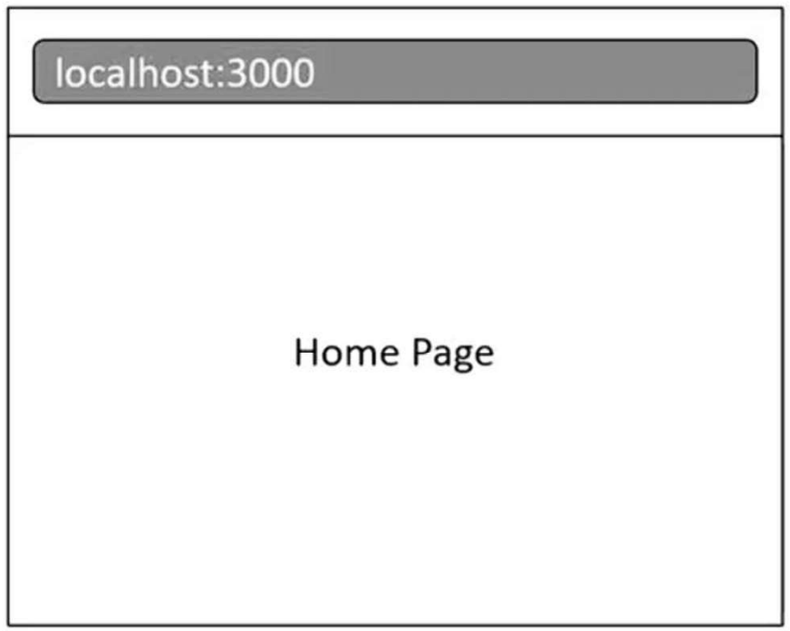


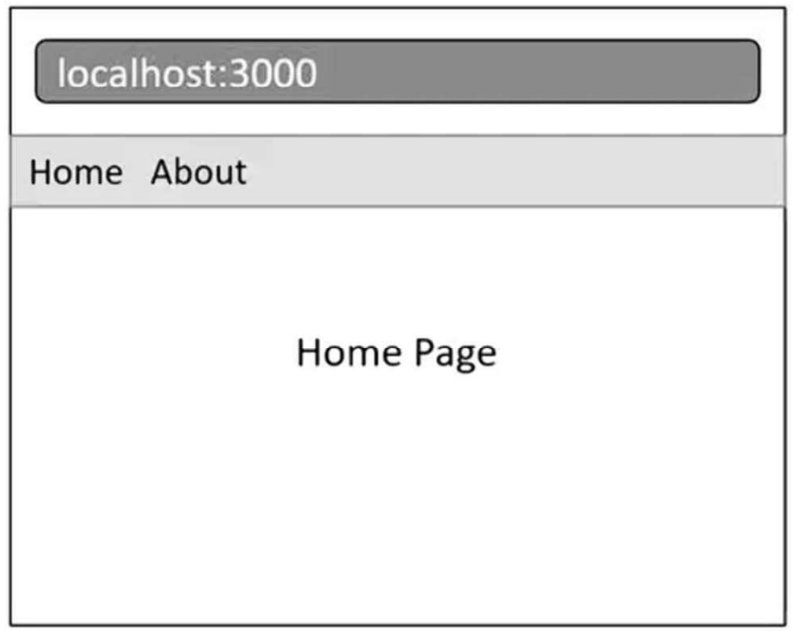
Links

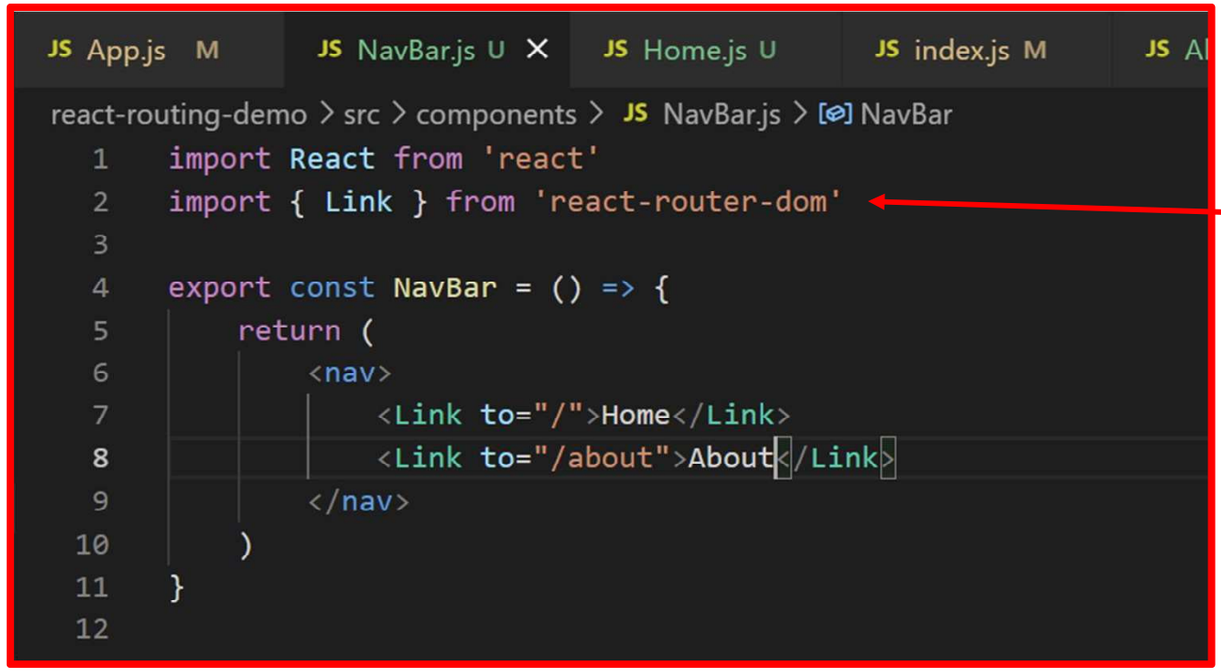
Links

Scenario : Let's add a Navbar in the app component with two links

- One to the **Home page** and One to the **About page**
- Click of these links we should be able to navigate between the two route







Clickable elements to navigate between the routes to navigate to another route

React Router provides us with the **Link component**

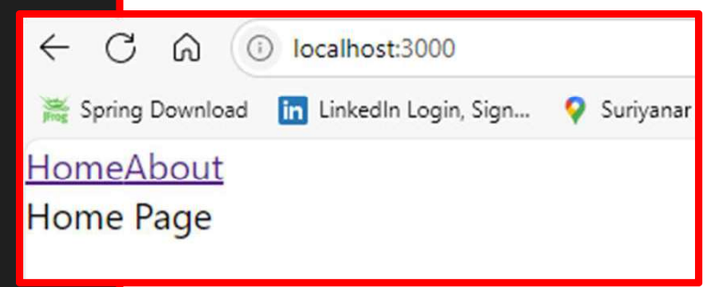
App.js M | NavBar.js U | Home.js U | index.js M | About.js U

react-routing-demo > src > App.js > ...

```
1  import { Routes, Route } from 'react-router-dom';
2  import { Home } from './components/Home'
3  import { About } from './components/About'
4  import { NavBar } from './components/NavBar'
5  function App() {
6    return (
7      <>
8      <NavBar />
9      <Routes>
10       <Route path="/" element={<Home />}></Route>
11
12       <Route path="about" element={<About />}></Route>
13
14     </Routes>
15     </>
16   );
17 }
18
19 export default App;
```

Fragments [Place Where Components are Rendered in SPA]

We Have Included NavBar Component

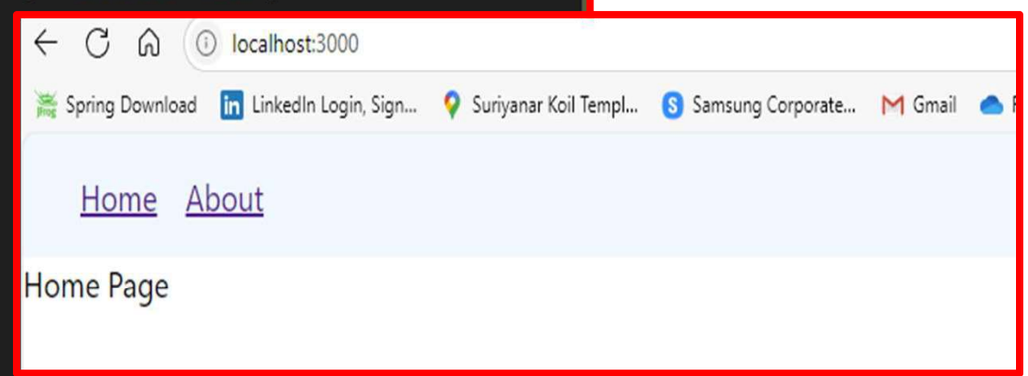


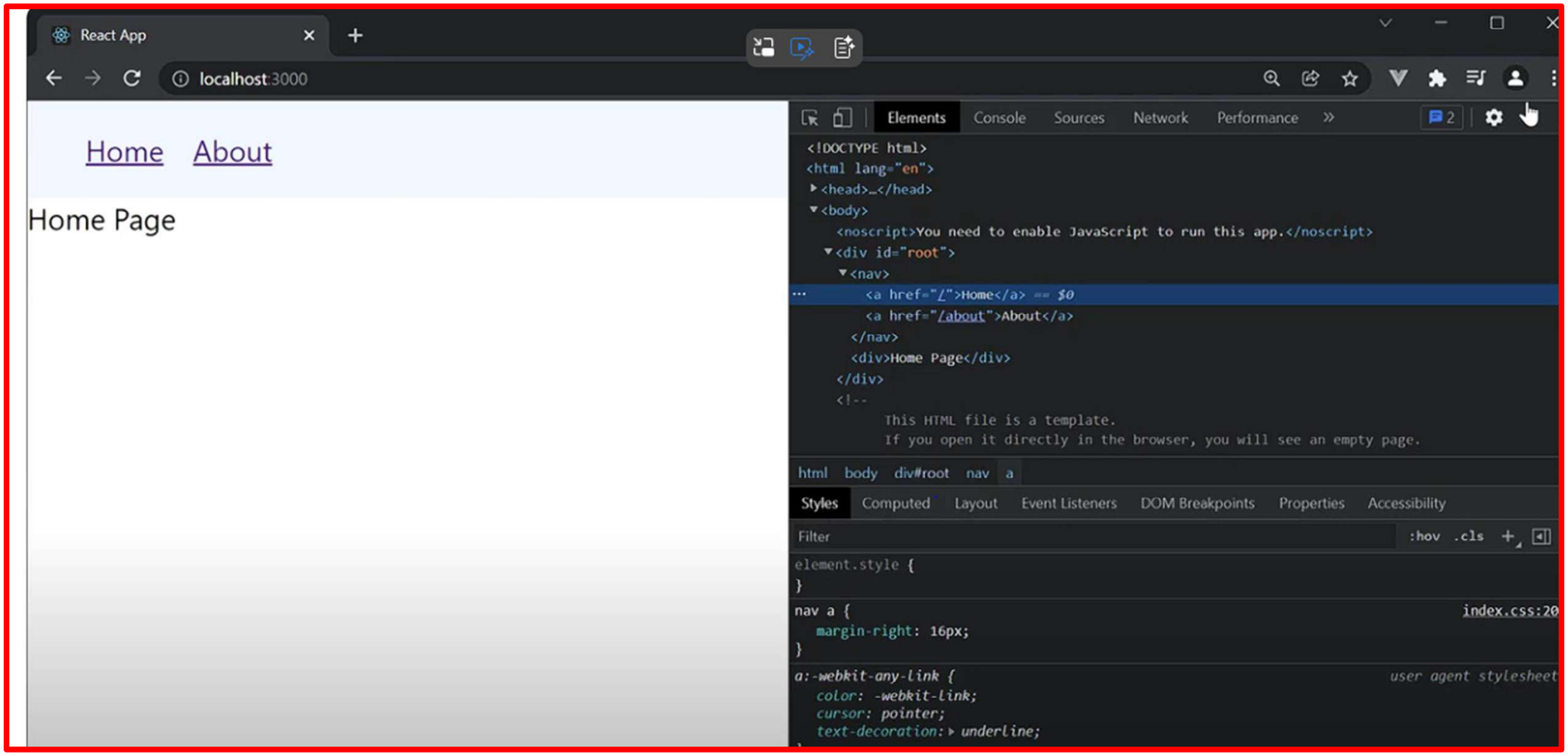
Let's Add Some Styling to Make it Presentable

JS App.js M × | # index.css M × | JS NavBar.js U | JS Home.js U | JS index.js M | JS Abou

react-routing-demo > src > # index.css > nav a

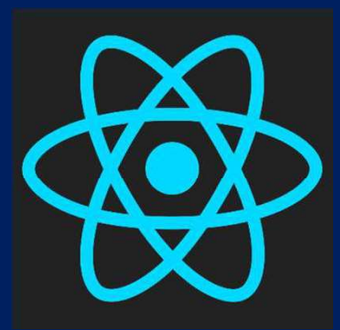
```css
body {
  margin: 0;
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',
    'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
    sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}

code {
  font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',
    monospace;
}

nav {
  background-color: aliceblue;
  padding: 16px 32px;
}

nav a {
  margin-right: 16px;
}
```




React App
localhost:3000
Home About
Home Page
Elements Console Sources Network Performance
<!DOCTYPE html>
<html lang="en">
<head>…</head>
<body>
<noscript>You need to enable JavaScript to run this app.</noscript>

<nav>
Home == $0
About
</nav>
Home Page

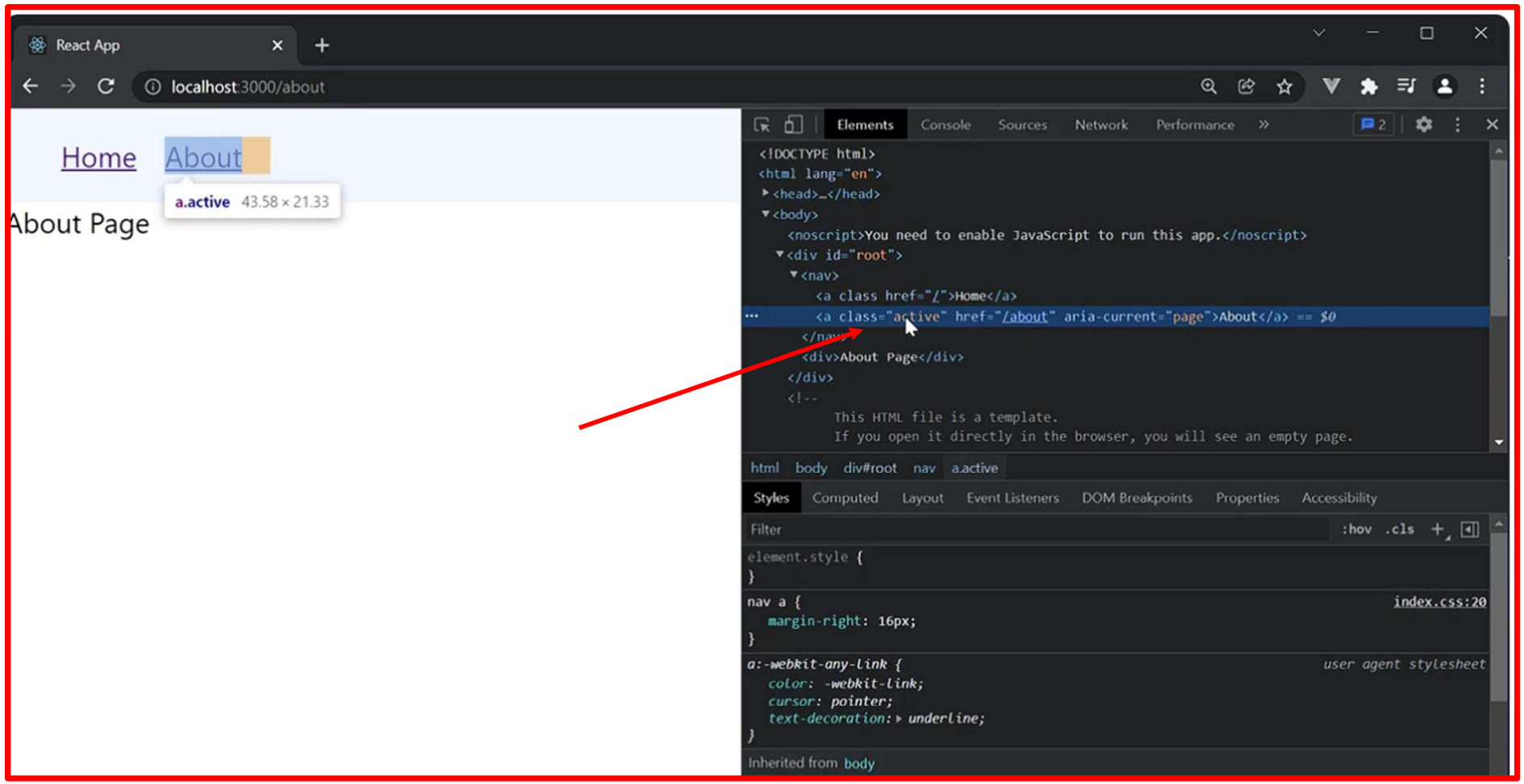
<!--
This HTML file is a template.
If you open it directly in the browser, you will see an empty page.
html body div#root nav a
Styles Computed Layout Event Listeners DOM Breakpoints Properties Accessibility
Filter :hov .cls
element.style {
}
nav a { index.css:20
margin-right: 16px;
}
a:-webkit-any-link { user agent stylesheet
color: -webkit-link;
cursor: pointer;
text-decoration: underline;



Active Links

- In practical web applications it is common to style the link corresponding to the active route in a different way this also benefits the user from a user experience point of view
- **React Router** provides another component called **Nav Link** which knows whether or not it is the active link we can use that component to style the **Active Link in our NavBa**r

- What is special about this **NavLink Component** though is that by default it receives an active class when the link is the current route.

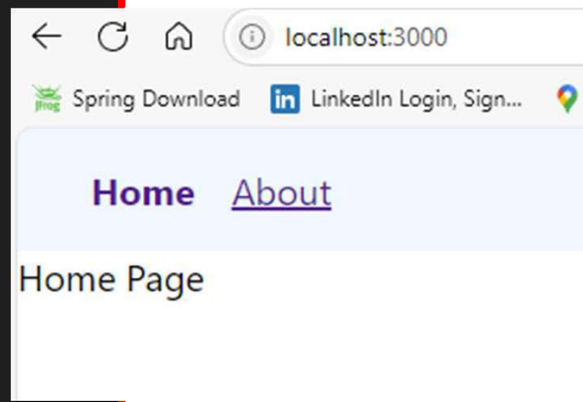

React App
localhost:3000/about
Home
About
a.active 43.58 × 21.33
About Page
Elements
Console
Sources
Network
Performance
<!DOCTYPE html>
<html lang="en">
<head>…</head>
<body>
<noscript>You need to enable JavaScript to run this app.</noscript>

<nav>
<a class href="/">Home
About == $0
About Page

<!--
This HTML file is a template.
If you open it directly in the browser, you will see an empty page.
html body div#root nav a.active
Styles
Computed
Layout
Event Listeners
DOM Breakpoints
Properties
Accessibility
Filter
:hov .cls
element.style {
}
nav a {
index.css:20
margin-right: 16px;
}
a:-webkit-any-link {
user agent stylesheet
color: -webkit-link;
cursor: pointer;
text-decoration: underline;
}
Inherited from body

In index.css

```
14
15  nav {
16    background-color: aliceblue;
17    padding: 16px 32px;
18  }
19
20  nav a {
21    margin-right: 16px;
22  }
23
24  nav a.active {
25    text-decoration: none;
26    font-weight: bold;
27  }
```



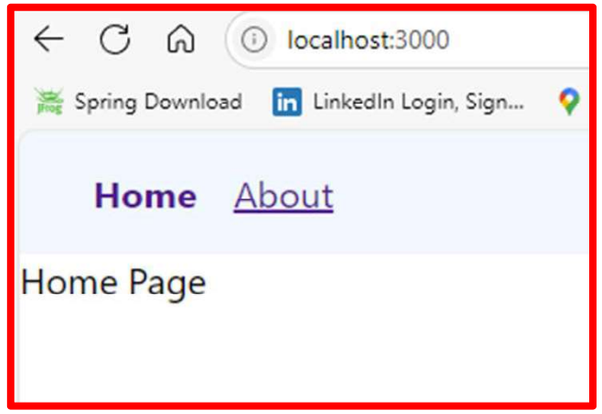
Active Link [Css in JS] with Style prop

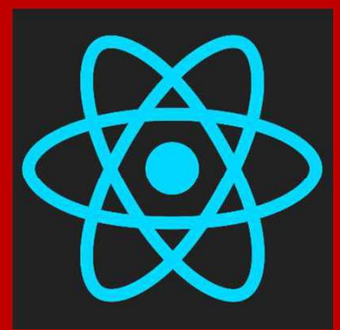
App.js M | index.css M | NavBar.js U | Home.js U | index.js M

react-routing-demo > src > components > NavBar.js > ...

```
1   import React from 'react'
2   import { NavLink } from 'react-router-dom'
3
4   export const NavBar = () => {
5
6       const navLinkStyles = ({ isActive }) => {
7           return {
8               fontWeight: isActive ? 'bold' : 'normal',
9               textDecoration: isActive ? 'none': 'underline'
10          }
11      }
12
13      return (
14          <nav>
15              <NavLink style={navLinkStyles} to="/">Home</NavLink>
16              <NavLink style={navLinkStyles} to="/about">About</NavLink>
17          </nav>
18      )
19  }
20
```

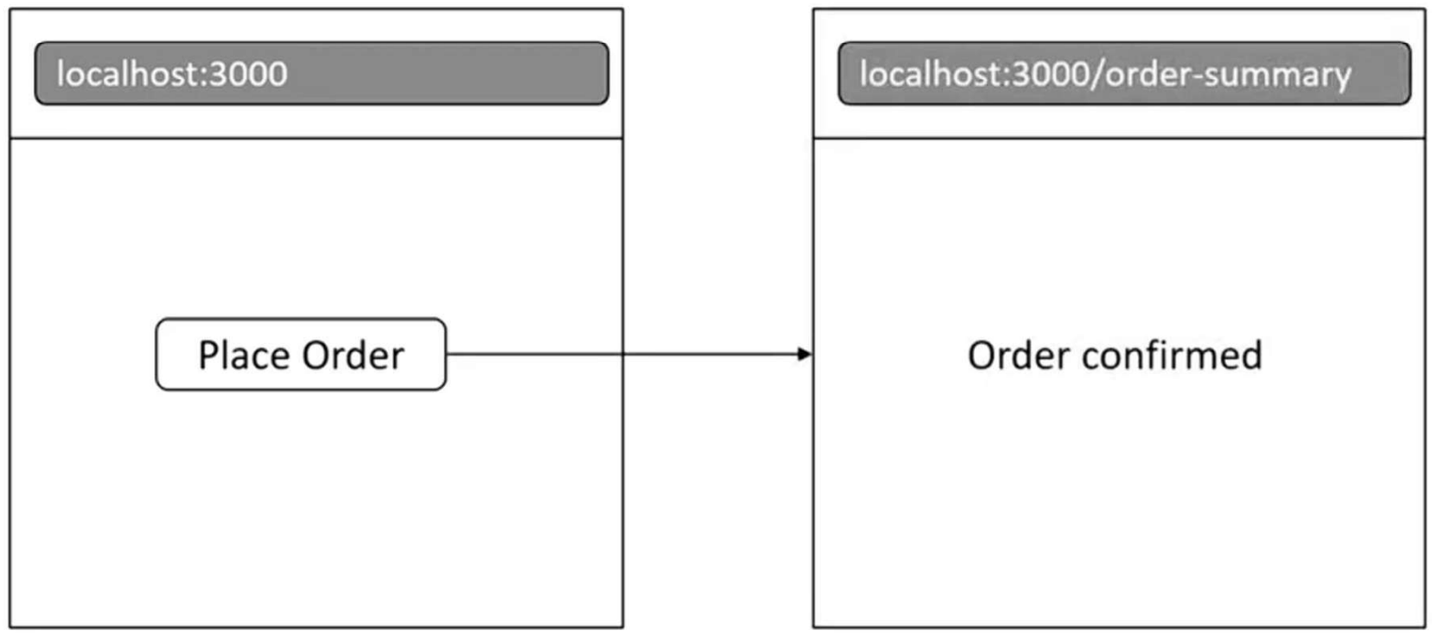
```
23
24  /* nav a.active {
25      text-decoration: none;
26      font-weight: bold;
27  } */
```



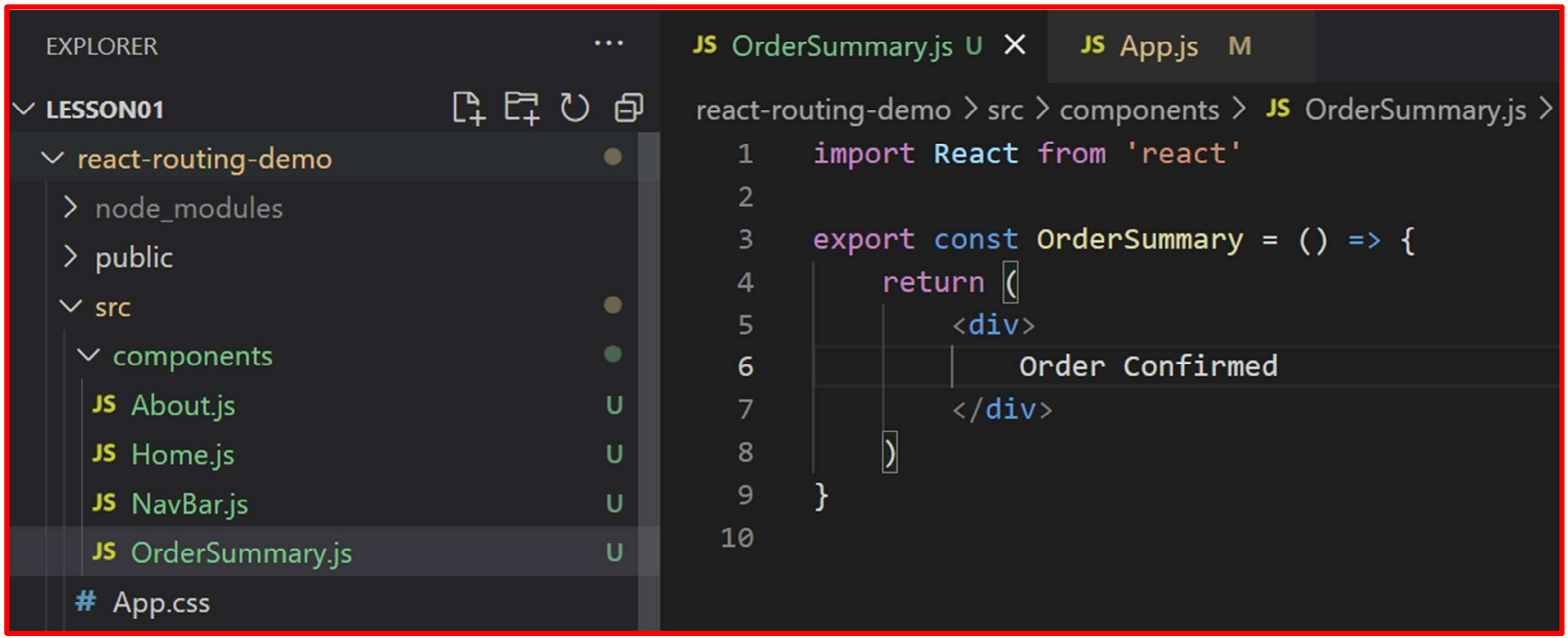


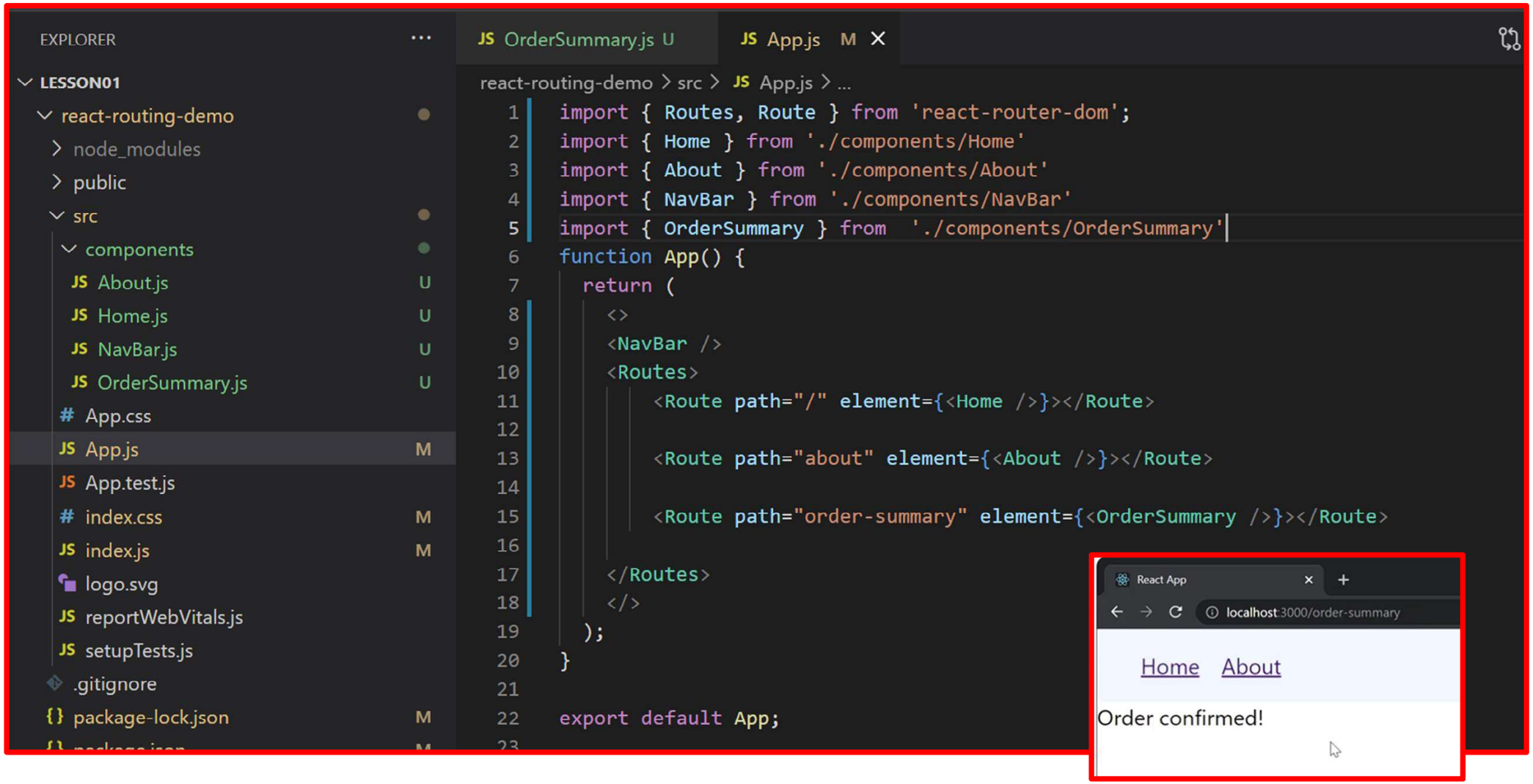
Navigating Programmatically

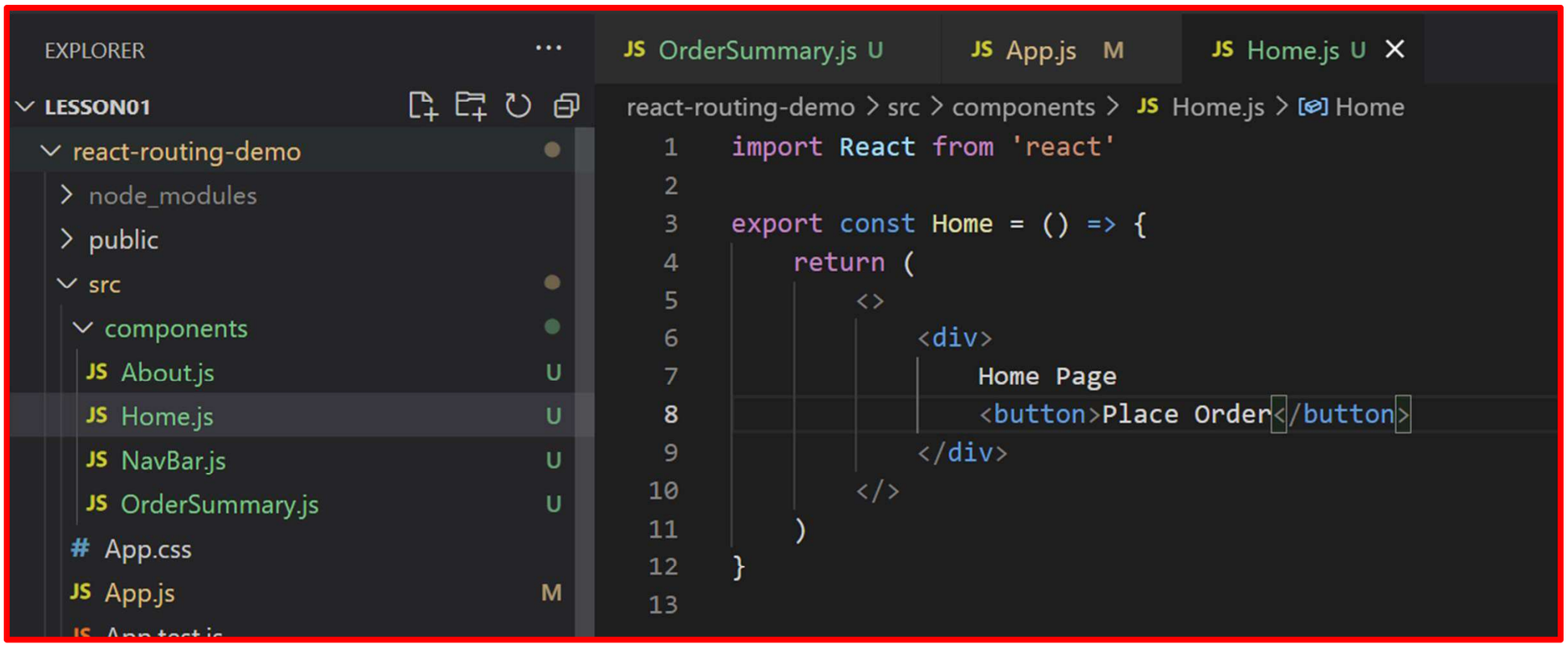
- Till now we learned how to navigate to different routes using the link and navlink components for the most part it is pretty much all you need however at times you might need to programmatically navigate to a particular route



- Let’s Implement the scenario, by a button click handler where we programmatically navigate to a different route in our application.
- On click of Order button in the home page let's navigate to the order summary page


EXPLORER
LESSON01
react-routing-demo
node_modules
public
src
components
About.js U
Home.js U
NavBar.js U
OrderSummary.js U
App.css
OrderSummary.js U
App.js M
react-routing-demo > src > components > OrderSummary.js >
1 import React from 'react'
2
3 export const OrderSummary = () => {
4 return (
5
6 Order Confirmed
7
8)
9 }
10


EXPLORER
LESSON01
react-routing-demo
node_modules
public
src
components
About.js U
Home.js U
NavBar.js U
OrderSummary.js U
App.css
App.js M
App.test.js
index.css M
index.js M
logo.svg
reportWebVitals.js
setupTests.js
.gitignore
package-lock.json M
OrderSummary.js U
App.js M
react-routing-demo > src > App.js > ...
1 import { Routes, Route } from 'react-router-dom';
2 import { Home } from './components/Home'
3 import { About } from './components/About'
4 import { NavBar } from './components/NavBar'
5 import { OrderSummary } from './components/OrderSummary'
6 function App() {
7 return (
8 <>
9 <NavBar />
10 <Routes>
11 <Route path="/" element={<Home />}></Route>
12
13 <Route path="about" element={<About />}></Route>
14
15 <Route path="order-summary" element={<OrderSummary />}></Route>
16
17 </Routes>
18 </>
19);
20 }
21
22 export default App;
React App
localhost:3000/order-summary
Home About
Order confirmed!

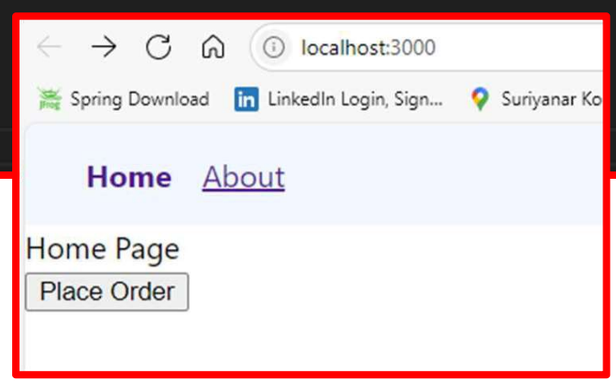

EXPLORER
LESSON01
react-routing-demo
node_modules
public
src
components
About.js U
Home.js U
NavBar.js U
OrderSummary.js U
App.css
App.js M
OrderSummary.js U
App.js M
Home.js U
react-routing-demo > src > components > Home.js > Home
1 import React from 'react'
2
3 export const Home = () => {
4 return (
5 <>
6
7 Home Page
8 <button>Place Order</button>
9
10 </>
11)
12 }
13

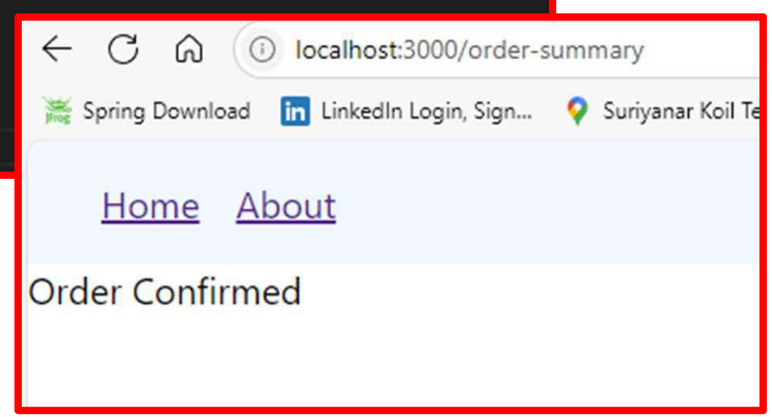
- To Navigate Programming React Router provides **useNavigate** Hook

JS OrderSummary.js U | JS App.js M | JS Home.js U

react-routing-demo > src > components > JS Home.js > ...

```
1   import React from 'react'
2   import { useNavigate } from 'react-router-dom'
3
4   export const Home = () => {
5       const navigate = useNavigate()
6       return (
7           <>
8               <div>
9                   Home Page
10              </div>
11                  <button onClick={() => navigate('order-summary')}>Place Order</button>
12
13          </>
14      )
15  }
16
```



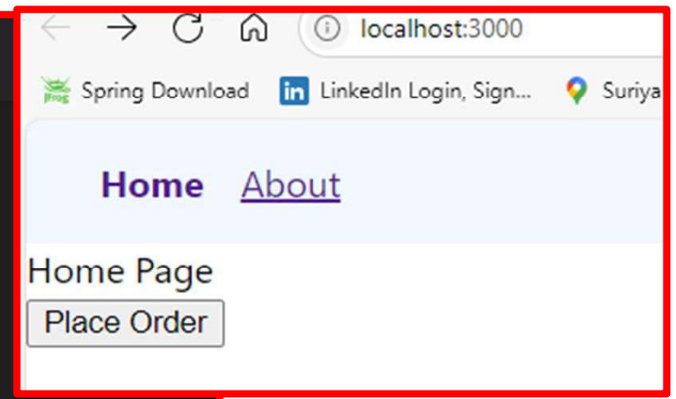


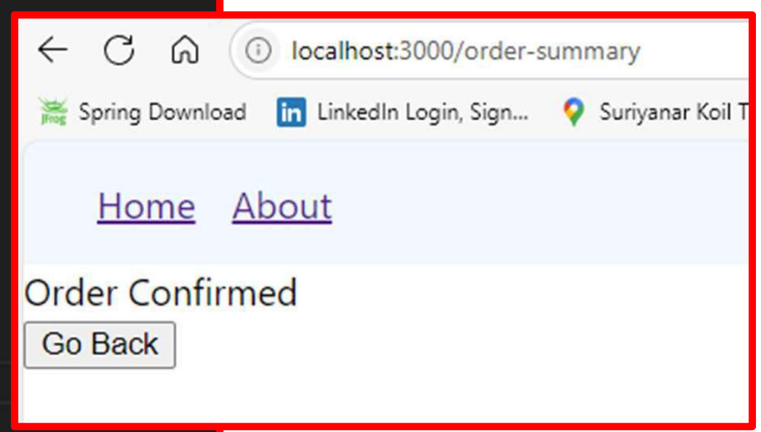
- Now implementing a back button programmatically is also done with the **useNavigate** hook

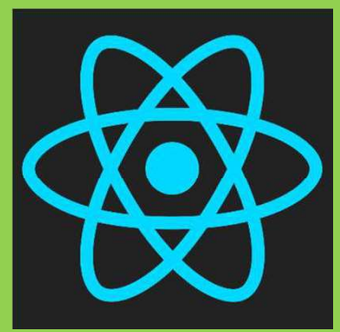
JS OrderSummary.js U | JS Home.js U | JS App.js M

react-routing-demo > src > components > JS OrderSummary.js > ...

```
1   import React from 'react'
2   import { useNavigate } from 'react-router-dom'
3
4   export const OrderSummary = () => {
5       const navigate = useNavigate()
6       return (
7           <>
8           <div>
9               Order Confirmed
10          </div>
11          <button onClick={() => navigate(-1)}>Go Back</button>
12          </>
13      )
14  }
15
```

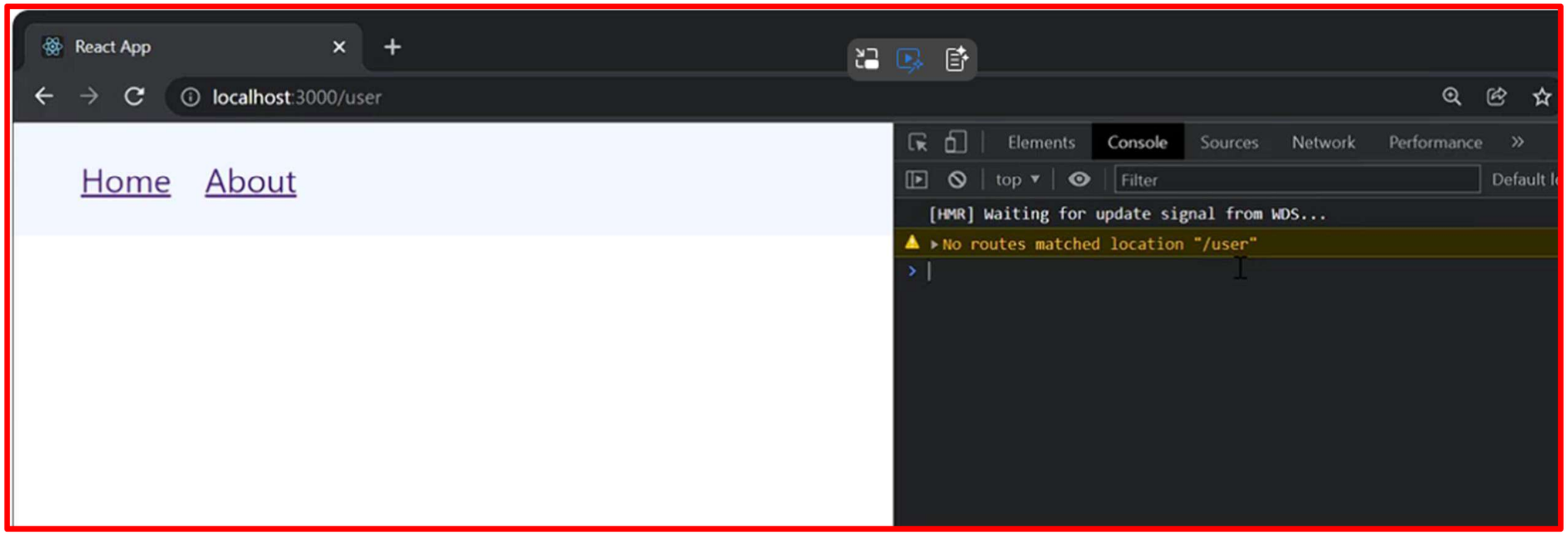




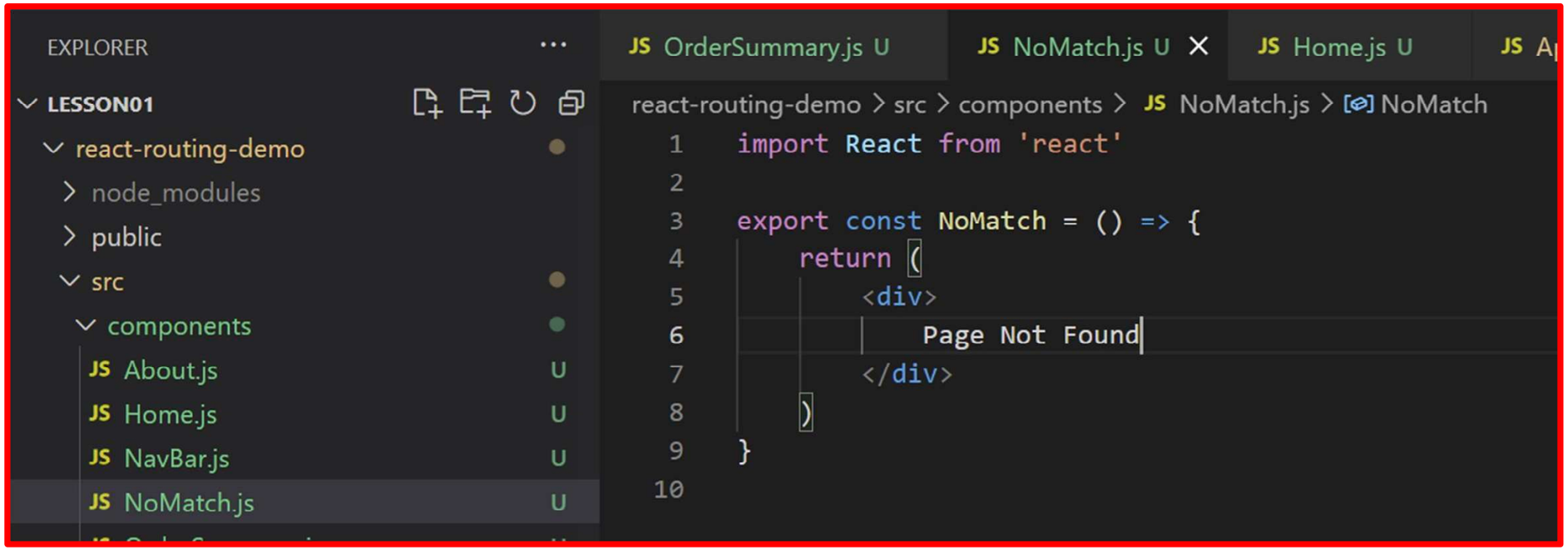


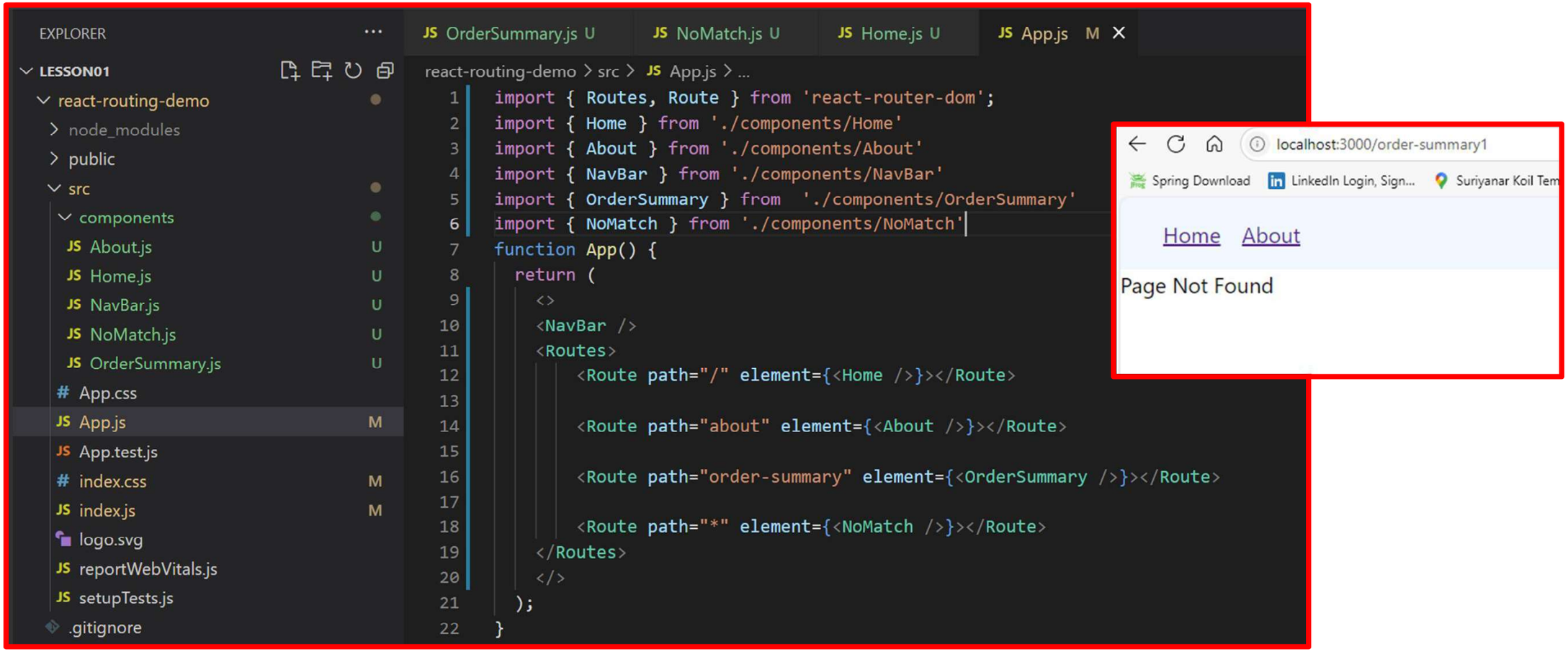
No Match Route

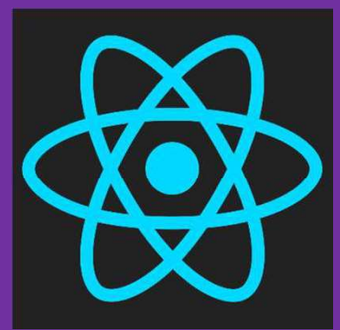
- This scenario is not favorable from a user point of view a user might assume that the app is still loading or there is an error in the **/user route**



- It would be better is if we inform the user that the url does not match any route in our application



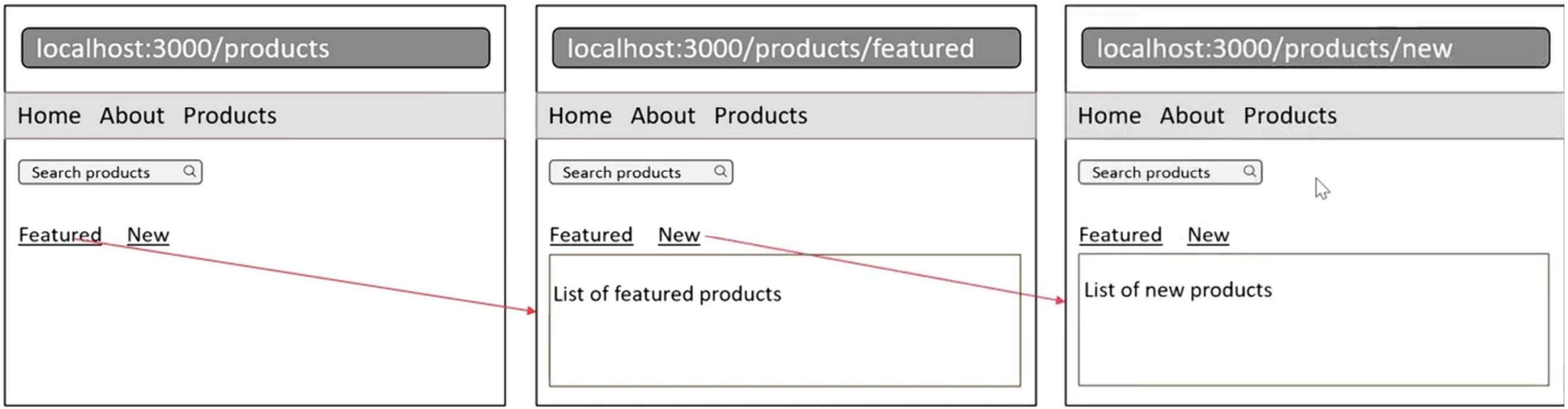

EXPLORER
LESSON01
react-routing-demo
node_modules
public
src
components
About.js U
Home.js U
NavBar.js U
NoMatch.js U
OrderSummary.js U
App.css
App.js M
App.test.js
index.css M
index.js M
logo.svg
reportWebVitals.js
setupTests.js
.gitignore
OrderSummary.js U
NoMatch.js U
Home.js U
App.js M
react-routing-demo > src > App.js > ...
1 import { Routes, Route } from 'react-router-dom';
2 import { Home } from './components/Home'
3 import { About } from './components/About'
4 import { NavBar } from './components/NavBar'
5 import { OrderSummary } from './components/OrderSummary'
6 import { NoMatch } from './components/NoMatch'
7 function App() {
8 return (
9 <>
10 <NavBar />
11 <Routes>
12 <Route path="/" element={<Home />}></Route>
13
14 <Route path="about" element={<About />}></Route>
15
16 <Route path="order-summary" element={<OrderSummary />}></Route>
17
18 <Route path="*" element={<NoMatch />}></Route>
19 </Routes>
20 </>
21);
22 }
localhost:3000/order-summary1
Spring Download
LinkedIn Login, Sign...
Suriyanar Koil Tem
Home About
Page Not Found



Nested Routes

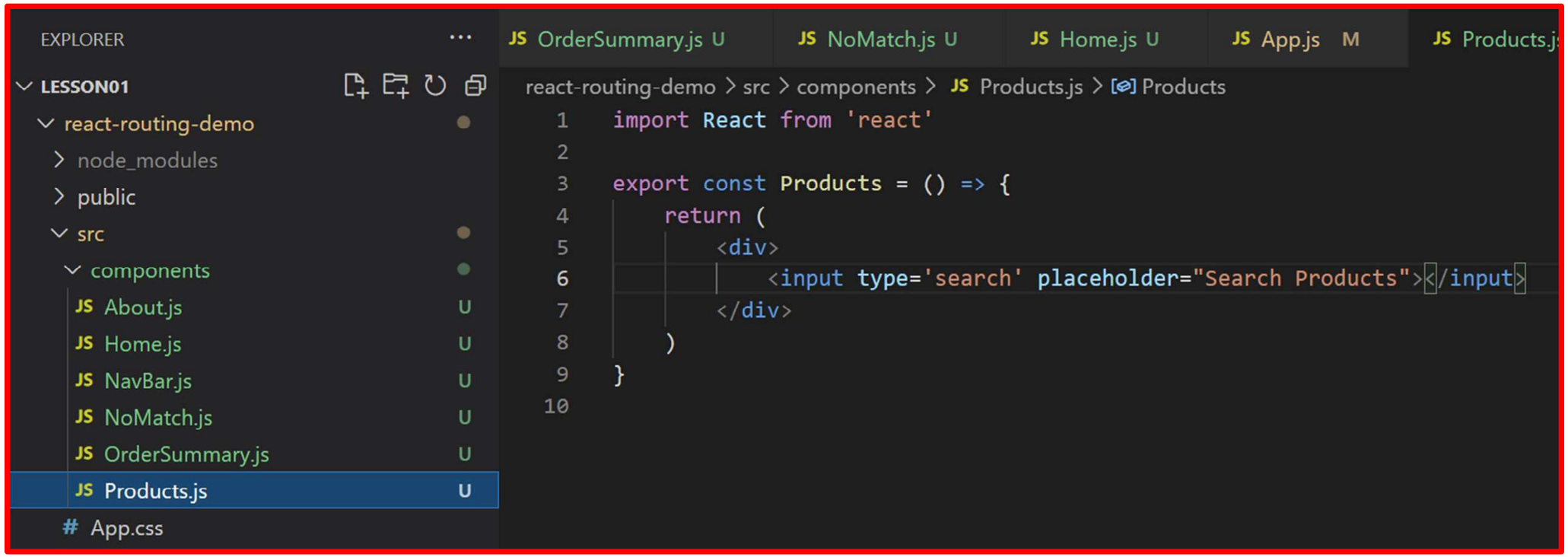
- So far we have seen how react router helps us navigate across pages in our application but what you should know is that react router also helps to switch between a portion of the view inside the page

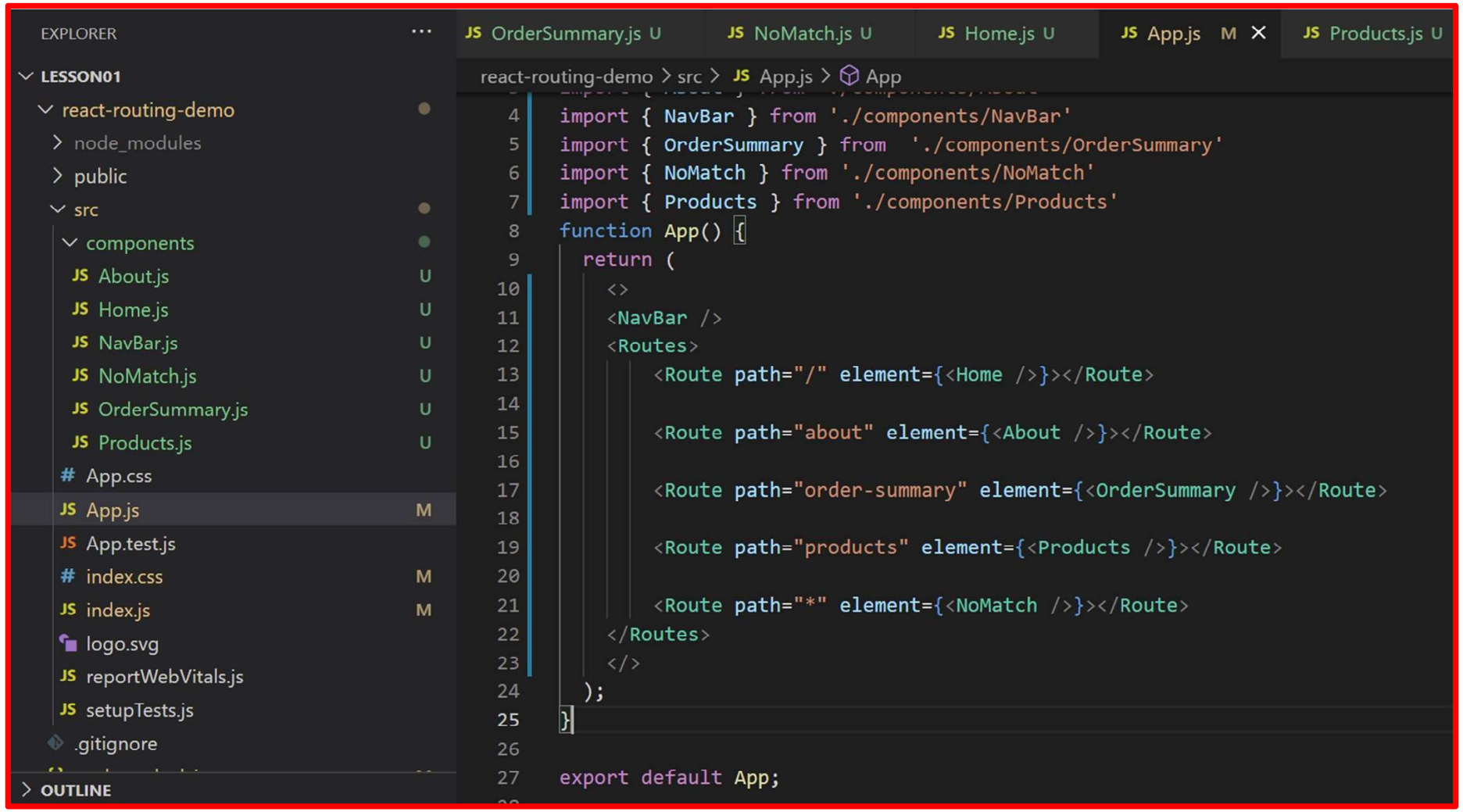
To Achieve this we make use of Nested Routes



Step 1 :

- We are going to configure a new route to the products page and add a link in the nav bar

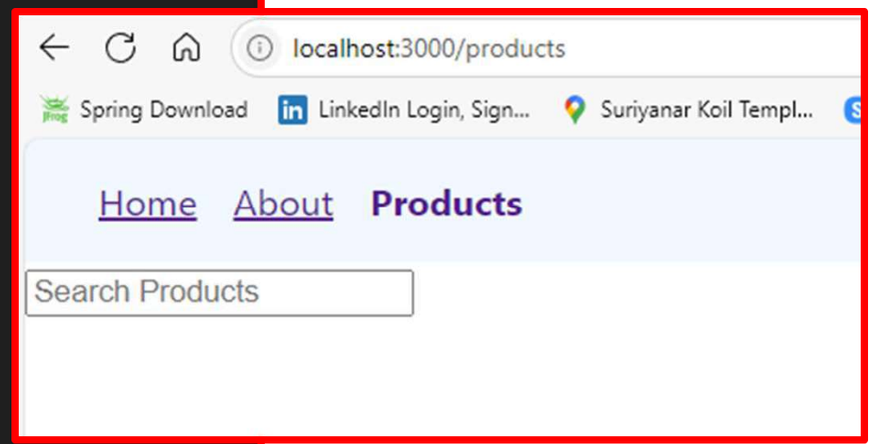



EXPLORER
LESSON01
react-routing-demo
node_modules
public
src
components
About.js U
Home.js U
NavBar.js U
NoMatch.js U
OrderSummary.js U
Products.js U
App.css
App.js M
App.test.js
index.css M
index.js M
logo.svg
reportWebVitals.js
setupTests.js
.gitignore
OUTLINE
OrderSummary.js U
NoMatch.js U
Home.js U
App.js M
Products.js U
react-routing-demo > src > App.js > App
import { NavBar } from './components/NavBar'
import { OrderSummary } from './components/OrderSummary'
import { NoMatch } from './components/NoMatch'
import { Products } from './components/Products'
function App() {
return (
<>
<NavBar />
<Routes>
<Route path="/" element={<Home />}></Route>
<Route path="about" element={<About />}></Route>
<Route path="order-summary" element={<OrderSummary />}></Route>
<Route path="products" element={<Products />}></Route>
<Route path="*" element={<NoMatch />}></Route>
</Routes>
</>
);
}
export default App;

Products.js U | App.js M | NavBar.js U

react-routing-demo > src > components > NavBar.js > ...

```
1   import React from 'react'
2   import { NavLink } from 'react-router-dom'
3
4   export const NavBar = () => {
5
6       const navLinkStyles = ({ isActive }) => {
7           return {
8               fontWeight: isActive ? 'bold' : 'normal',
9               textDecoration: isActive ? 'none': 'underline'
10          }
11      }
12
13      return (
14          <nav>
15              <NavLink style={navLinkStyles} to="/">Home</NavLink>
16              <NavLink style={navLinkStyles} to="/about">About</NavLink>
17              <NavLink style={navLinkStyles} to="/products">Products</NavLink>
18          </nav>
19      )
20  }
21
```



Step 2 :

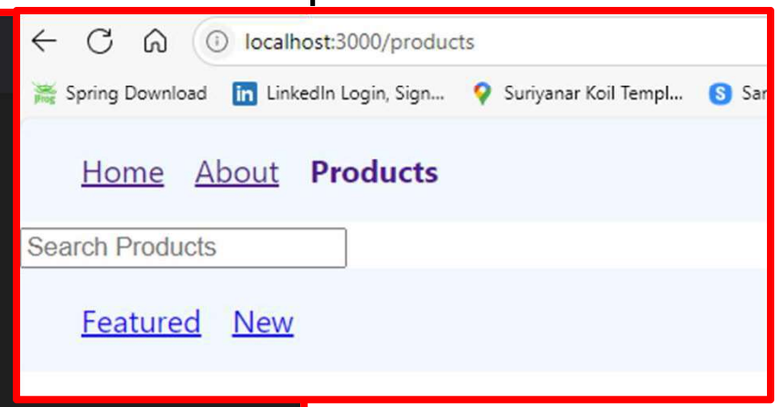
- Lets add another set of navigation links within the products component

Products.js U | App.js M | NavBar.js U

react-routing-demo > src > components > Products.js > ...

```
1   import React from 'react'
2   import { Link } from 'react-router-dom'
3
4   export const Products = () => {
5       return (
6           <>
7               <div>
8                   <input type='search' placeholder="Search Products"></input>
9               </div>
10              <nav>
11                  <Link to='featured'>Featured</Link>
12                  <Link to='new'>New</Link>
13
14              </nav>
15          </>
16      )
17  }
18
```

Relative Path
Without /



Products.js U | index.css M | App.js M

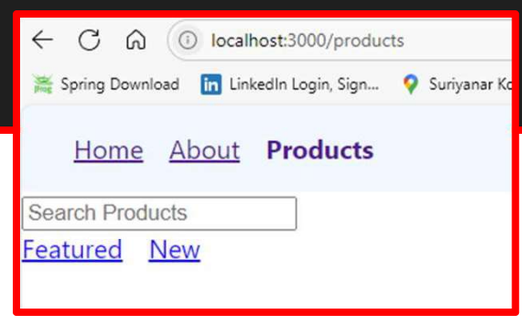
react-routing-demo > src > # index.css > .primary-nav

```
    sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}

code {
  font-family: source-code-pro, Menlo, M
    monospace;
}

.primary-nav {
  background-color: aliceblue;
  padding: 16px 32px;
}

nav a {
  margin-right: 16px;
}

/* nav a.active {
  text-decoration: none;
  font-weight: bold;
} */
```

Products.js U | index.css M | App.js M | NavBar.js U

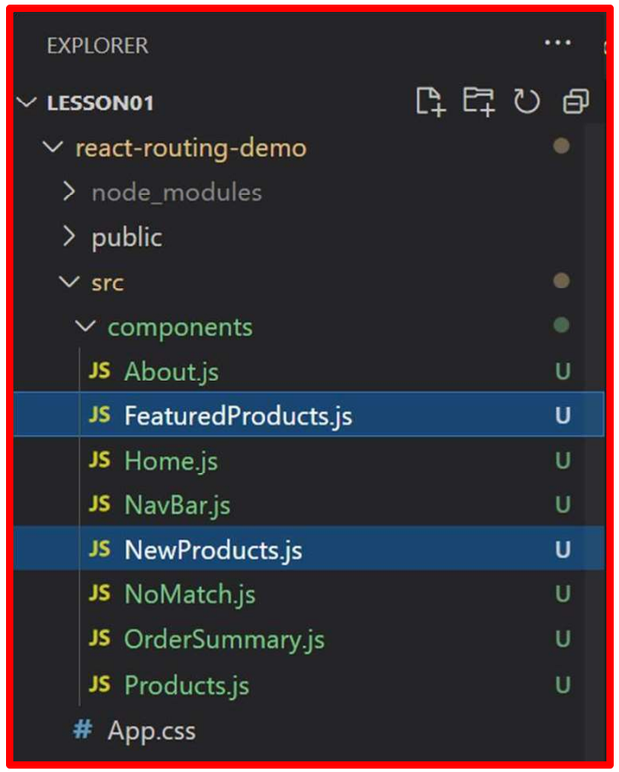
react-routing-demo > src > components > NavBar.js > NavBar

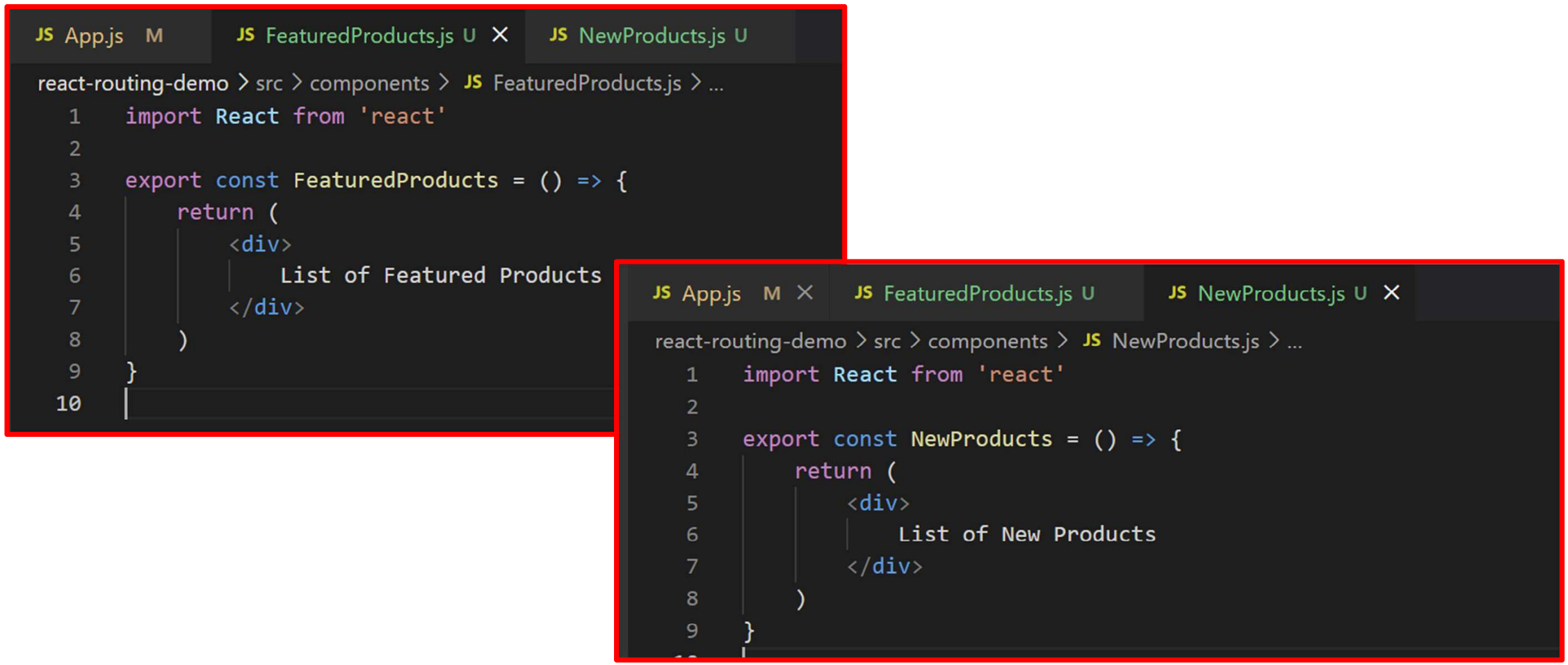
```
import React from 'react'
import { NavLink } from 'react-router-dom'

export const NavBar = () => {

    const navLinkStyles = ({ isActive }) => {
        return {
            fontWeight: isActive ? 'bold' : 'normal',
            textDecoration: isActive ? 'none': 'underline'
        }
    }

    return (
        <nav className='primary-nav'>
            <NavLink style={navLinkStyles} to="/">Home</NavLink>
            <NavLink style={navLinkStyles} to="/about">About</NavLink>
            <NavLink style={navLinkStyles} to="/products">Products</NavLink>
        </nav>
    )
}
```



Step 3 :

- We will create 2 New Components that need to be rendered for **Featured Products** and **New Products** and also configure the new routes




App.js M
FeaturedProducts.js U
NewProducts.js U
react-routing-demo > src > components > FeaturedProducts.js > ...
import React from 'react'
export const FeaturedProducts = () => {
return (

List of Featured Products

)
}
App.js M
FeaturedProducts.js U
NewProducts.js U
react-routing-demo > src > components > NewProducts.js > ...
import React from 'react'
export const NewProducts = () => {
return (

List of New Products

)
}

Now Configuring Nested Routes

- Both the components have to be nested within the **products** route

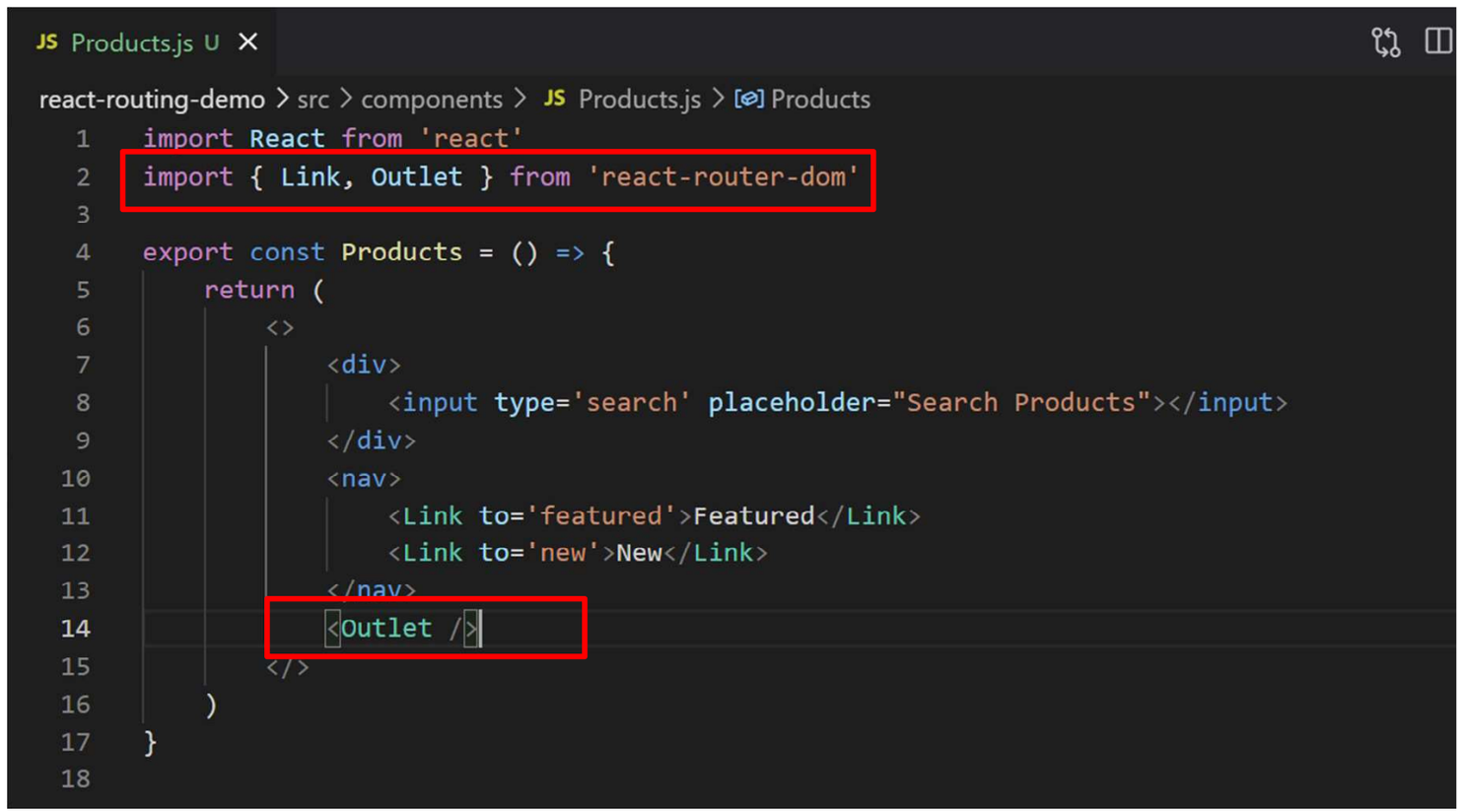
App.js M

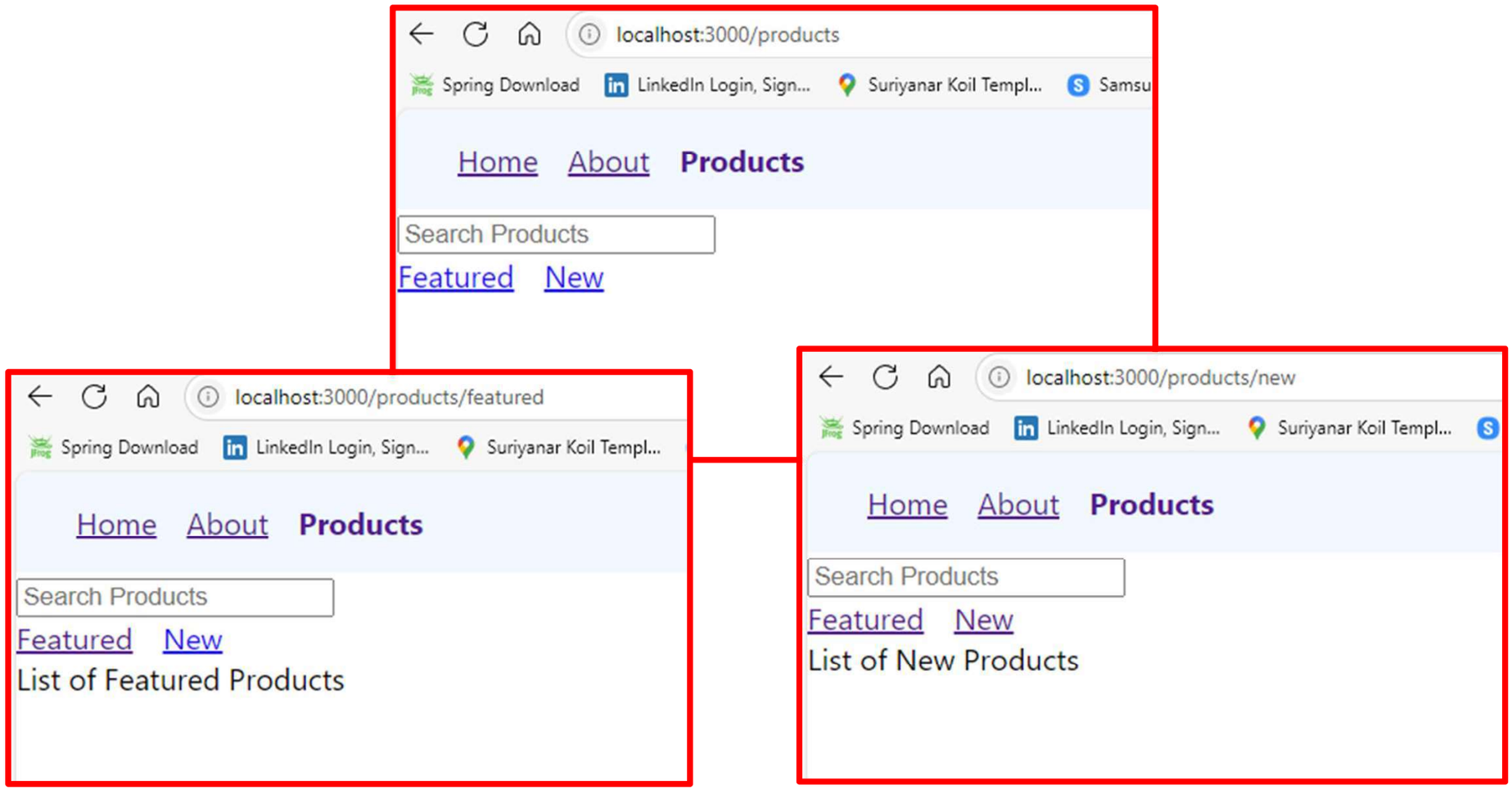
react-routing-demo > src > App.js > App

```
import { NoMatch } from './components/NoMatch'
import { Products } from './components/Products'
import { FeaturedProducts } from './components/FeaturedProducts'
import { NewProducts } from './components/NewProducts'
function App() {
  return (
    <>
    <NavBar />
    <Routes>
        <Route path="/" element={<Home />}></Route>
        <Route path="about" element={<About />}></Route>
        <Route path="order-summary" element={<OrderSummary />}></Route>
        <Route path="products" element={<Products />}>

            <Route path="featured" element={<FeaturedProducts />}></Route>
            <Route path="new" element={<NewProducts />}></Route>

        </Route>
        <Route path="*" element={<NoMatch />}></Route>
    </Routes>
    </>
  );
```

- What is special about nested routes is that react router automatically forms the full path to the children routes

- However Products Page Doesn't Know Where to Render Child Components
  - So React Router Provides Outlet Component




localhost:3000/products
Spring Download
LinkedIn Login, Sign...
Suriyanar Koil Templ...
Samsu
Home
About
Products
Search Products
Featured
New
localhost:3000/products/featured
Spring Download
LinkedIn Login, Sign...
Suriyanar Koil Templ...
Home
About
Products
Search Products
Featured
New
List of Featured Products
localhost:3000/products/new
Spring Download
LinkedIn Login, Sign...
Suriyanar Koil Templ...
Home
About
Products
Search Products
Featured
New
List of New Products