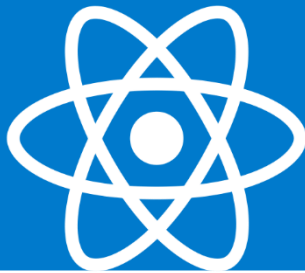




Routing in React JS

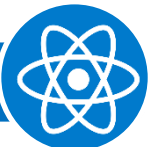
#React Notes

| Routing



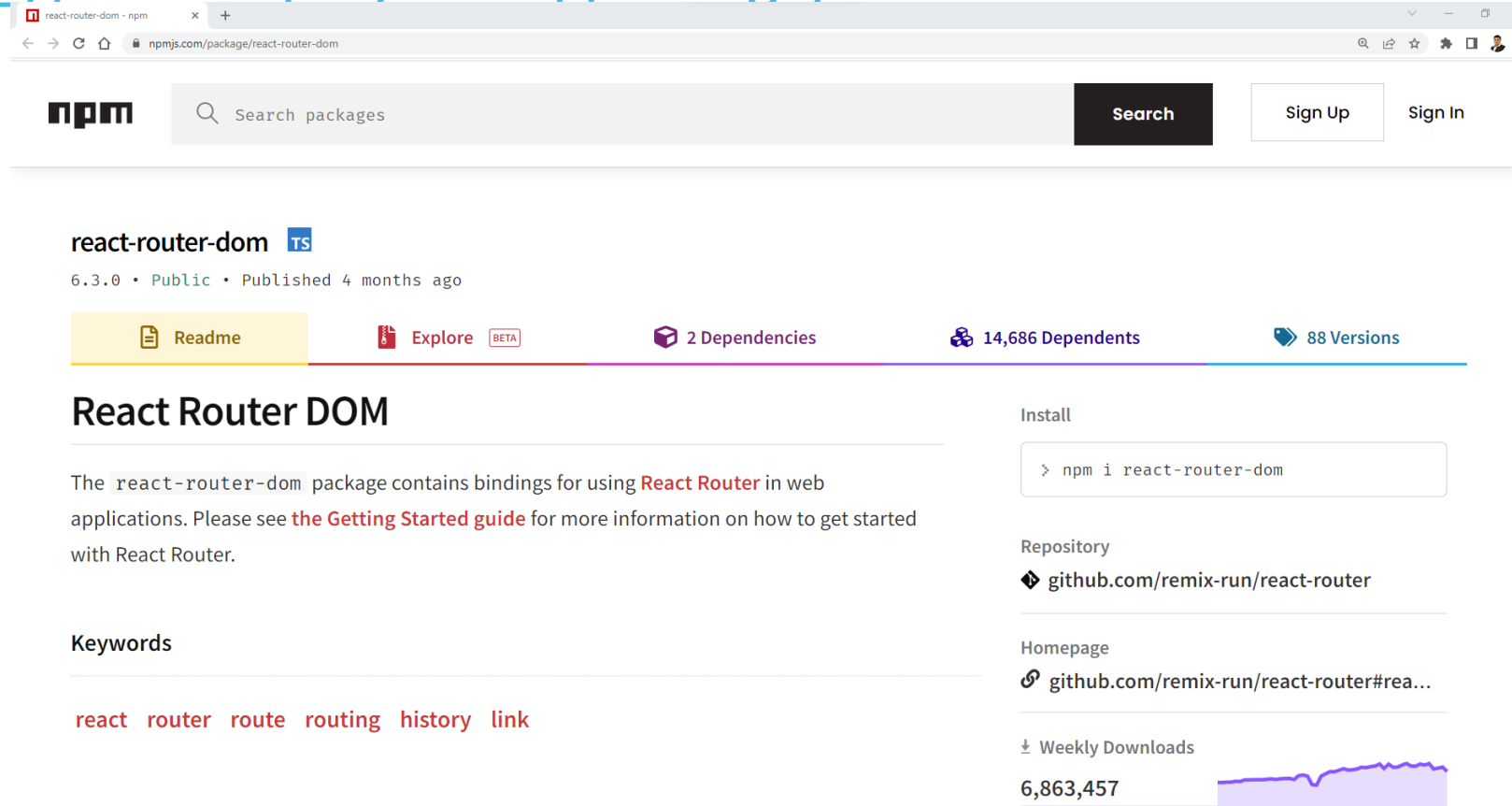
What is React Router?

- The React Router helps you define dynamic URLs and select a suitable Component to render on the user browser corresponding to each URL.
- `<BrowserRouter>` is used more commonly, it uses the History API included in HTML5 to monitor your router's history
- The **<Route> component** defines a **mapping between an URL and a Component**. That means when the user visits by an URL on browser, a corresponding Component shall be rendered on interface.
- The **exact attribute** is used in the `<Route>` to say that this `<Route>` only operates if the URL on the browser matches absolutely the value of its path attribute.



React Router DOM

- To Enable Navigation in React we need to Add React Router DOM package.
- <https://www.npmjs.com/package/react-router-dom>

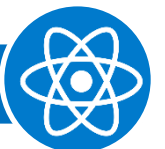
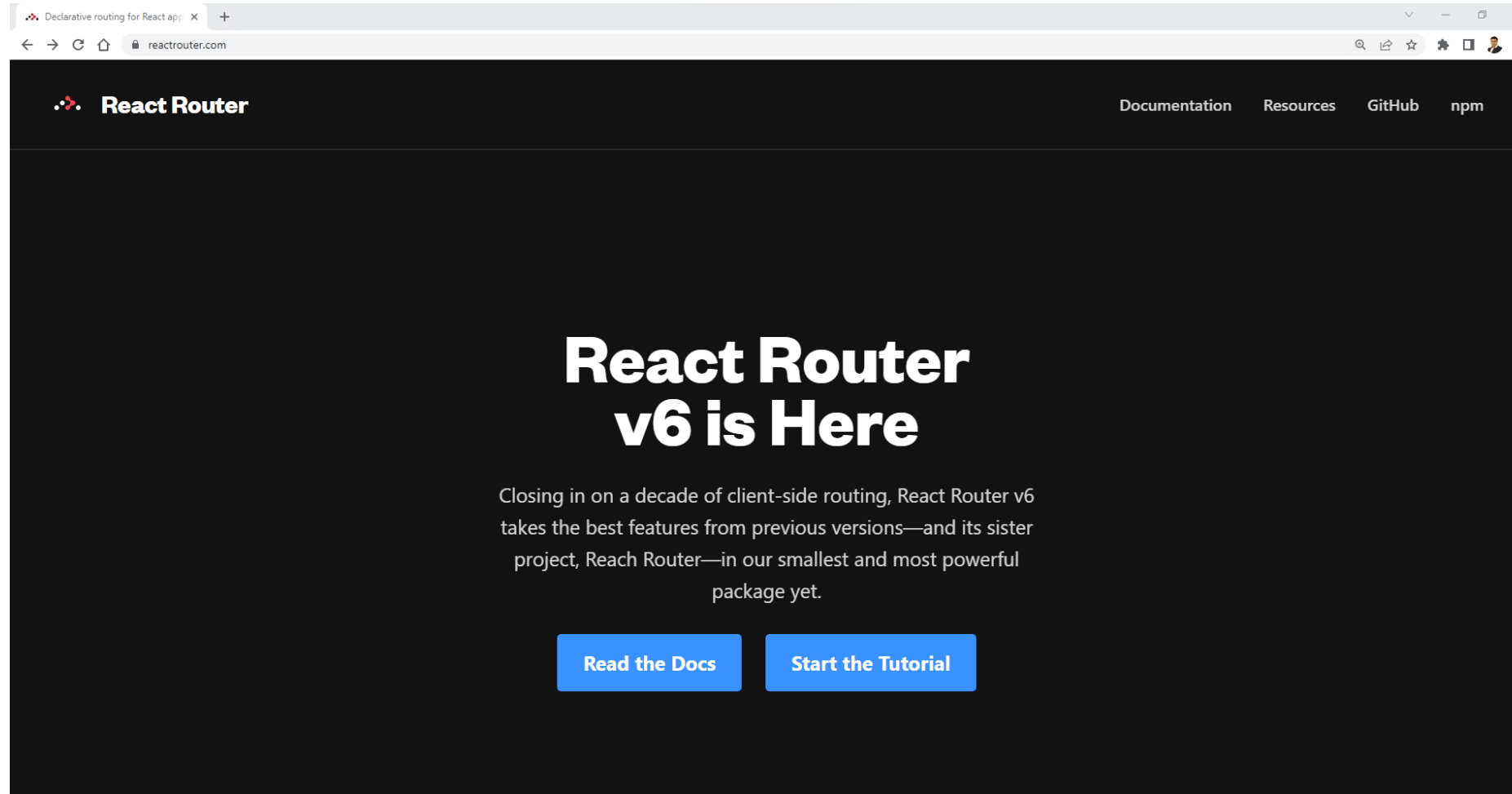


The screenshot shows the npm package page for `react-router-dom`. The page header includes the npm logo, a search bar, and links for Sign Up and Sign In. The package details section shows the package name `react-router-dom` with a TypeScript icon, version `6.3.0`, and a note that it is Public and published 4 months ago. Below this are tabs for Readme, Explore (marked BETA), 2 Dependencies, 14,686 Dependents, and 88 Versions. The main content area has a heading `React Router DOM` and a description: "The `react-router-dom` package contains bindings for using **React Router** in web applications. Please see the **Getting Started guide** for more information on how to get started with React Router." Below the description are keywords: `react router route routing history link`. On the right side, there is an "Install" section with a code block showing `> npm i react-router-dom`, a "Repository" section linking to `github.com/remix-run/react-router`, a "Homepage" section linking to `github.com/remix-run/react-router#rea...`, and a "Weekly Downloads" section showing a graph and the number `6,863,457`.



React Router DOM Official Website

- React Documentation : <https://reactrouter.com/>



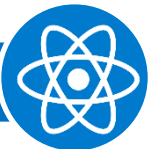
Types of Router

1. **BrowserRouter** /urlname

- makes pretty URLs like
 - `example.com/home.`
 - `example.com/about.`

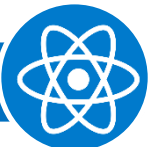
2. **HashRouter** #urlname

- makes URLs with the octothorpe (or hashtag, if you will) that would look like
 - `example.com/#about`
 - `example.com/#home`

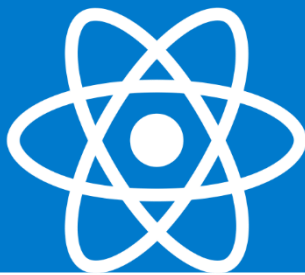


Component

- Router
 - Groups all your routes together, and ensures they take precedence from top-to-bottom.
- Routes
 - Grouping
- Route
 - Each URL will map with Component
- Link
 - If we used the traditional ``, it would make a completely new request and reload the page, so we have Link to help us out
 - `<Link>`

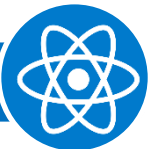


| New Version > 6.0



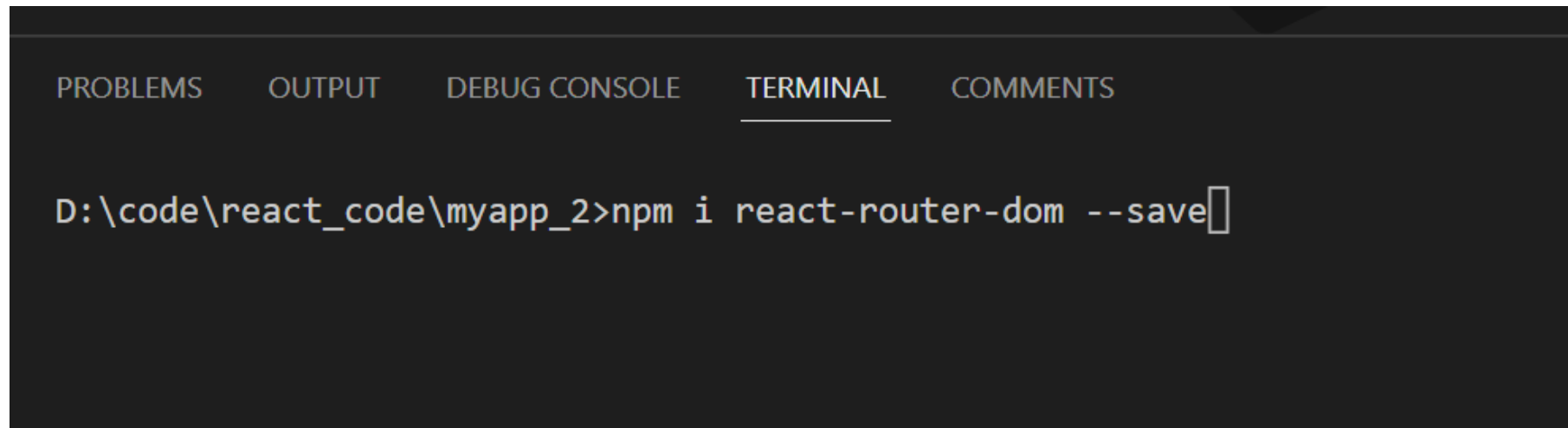
Steps >6.0 React Router DOM

- Create New Project
 - create-react-app react_spa
- Visit :
 - <https://www.npmjs.com/package/react-router-dom>
- Download:
 - npm install --save react-router-dom
- Create 3 New Components
 - Home.js
 - About.js
 - Error.js



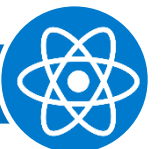
Install Package in React Project

- Npm I react-router-dom --save

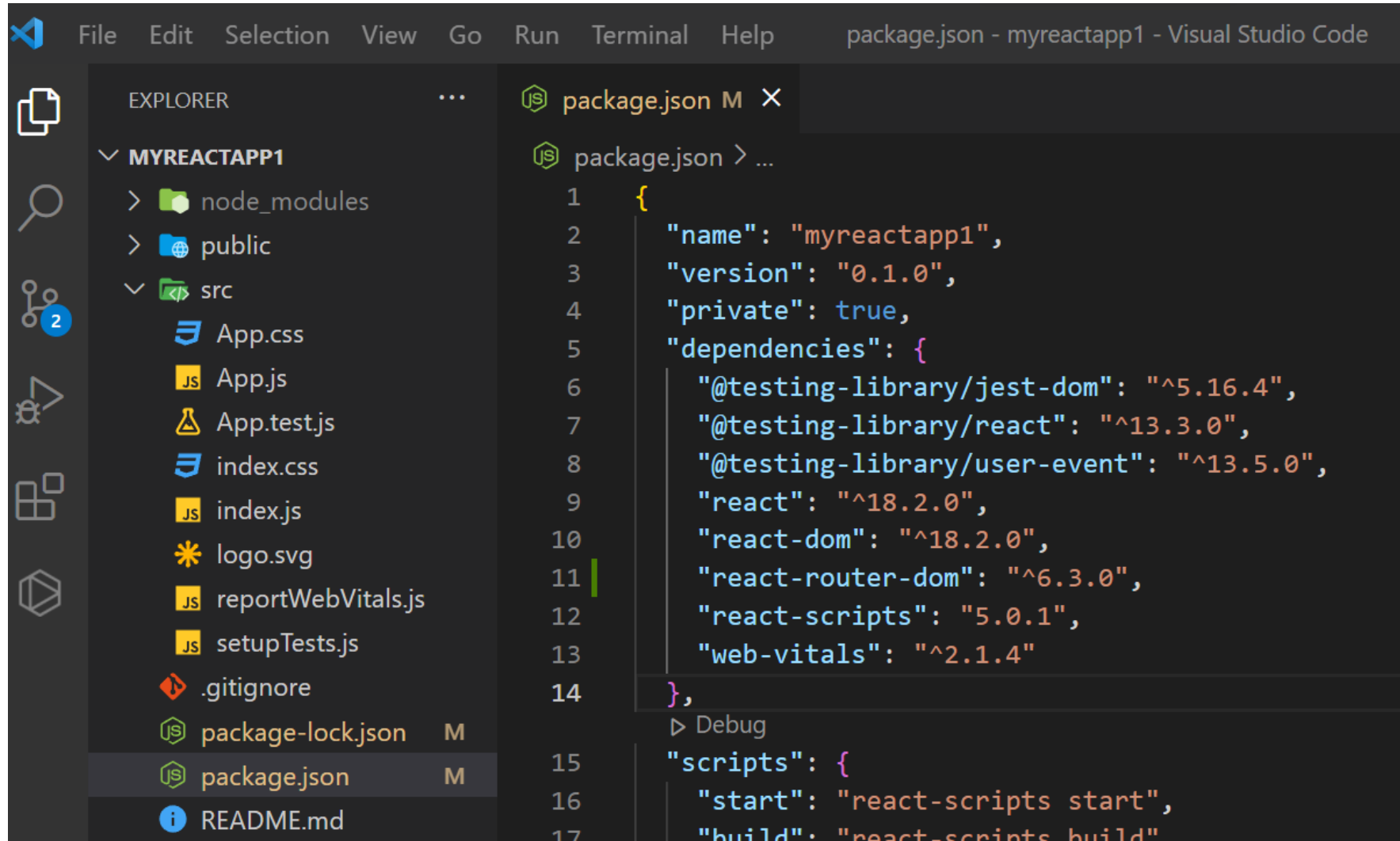


A screenshot of a Visual Studio Code terminal window. The terminal has a dark background with light-colored text. At the top, there is a tab bar with five tabs: 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is selected and underlined), and 'COMMENTS'. Below the tab bar, the terminal shows a command prompt with the text 'D:\code\react_code\myapp_2>' followed by the command 'npm i react-router-dom --save' and a cursor at the end of the line.

```
D:\code\react_code\myapp_2>npm i react-router-dom --save
```



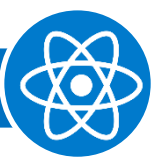
6.3 Version Downloaded



```
package.json - myreactapp1 - Visual Studio Code

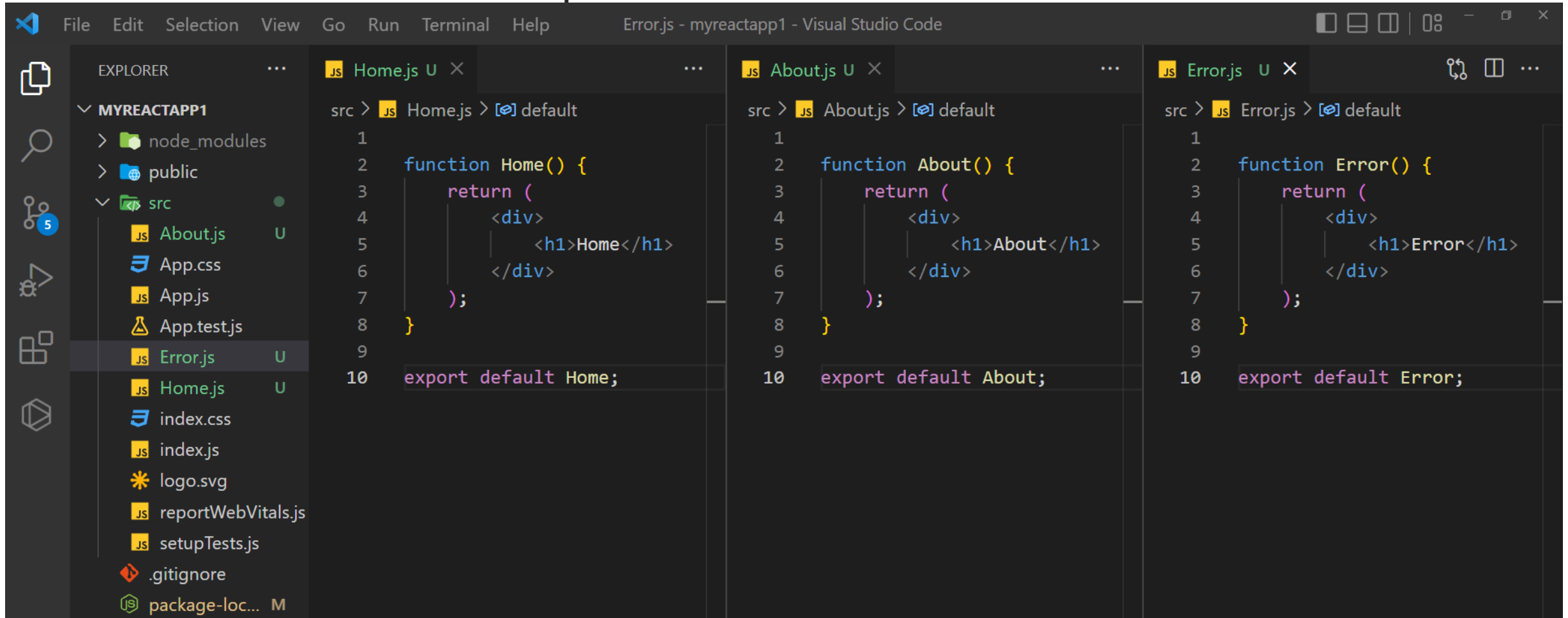
EXPLORER
MYREACTAPP1
  > node_modules
  > public
  < src
    App.css
    App.js
    App.test.js
    index.css
    index.js
    logo.svg
    reportWebVitals.js
    setupTests.js
    .gitignore
    package-lock.json M
    package.json M
    README.md

package.json M X
package.json > ...
1  {
2    "name": "myreactapp1",
3    "version": "0.1.0",
4    "private": true,
5    "dependencies": {
6      "@testing-library/jest-dom": "^5.16.4",
7      "@testing-library/react": "^13.3.0",
8      "@testing-library/user-event": "^13.5.0",
9      "react": "^18.2.0",
10     "react-dom": "^18.2.0",
11     "react-router-dom": "^6.3.0",
12     "react-scripts": "5.0.1",
13     "web-vitals": "^2.1.4"
14   },
15   "scripts": {
16     "start": "react-scripts start",
17     "build": "react-scripts build"
```



Create 3 Component

■ Home About Contact Component



The screenshot displays the Visual Studio Code interface with three files open in the editor: `Home.js`, `About.js`, and `Error.js`. The Explorer sidebar on the left shows the project structure for `MYREACTAPP1`, including `node_modules`, `public`, and a `src` directory containing the three component files. The `Home.js` file defines a `Home` function that returns a JSX element with an `h1>Home` tag. The `About.js` file defines an `About` function that returns a JSX element with an `h1>About` tag. The `Error.js` file defines an `Error` function that returns a JSX element with an `h1>Error` tag. Each file also includes an `export default` statement at the bottom.

```
src > JS Home.js > [🔗] default
1
2 function Home() {
3   return (
4     <div>
5       <h1>Home</h1>
6     </div>
7   );
8 }
9
10 export default Home;
```

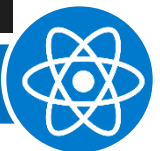
```
src > JS About.js > [🔗] default
1
2 function About() {
3   return (
4     <div>
5       <h1>About</h1>
6     </div>
7   );
8 }
9
10 export default About;
```

```
src > JS Error.js > [🔗] default
1
2 function Error() {
3   return (
4     <div>
5       <h1>Error</h1>
6     </div>
7   );
8 }
9
10 export default Error;
```

Import Browser Router and Component

- import { BrowserRouter as Router, Routes, Route, Link } from "react-router-dom";

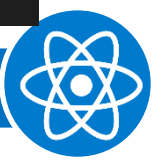
```
src > JS App.js > App
1  import './App.css';
2  import Home from './Home';
3  import About from './About';
4
5  import { BrowserRouter as Router, Routes, Route, Link } from 'react-router-dom';
6
7  function App() {
8    return (
9      <div>
10
11    </div>
12  );
13 }
14
15 export default App;
16
```



Add Routing and Mapping of Component

- Path Mapping with Component

```
JS App.js 1, M X
src > JS App.js > App
1 import './App.css';
2 import Home from './Home';
3 import About from './About';
4 import { BrowserRouter as Router, Routes, Route, Link } from "react-router-dom";
5 function App() {
6   return(
7     <div>
8       <Router>
9         <Routes>
10           <Route path="/" element={<Home/>} />
11           <Route path="/Home" element={<Home/>} />
12           <Route path="/About" element={<About/>} />
13         </Routes>
14       </Router>
15     </div>
16   )
17 }
18 export default App;
```



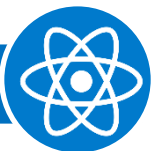
Routing

■ Routing with Link Navigation

```
JS App.js M X
src > JS App.js > ...
1  import './App.css';
2  import Home from './Home';
3  import About from './About';
4  import { BrowserRouter as Router, Routes, Route, Link } from "react-router-dom";
5
6  function App() {
7    return (
8      <div>
9        <Router>
10          <Link to="/Home">Home</Link> |
11          <Link to="/About">About</Link> |
12          <Routes>
13            <Route path="/" element={<Home />} />
14            <Route path="/Home" element={<Home />} />
15            <Route path="/About" element={<About />} />
16          </Routes>
17        </Router>
18      </div>
19    );
20  }
21  export default App;
```

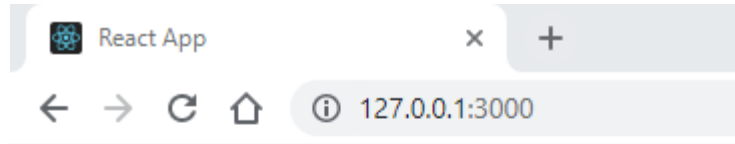
```
import { BrowserRouter as Router, Routes, Route, Link } from "react-router-dom";
```

```
<Router>
  <Link to="/Home">Home</Link> |
  <Link to="/About">About</Link>
  <Routes>
    <Route path="/" element={<Home />} />
    <Route path="/Home" element={<Home />} />
    <Route path="/About" element={<About />} />
  </Routes>
</Router>
```



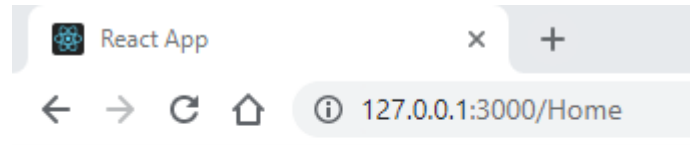
Browser Router Navigation

- <http://127.0.0.1:3000/>
- <http://127.0.0.1:3000/Home>
- <http://127.0.0.1:3000/About>



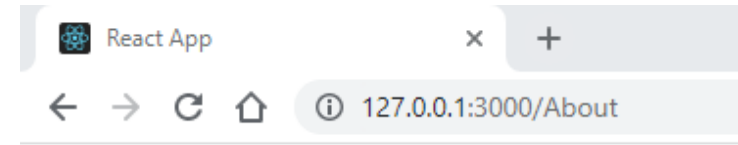
[Home](#) | [About](#) |

Home



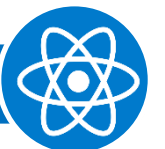
[Home](#) | [About](#) |

Home

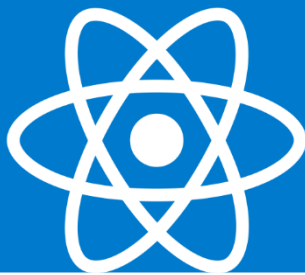


[Home](#) | [About](#) |

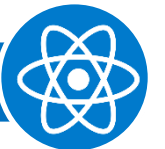
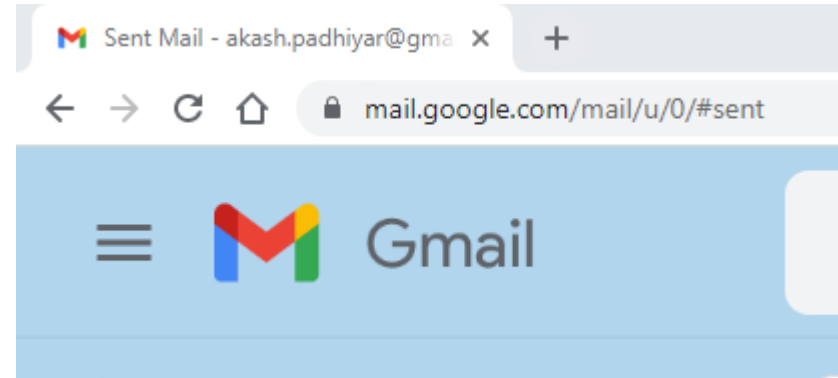
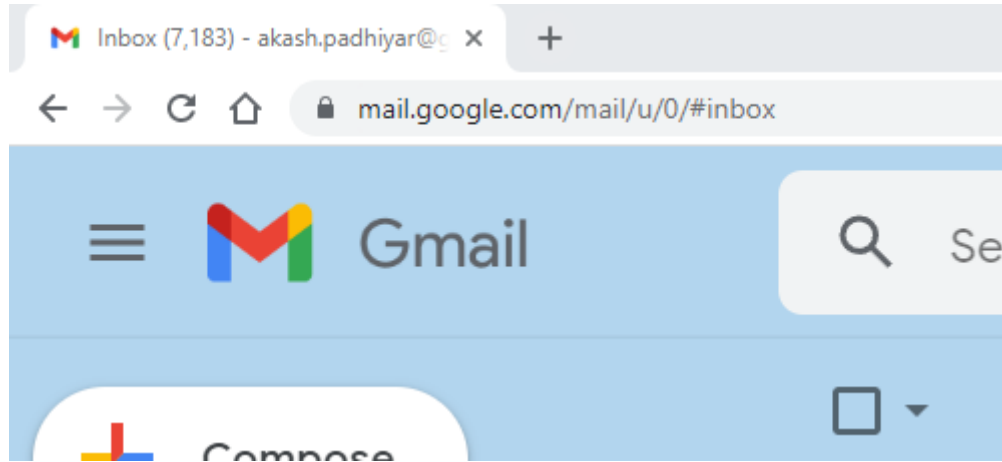
About



| Hash Router



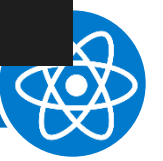
Example



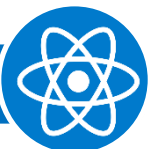
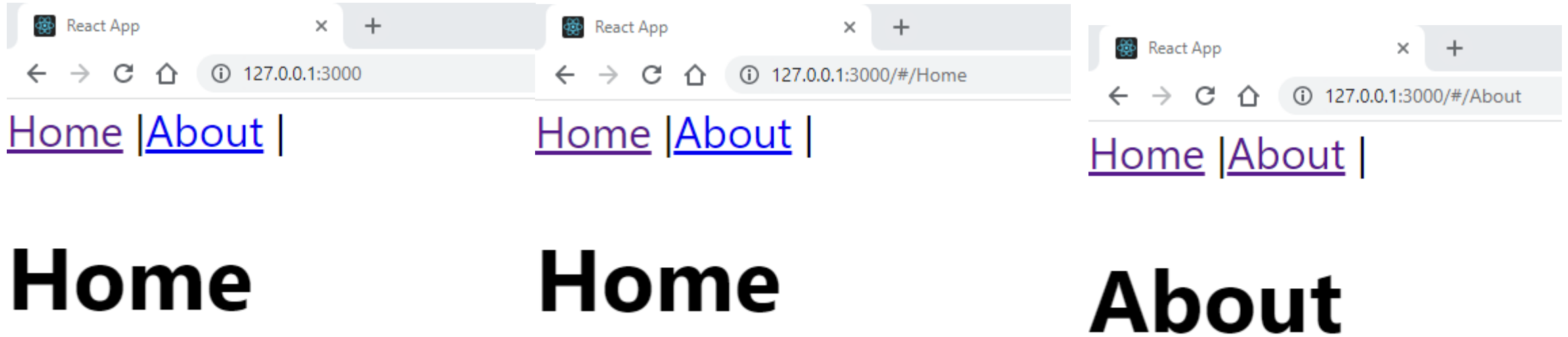
Hash Router

- HashRouter

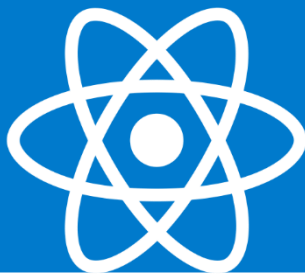
```
JS App.js M X
src > JS App.js > ...
1 | import './App.css';
2 | import Home from './Home';
3 | import About from './About';
4 | import { HashRouter as Router, Routes, Route, Link } from "react-router-dom";
5 |
6 | function App() {
7 |   return (
8 |     <div>
9 |       <Router>
10 |         <Link to="/Home">Home</Link> |
11 |         <Link to="/About">About</Link> |
12 |         <Routes>
13 |           <Route path="/" element={<Home/>}/>
14 |           <Route path="/Home" element={<Home/>}/>
15 |           <Route path="/About" element={<About/>}/>
16 |         </Routes>
17 |       </Router>
18 |     </div>
19 |   );
20 | }
21 | export default App;
22
```



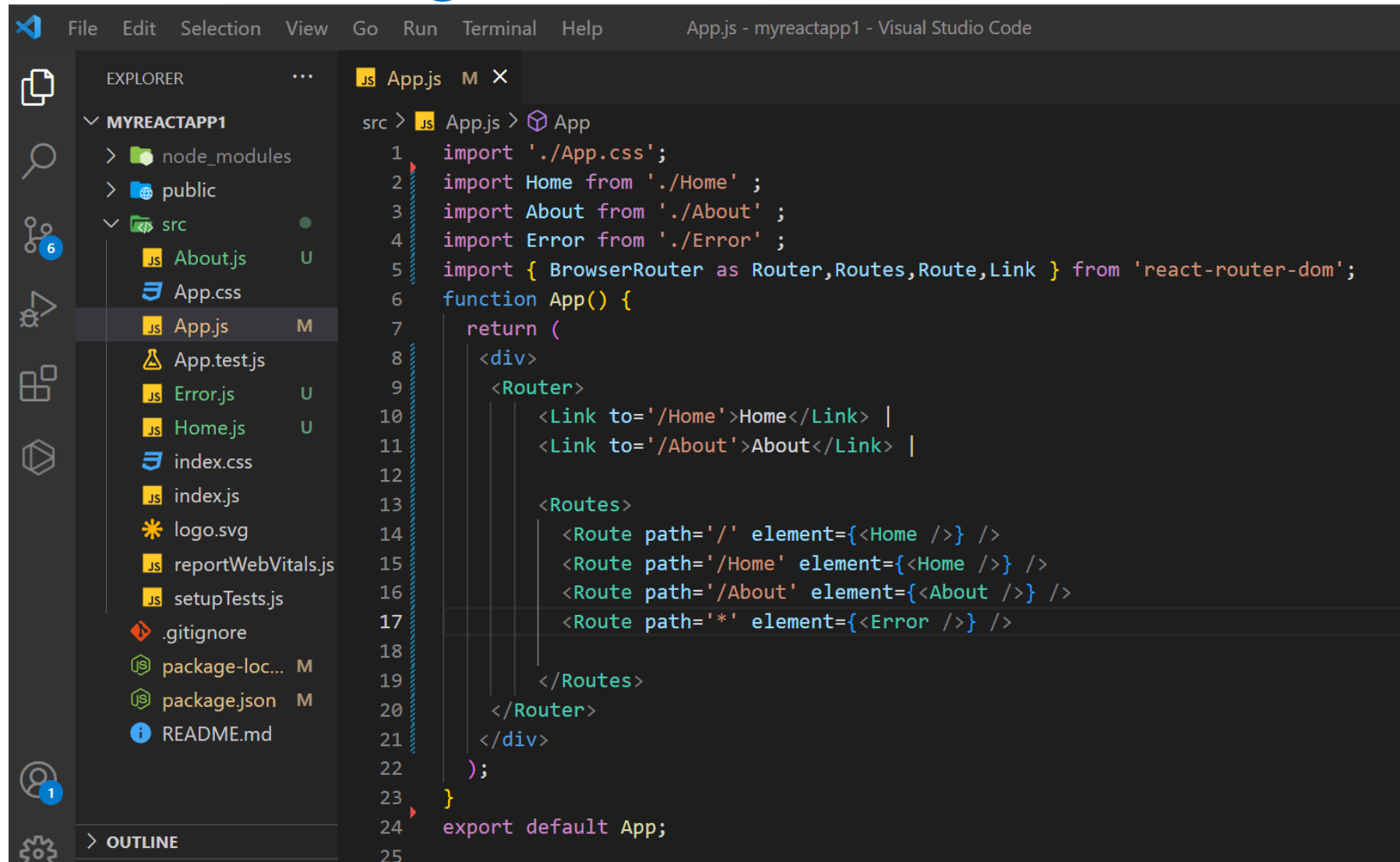
HashRouter Example



| Error Handling

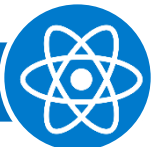


Error Routing * for All URL

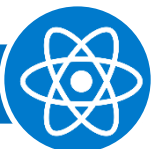
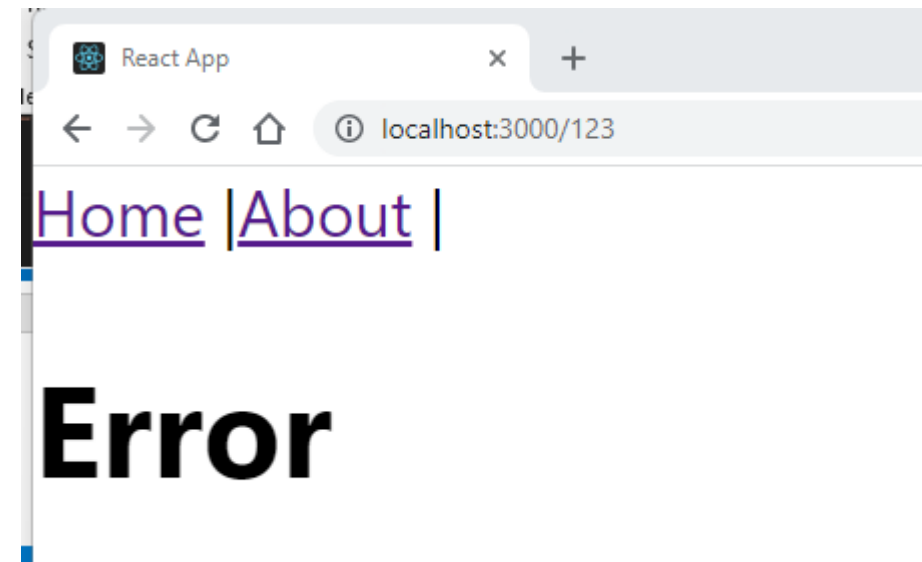


The screenshot displays the Visual Studio Code interface with a React application project named 'MYREACTAPP1'. The Explorer sidebar on the left shows the file structure, including a 'src' directory with files like 'App.js', 'App.css', 'App.test.js', 'Error.js', 'Home.js', 'index.css', 'index.js', 'logo.svg', 'reportWebVitals.js', 'setupTests.js', '.gitignore', 'package-lock.json', 'package.json', and 'README.md'. The 'App.js' file is selected and open in the editor. The code in 'App.js' shows the following structure:

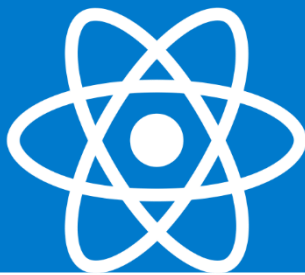
```
1 import './App.css';
2 import Home from './Home';
3 import About from './About';
4 import Error from './Error';
5 import { BrowserRouter as Router, Routes, Route, Link } from 'react-router-dom';
6 function App() {
7   return (
8     <div>
9       <Router>
10         <Link to='/Home'>Home</Link> |
11         <Link to='/About'>About</Link> |
12
13         <Routes>
14           <Route path='/' element={<Home />} />
15           <Route path='/Home' element={<Home />} />
16           <Route path='/About' element={<About />} />
17           <Route path='*' element={<Error />} />
18         </Routes>
19       </Router>
20     </div>
21   );
22 }
23
24 export default App;
```



```
JS Error.js U X
src > JS Error.js > ...
1
2 function Error() {
3   return (
4     <div>
5       <h1>Error</h1>
6     </div>
7   );
8 }
9
10 export default Error;
```



| Error With Button Back Click Event



Error with Back Navigation

JS Error.js U X

src > JS Error.js > ...

```
1 import React from 'react';
2 import {useNavigate} from 'react-router-dom';
3
4 function Error() {
5   let navigate = useNavigate();
6   return (
7     <div>
8       <h1>Error</h1>
9       <button onClick={()=>navigate('/Home')}>Go Back</button>
10    </div>
11  );
12 }
13
14 export default Error;
```

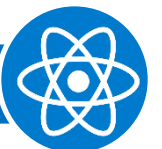
React App

localhost:3000/1111

[Home](#) | [About](#) |

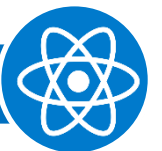
Error

Go Back



Navigate with Hooks

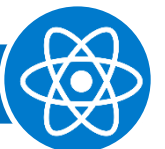
```
JS Page1.js ×
src > JS Page1.js > Page1
1  import {  useNavigate } from 'react-router-dom';
2
3  function Page1() {
4      let navigate = useNavigate();
5
6      function ap_demo(){
7          navigate('/home')
8      }
9      return (
10         <div>
11             <h1>Function Page 1</h1>
12             <button onClick={ap_demo}>Click to Home 1</button>
13
14             <button onClick={()=>navigate('/home')}>Click to Home 2</button>
15         </div>
16     );
17 }
```



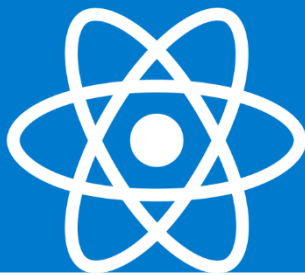
- To go back, like the browser's "Back" button, simply pass -1:
 - `navigate(-1);`
- To go forward, like the browser's "Forward" button, simply pass 1:
 - `navigate(1);`

```
const GoBack = () => {  
  const navigate = useNavigate();  
  const handleClick = () => navigate(-1);  
  
  return (  
    <button type="button" onClick={handleClick}>  
      Go Back ⏮  
    </button>  
  );  
};
```

```
const GoForward = () => {  
  const navigate = useNavigate();  
  const handleClick = () => navigate(1);  
  
  return (  
    <button type="button" onClick={handleClick}>  
      Go Forward ⏭  
    </button>  
  );  
};
```



| Routing Parameter



Contact Component Simple

```
App.js  M X
src > JS App.js > App
1  import './App.css';
2  import Home from './Home';
3  import About from './About';
4  import Contact from './Contact';
5
6  import Error from './Error';
7  import { BrowserRouter as Router, Routes, Route, Link } from 'react-router-dom';
8  function App() {
9    return (
10     <div>
11       <Router>
12         <Link to='/Home'>Home</Link> |
13         <Link to='/About'>About</Link> |
14         <Link to='/Contact'>Contact</Link> |
15         <Routes>
16           <Route path="/" element={<Home />} />
17           <Route path="/Home" element={<Home />} />
18           <Route path="/About" element={<About />} />
19           <Route path="/Contact" element={<Contact />} />
20           <Route path="*" element={<Error />} />
21         </Routes>
22       </Router>
23     </div>
24   );
25 }
26
27 export default App;
28
```

EXPLORER

MYREACTAPP1

- node_modules
- public
- src
 - About.js U
 - App.css
 - App.js M
 - App.test.js
 - Contact.js U
 - Error.js U
 - Home.js U

Contact.js U X

```
src > JS Contact.js > Contact
1
2 function Contact() {
3   return (
4     <div>
5       <h1>Contact</h1>
6     </div>
7   );
8 }
9
10 export default Contact;
```

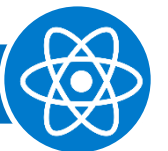
React App

localhost:3000/Contact/123

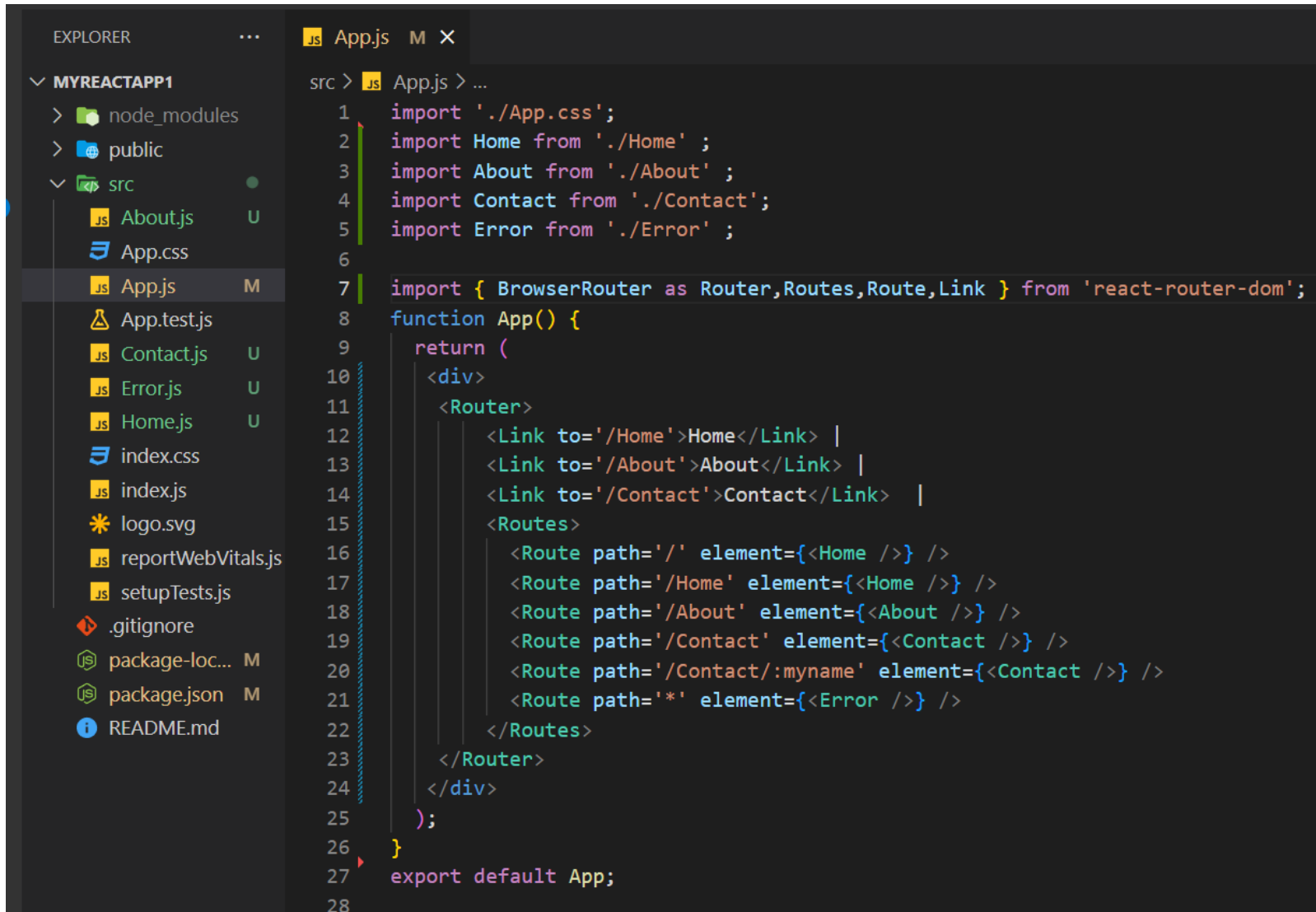
[Home](#) | [About](#) | [Contact](#) |

Error

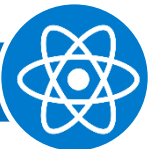
Go Back



Parameter Define

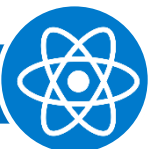


```
1  import './App.css';
2  import Home from './Home';
3  import About from './About';
4  import Contact from './Contact';
5  import Error from './Error';
6
7  import { BrowserRouter as Router, Routes, Route, Link } from 'react-router-dom';
8  function App() {
9    return (
10     <div>
11       <Router>
12         <Link to='/Home'>Home</Link> |
13         <Link to='/About'>About</Link> |
14         <Link to='/Contact'>Contact</Link> |
15         <Routes>
16           <Route path='/' element={<Home />} />
17           <Route path='/Home' element={<Home />} />
18           <Route path='/About' element={<About />} />
19           <Route path='/Contact' element={<Contact />} />
20           <Route path='/Contact/:myname' element={<Contact />} />
21           <Route path='*' element={<Error />} />
22         </Routes>
23       </Router>
24     </div>
25   );
26 }
27 export default App;
28
```

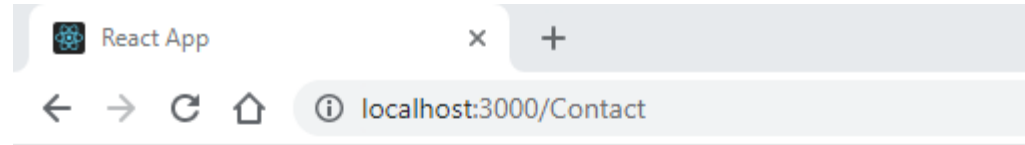


Get Parameter

```
JS Contact.js U X
src > JS Contact.js > [🔗] default
1  import React from 'react';
2  import {useParams} from 'react-router-dom';
3
4  function Contact() {
5      let {myname} = useParams();
6      return (
7          <div>
8              <h1>Contact </h1>
9              <h2>Params is : {myname}</h2>
10         </div>
11     );
12 }
13
14 export default Contact;
```



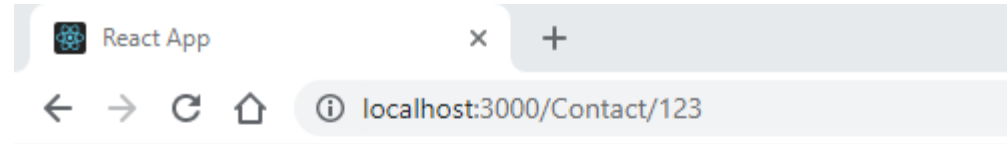
Contact with Parameter



[Home](#) | [About](#) | [Contact](#) |

Contact

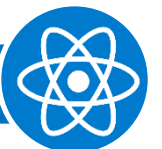
Params is :



[Home](#) | [About](#) | [Contact](#) |

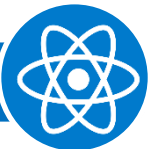
Contact

Params is : 123



Match multiple paths :

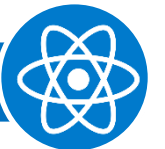
- Path will check URL and According to URL it will load Component
- `<Route path="/(about|who)/" component={Dashboard} />`



Inline rendering

- Instead of specifying a component property on Route, you can set a render prop:

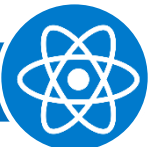
```
<Route
  path="/(about|who)/"
  render={() => (
    <div>
      <h2>About</h2>
      ...
    </div>
  )}
/>
```



Match dynamic route parameter

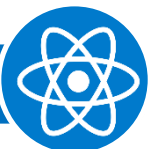
- `<Route exact path="/posts" component={Posts} />`

```
const Posts = () => (  
  <div>  
    <h2>Posts</h2>  
    ...  
  </div>  
)
```

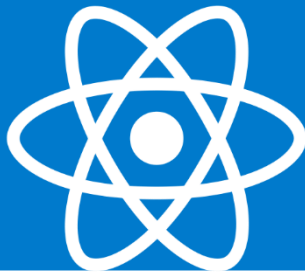


Parameters

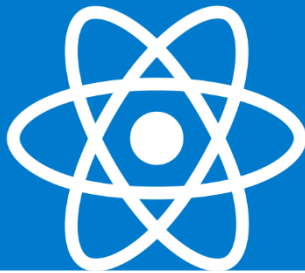
```
const Post = ({match}) => (  
  <div>  
    <h2>Post #{match.params.id}</h2>  
    ...  
  </div>  
)  
  
//...  
  
<Route exact path="/post/:id" component={Post} />
```



| Old Version

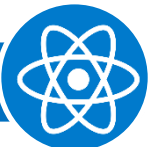


| Old Steps



Component

- Switch
 - Groups all your routes together, and ensures they take precedence from top-to-bottom.
- Route
 - Each individual route.
- Link
 - If we used the traditional ``, it would make a completely new request and reload the page, so we have Link to help us out
 - `<Link>`



Old Version Steps

- Create New Project
 - create-react-app react_spa
- Visit :
 - <https://www.npmjs.com/package/react-router-dom>
- Download 5.2 Version :
 - npm install --save react-router-dom@5.2
- Create 2 New Components
 - Home.js
 - About.js



React App

react-router-dom - npm

npmjs.com/package/react-router-dom

Readme

Explore BETA

7 Dependencies

11,839 Dependents

47 Versions

react-router-dom

DOM bindings for **React Router**.

Installation

Using **npm**:

```
$ npm install --save react-router-dom
```

Then with a module bundler like **webpack**, use as you would anything else:

```
// using ES6 modules
import { BrowserRouter, Route, Link } from "react-router-dom";

// using CommonJS modules
const BrowserRouter = require("react-router-dom").BrowserRouter;
const Route = require("react-router-dom").Route;
const Link = require("react-router-dom").Link;
```

The UMD build is also available on **unpkg**:

```
<script src="https://unpkg.com/react-router-dom/umd/react-router-dom.min.js"></script>
```

Install

```
> npm i react-router-dom
```

Weekly Downloads

4,260,160

Version

5.2.0

License

MIT

Unpacked Size

632 kB

Total Files

50

Issues

45

Pull Requests

22

Homepage

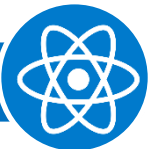
[github.com/ReactTraining/react-router...](https://github.com/ReactTraining/react-router-dom)

Repository

[github.com/ReactTraining/react-router](https://github.com/ReactTraining/react-router-dom)

Last publish

a year ago



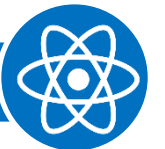
Home.js & About.js

Home.js

```
import React, { Component } from 'react';  
class Home extends Component{  
  render()  
  {  
    return(  
      <p> I am Home Page</p>  
    )  
  }  
}  
export default Home;
```

About.js

```
import React, { Component } from 'react';  
class About extends Component{  
  render()  
  {  
    return(  
      <p>I am About us</p>  
    )  
  }  
}  
export default About;
```



Browser Router Navigation

```
import React, { Component } from 'react';
```

```
import { BrowserRouter as Router, Switch, Route, Link } from  
'react-router-dom';
```

```
import './App.css';
```

```
import Home from './Home';
```

```
import About from './About';
```

```
class App extends Component {
```

```
  render() {
```

```
    return (
```

```
      <Router>
```

```
        <div>
```

```
          <ul>
```

```
            <li><Link to={'/'}>Home</Link></li>
```

```
            <li> <Link to={'/about'}>About</Link></li>
```

```
          </ul>
```

```
        <Switch>
```

```
          <Route exact path='/' component={Home} />
```

```
          <Route exact path='/about' component={About} />
```

```
        </Switch>
```

```
      </div>
```

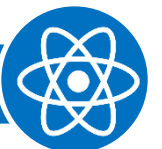
```
    </Router>
```

```
  );
```

```
}
```

```
}
```

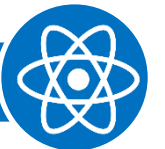
```
export default App;
```



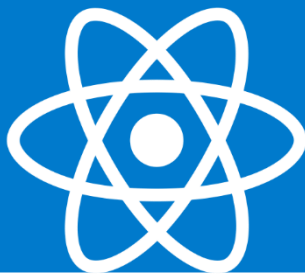
HashRouter Navigation

```
import React, { Component } from "react";  
  
import {Route,NavLink,HashRouter} from "react-router-dom";  
  
import Home from './Home';  
import About from './About';  
  
class Main extends Component {
```

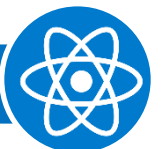
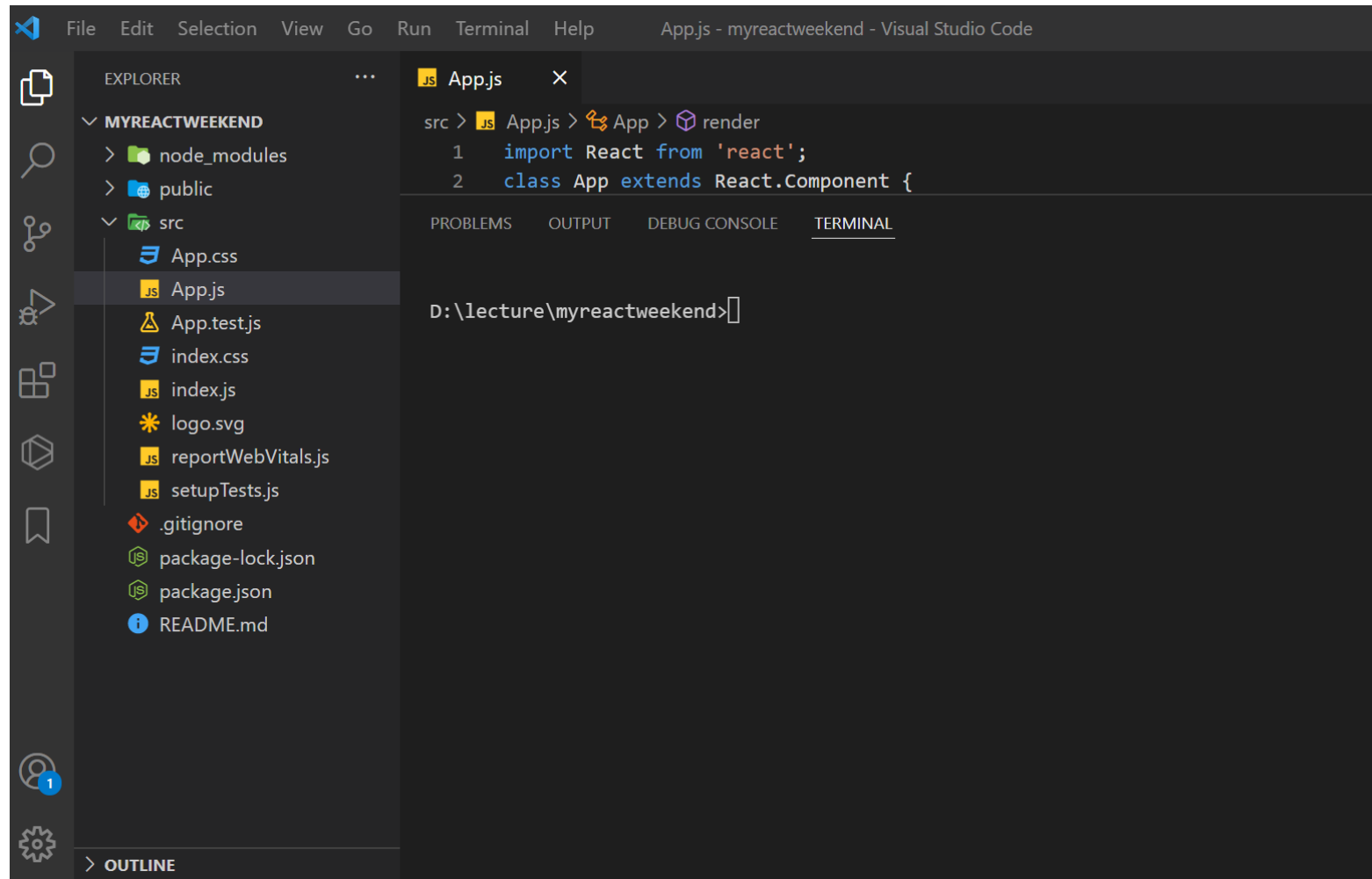
```
  render() {  
    return (  
      <div>  
        <HashRouter>  
          <h1>Simple SPA</h1>  
          <ul>  
            <li><NavLink to="/Home">Home</NavLink></li>  
            <li><NavLink to="/About">About</NavLink></li>  
          </ul>  
          <div>  
            <Route path='/home' component={Home} />  
            <Route path='/about' component={About} />  
          </div>  
        </HashRouter>  
      </div>  
    );  
  }  
}  
  
export default Main;
```



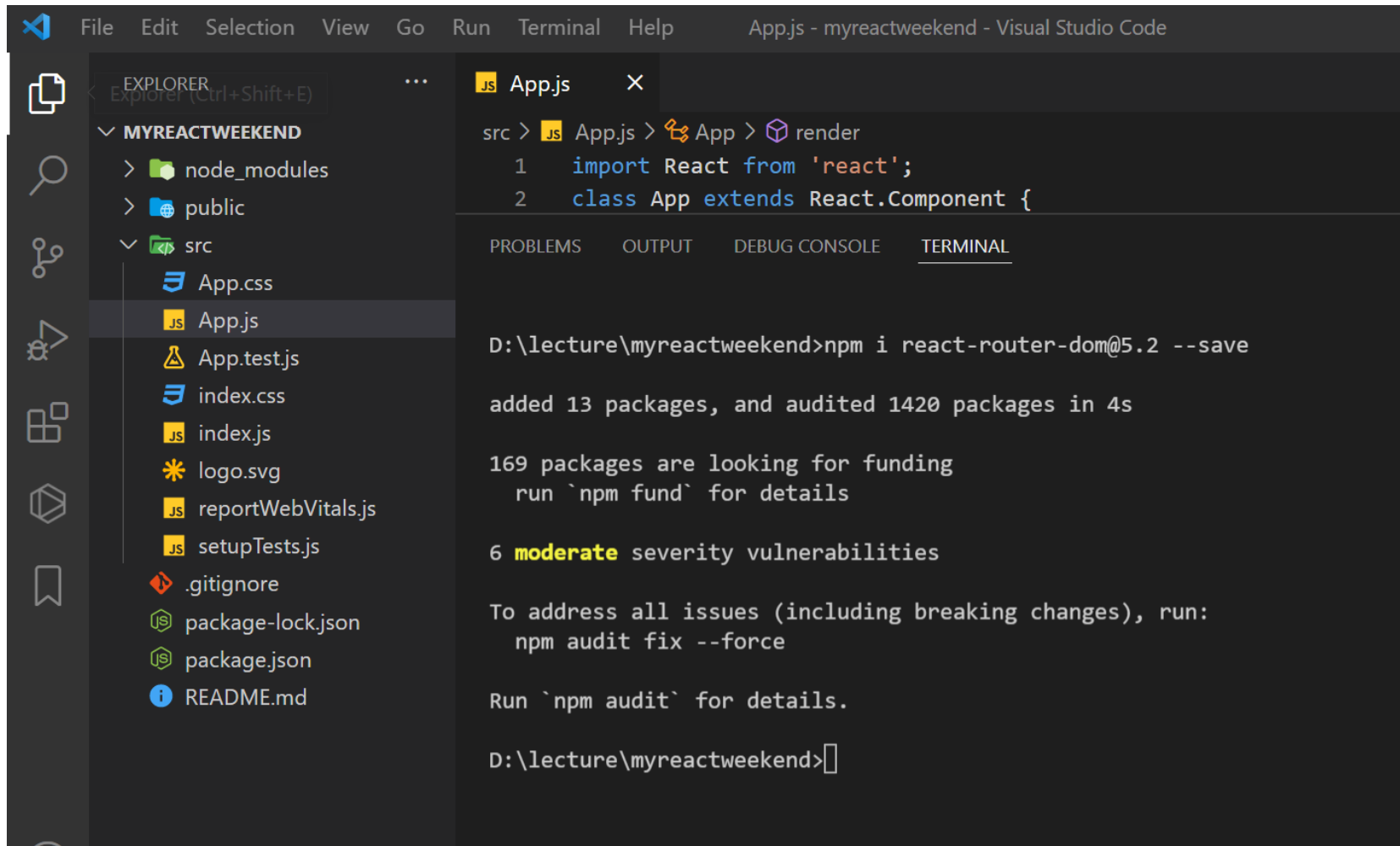
| Demo



Create New Project & Install React Router Dom



Install React Router Dom



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left displaying the file structure of a project named 'MYREACTWEEKEND'. The 'src' folder is expanded, showing files like 'App.css', 'App.js', 'App.test.js', 'index.css', 'index.js', 'logo.svg', 'reportWebVitals.js', 'setupTests.js', '.gitignore', 'package-lock.json', 'package.json', and 'README.md'. The 'App.js' file is selected. The main editor area shows the code for 'App.js' with the following content:

```
src > JS App.js > App > render
1 import React from 'react';
2 class App extends React.Component {
```

Below the code editor, the TERMINAL panel is active, showing the output of the command `npm i react-router-dom@5.2 --save` executed in the directory `D:\lecture\myreactweekend`. The output indicates that 13 packages were added and 1420 packages were audited in 4 seconds. It also mentions that 169 packages are looking for funding and that there are 6 moderate severity vulnerabilities. The terminal output is as follows:

```
D:\lecture\myreactweekend>npm i react-router-dom@5.2 --save

added 13 packages, and audited 1420 packages in 4s

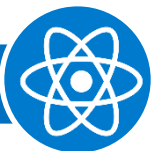
169 packages are looking for funding
  run `npm fund` for details

6 moderate severity vulnerabilities

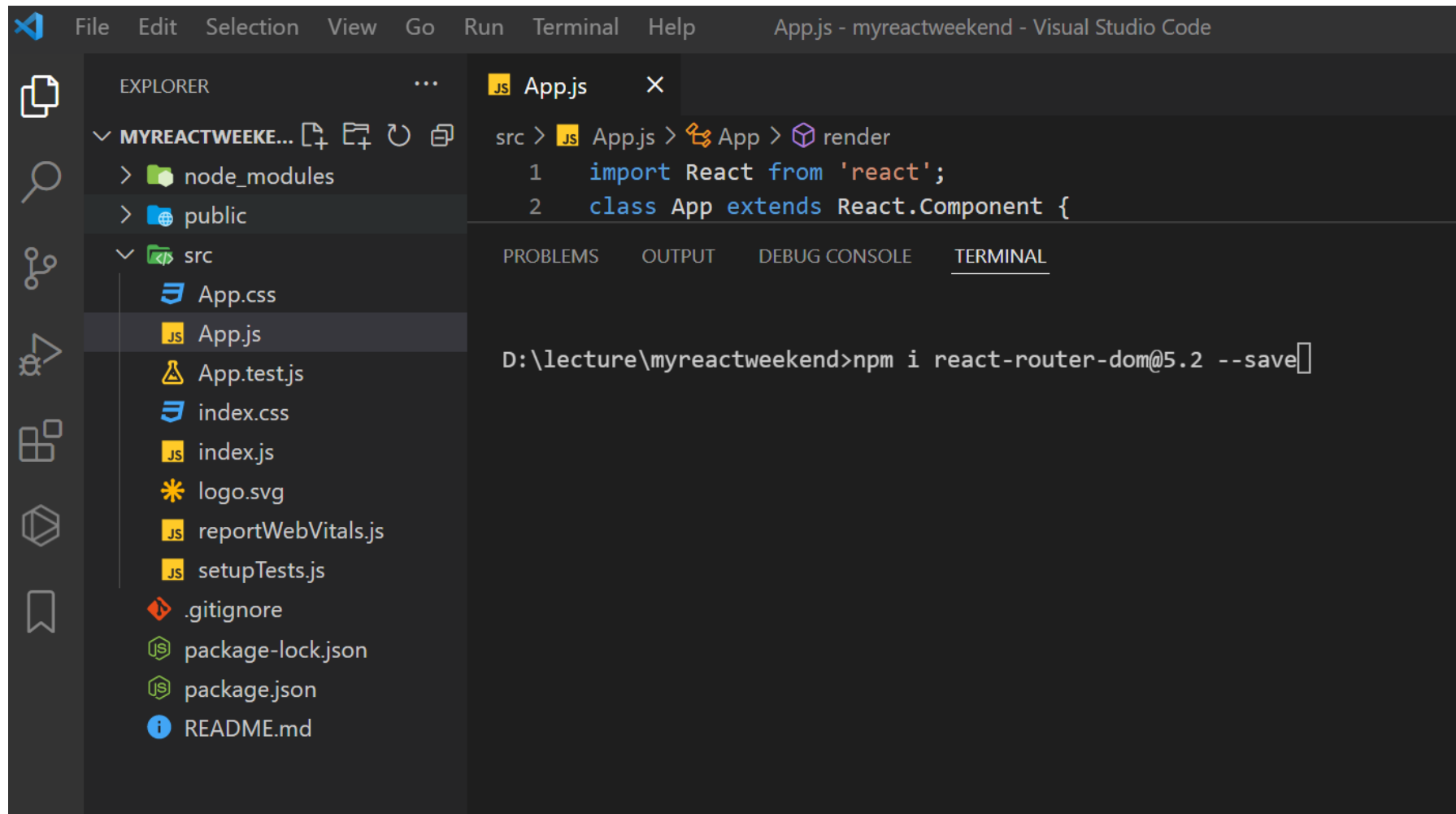
To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.

D:\lecture\myreactweekend>
```



Install React Router Dom 5.2

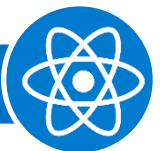


The screenshot shows the Visual Studio Code interface with the following details:

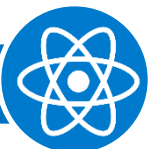
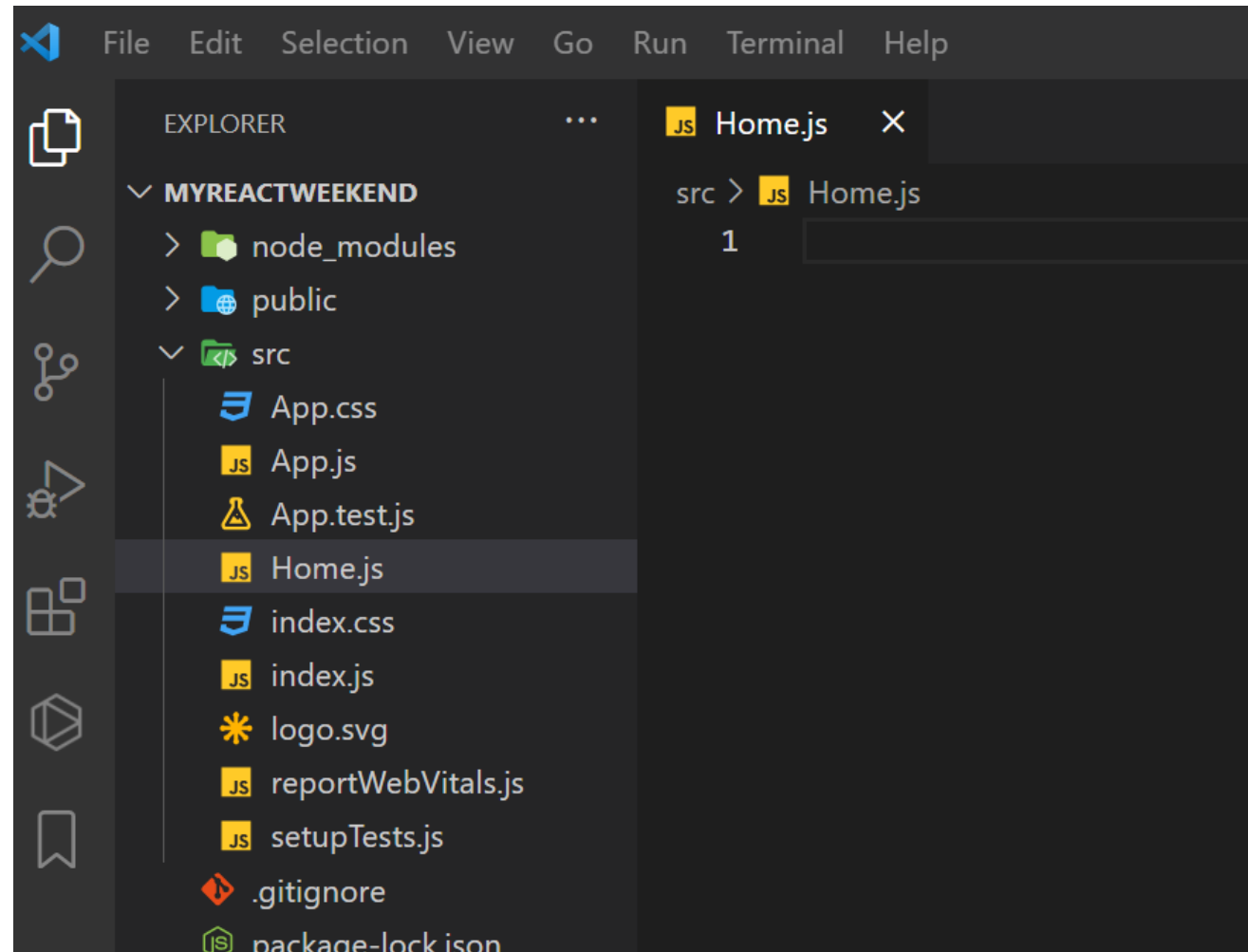
- Explorer Panel:** Displays the file structure of a project named 'MYREACTWEEKE...'. The 'src' folder is expanded, showing files: App.css, App.js (selected), App.test.js, index.css, index.js, logo.svg, reportWebVitals.js, setupTests.js, .gitignore, package-lock.json, package.json, and README.md.
- Editor Panel:** Shows the 'App.js' file with the following code:

```
src > JS App.js > App > render
1  import React from 'react';
2  class App extends React.Component {
```
- Terminal Panel:** Shows the command being executed in the terminal:

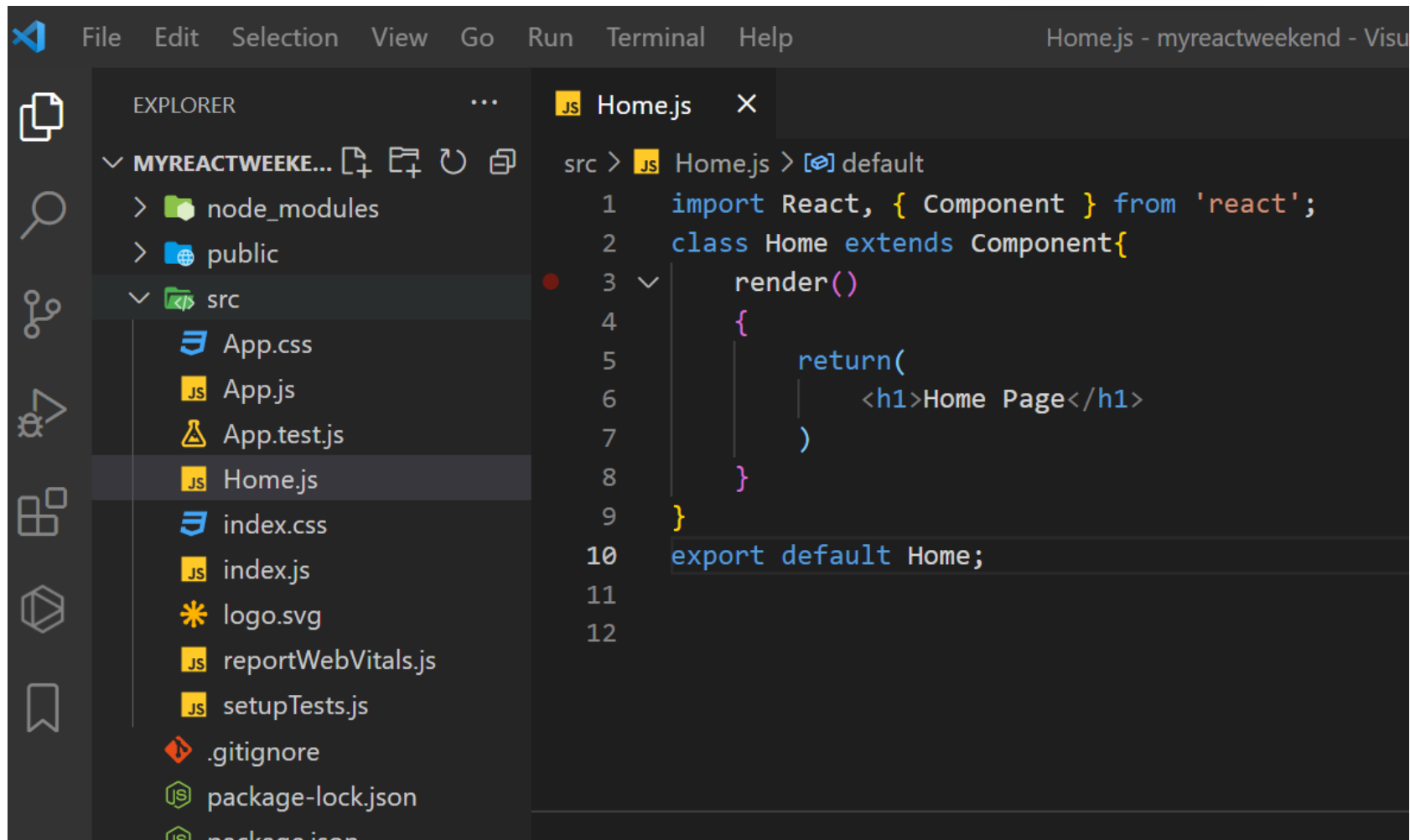
```
D:\lecture\myreactweekend>npm i react-router-dom@5.2 --save
```



Home.js

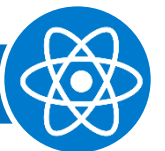


Home.js Component

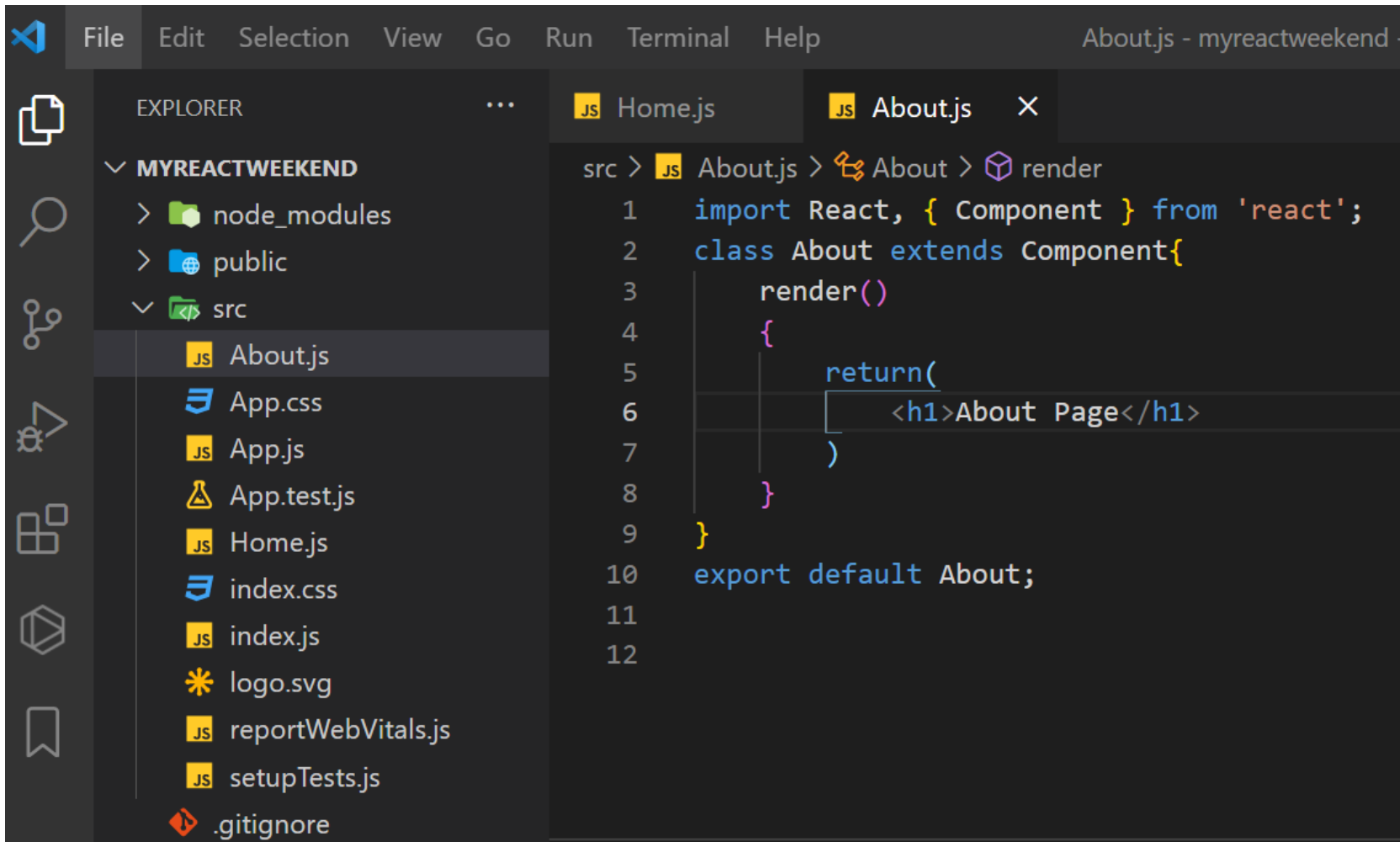


The screenshot shows the Visual Studio Code editor with the 'Home.js' file open. The Explorer sidebar on the left shows the project structure with the 'src' folder expanded, highlighting 'Home.js'. The main editor area displays the following code:

```
src > JS Home.js > [default]
1  import React, { Component } from 'react';
2  class Home extends Component{
3    render()
4    {
5      return(
6        <h1>Home Page</h1>
7      )
8    }
9  }
10 export default Home;
```

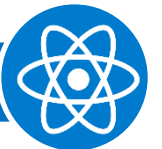


About.js Component

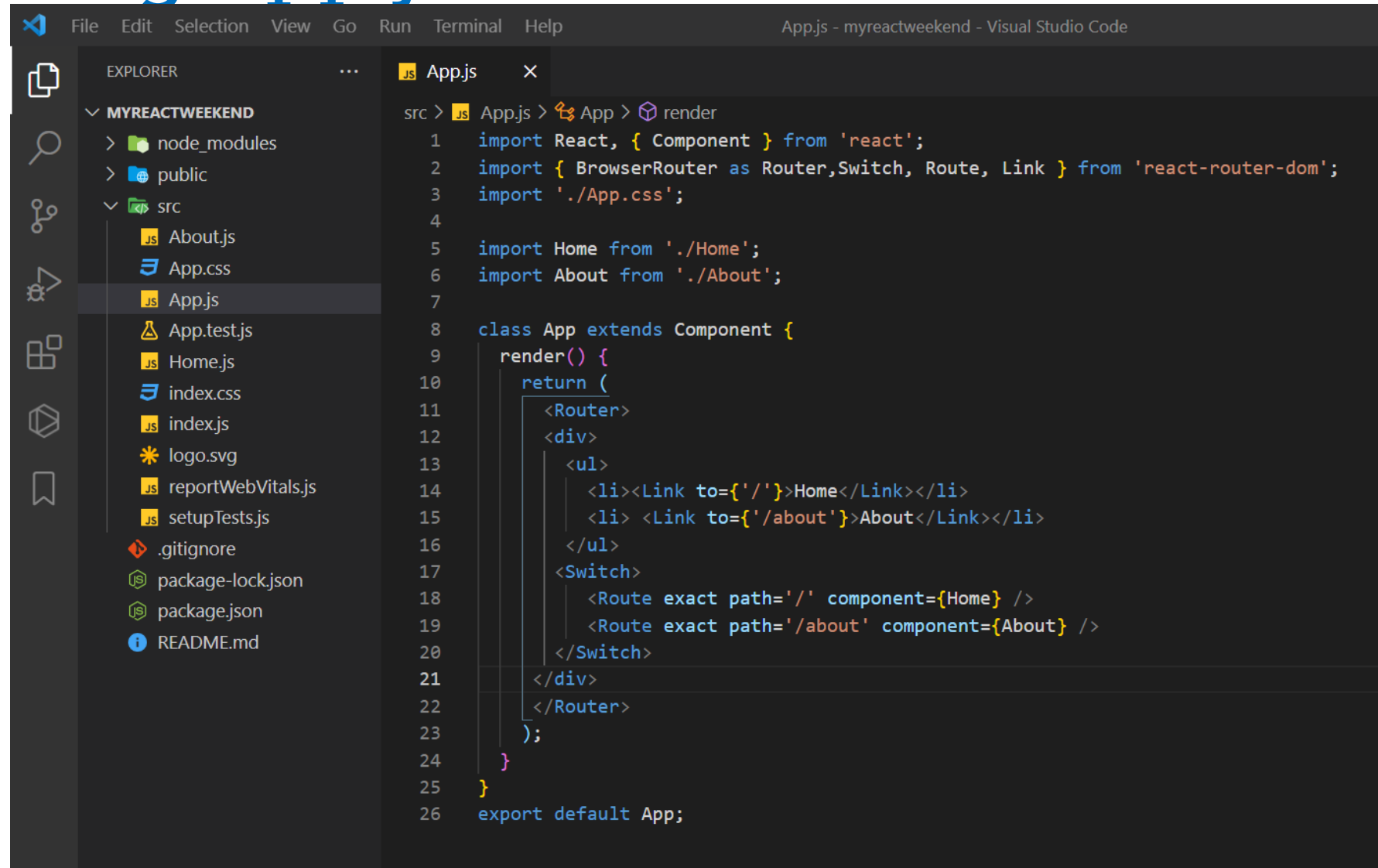


The screenshot shows the Visual Studio Code editor with the 'About.js' file open. The Explorer sidebar on the left shows the project structure: 'MYREACTWEEKEND' with subfolders 'node_modules', 'public', and 'src'. The 'src' folder is expanded, showing files like 'About.js', 'App.css', 'App.js', 'App.test.js', 'Home.js', 'index.css', 'index.js', 'logo.svg', 'reportWebVitals.js', 'setupTests.js', and '.gitignore'. The 'About.js' file is selected. The main editor area shows the code for the 'About' component, which extends 'Component' and has a 'render' method that returns a single JSX element: a heading 'About Page'.

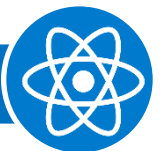
```
File Edit Selection View Go Run Terminal Help About.js - myreactweekend -  
  
EXPLORER  
▼ MYREACTWEEKEND  
  > node_modules  
  > public  
  ▼ src  
    JS About.js  
    App.css  
    JS App.js  
    App.test.js  
    JS Home.js  
    index.css  
    JS index.js  
    logo.svg  
    JS reportWebVitals.js  
    JS setupTests.js  
    .gitignore  
  
src > JS About.js > About > render  
1  import React, { Component } from 'react';  
2  class About extends Component{  
3      render()  
4      {  
5          return(  
6              <h1>About Page</h1>  
7          )  
8      }  
9  }  
10 export default About;  
11  
12
```



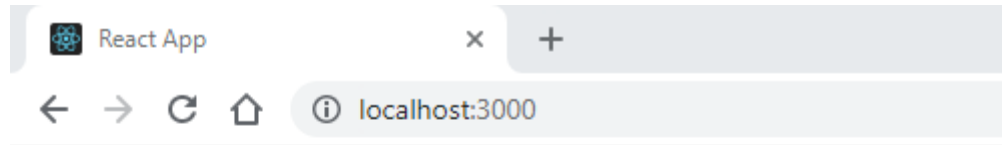
Routing App.js



```
src > JS App.js > App > render
1  import React, { Component } from 'react';
2  import { BrowserRouter as Router, Switch, Route, Link } from 'react-router-dom';
3  import './App.css';
4
5  import Home from './Home';
6  import About from './About';
7
8  class App extends Component {
9    render() {
10     return (
11       <Router>
12         <div>
13           <ul>
14             <li><Link to={'/'}>Home</Link></li>
15             <li><Link to={'/about'}>About</Link></li>
16           </ul>
17           <Switch>
18             <Route exact path='/' component={Home} />
19             <Route exact path='/about' component={About} />
20           </Switch>
21         </div>
22       </Router>
23     );
24   }
25 }
26 export default App;
```

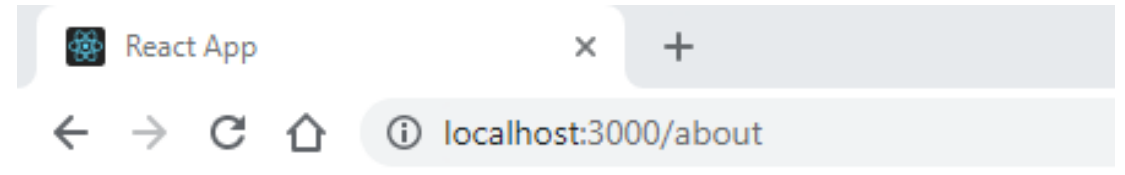


Browser Router Navigation



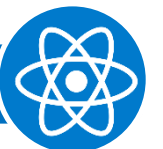
- [Home](#)
- [About](#)

Home Page

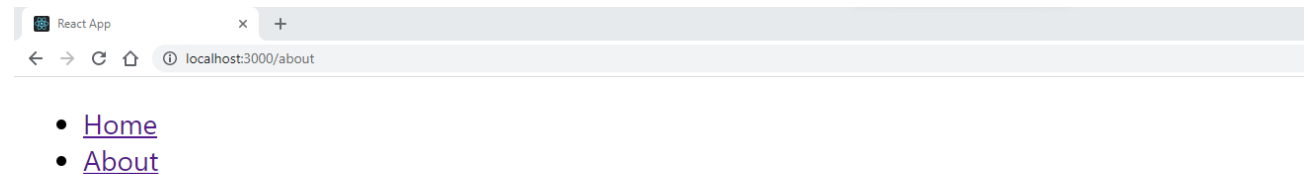


- [Home](#)
- [About](#)

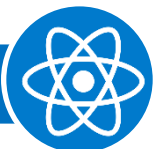
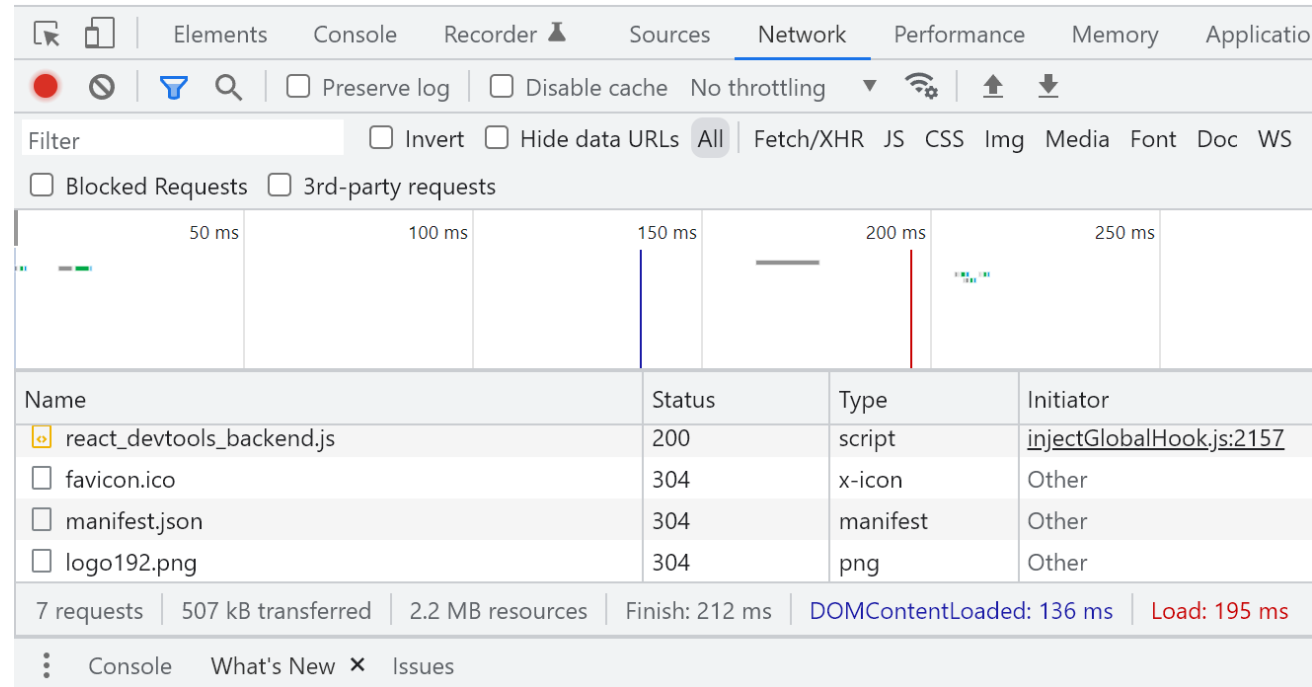
About Page



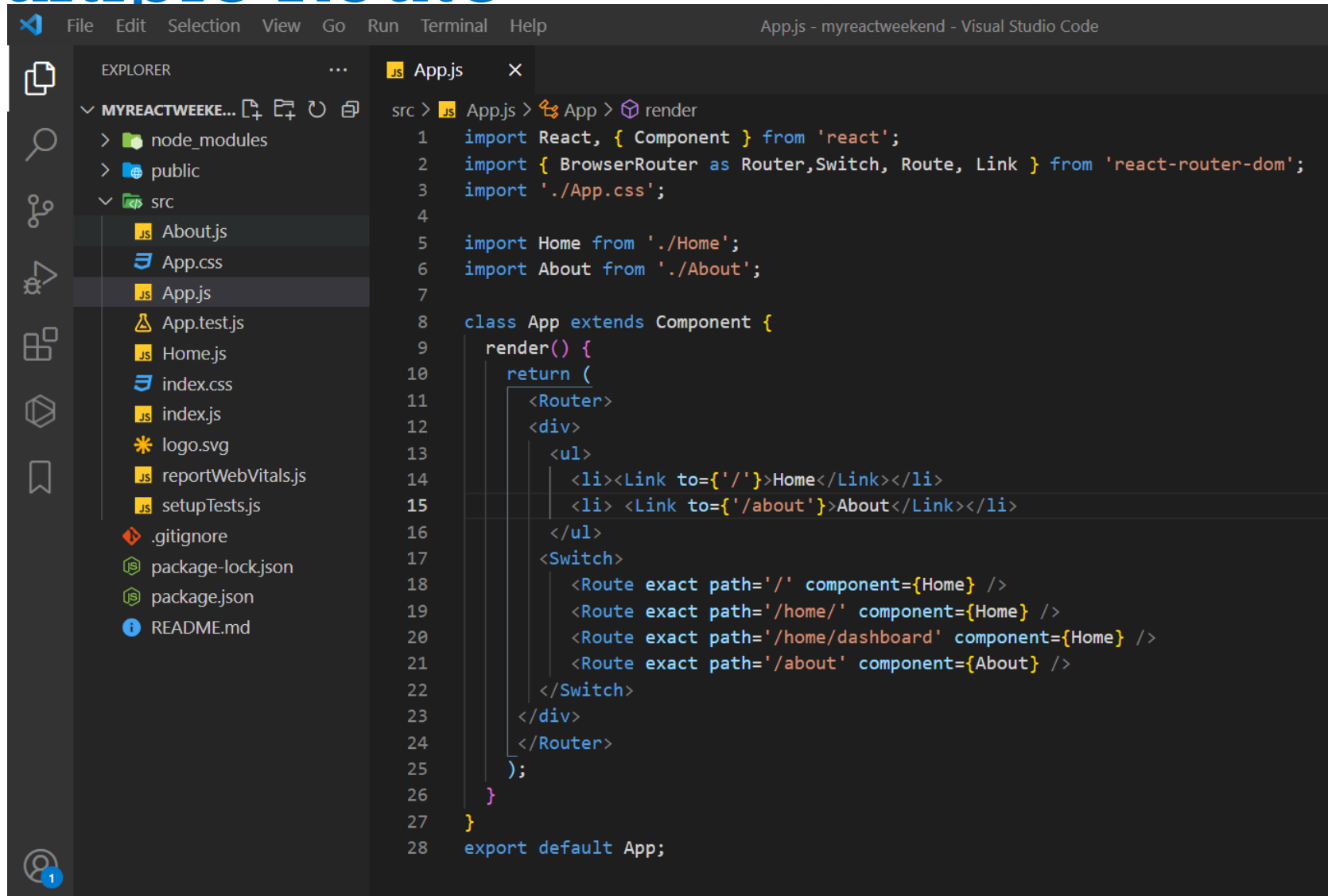
Loading Time will be same for all Component



About Page

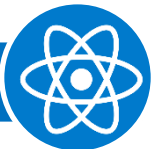


Multiple Route

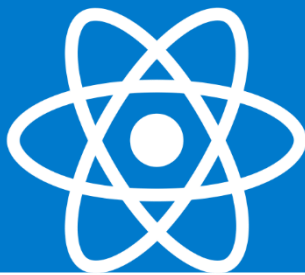


The screenshot shows a Visual Studio Code editor window titled 'App.js - myreactweekend - Visual Studio Code'. The Explorer sidebar on the left shows a project structure with a 'src' folder containing files like About.js, App.css, App.js, App.test.js, Home.js, index.css, index.js, logo.svg, reportWebVitals.js, setupTests.js, .gitignore, package-lock.json, package.json, and README.md. The main editor area displays the code for App.js, which imports React, ReactDOM, and react-router-dom, and defines a class App with a render method that uses Router, Switch, Route, and Link to create a multi-page application.

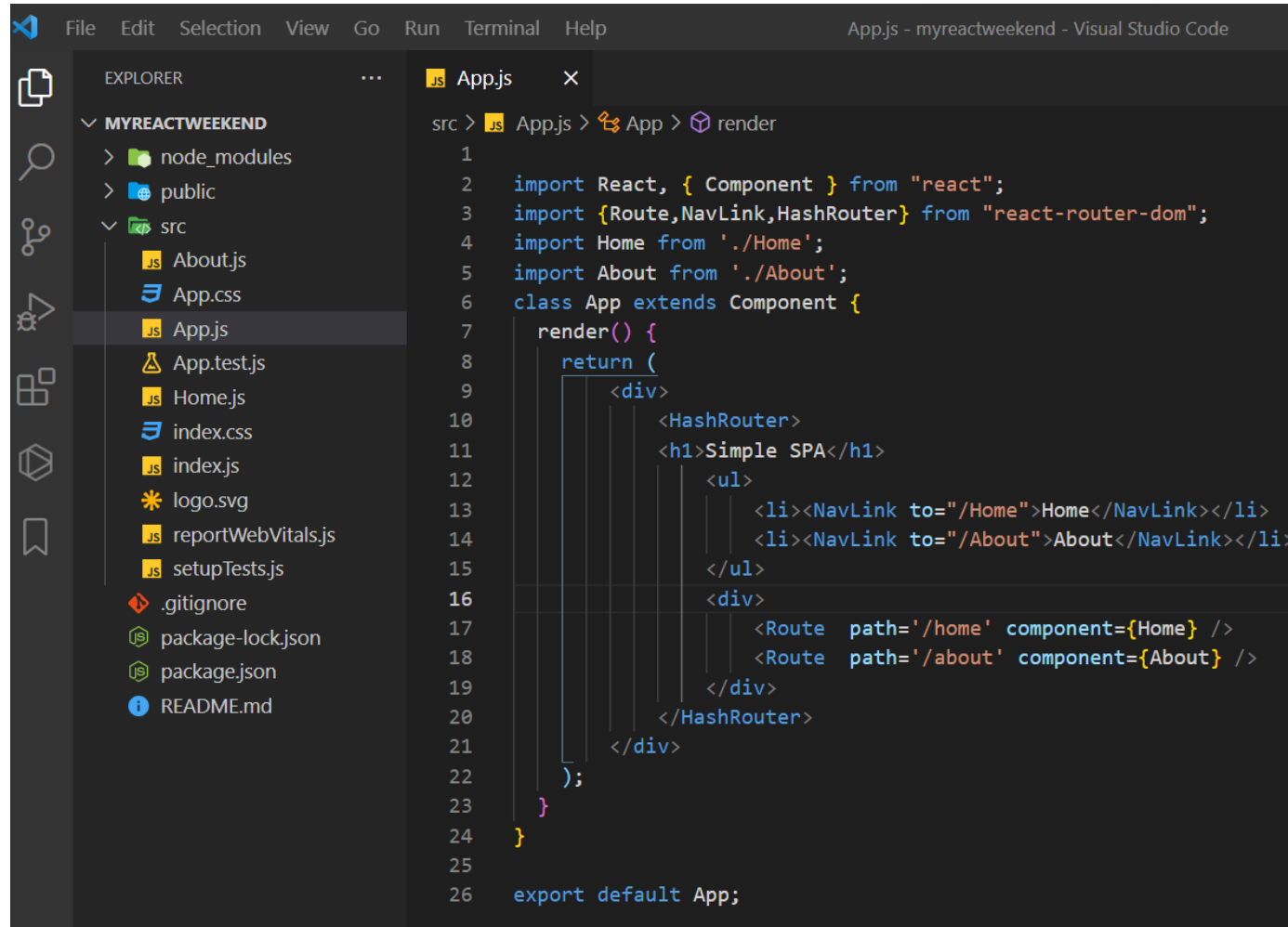
```
src > JS App.js > App > render
1  import React, { Component } from 'react';
2  import { BrowserRouter as Router, Switch, Route, Link } from 'react-router-dom';
3  import './App.css';
4
5  import Home from './Home';
6  import About from './About';
7
8  class App extends Component {
9    render() {
10     return (
11       <Router>
12         <div>
13           <ul>
14             <li><Link to={'/'}>Home</Link></li>
15             <li><Link to={'/about'}>About</Link></li>
16           </ul>
17           <Switch>
18             <Route exact path="/" component={Home} />
19             <Route exact path="/home/" component={Home} />
20             <Route exact path="/home/dashboard" component={Home} />
21             <Route exact path="/about" component={About} />
22           </Switch>
23         </div>
24       </Router>
25     );
26   }
27 }
28 export default App;
```



| HashRouter



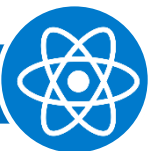
HashRouter



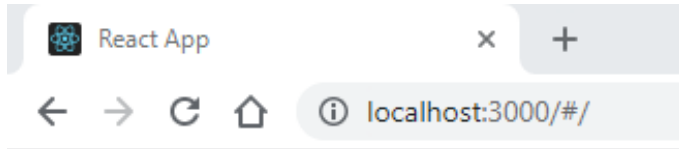
```
File Edit Selection View Go Run Terminal Help App.js - myreactweekend - Visual Studio Code

EXPLORER
MYREACTWEEKEND
  node_modules
  public
  src
    About.js
    App.css
    App.js
    App.test.js
    Home.js
    index.css
    index.js
    logo.svg
    reportWebVitals.js
    setupTests.js
  .gitignore
  package-lock.json
  package.json
  README.md

App.js
src > App.js > App > render
1
2 import React, { Component } from "react";
3 import {Route,NavLink,HashRouter} from "react-router-dom";
4 import Home from './Home';
5 import About from './About';
6 class App extends Component {
7   render() {
8     return (
9       <div>
10         <HashRouter>
11           <h1>Simple SPA</h1>
12           <ul>
13             <li><NavLink to="/Home">Home</NavLink></li>
14             <li><NavLink to="/About">About</NavLink></li>
15           </ul>
16           <div>
17             <Route path="/home" component={Home} />
18             <Route path="/about" component={About} />
19           </div>
20         </HashRouter>
21       </div>
22     );
23   }
24 }
25
26 export default App;
```

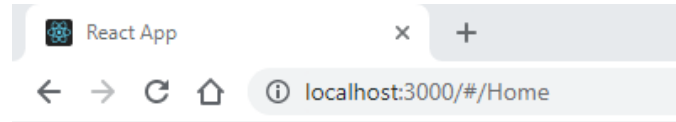


HashRouter Output



Simple SPA

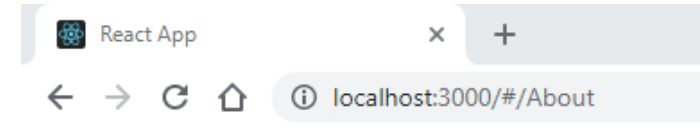
- [Home](#)
- [About](#)



Simple SPA

- [Home](#)
- [About](#)

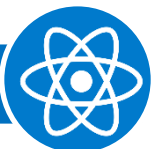
Home Page



Simple SPA

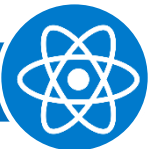
- [Home](#)
- [About](#)

About Page



Task

- Preform Same Program
- Create Website and Create 3 Component Home,About,Contact
 - Home : Basic Content
 - About : Basic Contact & Images
 - Contact : Contact Form



Get Exclusive Video Tutorials



www.apptutorials.com

<https://www.youtube.com/user/Akashtips>





Get More Details

www.akashsir.com



If You Liked It !

Rating Us Now



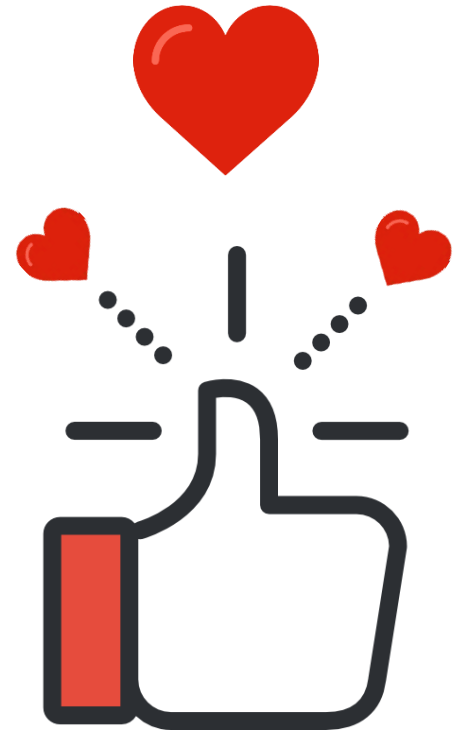
Just Dial

https://www.justdial.com/Ahmedabad/Akash-Technolabs-Navrangpura-Bus-Stop-Navrangpura/079PXX79-XX79-170615221520-S5C4_BZDET



Sulekha

<https://www.sulekha.com/akash-technolabs-navrangpura-ahmedabad-contact-address/ahmedabad>



Connect With Me



Akash Padhiyar
#AkashSir

www.akashsir.com

www.akashtechlabs.com

www.akashpadhiyar.com

www.apptutorials.com

Social Info



Akash.padhiyar



Akashpadhiyar



Akash_padhiyar



+91 99786-21654



#Akashpadhiyar

#apptutorials