

LAB ASSIGNMENT NO – 7a

Aim: Write a program to implement ReactJS Router. Create more than two class or functional components and implement program for Routing using browser router.

Theory:

React Router is a popular library for implementing client-side routing in React applications. It allows you to create single-page applications (SPAs) with multiple views or pages without the need for full-page refreshes. Here's a brief overview of React Router:

Routing Components:

React Router provides a set of components to define the routing structure of your application. The primary components include:

`<BrowserRouter>`: Wraps your entire application and provides routing functionality.

`<Route>`: Defines a route and associates it with a specific component to render when the URL matches the route.

`<Switch>`: Renders the first `<Route>` or `<Redirect>` that matches the current URL, helping you avoid multiple matches.

`<Link>` and `<NavLink>`: Create links to navigate between routes without causing a full page reload.

Route Matching:

Routes in React Router can match based on the exact URL path or using dynamic parameters (URL parameters) that allow you to extract values from the URL.

Nested Routing:

React Router supports nested routing, enabling you to create complex layouts with multiple levels of routing. Child routes can be nested within parent routes, and components can be structured accordingly.

Programmatic Navigation:

You can programmatically navigate between routes in your application using the history object provided by React Router or by using the `useHistory` hook.

Route Rendering:

React Router allows you to render different components based on the route match. This feature is useful for creating layouts that include headers,

footers, or sidebars that remain consistent across multiple pages.

Route Guards:

You can implement route guards or authentication checks by wrapping your route components with higher-order components (HOCs) or hooks to control access to certain routes.

Browser History Integration:

React Router uses the HTML5 History API to handle routing. This allows you to create clean, client-side URLs and handle navigation through the browser's back and forward buttons.

Code Splitting:

React Router integrates well with code-splitting techniques, allowing you to load route-specific components only when they are needed. This can improve application performance.

Server-Side Rendering (SSR):

React Router can be used in server-side rendering environments, ensuring that routes work both on the client and server sides.

Navigation Features:

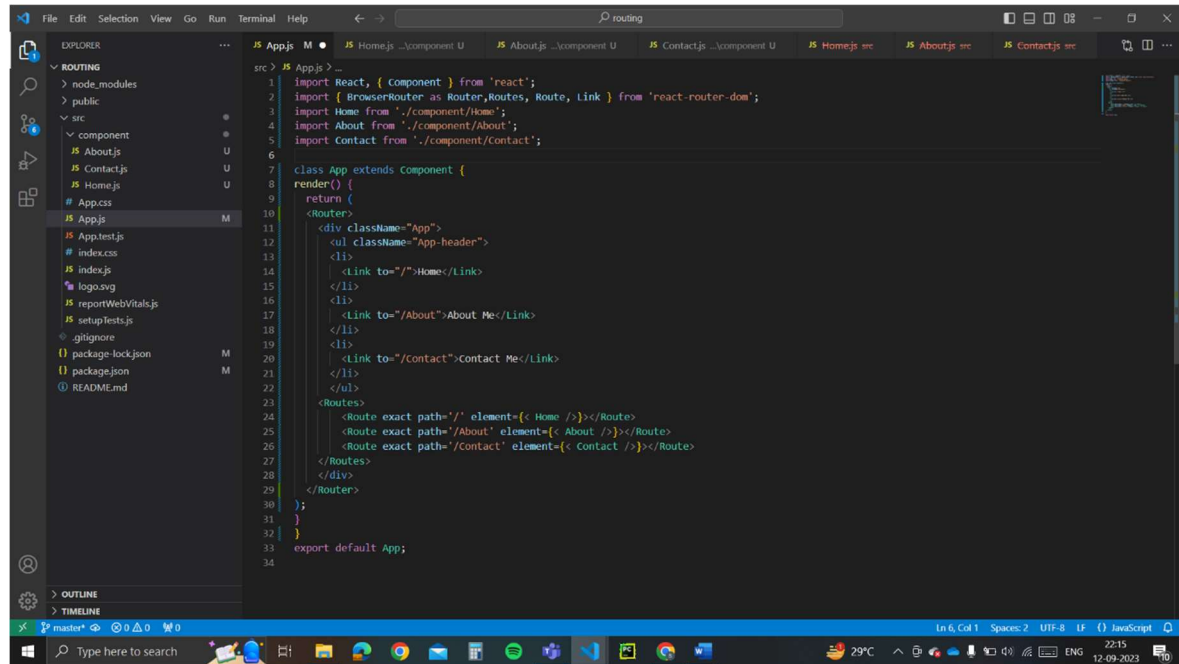
React Router includes navigation features such as route redirects, history manipulation, and scroll restoration to enhance the user experience.

Community and Ecosystem:

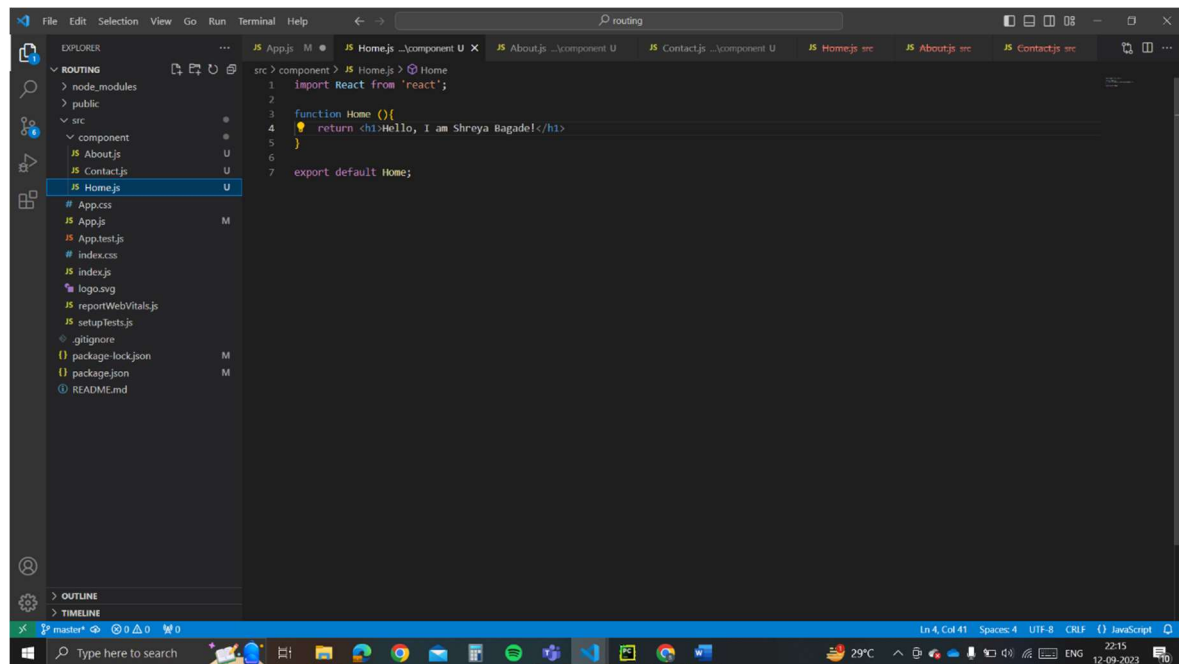
React Router is widely used and has an active community. It is well-documented and supported by various plugins and extensions for advanced routing needs.

Conclusion: This experiment with React Routing using BrowserRouter has demonstrated the power and flexibility of this library for creating dynamic single-page applications. We successfully defined routes, navigated between different views, and leveraged dynamic parameters to create a responsive user experience. React Router's ability to handle client-side routing efficiently makes it an essential tool for modern web application development.

Code:



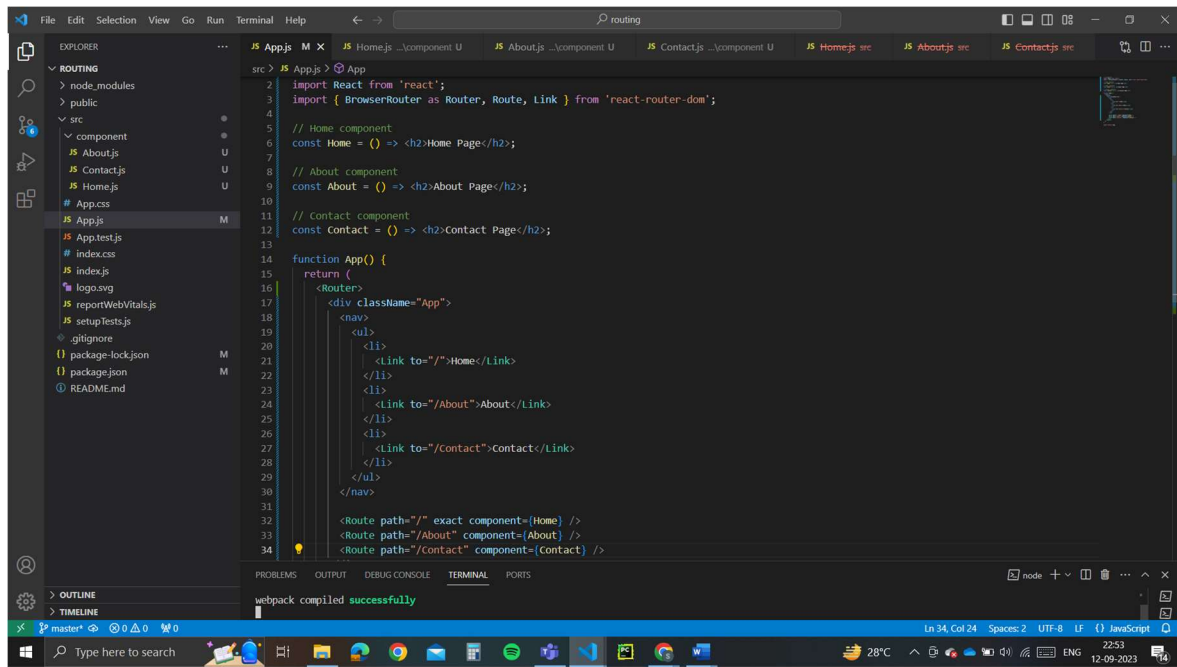
```
1 import React, { Component } from 'react';
2 import { BrowserRouter as Router, Routes, Route, Link } from 'react-router-dom';
3 import Home from './component/Home';
4 import About from './component/About';
5 import Contact from './component/Contact';
6
7 class App extends Component {
8   render() {
9     return (
10      <Router>
11        <div className="App">
12          <ul className="App-header">
13            <li>
14              <Link to="/">Home</Link>
15            </li>
16            <li>
17              <Link to="/About">About Me</Link>
18            </li>
19            <li>
20              <Link to="/Contact">Contact Me</Link>
21            </li>
22          </ul>
23          <Routes>
24            <Route exact path="/" element={< Home />}</Route>
25            <Route exact path="/About" element={< About />}</Route>
26            <Route exact path="/Contact" element={< Contact />}</Route>
27          </Routes>
28        </div>
29      </Router>
30    );
31  }
32 }
33 export default App;
```



```
1 import React from 'react';
2
3 function Home () {
4   return <h1>Hello, I am Shreya Bagade</h1>
5 }
6
7 export default Home;
```

```
1 import React from 'react';
2
3 function About () {
4   return <div>
5     <h2>I am a Third Year Engineering Student at TSEC!</h2>
6     <h2>I am proficient in HTML, CSS and JavaScript.</h2>
7   </div>
8 }
9
10 export default About;
```

```
1 import React from 'react';
2
3 function Contact (){
4   return <address>
5     You can contact me at: shreyabagade@abc.com
6   </address>
7 }
8
9
10
11 export default Contact;
```



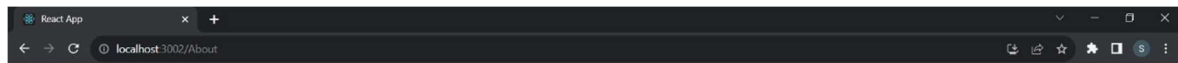
```
src > JS App.js > App
2 import React from 'react';
3 import { BrowserRouter as Router, Route, Link } from 'react-router-dom';
4
5 // Home component
6 const Home = () => <h2>Home Page</h2>;
7
8 // About component
9 const About = () => <h2>About Page</h2>;
10
11 // Contact component
12 const Contact = () => <h2>Contact Page</h2>;
13
14 function App() {
15   return (
16     <Router>
17       <div className="App">
18         <nav>
19           <ul>
20             <li>
21               <Link to="/">Home</Link>
22             </li>
23             <li>
24               <Link to="/About">About</Link>
25             </li>
26             <li>
27               <Link to="/Contact">Contact</Link>
28             </li>
29           </ul>
30         </nav>
31
32         <Route path="/" exact component={Home} />
33         <Route path="/About" component={About} />
34         <Route path="/Contact" component={Contact} />
35       </div>
36     </Router>
37   );
38 }
39
40 export default App;
```

webpack compiled successfully



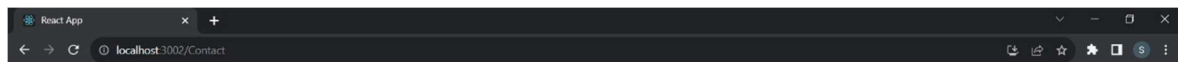
Hello, I am Shreya Bagade!





I am a Third Year Engineering Student at TSEC!

I am proficient in HTML,CSS and JavaScript.



You can contact me at:
shreyabagade@abc.com



Lab Outcome : LO5- Construct Front-end applications using React.