

Experiment -6: Design of React Single Page Application

- 1. Aim:** To design a single page application in react using react router.
- 2. Objective:** To understand and design front end applications using react
- 3. Lab Outcome:** Students will be able to construct front end applications using React (PO3, PO5, PSO3, PSO4)
- 4. Prerequisite:** HTML, CSS, node.js, react
- 5. Requirements:** The following are the requirements –
 - PC/Laptop
 - Visual Studio Code
 - Browser

6. Pre-Experiment Theory:

React Components are independent and reusable bits of code. They serve the same purpose as JavaScript functions, but work in isolation and return HTML.

Components come in two types –

- Class components - A class component must include the extends React.Component statement. This statement creates an inheritance to React.Component, and gives your component access to React.Component's functions. The component also requires a render() method, this method returns HTML.
- Function components - A Function component also returns HTML, and behaves much the same way as a Class component, but Function components can be written using much less code, are easier to understand.

Components can be passed as props, which stands for properties. Props are like function arguments, and you send them into the component as attributes.

We can refer to components inside other components.

React is all about re-using code, and it is recommended to split your components into separate files. To do that, create a new file with a .js file extension and put the code inside it. Create React App doesn't include page routing. React Router is the most popular solution. React Router is a standard library for routing in React. It enables the navigation among views of various components in a React Application, allows changing the browser URL, and keeps the UI in sync with the URL.

React Router can be installed via npm in your React application. After installing react-router-dom, add its components to your React application.

The main Components of React Router are:

- BrowserRouter - BrowserRouter is a router implementation that uses the HTML5

1 |Internet Programming Lab

history API (pushState, replaceState and the popstate event) to keep your UI in sync with the URL. It is the parent component that is used to store all of the other components.

- Route - Route is the conditionally shown component that renders some UI when its path matches the current URL.
- Link - Link component is used to create links to different routes and implement navigation around the application. It works like HTML anchor tag.
- Switch - Switch component is used to render only the first route that matches the location rather than rendering all matching routes. Although there is no defying functionality of SWITCH tag in our application because none of the LINK paths are ever going to coincide. But let's say we have a route (Note that there is no EXACT in here), then all the Route tags are going to be processed which start with '/' (all Routes start with /). This is where we need SWITCH statement to process only one of the statements.

7. Laboratory Exercise:

A. Procedure

- Add following extensions in vs code (Recommended) – Babel JavaScript, JavaScript ES6 Code Snippets, vs code icons
- Install node.js (npm installs automatically)
- Open terminal window (use cmd command)
- Check version of node.js (node -v)
- Check version of npm (npm -v)
- Install react from terminal using following commands -
 - npm install -g create-react-app
 - create-react-app --version
 - d:
 - mkdir <foldername>
 - cd <foldername>
 - create-react-app <projname> (may take a few minutes)
 - cd <projname>
 - npm start (localhost:3000 – react app installed in local system)
- Open vs-code and open the folder.
- Write your code

To view the output, reload the react app

Check output

B. Program Code

Design a single page application in react using react router using the suggestive guidelines as mentioned below. However, you are free to include other features covered earlier to make the design more feature rich.

2 |Internet Programming Lab

A Single Page Application

Internet ProgrammingHomeSessionsContact

Topics to be covered

Sr. No.	Topics
1	
2	
3	
4	
5	

Sessions=>

3-4 lines about every topic

Contacts

Contact Details like Phone No, Email-ID, Location

8. Post Experimental Exercise

- 1) What is HashRouter in react?
- 2) Differentiate between HashRouter and BrowserRouter.
- 3) Explain the use of basename, window and future properties in HashRouter with the help of an example.

9. Results/Observations/Program output:

Present the program code and output

10. Conclusion:

Write what was performed in the experiment

Write which all features of HTML, CSS and React you used to perform the experiment

11. References:

- <https://nodejs.org/en/>
- <https://www.thapatechnical.com/2020/05/install-reactjs-windows-install-nodejs.html> •
- https://www.kirupa.com/react/setting_up_react_environment.htm
- https://www.kirupa.com/react/creating_single_page_app_react_using_react_router.htm •
- <https://youtu.be/tg73NsiQOUE>
- <https://www.youtube.com/watch?v=8AJ3Kcz5FsM>

1. After creating a React app using Create React App (CRA), the default structure includes a file called **App.js**. To create a single-page application (SPA), we edited that file to incorporate **React Router**, allowing navigation between different components (pages) without reloading the entire application. Here's the code:

App.js

```
import { BrowserRouter as Router, Routes, Route, Link } from
'react-router-dom';
import { FontAwesomeIcon } from '@fortawesome/react-fontawesome';
import { faInstagram, faFacebook, faYoutube } from
'@fortawesome/free-brands-svg-icons';
import Home from './Home';
import Sessions from './Sessions';
import Contacts from './Contacts';
import './App.css';

function App() {
  return (
    <Router>
      <div className='App'>
        <header>
          <h1 className='header'>Single Page Application</h1>
          <nav className='navbar'>
            <span>Internet Programming</span>
            <div>
              <Link to="/">Home</Link>
              <Link to="/Sessions">Sessions</Link>
              <Link to="/Contacts">Contacts</Link>
            </div>
            <div className='Social-icons'>
              <FontAwesomeIcon icon={faFacebook} />
              <FontAwesomeIcon icon={faInstagram} />
              <FontAwesomeIcon icon={faYoutube} />
            </div>
          </nav>
        </header>
        <main>
          <Routes>
            <Route path="/" element={<Home />} />
            <Route path="/Sessions" element={<Sessions />} />
            <Route path="/Contacts" element={<Contacts />} />
          </Routes>
        </main>
      </div>
    </Router>
  );
}

export default App;
```

```
    </Routes>
  </main>
</div>
</Router>
);
}

export default App;
```

App.css

```
html, body {
  margin: 0;
  padding: 0;
  height: 100%;
}

body {
  text-align: center;
  background-image: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
  min-height: 100vh;
  color: #333;
}

.header{
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  color: white;
  margin-top: 2;
  padding: 0;
}

.navbar {
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 1rem 2rem;
  background-color: rgba(255, 255, 255, 0.2);
  backdrop-filter: blur(10px);
}
```

```

-webkit-backdrop-filter: blur(10px);
border-radius: 50px;
box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
border: 1px solid rgba(255, 255, 255, 0.3);
margin: 0 2rem;
}

.navbar div {
  display: flex;
  gap: 1.5rem;
}

.navbar a {
  color: white;
  text-decoration: none;
  font-weight: 600;
  font-size: 1.1rem;
  padding: 0.5rem;
  transition: all 0.3s ease;
  background-color: rgba(0, 0, 0, 0.2);
  border-radius: 8px;
}

.navbar a:hover {
  color: #764ba2;
  background-color: rgba(255, 255, 255, 0.2);
}

.navbar span {
  font-size: 1.8rem;
  font-weight: bold;
  color: white;
}

.Social-icons {
  display: flex;
  gap: 1rem;
}

.Social-icons .fa-facebook,
.Social-icons .fa-instagram,
.Social-icons .fa-youtube {

```

```

color: white;
font-size: 1.5rem;
transition: color 0.3s ease, transform 0.3s ease;
}

.Social-icons .fa-facebook:hover,
.Social-icons .fa-instagram:hover,
.Social-icons .fa-youtube:hover {
  color: #764ba2;
  transform: scale(1.2);
}

```

2. Here is the **Home Component**, which serves as the landing page of our single-page application. This component will be displayed when users navigate to the root URL ("/"). The component is imported into the App.js file and rendered as part of the routing system using Route.

Home.js

```

import React from 'react'
import './Home.css'
const Home = () => {
  return (
    <div className='Home'>
      <h3>Topics To Be Covered</h3>
      <table>
        <tr>
          <th>Sr No.</th>
          <th>Topics</th>
        </tr>
        <tr>
          <td>1</td>
          <td>HTML</td>
        </tr>
        <tr>
          <td>2</td>
          <td>Css</td>
        </tr>
        <tr>
          <td>3</td>
          <td>JavaScript</td>
        </tr>
      </table>
    </div>
  )
}

```

```

    </tr>
    <tr>
      <td>4</td>
      <td>React</td>
    </tr>
    <tr>
      <td>5</td>
      <td>Node.js</td>
    </tr>
  </table>
</div>
)
}

export default Home

```

Home.css

```

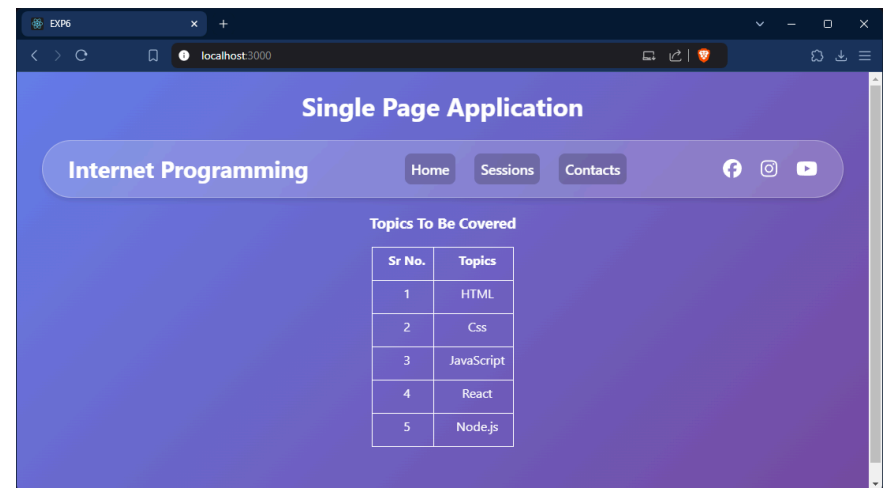
.Home{
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  color: white;
}

table, th, td {
  border: 1px solid white;
  border-collapse: collapse;
  color: white;
}

th, td {
  padding: 5px 10px 15px 20px;
}

```

Home Output



3. Here is the Sessions Component. This component will be displayed when users navigate to the /Sessions URL. Similar to the Home Component, it is imported into App.js and rendered as part of the routing system using the Route component.

Sessions.js

```

import React from 'react';
import './Sessions.css';

function Sessions() {
  return (
    <div className="sessions-container">
      <h2>Web Development Topics</h2>

      <section className="section-box">
        <h3>1. HTML</h3>
        <p>
          HTML (HyperText Markup Language) is the backbone of any web page.
          It provides the structure of a webpage by using elements such as headings,
          paragraphs, links, and images. It forms the foundation of all web
          development.
        </p>
      </section>
    </div>
  );
}

```

```

<section className="section-box">
  <h3>2. CSS</h3>
  <p>
    CSS (Cascading Style Sheets) is used to style and layout web
    pages. It allows developers to add colors, fonts, and spacing, and control
    how elements are displayed on different devices. CSS enables responsive
    design and improves user experience.
  </p>
</section>

<section className="section-box">
  <h3>3. JavaScript</h3>
  <p>
    JavaScript is a powerful programming language that enables
    interactivity on websites. From form validation to complex animations and
    asynchronous data fetching, JavaScript makes websites dynamic and
    interactive.
  </p>
</section>

<section className="section-box">
  <h3>4. React</h3>
  <p>
    React is a popular JavaScript library for building user
    interfaces, especially for single-page applications. It helps developers
    create reusable components, making the code more manageable and efficient.
  </p>
</section>

<section className="section-box">
  <h3>5. Node.js</h3>
  <p>
    Node.js is a server-side runtime environment that allows
    JavaScript to be used on the server. It is known for its efficiency and
    scalability, making it ideal for building fast, data-intensive applications.
  </p>
</section>
</div>
);
}

```

```
export default Sessions;
```

Sessions.css

```

.sessions-container {
  padding: 2rem;
  display: flex;
  flex-wrap: wrap;
  gap: 2rem;
  justify-content: center;
  align-items: flex-start;
  min-height: 100vh;
  background-image: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
}

.sessions-container h2 {
  color: white;
  font-size: 2.5rem;
  text-align: center;
  margin-bottom: 2rem;
  width: 100%;
}

.section-box {
  background: rgba(255, 255, 255, 0.1);
  border: 1px solid rgba(255, 255, 255, 0.3);
  backdrop-filter: blur(15px);
  -webkit-backdrop-filter: blur(15px);
  border-radius: 15px;
  padding: 2rem;
  max-width: calc(33.33% - 2rem);
  flex-basis: calc(33.33% - 2rem);
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.2);
  color: white;
  transition: transform 0.3s ease, box-shadow 0.3s ease;
}

.section-box h3 {
  font-size: 1.8rem;
  margin-bottom: 1rem;
}

.section-box p {
  font-size: 1.1rem;
}

```

```

    line-height: 1.6;
}

.section-box:hover {
    transform: translateY(-10px);
    box-shadow: 0 8px 12px rgba(0, 0, 0, 0.3);
}

```

Sessions Output



4. Here is the Contact Component. This component will be displayed when users navigate to the /Contacts URL. Similar to the Home Component, it is imported into App.js and rendered as part of the routing system using the Route component.

Contacts.js

```

import React from 'react';
import './Contacts.css';

function Contacts() {
    return (
        <div className="contact-container">
            <div className="contact-box">
                <h2>Contact Us</h2>
                <p><strong>Phone:</strong> +91-12345-67890</p>
                <p><strong>Email:</strong> <a
href="mailto:example@example.com">vism06@student.sfit.ac.in</a></p>
                <p><strong>Address:</strong> Gate no 5, SFIMAR, Mount Poincur, S.V.P
Road, Borivali(W)-400103</p>
            </div>
        </div>
    );
}

export default Contacts;

```

Contacts.css

```

.contact-container {
    display: flex;
    justify-content: center;
    align-items: center;
    margin: 0;
    padding: 0;
    background-image: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
    margin-top: 100px;
}

.contact-box {
    background: rgba(255, 255, 255, 0.1);
    border: 1px solid rgba(255, 255, 255, 0.3);
    backdrop-filter: blur(15px);
}

```



```

-webkit-backdrop-filter: blur(15px);
border-radius: 15px;
padding: 3rem;
max-width: 600px;
width: 100%;
box-shadow: 0 4px 6px rgba(0, 0, 0, 0.2);
color: white;
text-align: center;
}

.contact-box h2 {
  font-size: 2.5rem;
  margin-bottom: 1.5rem;
}

.contact-box p {
  font-size: 1.2rem;
  margin: 0.5rem 0;
}

.contact-box a {
  color: #61dafb;
  text-decoration: none;
  transition: color 0.3s ease;
}

```

Contacts Output

