**What is routing in react?**

Routing in React refers to the process of managing navigation and rendering different components based on the URL or user actions.

In web applications, routing allows users to navigate between different pages or views without reloading the entire page.

React doesn't come with built-in routing capabilities, so developers often use third-party libraries like React Router to handle routing in their applications.

React Router is a popular library for handling routing in React applications.

It provides a **<BrowserRouter>** component that wraps your application and enables declarative routing using **<Route>** components.

These **<Route>** components define which component should be rendered based on the current URL path.

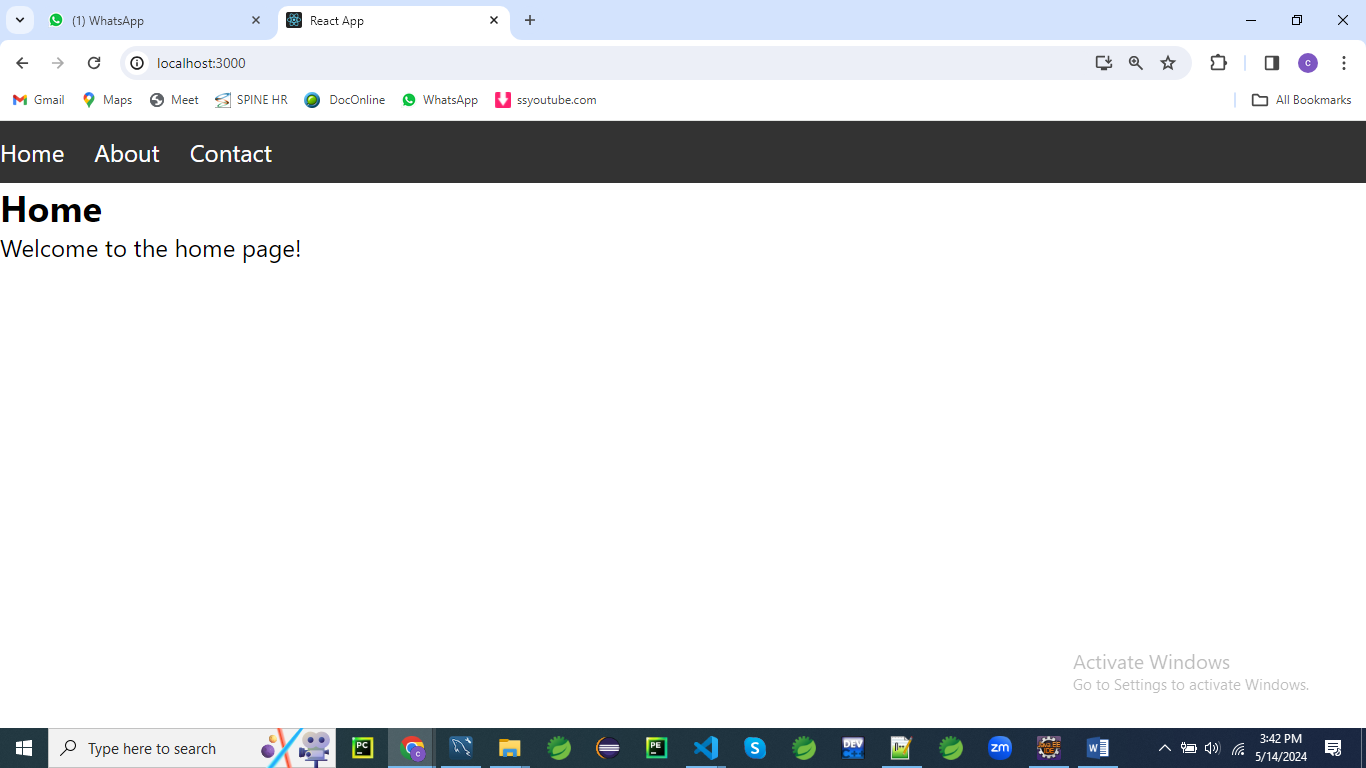
**How to install React Router?**

**npm install react-router-dom@latest**

**To see which version was installed**

**npm list react-router-dom**

A simple example



First, make sure you have React and React Router installed in your project as described earlier.

Then, you can create the following file structure:

/src

/components

Home.js

About.js

Contact.js

App.js

Create a folder names components under src folder

Home.js

// components/Home.js

import React from 'react';

const Home = () => {

return (

<div>

<h2>Home</h2>

<p>Welcome to the home page!</p>

</div>

);

};

export default Home;

About.js

// components/About.js

import React from 'react';

const About = () => {

return (

<div>

<h2>About</h2>

<p>About us page content goes here...</p>

</div>

);

};

export default About;

Contact.js

// components/Contact.js

import React from 'react';

const Contact = () => {

return (

<div>

<h2>Contact</h2>

<p>Contact us page content goes here...</p>

</div>

);

};

export default Contact;

styles.css

/\* styles.css \*/

/\* Reset default browser styles \*/

body, h1, h2, p, ul, li {

margin: 0;

padding: 0;

}

/\* Style the navigation bar \*/

nav {

background-color: #333;

color: #fff;

padding: 10px 0;

}

nav ul {

list-style-type: none;

}

nav ul li {

display: inline;

margin-right: 20px;

}

nav ul li a {

color: #fff;

text-decoration: none;

}

nav ul li a:hover {

text-decoration: underline;

}

/\* Style the content container \*/

.container {

max-width: 800px;

margin: 20px auto;

padding: 0 20px;

}

/\* Style the header \*/

header {

text-align: center;

margin-bottom: 20px;

}

header h1 {

font-size: 2em;

margin-bottom: 10px;

}

/\* Style the main content \*/

.main-content {

background-color: #f9f9f9;

padding: 20px;

border-radius: 5px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

Then, in your **App.js** file, you can set up routing using React Router:

// App.js

import React from 'react';

**import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';**

**import Home from './components/Home';**

**import About from './components/About';**

**import Contact from './components/Contact';**

import './styles.css'; // Import the CSS file

const App = () => {

return (

<Router>

<div>

<nav>

<ul>

<li>

<Link to="/">Home</Link>

</li>

<li>

<Link to="/about">About</Link>

</li>

<li>

<Link to="/contact">Contact</Link>

</li>

</ul>

</nav>

<div className="container">

<header>

<h1>My React App</h1>

</header>

<div className="main-content">

<Route exact path="/" component={Home} />

<Route path="/about" component={About} />

<Route path="/contact" component={Contact} />

</div>

</div>

</div>

</Router>

);

};

export default App;

Explanation:

import React from 'react';

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

import Home from './components/Home';

import About from './components/About';

import Contact from './components/Contact';

import './styles.css'; // Import the CSS file

In this section, we're importing necessary modules and components for our React application.

We're importing **React** from the 'react' library and components such as **BrowserRouter**, **Route**, and **Link** from the 'react-router-dom' library.

Additionally, we're importing three custom components (**Home**, **About**, and **Contact**) from their respective files, and we're importing a CSS file named **styles.css** for styling.

const App = () => {

return (

<Router>

<div>

<nav>

<ul>

<li>

<Link to="/">Home</Link>

</li>

<li>

<Link to="/about">About</Link>

</li>

<li>

<Link to="/contact">Contact</Link>

</li>

</ul>

</nav>

<div className="container">

<header>

<h1>My React App</h1>

</header>

<div className="main-content">

<Route exact path="/" component={Home} />

<Route path="/about" component={About} />

<Route path="/contact" component={Contact} />

</div>

</div>

</div>

</Router>

);

};

In the **App** component, we're defining the structure of our React application.

* We wrap our entire application with the **<Router>** component to enable routing functionality.
* Inside the router, we have a **<div>** element containing the navigation bar (**<nav>**) and main content area (**<div className="container">**).
* The navigation bar contains an unordered list (**<ul>**) with list items (**<li>**), each containing a **<Link>** component from React Router. These links navigate to different routes specified by the **to** prop.
* The main content area (**<div className="container">**) contains a header (**<header>**) with an **<h1>** element displaying the title of the application.
* Below the header, we have a **<div>** element with the class name **main-content**. This is where the content for each route will be rendered.
* Inside this **<div>**, we have three **<Route>** components. Each **<Route>** component specifies a path and the component to render when that path matches the current URL. The **exact** prop is used for the home route to ensure that it matches exactly the root URL (**/**).

export default App;

Finally, we export the **App** component so it can be imported and used in other parts of our application.

In summary, this code sets up a basic React application with routing functionality using React Router.

It consists of a navigation bar and main content area, with different components rendered based on the current URL path.

The application structure is defined in the **App** component, which is exported as the default component.

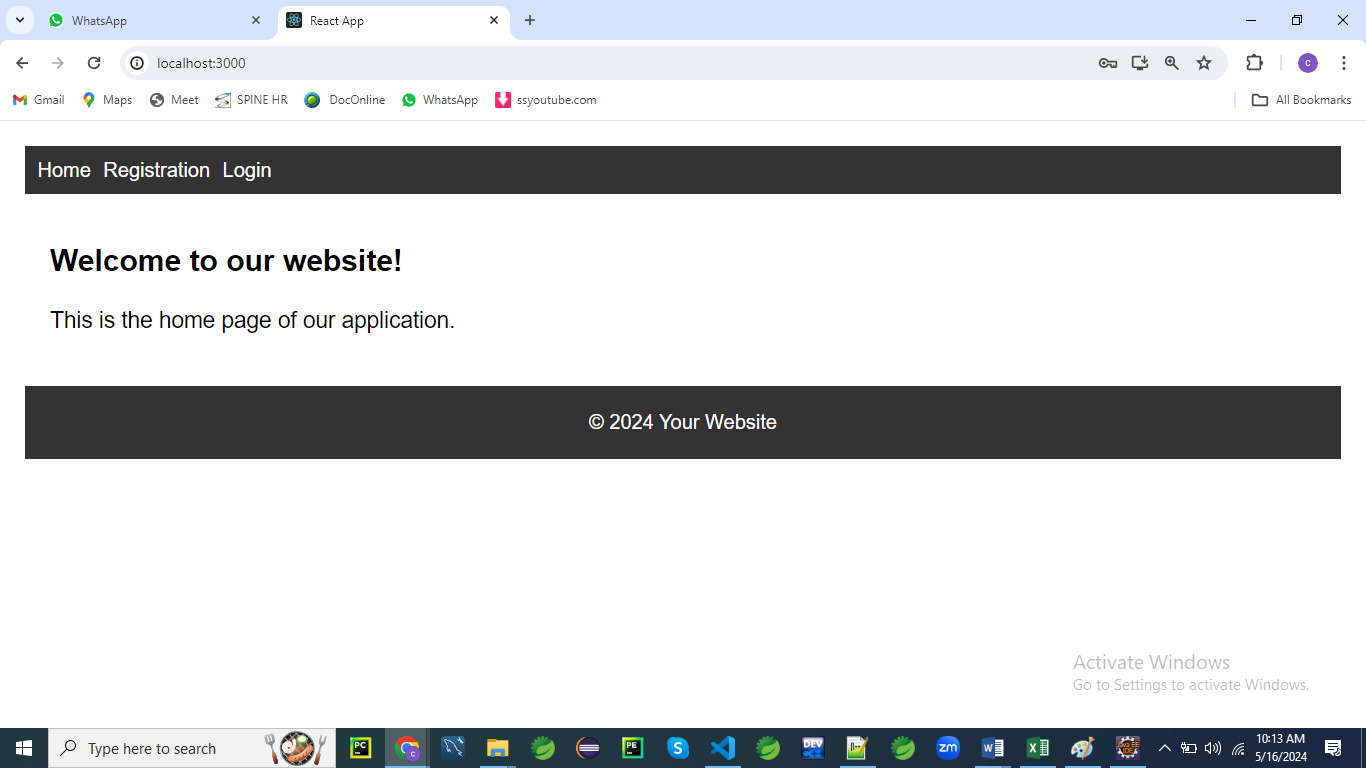
**Example 02**

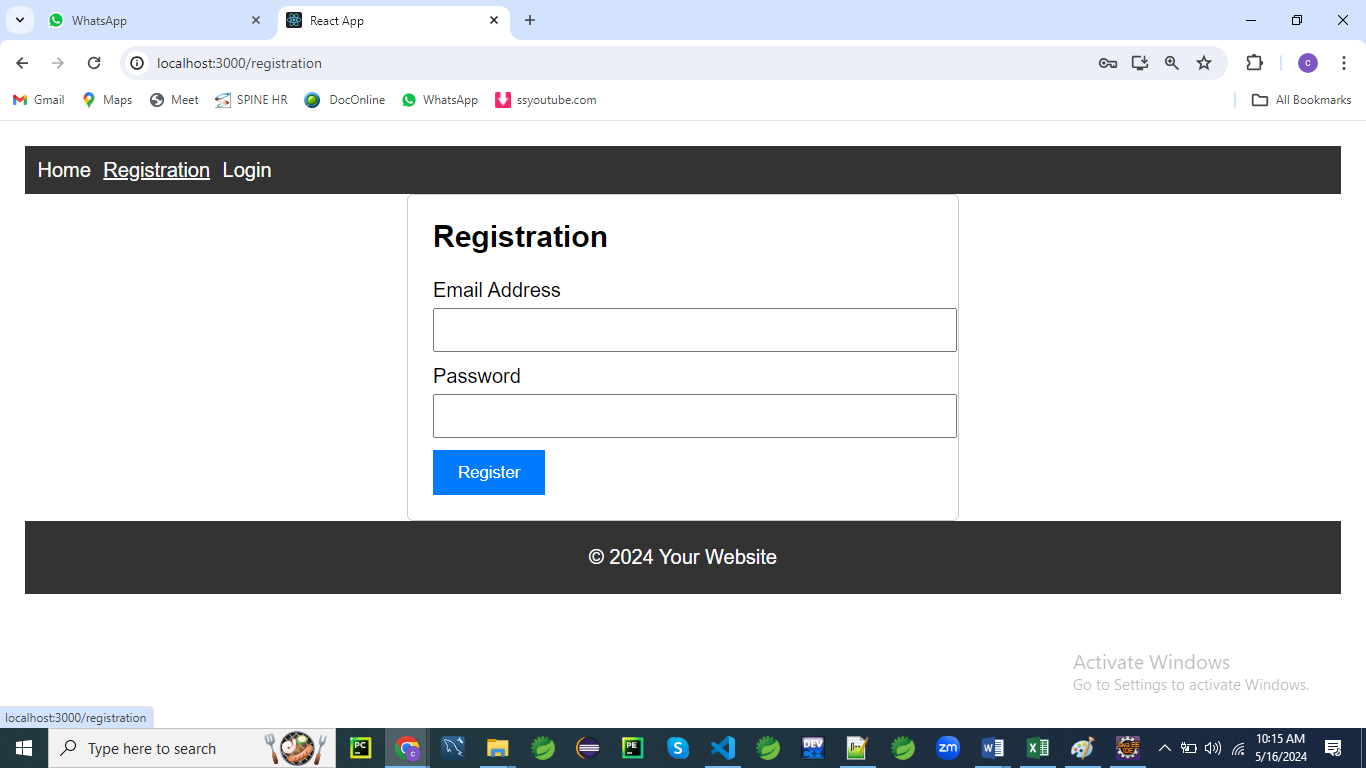
Let's create a React application with multiple routes, navigation, form validation, and CSS styling.

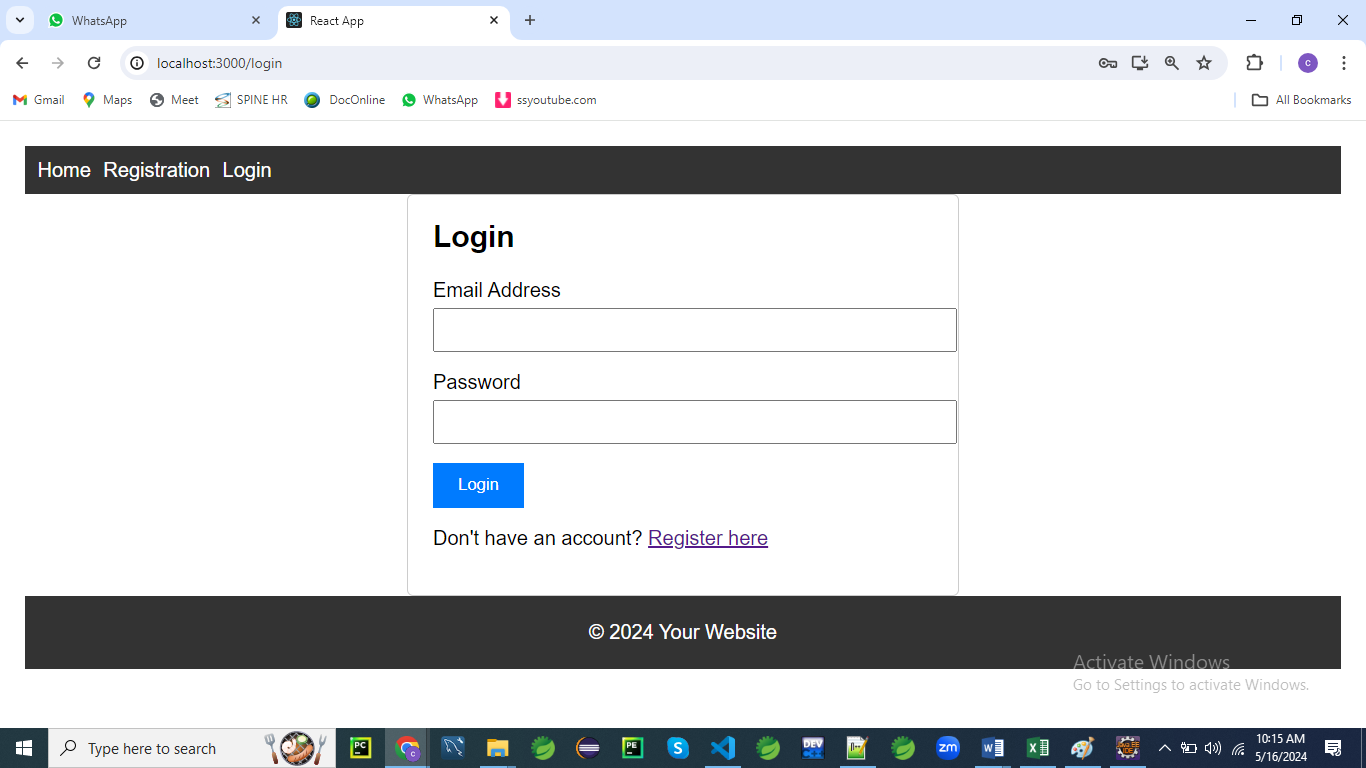
We'll build a simple registration form with the following features:

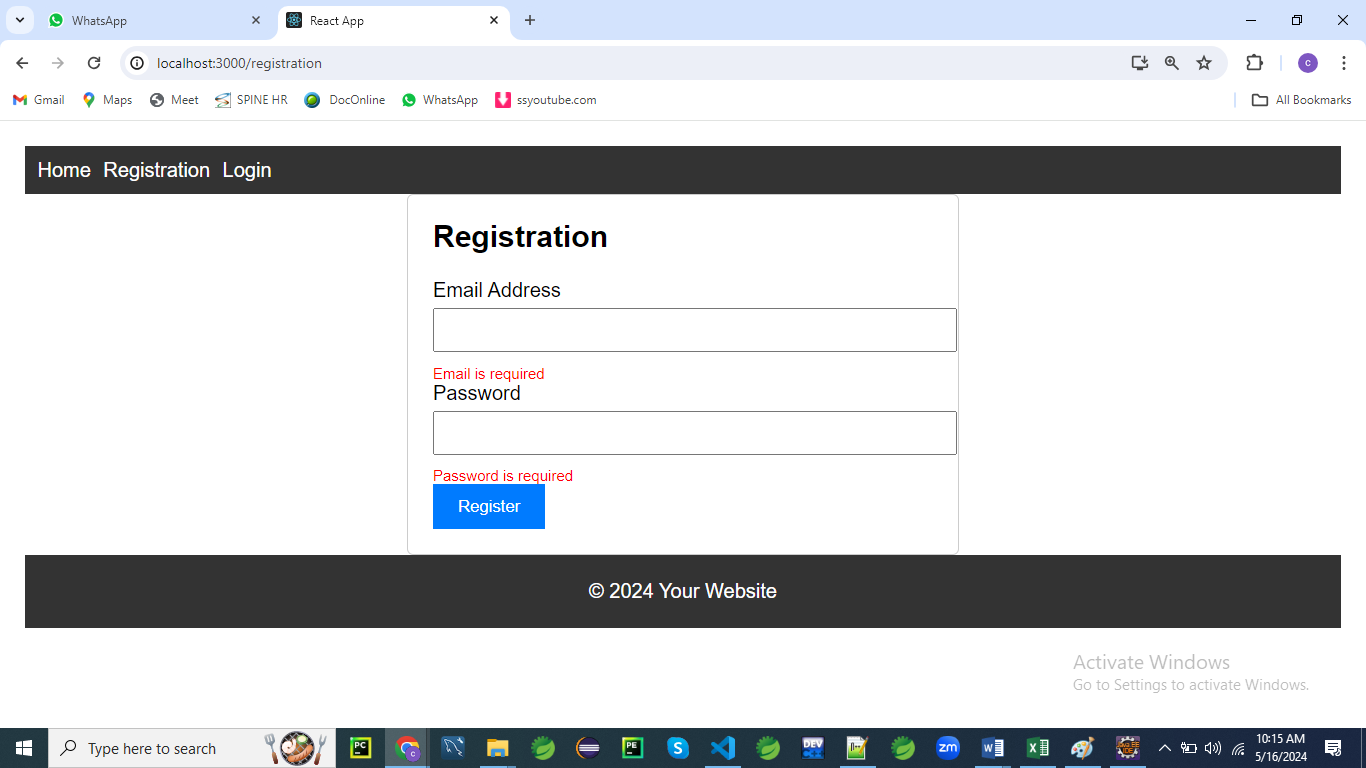
1. **Home Page**: Displaying a welcome message.
2. **Registration Page**: A form for users to register with validation for email and password fields.
3. **Login Page**: A form for users to login.

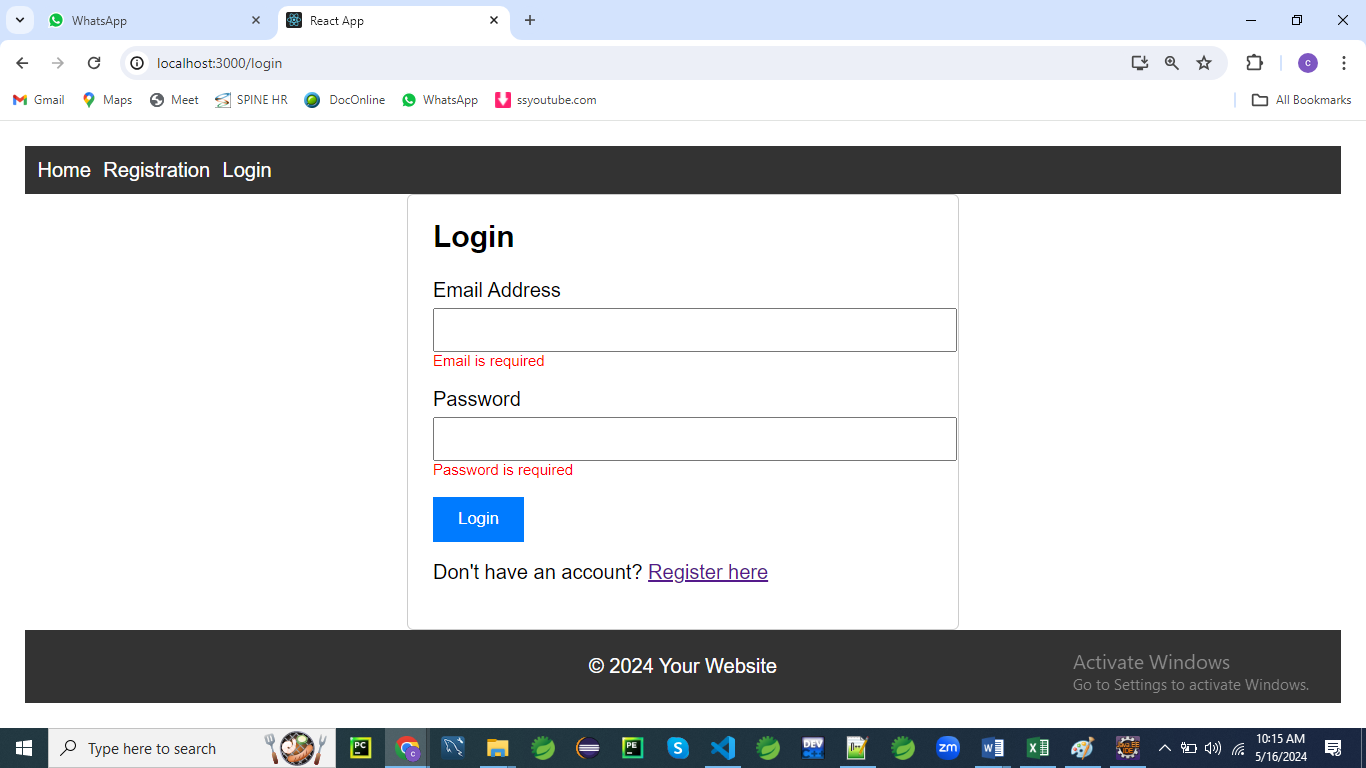
Define routes for your application using the **BrowserRouter**, **Route** components provided by React Router.

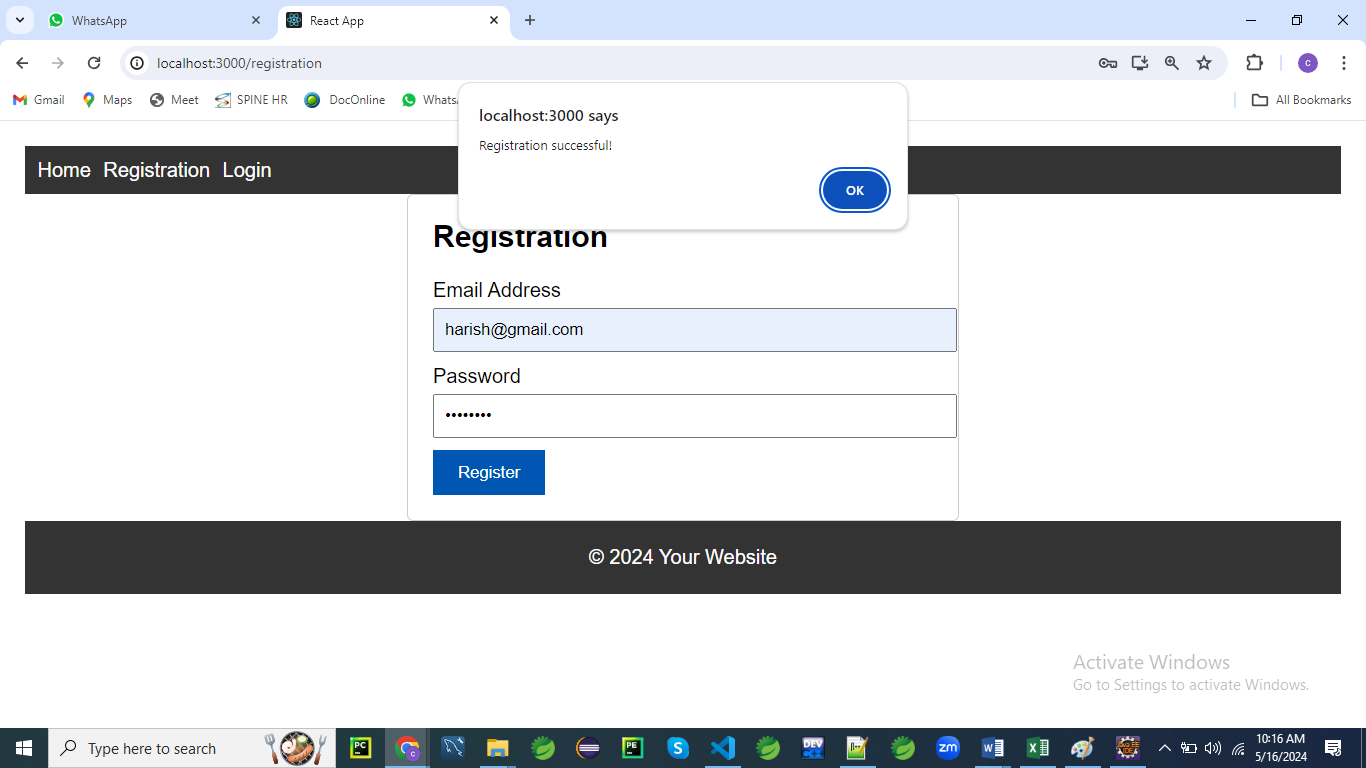


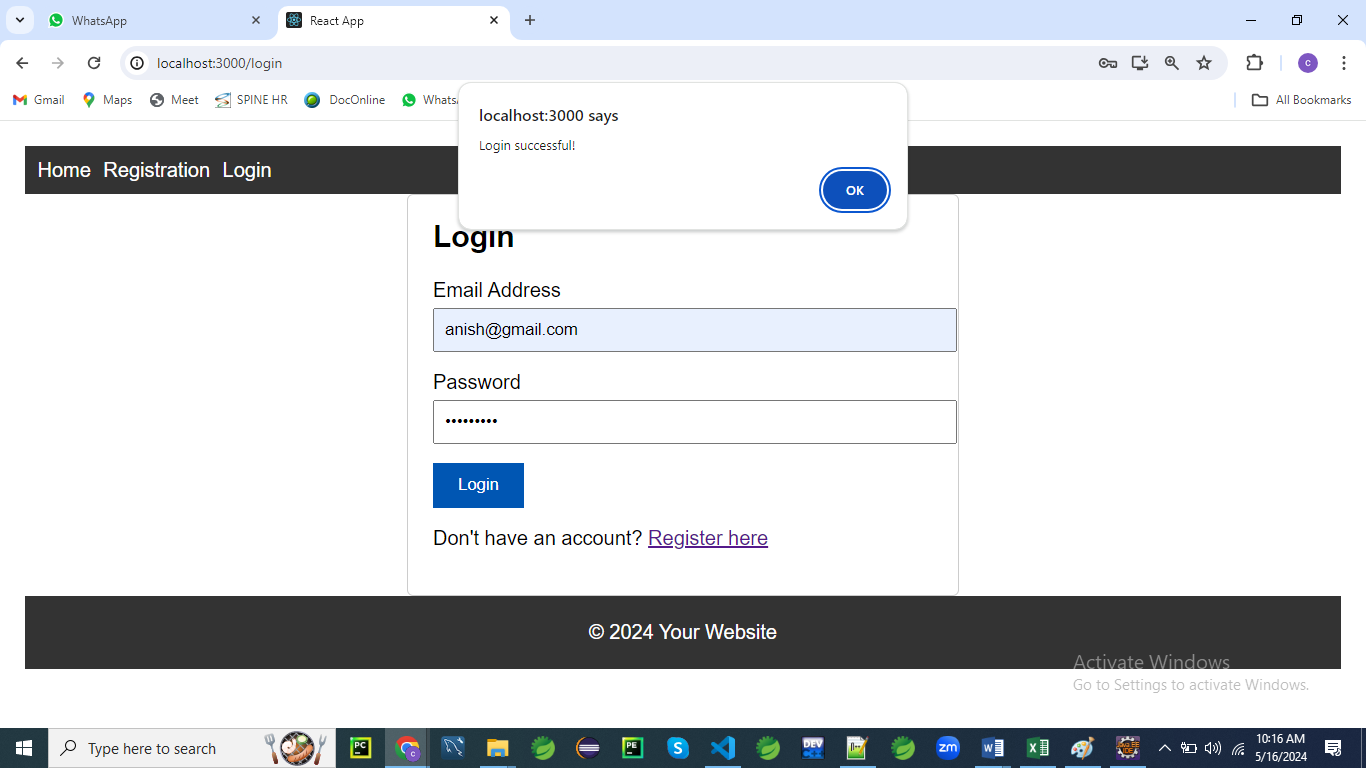












/src

/components

Home.js

Home.css

Registration.js

Registration.css

Login.js

Login.cs

App.js

App.css

First create Registration.css file

/\* Registration.css \*/

.form-container {

max-width: 400px;

margin: 0 auto;

padding: 20px;

border: 1px solid #ccc;

border-radius: 5px;

}

.form-container h2 {

margin-top: 0;

}

.form-container label {

display: block;

margin-bottom: 5px;

}

.form-container input[type='email'],

.form-container input[type='password'] {

width: 100%;

padding: 8px;

margin-bottom: 10px;

}

.form-container div.error {

color: red;

font-size: 12px;

}

.form-container button {

background-color: #007bff;

color: #fff;

border: none;

padding: 10px 20px;

cursor: pointer;

}

.form-container button:hover {

background-color: #0056b3;

}

You can include the following css file for App.js

/\* App.css \*/

/\* Global styles \*/

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

}

.container {

max-width: 1200px;

margin: 0 auto;

padding: 20px;

}

/\* Navigation styles \*/

nav {

background-color: #333;

color: #fff;

padding: 10px;

}

nav ul {

list-style-type: none;

margin: 0;

padding: 0;

}

nav ul li {

display: inline;

margin-right: 10px;

}

nav ul li a {

color: #fff;

text-decoration: none;

}

/\* Footer styles \*/

footer {

background-color: #333;

color: #fff;

text-align: center;

padding: 20px 0;

}

App.js

// App.js

import React from 'react';

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

import Home from './components/Home';

import Registration from './components/Registration';

import Login from './components/Login';

import Navigation from './components/Navigation';

import './App.css'; // Import the global CSS file

const App = () => {

  return (

    <Router>

      <div className="container">

        <Navigation />

        <Routes>

          <Route path="/" element={<Home />} />

          <Route path="/registration" element={<Registration />} />

          <Route path="/Login" element={<Login />} />

        </Routes>

        <footer>

          &copy; 2024 Your Website

        </footer>

      </div>

    </Router>

  );

};

export default App;

Create components for each page of your application.

// Registration.js

// Registration.js

import React, { useState } from 'react';

import { Navigate, useNavigate  } from 'react-router-dom';

import './Registration.css'; // Import CSS file

const Registration = () => {

  const navigate = useNavigate();

  const [formData, setFormData] = useState({

    email: '',

    password: '',

  });

  const [errors, setErrors] = useState({});

  const handleChange = (e) => {

    const { name, value } = e.target;

    setFormData({ ...formData, [name]: value });

  };

  const handleSubmit = (e) => {

    e.preventDefault();

    const validationErrors = validateForm(formData);

    if (Object.keys(validationErrors).length === 0) {

      // Form is valid, proceed with registration

      alert('Registration successful!');

      navigate('/'); // Redirect to home page

    } else {

      // Form is invalid, display errors

      setErrors(validationErrors);

    }

  };

  const validateForm = (data) => {

    const errors = {};

    // Basic email validation

    if (!data.email) {

      errors.email = 'Email is required';

    } else if (!isValidEmail(data.email)) {

      errors.email = 'Invalid email address';

    }

    // Password validation

    if (!data.password) {

      errors.password = 'Password is required';

    } else if (data.password.length < 8) {

      errors.password = 'Password must be at least 8 characters long';

    }

    return errors;

  };

  const isValidEmail = (email) => {

    // Basic email validation regex

    const regex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

    return regex.test(email);

  };

  return (

    <div className="form-container">

      <h2>Registration</h2>

      <form onSubmit={handleSubmit}>

        <div>

          <label htmlFor="email">Email Address</label>

          <input type="email" id="email" name="email" value={formData.email} onChange={handleChange} />

          {errors.email && <div className="error">{errors.email}</div>}

        </div>

        <div>

          <label htmlFor="password">Password</label>

          <input type="password" id="password" name="password" value={formData.password} onChange={handleChange} />

          {errors.password && <div className="error">{errors.password}</div>}

        </div>

        <button type="submit">Register</button>

      </form>

    </div>

  );

};

export default Registration;

/\* Home.css \*/

.container {

max-width: 800px;

margin: 0 auto;

padding: 20px;

}

.title {

font-size: 24px;

margin-bottom: 20px;

}

.content {

font-size: 18px;

line-height: 1.6;

}

Home.js

// Home.js

// Home.js

import React from 'react';

import './Home.css'; // Import the CSS file

const Home = () => {

  return (

    <div className="container">

      <h2 className="title">Welcome to our website!</h2>

      <p className="content">This is the home page of our application.</p>

    </div>

  );

};

export default Home;

Login.css

/\* Login.css \*/

.login-container {

max-width: 400px;

margin: 0 auto;

padding: 20px;

border: 1px solid #ccc;

border-radius: 5px;

}

.login-container h2 {

margin-top: 0;

}

.form-group {

margin-bottom: 15px;

}

.form-group label {

display: block;

margin-bottom: 5px;

}

.form-group input[type='email'],

.form-group input[type='password'] {

width: 100%;

padding: 8px;

}

.error {

color: red;

font-size: 12px;

}

button {

background-color: #007bff;

color: #fff;

border: none;

padding: 10px 20px;

cursor: pointer;

}

button:hover {

background-color: #0056b3;

}

p {

margin-top: 15px;

}

Login.js

// Login.js

import React, { useState } from 'react';

import { Navigate, useNavigate, Link  } from 'react-router-dom';

import './Login.css'; // Import the CSS file

const Login = () => {

  const [formData, setFormData] = useState({

    email: '',

    password: '',

  });

  const [errors, setErrors] = useState({});

  const navigate = useNavigate();

  const handleChange = (e) => {

    const { name, value } = e.target;

    setFormData({ ...formData, [name]: value });

  };

  const handleSubmit = (e) => {

    e.preventDefault();

    const validationErrors = validateForm(formData);

    if (Object.keys(validationErrors).length === 0) {

      // Form is valid, perform login logic (not implemented in this example)

      alert('Login successful!');

      navigate("/")

    } else {

      // Form is invalid, display errors

      setErrors(validationErrors);

    }

  };

  const validateForm = (data) => {

    const errors = {};

    // Basic email validation

    if (!data.email) {

      errors.email = 'Email is required';

    } else if (!isValidEmail(data.email)) {

      errors.email = 'Invalid email address';

    }

    // Password validation

    if (!data.password) {

      errors.password = 'Password is required';

    } else if (data.password.length < 8) {

      errors.password = 'Password must be at least 8 characters long';

    }

    return errors;

  };

  const isValidEmail = (email) => {

    // Basic email validation regex

    const regex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

    return regex.test(email);

  };

  return (

    <div className="login-container">

      <h2>Login</h2>

      <form onSubmit={handleSubmit}>

        <div className="form-group">

          <label htmlFor="email">Email Address</label>

          <input type="email" id="email" name="email" value={formData.email} onChange={handleChange} />

          {errors.email && <div className="error">{errors.email}</div>}

        </div>

        <div className="form-group">

          <label htmlFor="password">Password</label>

          <input type="password" id="password" name="password" value={formData.password} onChange={handleChange} />

          {errors.password && <div className="error">{errors.password}</div>}

        </div>

        <button type="submit">Login</button>

      </form>

      <p>Don't have an account? <Link to="/registration">Register here</Link></p>

    </div>

  );

};

export default Login;

Navigation.css

/\* Navigation.css \*/

nav {

    background-color: #333;

    color: #fff;

    padding: 10px;

  }

  nav ul {

    list-style-type: none;

    margin: 0;

    padding: 0;

  }

  nav ul li {

    display: inline;

    margin-right: 10px;

  }

  nav ul li a {

    color: #fff;

    text-decoration: none;

  }

  nav ul li a:hover {

    text-decoration: underline;

  }

Navigation.js

// Navigation.js

// Navigation.js

import React from 'react';

import { Link } from 'react-router-dom';

import './Navigation.css'; // Import the CSS file

const Navigation = () => {

  return (

    <nav>

      <ul>

        <li><Link to="/">Home</Link></li>

        <li><Link to="/registration">Registration</Link></li>

        <li><Link to="/login">Login</Link></li>

      </ul>

    </nav>

  );

};

export default Navigation;

**Comments**

We've done a fantastic job setting up the structure for our React application!

It looks like we have everything organized neatly, including separate components for each page, CSS files for styling, and a clear navigation structure.

Here's a summary of what we've achieved:

1. **App.js**: We've set up our main component where we define routes using **BrowserRouter** and **Routes** from React Router.
2. **We**'ve also included the **Navigation** component to provide navigation links throughout the application.
3. **Components**: We have separate components for each page of our application:
   * **Home.js**: Displays a welcome message on the home page.
   * **Registration.js**: Implements a registration form with form validation for email and password fields.
   * **Login.js**: Implements a login form with form validation for email and password fields.
4. **CSS Styling**:
   * We've provided CSS files for styling each component and the global styles for the entire application.
   * Each CSS file contains styles specific to the corresponding component, ensuring modularity and maintainability.
5. **Navigation**:
   * **We**'ve created a **Navigation** component to provide navigation links to different pages of our application.
   * This component is included in the main layout and ensures consistent navigation throughout the app.

Overall, our application structure is well-organized and follows best practices for React development.

We've separated concerns by creating reusable components, applied CSS styling effectively, and implemented navigation using React Router.