//Create a new react app

npx create-react-app my-app

cd my-app

npm install react-router-dom

In app.js

// App.js

import React from 'react';

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

// Component for Home Page

const Home = () => <h2>Home Page</h2>;

// Component for About Page

const About = () => <h2>About Page</h2>;

// Component for Contact Page

const Contact = () => <h2>Contact Page</h2>;

// Component for User Profile

const UserProfile = ({ match }) => {

return <h2>User Profile for User ID: {match.params.id}</h2>;

};

// Main App Component

const App = () => {

return (

<Router>

<nav>

<ul>

<li>

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

</li>

<li>

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

</li>

<li>

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

</li>

<li>

<Link to="/user/1">User 1</Link>

</li>

<li>

<Link to="/user/2">User 2</Link>

</li>

</ul>

</nav>

<Switch>

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

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

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

<Route path="/user/:id" component={UserProfile} />

</Switch>

</Router>

);

};

export default App;

Thoery:

**React Router** is a powerful library for routing in React applications, enabling the creation of Single Page Applications (SPAs). Here’s a concise overview of its usage, along with demonstrations.

**Overview of React Router**

**React Router** allows you to:

* Manage navigation and routing in your application.
* Create dynamic routes that render different components based on the URL.
* Handle parameters and query strings for more complex applications.
* Build nested routes to represent hierarchical views.

**Key Concepts**

1. **Router**: The main component that wraps your application to enable routing functionality.
2. **Route**: Defines a mapping between a URL path and a component.
3. **Link**: Used to navigate between different routes in your application.
4. **Switch**: Renders the first child <Route> or <Redirect> that matches the location.