

Day 7: React Router - Navigation & Dynamic Pages

Today, you'll learn how to **navigate between pages** in a React app using **React Router**. This will allow you to create **multi-page applications** with seamless routing.

1 What is React Router?

React Router is a **library** that helps manage navigation in a React app.

- ✓ Enables **multi-page navigation** without refreshing the page
- ✓ Supports **dynamic routes** and **URL parameters**
- ✓ Provides **history management**

Installing React Router

First, install React Router in your project:

```
npm install react-router-dom
```

2 Setting Up React Router

To enable routing, wrap your app with the **BrowserRouter** component.

Example: Basic Router Setup

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

function Home() {
  return <h2>Home Page</h2>;
}

function About() {
  return <h2>About Page</h2>;
}

function App() {
  return (
    <Router>
      <nav>
        <Link to="/">Home</Link> | <Link to="/about">About</Link>
      </nav>

      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/about" element={<About />} />
      </Routes>
    </Router>
  );
}
```

```
export default App;
```

- ✓ Navigation works without a full-page refresh
 - ✓ Clicking the links updates the URL & content dynamically
-

3 Navigating Between Pages

React Router provides two main ways to navigate:

1. **Link Component** (for static navigation)
2. **useNavigate Hook** (for programmatic navigation)

Example: Using Link for Navigation

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

Example: Using useNavigate for Navigation

```
import { useNavigate } from "react-router-dom";
```

```
function Home() {  
  const navigate = useNavigate();  
  
  return (  
    <div>  
      <h2>Home Page</h2>  
      <button onClick={() => navigate("/about")}>Go to About</button>  
    </div>  
  );  
}
```

- ✓ Use useNavigate for buttons or conditional navigation.
-

4 Dynamic Routes with URL Parameters

You can **pass dynamic values** through the URL using **route parameters (:id)**.

Example: Defining a Dynamic Route

```
import { useParams } from "react-router-dom";
```

```
function UserProfile() {  
  const { username } = useParams();  
  return <h2>Welcome, {username}!</h2>;  
}
```

```
function App() {  
  return (  
    <Router>  
      <Routes>  
        <Route path="/user/:username" element={<UserProfile />} />  
      </Routes>  
    </Router>  
  )  
}
```

```
  );  
}
```

✅ Visiting `/user/Danish` will display “Welcome, Danish!”

5 Redirects & 404 Pages

Redirecting with `useNavigate`

```
import { useNavigate } from "react-router-dom";  
  
function NotFound() {  
  const navigate = useNavigate();  
  
  return (  
    <div>  
      <h2>Page Not Found</h2>  
      <button onClick={() => navigate("/")}>Go Home</button>  
    </div>  
  );  
}
```

Handling 404 Pages

```
<Routes>  
  <Route path="*" element={<NotFound />} />  
</Routes>
```

✅ Handles unknown URLs with a custom 404 page.

Summary of Day 7

- ✓ Installed & set up **React Router**
 - ✓ Created **multiple pages** with `<Route>`
 - ✓ Used **Link** and `useNavigate` for navigation
 - ✓ Implemented **dynamic routes** (`useParams`)
 - ✓ Added **404 page & redirects**
-

Next Step: Day 8 - React Forms & Controlled Components