**ROUTER**

Nếu bạn có nhiều trang JSX trong ứng dụng React và muốn liên kết chúng với nhau thông qua điều hướng (navigation), bạn có thể sử dụng react-router-dom để quản lý các tuyến (routes). Đây là cách thiết lập hệ thống Router trong ứng dụng React để liên kết giữa các trang.

**Bước 1: Cài đặt react-router-dom**

bash

Copy code

npm install react-router-dom

**Bước 2: Tạo cấu trúc Router**

1. **Tạo các trang JSX (các component đại diện cho từng trang)**:
   * Giả sử bạn có các trang như Home.jsx, About.jsx, và Contact.jsx.

jsx

Copy code

// Home.jsx

export default function Home() {

return <h1>Home Page</h1>;

}

// About.jsx

export default function About() {

return <h1>About Page</h1>;

}

// Contact.jsx

export default function Contact() {

return <h1>Contact Page</h1>;

}

1. **Cấu hình Router trong ứng dụng**:

Bạn sẽ cần sử dụng BrowserRouter, Routes, và Route từ react-router-dom để định nghĩa các tuyến đường trong ứng dụng.

import React from "react";

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

import Home from "./Home";

import About from "./About";

import Contact from "./Contact";

import Layoutmenu from "./Layoutmenu"; // Đây là component menu bạn đã tạo

function App() {

return (

<Router>

<Layoutmenu /> {/\* Thêm menu vào để nó luôn xuất hiện \*/}

<Routes>

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

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

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

</Routes>

</Router>

);

}

export default App;

* + **<Router>**: Bao bọc toàn bộ ứng dụng của bạn, cho phép sử dụng Router.
  + **<Routes>**: Định nghĩa tất cả các tuyến đường mà ứng dụng của bạn sẽ có.
  + **<Route>**: Định nghĩa từng tuyến đường riêng lẻ. Mỗi tuyến có thuộc tính path là URL và element là trang JSX mà tuyến đó hiển thị.
  + **Layoutmenu**: Đặt thành phần Layoutmenu để nó xuất hiện trên tất cả các trang.

1. **Sử dụng Link trong Menu**:

Trong Layoutmenu, bạn đã sử dụng Link để liên kết đến các trang. Hãy đảm bảo rằng các Link này tương ứng với các tuyến mà bạn đã định nghĩa:

import React, { useEffect, useState } from "react";

import { Disclosure, DisclosureButton, DisclosurePanel } from "@headlessui/react";

import { Bars3Icon, XMarkIcon } from "@heroicons/react/24/outline";

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

import axios from "axios";

function classNames(...classes) {

return classes.filter(Boolean).join(" ");

}

export default function Layoutmenu() {

const [navigation, setNavigation] = useState([]);

useEffect(() => {

axios

.get("/data/navigation.json")

.then((response) => setNavigation(response.data))

.catch((error) => console.error("Error loading JSON:", error));

}, []);

return (

<Disclosure as="nav" className="bg-gray-800 items-center">

<div className="mx-auto max-w-7xl px-2 sm:px-6 lg:px-8">

<div className="relative flex h-16 items-center justify-between">

<div className="absolute inset-y-0 left-0 flex items-center sm:hidden">

<DisclosureButton className="group relative inline-flex items-center justify-center rounded-md p-2 text-gray-400 hover:bg-gray-700 hover:text-white focus:outline-none focus:ring-2 focus:ring-inset focus:ring-white">

<span className="absolute -inset-0.5" />

<span className="sr-only">Open main menu</span>

<Bars3Icon

aria-hidden="true"

className="block h-6 w-6 group-data-[open]:hidden"

/>

<XMarkIcon

aria-hidden="true"

className="hidden h-6 w-6 group-data-[open]:block"

/>

</DisclosureButton>

</div>

<div className="flex flex-1 items-center justify-center sm:items-stretch sm:justify-start">

<div className="hidden sm:ml-6 sm:block">

<div className="flex space-x-4">

{navigation.map((item) => (

<Link

key={item.name}

to={item.href} // Sử dụng 'to' thay vì 'href'

aria-current={item.current ? "page" : undefined}

className={classNames(

item.current

? "bg-gray-900 text-white"

: "text-gray-300 hover:bg-gray-700 hover:text-white",

"rounded-md px-3 py-2 text-sm font-medium"

)}

>

{item.name}

</Link>

))}

</div>

</div>

</div>

</div>

</div>

<DisclosurePanel className="sm:hidden">

<div className="space-y-1 px-2 pb-3 pt-2">

{navigation.map((item) => (

<DisclosureButton

key={item.name}

as={Link} // Sử dụng 'as={Link}' để sử dụng React Router

to={item.href}

aria-current={item.current ? "page" : undefined}

className={classNames(

item.current

? "bg-gray-900 text-white"

: "text-gray-300 hover:bg-gray-700 hover:text-white",

"block rounded-md px-3 py-2 text-base font-medium"

)}

>

{item.name}

</DisclosureButton>

))}

</div>

</DisclosurePanel>

</Disclosure>

);

}

**Bước 3: Điều hướng giữa các trang**

Bây giờ, khi người dùng nhấp vào các liên kết trong menu, ứng dụng sẽ điều hướng đến các trang tương ứng (Home, About, Contact) mà không cần tải lại trang.

**Lợi ích:**

1. **Điều hướng một cách liền mạch**: Không tải lại trang, giúp trải nghiệm người dùng mượt mà.
2. **Tối ưu hóa ứng dụng React**: Việc sử dụng react-router-dom giúp bạn dễ dàng quản lý các tuyến trong ứng dụng React và xây dựng hệ thống điều hướng phức tạp hơn khi cần thiết.

Bạn có thể tiếp tục mở rộng các tuyến đường và trang khác trong ứng dụng của mình bằng cách thêm các Route và liên kết chúng với các component JSX tương ứng.