Form - Style - Router

Form - Style - Router

1. Form
2. Style
3. Router

Form

- Forms là một thành phần quan trọng trong các ứng dụng React vì chúng cho phép người dùng nhập dữ liệu. React cung cấp một cách linh hoạt để xử lý forms, sử dụng sự kết hợp giữa HTML và JavaScript.

Form

```
function MyForm() {
  return (
    <form>
      <label>Enter your name:
        <input type="text" />
      </label>
    </form>
  )
}
```

Form

- Để sử dụng form cần chú ý những điều sau đây:
  - Phải sử dụng state trong component
  - Phải sử dụng sự kiện onChange của các thẻ input trong component nếu lấy dữ liệu.
  - Thêm thuộc tính name của thẻ input trong component.

Form

```
import { useState } from 'react';
import ReactDOM from 'react-dom/client';
```

```
function MyForm() {
  const [name, setName] = useState("");

  const handleSubmit = (event) => {
    event.preventDefault();
    alert(`The name you entered was: ${name}`)
  }

  return (
    <form onSubmit={handleSubmit}>
      <label>Enter your name:
        <input
          type="text"
          value={name}
          onChange={(e) => setName(e.target.value)}
        />
      </label>
      <input type="submit" />
    </form>
  )
}
```

Form

```
import { useState } from 'react';
import ReactDOM from 'react-dom/client';

function MyForm() {
  const [inputs, setInputs] = useState({});

  const handleChange = (event) => {
    const name = event.target.name;
    const value = event.target.value;
    setInputs(values => ({...values, [name]: value}))
  }

  const handleSubmit = (event) => {
    event.preventDefault();
    alert(inputs);
  }
```

```
  return (
    <form onSubmit={handleSubmit}>
      <label>Enter your name:
      <input
        type="text"
        name="username"
        value={inputs.username || ""}
        onChange={handleChange}
      />
      </label>
      <label>Enter your age:
        <input
          type="number"
          name="age"
          value={inputs.age || ""}
          onChange={handleChange}
        />
        </label>
        <input type="submit" />
    </form>
  )
}
```

Style

- Có nhiều các sử dụng Style trong React
  - Viết trong thẻ html
  - Sử dụng file css
  - Sử dụng css module
  - Sử dụng Sass

Style

- Viết style trong thẻ html phải chú ý
  - Các thuộc tính bên trong của css phải viết trong dấu {{}}
  - Các thuộc tính css có dấu gạch nối phải viết theo kiểm camel case (background-color => backgroundColor)
- Đối với sử dụng file css thì chỉ cần import file css vào

- Khi sử dụng css module thì file phải ghi thêm phần mở rộng là .module.css. Rồi sau đó import vào sử dụng như bình thường

```css
.bigblue {
  color: DodgerBlue;
  padding: 40px;
  font-family: Sans-Serif;
  text-align: center;
}
```

```jsx
import styles from './my-style.module.css';

const Car = () => {
  return <h1 className={styles.bigblue}>Hello Car!</h1>;
}

export default Car;
```

Style

- Khi sử dụng Sass
  - Thêm thư viện Sass bằng câu lệnh sau npm i sass
  - Viết code css vào file scss
  - Import vào react để sử dụng

Style

```scss
$myColor: red;

h1 {
  color: $myColor;
}
```

```jsx
import React from 'react';
import ReactDOM from 'react-dom/client';
import './my-sass.scss';

const Header = () => {
  return (
    <>
      <h1>Hello Style!</h1>
      <p>Add a little style!.</p>
    </>
  );
}

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<Header />);
```

Router

- Router là một thành phần cơ bản của thư viện react-router, được sử dụng để quản lý routing trong ứng dụng web. Router cho phép bạn định nghĩa các route và ánh xạ chúng với các thành phần React cụ thể để hiển thị nội dung tương ứng khi URL thay đổi.

Cách 1

- Bước 1: sử dụng câu lệnh npm i -D react-router-dom để thêm thư viện react-router vào ứng dụng
- Bước 2: cấu hình router trong ứng dụng
  - Trong đó <Outlet> được sử dụng để render các thành phần con. <Outlet> sẽ được sử dụng trong thành phần của trang chính để hiển thị nội dung của các trang con khi người dùng điều hướng tới chúng.

Router

```
import ReactDOM from "react-dom/client";
import { BrowserRouter, Routes, Route } from "react-router-dom";
import Layout from "./pages/Layout";
import Home from "./pages/Home";
import Blogs from "./pages/Blogs";
import Contact from "./pages/Contact";
import NoPage from "./pages/NoPage";
```

```
export default function App() {
  return (
    <BrowserRouter>
      <Routes>
        <Route path="/" element={<Layout />}>
          <Route index element={<Home />} />
          <Route path="blogs" element={<Blogs />} />
          <Route path="contact" element={<Contact />} />
          <Route path="*" element={<NoPage />} />
        </Route>
      </Routes>
    </BrowserRouter>
  );
}
```

Router

```
let Layout = function (){

    return(
        <>
            <NavBar/>
            {/*Outlet là nơi hiển thị nội dung của các trang khác của website*/}
            {/* <Home/>  <Shop/> , <Shopdetail/>*/}
            <Outlet/>
            <Footer/>
        </>
    )
}

export default Layout;
```

Cách 2

- Bước 1: Cài đặt React Router npm install react-router-dom
- Bước 2: Sử dụng createBrowserRouter
  - Path: là thuộc tính dùng để xác định đường dẫn URL
  - Element là thuộc tính dùng để xác định component hoặc JSX sẽ được render khi một route tương ứng với path được kích hoạt

Router

- Element là thuộc tính dùng để xác định component hoặc JSX sẽ được render khi một route tương ứng với path được kích hoạt
- Loader: Là một hàm bất đồng bộ được định nghĩa trong route. Cho phép tải dữ liệu trước khi render một component.

Router

- Lazy là thuộc tính dùng để tải component của route khi route đó được kích hoạt (người dùng truy cập vào nó)
- errorElement là thuộc tính chỉ định một component sẽ được hiển thị khi có lỗi xảy ra trong route đó.

Router

```
const router = createBrowserRouter([
  {
    path : '/',
    element: <Layout/>,
    //loader:
    //lazy:
    //errorElement:
    children: [
      {
        index: true,//http://localhost
        element: <Home/>
      },
      {
        path: 'lien-he',//http://localhost/lien-he
        element: <Contact/>
      },
      {
        path: 'danh-muc',//http://localhost/danh-muc
        element: <Shop/>
      },
      {
        path: 'chi-tiet-sp',//http://localhost/chi-tiet-sp
        element: <ShopDetail/>
      }
    ]
  }
])
```

```
function App () {
  return (
    <>
      <RouterProvider router={router}/>
    </>
  )
}
```

Router

- Tham số trong đường dẫn (path) bằng cách sử dụng dấu hai chấm (:). Tham số có thể là bất kỳ tên.

```
{
  path: ':cat',//http://localhost/
  element: <Shop/>
},
{
  path: ':cat/san-pham/:id',//http
  element: <ShopDetail/>
}
```

```
<Route path="/" element={<Layout/>}>
  <Route index element={<Home/>}/>
  <Route path="lien-he" element={<Contact/>}/>
  <Route path=":cat" element={<Shop/>}/>
  <Route path=":cat/san-pham/:id"
  element={<ShopDetail/>}/>
</Route>
```

- Sử dụng Tham số Tùy chọn bằng cách sử dụng dấu hỏi (?) trong đường dẫn.

```
{
  path: ":cat",
  element: <Shop/>
},
{
  path: ":cat/san-pham/:id?",
  element: <ShopDetail/>
}
```

```
<Route index element={<Home/>}/>
<Route path="lien-he" element={<Contact/>}/>
<Route path=":cat" element={<Shop/>}/>
<Route path=":cat/san-pham/:id?"
element={<ShopDetail/>}/>
```

- Truy cập Tham số trong Component: Để truy cập các tham số được định nghĩa trong route, có thể sử dụng hook useParams.

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

let ShopDetail = function() {

    const {id} = useParams()

    return (
```

Router

- Để truy cập các tham số sau dấu chấm hỏi (?) trong URL trong React Router v6, bạn cần sử dụng search parameters.

  Ví dụ: http://localhost/dtdd/san-pham?id=1

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

let ShopDetail = function() {

    const [searchParams, setSearchParams] = useSearchParams();

    const id = searchParams.get('id') || "";

    return (
```