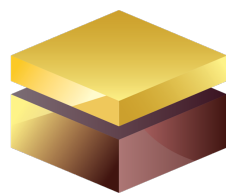
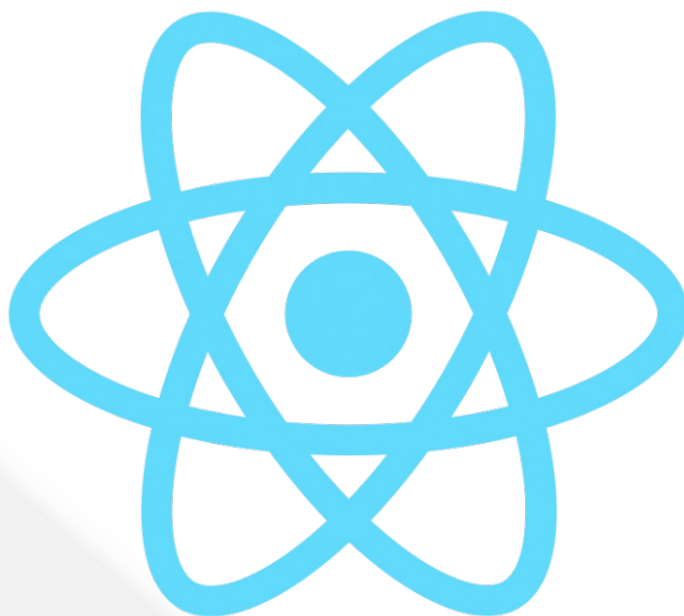
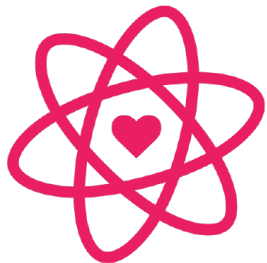


React js



CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

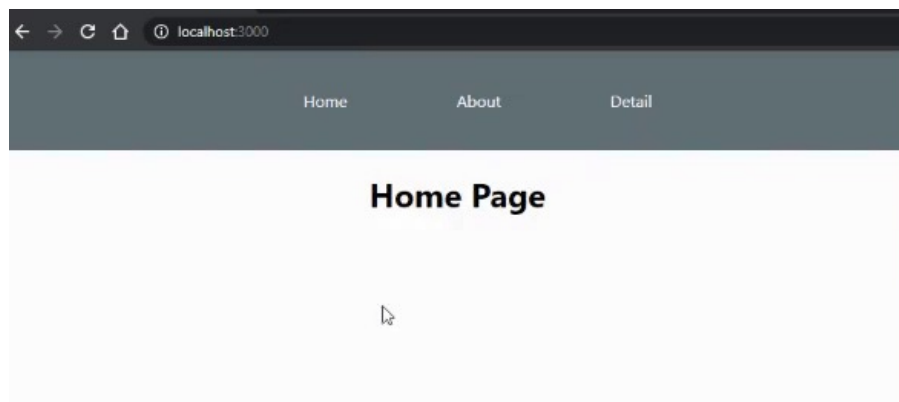
Hướng dẫn : Trương Tấn Khải



React JS

Routing React router dom Ver.6

React Form - Validation



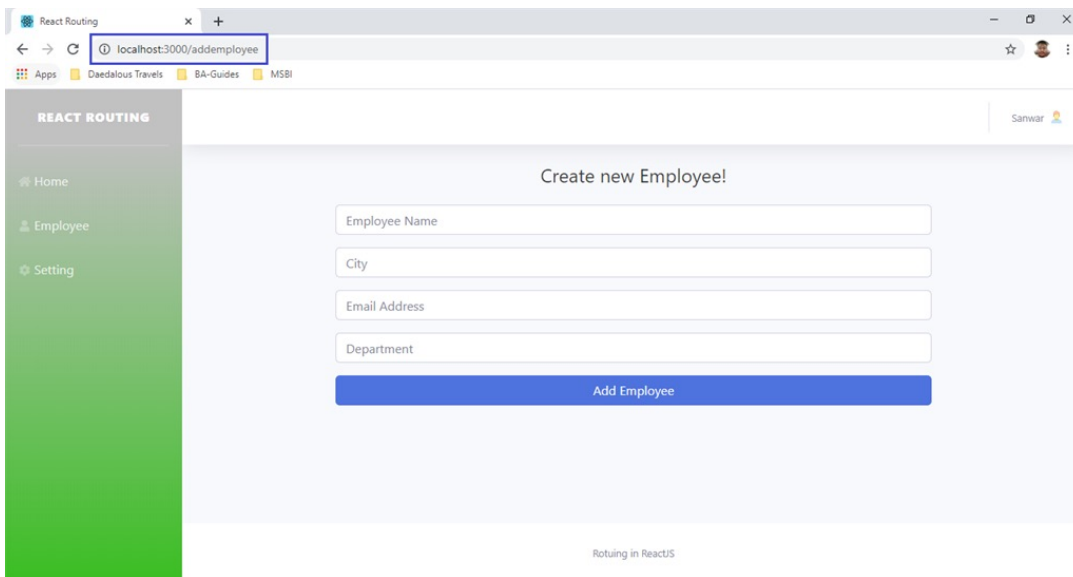
Routing (react-router-dom)

Routing là gì ?

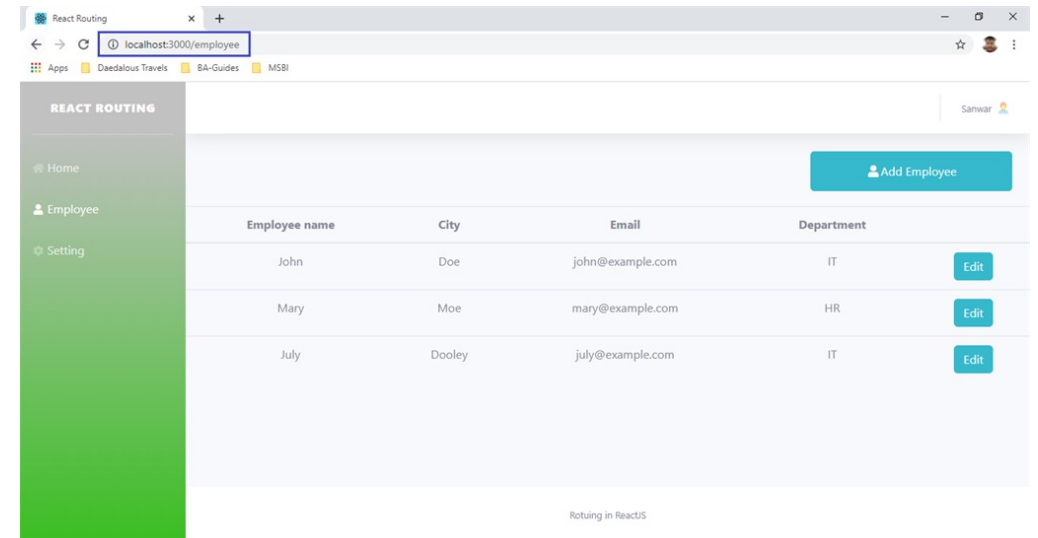
- ❖ Routing là cơ chế trong single page giúp ta chuyển đổi qua lại giữa các component
- ❖ Để sử dụng được routing với reactjs, ta cần package hỗ trợ đó là React-router-dom
- ❖ Tiến hành cài đặt: `npm install --save react-router-dom`
- ❖ Sử dụng: tại app.js, ta đang có 2 component không thể hiện cùng lúc, đó là danh sách khóa học, và danh sách sinh viên . Do đó ta sẽ dùng routing để quản lý, dựa theo đường dẫn url để hiển thị component tương ứng

Document: <https://reactrouter.com/docs/en/v6/getting-started/overview>

Ví dụ:



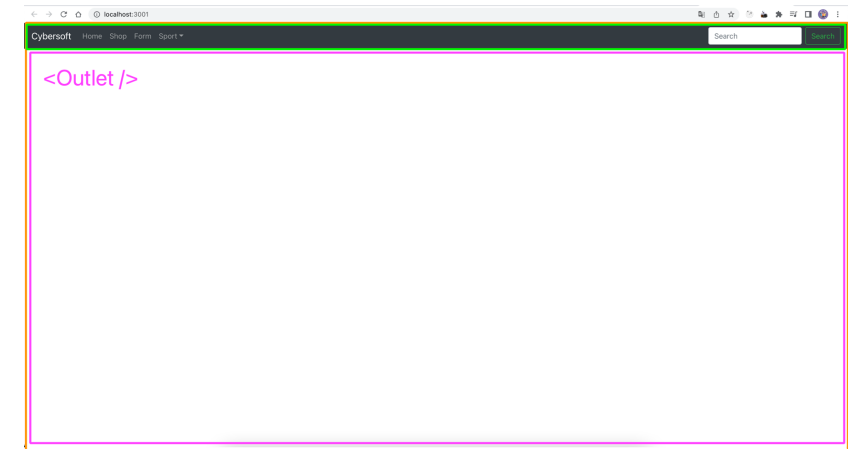
url: <https://localhost:3000/addemployee>



url: <https://localhost:3000/employee>

Routing (react-router-dom)

- Đối với react router dom phiên bản 6. Cơ chế hoạt động như sau:
- React router dom sẽ dùng thẻ BrowserRouter để định nghĩa phần phạm vi nội dung được load ra thành bởi thành url.
- Mỗi phần nội dung load ra được định nghĩa trên 1 thẻ Routes
- Trong mỗi thẻ **Routes** sẽ có các thẻ **Route** quy định nội dung được load.
- Ví dụ bên phải:
 - Thẻ **Route** quy định khi người dùng gõ path là “(rỗng)” thì nội dung mặc định load ra trên trình duyệt là Component **HomeLayout**. (Viền **cam** bên hình mà chúng ta thấy)
 - Bên trong nội dung của **HomeLayout** sẽ được chia nhỏ các phần con như **shop** và **form** theo đường dẫn / tiếp theo

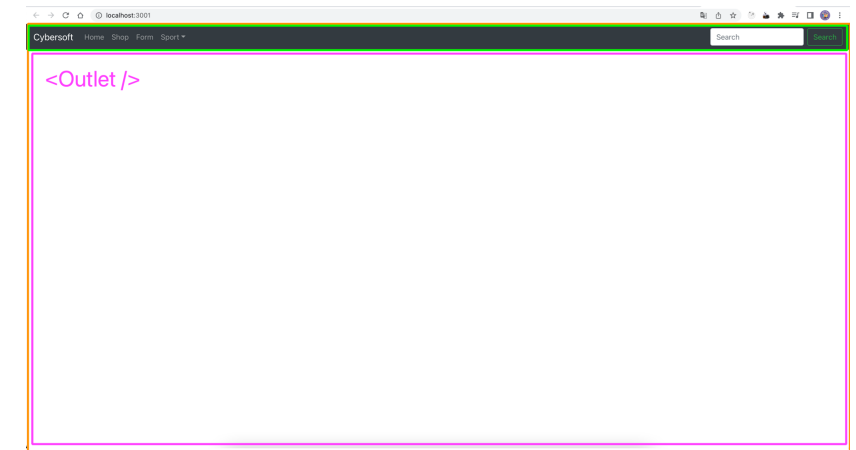


```
import './App.css';
import { Routes, Route, BrowserRouter } from 'react-router-dom'
import Shop from './Pages/Shop';
import HomeLayout from './templates/HomeLayout';
import ReactForm from './Pages/ReactForm/ReactForm';
function App(props) {
  return (
    <BrowserRouter>
      <Routes>
        <Route path="/" element={<HomeLayout />} >
          <Route path="shop" element={<Shop />} />
          <Route path="form" element={<ReactForm />} />
        </Route>
      </Routes>
    </BrowserRouter>
  );
}

export default App;
```

Routing (react-router-dom)

- Nội dung home layout sẽ được load ra với **Header** và 1 thẻ tên **<Outlet />**
- Thẻ header là thẻ cố định còn thẻ Outlet sẽ đại diện khi người dùng tiếp tục gõ /link
- Localhost:3000 / shop => Thẻ outlet sẽ được thế = thẻ ShopComponent
- Localhost:3000/ form => Thẻ outlet sẽ được thế = thẻ FormComponent



```
<Route path="/" element={<HomeLayout />} >
  <Route path="shop" element={<Shop />} />
  <Route path="form" element={<ReactForm />}
/>
</Route>
```

App.jsx

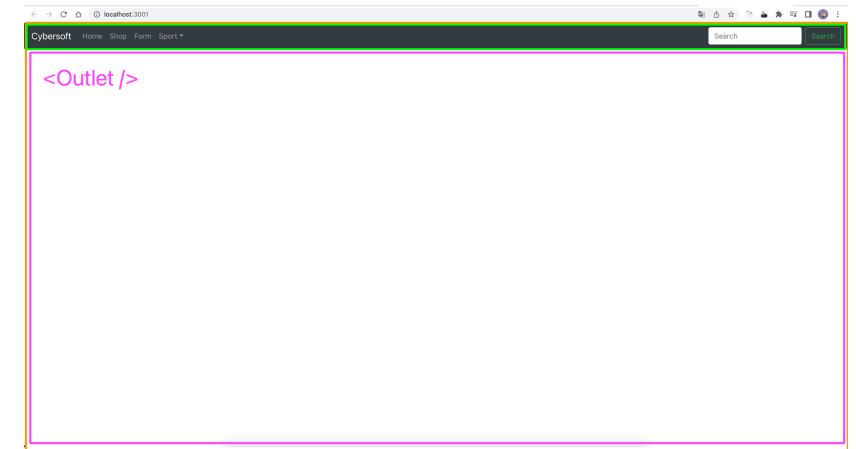
```
import React from 'react'
import { Outlet } from 'react-router-dom'
import HeaderHome from '../components/HeaderHome'

export default function HomeLayout(props) {
  return (
    <>
      <HeaderHome />
      <Outlet />
    </>
  )
}
```

HomeLayout.jsx

Routing (react-router-dom)

- Nội dung home layout sẽ được load ra với **Header** và 1 thẻ tên **<Outlet />**
- Thẻ header là thẻ cố định còn thẻ Outlet sẽ đại diện khi người dùng tiếp tục gõ /link
- Localhost:3000 / shop => Thẻ outlet sẽ được thế = thẻ ShopComponent
- Localhost:3000/ form => Thẻ outlet sẽ được thế = thẻ FormComponent



```
<Route path="/" element={<HomeLayout />} >
  <Route path="shop" element={<Shop />} />
  <Route path="form" element={<ReactForm />}
/>
</Route>
```

App.jsx

```
import React from 'react'
import { Outlet } from 'react-router-dom'
import HeaderHome from '../components/HeaderHome'

export default function HomeLayout(props) {
  return (
    <>
      <HeaderHome />
      <Outlet />
    </>
  )
}
```

HomeLayout.jsx

Routing (react-router-dom)

- Tương tự như vậy ở 1 hệ thống website ta có thể định nghĩa nhiều Routes để build nhiều template khác nhau. Và phân Route theo đặc trưng của từng hệ thống.

```
import './App.css';
import { Routes, Route, BrowserRouter } from 'react-router-dom'
import Shop from './Pages/Shop';
import HomeLayout from './templates/HomeLayout';
import ReactForm from './Pages/ReactForm/ReactForm';
import AdminLayout from './templates/AdminLayout';
import UserManagement from './Pages/Admin/UsersManagement/UserManagement';
import ProductManagement from './Pages/Admin/ProductManagement/ProductManagement';
function App(props) {
  return (
    <BrowserRouter>
      <Routes>
        <Route path="/" element={<HomeLayout />} >
          <Route path="shop" element={<Shop />} />
          <Route path="form" element={<ReactForm />} />
        </Route>
      </Routes>

      <Routes>
        <Route path="/admin" element={<AdminLayout />} >
          <Route path="users" element={<UserManagement />} />
          <Route path="products" element={<ProductManagement />} />
        </Route>
      </Routes>
    </BrowserRouter>
  );
}

export default App;
```

Routing (react-router-dom)

➤ NavLink (Hoặc <Link>)

- Thư viện react router dom hỗ trợ ta thẻ **NavLink** thay thế cho thẻ **<a>** với **href** đổi thành **to**, giúp cho ta có thể chuyển đổi qua lại giữa các trang mà không cần load lại toàn bộ html của trang đó.
- Ngoài ra **<NavLink>** còn cho phép ta custom link css, khi người dùng gõ đúng đường dẫn thì thẻ NavLink sẽ được active class hoặc style thông qua thuộc tính tương ứng **className, style**.
Style và class name sẽ trả ra cho ta thuộc tính active = true (Đối với class ta truyền vào function trả về string Class, đối với style ta truyền vào function trả về object style).

```
<NavLink  
  className={({isActive}) => isActive ? 'nav-link bg-warning text-white' : 'nav-link'}  
  to="/home"  
  style={({isActive}) => isActive ? {color:'red'} : {}}>  
    Home  
</NavLink>
```


Routing (react-router-dom)

➤ Navigate (react-class-component)

- Component navigate được cung cấp bởi react-router-dom hỗ trợ ta điều hướng từ path này tới path khác

```
import React, { Component } from 'react'

export default class UserManagement extends Component {
  render() {
    if(!localStorage.getItem('userLogin')) {
      return <Navigate to="/" replace={false} />
    }
    return (
      <div>
        <h3>admin/users</h3>
      </div>
    )
  }
}
```

- Khi nội dung component gặp đối tượng component `<Navigate />` thì sẽ chuyển hướng về path tương ứng. Lưu ý: có thể chuyển hướng dạng replace