**홈에 홈, 프로필로 가는 Navigation 추가**

1. components에 Navigation.js 파일 추가

import React from "react";

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

const Navigation = () => <nav>

    <ul>

        <li>

            <Link to="/">Home</Link>

        </li>

        <li>

            <Link to="profile">Profile</Link>

        </li>

    </ul>

</nav>;

export default Navigation;

1. Link를 쓰기 위해선, react-router-dom을 import 해야함
2. Link 컴포넌트 안에는 to=””를 지정할 수 있는데, 파일구조에서가 아닌, 현재 화면에서 어디로 가야하는지 지정해주는 거임.
3. 따라서 Home과 profile.js는 둘다 routes 폴더 안에 있지만, home은 로그인 하면 나오는 첫 화면이기에 to=”/”로 지정해주고, profile은 로그인하고 들어가서 있는 화면이기에 to=”profile”로 함.
4. Router.js에 Navigation 컴포넌트를 추가하고, 로그인 되었고, profile로 이동했으면 profile을 보여주도록 요소들을 추가함

**import Navigation from “components/Navigation”;**

**import Profile from “routes/Profile”;**

…

return (

<Router>

**{isLoggedIn && <Navigation />}**

<Switch>

{ isLoggedIn ? (

<>

**<Route exact path=”/”>**

**<Home />**

**</Route>**

**<Route exact path=”/profile”>**

**<Profile />**

**</Route>**

</> )

**log out 하는 버튼 추가하는 파트**

1. 로그인하면 home으로 이동하고, 아까 home에서 profile로 이동하는 로직을 추가했으니, profile에서 로그아웃해보도록 하자.
2. profile.js 에 로그아웃 버튼 추가

return (

<>

<span>Profile</span>

<button>Log out</button>

</>

);

1. 로그아웃은 auth.signOut()으로 할 수 있음. onLogOutClick() 함수를 추가하고 그 안에서 로그아웃

const onLogOutClick = () => {

authService.signOut();

}

…

<button onClick={onLogOutClick}>Log Out</button>

1. 여기까지 하고 로그아웃 버튼을 누르면 로그아웃은 되는지 첫화면으로 이동하지 않음

**Log out하고 첫 화면으로 이동하기**

1. react-router-dom의 Redirect 사용하기

//Router.js

<Switch>

                { isLoggedIn ? (

                <>

                <Route exact path="/">

                    <Home />

                </Route>

                <Route exact path="/profile">

                    <Profile />

                </Route>

                <Redirect from="\*" to="/" />

                </>) :(

                <>

                <Route exact path="/">

                    <Auth />

                </Route>

                <Redirect from="\*" to="/" />

                </>

                )}

            </Switch>

1. 로그인 됐을 때랑, 안됐을 때에 각각 Redirect를 추가한 모습을 확인할 수 있다.
2. Redirect 안에 from to는 말그대로 from에서 to로 간다는 의미
3. Redirect 실행조건은 위 Route들이 정상적으로 실행되면 실행하는데 안되면 Redirect하라는 뜻.
4. 위에선 <Redirect from=”\*” to=”/” />는 현재 화면에서 첫화면으로 가라는 뜻.
5. useHistory 사용법
   1. 만약 profile에서 로그아웃 한 것이므로, profile 내에서 프로그램적으로 해결하고 싶다면 useHistory를 사용할 수도 있다.
   2. 사용법은 다음과 같이 useHistory를 생성하고, .push() 함수를 써주면 명시된 곳으로 이동한다.

 const history = useHistory();

    const onLogOutClick = () => {

         authService.signOut();

         history.push("/");

    }

* 1. 위는 sighOut하고 바로 / 로 이동하라고 코드는 짠 모습이다.