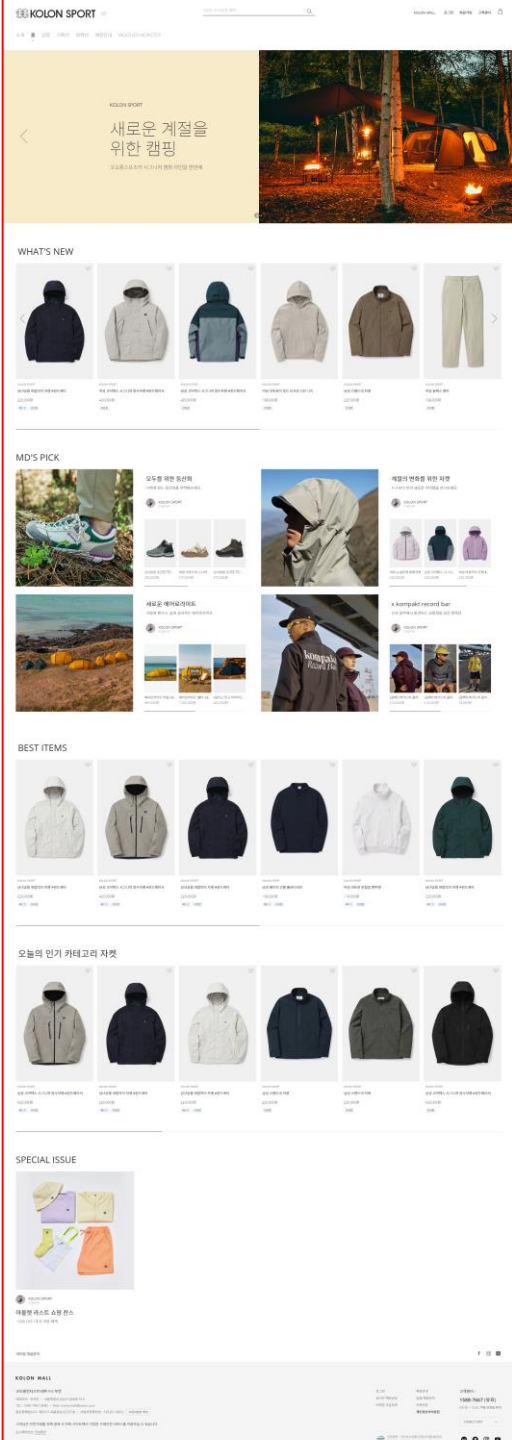


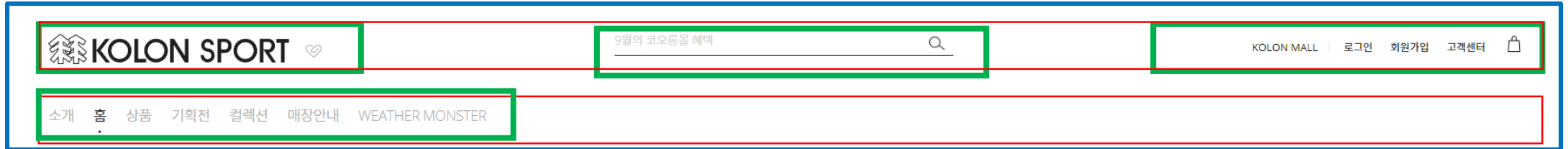
코오롱스포츠 클론코딩

React, Router, json

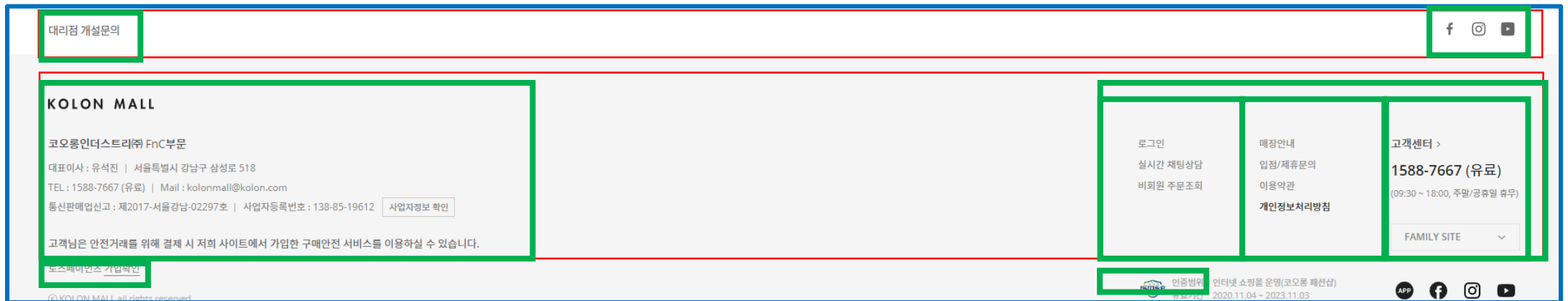


Header
Visual
WHAT'S NEW
MD'S PICK
BEST ITEMS
FOOTER

Header



Footer



Visual

KOLON SPORT

새로운 계절을 위한 캠핑

코오롱스포츠의 시그니처 캠핑 라인을 한번에



WHAT'S NEW

WHAT'S NEW



KOLON SPORT

남녀공용 바람막이 자켓 #윈드케이

220,000원

베스트 신상품



KOLON SPORT

여성 고어텍스 시그니처 방수자켓 #윈드체이서

420,000원

신상품



KOLON SPORT

남성 고어텍스 시그니처 방수자켓 #윈드체이서

420,000원

신상품



KOLON SPORT

여성 코듀로이 후드 티셔츠 (SET UP)

138,000원

신상품



KOLON SPORT

남성 스탠드넥 자켓

220,000원

신상품



KOLON SPORT

여성 슬랙스 팬츠

158,000원

신상품

MD'S PICK

MD'S PICK



모두를 위한 등산화

나에게 맞는 등산화를 선택해보세요



KOLON SPORT
originals



남녀공용 GORE-TEX ...
290,000원



여성 아웃도어 스니커...
270,000원



남녀공용 GORE-TEX ...
370,000원



계절의 변화를 위한 자켓

누구보다 먼저 새로운 아이템을 만나보세요



KOLON SPORT
originals



여성 뉴실루엣 방풍자켓
220,000원



남성 고어텍스 시그니...
420,000원



여성 바람막이 자켓 #...
220,000원



새로운 에어로라이트

가볍게 챙기고, 쉽게 설치하는 에어로라이트



KOLON SPORT
originals



에어로라이트 터널 (AE...
990,000원



에어로라이트 셸터 (AE...
1,050,000원



AEROLITE 3 리미티드...
640,000원



x kompakt record bar

산과 음악에서 발견되는 공통점을 담은 컬렉션



KOLON SPORT
originals



[콤팩트레코드바 콜라...
219,000원



[콤팩트레코드바 콜라...
219,000원



[콤팩트레코드바 콜라...
78,000원

BEST

BEST ITEMS



KOLON SPORT
남녀공용 바람막이 자켓 #윈드케이

220,000원

베스트 신상품



KOLON SPORT
남성 고어텍스 시그니처 방수자켓 #윈드체이서

420,000원

베스트 신상품



KOLON SPORT
남녀공용 바람막이 자켓 #윈드케이

220,000원

베스트 신상품



KOLON SPORT
남성 베이직 긴팔 폴로티셔츠

108,000원

베스트 신상품



KOLON SPORT
여성 비트윈 반집업 맨투맨

118,000원

베스트 신상품



KOLON SPORT
남녀공용 바람막이 자켓 #윈드케이

220,000원

베스트 신상품

복붙설치+6버전 꼭 확인할 것



리액트 라우터

`npm install react-router@6 react-router-dom@6`

클릭

the other, however, it is nice to be able to make a change and refresh the page to see that you did that one step correctly.

1

Install React Router v6

```
npm install react-router@6 react-router-dom@6
```

Update `LocationProvider` to `BrowserRouter`

```
1 // @reach/router
2 import { LocationProvider } from "@reach/router";
3
4 ReactDOM.render(
5   <LocationProvider>
6     <App />
7   </LocationProvider>,
8   el
9 );
```

사용방법

```
11 // React Router v6
12 import { BrowserRouter } from "react-router-dom";
13
14 ReactDOM.render(
15   <BrowserRouter>
16     <App />
17   </BrowserRouter>,
18   el
19 );
```

Update `Router` to `Routes`

You may have more than one, but usually there's just one somewhere near the top of your app. If you have multiple, go ahead and do this for each one.

- 1_1
브라우저라우터를 사용하기 위하여 `index.js` 수정
사용방법 참고 `BrowserRouter`
- 1-2
특정 페이지를 연결할 수 있는 컴포넌트 생성하기
`src` 폴더 밑에, `pages`라는 폴더를 새로 생성
`Pages` 만들 페이지 이름.js 로 저장

Snippets(단축키) 사용
(vscode 에서 확장기능ES7+ 설치해야 작동됨)

rafce 화살표 함수만들기
(react arrow functional export component)
카테고리 중 (RN)은 리액트 네이티브 부분

<https://reactrouter.com/en/main/upgrading/reach>

리액트 라우터

```
Aboutpage.js
src > pages > Aboutpage.js > default
1 import React from 'react'
2
3 const Aboutpage = () => {
4   return (
5     <div>Aboutpage</div>
6   )
7 }
8
9 export default Aboutpage
```

생략해도 됨 리액트 컴포넌트라는 것을 알려주는 코드 작성하고 있는 컴포넌트가 상위 컴포넌트 인지 하위 컴포넌트 인지 파악하고 만약 상위 컴포넌트에 import React from 'react'가 작성되었다면 하위에서는 생략해도 작동이 된다.

- import React from 'react' 브라우저가 읽을 수 있도록 jsx를 바꾸기 위해 작성
- 상위 컴포넌트에 작성되었다면 하위에서는 생략 가능
- React 17버전부터 작성하지 않아도 된다.
- 하지만 굳이 지우지 않아도 됨

화면에 출력 내용

```
return(
  <>
    출력 영역
  </>
)
```

App.js에서 받아 사용할 수 있게 내보내기



페이지의 특성이 들어가는 이름을 지어주세요.

파일명.js/const 파일명/export default 파일명

이름이 같아야 덜 헷갈립니다. 꼭 app.js에서 import 를 해야 됨

더 자세한 내용은 리액트라우터 공식홈페이지에서 확인

json-server component

다운 받는 곳 <https://www.npmjs.com/package/json-server>

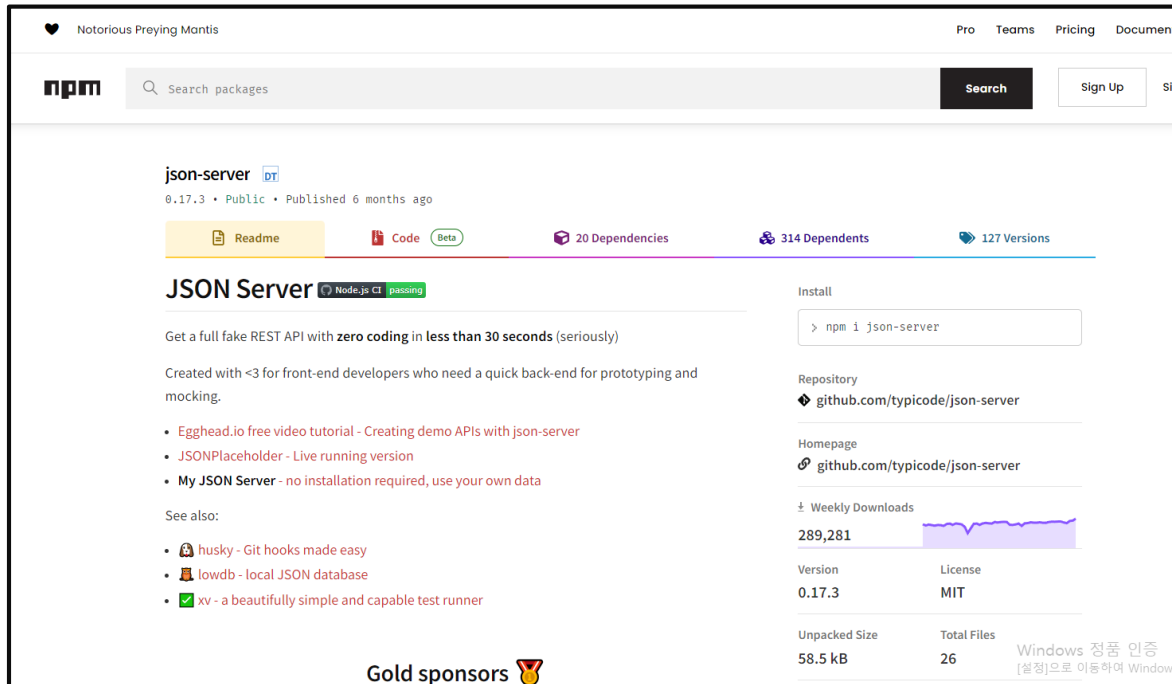


설치

```
npm install -g json-server
```

시작

```
json-server --watch db.json --port 3004
```



1-1

son server는 짧은 시간에
REST API를 구축해주는 라이브러리

1_2




getting started ,
Alternative port
공식홈페이지 참고 해서 작성

관련자료참조

<https://react.vlpt.us/redux-middleware/08-json-server.html>

<https://url.kr/us6gvn>

이번 프로젝트에서 자주 사용하게 될 ReactRouter hooks +

Link(라우터에서 a태그 담당)  새로운 페이지로 이동
Navigate(페이지 이동)
useParams(파라미터값 사용)
useSearchParams(파라미터 읽기)

Redirect (URL 응답)

Restful Route(url작성규칙)

참고

<https://reactrouter.com/en/main/components/link>

<https://reactrouter.com/en/main/components/navigate>

<https://reactrouter.com/en/main/hooks/use-params>

<https://reactrouter.com/en/main/hooks/use-search-params>

<https://medium.com/@shubhangirajagrawal/the-7-restful-routes-a8e84201f206>

<https://hajang.tistory.com/67>

<https://www.learnhowtoprogram.com/c-and-net/basic-web-applications/introduction-to-restful-routing>

Link / `<Link to=""></Link>`

경로 href역할

- React-Router-Dom 이 제공하는 컴포넌트.
- Link 컴포넌트로 만든 부분을 누르면 URL 경로가 바뀌면서 해당 경로로 지정된 컴포넌트가 보여진다.
- 클릭만 하면 이동하기 때문에 다른 연산 과정 없이 페이지를 이동할 때 쓰인다.
- 개발자 도구에는 `` 로 보인다.

◆ Link 컴포넌트와 `<a>`의 차이점?

``는 전체 페이지를 재렌더링 시킨다.

브라우저 주소를 이동하고 페이지 자체를 새로고침 한다. 상태 값이 유지되지 못하고 속도도 저하된다.

`<Link />`는 SPA의 특징에 맞게 필요한 부분만 재렌더링하고 나머지 부분은 그대로 유지된다.

데이터를 필요한 부분만 불러올 수 있기 때문에 속도 향상에 도움이 된다.

◆ 쓰임의 차이

`` 는 외부 프로젝트와 연결 할 때 주로 사용한다.

`<Link />`는 프로젝트 내에서 페이지 전환하는 경우 사용한다.

useNavigate

- 특정 이벤트가 실행됐을 때 동작하도록 하거나 추가적인 로직이 필요한 경우 useNavigate를 사용
- 반환하는 함수를 navigate라는 변수에 저장 후 navigate의 인자로 설정한 path값을 넘겨주면 해당 경로로 이동할 수 있다.

ex) 로그인 페이지에서 아이디와 비밀번호를 입력하고 로그인 버튼을 클릭했을 때 다른 페이지로 이동할 때

사용예시

1. useNavigate를 import
2. 변수에 담는다.(호출하면 양식이 제출된 후 프로그래밍 방식으로 탐색할 수 있는 함수를 반환한다)
3. 함수의 파라미터값으로 가고 싶은 경로를 적는다.

```
1 import { useNavigate } from 'react-router-dom';

const test = () => {
  2 const navigate = useNavigate();
  3 const goToProductDetail = () => {navigate( ' 경로' )}
  return (<>
    <button onClick= { goToProductDetail } >내용</button>
    </> )
  }
  export default test;
```

Link vs useNavigate

1. Link

클릭 시 바로 이동하는 로직 구현 시에 사용

ex) 상품 리스트에서 상세 페이지 이동 시

2. useNavigate

페이지 전환 시 추가로 처리해야 하는 로직이 있을 경우 useNavigate 사용

ex) 로그인 버튼 클릭 시

회원가입 되어 있는 사용자 -> Main 페이지로 이동

회원가입이 되어 있지 않은 사용자 -> SignUp 페이지로 이동

클릭 시 페이지 이동 하는 a 태그처럼 사용 것 **Link**
어떠한 조건을 만족했을 때, 이동을 할 수 있도록 하는 것 **useNavigate**

restful routes

<https://www.learnhowtoprogram.com/c-and-net/basic-web-applications/introduction-to-restful-routing>

https://developer.mozilla.org/ko/docs/Web/API/Fetch_API/Using_Fetch

REST 규칙을 이용해서 만든 api 디자인 패턴

Http 명령어와 URL을 매칭시켜 url디자인을 좀더 단순하고 통일성 있게 만듦

HTTP Method 혹은 HTTP verb 라고 하는 URL로 통신하는 방법

"Representational State Trasfer" 의 약자

자원을 이름(자원의 표현)으로 구분해 해당 자원의 상태(정보)를 주고 받는 모든 것을

-> 자원(resource)의 표현(representation)에 의한 상태 전달

POST 우리가 지금까지 fetch 해서 데이터 가져올 때 항상 씬

GET 새로운데이터 생성,

PUT 기존에 있는 데이터를 수정 할 때

DELETE 데이터를 삭제할 때

네가지 방식으로 url 이 만들어진다.

파라미터란?

컴퓨터 프로그래밍에서 매개변수(parameter 파라미터)란 변수의 특별한 한종류로서, 함수 등과 같은 서브루틴의 인풋으로 제공되는 여러 데이터 중 하나를 가리키기 위해 사용된다

추가적으로 알아야 될 부분 url에서 물음표 뒤에 있는 값 읽어보기

쿼리 스트링(Query string)

쿼리 스트링은 URL 의 한 부분으로, 요청하는 URL 에 부가정보를 포함할 때 사용.

기존 URL 은 단순한 형태의 요청과 응답을 주고 받았지만 쿼리 스트링을 사용하면 조건에 맞게 정렬된 특정 형태의 정보를 요청하고 받을 수 있음.

hyper text transfer protocol path
http://a.com/topic?id=1
http://a.com/topic?id=2
http://a.com/topic?id=3
domain **query string**

쿼리 스트링 형태

쿼리 스트링은 문자열의 형태를 띄며 key=value 로 표현함.

URL 의 일부

? 를 통해 시작 부분을 구분

각 영역의 구분은 &

예시

// 인기순으로 정렬된 정보

https://www.example.com/products?sort=popular

// 인기순으로 정렬된 정보를 내림차순으로 보고 싶다면

https://www.example.com/products?sort=popular&direction=desc

: (콜론)이 있다는 것은 파라미터다 라는 것, :뒤에 오는 내용은 가변적
정리

Restful Route의 필요성

우리가 쇼핑몰 아이템을 보여주는 페이지가 있다고 가정하자
어떤사람은

/showitem 이라고 만들고 /getitem 이라고 만들 수 있다

어떤 아이템은 삭제하는 경우에는 /removeitems or / deleteitems

이렇게 하면 url은 길고 복잡해진다. 이를 해결하기 위해 나온게 restful 패턴이다

url에서 동사는 빼고 이를 Http 명령어로 대체한다

/items + get 명령어 = 아이템읽어오기

/items + post 명령어 = 아이템 생성하기

/items + put 명령어 = 아이템 수정하기

/items + delete명령어 = 아이템 삭제하기

이런 규칙으로 바뀐다

하나의 아이템만 가져오고 싶다면 뒤에 아이템의 id를 붙이는것도 restfulroute의 규칙이다

useParams()

파라미터 정보를 가져와 활용하고 싶을 때 사용

ProductDetail.js

사용예시

1. useParams를 import

2. const{id} = useParams();

오브젝트값을 리턴해서 키값을 가져올 수 있음. 분해하여 가져옴

<https://poiemaweb.com/es6-destructuring>

1. url값 바꾸고 뒤에 /숫자

2. 콘솔에서 파라미터 값 확인

```
import React from 'react';
1 import { useParams } from 'react-router-dom';

const test = () => {
  return (
    2 const { id } = useParams();
    3 <>
      <p> ProductDetail {id} </p>
    </>
  )
}

export default test;
```

useLocation()

현재 URL 에 포함된 여러 가지 정보를 담은 객체를 반환해주는 역할

ProductAll.js

사용예시

1. useLocation를 import
2. 컴포넌트 안에서 데이터를 변수에 담는다.
3. 콘솔에서 확인

```
1 import React from 'react';  
  import { useLocation } from "react-router-dom";  
  
  const test = () => {  
    2 const location = useLocation();  
    3 console.log(location);  
    return (  
      <>  
      </>  
    )  
  }  
  
  export default test;
```

▶ {pathname: '/pages/ProductAll', search: '?q=koLonproduct', hash: '', state: null, key: 'default'}

useSearchParams()

현재 URL 에 포함된 여러 가지 정보를 담은 객체를 반환해주는 역할

ProductAll.js

사용예시

1. useSearchParams 를 import
2. 배열로 만들자

```
const [searchParams,  
setSearchParams] =  
useSearchParams();
```

3. 콘솔에서 setSearchParams에
q뒤에 있는 값을 가져오기

```
import React from 'react';  
1 import {useSearchParams } from "react-router-dom";  
  
const ProductAll = () => {  
  2 const [searchParams, setSearchParams] = useSearchParams();  
  3 console.log( ' 아무말', searchParams.get('q'))};
```

url입력

http://localhost:3000/pages/ProductAll?q=kolonproduct

url읽어 오기 kolonproduct

Redirect(리다이렉트) 경로이동 처리 방법

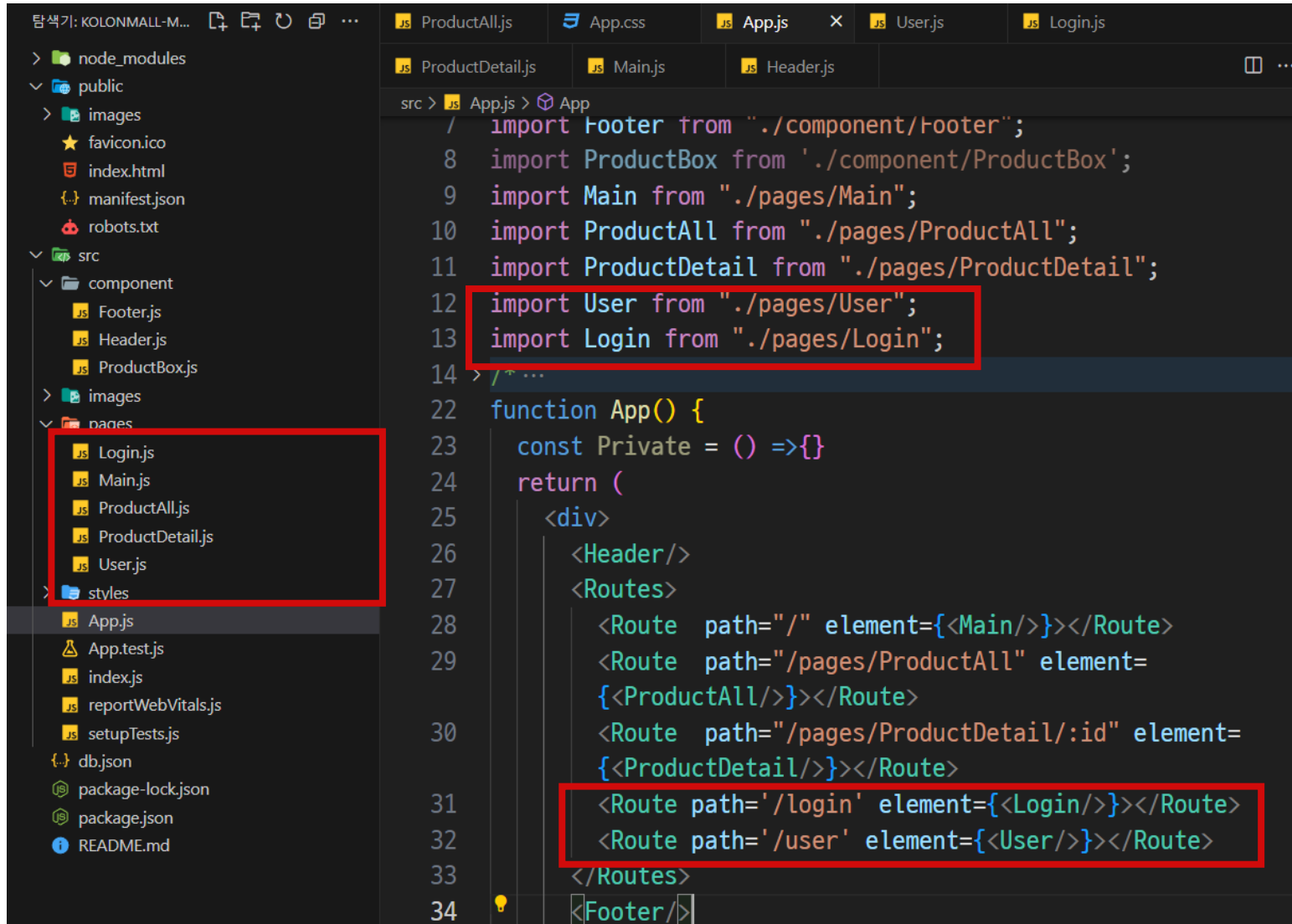
리액트 라우터 v6 에서 새로그침 없이 경로 이동 및 리다이렉션 처리 방법

Pages 폴더에
User.js
Login.js
페이지를 추가한다.

app.js로 와서 Route추가
<Route path='/login' element='<Login/>'></Route>
<Route path='/user' element='<User/>'></Route>

app.js로 와서 Route추가
import User from "./pages/User";
import Login from "./pages/Login";

Redirect(리다이렉트) 경로이동 처리 방법



```
src > App.js > App
/ import Footer from "../component/Footer";
8 import ProductBox from "../component/ProductBox";
9 import Main from "../pages/Main";
10 import ProductAll from "../pages/ProductAll";
11 import ProductDetail from "../pages/ProductDetail";
12 import User from "../pages/User";
13 import Login from "../pages/Login";
14 > / * ...
22 function App() {
23   const Private = () => {}
24   return (
25     <div>
26       <Header/>
27       <Routes>
28         <Route path="/" element={<Main/>}></Route>
29         <Route path="/pages/ProductAll" element=
30           {<ProductAll/>}></Route>
31         <Route path="/pages/ProductDetail/:id" element=
32           {<ProductDetail/>}></Route>
33         <Route path="/login" element={<Login/>}></Route>
34         <Route path="/user" element={<User/>}></Route>
35       </Routes>
36     </div>
37   );
38 }
```

상황 A

유저가 로그인 안한상태 false

```
const [authenticate, SetAuthenticate] = useState(false);
```

```
function App() {  
  const [authenticate, SetAuthenticate] = useState(false);  
  const Private = () =>{  
    return authenticate == true?<User/>:<Navigate to='/login'/>;  
  };  
}
```

URL 확인

http://localhost:3000/user 입력시 return되어서 login페이지로 이동
useState값을 true로 바꾸면 url 이동하지x