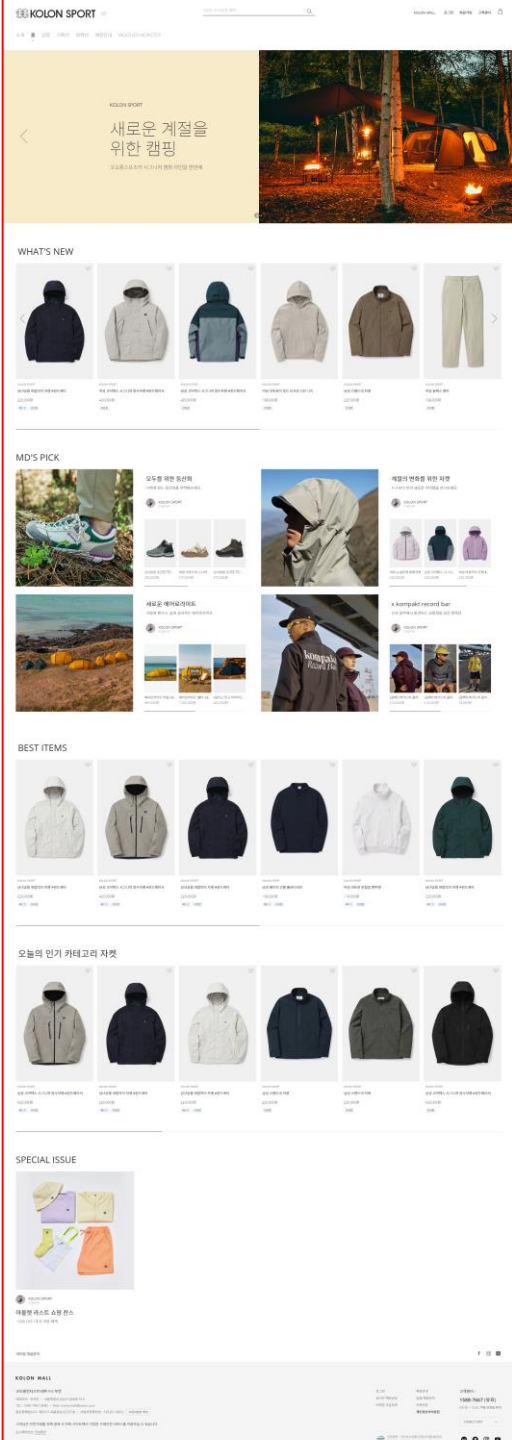


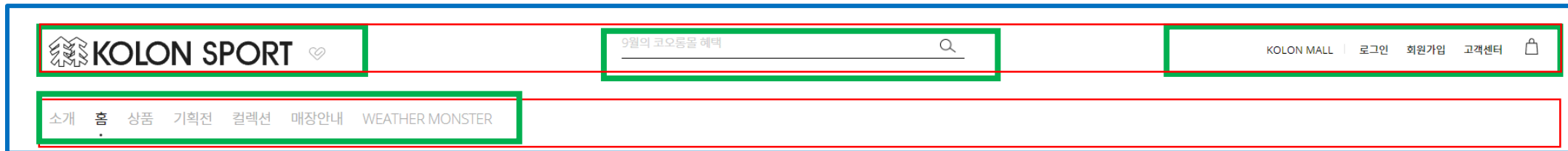
코오롱스포츠 클론코딩

React, Router, json

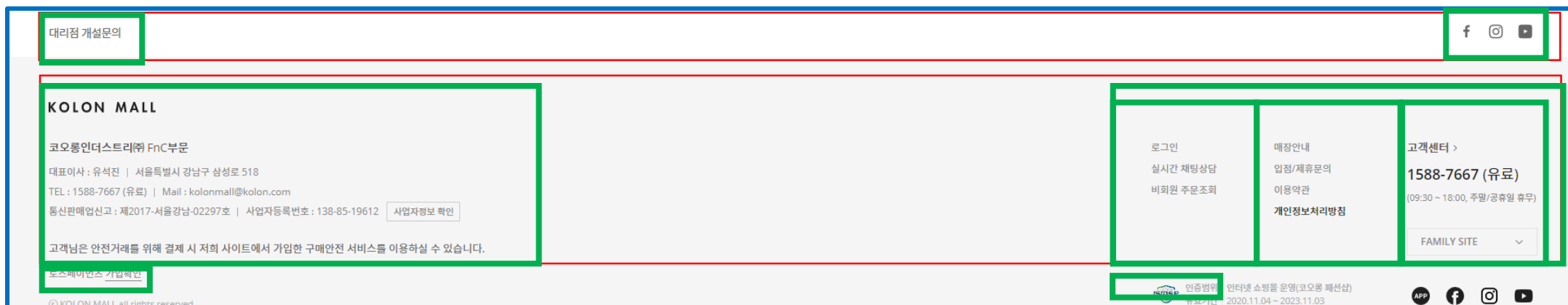


Header
Visual
WHAT'S NEW
MD'S PICK
BEST ITEMS
FOOTER

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



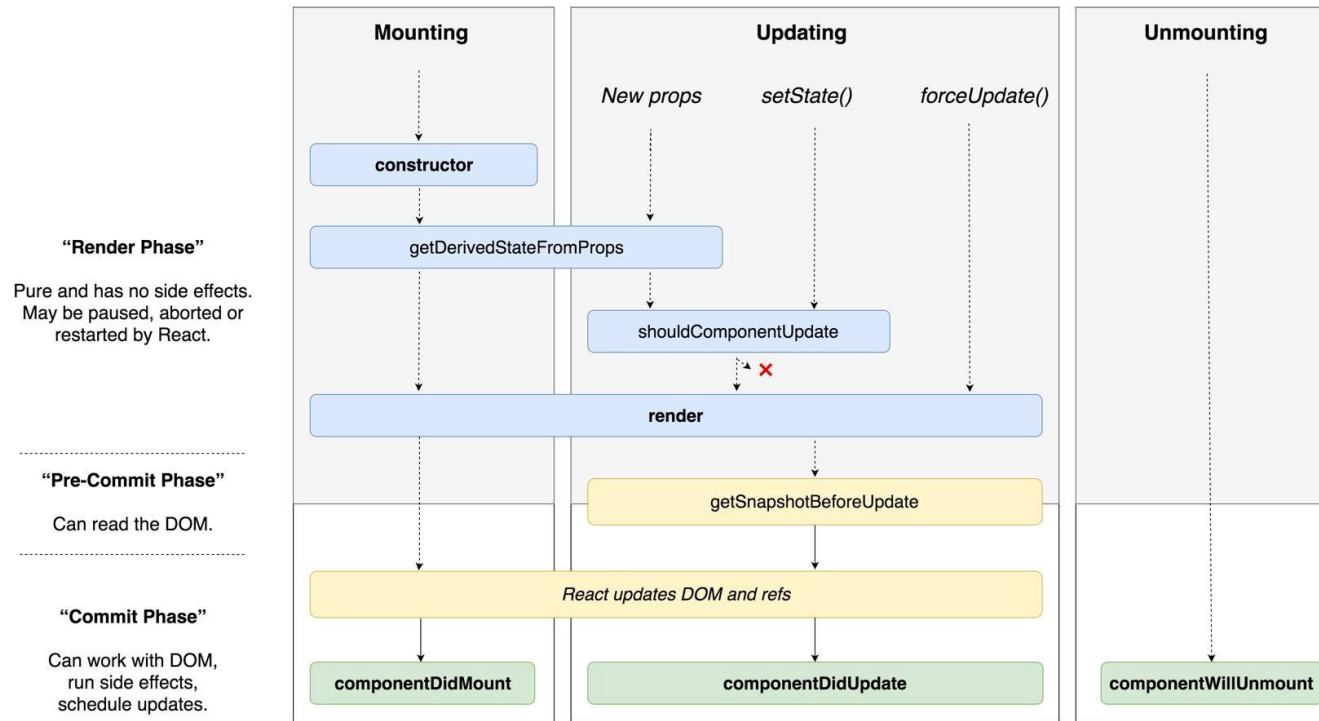
KOLON SPORT
남녀공용 바람막이 자켓 #윈드케이

220,000원

베스트 신상품

useEffect()

useEffect() 함수는 React component가 렌더링 될 때마다 특정 작업(Sied effect)을 실행할 수 있도록 하는 리액트 Hook



- `componentDidMount`: 컴포넌트를 만들고, 첫 렌더링을 다 마친 후 실행.
- `componentDidUpdate`: 리렌더링을 완료한 후 실행. 즉 `render()`가 업데이트될 때마다 실행
- `componentWillUnmount`: 컴포넌트를 DOM에서 제거할 때 실행.

유즈이펙트 사용방법

기본 형태

```
useEffect(function, deps)
```

- function : 실행하고자 하는 함수
- deps : 배열 형태. function을 실행시킬 조건.
deps에 특정값을 넣게 되면 컴포넌트가 mount 될 때, 지정한 값이 업데이트될 때 useEffect를 실행.

useEffect 함수 불러오기

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

유즈이펙트 사용방법

1. Component가 Mount 되었을 때(나타날 때)

```
useEffect(() => { console.log("렌더링 될때마다 실행"); });
```

deps부분을 생략한다면 해당 컴포넌트가 렌더링 될 때마다 useEffect가 실행되게 됩니다. 만약 맨 처음 렌더링 될 때 한 번만 실행하고 싶다면 deps위치에 빈 배열을 넣어준다.

```
useEffect(() => { console.log("맨 처음 렌더링될 때 한 번만 실행"); }, []);
```

유즈이펙트 사용방법

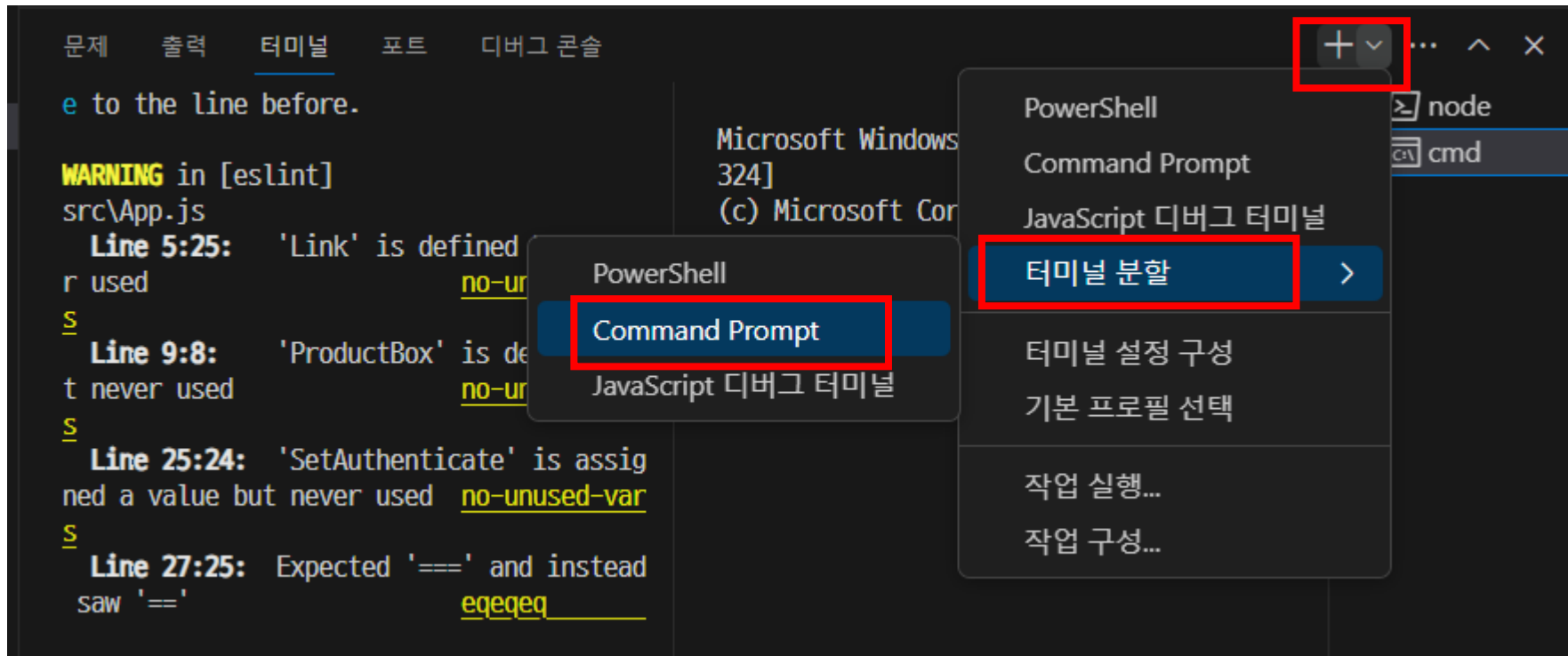
3. Component가 Unmount 되었을 때(사라질 때) & Update 되기 직전에

```
useEffect(() => {  
    console.log("컴포넌트 나타남");  
    console.log(name);  
    return () => {  
        console.log("cleanUp 함수");  
    };  
});
```

- useEffect는 함수를 반환할 수 있는데 이 함수를 cleanup이라고 한다.
- Unmount 될 때만 cleanup 함수를 실행시키고 싶다면 deps에 빈 배열을,
특정 값이 업데이트되기 직전에 cleanup 함수를 실행시키고 싶다면 deps에 해당 값을 넣어주면 됨.

Api호출 fetch()를 이용하여 API불러오기

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



json서버를 설치

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

이런식으로 port번호를 지정하면 된다.

```
json server --watch ./(파일 경로)/(json 파일) --port 숫자()
```

실습

ProductAll.js

```
ProductAll.js x ProductDetail.js
src > pages > ProductAll.js > ProductAll
1 import React, { useEffect } from 'react';
2
3 const ProductAll = () => {
4   useEffect(()=>{},[]);
5   return (
6     <div className='Maininner'>
7       <div>ProductAll</div>
8     </div>
9   )
10 }
11
12 export default ProductAll
```

useEffect(()=>{},[]);

https://developer.mozilla.org/ko/docs/Web/JavaScript/Reference/Statements/async_function
<https://lovethefeel.tistory.com/126>

ProductAll.js

```
1 import React, { useEffect } from 'react';
2
3 const ProductAll = () => {
4   2 const getProduct = async() => {
5     let url = `http://localhost:3004/products`;
6     let response = await fetch(url);
7     let data = await response.json();
8     console.log(data);
9   }
10  3 useEffect(()=>{
11    getProduct()
12  },[]);
13  return (
14    <div className='Maininner'>
15      <div>ProductAll</div>
16    </div>
17  )
18 }
19
20 export default ProductAll
```

import React, { useEffect } from 'react';

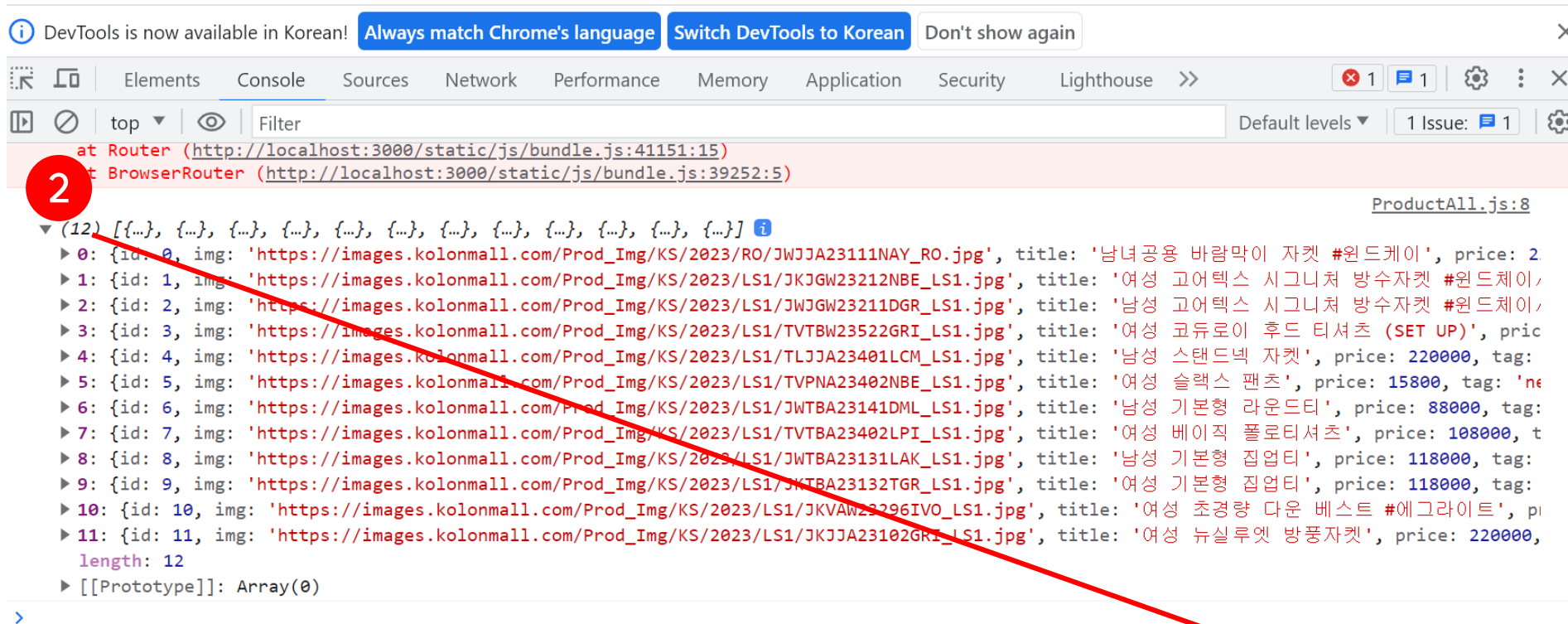
const getProduct = async() => {
 let url = 'http://localhost:3004/products';
 let response = await fetch(url);
 let data = await response.json();
 console.log(data);
}

useEffect(()=>{
 getProduct()
},[]);

확인



http://localhost:3000/pages/ProductAll



데이터값 콘솔에서 확인

useState 사용

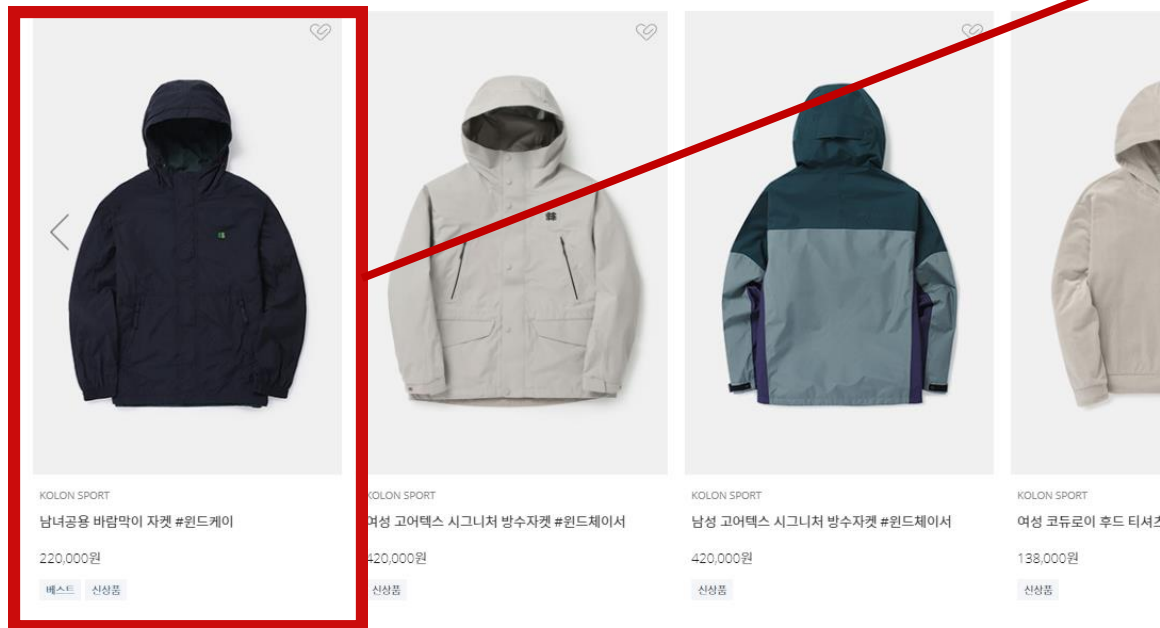
ProductAll.js

```
ProductAll.js x ProductDetail.js
src > pages > ProductAll.js > ProductAll > useEffect() callback
1  import React, { useEffect, useState } from 'react';
2
3  const ProductAll = () => {
4    const [productList, setProductList] = useState([]);
5    const getProduct = async () => {
6      let url = `http://localhost:3004/products`;
7      let response = await fetch(url);
8      let data = await response.json();
9      //console.log(data);
10     setProductList(data);
11   }
12   useEffect(() => {
13     getProduct()
14   }, []);
```

```
const [productList, setProductList] = useState([]);
```

Component > ProductBox.js

WHAT'S NEW



```
ProductBox.js | ProductDetail.js
src > component > ProductBox.js > ProductBox
1 import React from 'react'
2
3 const ProductBox = () => {
4   return (
5     <div>
6       <img src='https://images.kolonmall.com/Prod_Img/KS/2023/R0/JWJJA23111NAY_R0.jpg'></img>
7       <p>KOLON SPORT</p>
8       <h2>제목</h2>
9       <h3>가격</h3>
10      <p>신제품</p>
11    </div>
12  )
13 }
14
15 export default ProductBox
```

```
src > component > ProductBox.js > ProductBox
1 import React from 'react'
2
3 const ProductBox = () => {
4   return (
5     <div>
6       <img src='https://images.kolonmall.com/Prod_Img/KS/2023/R0/JWJJA23111NAY_R0.jpg'></img>
7       <p>KOLON SPORT</p>
8       <h2>제목</h2>
9       <h3>가격</h3>
10      <p>신제품</p>
11    </div>
12  )
13 }
14
15 export default ProductBox

src > pages > ProductAll.js > ProductAll
1 import React, { useEffect, useState } from 'react';
2 import ProductBox from '../component/ProductBox';
3
4 const ProductAll = () => {
5   const [productList, setProductList] = useState([]);
6   const getProduct = async () => {
7     let url = `http://localhost:3004/products`;
8     let response = await fetch(url);
9     let data = await response.json();
10    //console.log(data);
11    setProductList(data);
12  }
13  useEffect(()=>{
14    getProduct()
15  },[]);
16  return (
17    <div className='Maininner'>
18      <div>
19        <ProductBox/>
20      </div>
21    </div>
22  )
23 }
24
25 export default ProductAll
```

혼자 해보기

```
1 import React, { useEffect, useState } from 'react';
2 import ProductBox from '../component/ProductBox';
3
4 const ProductAll = () => {
5   const [productList, setProductList] = useState([]);
6   const getProduct = async () => {
7     let url = `http://localhost:3004/products`;
8     let response = await fetch(url);
9     let data = await response.json();
10    //console.log(data);
11    setProductList(data);
12  }
13  useEffect(()=>{
14    getProduct()
15  },[]);
16  return (
17    <div className='Maininner'>
18      <div>
19        <ProductBox/>
20        <ProductBox/>
21        <ProductBox/>
22        <ProductBox/>
23      </div>
24    </div>
25  )
26 }
27
28 export default ProductAll
```

ProductBox를 map()활용해서
보여주세요

ProductAll.js

```
return (  
  <div className='Maininner'>  
    <div>  
      <ul>  
        {productList.map((menu)=>(  
          <li>  
            <ProductBox item={menu}/>  
          </li>  
        ))}  
      </ul>  
    </div>  
  </div>  
)
```

ProductBox를 map()활용해서
보여주세요

```
<ul>  
  {productList.map((menu)=>(  
    <li>  
      <ProductBox item={menu}/>  
    </li>  
  ))}  
</ul>
```

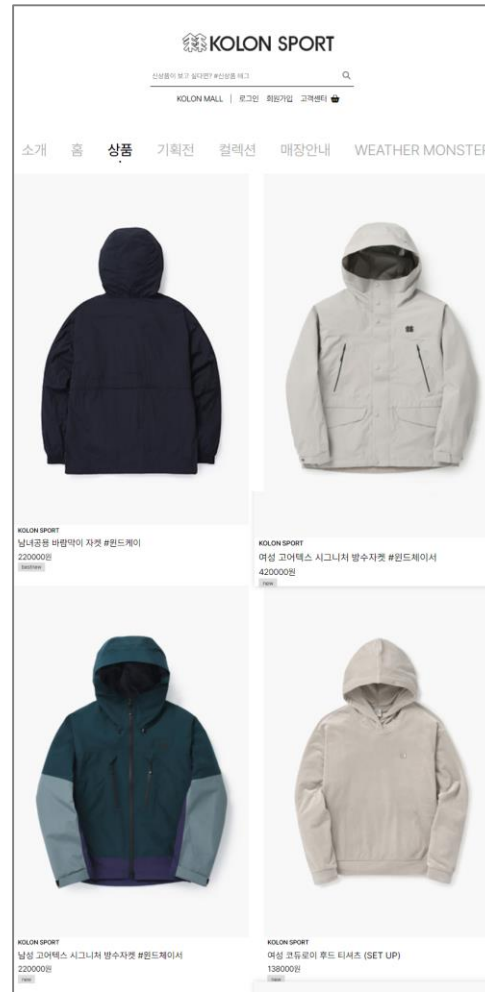
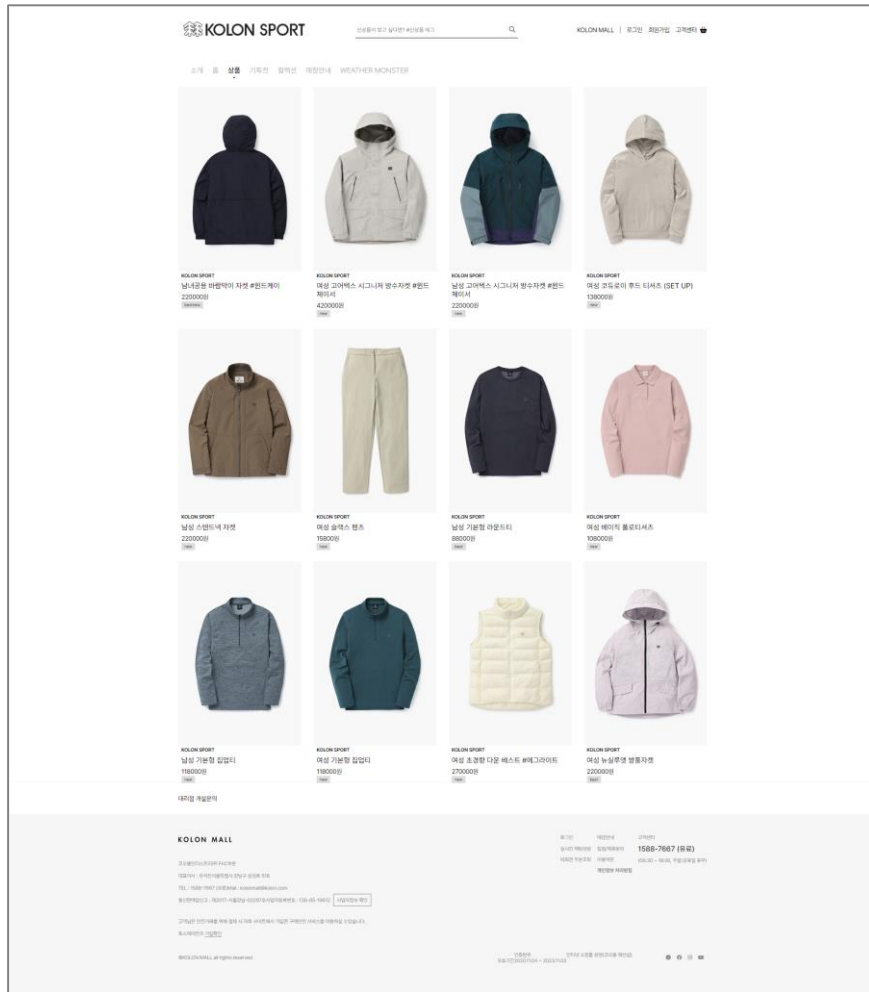

ProductBox.js

```
return (  
  <div className='Maininner'>  
    <div>  
      <ul>  
        {productList.map((menu)=>(  
          <li>  
            <ProductBox item={menu}/>  
          </li>  
        ))}  
      </ul>  
    </div>  
  </div>  
)
```

ProductBox를 map()활용해서
보여주세요

```
<ul>  
  {productList.map((menu)=>(  
    <li>  
      <ProductBox item={menu}/>  
    </li>  
  ))}  
</ul>
```

미디어 쿼리 사용해서 각 분기점 디자인해주세요



```
/* desktop 규격 */
@media screen and (min-width: 1024px){
/* 4단배열 */
}

/* tablet 규격 */
@media screen and (max-width: 1023px){
/* 2단배열 , 헤더 영역 1단 */
}

/* mobile 규격 */
@media screen and (max-width: 540px){
/* 2단배열 , 헤더 영역 1단 */
}
```

Form, Submit 로그인 버튼을 누르면 로그인 페이지를 출력

Header.js

```
Header.js
1 import React, {useState} from 'react';
2 import {Link, useNavigate} from 'react-router-dom';
3 const Header = () => {
4   const [activeLink, setActiveLink] = useState(0); // 초기값 0
5   const LinkClick = (index) => {setActiveLink(index); // 클릭시 이벤트
6 };
7 const navigate = useNavigate();
8 const goToLogin = () => {
9   navigate('/pages/login')
10 };
11 return (
12   <div className='header'>
13     <div className='HeaderTop inner'>
14       <Link to="/" className='logo'>
15     </Link>
16     <form>
17       <input type='search' placeholder='신상품이 보고 싶다면? #신상품 태그' />
18       <i class="fa-solid fa-magnifying-glass"></i>
19     </form>
20     <div className='snb'>
21       <ul>
22         <li>KOLON MALL</li>
23         <li onClick={goToLogin}>로그인</li>
24         <li>회원가입</li>
25         <li>고객센터</li>
26         <li><Link to = '/'><i class="fa-solid fa-basket-shopping"></i></Link></li>
27       </ul>
28     </div>
29   </div>
```

useNavigate

1. react-router-dom으로부터 useNavigate를 import.

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

2. useNavigate 변수에 담는다

```
const navigate = useNavigate();
```

3. 클릭이벤트를 추가해 함수를 담아주고

```
<li onClick={goToLogin}>로그인</li>
```

4. 함수 실행 시 가는 페이지를 경로로 지정한다

```
const goToLogin = () => {
  navigate('이동할 페이지 경로')
};
```

로그인페이지 디자인 해주세요

```
Header.js  Login.js  App.js  Header.css
src > pages > Login.js > Login
1  import React from 'react'
2
3  const Login = () => {
4    const loginUser = (event) => {
5      event.preventDefault();
6      console.log('login user');
7    }
8    return (
9      <div className='inner'>
10        <form onSubmit={loginUser}>
11          <label for='email'>이메일을 입력해주세요</label>
12          <input type='email' placeholder='이메일을 입력 해주세요' id='email' /> <br />
13          <label for='password'>비밀번호를 입력해주세요</label>
14          <input type='password' placeholder='비밀번호를 입력 해주세요' id='password' /> <br />
15          <button type='submit'>로그인</button>
16        </form>
17      </div>
18    )
19  }
20
```

Form태그 사용시
페이지가 새로고침 되는 것을
막기위해
onSubmit함수를 사용

event.preventDefault();
함수를 꼭 써주세요

상품 디테일을 눌렀으나, 로그인 안되었을 경우 로그인 페이지가 먼저나온다

```
function App() {  
  const [authenticate, SetAuthenticate] = useState(true); //true면 로그인이 되고 f  
  const Private = () => {  
    return authenticate == true ? <User/> : <Navigate to='/login' />;  
  }  
  return (  
    <div>  
      <Header/>  
      <Routes>  
        <Route path="/" element={<Homepage/>}></Route>  
        <Route path="/pages/Aboutpage" element={<Aboutpage/>}></Route>  
        <Route path="/pages/ProductAllpage" element={<ProductAllpage/>}></Route>  
        <Route  
          path="/pages/ProductDetailpage/:id"  
          element={<Private authenticate={authenticate}/>}></Route>  
      </Routes>  
    </div>  
  );  
}
```

1. src폴더에 Route 폴더 만들고 Private.js 만들기
2. app.js에서 import 해주세요.
3. <Route
 path="/pages/ProductDetailpage/:id"
 element={<Private authenticate={authenticate}/>}>
 </Route>
4. Authenticate를 props로 받아오기


```

Header.js  JS Login.js  JS App.js  JS ProductBox.js X  JS Private.js  Header.css
src > components > JS ProductBox.js > [ProductBox]
1 import React from 'react'
2 2 import {useNavigate} from 'react-router-dom';
3
4 const ProductBox = ({item}) => {
5 3 const navigate = useNavigate();
6   const showDetail = () =>{
7     navigate(`/product/${item.id}`);
8   }
9   return (
10 1 <div className='List' onClick={showDetail}>

```

4

localhost:3000/product/0

ProductBox.js에서

1. div를 onClick 했을 때 {showDetail}함수로 넘어가게 한다.
2. useNavigate를 import한다
3. ProductBox에 {item}함수를 인자로 받아서 useNavigatehook으로 원하는 페이지로 이동하게 만든다
\${item.id}

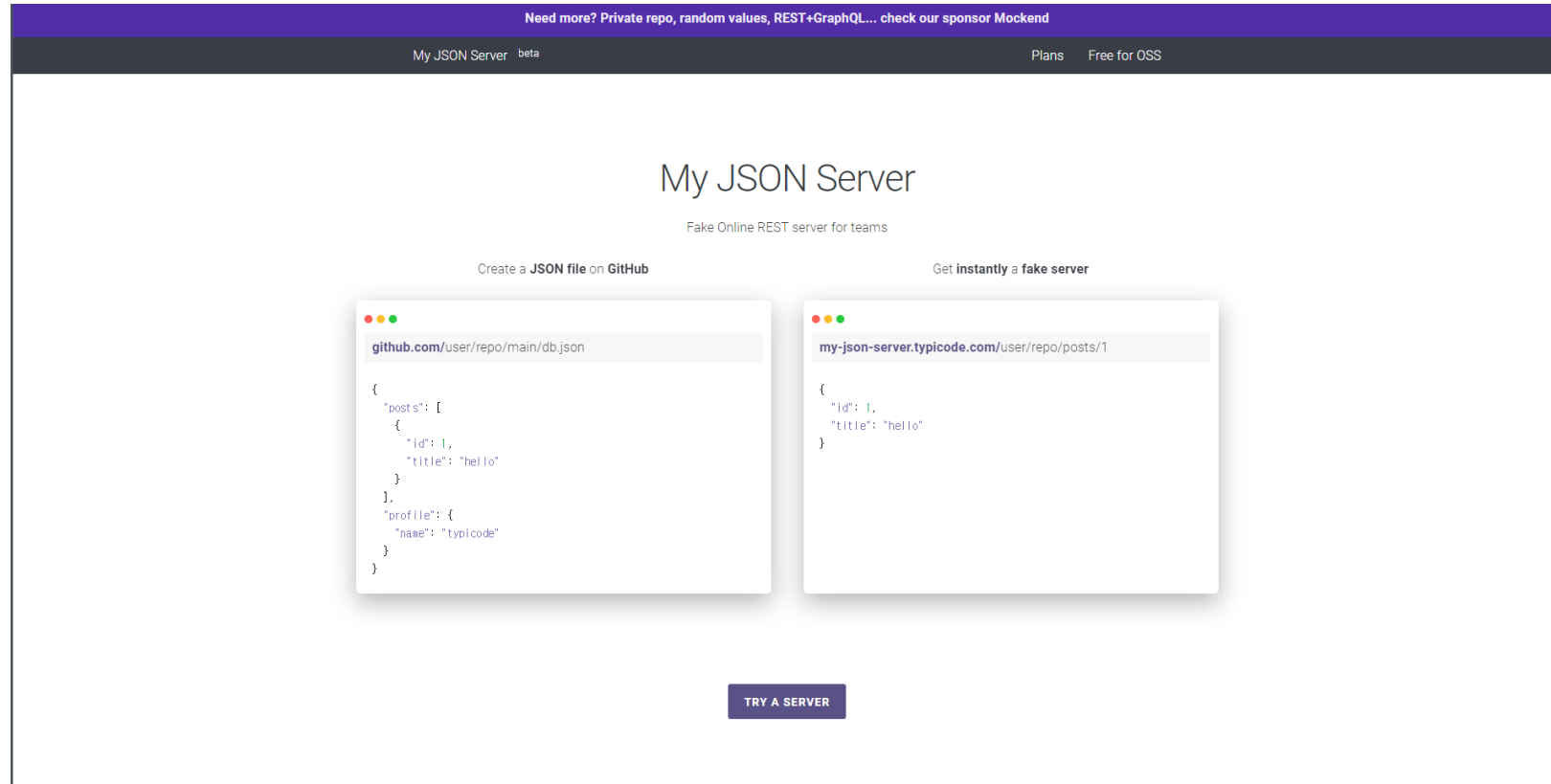
4. 클릭시 url이 아이디 값을 받아 오는지 확인한다.

템플릿 리터럴 사용방법 참고

<https://url.kr/bgxhqm>

https://my-json-server.typicode.com/

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



https://my-json-server.typicode.com/

1. db.json 구축

2. Github 래퍼지토리 확인

my-json-server.typicode.com/<github닉네임><repo주소>

