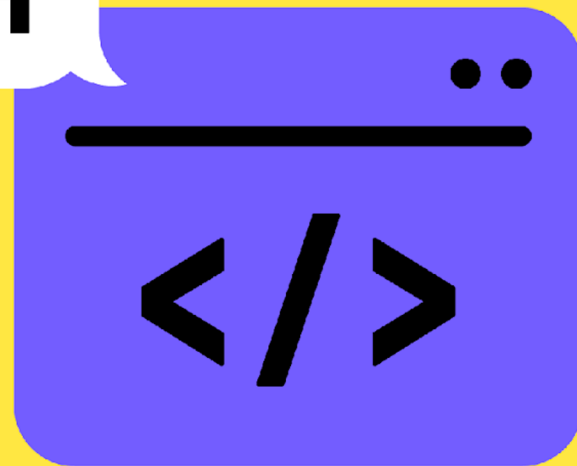


카카오 테크 캠퍼스

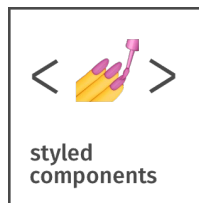
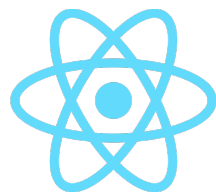
2단계 수상작 발표 자료

발표자 : 전남대 FE 경주원

TECH



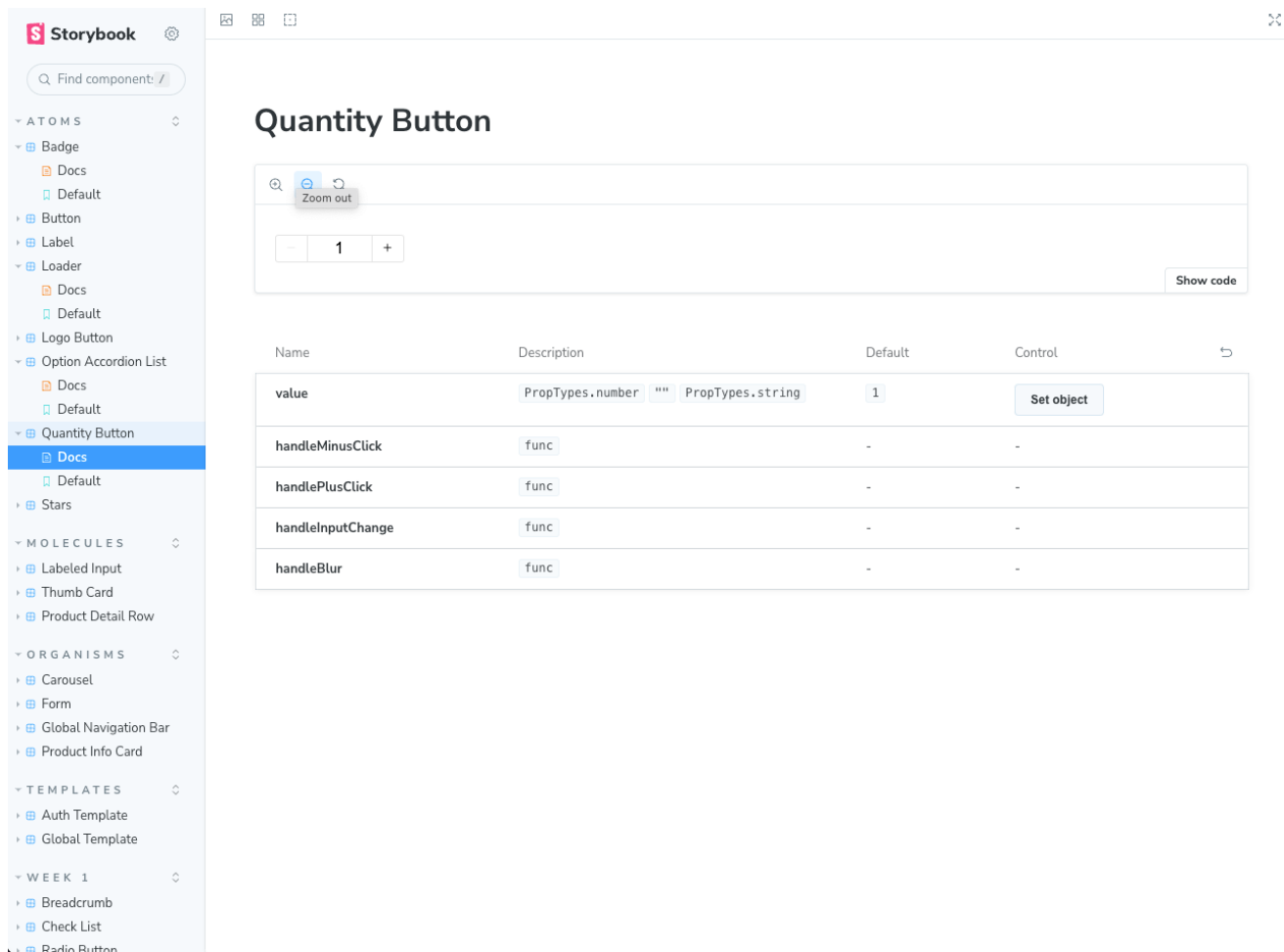
기술 스택



Jōtai




01 최종 과제 구현 모습



storybook을 통해 컴포넌트를 문서화하고 재사용성을 높이자

- 1주차 과제의 과제 3: 컴포넌트 테스트 화면 구현
- 2주차 과제의 과제 1: 아토믹 컴포넌트 디자인 패턴
- 컴포넌트 테스트? 아토믹 컴포넌트? 그렇다면 스토리북이지!
- 나만의 목표: 모든 컴포넌트에 스토리북을 적용해 보자
- 그러나 주차가 늘어날수록 컴포넌트는 매우 많았다 🤖
- 개발 전에 UI 디자인을 재사용 가능하게 나누었다면 더 도움이 되었을 것


src/hooks

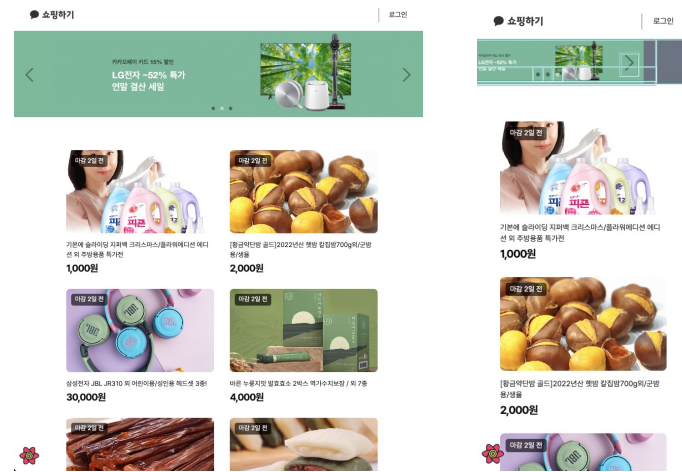
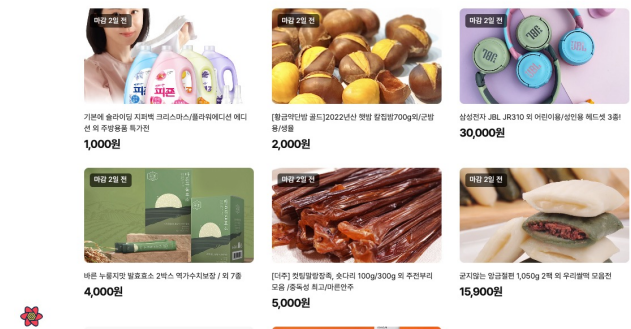
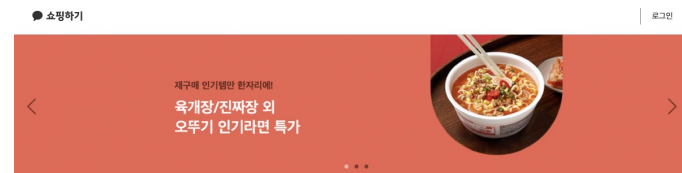
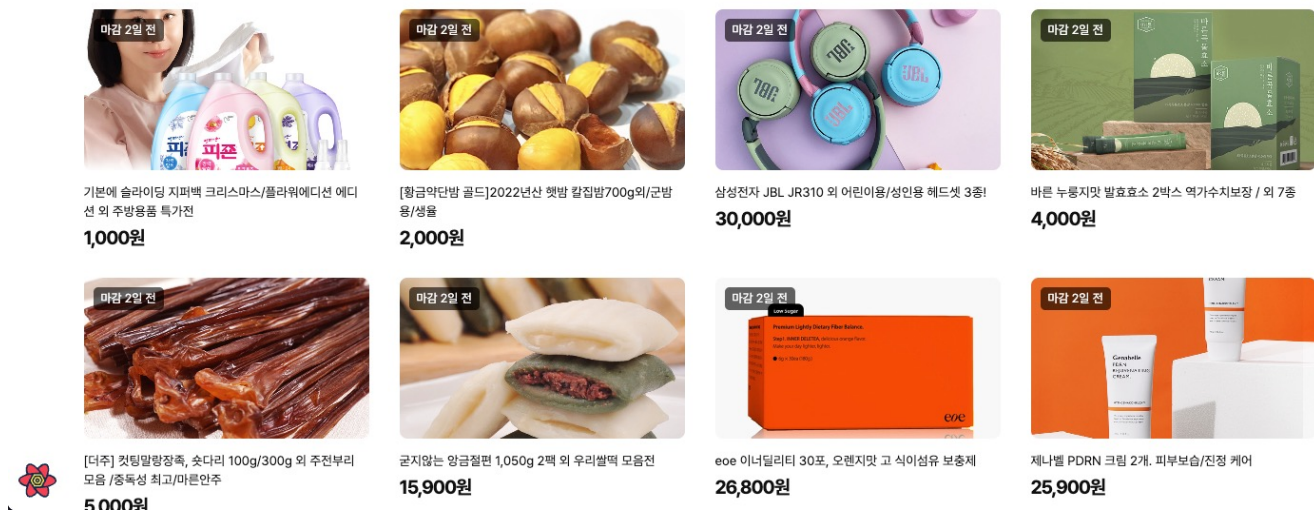
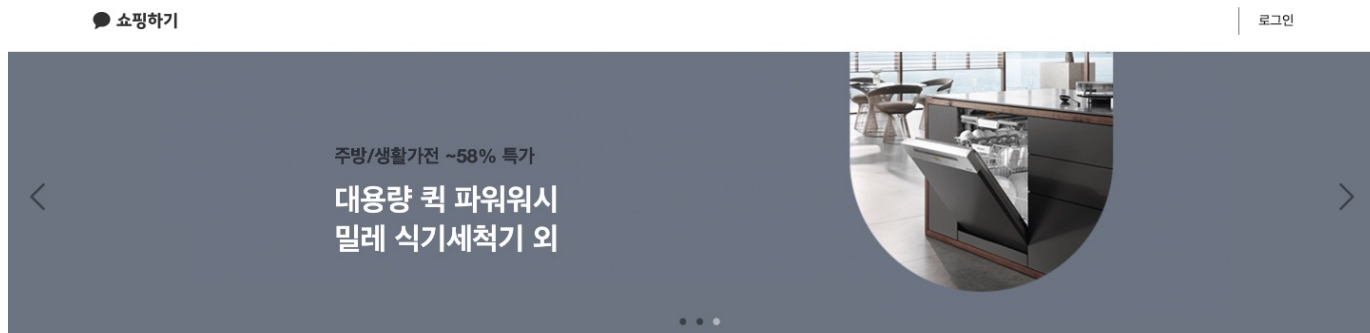
- ├─ useAddCartItemMutation.js
- ├─ useBodyStyleFixed.js
- ├─ useGetCartItemQuery.js
- ├─ useGetInfiniteProductsQuery.js
- ├─ useGetOrderResult.js
- ├─ useGetProductByIdQuery.js
- ├─ useIntersectionObserver.js
- ├─ useOutsideClick.js
- ├─ usePostOrderMutation.js
- ├─ useSaveOrderMutation.js
- └─ useUpdateCartItemMutation.js

custom hook으로 다양한 로직을 분리하자

- react query의 query, mutation 관련 훅 분리
- useIntersectionObserver: Intersection Observer API 로직을 분리한 커스텀 훅
- useOutsideClick({ref, callback}): ref 외부에 클릭 이벤트 발생시 callback 함수를 실행하는 커스텀 훅
- useBodyStyleFixed: root의 scroll을 막아주는 커스텀 훅

01 최종 과제 구현 모습

여러 디바이스에서 쉽게 접근할 수 있는 반응형 디자인 - grid



01 최종 과제 구현 모습

여러 디바이스에서 쉽게 접근할 수 있는 반응형 디자인 - flex

쇼핑하기

로그인



★★★★★

기본에 슬라이딩 지퍼백 크리스마스/플라워에디션 에디션 외 주방용품 특가전

1,000원~

옵션 선택

구성

01. 슬라이딩 지퍼백 크리스마스에디션 4종

- 2 + 20,000원

02. 슬라이딩 지퍼백 플라워에디션 5종

- 1 + 10,900원

뽑아쓰는 키친타올 130매 12팩

- 1 + 16,900원

총 수량 4개

총 주문금액 47,800원

장바구니

구매하기

쇼핑하기

로그인



★★★★★

기본에 슬라이딩 지퍼백 크리스마스/플라워에디션 에디션 외 주방용품 특가전

1,000원~

옵션 선택

구성

01. 슬라이딩 지퍼백 크리스마스에디션 4종

- 1 + 10,000원

총 수량 1개

총 주문금액 10,000원

장바구니

구매하기

쇼핑하기

로그아웃



★★★★★

기본에 슬라이딩 지퍼백 크리스마스/플라워에디션 에디션 외 주방용품 특가전

1,000원~

옵션 선택

구성



01 최종 과제 구현 모습

다양한 라이브러리를 사용해 상태 혹은 데이터 관리하자 - jotai

```
import { atomWithStorage } from "jotai/utils";

const accessTokenAtom = atomWithStorage("accessToken", "");

function SignIn() {
  const setAccessToken = useSetAtom(accessTokenAtom);

  const onSignInSubmit = async (data) => {
    try {
      const response = await authAPI.signIn(data);
      setAccessToken(response.headers.authorization);
    }
  }
}
```

- atomic한 전역 상태 관리
- bottom-up 방식
- accessToken을 localStorage에 저장하자!
- atomWithStorage: localStorage나 sessionStorage에 상태를 저장하고 사용할 수 있다

01 최종 과제 구현 모습

다양한 라이브러리를 사용해 상태 혹은 데이터 관리하자 - react hook form

이메일

joowoni99@gmail.com

중복된 이메일입니다

비밀번호

.....

영문, 숫자, 특수문자 모두 포함

이름

joowon

회원가입

- 사용하기 쉬운 유효성 검사
- 비제어 컴포넌트 ➡ 리렌더링 횟수 및 유효성 검사 계산 최소화
- FormProvider, useFormContext를 사용해 컴포넌트 분리 및 재사용성 높이기
- constant로 form 관련 데이터(id, label, type, placeholder, validation) 관리 ➡ 가독성 좋은 코드

01 최종 과제 구현 모습

다양한 라이브러리를 사용해 상태 혹은 데이터 관리하자 - react query의 useInfiniteQuery

```
const { data, fetchNextPage } = useInfiniteQuery(
  [API.KEYS.GET_ALL_PRODUCTS],
  async ({ pageParam = 0 }) => {
    const { data } = await productAPI.getAllProducts({
      pageIndex: pageParam,
    });
    return data.response;
  },
  {
    getNextPageParam: (lastPage, allPages) => {
      if (lastPage.length === 0) {
        changeLoaderDisplay();
        return undefined;
      }
      return allPages.length;
    },
    suspense: true,
    retry: 1,
  }
);
```

- 파라미터 값만 변경하여 동일한 useQuery를 무한정 호출할 때 사용
- getNextPageParam: 다음 API 호출 때 쿼리 함수에 넘어갈 pageIndex를 리턴
- 무한 스크롤을 구현할 때 데이터를 쉽게 관리할 수 있다

01 최종 과제 구현 모습

다양한 라이브러리를 사용해 상태 혹은 데이터 관리하자 - react query의 Query Invalidation

```
const { mutate } = useMutation(  
  [API.KEYS.UPDATE_CART_ITEMS],  
  async ({ items }) => {  
    await cartAPI.updateCartItems({ items });  
  },  
  {  
    onSuccess: () => {  
      queryClient.invalidateQueries(API.KEYS.GET_CART_ITEMS);  
    },  
    onError: (error) => {  
      alert(error.response.data.error.message);  
    },  
  }  
);
```

- 서버 데이터가 오래됐는지 파악하고 업데이트하자
- `QueryClient.invalidateQueries()`: 캐시된 데이터를 새로 fetch한 데이터로 변경 (refetch)
- CRD 작업 후 데이터를 다시 불러올 때 매우 편리

01 최종 과제 구현 모습

조금은 까다로웠던 컴포넌트- 상품 페이지에서 장바구니에 넣을 상품 옵션 선택하기

옵션 선택

구성 ^

01. 슬라이딩 지퍼백 크리스마스에디션 4종
10,000원

02. 슬라이딩 지퍼백 플라워에디션 5종
10,900원

고무장갑 베이지 S(소형) 6팩
9,900원

뽕아쓰는 키친타올 130매 12팩
16,900원

2겹 식빵수세미 6매

총 수량 2개

총 주문금액 20,900원

장바구니

구매하기

옵션 선택

구성 v

01. 슬라이딩 지퍼백 크리스마스에디션 4종 ×
- 4 + 40,000원

02. 슬라이딩 지퍼백 플라워에디션 5종 ×
- 1 + 10,900원

총 수량 5개

총 주문금액 50,900원

장바구니

구매하기

1. 옵션을 선택한 경우

- 해당 옵션이 cart에 있는지 확인
- 없다면 cart에 옵션의 수량을 1로 추가
- 있다면 구성 컴포넌트 닫기

2. 옵션의 개수를 변경하는 경우

- 옵션의 개수가 변하더라도 옵션의 순서는 바뀌지 않아야 함
- 옵션의 수량은 1 미만일 수 없음 ➡
handleMinusClick은 수량이 2 이상일 때만 실행
- 수량은 keyboard 입력으로 변경 가능 ➡ onChange
이벤트로 수량이 1 미만으로 변경되었다면, onBlur 이벤트를
통해 수량을 1로 변경

01 최종 과제 구현 모습

장바구니 페이지

쇼핑하기

로그아웃

기본에 슬라이딩 지퍼백 크리스마스/플라워에디션 에디션 외 주방용품 특가전

01. 슬라이딩 지퍼백 크리스마스에디션 4종

삭제

-

1

+

10,000원

02. 슬라이딩 지퍼백 플라워에디션 5종

삭제

-

1

+

10,900원

굳지않는 양금절편 1,050g 2팩 외 우리쌀떡 모음전

굳지않는 썩 양금 절편 1050g

삭제

-

1

+

15,900원

주문 예상금액

36,800원

2건 주문하기

쇼핑하기

로그아웃

기본에 슬라이딩 지퍼백 크리스마스/플라워에디션 에디션 외 주방용품 특가전

01. 슬라이딩 지퍼백 크리스마스에디션 4종

삭제

-

1

+

10,000원

02. 슬라이딩 지퍼백 플라워에디션 5종

삭제

-

1

+

10,900원

굳지않는 양금절편 1,050g 2팩 외 우리쌀떡 모음전

굳지않는 썩 양금 절편 1050g

삭제

-

1

+

15,900원

주문 예상금액

36,800원

2건 주문하기

01 최종 과제 구현 모습

주문하기 페이지

쇼핑하기

로그아웃

주문하기

배송지 정보

임시 이름
010-1234-5678
서울시 종로구

부재시 경비실에 맡겨주세요.

부재시 경비실에 맡겨주세요.

기본에 슬라이딩 지퍼백 크리스마스/플라워에디션 에디션 외 주방용품 특가전
[옵션] 옵션명: 01. 슬라이딩 지퍼백 크리스마스에디션 4종, 1개
10,000원

기본에 슬라이딩 지퍼백 크리스마스/플라워에디션 에디션 외 주방용품 특가전
[옵션] 옵션명: 02. 슬라이딩 지퍼백 플라워에디션 5종, 1개
10,900원

굳지않는 앙금절편 1,050g 2팩 외 우리쌀떡 모음전
[옵션] 옵션명: 굳지않는 썩 앙금 절편 1050g, 1개
15,900원

☒ 전체 동의하기

☒ 구매조건 확인 및 결제 진행 동의

☒ 개인정보 제3자 제공 동의

총 36,800원 결제하기



쇼핑하기

로그아웃

주문하기

배송지 정보

임시 이름
010-1234-5678
서울시 종로구

직접 입력

기본에 슬라이딩 지퍼백 크리스마스/플라워에디션 에디션 외 주방용품 특가전
[옵션] 옵션명: 01. 슬라이딩 지퍼백 크리스마스에디션 4종, 1개
10,000원

기본에 슬라이딩 지퍼백 크리스마스/플라워에디션 에디션 외 주방용품 특가전
[옵션] 옵션명: 02. 슬라이딩 지퍼백 플라워에디션 5종, 1개
10,900원

굳지않는 앙금절편 1,050g 2팩 외 우리쌀떡 모음전
[옵션] 옵션명: 굳지않는 썩 앙금 절편 1050g, 1개
15,900원

부재시 경비실에 맡겨주세요.

부재시 경비실에 맡겨주세요. 문앞도 괜찮습니다!!



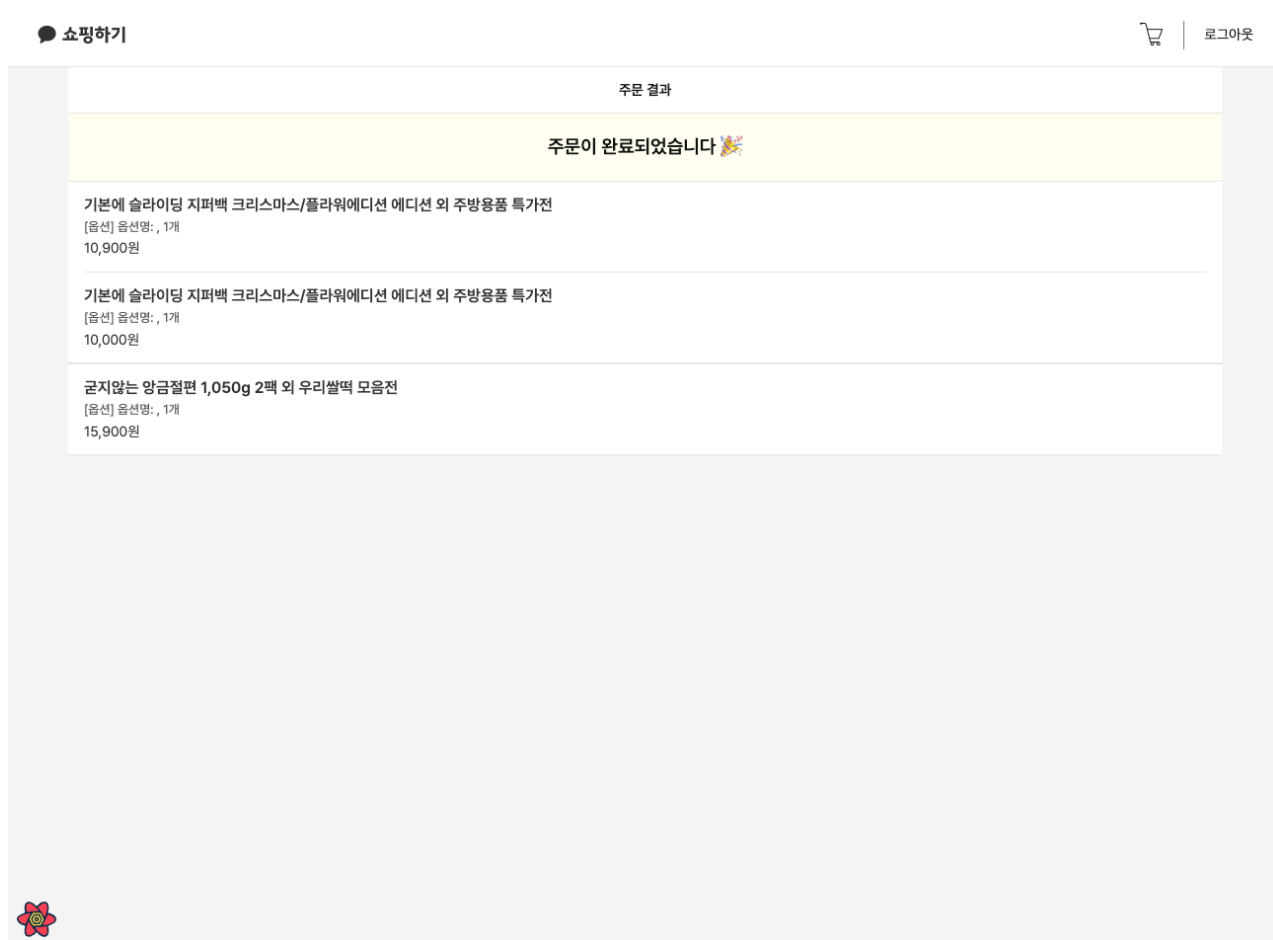
직접 입력

직접 입력을 선택하면 textarea가 빈칸입니다



01 최종 과제 구현 모습

주문 완료 페이지



01 최종 과제 구현 모습

Not Found 페이지

🗨 쇼핑하기  | [로그아웃](#)

Not Found

해당 페이지를 찾을 수 없습니다

[홈](#) [뒤로 가기](#)

🗨 쇼핑하기

 | [로그아웃](#)

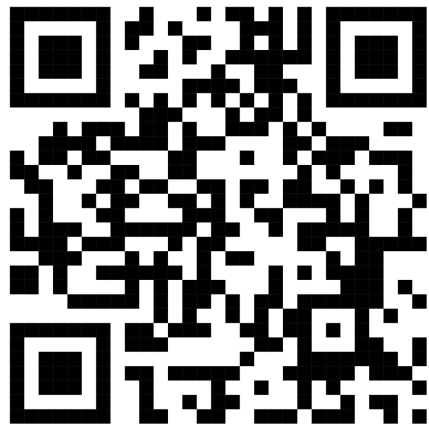
상품 아이디 에러

해당 상품 아이디가 존재하지 않습니다

[홈](#) [뒤로 가기](#)

- createBrowserRouter의 `errorElement`을 사용해 페이지별 에러 페이지 관리

[github repository](#)



[배포 URL](#)



- 평소에 사용해 보고 싶었지만 왠지 겁이 났던 storybook과 react query를 접할 수 있었다!
- 다양한 상태 관리 라이브러리를 사용하고 비교했다
- 현업 개발자로부터 얻는 꿀 코드 리뷰! 조언과 더불어 코드에 확신을 가졌다

- 기획 단계에서 공통 디자인을 정해서 컴포넌트 재사용성을 높이고 싶다
- storybook을 활용해 공통 디자인 컴포넌트를 문서화하고 싶다
- 이슈 및 브랜치 관리, 팀원간 코드리뷰를 통해 원활한 협업을 진행하고 싶다
- socket 통신이나 WebRTC를 활용해 다양한 서비스를 구현하고 싶다

End of Document