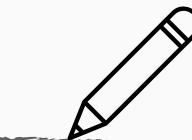




SNACK

여러 플랫폼에서 구매한 간식 내역을 한곳에 모아 관리할 수 있는

- **원스톱 간식 관리 서비스**



이태빈 이지수 김우주 김홍섭 장원빈 조성빈

CONTENTS



01 프로젝트 개요

- 서비스 및 컨셉 소개
- 활용 방안 및 기대 효과

05 프로젝트 수행 절차 및 방법

- 프로젝트 절차
- API 명세, ERD
- 프로젝트 아키텍쳐

02 시연

- 실시간 시연
- 서비스 배포 링크 첨부 (QR, URL)

06 트러블슈팅

- 백엔드
- 기획의도

03 심화기능

- 짐하기
- 토스페이먼츠 결제 API 연동
- 노드메일러
- node-cron
- OPEN AI

07 회고

- Keep
- Problem
- Try

04 팀 구성 및 역할, 협업 방식 소개

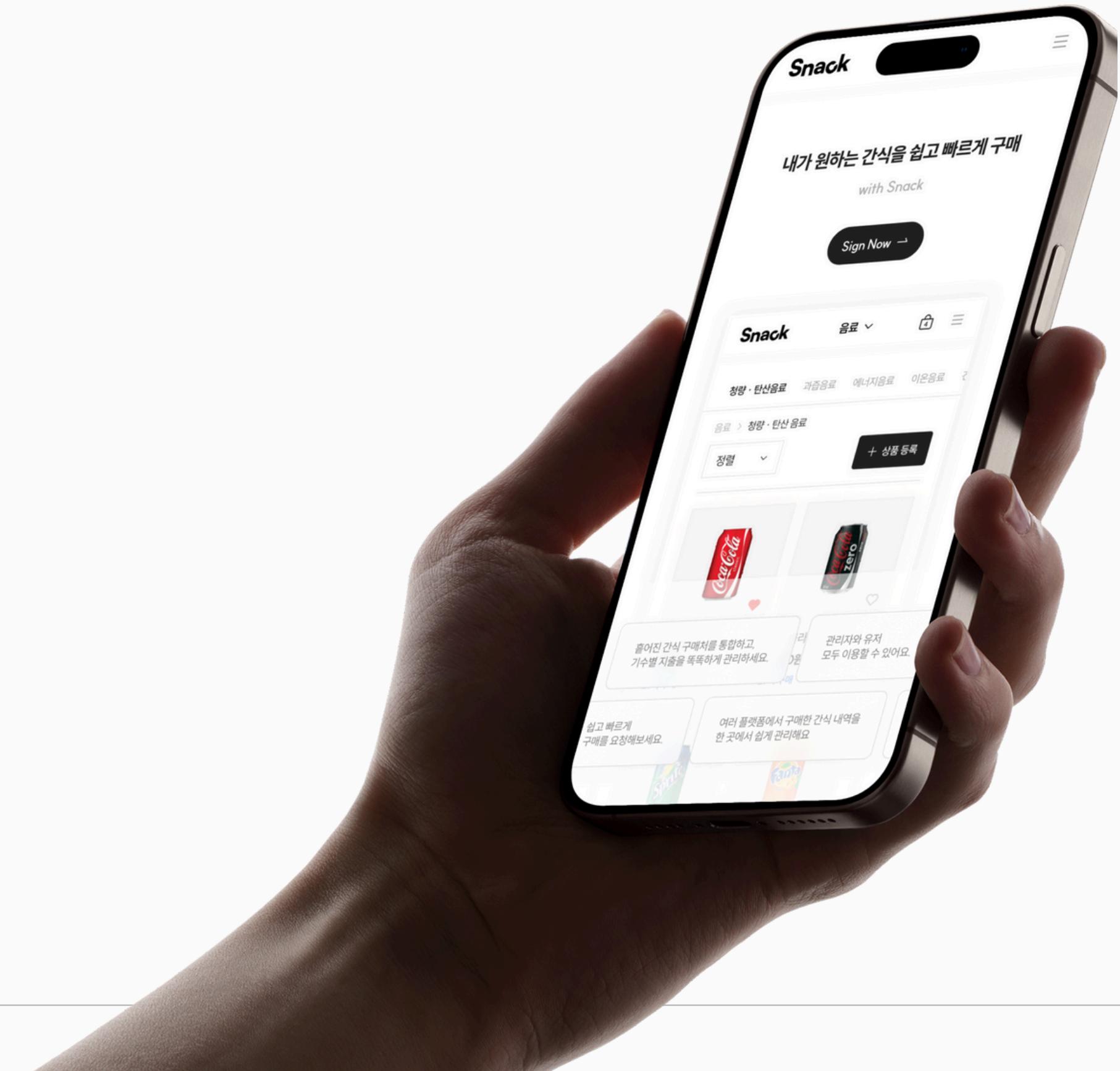
- 팀원 소개 & R&R
- 협업 방식

08 Q&A

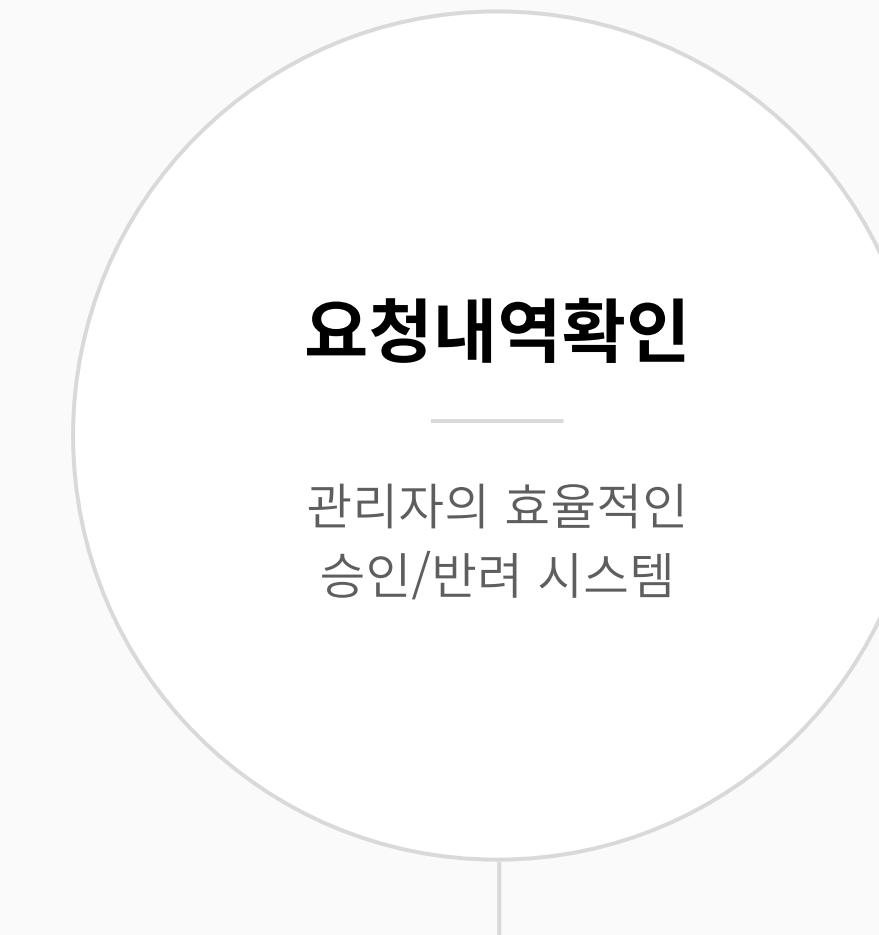
01

프로젝트 개요

- 서비스 및 컨셉 소개
- 활용 방안 및 기대 효과



• 서비스 소개



• 활용 방안 및 기대효과

분산된 구매 요청

여러 곳에서 진행되는 구매 관리



통합된 구매 요청 관리

언제 어디서나 한눈에 확인!

예상치 못한 비용 지출

어려운 비용 관리



불필요한 지출 비용 절감

직원들의 간편한 간식/비품 구매요청 시스템

번거로운 비용 처리

기하급수적으로 늘어나는 업무시간



단순한 비용 처리

비용 처리 업무시간 단축

02

시연

- 실시간 시연
- 서비스 배포 링크 첨부



• 시연

QR 코드



<https://5nack.site>

03

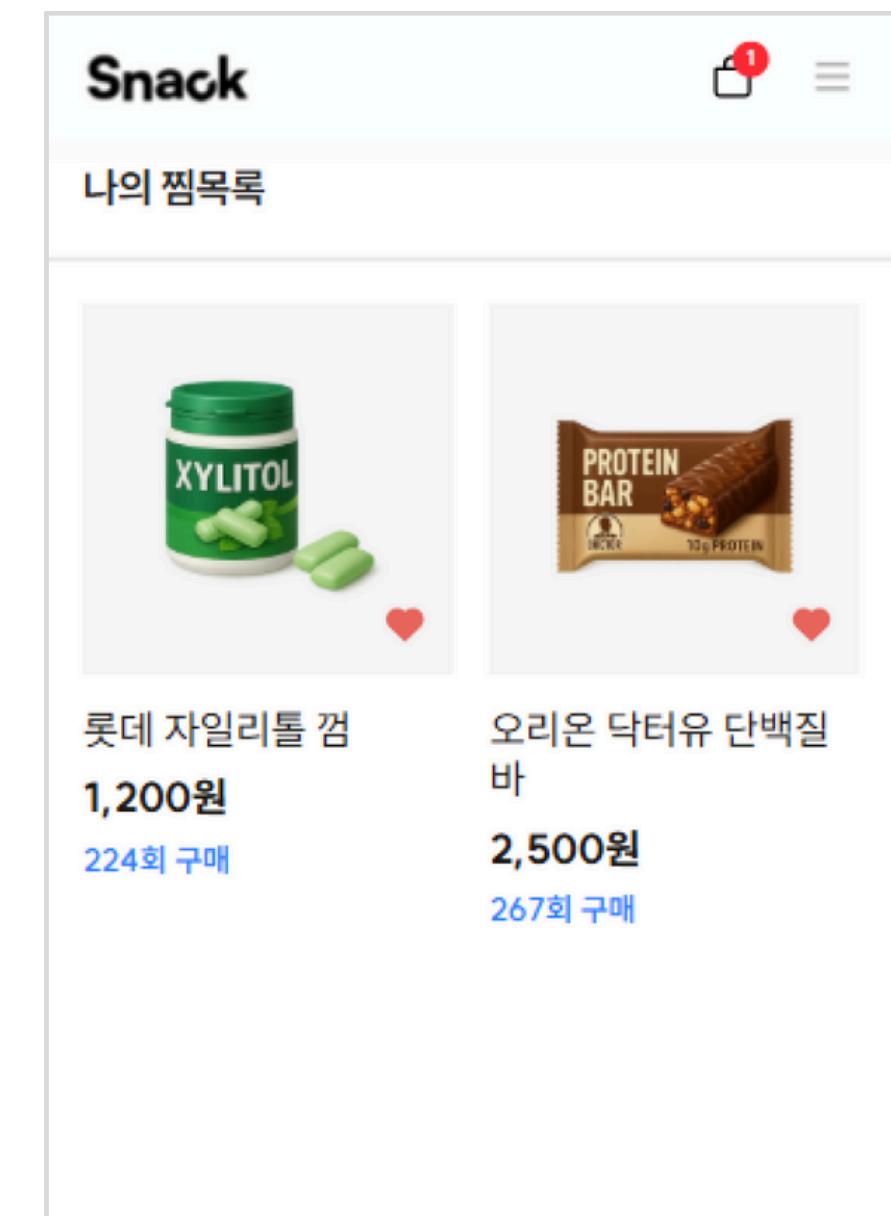
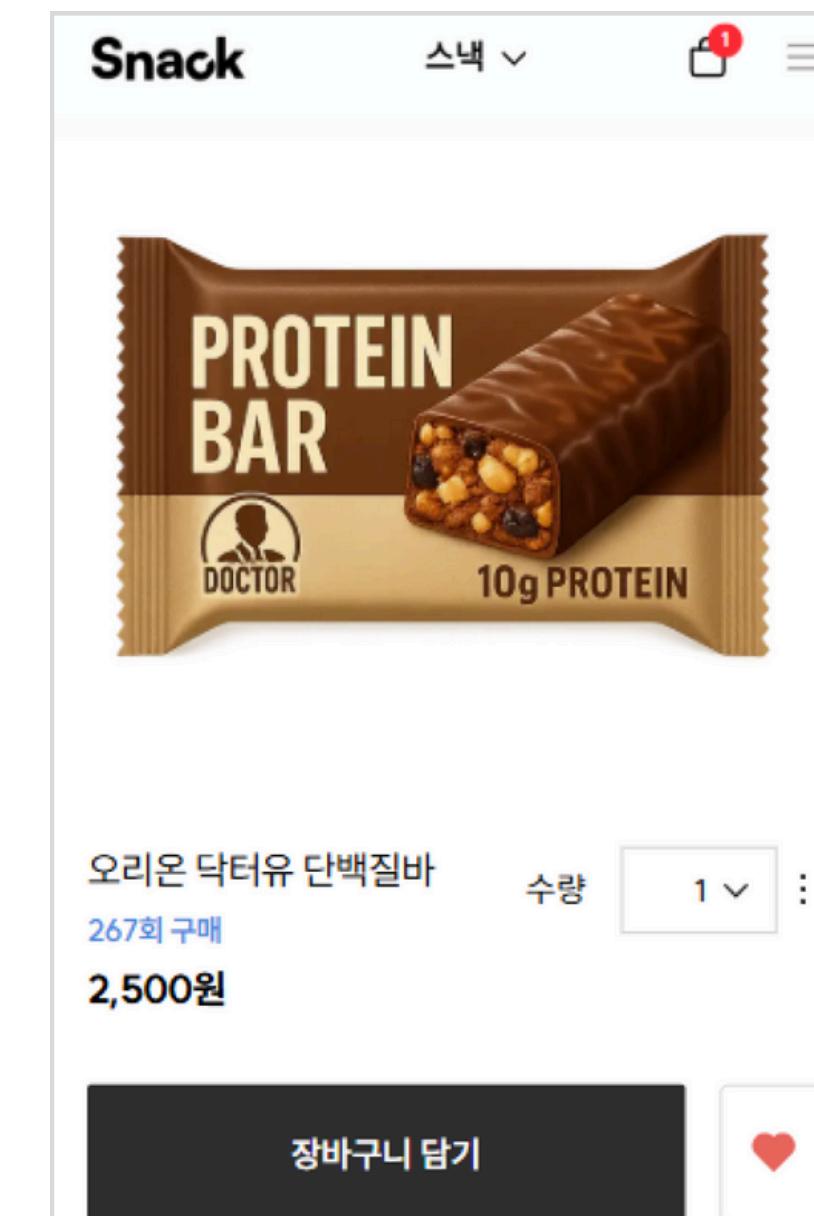
심화기능

- 짐하기
- 토스페이먼츠 결제 API 연동
- 노드메일러
- node-cron
- OPENAI



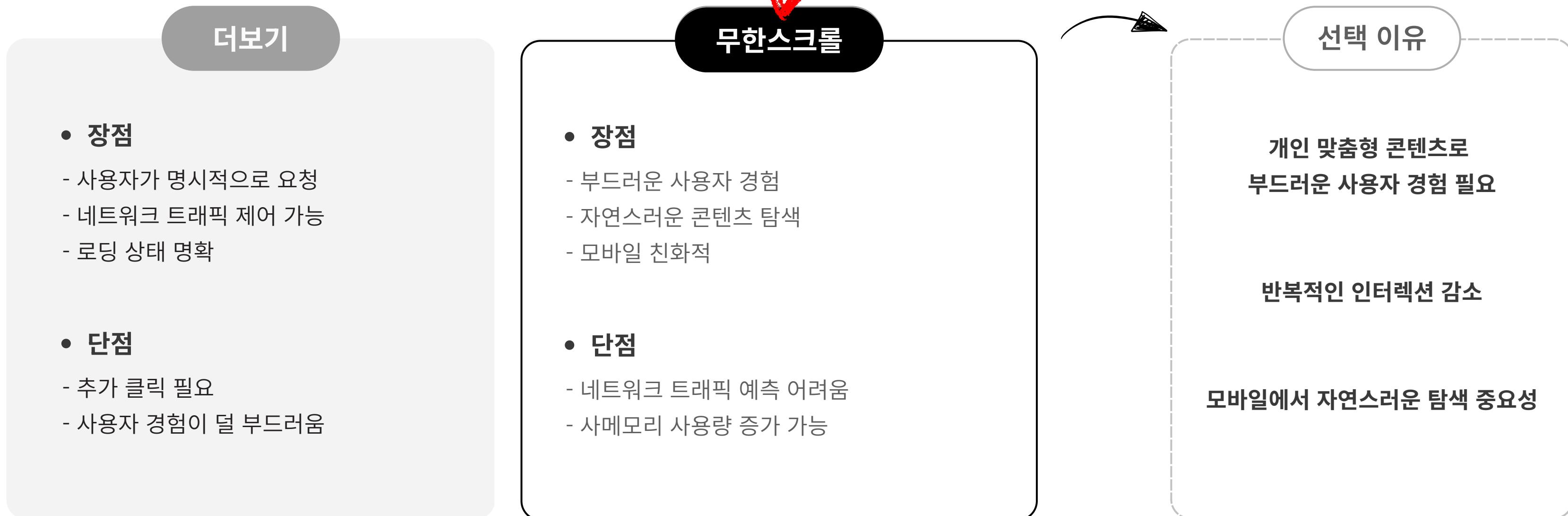
• 심화기능

상품 찜하기 + 내가 찜한 상품 목록 페이지



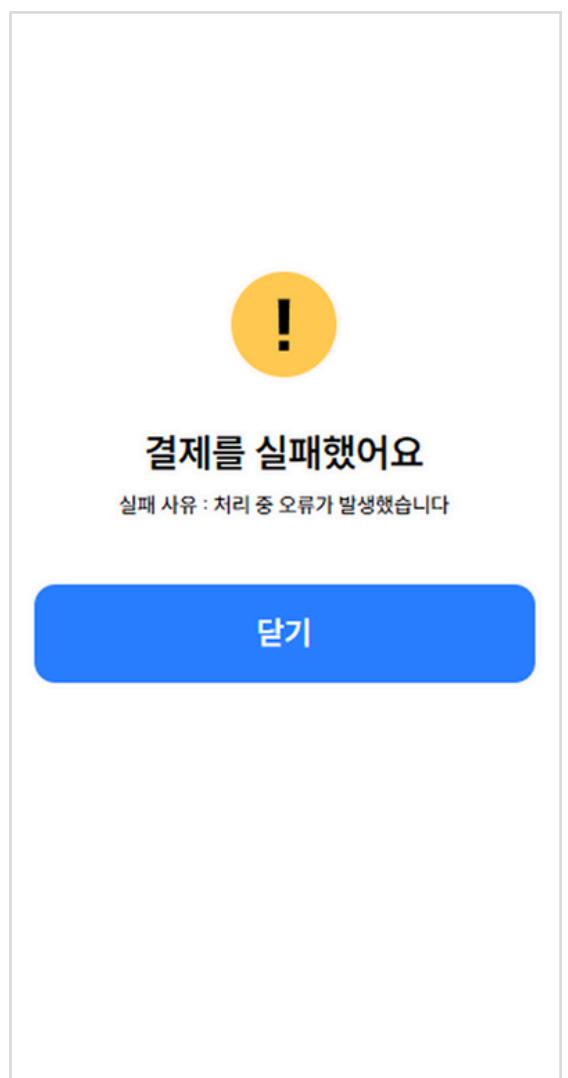
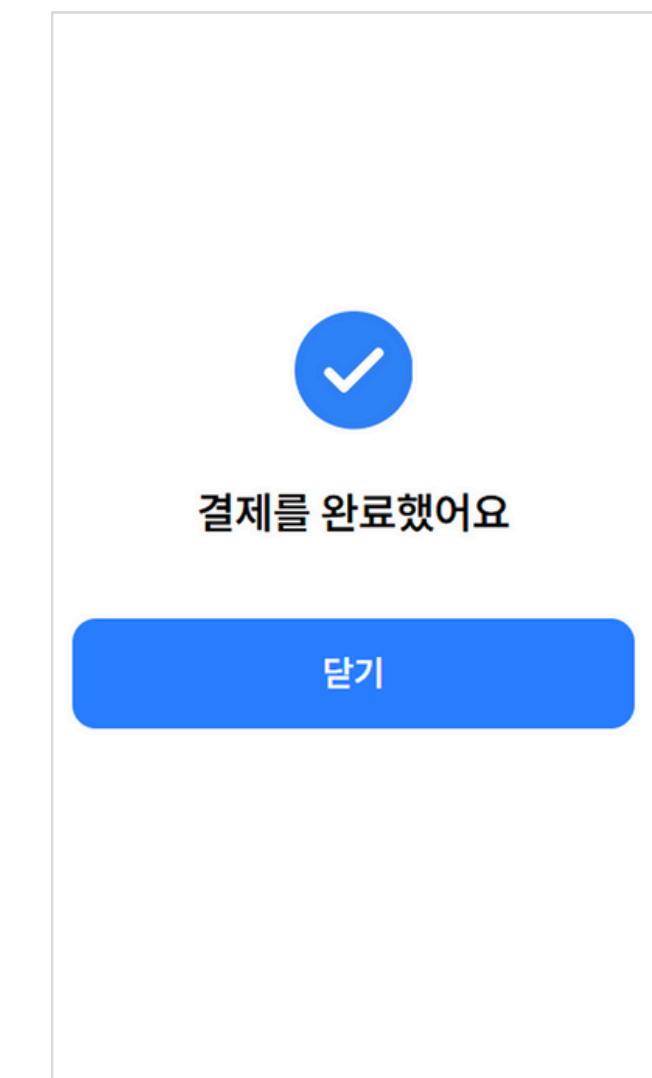
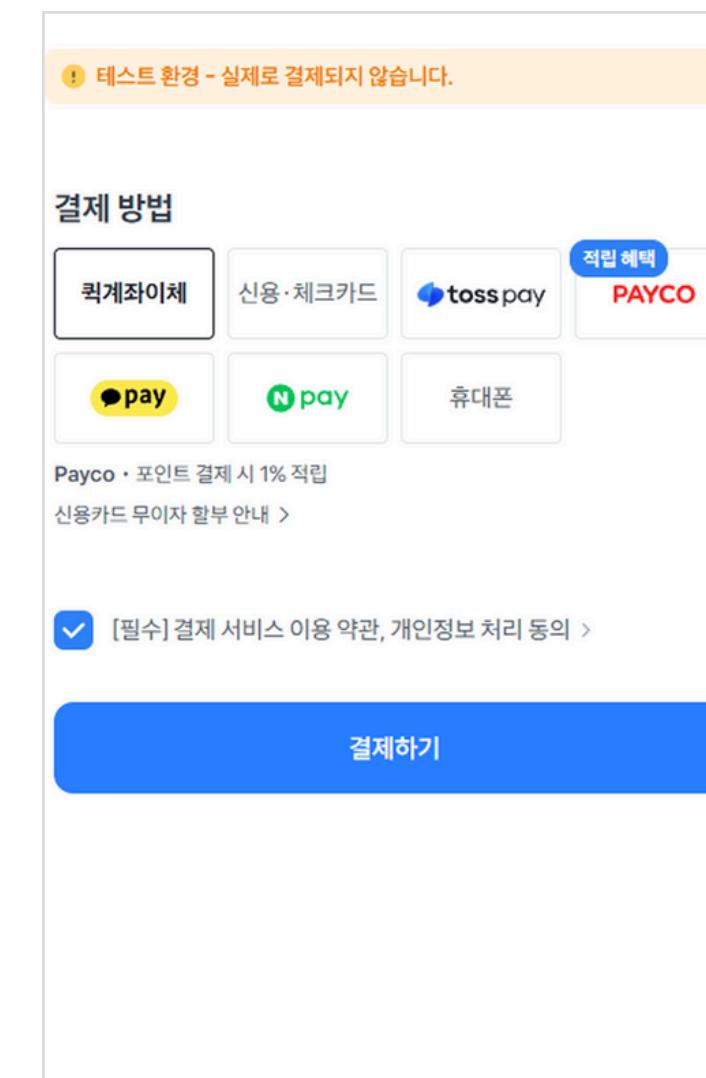
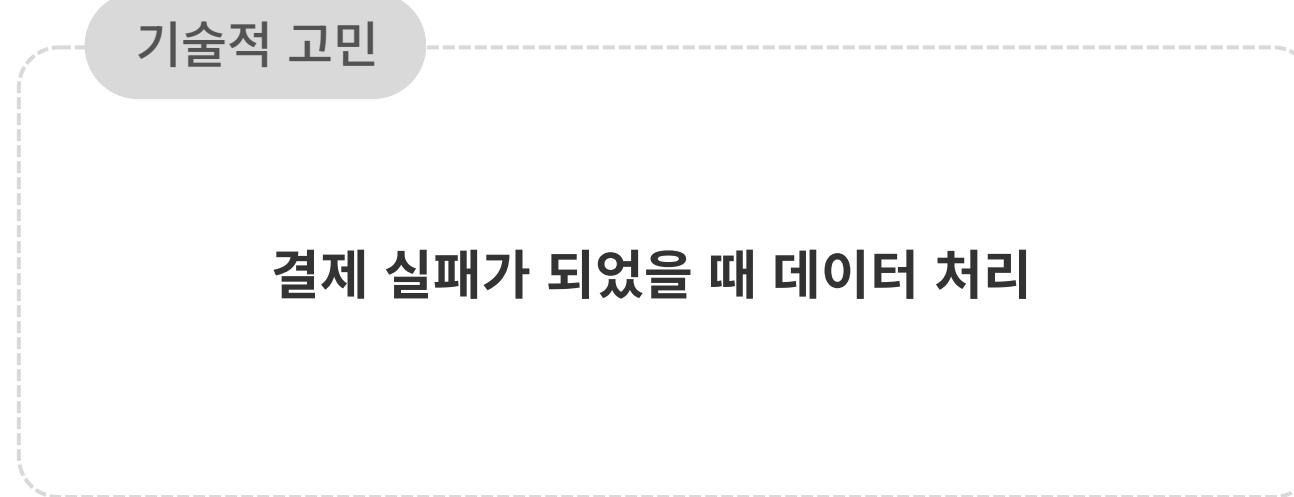
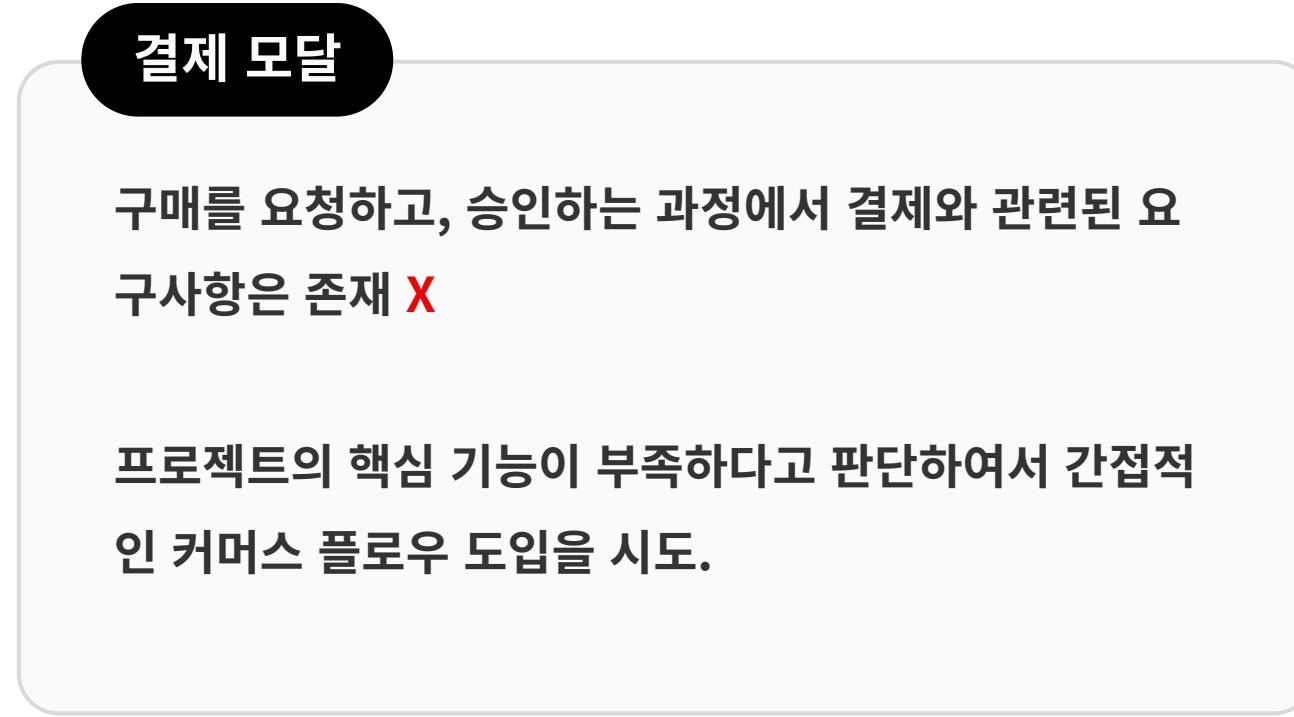
• 심화기능

기술적 고민 & 선택이유 : 무한 스크롤 방식 VS 더보기 버튼 방식



• 심화기능

토스 페이먼츠 결제 모달



결제 모달

성공 모달

실패 모달

• 심화기능

노드 메일러 회원 초대 기능

노드 메일러

- 이메일 초대 기반 회원가입
- 만료 링크로 안전한 가입 유도
- 초대를 통한 간편한 가입 경험

해결방안

트랜잭션 분리

- 초대 생성 후, 이메일 발송은 별도 TRY-CATCH로 분리 처리

실패 대응:

- 이메일 실패 시에도 초대 유지,
EMAILSENT: FALSE로 상태 전달

기술적 고민

이메일 실패 시 초대 생성 유지

VS

전체 롤백, 트랜잭션 처리 고민

Snack

회사 초대

안녕하세요, 테스트 관리자님!

회사 관리자가 귀하를 관리자로 초대했습니다.

초대 정보:

- 역할: 관리자
- 만료일: 2025년 8월 1일 오후 04:19

아래 버튼을 클릭하여 회원가입을 완료해주세요:

회원가입 완료하기

주의사항:

- 이 초대 링크는 2025년 8월 1일 오후 04:19까지 유효합니다.
- 링크를 클릭하면 비밀번호를 설정하여 회원가입이 완료됩니다.
- 본인이 요청하지 않은 초대라면 이 이메일을 무시하셔도 됩니다.

초대 메일 HTML

• 심화기능

매달 예산 자동 생성

NODE-CRON

NODE 환경에서 사용할 수 있는
스케줄러 라이브러리

기술적 고민

- 예산 자동 생성 시 초기 예산 값 고민

```
import cron from "node-cron";
import createMonthlyBudget from "./createMonthlyBudget";

const autoCreateMonthlyBudget = cron.schedule(
  "0 0 0 1 * *",
  async () => {
    try {
      await createMonthlyBudget();
    } catch (e) {
      if (e instanceof Error) {
        console.error("⚠️ MonthlyBudget Create Error");
        console.error(e.message);
      }
    }
  },
  { timezone: "Asia/Seoul" },
);

export default autoCreateMonthlyBudget;
```

NODE-CRON 표현식

"0 0 0 1 * *",

초 분 시 일 월曜일

매월 1일 실행

함수 실행

await createMonthlyBudget();

NODE-CRON 함수와
서비스 로직은 별도로 분리

• 심화기능

OPENAI API를 활용한 AI 챗봇 도우미

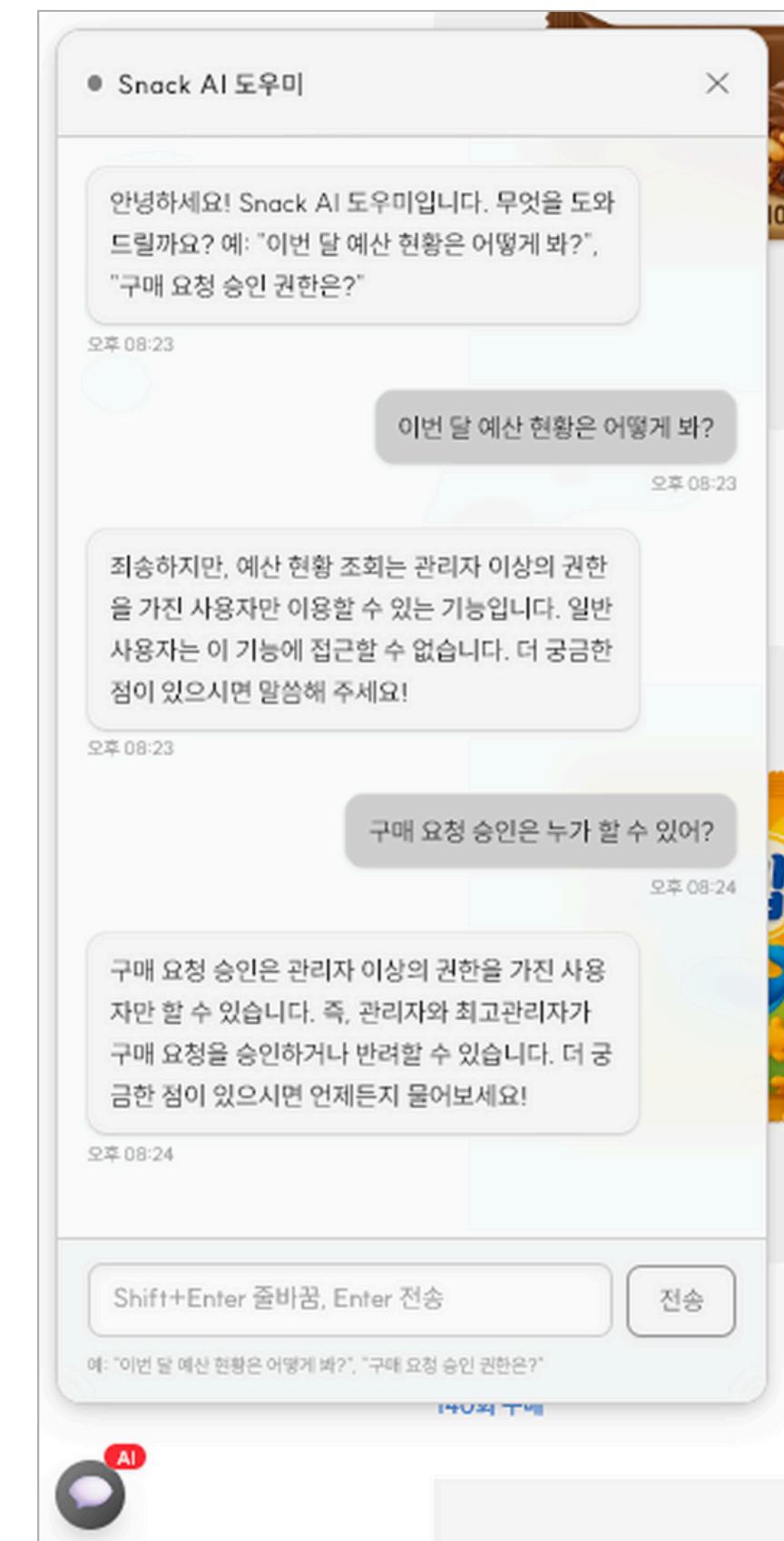
SNACK AI 도우미

SNACK 사용자를 위한 특별한 챗봇 서비스

SNACK 서비스를 사용하며 궁금한 점을 자유롭게 질문하세요!

고민했던 점

1. 챗봇 기능 구현을 위한 외부 API로 무엇을 선택해야 할까?
2. 챗봇에게 어떻게 SNACK 서비스만의 특별한 정보를 학습시킬 수 있을까?
3. 기존 기능에 영향을 미치지 않으면서 챗봇 기능을 구현할 수 있을까?



• 심화기능

OPENAI API를 활용한 AI 챗봇 도우미

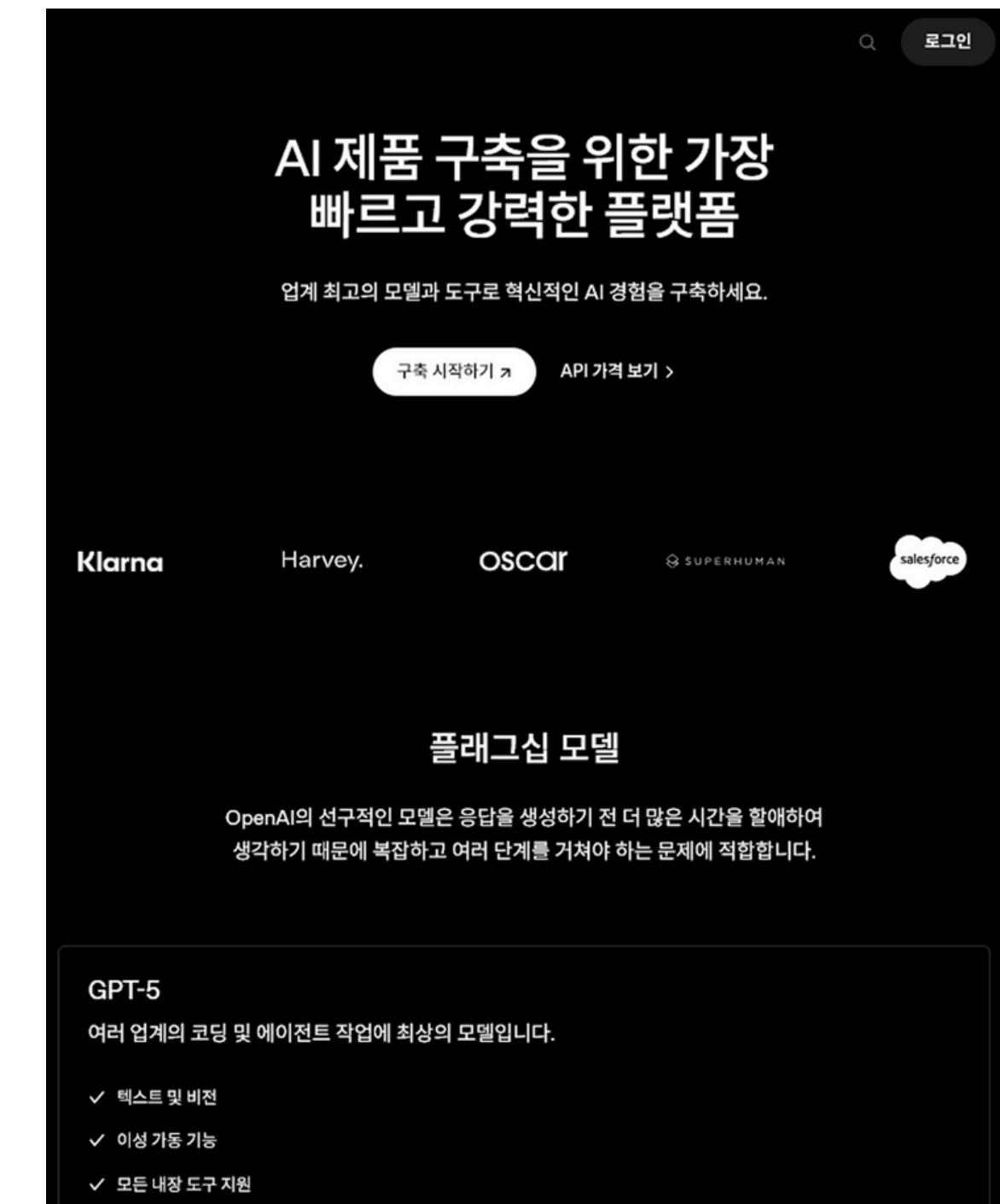
1. 챗봇 기능 구현을 위한 외부 API로 무엇을 선택해야 할까?

• 다양한 서비스 중 어떤 기준으로 API를 선택하셨나요?

무료 사용량을 제공하며 인지도가 높고 개발자 커뮤니티가 비교적 활성화된 OPENAI API 플랫폼을 선택했습니다

• 개발자 커뮤니티 활성화를 고려한 이유가 있나요?

프로젝트 진행 중 발생할 수 있는 기술적 문제를 빠르게 해결하고, 최신 노하우를 공유받아 프로젝트 완성도를 높이기 위해 개발자 커뮤니티 활성화를 중요하게 고려했습니다.



OPENAI API 플랫폼

• 심화기능

OPENAI API를 활용한 AI 챗봇 도우미

2. 챗봇에게 어떻게 SNACK 서비스만의 특별한 정보를 학습시킬 수 있을까?

- ✓ **프롬프트 엔지니어링을 통해 API 요청 시,
Snack의 고유한 데이터를 담은 추가 메시지를 함께 보내도록 구현**

```
// 프롬프트 엔지니어링: system 메시지 추가
const systemMessage: ChatMessage = {
  role: 'system',
  content: `
    당신은 '스낵', 'Snack' 서비스의 AI 어시스턴트입니다. 주요 목표는 간식 구매 내역 관리와 총무업무 지원입니다. 사용자 권한에 따라 기능을 제공합니다:
    - 일반 유저: 상품 조회/등록, 장바구니, 구매 요청/내역 관리, 정보 수정.
    - 관리자: 상품 수정/삭제, 예산/지출 조회, 구매 요청 승인/반려.
    - 최고 관리자: 회원 관리, 예산 설정, 기업 정보 수정.
    답변은 제공된 정보만 사용하며, 친절한 한국어로 작성하세요. 권한별 기능을 명확히 설명하고, 정보가 없을 경우 정중히 답변하세요.
  `,
};
const messagesWithSystem = [systemMessage, ...messages];
```

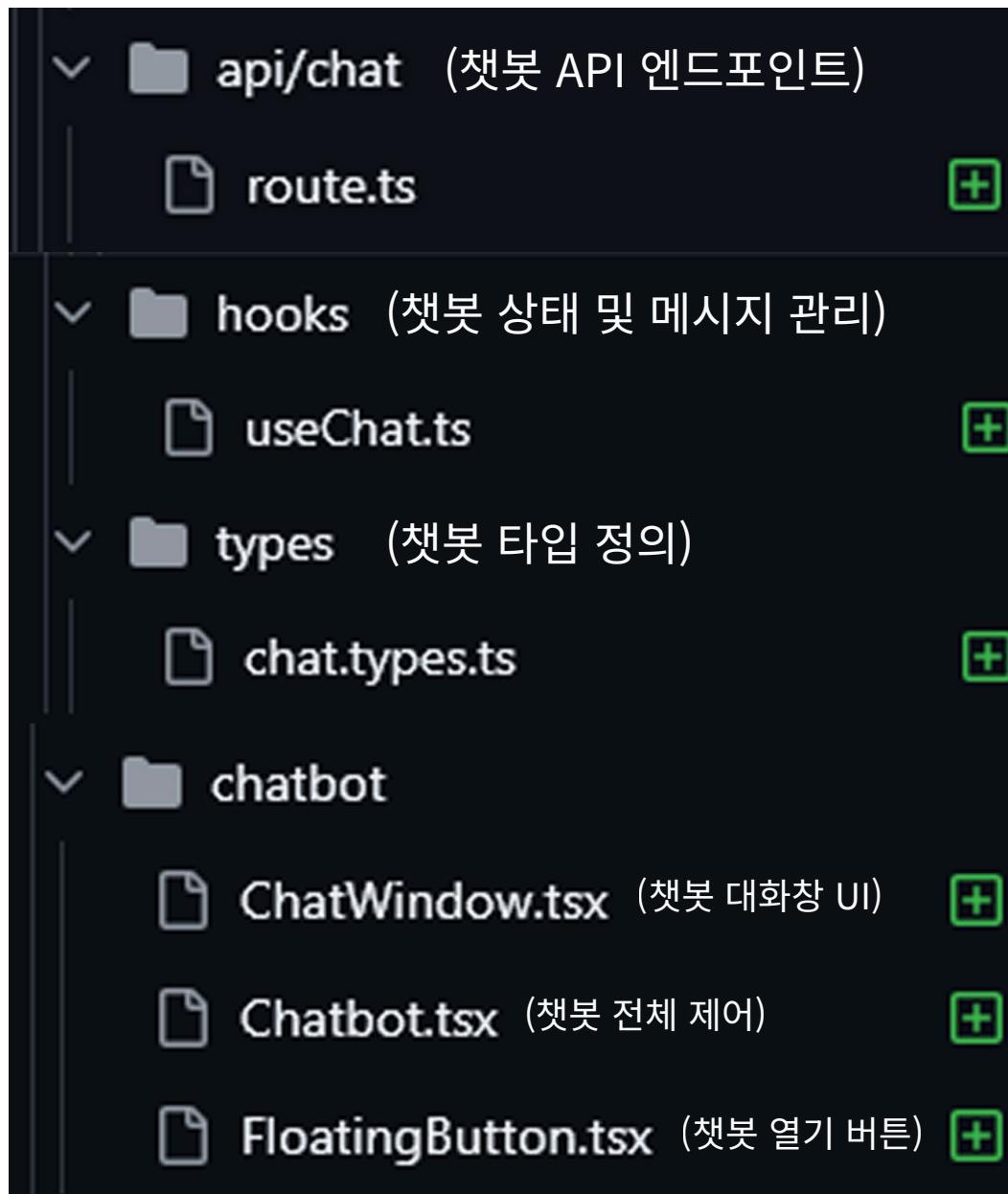
* 프롬프트
엔지니어링이란?

프롬프트 엔지니어링은 AI 모델에게 더 명확하고 구체적인 맥락을 제공하여, 원하는 결과를 얻기 위해 질문이나 지시에 추가 정보를 더하는 기술입니다.

• 심화기능

OPENAI API를 활용한 AI 챗봇 도우미

3. 기존 기능에 영향을 미치지 않으면서 챗봇 기능을 구현할 수 있을까?



기존 기능에 영향을 미치지 않으려면 결합도는 Down, 응집도는 Up

결합도 COUPLING

결합도는 모듈 간의 의존성, 즉 서로 얼마나 긴밀하게 연결되어 있는지를 나타냅니다.

- UI, 로직, API를 독립적인 모듈로 분리하고, HOOK과 API 호출 같은 표준화된 인터페이스를 통해 소통하여 직접적인 코드 의존성을 최소화했습니다

응집도 COHESION

응집도는 하나의 모듈이 얼마나 단일한 목적을 가지고 있는지를 나타냅니다.

- 각 파일이 UI, 로직, API 처리 등 단일 목적에만 집중하고, 챗봇 로직은 USECHAT.TS 파일에 응집시켜 가독성과 유지보수성을 높였습니다.

04

팀 구성 및 역할, 협업 방식 소개

- 팀원 소개 & R&R
- 협업 방식



• 팀원 소개 (R&R)



이태빈
팀장

[비개발]

- 프론트엔드 디자인 총괄
- AWS IAM 관리
- DB 스키마 관리/담당

[개발]

API

- 구매 관리
- 예산 관리
- 좋아요
- 결제

페이지

- 로그인 / 상품 등록 내역 / 장바구니 / 장바구니 주문 / 찜 목록 / 권한 에러 / 결제 모듈

공통 컴포넌트

- 내가 등록한 상품 리스트
- 구매 요청 리스트
- 주문 승인 / 반려 모달
- 카테고리 메뉴



이지수
부팀장

- 팀 노션 문서 관리·담당

API

- 회원 관리 API
- 프로필 관리 API

페이지

- 상품 리스트 페이지
- 구매 요청 관리 상세 페이지
- 회원가입 페이지 (일반 & 관리 자용)

공통 컴포넌트

- 헤더
- 탈퇴 알림 모달
- 메뉴
- 상품 상세정보



조성빈
팀원

- API & 컴포넌트 중점 개발

API

- 장바구니

페이지

- 상품 상세
- 내 구매 요청 내역
- 랜딩

공통 컴포넌트

- text area
- 회원관리 리스트
- dropdown
- toast

• 팀원 소개 (R&R)

[비개발]



장원빈
팀원

- AWS RDS 관리·담당



김홍섭
팀원

- AWS EC2 관리·담당



김우주
팀원

- API & 컴포넌트 중점 개발

[개발]

API

- 구매 요청 API

페이지

- 구매 내역 상세 페이지
- 구매 요청 완료 페이지
- 내 구매 요청 내역 상세 페이지

공통 컴포넌트

- 회원초대 모달
- 페이지네이션
- 서브 카테고리 메뉴
- 사이드 메뉴

API

- 인증/인가 API
- 회원가입 이메일 전송 API

페이지

- 최고 관리자 회원가입 페이지
- 예산 관리 페이지
- 구매 내역 확인 페이지

공통 컴포넌트

- Input
- product list
- 상품 수정/등록 모달
- 삭제/구매 요청 취소 alert 모달

API

- 상품 관련 API
- 카테고리 API

페이지

- 회원 관리 페이지
- 구매 요청 관리 페이지
- 프로필 페이지

공통 컴포넌트

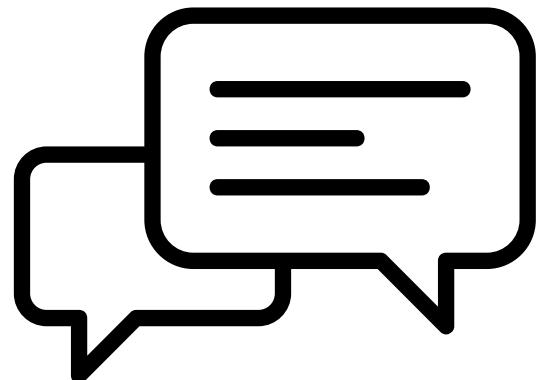
- 버튼
- 카드
- 검색
- badge

• 협업 방식



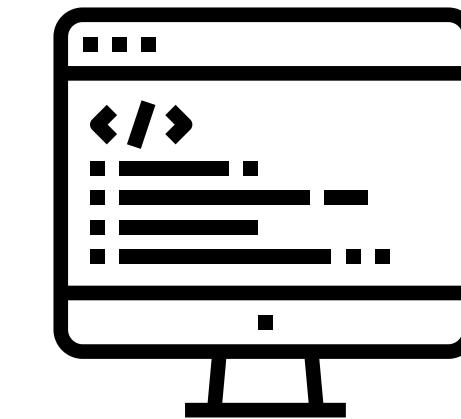
다양한 시각의 피드백

CODE REVIEW



원활한 커뮤니케이션

GITHUB DISCUSSIONS



향상된 코드 퀄리티

GITHUB ISSUES

• 협업 방식



다양한 시각의 피드백 CODE REVIEW

- 다양한 시각의 코드 리뷰
- 리팩토링 포인트 사전 발견
- 기술적인 토론을 통한 성장
- 코드 컨벤션 통일성 증가

Author ▾	Label ▾	Projects ▾	Milestones ▾	Reviews ▾	Assignee ▾	Sort ▾
Route					27	
발 확인 [김홍섭]	Chore Feature Refactor Route				25	
삭제 추가 [김우주]	Feature Refactor				21	
]	Feature				15	
					14	

• 협업 방식



원활한 커뮤니케이션
GITHUB DISCUSSIONS

- 개발 중 새롭게 알게 된 정보 및 기술 자유롭게 공유
- 특정 기술 선택에 대한 의견 교환 및 논의
- 자연스러운 커뮤니케이션으로 협업 효율 증대

The screenshot shows a list of discussions from a GitHub repository. Each post includes a timestamp, a user icon, a title, a category (e.g., Feature, Chore), and a reply count.

- [라이브러리] Zustand 도입 제안 및 장단점 (Feature) by rakaso598, started 2 weeks ago in 라이브러리. 0 replies.
- React에서 모달을 간편하게! ModalProvider 사용법 정리 [이태빈] (Feature) by De-cal, started 3 weeks ago in 일반. 1 reply.
- 코드 리뷰어 운영 방식 변경 안내 [이태빈] (Chore) by De-cal, announced last month in 변경사항. 1 reply.
- [트러블슈팅] SVG 관련 트러블슈팅 및 가이드라인 (Chore) by rakaso598, started 3 weeks ago in 트러블슈팅. 0 replies.
- [웹 표준 API] IntersectionObserver를 활용한 무한스크롤 팀 (Feature) by rakaso598, started 3 weeks ago in 일반. 1 reply.
- 스낵 시스템에 대한 질문 ② — 상품 등록 시 가격 필드의 목적 [이태빈] (Chore) by De-cal, asked 3 weeks ago in 질의응답 · Unanswered. 1 reply.

• 협업 방식



향상된 코드 퀄리티

GITHUB ISSUES

- QA 과정 중 발견한 이슈를 기록
- 기능 누락, 예외 처리, UI 개선 등
사소한 문제도 빠짐없이 공유
- 실시간 피드백을 통해 수정 속도 및
정확도 향상
- 완성도와 안정성 크게 향상

<input type="checkbox"/>	Open 0	Closed 32
<input type="checkbox"/>	<input checked="" type="checkbox"/> [QA] 토스 페이먼츠 에러 [이지수 -> 이태빈] #239 · by Jam1eL1 was closed 13 hours ago	
<input type="checkbox"/>	<input checked="" type="checkbox"/> [QA] 상품 등록 내역 페이지 [김우주 -> 이태빈] UI #233 · by wooju01 was closed last week	
<input type="checkbox"/>	<input checked="" type="checkbox"/> [QA] 내 구매 요청 내역 페이지 [김우주 -> 조성빈] Bug UI #232 · by wooju01 was closed 9 minutes ago	
<input type="checkbox"/>	<input checked="" type="checkbox"/> [QA] 구매요청내역 상세 페이지 [김홍섭 -> 이지수] QA UI #231 · by rakaso598 was closed 10 minutes ago	
<input type="checkbox"/>	<input checked="" type="checkbox"/> [QA] 회원가입 페이지(일반, 관리자) [김우주 -> 이지수] QA #218 · by wooju01 was closed last week	
<input type="checkbox"/>	<input checked="" type="checkbox"/> [QA] 상품 리스트 페이지 [이태빈 -> 이지수] QA #217 · by De-cal was closed last week	
<input type="checkbox"/>	<input checked="" type="checkbox"/> [QA] 상품 등록 모달 [이태빈 -> 이지수] QA #216 · by De-cal was closed last week	

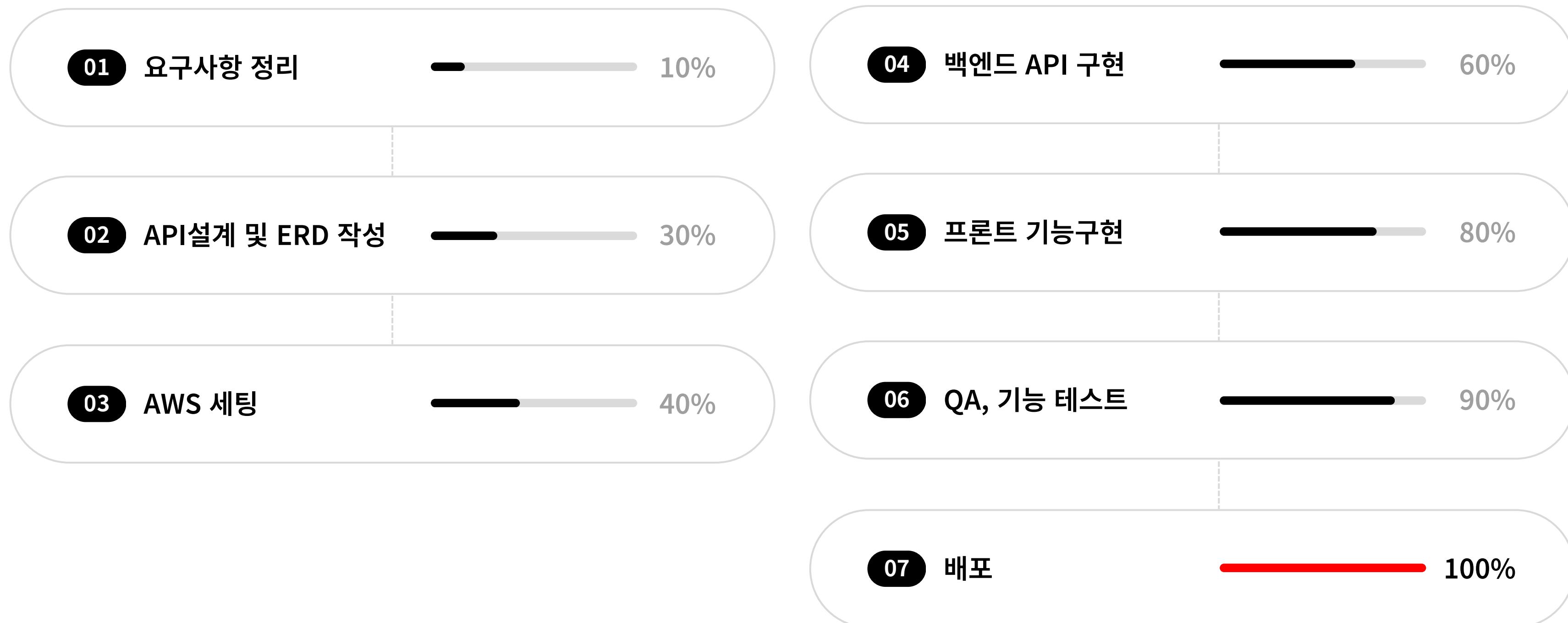
05

프로젝트 수행 절차 및 방법

- 프로젝트 절차
- API 명세, ERD
- 프로젝트 아키텍처
- 최적화

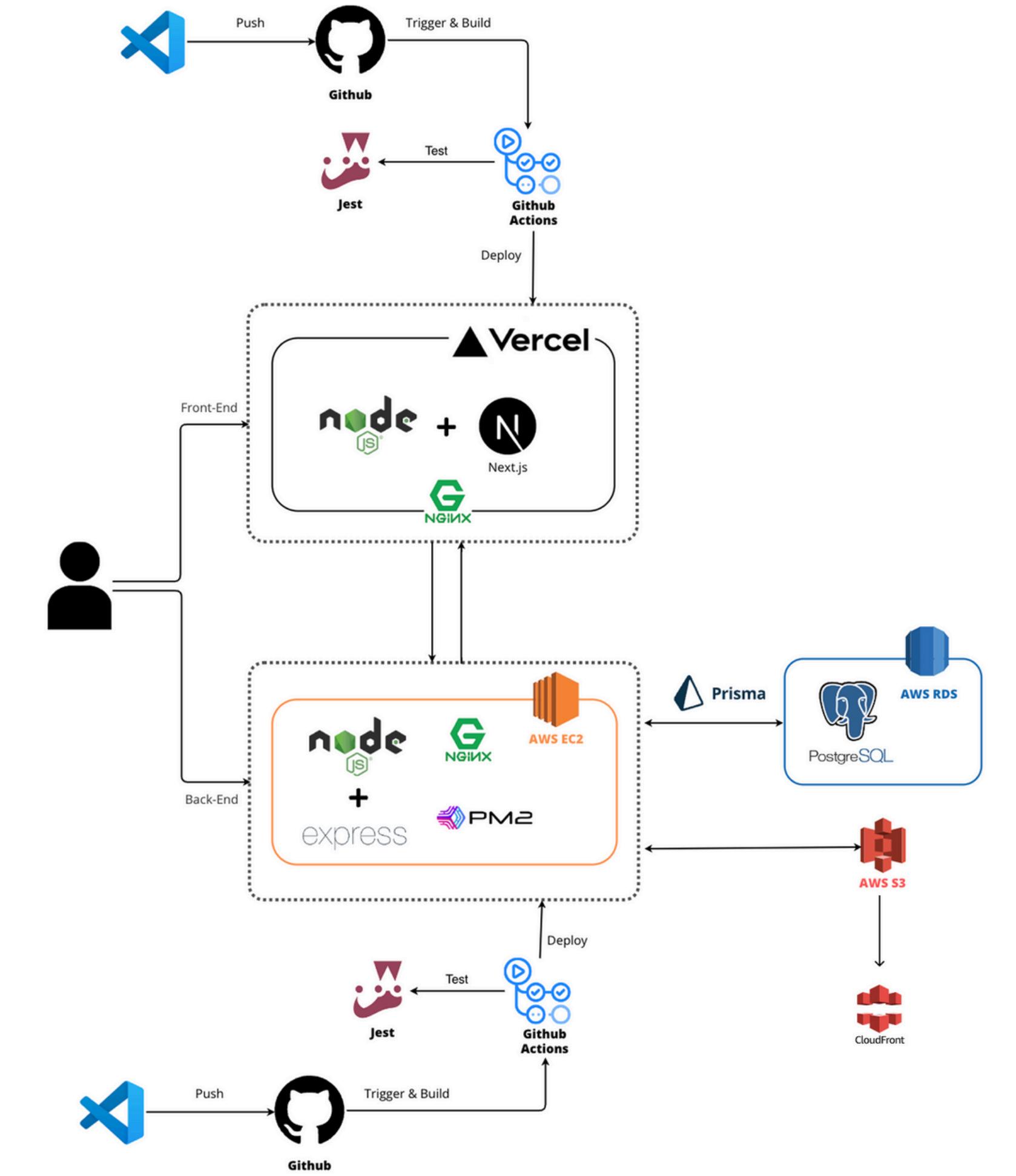
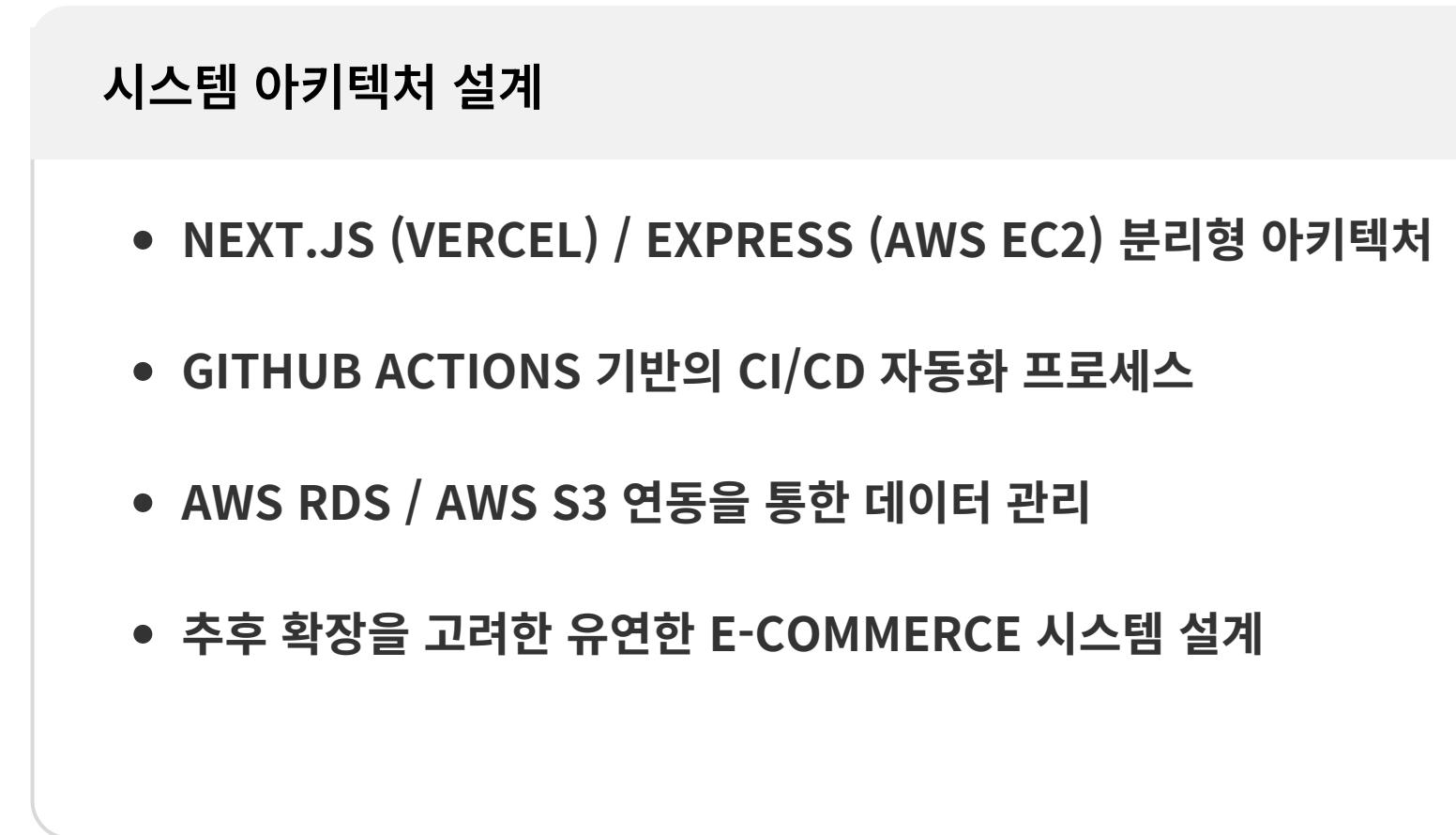


• 프로젝트 수행 절차 및 방법



• 프로젝트 수행 절차 및 방법

시스템 아키텍처 다이어그램

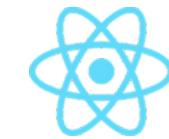


• FE 기술 스택

프레임워크 & 언어



NEXT.JS



REACT



TYPESCRIPT

상태 & 데이터 관리



REACT-QUERY



ZUSTAND



ZOD



REACT HOOK FORM

개발 도구



ESLINT



PRETTYER



SENTRY

UI & 스타일링



TAILWIND CSS



TAILWIND-MERGE

유틸리티



CLSX
LODASH.DEBOUNCE
DATE-FNS
INTERSECTION-OBSERVER



OPENAI



TOSSPAYMENTS-
SDK

배포 & 협업 도구



VERCEL



GITHUB



FIGMA



NOTION

• BE 기술 스택

프레임워크 & 언어



TYPESCRIPT



NODE.JS



EXPRESS.JS

인프라 & 유틸리티

보안 & 인증



MORGAN
HELMET
BCRYPT



REACT

테스트 & 품질 관리



JEST



SENTRY



SWAGGER



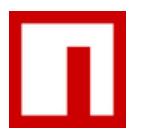
AXIOS



CLOUD FRONT



S3



NODE-CRON
NODEMAILER



DOCKER

DB & 캐시



PRISMA



POSTGRESQL

배포



NEXT.JS



ROUTE53



RDS



EC2

협업 도구



GITHUB



NOTION

• 프로젝트 수행 절차 및 방법

API 명세서

API 명세서 제작

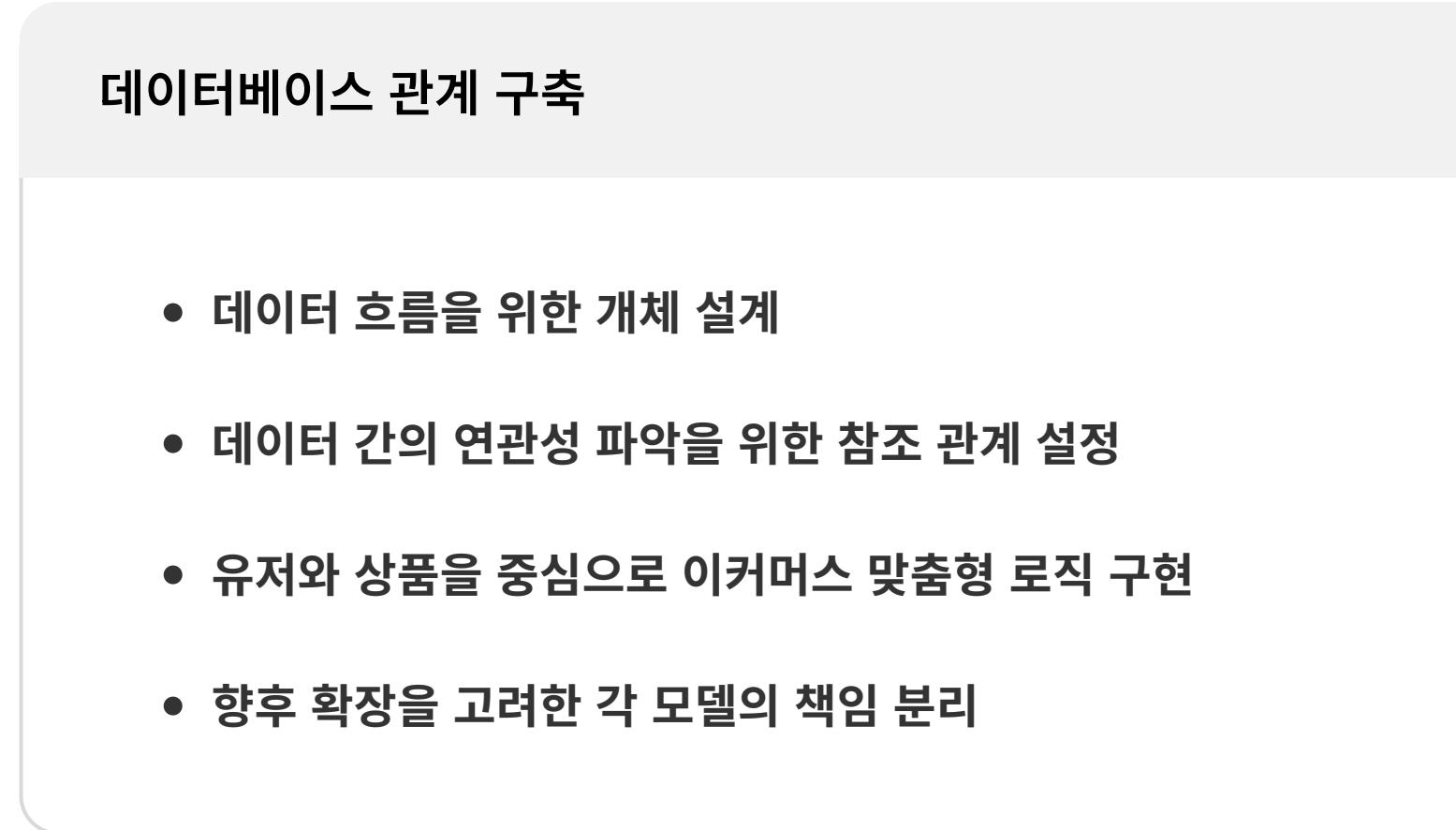
- SWAGGER를 활용하여 프론트엔드와 백엔드 간 주고받을 데이터 형식 자동 문서화
- HTTP 메서드를 기반으로 41개의 API 소통 규칙 정의
- 기능별 경로를 구분해 명확한 데이터 흐름 확보 및 유지보수 용이성 향상

The screenshot shows the Snack API documentation interface. At the top, it displays "Snack API 1.0.0 OAS 3.0" and a message: "원스톱 간식 구매 솔루션 Snack API 문서에 오신 걸 환영합니다." Below this is a "Servers" dropdown set to "http://localhost:8080 - Development server" and an "Authorize" button. The main content area is organized into sections: "Auth" (인증/인가 API), "Budget" (예산 API), and "Order" (구매 API) and "Product" (상품 API). Each section lists its available endpoints with their methods, paths, descriptions, and security requirements (indicated by lock icons).

Category	Method	Path	Description	Security
Auth	POST	/auth/signup	회원가입(최고관리자)	
	POST	/auth/signup/{inviteId}	회원가입(초대)	
	POST	/auth/login	로그인	
	POST	/auth/refresh-token	Access Token 재발급	
	POST	/auth/logout	로그아웃	🔒
Budget	GET	/admin/{companyId}/budgets	예산 및 지출 조회	🔒
	PATCH	/super-admin/{companyId}/budgets	예산 수정	🔒
Order	GET	/admin/orders	구매 내역 조회 (대기 or 승인)	🔒
	GET	/admin/orders/{orderId}	구매 내역 상세 조회 (대기 or 승인)	🔒
	PATCH	/admin/orders/{orderId}	구매 승인/반려	🔒
	POST	/orders	구매 요청 생성	🔒
	GET	/orders	내 구매 요청 조회	🔒
	GET	/orders/{orderId}	구매 요청 상세 조회	🔒
	PATCH	/orders/{orderId}	구매 요청 취소	🔒
Product	POST	/product	상품 등록	🔒
	GET	/products	상품 조회	🔒
	GET	/products/my	내가 등록한 상품 조회	🔒
	GET	/products/{id}	상품 상세 조회	🔒
	DELETE	/products/{id}	상품 삭제	🔒

• 프로젝트 수행 절차 및 방법

ERD (개체 관계 다이어그램)



User

```
id      String @id uuid()
email   String
name    String
password String
createdAt DateTime now()
updatedAt DateTime updatedAt()
deletedAt DateTime?
hashedRefreshToken String?
role    Role
```

Product

```
id      Int @id autoincrement()
name   String
price  Int
imageUrl String
linkUrl String
cumulativeSales Int 0
createdAt DateTime now()
updatedAt DateTime updatedAt()
deletedAt DateTime?
```

Order

```
id      String @id uuid()
approver String?
adminMessage String?
requestMessage String?
deliveryFee Int
productsPriceTotal Int
createdAt DateTime now()
updatedAt DateTime updatedAt()
status   OrderStatus
```

Receipt

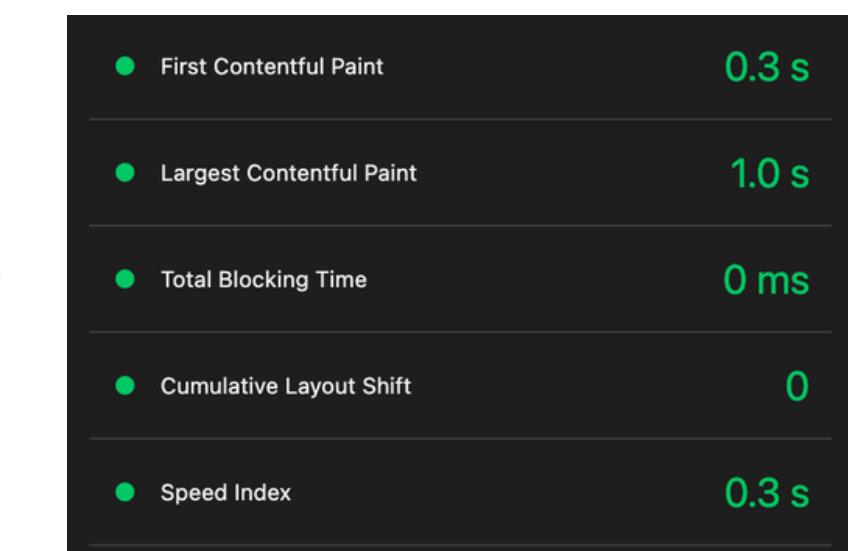
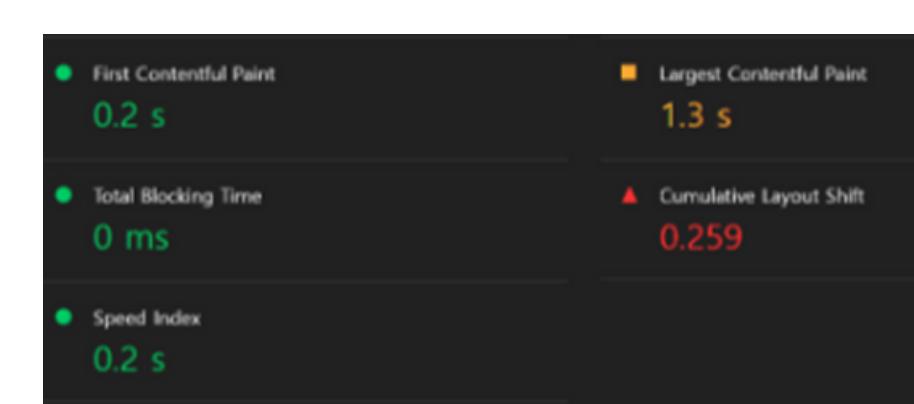
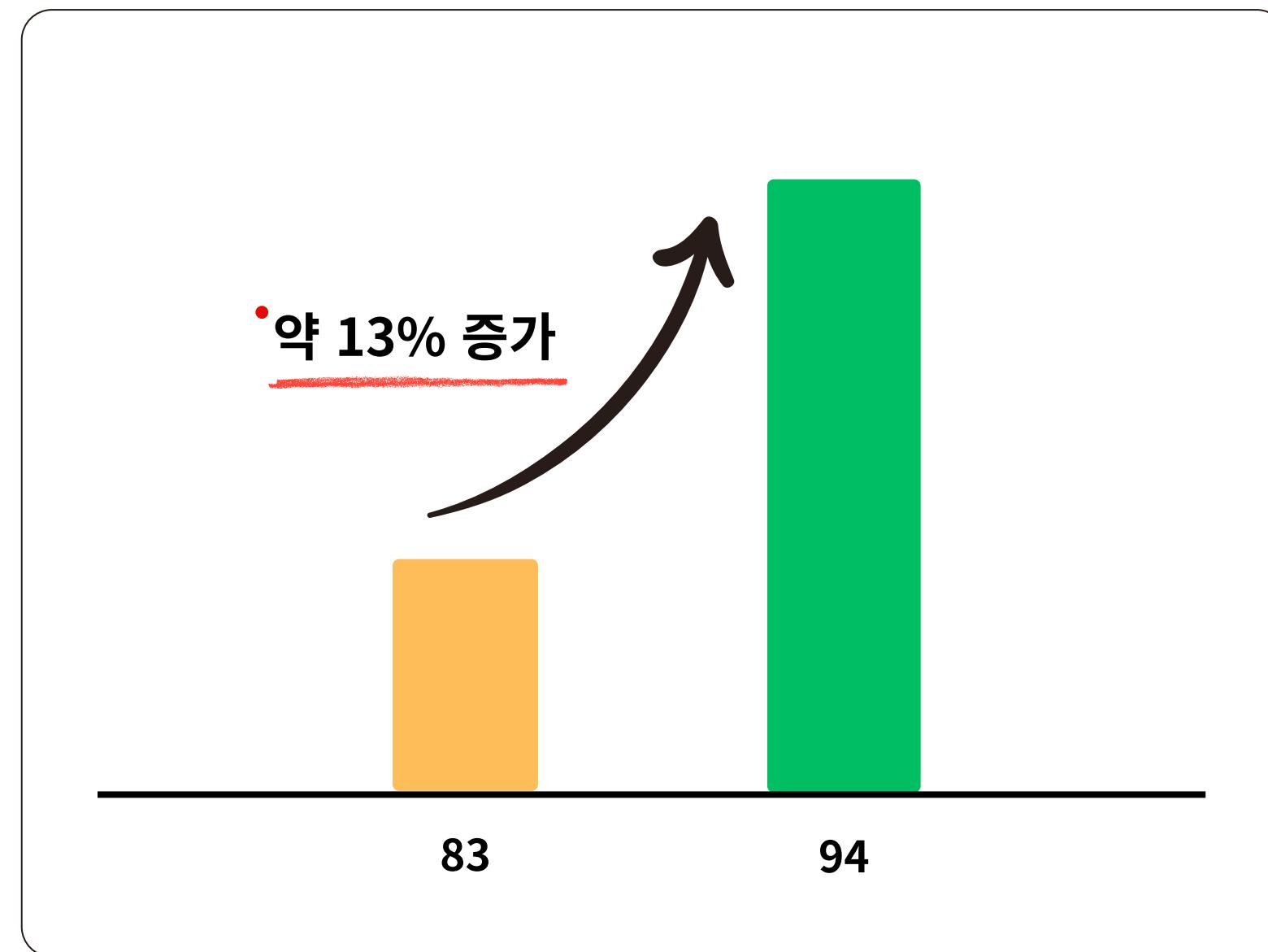
```
id      Int @id autoincrement()
productName String
price  Int
imageUrl String
quantity Int
createdAt DateTime now()
```

• 프로젝트 수행 절차 및 방법

최적화

LIGHTHOUSE

- 성능 진단: GOOGLE LIGHTHOUSE를 활용해 페이지 로딩·렌더링 지표 분석
- 개선 포인트
 - 이미지 및 컴포넌트 LAZY LOADING 적용
 - 부분 컴포넌트 분리로 리렌더 범위 축소
 - 서버/클라이언트 분리로 번들 최소화



06

트러블슈팅

- 백엔드
- 기획 의도



• 트러블슈팅

백엔드 - 장바구니와 주문 기록의 분리를 통한 결제 시스템 안정화

< 장바구니 >

```
CartItem  
id      Int @id autoincrement()  
quantity Int  
isChecked Boolean true  
deletedAt DateTime?  
createdAt DateTime now()  
updatedAt DateTime updatedAt()  
user    User  
product Product
```

“장바구니”의 상품이 결제되면,
“주문된 상품”을 기록하는 모델이 존재

< 주문된 상품 >

```
OrderedItem  
id      Int @id autoincrement()  
product Product  
order   Order  
price   Int  
quantity Int
```

“주문된 상품”은 “장바구니”에 연동되어 있어,
상품 정보가 수정(UPDATE) 되는 경우 결제된 상품의 정
보까지 변경되는 치명적인 결함이 발견

< 영수증 >

```
Receipt  
id      Int @id autoincrement()  
productName String  
price   Int  
imageUrl String  
quantity Int  
createdAt DateTime now()  
product Product  
order   Order
```

“주문된 상품” 모델 대신, ”영수증”이라는
뚜렷한 목적을 나타내는 모델로 개선

주문된 상품 기록은 **불변**해야 하지만 상품 정보 수정에 따라 **가변**하는 문제



결제 시점 상품의 정보를 안전하게 **보존**하는 영수증 모델로 개선



• 트러블슈팅

기획의도 수정 - 문제 발생

01 구매 횟수 집계 오류



코카콜라 29회 구매

2,000원

코카콜라 - 29회 구매



환타 오렌지 29회 구매

2,400원

환타 오렌지 - 29회 구매



&

코카콜라 구매 이력 손실

• 트러블슈팅

기획의도 수정 - 문제 발생

02 장바구니 데이터의 동기화 문제

The image displays two side-by-side screenshots of a mobile application's shopping cart screen. Both screens show a list of items with a total price at the bottom and a 'Buy Now' button at the bottom right.

Left Screenshot:

- Header:** 전체 선택 (1개) | 선택 삭제
- Item 1:** 롯데 자일리톨 껌 1,200원 (with a checked checkbox)
- Bottom:** 5 ▼ 총 6,000원 | 즉시 구매

Right Screenshot:

- Header:** 전체 선택 (1개) | 선택 삭제
- Item 1:** 오리온 닥터유 단백질바 2,500원 (with a checked checkbox)
- Bottom:** 5 ▼ 총 12,500원 | 즉시 구매

A large black arrow points from the left screenshot to the right one. Above the arrow, a callout bubble contains the text "상품 정보 변경".

• 기획의도 수정 - 문제 발생

정확한 구매 횟수 집계 **불가능**

기존 구매 횟수 이력 **손실**

유저가 장바구니에 담은 상품 **강제** 정보 변경



모든 것은 기본 요구사항인

“상품 수정”

기능으로부터 비롯된
문제점이라는 것을 파악했습니다.

• 기획의도 수정 - 해결

상품 수정 기능에 대한 기획 방향 결정

상품 정보 수정 항목 제한

초기에는 상품 정보 수정 기능을 제공하되,

일부 항목(가격)만 수정 가능하도록 제한하는 방향을 고려

개선 방향 문제점 인식

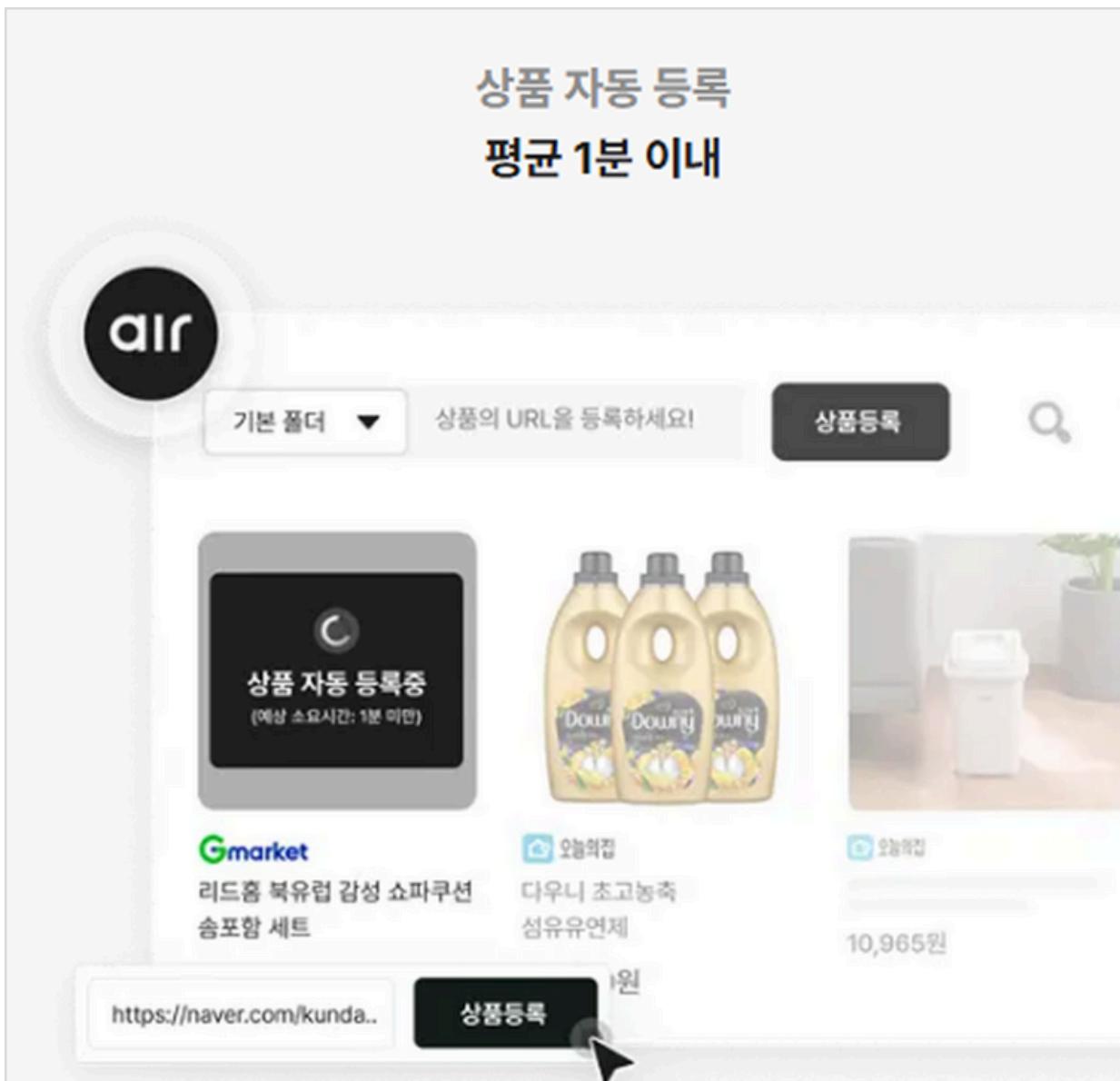
사용자가 직접 상품을 등록하는 것이 아니라,

이미 시중에 존재하는 상품의 URL을 공유하는 구조

• 이 구조에서 사용자가 가격을 수정할 수 있도록 하는 것도 플랫폼의 방향성과 맞지 않는, 비자연스러운 로직이라는 판단에 도달했습니다.

• 기획의도 수정 - 해결

벤치마크 사례 참고



air 에어서플라이

벤치마크한 플랫폼인 에어서플라이에서는 URL 입력 시
상품 정보가 자동 완성되는 구조를 사용하고 있었습니다

• 기획의도 수정 - 해결

최종 결정

- ✓ ‘상품 수정’ 기능은 UX와 기획 의도 모두에 부자연스럽다고 판단
- ✓ 결국 해당 기능은 제외하는 것으로, 더 명확한 흐름을 갖는 방향으로 기획을 재조정



• 우리는 단순히 주어진 요구사항을 그대로 구현하는 개발자가 아니라,
기획 의도를 고민하고, 사용자 경험을 고려하며 판단하는 개발자입니다.

07

회고

- Keep
- Problem
- Try



• 회고 KEEP

간접적 실무 경험

★ 4.9



페이지 중심의 국한된 기능을 맡지 않고, 서로 다른 FE, BE를 맡아서 실제 다른 개발자와 협업하는 것처럼 느껴져서 좋았습니다

균형적인 역할 분담

★ 4.7



업무를 랜덤하게 배정함으로써 공통 컴포넌트, API, 페이지 등 다양한 영역을 모두가 경험하며 역할을 고르게 분담했습니다.

원활한 의사소통

★ 4.4



GITHUB DISCUSSION, ISSUES 탭 등을 적극적으로 활용하여 프로젝트의 퀄리티가 크게 향상된 것 같다고 느꼈습니다.

활발한 분위기 조성

★ 4.5



긴 프로젝트 동안 각 팀원들을 배려하고 협력하여 팀을 이끌어 활발한 분위기를 조성

• 회고 PROBLEM

컨벤션 미준수

★ 1.0



정해진 컨벤션을 제대로 준수하지
않아서 일을 반복하게 되는 현상
발생

개발 지연

★ 2.0



계획한 일정에 맞게 개발이 끝나
지 않아 리팩터링 시간이 짧았음

긴 회의시간

★ 2.5



잦고 긴 회의시간으로 인해 기능
구현 중 집중이 자주 흐트러지는
때가 있었음

아쉬운 출결 보고

★ 3.0



출결에 대한 보고 및 기록 체계를
명확히 구축하지 못함

• 회고 TRY

컨벤션 리마인더

★3.5



한 번에 많은 컨벤션을 숙지하기
에 어려움이 있으므로 리마인더
시도

체계적 일정관리

★3.6



특정 업무가 예상 일정보다 늦게
구현되어서 다음 할일이 밀리는
것을 방지하기 위한 대안이 필요

프로젝트 목적 구체화

★4.1



기획 단계에서 흐름이 명확하지
않은 부분을 담당자와 적극적으로
소통하여 프로젝트 목적을 구체화
하는 것이 중요하다고 느낌

명확한 초기 설계

★3.8



초반 설계를 명확하게 하여
중간에 큰 구조를 변경 없이 안정
적으로 진행할 필요를 느꼈다.

08

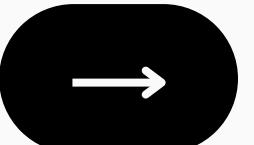
Q&A





SNACK

• **감사합니다**



이태빈 이지수 김우주 김홍섭 장원빈 조성빈
Snack 사이트 : <https://5nack.site>