

BALANCE EAT



밸런스일조 | 코알누 리액트 스터디 1조



발표 순서



- 1 팀원 소개
- 2 프로젝트 소개
- 3 프로젝트 협업과정
- 4 프로젝트 시현
- 5 팀원별 느낀점

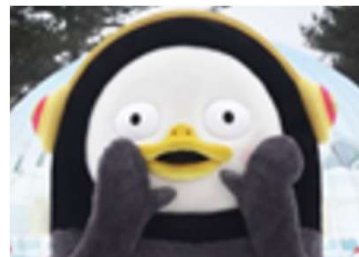
팀원 소개



S.M 이소영



FE 안윤희



FE 장유진



P.O 김은혜

밸런스일조가 해결하려고 했던 문제

 식단 관리를 더 편하게, 다음 끼니의 고민까지 덜어주는 경험을 제공하자

● POINT.01

직접 계산하지 않아도
영양 파악이 가능하게 하자

● POINT.02

기록뿐 아니라 이해를
돕는 코멘트를 함께 제공하자

● POINT.03

영양 상태에 맞춰 다음
끼니를 자연스럽게 추천하자

● POINT.04

직접 계산할 필요 없이 자연스럽게
관리가 이어지도록 만들자



Tech Stack / API Integration

📌 농촌진흥청 국립식량과학원_농식품 식단관리 API

- 음식별 영양 성분(칼로리, 탄단지, 나트륨 등) 조회
- 사용자 입력 음식의 영양 정보를 받아 끼니별·하루 단위 영양 분석의 기초 데이터로 활용

📌 Gemini AI API

- 하루 식단 데이터를 기반으로 **영양 분석 코멘트** 생성
- 부족하거나 과한 영양소 판단, 다음 끼니 추천 등
- “기록 → 분석 → 인사이트 제공” 흐름을 담당하는 프로젝트 핵심 AI 엔진

📌 로컬 스토리지 기반 저장

- 아직 서버 없이도 사용자의 입력 기록이 유지되도록 구현했습니다.
- (추후 확장성을 위해 DB 연동 계획이 있습니다.)



Team Cooperation

밸런스일조의 협업 (디자인, 깃, 코드 컨벤션)

Design_ Figma / 디자인 가이드

→ 서비스 구조·UI 흐름 정리, 주요 페이지 와이어프레임 공유

Git_Git Flow 전략

main/develop 분리 · 기능 단위 feature 브랜치로 개발

→ 충돌 최소화 & 안정적 머지

Commit Convention

feat / fix / refactor / style / doc

→ 코드 리뷰 효율 증가

Folder Convention

역할 기반 구조

(pages / components / hooks / stores / utils)

→ 유지보수성과 협업 효율 향상

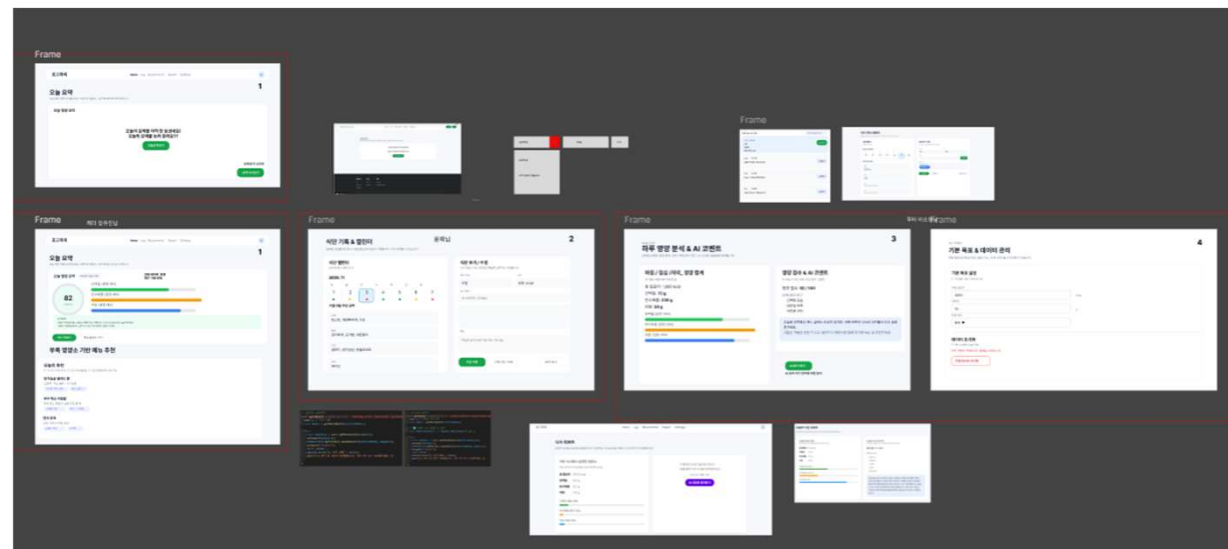
CSS Style Guide _ CSS Modules

→ 파일명 일관성, 컴포넌트 단위 스타일링, 스타일 충돌 방지



Design_ Figma / 디자인 가이드

서비스 구조·UI 흐름 정리, 주요 페이지 와이어프레임 공유



Folder Convention

역할 기반 구조

(pages / components / hooks / stores / utils)

→ 유지보수성과 협업 효율 향상

폴더 구조

```
├─ node_modules/
├─ public/
├─ src/
│  ├─ assets/      # 이미지 등 정적 리소스
│  ├─ common/      # 공용 컴포넌트
│  ├─ hooks/        # React Query 기반 API 요청 혹 (useQuery)
│  ├─ layouts/      # 페이지 전체 구조(레이아웃) 컴포넌트
│  ├─ pages/        # 라우트 페이지 컴포넌트
│  ├─ stores/       # Zustand 상태 관리 스토어
│  └─ utils/        # 유틸 함수, axios 인스턴스 등 공통 로직
├─ App.jsx
├─ App.css
├─ index.css
├─ main.jsx
├─ .gitignore
├─ eslint.config.js
├─ index.html
├─ package.json
├─ package-lock.json
├─ README.md
└─ vite.config.js
```

Git_ Git Flow 전략 / Commit Convention

main/develop 분리 · 기능 단위 feature 브랜치로 개발 충돌 최소화 & 안정적 머지

✓ 브랜치 전략

기본 브랜치

- `main` → 배포/완성본
- `develop` → 개발 통합 브랜치

기능 개발 브랜치

- 기능 단위로 브랜치 생성:

예시) `feat/login-page`, `feat/user-api`

커밋 규칙 (Commit Convention)

<타입>: <변경 내용 요약>

사용 타입

- `feat`: 새로운 기능 추가
- `fix`: 버그 수정
- `docs`: 문서 수정
- `style`: CSS/레이아웃/스타일 변경(로직 없음)
- `refactor`: 리팩토링(기능 변화 없음)
- `chore`: 패키지, 설정 등 기타 변경
- `test`: 테스트 코드 추가/수정

CSS 스타일 가이드 (CSS Modules)

CSS Modules

→ 파일명 일관성, 컴포넌트 단위 스타일링, 스타일 충돌 방지

✓ CSS 스타일 가이드 (CSS Modules)

- CSS Modules 사용

사용 예시

1) 파일명 규칙

ComponentName.module.css

2) import 방식

```
import styles from "../HomeCard.module.css";

export default function HomeCard() {
  return <div className={styles.container}>Hello</div>;
}
```



프로젝트 시현

<https://balanceeat.vercel.app/>



✓ 이소영님 _ 느낀점 및 소감



소감:

프로젝트를 진행하면서 리액트에 대해 더 깊게 공부할 수 있었고, 특히 팀원들과 함께 공통으로 사용하는 상태 관리를 고민하고 구현해본 경험이 큰 도움이 되었습니다. 부족했던 점도 분명히 있었지만, 직접 만들고 부딪히면서 확실히 성장할 수 있었던 시간이었습니다.

✓ 안윤희님의 느낀점 및 소감



소감:

이번 프로젝트를 진행 하며 처음으로 AI API를 활용해보는 경험을 했습니다. 시작 단계에서 접근방식조차 어떻게 해야 할지 막막했으나, 제미나이 API 공식 문서와 공개된 깃허브 코드들을 공부하여 흐름을 이해해 나갔습니다. 프롬프트 구성, 데이터 전달 방식, 출력 형식 등에 따라 결과가 크게 달라진다는 것을 확인했고, 이는 설계단계에서 명확한 지시와 구조화가 필수라는 것을 알았습니다. 또한 로컬 스토리지와 zustand 를 결합해 유저 플로우를 구축 하는 과정에서 충분히 안정적인 사용자 경험을 설계 할 수 있었습니다. 무엇보다 각자의 강점을 발휘하여 부족한 부분을 서로 채워준 팀원들 덕분에 어려운 순간에도 넘어지지 않고 끝까지 완성 할 수 있었던 점이 가장 큰 의미로 남았습니다.

✔ 장유진님의 느낀점 및 소감



소감:

혼자하는 프로젝트가 아닌 만큼 책임감도 느끼고 본업으로 시간이 없었는데
팀원들이 도와주셔서 프로젝트를 잘 끝낼 수 있었습니다.

프로젝트를 진행하면서 어떻게 하면 좀 더 효율적으로 코드를 짤 수 있을지 고민할 수 있는 시간을 가지게 되어서
너무 좋았습니다. 나중에 이 기획으로 더 디벨롭 시켜서 완성도 있는 아웃풋도 만들어봤으면 좋겠습니다.

✓ 김은혜님의 느낀점 및 소감



소감:

이번 프로젝트에서 처음으로 PO 역할을 맡으면서 정말 큰 도전이었습니다. 처음 해보는 자리다 보니 무엇을 어떻게 결정해야 하는지 막막할 때도 있었고 팀의 속도에 충분히 보조를 맞추지 못해 스스로 아쉬움도 컸습니다. 그런데도 팀원들은 각자 맡은 기능을 책임감 있게 구현하고 일정을 스스로 조율해가며 놀라울 만큼 프로젝트를 진행해 가는 모습을 보며 존경스러움과 저 스스로 부족하다는 생각이 들었습니다. 제가 충분히 도움을 드리지 못한 순간들이 떠올라 미안한 마음도 크지만 그럼에도 불구하고 이렇게 멋진 결과물을 함께 만들어낸 팀원들께 진심으로 감사하다는 말을 꼭 전하고 싶습니다. 이번 경험을 통해 팀워크의 힘을 다시 느낄 수 있었고, 앞으로는 더 나은 팀원이 되기 위해 노력하겠습니다!



발표를 마칩니다

감사합니다

