

● ● 요건 다 내꺼

웹 프rototype 과정 4조

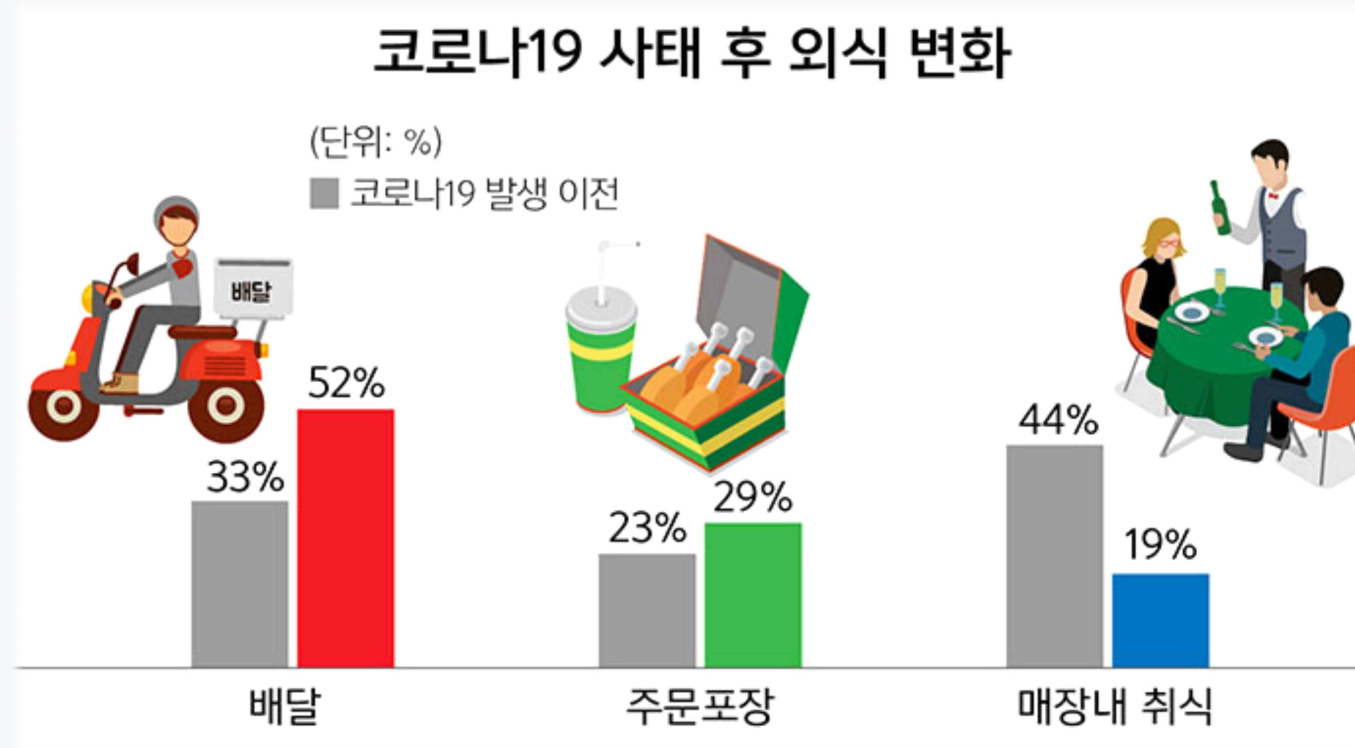
김민성 강래헌 박지훈 조은혜



01. 프로젝트 개요

프로젝트 선정 배경

#코로나19 #팬데믹 #집콕 #배달음식 #확진자



프로젝트 선정 배경

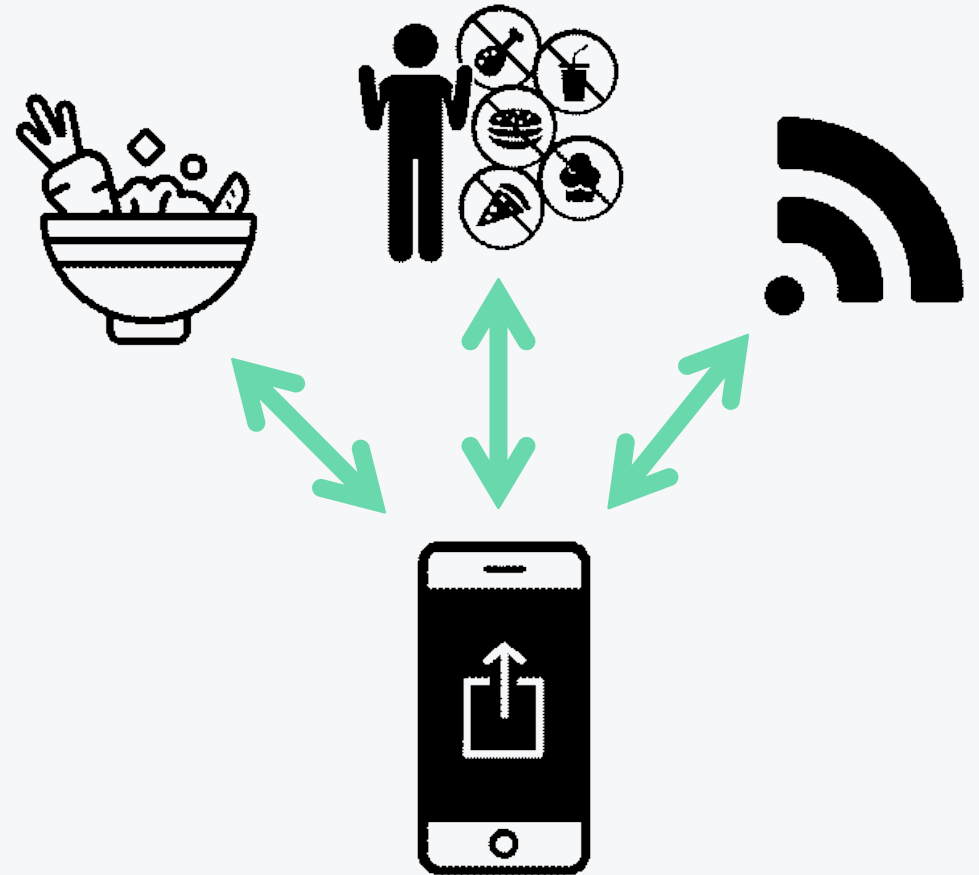
#건강 #집밥 #다이어트 #습관형성

- ✓ 손쉬운 요리 레시피 작성과 공유를 돕는 '레시피 서비스'

사용자의 신체조건에 기반하여 최적의 영양 섭취와 식단 공유를 돕는 '다이어리 서비스'

- ✓ 다른 사용자와 함께 도전과제를 수행하며 콘텐츠 참여와 성취감 달성을 유도하는 '챌린지 서비스' ⑦

“사용자는 『요건 다 내꺼』를 통해 건강한 식생활 습관을 형성하는 데 도움을 받을 수 있다.”



핵심 구현 기능

1

회원가입 다원화

- 이메일 인증절차를 거치는 회원가입 기능 구현
- OAuth 2.0 기반 Google, Kakao REST API 이용한 로그인 기능 구현

2

다이어리 서비스

- 신체조건, 관리 목표 설정 옵션에 따라 맞춤형 인터페이스 제공
- 날짜별 식단과 일기 CRUD 가능
- 식단 작성시 레시피와 식품 영양정보 실시간으로 검색 가능

3

레시피 서비스

- 레시피 접근성을 높이기 위하여 다양한 검색, 조회 기능 제공
- 작성된 레시피는 다이어리 서비스에서 사용 가능
- 신뢰성있는 영양정보 제공을 위해 식약처DB 이용

4

챌린지 서비스

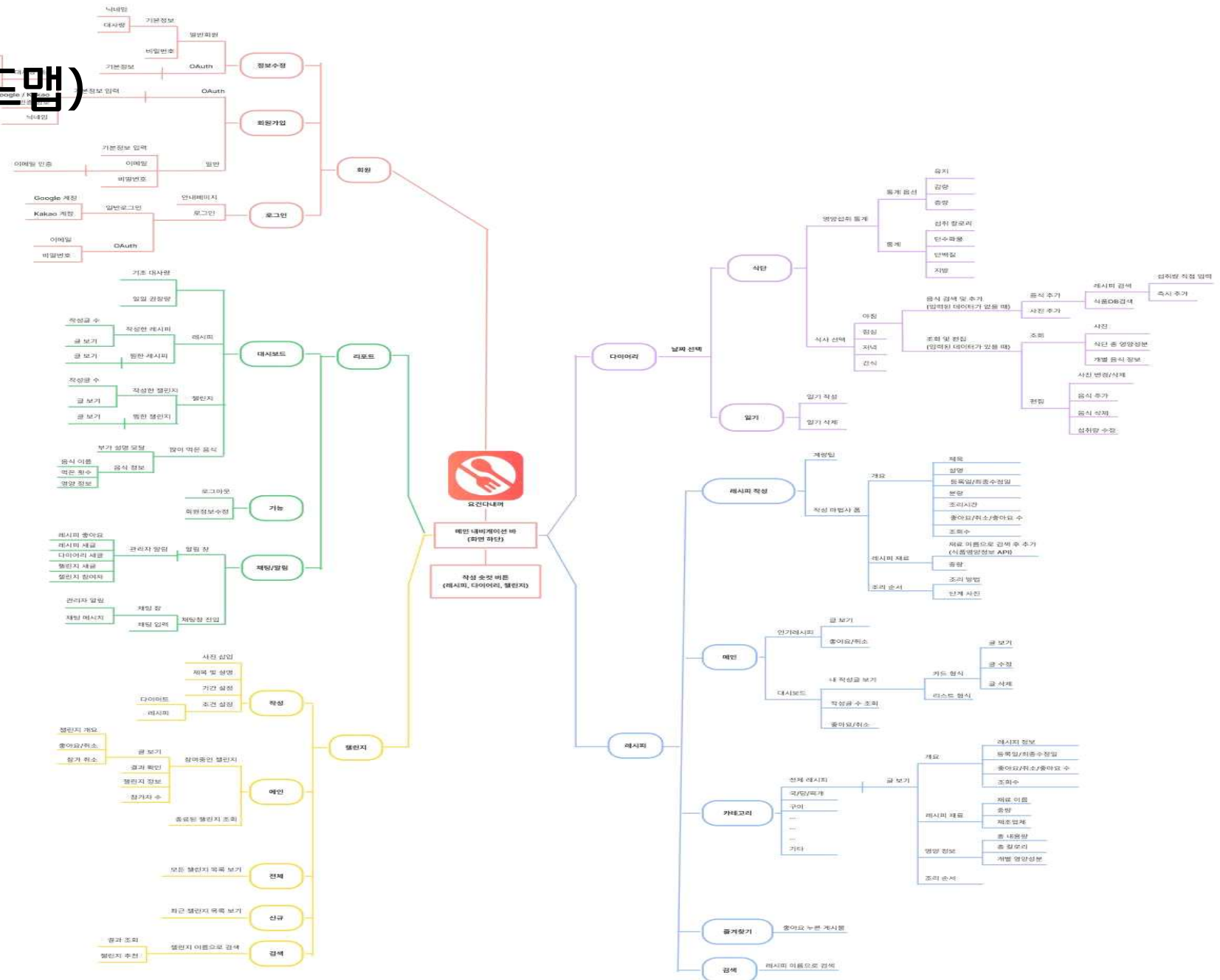
- 레시피, 다이어리 서비스와 연계한 도전 과제 서비스 제공
- 챌린지 검색 및 공동 참여 인터페이스 제공
- 챌린지 결과 조회 및 챌린지 추천 기능 구현

5

리포트 서비스

- 회원 활동 데이터를 가공하여 개인별 통계 페이지 제공
- 내가 많이 먹은 음식, 내가 좋아요 누른 콘텐츠 확인
- 실시간 서비스 알림 / 채팅기능

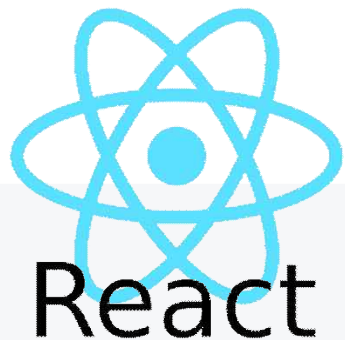
프로젝트 구조 (로드맵)





02. 개발 환경 및 기술 스택

개발 환경



SPA(Single Page Application)

방대한 레퍼런스와 전용 라이브러리

컴포넌트 단위 개발 유지보수 및 확장 용이

지속적으로 상승하는 점유율

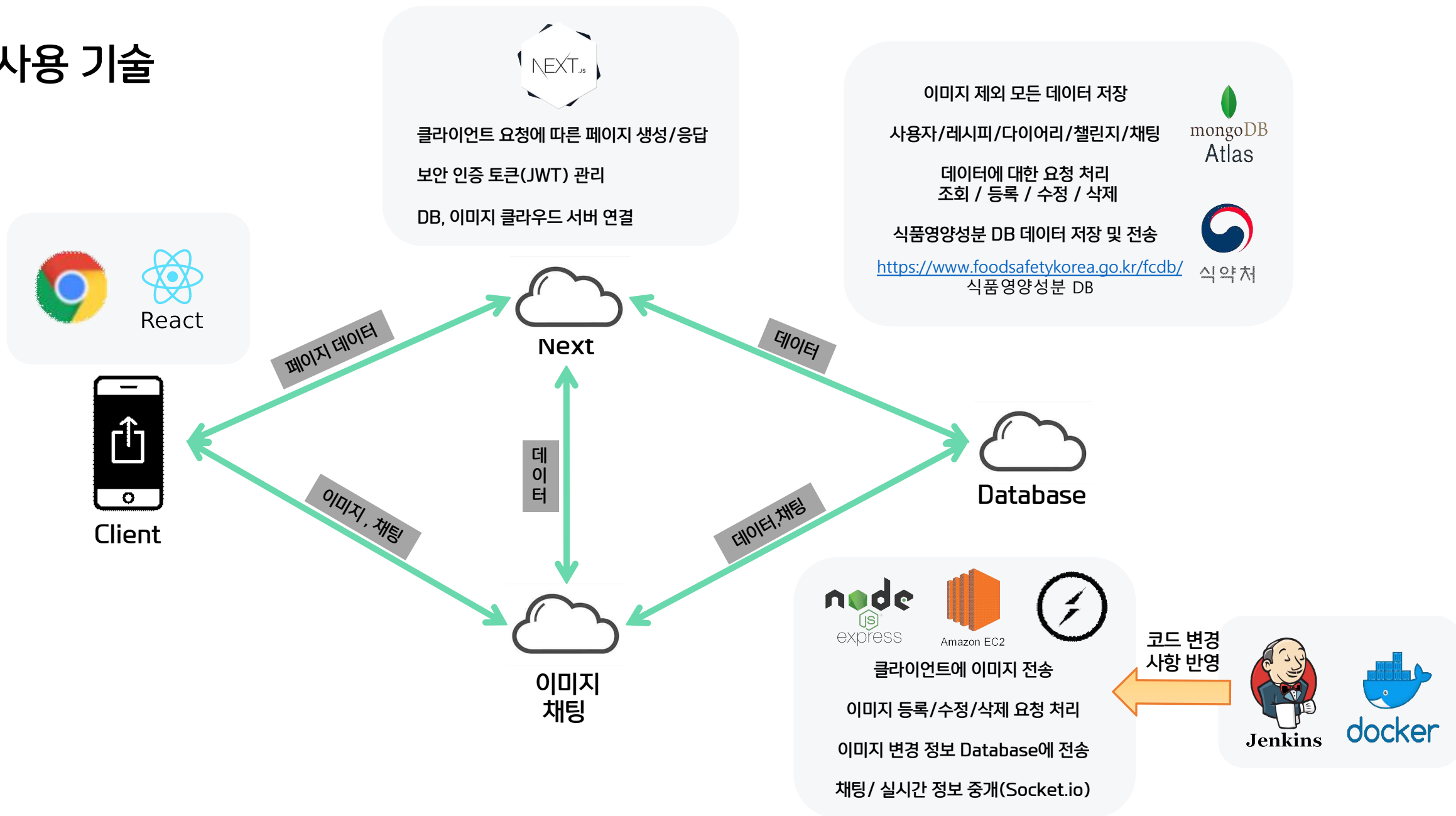


SSR

- 요청이 들어올 때 Next.js 서버에서 데이터를 준비하고 html 페이지에 미리 렌더링 후 브라우저에 전달
- 검색 엔진이 콘텐츠를 확인할 수 있으므로 SPA의 SEO(검색 엔진 최적화)문제 해결

File System Routing

- Page 폴더 내부의 파일 이름을 기준으로 URL 구분
- 구분된 페이지는 개별적으로 렌더링
- 첫 렌더링에 모든 URL에 대한 페이지를 불러오는 SPA의 속도 문제를 완화





03. 프로젝트 팀 구성 및 역할

팀 구성 및 역할 소개

김민성

데이터베이스 구축 및 관리
MongoDB Atlas

정적 파일 API 서버 생성 및 관리
AWS EC2 Express.js

회원 관리 / 인증 구현

OAuth 2.0 인증 구현

통합 개발 환경 구축 및 관리

클라이언트 비동기 처리 구현 지원

스키마 및 ERD 작성

개인 리포트 페이지 기능 구현

실시간 채팅/공지 기능 구현
(WebSocket 활용)

HTTPS 도메인 등록



강래현

레시피 서비스 설계/기능 구현/UI 디자인

레시피 데이터 설계

레시피 서비스 API 작성

공공 API 이용, 음식 데이터 설계



박지훈

챌린지 서비스 설계/기능 구현/UI 디자인

챌린지 데이터 설계

챌린지 참가 기능 및 취소 기능 구현

챌린지 좋아요 기능 및 검색 기능 구현

챌린지 결과 확인 기능 구현



조은혜

다이어리 서비스 설계/기능 구현/UI 디자인

로그인, 회원가입, BMR, 회원정보 수정 CSS

BMR+하루 권장 칼로리 계산 로직 및 폼

칼로리 및 영양 섭취 통계 프로그레스 기능

식단 옵션 선택에 따른 파이차트 변환

레시피, 식품영양성분 DB 검색 필터 기능

입력 값에 따른 영양 성분 동적 변화

식단 추가 및 삭제, 편집, 조회 기능

일기 작성 및 삭제 기능

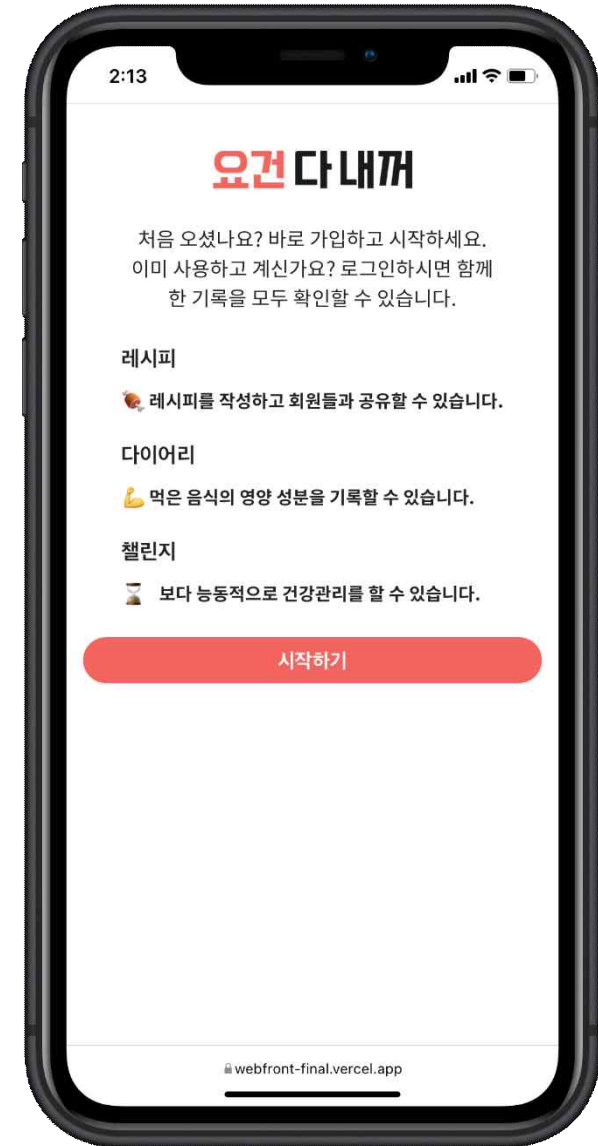
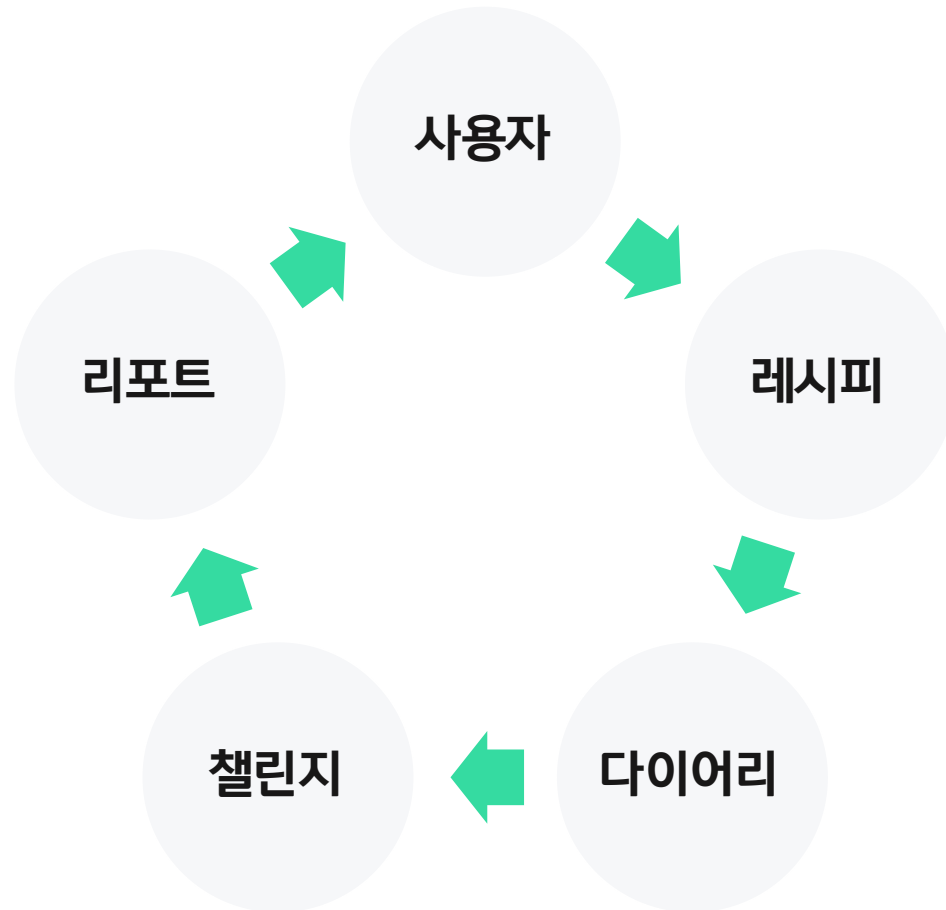
이미지 업로드, 삭제, 대시보드 반영





04. 프로젝트 수행 및 결과

시연 동영상 목차



사용자

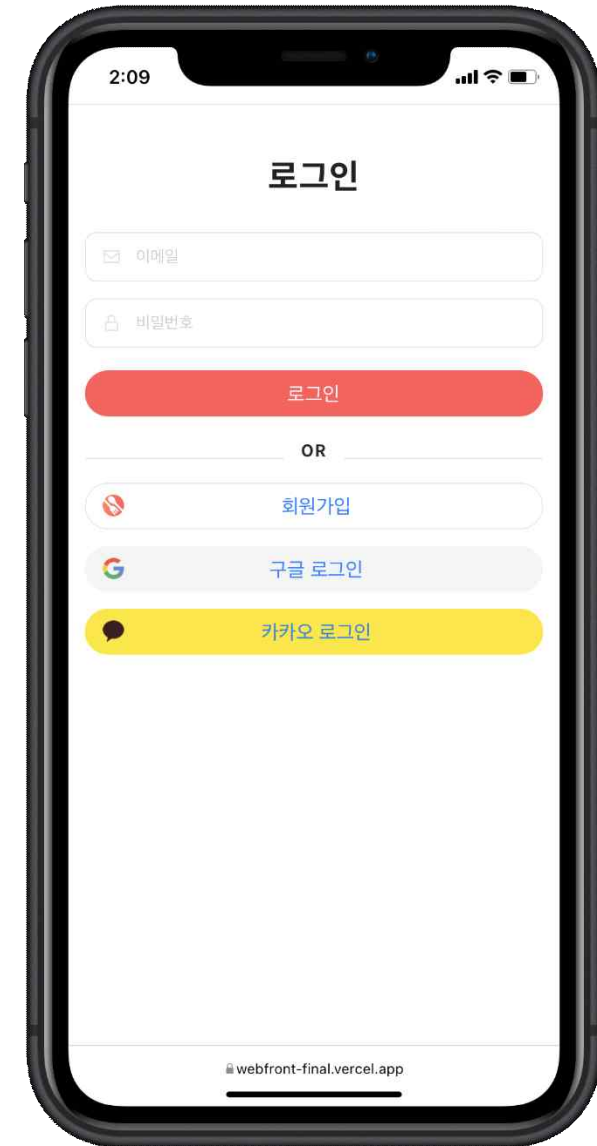
* 좌우 화면 이동을 통해 사용자가 페이지가 아닌 모바일 앱을 사용한다는 느낌 부여

1. 회원가입 / 정보 수정 페이지

- 키, 성별, 몸무게 입력을 받아서 사용자 기초 대사량 데이터 계산 및 저장 / 및 수정
- 좌/우 페이지 슬라이드로 모바일 입력 편의성 강화
- 회원 가입 시 이메일 인증 메일을 사용자의 이메일이 맞는지 확인
- OAuth 회원 입력 내용 간소화(이메일, 비밀번호 X)

2. 로그인 페이지

- 일반 로그인:
이메일 인증 여부 확인을 통해 사용자가 소유한 이메일이 맞는지 확인
- OAuth 로그인
서비스 계정(카카오, 구글)의 고유 아이디가 사용자 DB에 있는지 인증
이메일, 비밀번호 입력이 번거로운 모바일 환경에서 손쉬운 로그인 가능



레시피

1. 메인 페이지

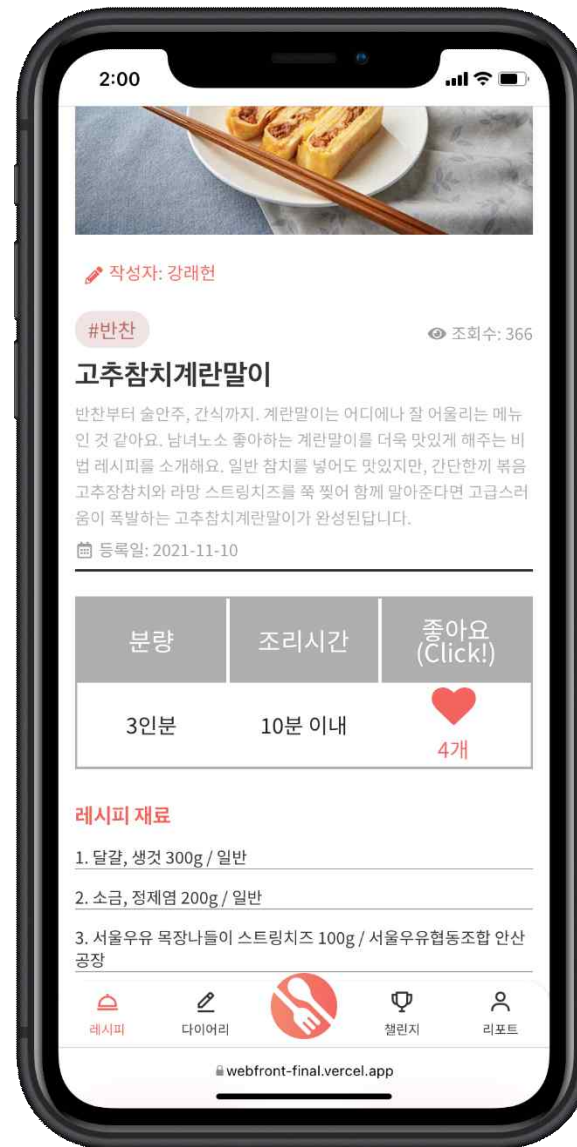
- 이용자 아이캐치를 위한 슬라이드 형식의 UI
- 다양한 서비스에 손쉽게 접근 가능

2. 작성 페이지

- hook을 이용한 state관리로 깜빡임 없는 서비스 이용 가능
- 식약처 데이터를 DB에 적재하여 응답시간 최소화
- 작성시 추가한 재료에 따라 브라우저가 로직에 따라 영양정보를 계산하여 DB에 저장
- 이미지 업로드 기능

3. 조회 및 수정

- 원하는 레시피에 쉽게 접근할 수 있도록, 레시피 데이터의 name이나 category 필드값에 따라 다른 결과가 출력되도록 함
- 사용자 경험을 고려한 좋아요/취소 기능 탑재
- 수정 페이지 진입 시 DB에서 각 필드값 받아온 후 기존 데이터 표시 후 수정



다이어리

1. 서비스간 높은 상호 연동성

- 사용자: 회원가입 시 작성한 신체 조건 데이터를 기준으로 식단 관리 목적에 따라 맞춤형 인터페이스를 제공
- 레시피: 식단 추가 시 작성된 레시피 실시간 검색 가능
- 챌린지: 챌린지 작성 시 다이어리 서비스 기반 도전 과제 제공

2. 사용자 친화적 모바일 서비스

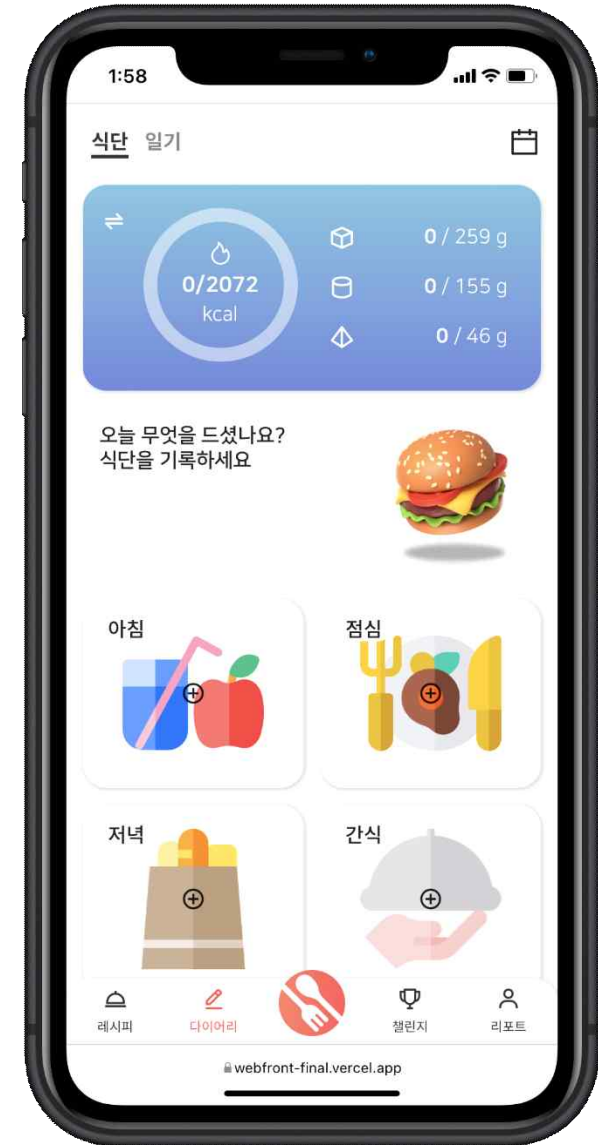
- 빠른 반응성, 화면 전환 애니메이션 등 사용자 친화적 모바일 서비스를 위해 React hooks를 이용한 state관리로 깜빡임 없이 이용 가능
ex) useState 사용 writingMode 생성, 데이터 유무에 따라 상이한 콘텐츠 노출

3. 식단 추가 및 실시간 수정 (저장된 데이터가 없는 경우)

- 식약처 데이터를 DB에 적재하여 검색 시 응답시간 최소화- 레시피 또는 재료 추가 시 입력 값에 따라 영양 성분을 계산
- 추가된 재료 및 음식이 실시간으로 리스트에 출력되며 삭제 및 계량 수정 가능
- 이미지 업로드 및 삭제 가능

4. 식단 조회 및 수정 (데이터가 있는 경우)

- 저장된 식단과 이미지 출력
- 수정 시 식단추가 콘텐츠로 이동



챌린지

1. 메인 페이지

- 사용자 참여중인 챌린지와 종료된 챌린지를 보여줌으로써 자신의 동기 부여를 만들어줌
- 다른 페이지를 자유자재로 이동 가능

2. 작성 페이지

- 요건 다 내꺼의 챌린지만의 고유 조건과 데이터 구축 및 작성 시 DB에 챌린지 고유의 데이터 저장
- 이미지 업로드 기능

3. 조회 및 결과

- 진행중인 챌린지 조회 기능
- 조회한 챌린지가 없을 경우 추천, 인기 챌린지를 표시
- 사용자가 결과를 확인하고 싶으면 진행중인 챌린지의 결과를 조건에 따라 검증해서 성공 여부를 확인할 수 있는 기능



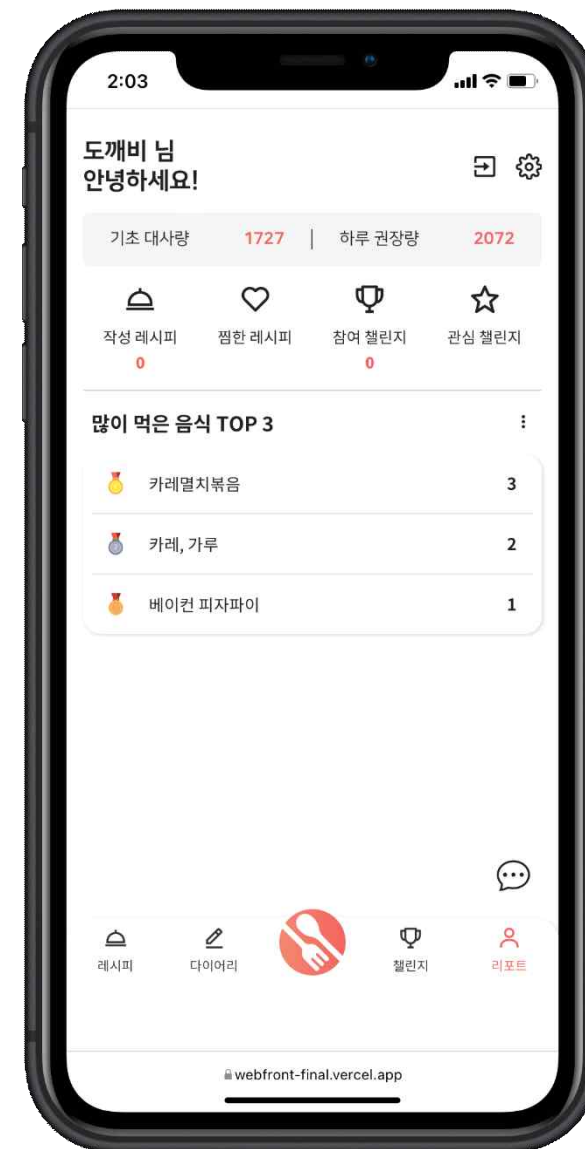
리포트

1. 개인화 대시보드

- 사용자 정보 시각화
- 대시보드 클릭시 추가적인 정보 제공
- Ajax와 React Hooks를 통한 비동기 상태 변경
- 식단 Top 3 데이터 등을 통해 식습관 확인 가능
- 각 항목 별 개인 콘텐츠에 접근할 수 있는 링크 제공

2. 알림 / 채팅

- 애플리케이션 전체 사용자 활동 실시간 알림
- 부가적인 채팅 기능
- 애플리케이션의 경량화된 커뮤니티 기능.





05. 자체 평가 의견 및 향후 계획

GOOD ☺

1

지속적인 통합 개발 / 배포 환경 설정

실제로 사용자가 사용할 수 있는 수준까지 개발 환경을 설정해서 개발 상황에 대한 현실적인 피드백을 받을 수 있었다.

특히 렌더링 속도가 로컬 개발 환경보다 상당한 차이가 있다는 것을 경험해서, 좋은 코드의 중요성을 체감할 수 있었다.

2

빠른 기본 기능 구현

관계형 데이터베이스보다 보다 자유롭게 데이터를 다룰 수 있는 NOSQL(MongoDB)를 활용해서 빠르게 데이터 구조를 만들 수 있었다.

또한 팀원 전부가 능동적으로 프로젝트에 참여하고, 할당된 역할에 필요한 기술을 빠르게 익혀서 계획보다 이른 시점에 기본 기능들을 완성할 수 있었다.

덕분에 팀원 일부가 기피했었던 영역(CSS, React Hooks 활용) 또한 충분히 연습하고 활용할 수 있는 기회가 생겨서 프론트엔드에서 요구하는 역량을 키울 수 있었다.

3

Only JavaScript

프론트엔드(React), 백엔드(NodeJS), 데이터베이스(MongoDB) 모두 자바스크립트를 활용한다.

덕분에 다른 역할군의 코드를 이해하고 서로의 요구 사항을 쉽게 전달할 수 있었다.

BAD ☹

1

사용자 참여 유인책 부재

회원 활동 포인트, 챌린지 달성 상품 제공 등 사용자 참여를 유도하는 기능을 구현하지 못했다.

즉 사용자 건강 습관 참여를 유도하는 애플리케이션의 목적을 지속적으로 인지하지 못하고 파편화된 구현에 시간을 보냈다는 아쉬움이 있다.

2

미흡했던 기획 및 설계

프로젝트에 대한 장기적인 설계, 기획의 중요성을 간과하고 기술에 대한 이해와 구현에 초점을 두었다.

애플리케이션의 기본 기능을 계획보다 빠르게 구현하는 것은 성공했으나, 확장성이 떨어지는 설계로 인해 애플리케이션을 차별화할 수 있는 기능을 추가하지 못했다.

3

역할 배분 실패

프론트 엔드 역할군에 과도한 인원을 할당했다. 그 결과 부실한 프로젝트 구조 설계로 이어졌고, 앞서 말한 문제들을 초래했다.

또한 백엔드에서 통합 개발 환경설정, 실제 서비스 배포에 지나치게 많은 시간을 투자했기 때문에, 프론트엔드에서 활용할 수 있는 다양한 데이터를 제공하지 못했다.

다음 프로젝트에는?

1

설득력 있는 주제 선정 및 기획

본격적인 프로젝트를 시작하기 전에 충분한 토론을 통해 설득력 있는 주제를 선정해서, 개발하는 제품이 실제 사용자 경험과 괴리되지 않도록 한다.

2

정기적인 피드백

팀원 전체가 규칙적인 피드백을 통해 계획했던 제품의 완성도와 실제 진행 상황을 비교한다.

개발하고 있는 애플리케이션이 프로젝트 주제에서 벗어나지 않는지, 우선순위가 낮은 단기 목표에 지나치게 많은 자원을 사용하고 있지 않는지, 프로젝트의 설계에 근본적인 결함이 있는지 등 전체 팀 차원에서 점검하고 발견된 문제점들을 해결한다.

3

균형 있는 역할 배분

백엔드 프론트엔드 역할을 균형있게 배분하여 한 쪽 역할의 업무 지체로 인해 프로젝트 전체의 진행에 차질이 생기는 것을 방지한다.