

React · Spring Boot 기반 소셜 피트니스 플랫폼



fitHub

프로젝트 개발 보고서

최승현 | 전한기 | 양현석 | 나한빈 | 정호균

목차

| 01

프로젝트 소개

| 02

기획 의도 및 유사 서비스 분석

| 03

팀원 소개 및 역할 분담

| 04

개발 목표 및 설계 주안점

| 05

개발 환경 및 기술 스택 요약

| 06

버전 관리 전략 및 커밋 컨벤션

목차

| 07

개발 일정

Development Schedule

| 08

와이어 프레임 - Figma

| 09

DB 설계 - ERD

| 10

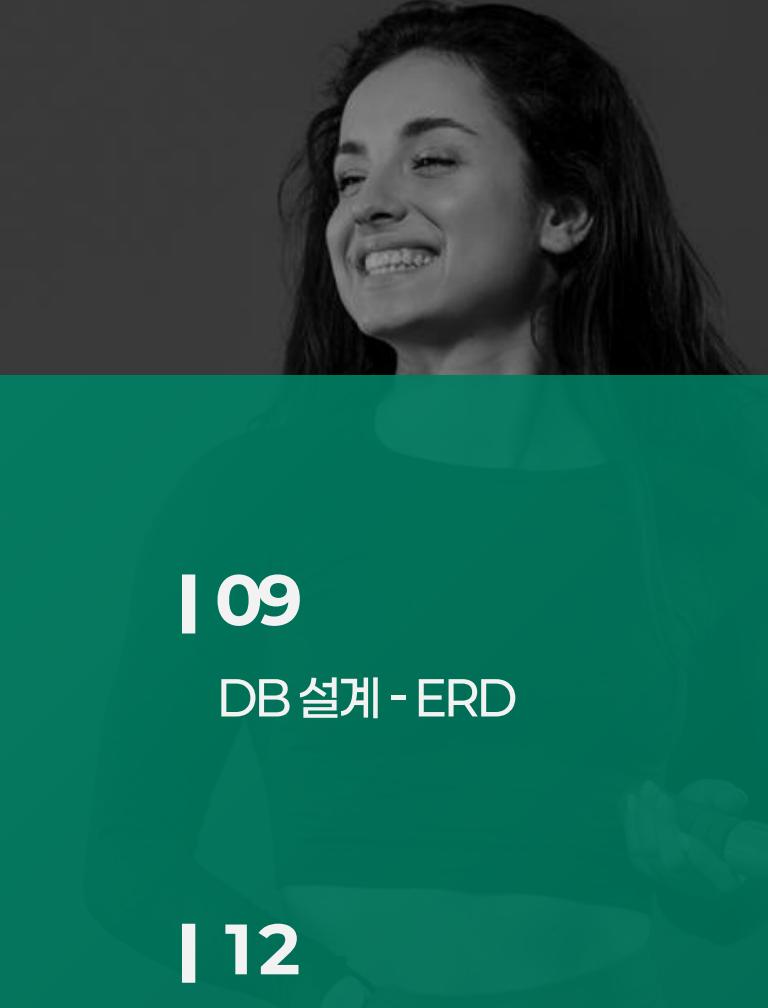
아키텍처 및 핵심 코드 구조 설명

| 11

주요 기능 및 상세 기능 설명

| 12

API 명세서 - Swagger



목차

| 13

트러블 슈팅 및
기술적 이슈 해결 사례

| 14

실제 구현 화면

| 15

프로젝트 회고 (팀 & 개인)





1. 프로젝트 소개

소셜 피트니스 플랫폼

fitHub는 사용자의 운동 기록과 목표 관리를 중심으로, 커뮤니티 기반의 소셜 피드 기능과 운동 상품 거래 기능(관리자 등록 · 회원 구매)을 결합한 웹 기반 피트니스 플랫폼입니다.

React 기반의 **CSR**(Client Side Rendering) 방식으로 사용자 경험을 향상시키고, Spring Boot 기반의 **RESTful API** 서버와의 연동을 통해 백엔드 기능을 구현한 풀스택 구조의 웹 프로젝트입니다.

사용자 편의성과 서비스 확장성을 고려하여, **SPA**(Single Page Application) 특유의 빠른 화면 전환과 비동기 처리 기반의 유연한 기능 제공을 목표로 개발되었습니다.

2. 기획 의도 및 유사서비스 분석

기획 배경 및 프로젝트 의도

기획 배경



최근 MZ세대를 중심으로 한 홈트레이닝, 퍼스널 피트니스 수요의 증가와 함께 **운동 습관 형성**에 대한 관심이 높아지고 있습니다. 하지만 기존 운동 플랫폼은 단순 운동 기록 저장 또는 루틴 제공에만 초점이 맞춰져 있어, 사용자의 **지속적인 동기 부여**와 **소셜 커뮤니티 기능**이 부족한 경우가 많습니다.

핏허브 기획 포인트



- ✓ 운동 기록 + 목표 설정 → 습관 형성 유도
- ✓ 소셜 피드 기능으로 사용자 간 피드백 및 동기 공유
- ✓ 운동 상품 거래 기능 통합으로 실질적인 운동 실천 유도
- ✓ CSR(Client Side Rendering) 기반으로 빠르고 유연한 UX 제공



“
**운동을 쉽게,
습관은 자연스럽게**

2. 기획 의도 및 유사서비스 분석

유사 서비스 분석

운동 기록 기반의 서비스와 콘텐츠·커머스 플랫폼을 분석하여,
fitHub 만의 차별화된 핵심 기능과 설계 방향을 정리하였습니다.

fitHub의 차별화 전략

기존 운동 플랫폼의 한계를 넘어, 기록 → 소통 → 실천까지
연결되는 “소셜 피트니스 생태계” 구축을 목표로 설계되었습니다.



- ✓ 웹 기반 피트니스 플랫폼
- ✓ 소셜 피드 중심 상호작용
- ✓ 운동 상품 거래 기능 통합
- ✓ React 기반 CSR + REST API 구조



주요 피트니스 서비스 비교

서비스명	주요기능	아쉬운 점	fitHub 차별점
Planfit	Ai 루틴 추천, 운동 기록	웹 미지원, 커뮤니티 부재	웹 기반 + 소셜 피드 제공
Hevy	운동 로그, 분석	한글 미지원, SNS 미흡	국내 사용 환경에 맞춘 UI/UX + 커뮤니티 중심설계
Nike Training Club	콘텐츠 스트리밍	단방향 정보 제공	소셜 피드 기반 상호작용 기능

“**기록 + 피드 + 상품 거래**” 세 가지 기능이 통합된
웹 피트니스 플랫폼은 **fitHub** 가 유일합니다.

3. 팀원 소개 및 역할 분담

Fithub 팀구성 및 역할

핵심 기능 단위로 전원이 풀스택 역할을 수행하며,
유기적인 협업을 통해 안정적인 서비스를 완성했습니다.



최승현
Member

회원가입 · 로그인 · 소셜 로그인
JWT 인증 시스템 · 마이페이지
공통 레이아웃 구성 · 메인페이지 설계
로고 디자인 · 프로젝트 문서화



전한기
Myfit

운동 기록 · 나의 루틴 · 활동 로그
운동 요약 · 통계 · 그래프 시각화
운동 기록 AI 요약
DM 채팅



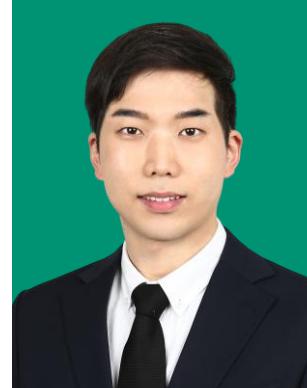
양현석
Admin

사이트(회원 · 게시글) 관리
사이트 통계 조회 및 시각화
사용자 문의 채팅
광고 관리



나한빈
Community

커뮤니티 피드 · 상세보기
게시물 등록 · 수정 · 삭제 · 검색
댓글 · 좋아요
팔로잉 · 팔로워



정호균
Market

마켓 상품 목록 · 상세 페이지
마켓 상품 등록 · 수정 · 삭제
장바구니 · 상품 구매
리뷰 · 구매 목록

4. 개발 목표 및 설계 주안점

개발 목표 및 설계 방향



개발 목표

기록 + 소통 + 실천 연결

운동 기록, 커뮤니티 피드, 상품 거래를 하나의 흐름으로 구성

CSR 기반 사용자 경험

React 기반 CSR 방식으로 빠르고 부드러운 사용자 경험 구현

SPA 구조 설계 최적화

페이지 전환 최소화 및 상태 관리 최적화를 고려한 설계

REST API 연동 백엔드

Spring Boot 기반 REST API 구조로 프론트와 유연하게 연동

사용자 중심 설계

정보 구조 단순화
UX 흐름 최적화



기능 모듈화

컴포넌트별 책임 분리
유지보수 용이



인증 구조 설계

JWT 기반 로그인 +
токен 관리 최적화



기술적 확장성

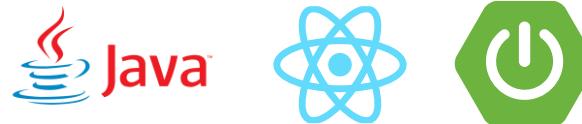
React CSR +
REST API 기반
SPA 구조



5. 개발환경 및 기술스택 요약

Development Environment

- React + Spring Boot 기반 풀스택 SPA 아키텍처
- AWS EC2 서버 운영 + GitHub Actions로 CI 적용
- REST API 통신 / JWT 인증 기반 로그인 처리
- 협업 도구: GitHub, Notion, Figma, ERDCloud



“

유기적인 기술연동과
실제 구현 중심의 구조설계에
중점을 두었습니다.

Language	라이브러리	프레임 워크	DataBase & 개발 도구 (Tool & IDE)	Infra & 배포	협업 & Docs	스타일 & 마크업
Java 17	React, React Router	Spring Boot 3.4.3	Oracle 11g XE SQL Developer	GitHub Actions	GitHub	HTML5
JavaScript (Node.js)	Axios	Spring Security	STS 4 (Spring Tool Suite)	AWS EC2	Notion , GitMind	CSS
SQL	Recoil	MyBatis	VS Code	Google Cloud Console	Figma	MUI
HTML5	MUI		Swagger	Kakao Developers Console	ERDCloud	

6. 버전 관리 전략 및 커밋 컨벤션

Version Control & Commit Convention

브랜치 전략



팀원별 브랜치 생성 → main 브랜치에 병합

- main : 배포용 안정화 브랜치
- [기능별] 브랜치 : 팀원이 독립적으로 작업
예) feature/login, feature/myfit 등
- 병합 방식 : 개별 기능 개발 후 main 브랜치에 직접 머지
- CI : main 브랜치 push 시 GitHub Actions로 테스트 자동 실행



GitHub 기반 협업으로 버전 이력 관리와 병합 충돌 방지에 집중

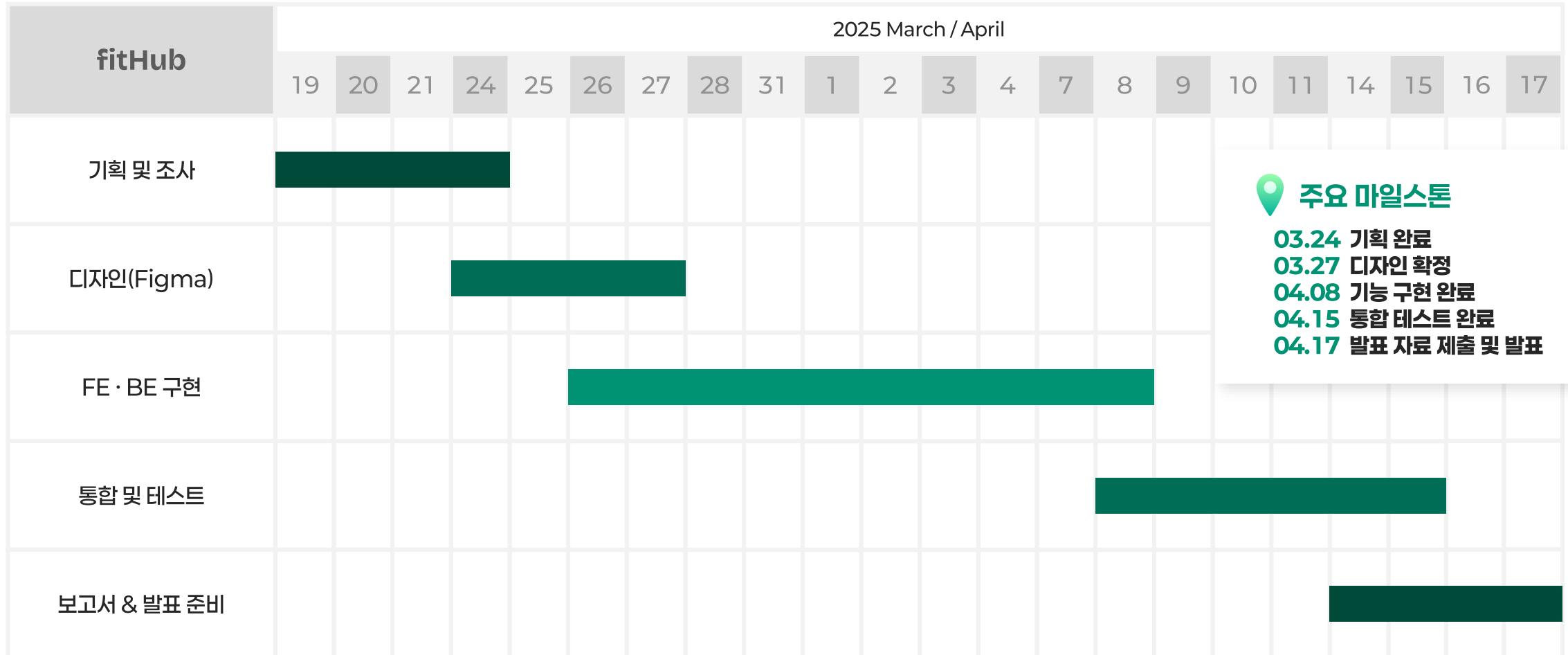
커밋 컨벤션

Prefix	설명
feat	새로운 기능 추가
fix	버그 수정
refactor	리팩토링 (기능 변화 없음)
style	코드 포맷, 세미콜론 등 스타일 수정
docs	문서 및 주석 수정
test	테스트 코드 추가 또는 수정
chore	빌드 설정, 패키지 등 기타 변경

Ex) feat: 마이페이지 회원정보 수정 기능 구현

7. 개발 일정

Development Schedule



7. 개발 일정

Development Schedule

구간	기간	주요 내용
기획 및 조사	03.19 ~ 03.24	기획안 정리, 유사 서비스 분석, ERD 구조 확정
디자인(Figma)	03.24 ~ 03.27	와이어프레임 제작 및 화면 구성 협의
FE · BE 구현	03.26 ~ 04.08	메인 페이지, 회원가입/로그인 페이지, 마이페이지, My Fit, 마켓 등 DB 설계, API 구현, 인증/관리자 기능 개발
통합 및 테스트	04.08 ~ 04.15	프론트-백 연결, 기능 통합, 오류 수정 QA 체크리스트 기반 디버깅 수행 및 반복 테스트
보고서 & 발표 준비	04.14 ~ 04.17	최종 보고서 작성, 발표 자료 정리 및 리허설



요약 포인트

- 담당자별 기능 모듈화를 기반으로 병렬 개발
→ 통합 단계에서 효율적인 연계 도모
- 마무리 2주간 집중적으로 통합 및 발표 준비
- 기능 구현 후 UI 흐름 및 마이페이지 연동 등

전체 사용자 경험 흐름 점검에 집중



협업 및 진행 방식

- 기능별 브랜치 분리 후 main 브랜치에 직접 머지
- [Figma](#)로 프로토타입 공유
- [Notion](#)을 통한 일정 관리 및 회의 기록
- [GitMind](#)로 기능 설계 흐름 시각화
→ 다양한 협업 도구를 활용한 팀 피드백을 통해

UI 통일성과 데이터 흐름 개선을 병행

8. 와이어프레임 - Figma

Wireframe - Member

공통 레이아웃 및 사용자 메인 흐름

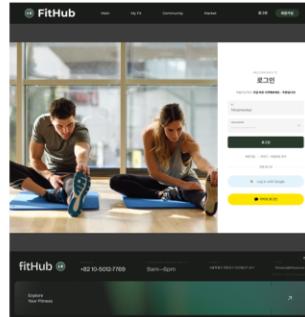
Header/Footer



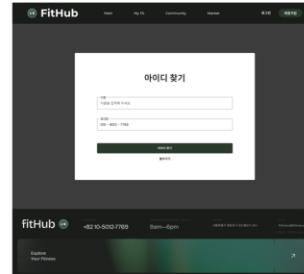
메인페이지



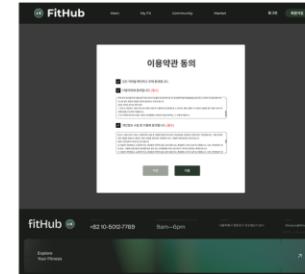
로그인



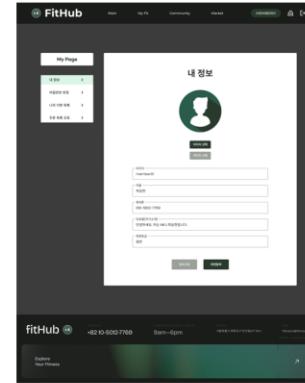
아이디찾기



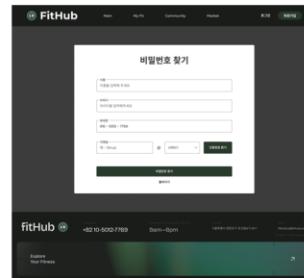
약관동의



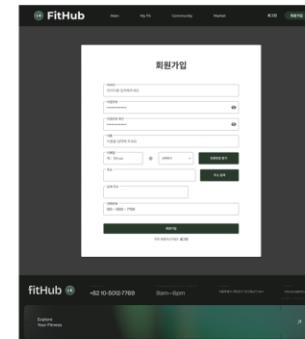
マイ페이지



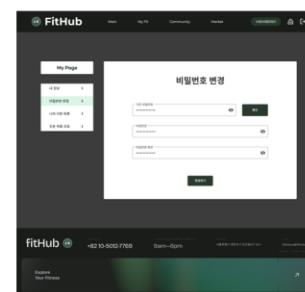
비밀번호 찾기



회원가입



비밀번호 변경



공통 레이아웃을 기반으로, 메인 화면부터 회원가입, 로그인, 정보 수정까지의 사용자 주요 흐름을 설계하여 전체 사용자 경험이 일관되고 명확하게 전달되도록 구성하였습니다.

8. 와이어프레임 - Figma

Wireframe - My Fit

운동 루틴 등록부터 기록, 요약까지의 흐름

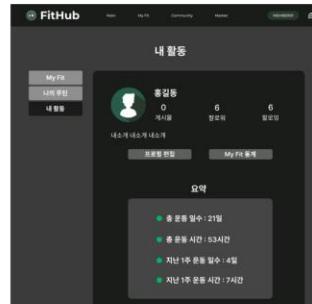
My Fit - 기록 캘린더화면



과거



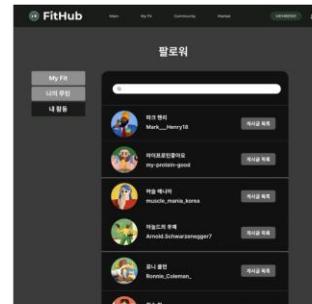
프로필



운동 루틴 등록



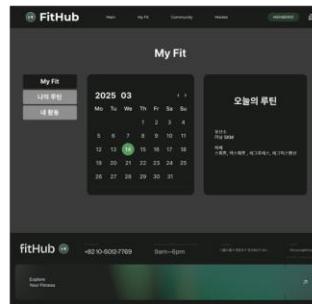
팔로워 - 커뮤니티 연계 기능



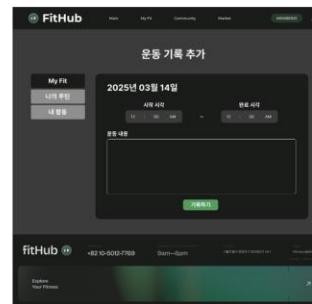
기록 존재



미래



운동 기록하기

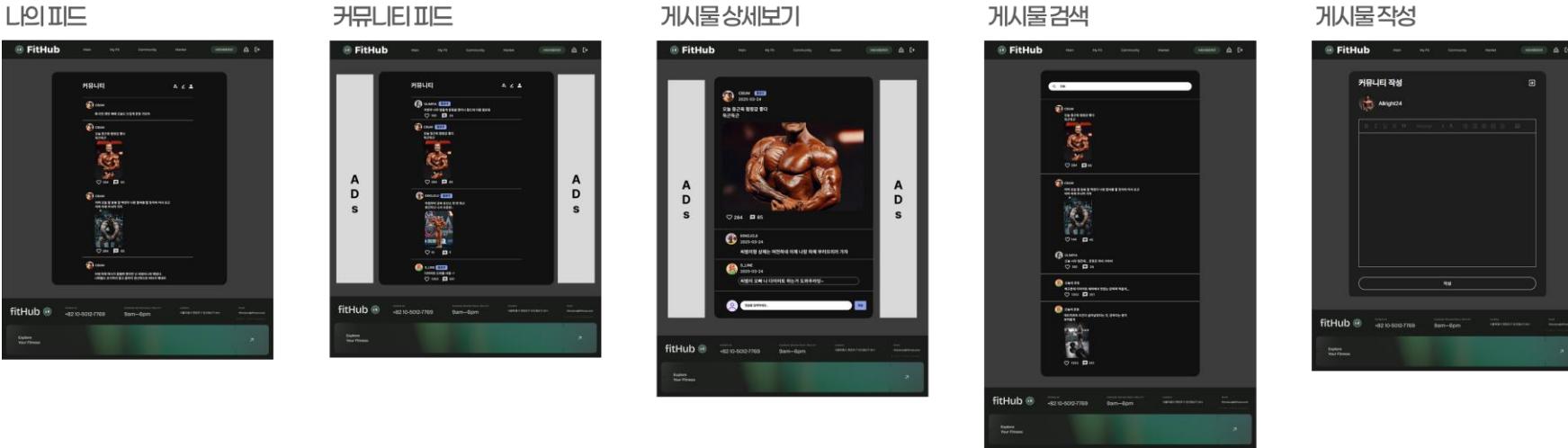


운동 루틴의 생성부터 실시간 기록, 통계 요약까지의 흐름을 설계하여 사용자가 자신의 운동을 체계적으로 관리할 수 있도록 구성했습니다.
캘린더 기반 UI로 시각적인 피드백을 제공하며 사용자 경험을 강화했습니다.

8. 와이어프레임 - Figma

Wireframe - Community

회원 간 소통 중심의 커뮤니티 흐름



자유롭게 글을 작성하고, 피드백을 주고받을 수 있는 커뮤니티 흐름을 설계하였으며
댓글과 좋아요, 검색 등 다양한 상호작용 기능을 통해 사용자 간 소통을 유도했습니다.

8. 와이어프레임 - Figma

Wireframe - Admin

관리자 회원·통계 관리 및 광고/문의 응대 흐름

회원 및 게시글 관리



통계 관리



문의 내역



문의 내역 상세보기



광고 관리



문의 페이지



특정 회원 문의 목록



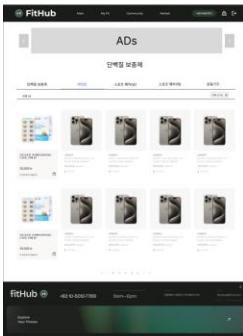
관리자는 회원 목록 및 상세정보 조회, 광고 설정, 1:1 문의 응대 등 서비스 전반을 모니터링하고 관리할 수 있도록 구성하였습니다.

8. 와이어프레임 - Figma

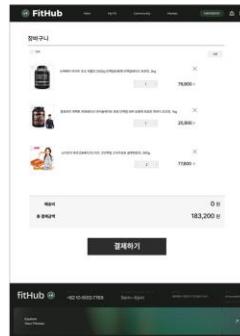
Wireframe - Market

상품 탐색부터 구매, 리뷰까지의 흐름

상품 리스트페이지



장바구니



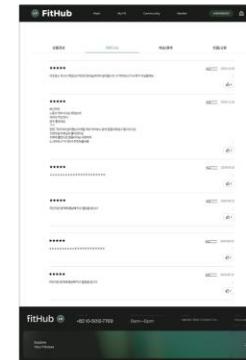
결제페이지



주문배송



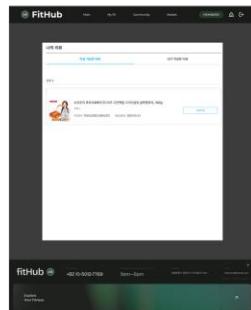
리뷰페이지



리뷰작성



나의 리뷰



상품 상세 페이지

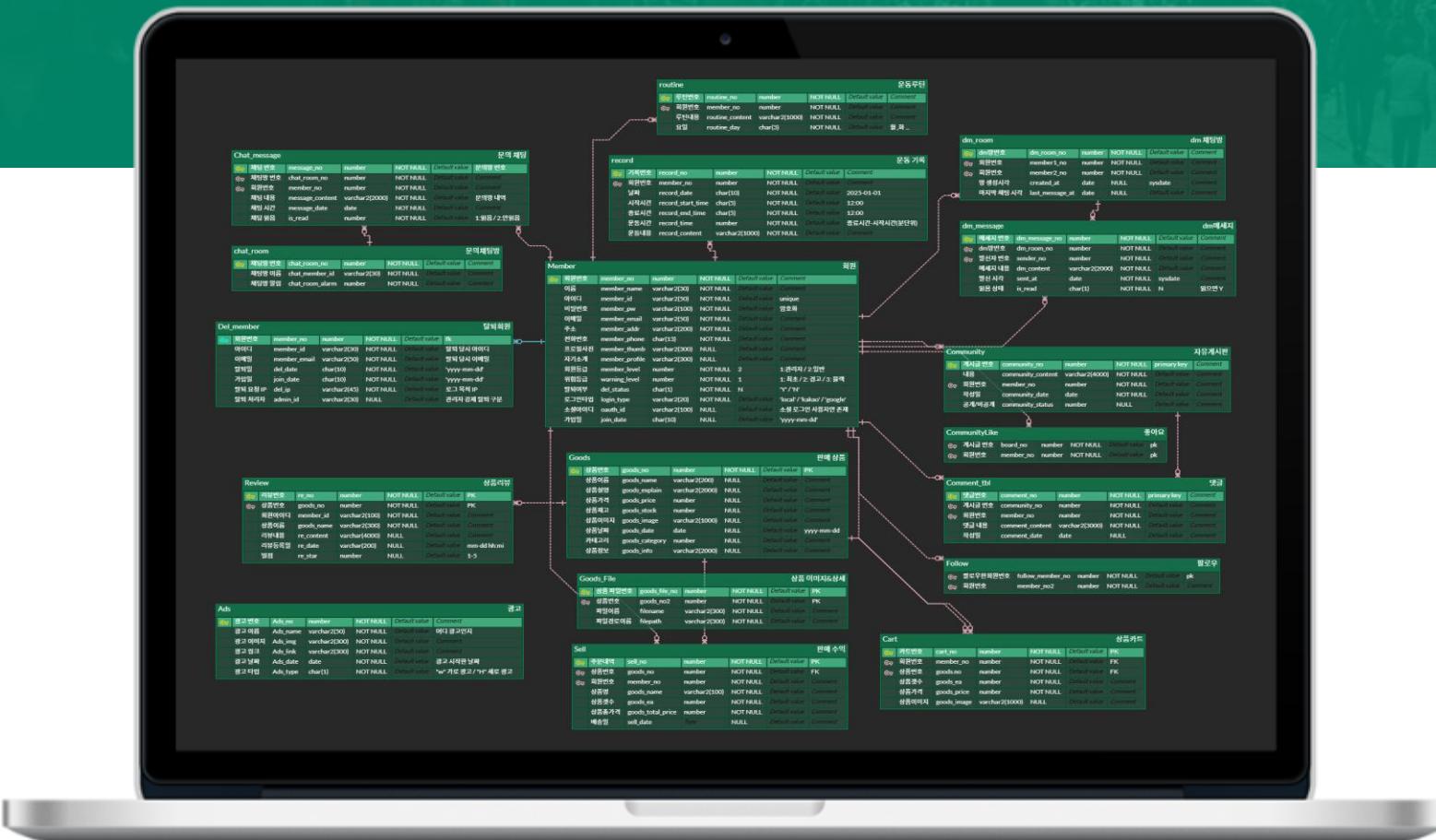


상품 목록 조회부터 상세 보기, 장바구니, 구매 및 리뷰 등록까지 전체 흐름을 설계하여

사용자가 필요한 운동 상품을 쉽고 빠르게 구매할 수 있도록 구성하였습니다.

9.DB 설계-ERD

ERD(Entity Relationship Diagram) – 전체 구성



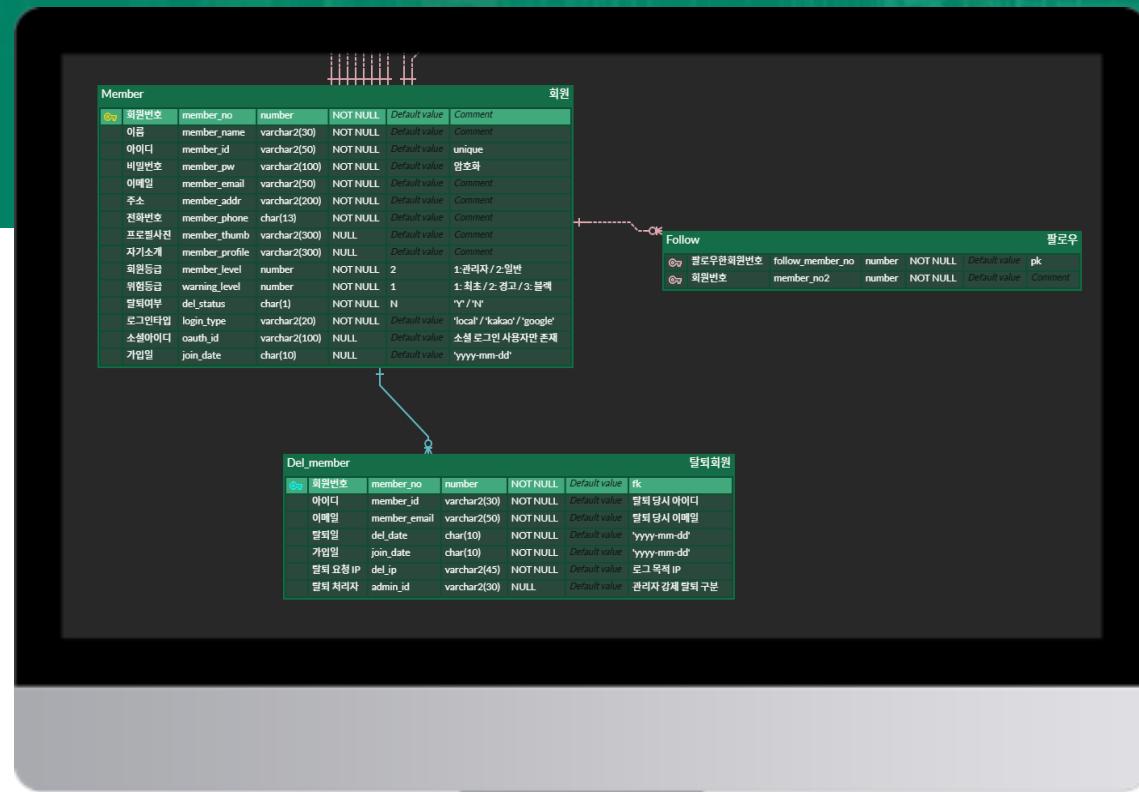
요약 포인트

FitHub의 주요 기능을 중심으로 사용자(Member), 운동(MyFit), 커뮤니티(Community), 마켓(Shop), 관리자/Admin) 등 도메인 별 테이블을 설계하고, 외래키를 활용해 데이터 간의 연관성을 명확히 구성했습니다.

- 사용자 중심 설계
- 기능별 도메인 구분
- 외래키 연계 및 무결성 확보
- 확장성 고려한 정규화

9.DB설계-ERD

사용자관리파트 – ERD 구성



주요 테이블 구성

테이블명

Member

설명

사용자 기본 정보
(아이디, 비밀번호, 이름, 이메일, 권한 등)

Del_member

탈퇴 회원 이력 보관용 테이블

Follow

회원 간 팔로우/팔로잉 관계 관리
(팔로우 요청자와 대상 회원 정보 저장)

9.DB설계-ERD

운동(My Fit) 파트 – ERD 구성



주요 테이블 구성

테이블명

Routine

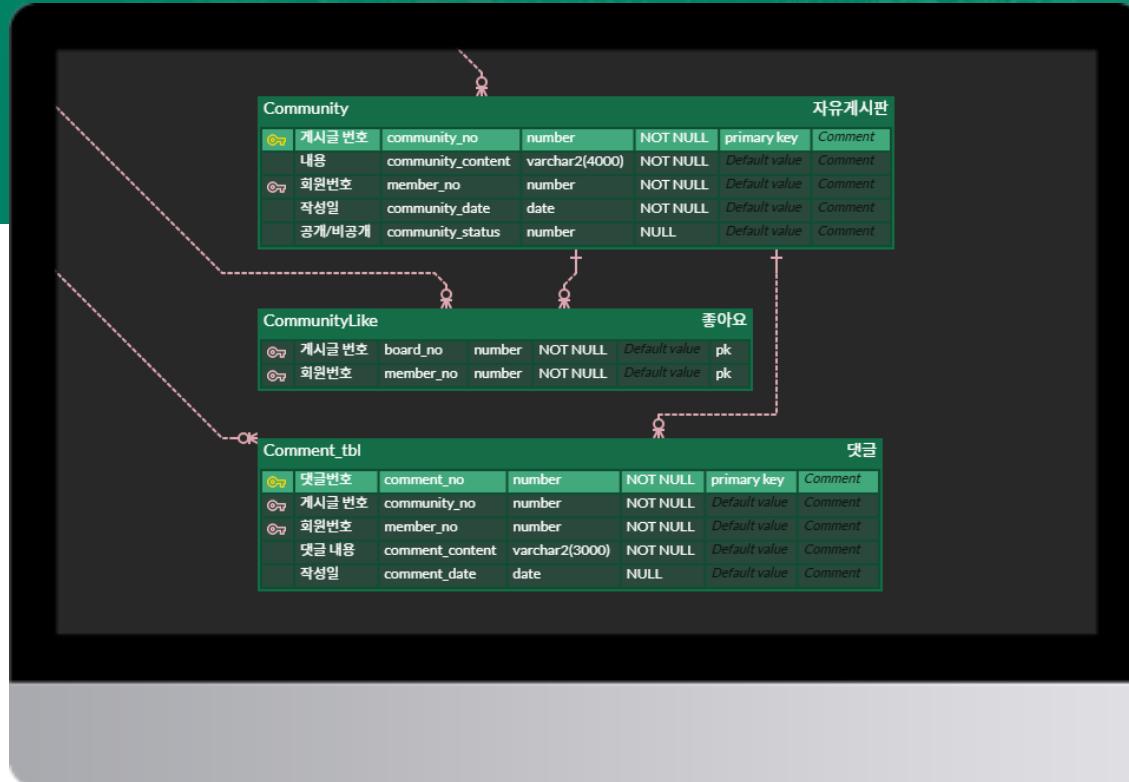
설명

사용자가 생성한 운동 루틴 정보 (루틴명, 내용, 요일 등)

Record

운동 수행 기록 데이터 (날짜, 루틴명, 수행 내용 등)

커뮤니티(Community) 파트 – ERD 구성



주요 테이블 구성

테이블명

Community

설명

게시글 정보 (제목, 내용, 작성자 등)

CommunityLike

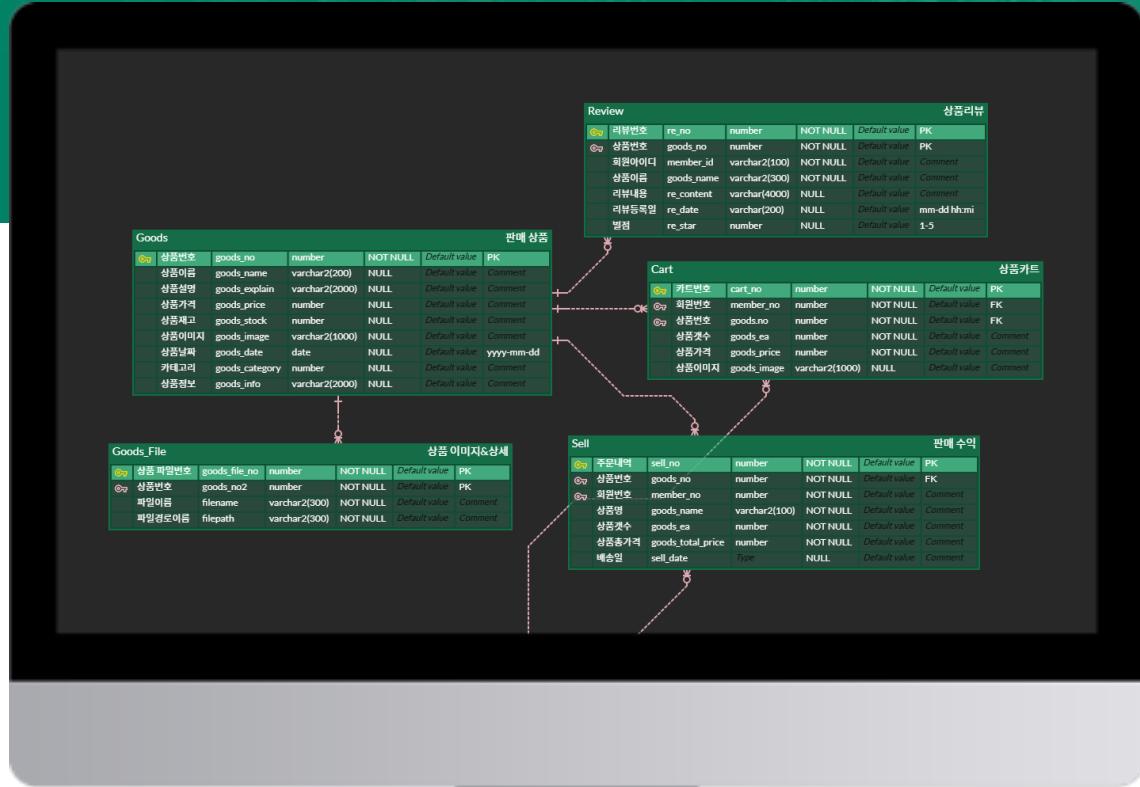
게시글 좋아요 정보 (게시글과 좋아요 누른 회원 매피)

Comment_tbl

댓글 정보 (내용, 작성자, 작성일 등)

9.DB 설계-ERD

마켓(Market) 파트 – ERD 구성



주요 테이블 구성

테이블명

Goods

상품 정보 (이름, 가격, 설명, 판매 상태 등)

Goods_file

상품 이미지 파일 경로 정보

Sell

실제 판매 내역 정보 (구매자, 수량, 금액 등)

Cart

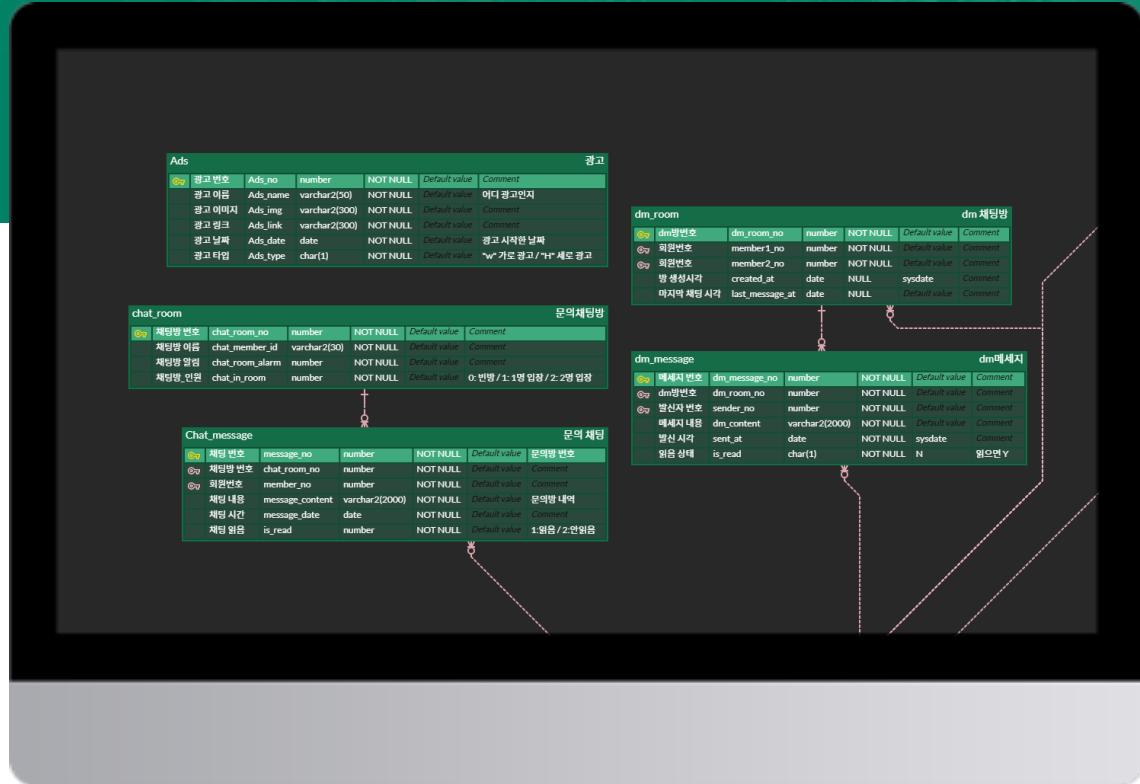
장바구니 정보 (회원이 담은 상품과 수량 등)

Review

구매 후 리뷰 정보 (평점, 후기 내용 등)

9.DB설계-ERD

관리자(Admin) 및 DM/채팅 파트 – ERD 구성



- 관리자 기능은 Ads 테이블 이외에도 회원 등급 수정, 강제 탈퇴 등의 작업에서 Member 테이블을 활용합니다.
이 외 테이블은 주로 조회 및 모니터링 용도로 사용됩니다.



주요 테이블 구성

테이블명

설명

Ads

관리자 등록 광고 정보 (이미지, 광고 문구 등)

Dm_room

1:1 쪽지방 정보 (송신자, 수신자)

Dm_message

쪽지 내역 정보 (쪽지 내용, 수신 확인 등)

Chat_room

실시간 채팅방 정보 (방 번호, 알림 설정 등)

Chat_message

실시간 채팅 메시지 정보 (내용, 보낸 회원 등)

fitHub SPA 아키텍처 & 핵심 로직 구성

fitHub SPA 아키텍처 개요

핏허브는 Single Page Application(SPA) 구조를 기반으로 하여, React와 Recoil을 사용한 클라이언트 측 렌더링 방식으로 사용자 경험을 향상시켰습니다. Spring Boot 기반의 백엔드는 RESTful API 기반의 명확한 요청-응답 구조를 통해 데이터를 전달하며, 프론트엔드와 백엔드의 역할을 분리하여 유지보수가 용이하고, 서비스 확장에 유리한 구조로 설계되었습니다.



핵심 처리 흐름 요약

- React 기반의 CSR 구조로 구성
- 전역 상태 관리는 Recoil로 관리, Axios로 API 통신 수행
- 서버는 Spring Boot + MyBatis 기반 RESTful API 설계
- 로그인, 회원가입, 마이페이지 등 주요 요청은 JWT 인증 기반으로 처리



핵심 처리 영역

- **Auth**
: JWT 발급 → Axios Interceptor → Header 자동 전송
- **State**
: Recoil 전역 상태로 로그인/회원정보 관리
- **Data**
: MyBatis 기반 SQL 처리로 빠른 CRUD 구현

11. 주요 기능 및 상세 기능 설명

홈화면 기능 상세 설명

헤더 · 메인 · 푸터

Header · Main · Footer



헤더

Header

- 상단 고정형 네비게이션 구성
- 로고 클릭 시 메인페이지 이동
- 메뉴
 - MyFit, 커뮤니티, 마켓, 로그인 등 메뉴 별 라우팅 처리
- 사용자 상태에 따른 표시 변화
 - └ 로그인 상태 : 아이디 표시 – 클릭 시 마이페이지 이동
 - └ 비로그인 상태 : 로그인 버튼 표시
 - └ 관리자 로그인 상태 : 관리자 버튼 추가
- DM 아이콘
 - 수신 여부 표시 → 클릭 시 DM List 이동



메인

Main

- 메인 배너
 - 핵심 기능 안내 및 Fade in/out 애니메이션 효과
- "지금 시작하기" CTA 버튼
 - 로그인 상태에 따라 로그인 또는 My Fit으로 라우팅 처리
- 주요 기능 안내 배너 → 기능 설명 및 이미지 슬라이드 형식 출력
- 슬로건 → 타이핑 애니메이션 효과



푸터

Footer

- 하단 고정형 구성
 - 스크롤 탑 버튼 / 문의 아이콘(관리자 문의 채팅) 포함
- 브랜드 정보 제공(로고, 전화번호, 고객센터 운영시간, 주소, 이메일)
- Explore Your Fitness 배너
 - 클릭 시 My Fit 이동

11. 주요 기능 및 상세 기능 설명

회원 기능 상세 설명

회원가입 · 로그인

Join · Login



회원가입

Join



로그인

Login

- **약관 동의**
 - 이용약관 및 개인정보 처리방침 확인 후 체크 → 다음 단계로 이동
- **사용자 정보 입력 및 유효성 검사**
 - 아이디 중복 확인 (API)
 - 비밀번호 강도 시각화 및 일치 여부 확인 (정규식 기반 + UI 피드백)
 - 전화번호 자동 하이픈 처리 및 정규식 검사
 - 이메일 주소 선택 or 직접 입력
 - 이메일 인증번호 전송 (중복 클릭 방지 및 타이머 표시)
 - 인증번호 확인 후 통과 시 다음 단계로 이동
 - 주소 검색 및 선택 기능 (다음 주소 API 연동) → 상세 주소 입력
- **가입 완료 : SweetAlert 토스트 알림 → 로그인 페이지로 이동**
- **일반 로그인**
 - 아이디, 비밀번호 입력 후 로그인
 - JWT 발급 → Axios Interceptor로 Header 자동 설정 → 인증 처리
- **소셜 로그인**
 - 구글, 카카오 로그인 연동 (OAuth 기반)
 - 소셜 로그인 후 추가 정보 입력 페이지로 이동 → 회원가입 완료
- **로그인 성공 시 전역 상태에 사용자 정보 저장**
- **아이디/비밀번호 찾기**
 - 아이디 찾기 : 이름 + 이메일 일치 검증 → 아이디 반환
 - 비밀번호 찾기 : 아이디 + 이메일 검증 → 임시 비밀번호 발급 메일 전송

11. 주요 기능 및 상세 기능 설명

회원 기능 상세 설명

마이페이지

Mypage



마이페이지

MyPage

- 회원 정보 관리
 - 이름, 연락처, 주소, 비밀번호, 자기소개 변경 가능
 - 프로필 이미지 미리보기 및 업로드
 - 회원 탈퇴 : 탈퇴 확인 모달 출력 → 회원 탈퇴 (API)
- 비밀번호 변경
 - 현재 비밀번호 확인 → 새 비밀번호 설정
 - 단계별 진행 상태 : MUI Stepper 시각화 UI 적용

11. 주요 기능 및 상세 기능 설명

마이핏 기능 상세설명

마이핏 · 나의 루틴

My Fit · My Routine



마이핏

My Fit



나의 루틴

My Routine

- 날짜 선택 시 조건별 출력
 - 과거 : 기록 출력
 - 오늘 : 기록 유무 따라 루틴/기록 표시
 - 미래 : 루틴 출력만 가능
- 운동 시작/종료 시간 입력 + 텍스트 기록
- 유효성 검사
 - 시작 > 종료 시 등록 불가
 - 빈 내용일 경우 : 신규 X, 기존 기록 삭제
- 기록 존재 시 AI 분석 → 소모 칼로리, 요약 제공

- 캘린더에서 날짜 선택 → 요일 기반 루틴 작성
- 요일별 내용 입력 가능 (다른 요일 이동 시 유지)
- [루틴 설정] 클릭 시만 적용됨
- 빈 루틴일 경우
 - 신규 : 반영 X
 - 기존 : 루틴 삭제 처리

11. 주요 기능 및 상세 기능 설명

마이핏 기능 상세설명

활동 로그 · DM

Activity Log · DM



활동 로그

Activity Log

- 내 운동 통계 확인 (토글 전환)
 - 요약형 / 그래프형 통계 보기
- 프로필 정보, 내가 쓴 글 목록
- 팔로워 / 팔로잉 리스트 확인
- 로그인 유저 = 해당 프로필이면 : [프로필 편집] 버튼 표시
- 다른 유저일 경우 : [팔로우] 또는 [팔로우 취소] 버튼 표시



DM

Direct Message

- 활동 로그의 [DM 보내기] 버튼 클릭 시 채팅방 입장
- 첫 메시지 전송 시 방 생성, 이후 DM 목록에서 유지
- DM List 정렬
 - 최근 메시지 순
 - 안 읽은 메시지
- 헤더 아이콘 + DM List 우측에 개수 표시
- 채팅방 입장 시 읽음 처리
- 메시지 UX
 - 같은 시간에 여러 메시지 → 마지막 메시지에만 시간 표시
 - 엔터 : 전송
 - 쉬프트 + 엔터 : 줄바꿈

11. 주요 기능 및 상세 기능 설명

커뮤니티 기능 상세설명

게시물 · 게시물 상세보기

Post · Post Details



게시물

Post



게시물 상세보기

Post Details

- 전체 조회
 - 모든 회원의 공개 게시물을 목록 조회
 - 비공개 게시물은 전체 리스트에서 숨김 처리되며, 본인만 열람 가능
- 작성 / 수정 / 삭제
 - 로그인한 회원만 게시물 작성 가능 (TextEditor 활용)
 - 작성된 게시물은 수정 및 삭제 가능 (작성자 본인만)
- 게시물 검색
 - 회원 아이디 및 키워드 기반 검색

- 커뮤니티 리스트에서 특정 게시물 클릭 시
→ 상세 내용, 이미지, 작성자, 좋아요, 댓글 등 전체 정보 확인 가능

11. 주요 기능 및 상세 기능 설명

커뮤니티 기능 상세설명

댓글 · 좋아요 · 팔로우

Comment · Like · Follow



댓글

Comment

- 댓글 작성
 - 특정 게시물에 로그인한 회원이 댓글 등록
- 댓글 수정
 - 본인이 작성한 댓글만 수정 가능
- 댓글 삭제
 - 본인이 작성한 댓글만 삭제 가능



좋아요

Like

- 등록
 - 특정 게시물에 좋아요 등록
- 취소
 - 기존 좋아요 취소 가능
- 개수 조회
 - 게시물에 등록된 총 좋아요 수 조회



팔로우

Footer

- 팔로우
 - 로그인한 회원이 다른 회원을 팔로우 등록 → '팔로잉'으로 전환
- 팔로잉 취소
 - 기존 팔로우 관계 취소 가능

11. 주요 기능 및 상세 기능 설명

마켓 기능 상세설명

상품 리스트 · 등록 · 상세보기

Goods List · Registration

Goods Details



- 메인 마켓 페이지에서 썸네일 형식으로 출력
- 최신순, 가격순 정렬 가능
- 관리자는 상품 수정 및 삭제 가능

- 관리자 페이지에서 상품 정보(이미지, 상세 설명, 카테고리) 입력 후 등록
- 등록된 상품은 자동으로 카테고리별 리스트에 분류됨

- 상품 클릭 시 상세페이지 전환
- 관리자 등록 정보 기반으로 출력

11. 주요 기능 및 상세 기능 설명

마켓 기능 상세설명

장바구니 · 상품 구매 · 리뷰

Cart · Goods Purchase · Review



장바구니

Cart

- 장바구니 추가
 - 상품 상세 페이지에서 장바구니 버튼 클릭 시 정보 저장
 - 여러 상품 담기 가능
- 장바구니 구매
 - 선택한 상품만 골라 구매 가능
 - 결제 완료 시 판매 기록 저장 및 매출 통계 연동
 - 구매한 상품은 장바구니 리스트에서 삭제
→ 마이페이지 주문 목록에 기록



상품 구매

Goods Purchase

- 수량 선택 후 구매 클릭 → 구매 페이지로 이동
- 구매자 정보 및 총 가격 자동 출력
- 구매 확정 시 DB에 판매 기록 저장
- 마이페이지에서 구매 내역 확인 가능



리뷰

Review

- 리뷰 등록
 - 구매한 상품에 대해 별점과 코멘트 작성
 - 등록된 리뷰는 상품 상세 페이지에서 노출됨
- 리뷰 관리
 - 마이페이지에서 내가 작성한 리뷰 조회 및 삭제 가능

11. 주요 기능 및 상세 기능 설명

관리자 기능 상세설명

회원 관리 · 게시물 및 댓글 관리

Member Management

Content Moderation



회원 관리

Member Management

- 전체 회원 정보 조회 가능
- 회원 등급 변경
 - 일반회원 또는 관리자
- 일반 → 경고 → 블랙 (2회 경고 시 자동 블랙 처리)
- 강제 탈퇴(강퇴) 기능
- 블랙 처리된 회원은 로그인 불가
- 탈퇴/강제탈퇴 회원 리스트 확인 가능



게시물 및 댓글 관리

Content Moderation

- 커뮤니티 전체 게시글 조회
- 전체 댓글 리스트 조회 가능
- 게시글 클릭 시 해당 커뮤니티 상세 페이지로 이동

11. 주요 기능 및 상세 기능 설명

관리자 기능 상세설명

통계 조회 · 문의 채팅 · 광고 관리

Analytics & Statistics · Chat Support

Ad Management



통계 조회

Analytics & Statistics



문의 채팅

Chat Support



광고 관리

Ad Management

- Google Analytics(GA4) 연동 통계
 - 방문자 수, 재방문자 수, 페이지 체류 시간 등
- 내부 DB 기반 통계
 - 전체 회원 수, 전체 매출, 전체 상품 판매 수 조회
 - 일/주/월 기준 판매량 시각화 (그래프 형태로 출력)
- 문의 채팅 리스트
 - 최신순 문의 확인 가능
 - 새로운 문의 발생 시 상단 이동 및 미확인 메시지 배지 표시
- 문의 채팅방
 - 실시간 소켓 기반 채팅방 오픈
 - 채팅방 입장 시 읽음 처리 / 메시지 타임스탬프 기록
 - 본인 메시지는 오른쪽 정렬, 실시간 반응 표시
- 기존 광고 리스트 조회 가능
- 제목, 링크, 이미지, 태입(와이드: w, 하이트: h) 등록 가능
- 등록된 광고는 랜덤하게 태입별로 페이지에 출력됨
- 광고 클릭 시 외부 링크 이동 처리

12. API 명세서 - Swagger

Swagger 적용 개요 및 설정



Swagger 도입 목적

- API 자동 문서화로 협업 효율성 증가
- Postman 없이도 웹 기반 API 테스트 가능



기술 정보

- 사용 라이브러리
: springdoc-openapi-starter-webmvc-ui
- 접근경로
: <http://localhost:8888/swagger-ui/index.html>



주요 설정

- @OpenAPIDefinition 대신 OpenAPI Bean 방식으로 설정
- JWT 기반 인증을 위한 bearer 방식 토큰 보안 구성 포함
- SecurityFilterChain을 이용해 Swagger 테스트가 가능하도록 열어둠

```
SwaggerConfig.java
@Configuration
public class SwaggerConfig {

    @Bean
    public OpenAPI customOpenAPI() {
        return new OpenAPI()
            .info(new Info()
                .title("FitHub API 문서")
                .version("1.0")
                .description("FITHUB 백엔드 API 명세서입니다.")
                .contact(new Contact()
                    .name("FitHub")
                    .email("admin@fithub.com"))
                .license(new License()
                    .name("FitHub 프로젝트 정보")
                    .url("https://ubertech.notion.site/finalproject-schedule")))
            .servers(List.of(
                new Server().url("http://localhost:8888").description("로컬 서버")
            ))
            .components(new Components()
                .addSecuritySchemes("AccessToken", new SecurityScheme()
                    .type(SecurityScheme.Type.HTTP)
                    .scheme("bearer")
                    .bearerFormat("JWT")
                    .in(SecurityScheme.In.HEADER)
                    .name("Authorization")
                )
            )
            .addSecurityItem(new SecurityRequirement().addList("AccessToken"));
    }
}

SecurityConfig.java
@Configuration
@EnableWebSecurity
public class SecurityConfig {

    @Bean
    public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {
        http
            .csrf(csrf -> csrf.disable())
            .authorizeHttpRequests(auth -> auth
                .anyRequest().permitAll()
            );
        return http.build();
    }
}
```

12. API 명세서 - Swagger

API 기능 분류

❖ API 카테고리 구성

카테고리	설명
👤 회원 API	일반 회원가입, 로그인, 내 정보 수정, 탈퇴 등 회원 기능
🔗 소셜 로그인 API	Google/Kakao 소셜 로그인 및 회원가입 처리
✉️ 이메일 인증 API	이메일 인증번호 발송 및 확인 기능
⚒️ 관리자 API	상품/회원 관리, 통계 대시보드 등 운영자 전용 기능
⌚ 운동 기록 & 활동 API	운동 루틴 등록, 운동 기록 저장 등 My Fit 기능
👫 커뮤니티 API	게시글 작성, 댓글, 좋아요, 신고 등 커뮤니티 기능
🛒 마켓 API	상품 등록, 수정, 장바구니, 결제, 리뷰 기능
💬 문의 채팅 API	실시간 1:1 문의 채팅 기능



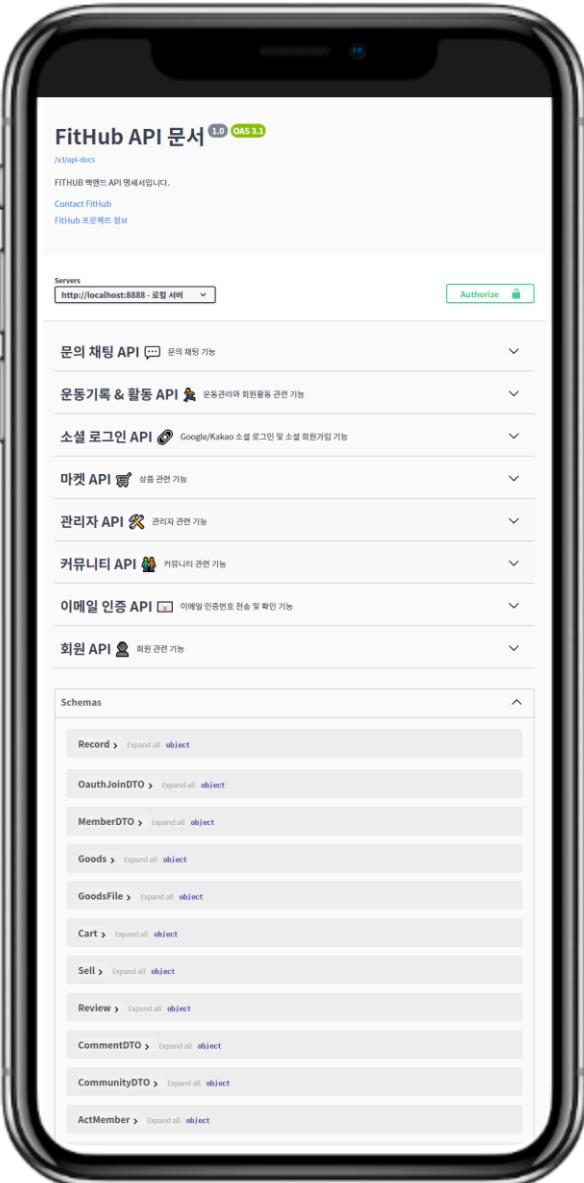
API 명세서 작성 & 활용

- 카테고리별로 컨트롤러를 분리하여 기능 단위 관리
→ 예 : MemberController, OauthController 등
- Swagger 어노테이션 활용
→ @Operation, @ApiResponses 등으로 명확한 문서화
- Try it out 기능을 통해 실시간 API 테스트 가능
- 팀 협업에서 API 명세서로 활용
→ 프론트, 백엔드, 기획 간 커뮤니케이션 도구 역할

```
@Operation(
    summary = "회원 탈퇴",
    description = "회원 ID를 통해 해당 회원을 탈퇴 처리합니다."
)
@ApiResponses({
    @ApiResponse(responseCode = "200", description = "회원 탈퇴 성공"),
    @ApiResponse(responseCode = "500", description = "회원 탈퇴 실패",
        content = @Content(
            mediaType = "text/plain",
            examples = @ExampleObject(value = "회원 탈퇴에 실패했습니다.")
        )
})
@DeleteMapping("/{memberId}")
```

12. API 명세서 - Swagger

Swagger UI 실제 화면



주요 특징

- Swagger UI를 통해 프로젝트 전체 API 구조와 흐름을 직관적으로 파악할 수 있습니다.
- 각 API별 “summary, description”, 요청 파라미터, 응답 형식을 명확히 확인할 수 있습니다.
- “Try it out” 기능을 통해 브라우저에서 직접 API 요청 테스트가 가능합니다.
- JWT 인증이 필요한 API는 “Authorize” 버튼을 통해 토큰 입력 후 테스트할 수 있습니다.
- 명확한 문서화를 통해 팀원 간 기능 공유와 협업 효율을 높일 수 있습니다.

The screenshot shows the '회원 API' (Member API) section of the Swagger UI. It lists various endpoints with their methods, URLs, and descriptions:

Method	Endpoint	Description
POST	/member	회원가입
PATCH	/member	회원 정보 수정
POST	/member/{memberId}/verify-password	비밀번호 확인
POST	/member/{memberId}/password/reset	비밀번호 찾기
POST	/member/auth/login	로그인
PATCH	/member/{memberId}/password	비밀번호 변경
GET	/member/{memberId}	회원 정보 조회
DELETE	/member/{memberId}	회원 탈퇴
GET	/member/recovery/id	아이디 찾기
GET	/member/exists/id	아이디 중복 확인
GET	/member/exists/email	이메일 중복 확인
GET	/member/auth/refresh	Access Token 재발급



API 문서는 다음 경로에서 확인할 수 있습니다.

- 개발 환경 : <http://localhost:8888/swagger-ui/index.html>
- 운영 환경 : <https://www.fithub.co.kr/swagger-ui/index.html>

트러블 슈팅 사례 ① - JWT 인증 오류



문제 상황

- 로그인 후 accessToken은 정상 발급되었으나, 페이지 새로고침 이후 일부 API 요청에서 401 Unauthorized 오류 발생.



원인 분석

- 로그인 시 accessToken은 axios 헤더에 저장되지만, 새로고침 시 헤더 초기화로 인증 정보가 사라짐
- Interceptor가 아닌 직접 세팅 방식이라, 앱이 다시 로드될 때 헤더가 복구되지 않음.



해결 방법

- accessToken은 recoil-persist로 Recoil 상태와 함께 localStorage에 저장되어 유지됨.
- 새로고침 후 App.js의 refreshLogin() 함수가 실행되어, localStorage에 저장된 refreshToken을 통해 새로운 accessToken을 재발급받고 헤더에 재설정함

```
// Inside `refreshLogin()` function (App.js)

axios.defaults.headers.common["Authorization"] = refreshToken;

axios
  .get(`${backServer}/member/auth/refresh`)
  .then((res) => {
    setMemberInfo(res.data);
    axios.defaults.headers.common["Authorization"] = res.data.accessToken;
    localStorage.setItem("refreshToken", res.data.refreshToken);
  })
  .catch((error) => {
    setMemberInfo(null);
    delete axios.defaults.headers.common["Authorization"];
    window.localStorage.removeItem("refreshToken");
  });
}
```



개선 결과 및 인사이트

- 새로고침 이후에도 accessToken이 자동 재발급되고 인증이 유지됨
- Interceptor 없이 Axios 헤더를 수동 관리하는 경우, 토큰 재설정 흐름(App 단 진입 시점)을 명확히 설계하는 것이 중요
- recoil-persist + refreshToken 재요청 흐름은 실질적인 자동 로그인 유지 전략으로 안정적

트러블 슈팅 사례 ② - 프로필 이미지 업로드 및 삭제 처리 문제



문제 상황

- 마이페이지에서 프로필 이미지를 삭제하거나 새 이미지로 교체한 후, 프론트엔드에서는 정상 반영되어도 백엔드에서는 기존 이미지가 남아있는 등 DB, 파일, UI 상태 불일치가 발생함.



원인 분석

- 이미지 삭제나 재등록 시 "null" 값을 백엔드에 명확히 전달하지 않으면, 기존 이미지가 파일/DB에서 제거되지 않고 유지됨
- 삭제 요청과 단순 누락이 명확히 구분되지 않는 조건 처리로 인해 이미지 상태 불일치 발생



해결 방법

- 삭제 시 memberThumb = "null" 설정하여 삭제 의도 명확화
- 이미지 변경 시 기존 썸네일을 먼저 null 처리하고 새 이미지 등록
- FormData로 조건 분기 처리하여 백엔드로 상황 명확 전달
- 백엔드에서 "null" 값인 경우 파일 삭제 로직 추가

```
// Inside `handleUpdate()` function (MemberInfo.js)
if (thumbnailFile) {
    formData.append("thumbnail", thumbnailFile);
} else if (
    !thumbnailFile &&
    member.memberThumb === null &&
    loginMember.memberThumb
) {
    formData.append("memberThumb", "null");
}

// Inside `updateMember()` function (MemberController.java)
if ("null".equals(member.getMemberThumb())) {
    if (origin.getMemberThumb() != null && !origin.getMemberThumb().isEmpty()) {
        File file = new File(savepath + origin.getMemberThumb());
        if (file.exists()) file.delete();
    }
    member.setMemberThumb(null);
}
```



개선 결과 및 인사이트

- "null" 전달 분기로 이미지 삭제·재등록 시 UI, DB, 파일 간 불일치 해결
- 삭제, 재등록, 유지 흐름을 명확히 구분해 모든 예외 상황 안정 대응
- FormData 요청에서 상태별 처리 분기가 중요함을 다시 확인

서비스 동작 화면



메인페이지

운동 루틴 기록부터 커뮤니티 소통, 마켓 구매까지
한눈에 시작할 수 있는 메인 허브 역할의 홈 화면

“

**핵심 기능을 바로 확인할 수 있는 인트로 구조로
사용자의 탐색 흐름을 유도합니다.**

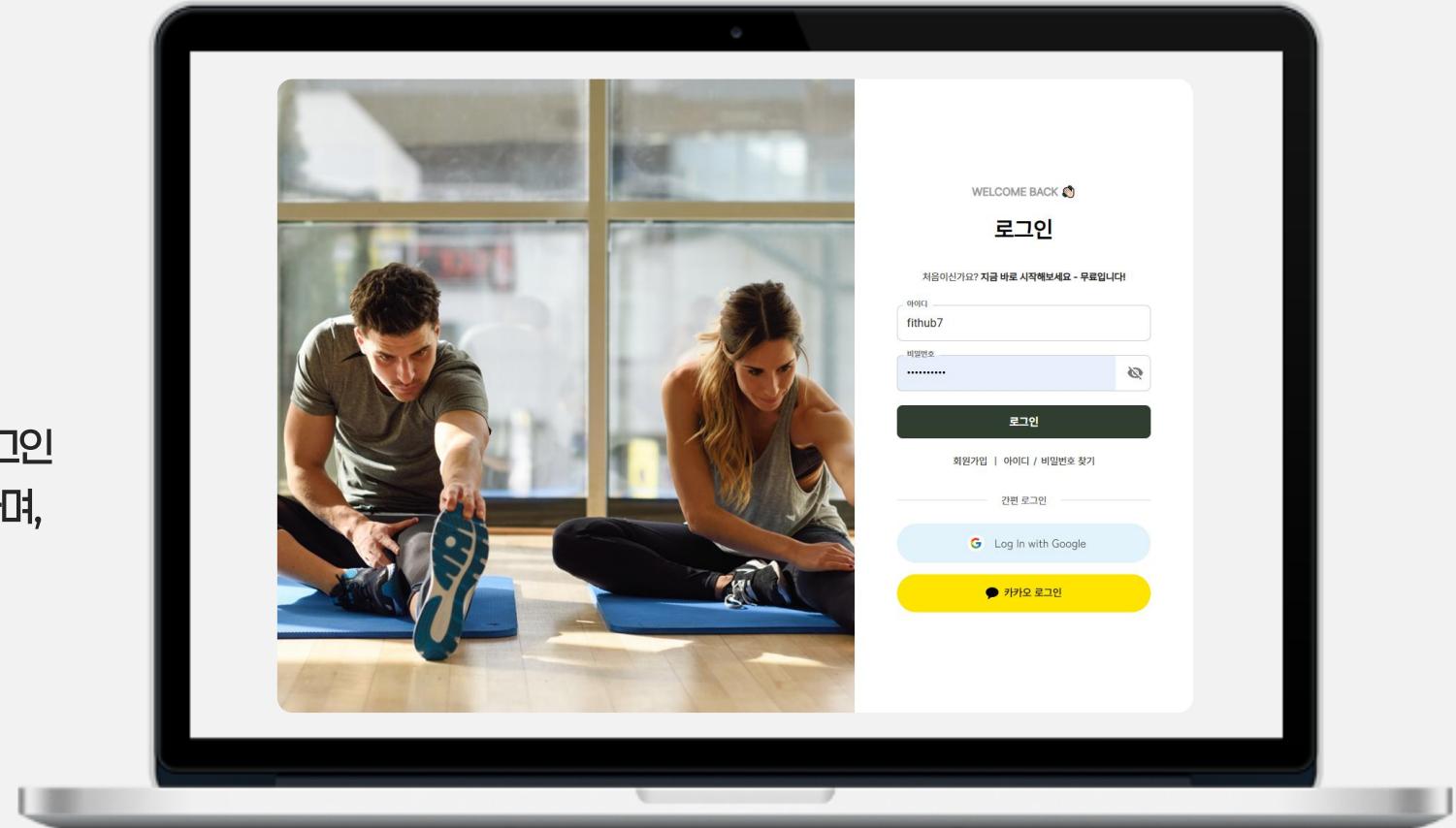


14. 실제 구현 화면

서비스 동작 화면



일반 로그인뿐만 아니라 Google, Kakao 소셜 로그인
까지 지원하여 다양한 방식으로 손쉽게 접속 가능하며,
ID/PW 찾기 및 회원가입 연동 기능 포함



14. 실제 구현 화면

서비스 동작 화면

이용약관 동의

모든 약관을 확인하고 전체 동의합니다.

이용약관에 동의합니다. (필수)

제 1 조 (목적)
본 약관은 회사(이하 "회사")가 제공하는 FitHub 서비스(이하 "서비스")의 이용 조건 및 절차, 권리의 무 및 책임사항 등을 규정함을 목적으로 합니다.

제 2 조 (용어의 정의)
1. "회원"이라 함은 본 약관에 동의하고 서비스를 이용하는 자를 말합니다.
2. "콘텐츠"란 회사가 서비스에서 제공하는 정보 및 자료를 의미합니다.

개인정보 수집 및 이용에 동의합니다. (필수)

회사는 회원가입, 서비스 제공 및 고객분의 응대 등을 위해 아래와 같은 최소한의 개인정보를 수집·이용합니다.

1. 수집 항목
- 필수 항목: 아이디, 비밀번호, 이름, 이메일, 휴대전화번호, 주소
- 선택 항목: 프로필 사진, 자기소개

2. 이용 목적

[이전](#) [다음](#)

소셜 회원 추가 정보 입력

아이디
fithubkakao
① 사용 가능한 아이디입니다.

이름
핏소설
① 사용 가능한 아이디입니다.

전화번호
010-7777-7777
① 유통번호 형식입니다.

주소
서울 영등포구 선유동2로 57
주소 검색

상세 주소
19층

[가입 완료](#)

가입을 원하지 않으신가요? [로그인 페이지로 돌아가기](#)

회원가입

아이디
fithub77
① 사용 가능한 아이디입니다.

비밀번호
.....
② 비밀번호 확인

비밀번호 확인
.....
③ 비밀번호가 일치합니다.

보안 수준: 강

이름
핏허브

이메일 아이디
fithub77
도메인 선택
naver.com 인증번호 반기

인증번호
556664
④ 인증번호가 확인되었습니다.

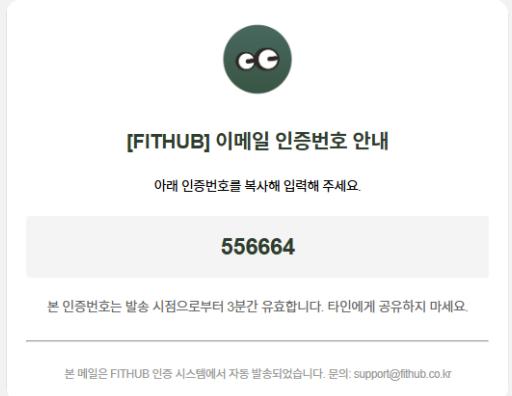
주소
서울 영등포구 선유동2로 57
주소 검색

상세 주소
19층

전화번호
010-7777-7777
⑤ 유통번호 형식입니다.

[회원가입](#)

이미 회원이신가요? [로그인](#)



회원가입

이메일 인증, 주소 검색, 전화번호 하이픈 자동 처리 등
다양한 유효성 검사와 사용자 편의 기능을 갖춘 단계별 회원가입 시스템

14. 실제 구현 화면

서비스 동작 화면

아이디 / 비밀번호 찾기

아이디 찾기 비밀번호 찾기

이름
이메일

아이디 찾기

아이디를 확인하셨나요? [로그인](#)

아이디 찾기 결과

회원님의 아이디는 다음과 같습니다:

7777tmdgus
777tmdgus2
77777tmdgus

확인

아이디 / 비밀번호 찾기

아이디 찾기 비밀번호 찾기

아이디
이메일

임시 비밀번호 받기

임시 비밀번호를 받으셨나요? [로그인](#)

[FITHUB] 임시 비밀번호 안내

아래 임시 비밀번호를 사용하여 로그인한 뒤 반드시 비밀번호를 변경해 주세요.

71b9cd9a55

보안을 위해 해당 비밀번호는 반드시 변경해 주세요.

본 메일은 FITHUB 시스템에서 자동 발송되었습니다. 문의: support@fithub.co.kr



아이디·비밀번호 찾기

입력한 이름·이메일·아이디 정보를 기반으로
아이디 반환 또는 임시비밀번호 메일 발송 기능 제공

14. 실제 구현 화면

서비스 동작 화면

내 정보



이미지 등록 이미지 삭제

아이디
fithubkakao

이름
핏소셜

전화번호
010-777-7777

주소
서울 영등포구 선유동2로 57

주소 검색

상세주소
19층

회원등급
일반 회원

프로필(자기소개)
안녕하세요. 핏소셜입니다.
저의 MBTI는 INFJ 입니다. 잘 부탁드립니다!

정보수정 회원탈퇴

비밀번호 변경



현재 비밀번호 확인
새 비밀번호 설정
변경 완료

현재 비밀번호
.....

비밀번호 확인
.....

비밀번호 확인
비밀번호가 일치합니다.
보안 수준: 강한

비밀번호 확인

비밀번호 변경



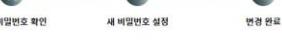
현재 비밀번호 확인
새 비밀번호 설정
변경 완료

새 비밀번호
.....

비밀번호 확인
.....

비밀번호 변경
비밀번호가 일치합니다.
보안 수준: 강한

비밀번호 변경



현재 비밀번호 확인
새 비밀번호 설정
변경 완료

비밀번호가 성공적으로 변경되었습니다.



마이페이지

이름, 연락처, 주소, 자기소개 등 정보 수정과
프로필 이미지 변경, 회원 탈퇴 기능을 제공하는 사용자 정보 관리 페이지

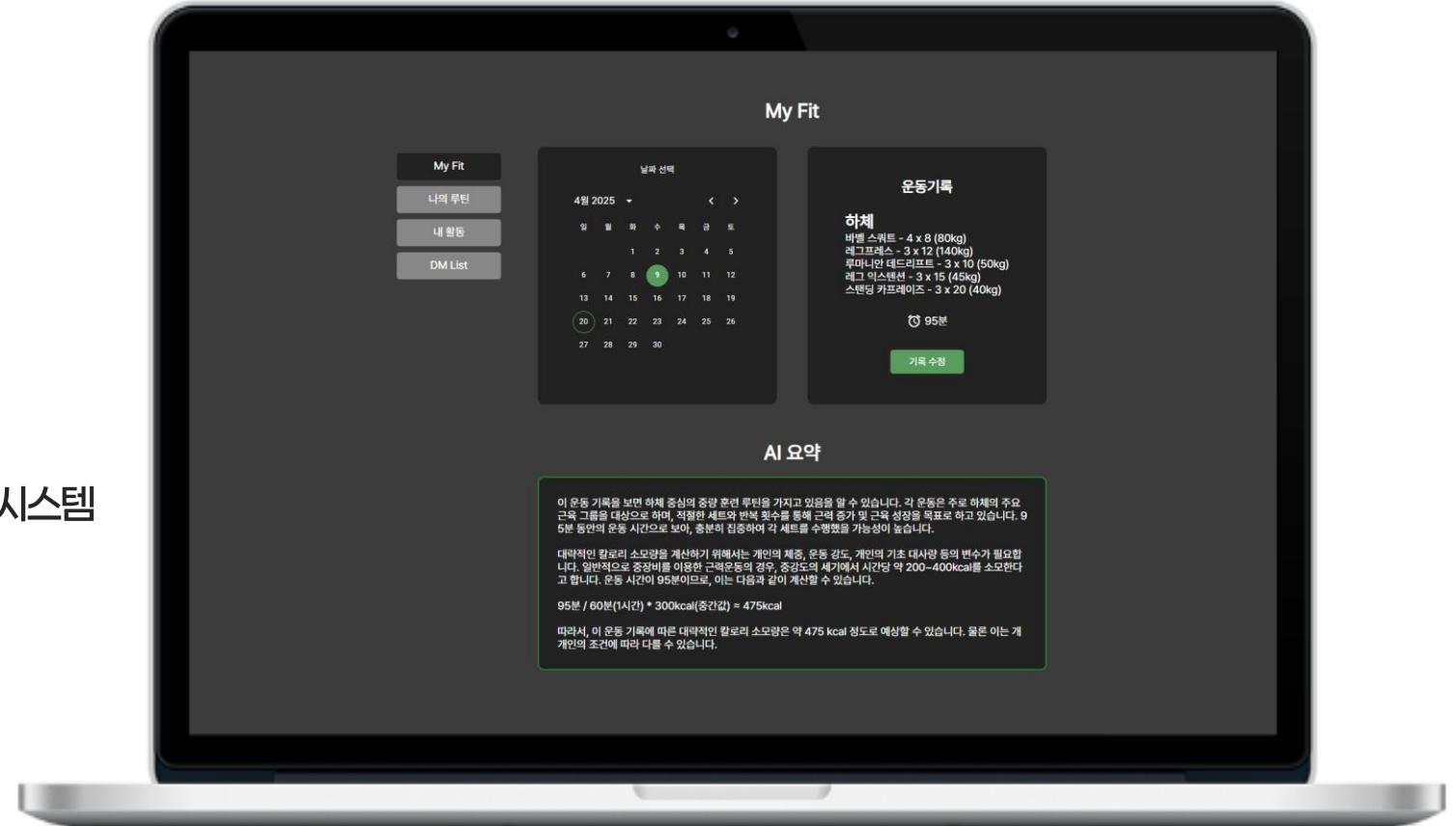
14. 실제 구현 화면

서비스 동작 화면



루틴 등록부터 운동 기록, AI 분석까지

운동 흐름에 최적화된 달력 기반 자기주도형 운동 관리 시스템



14. 실제 구현 화면

서비스 동작 화면

나의 루틴

나의 루틴

내 활동

DM List

나의 루틴

요일별로 나만의 루틴을 작성해보세요!

루틴

어깨 + 복부

- 비幡 슬더프레스 - 4 x 10 (35kg)
- 사이드 레티컬 레이즈 - 3 x 15 (7kg 양방)
- 리어 델트 폴라이 (마신) - 3 x 12 (30kg)
- 크런치 마신 - 3 x 15 (25kg)
- 햄엄 레그레이즈 - 3 x 최대반복 (엔딩)

루틴 설정

운동 기록

2025년 4월 10일 (목)

시작 시간 ~ 종료 시간

오전 12:00 ~ 오후 12:00

운동 내용

Normal

기록을 입력하세요...

기록하기



나의 루틴 & 운동 기록

사용자가 직접 등록한 운동 루틴을 기반으로
날짜별 수행 기록을 입력하고 수정할 수 있는 통합 운동 관리 시스템

14. 실제 구현 화면

서비스 동작 화면



활동로그

게시물, 팔로우, 프로필 편집, My Fit 통계 등

서비스 내 모든 사용자 활동 내역을 한 눈에 확인 가능한 통합 히스토리 기능 제공

14. 실제 구현 화면

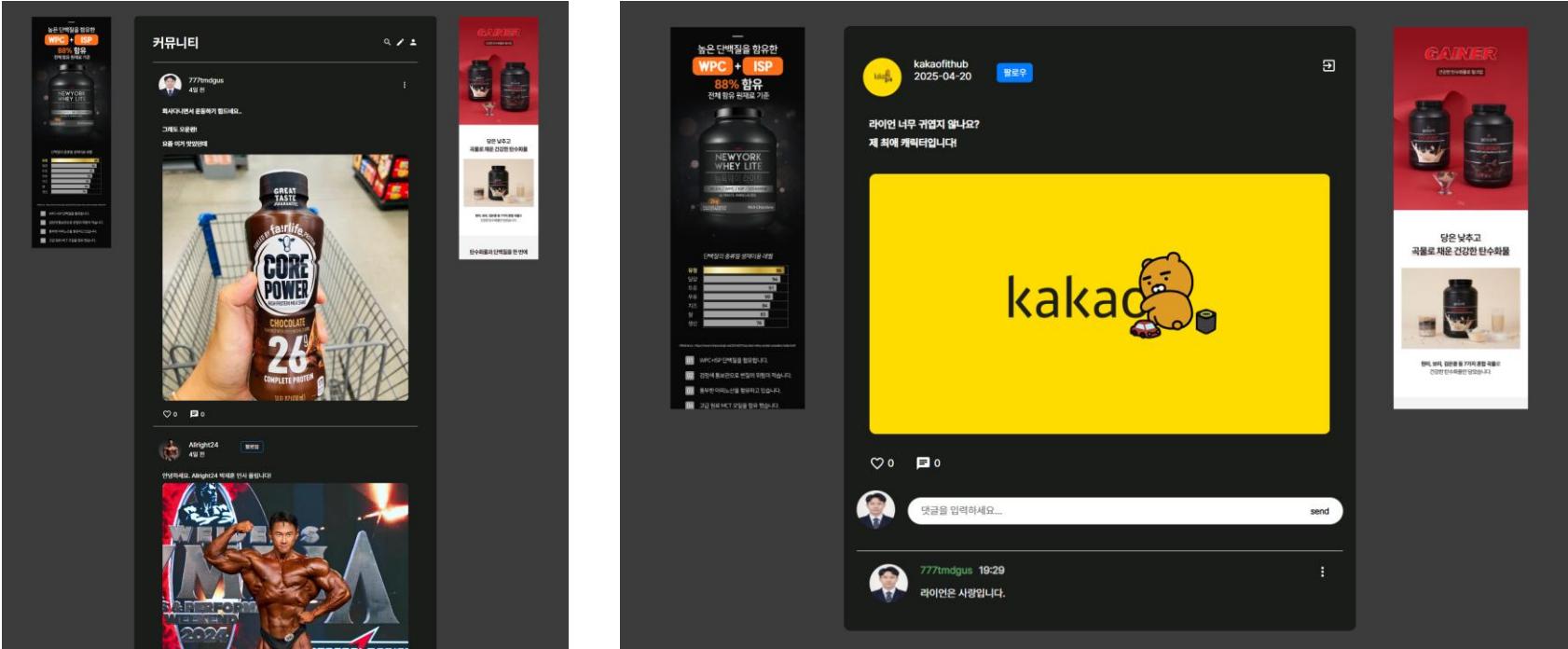
서비스 동작 화면



회원간 1:1 대화를 위한 실시간 디렉트 메시지 기능으로
운동 정보 공유 및 소통을 돋는 커뮤니티형 커뮤니케이션 시스템 제공

14. 실제 구현 화면

서비스 동작 화면

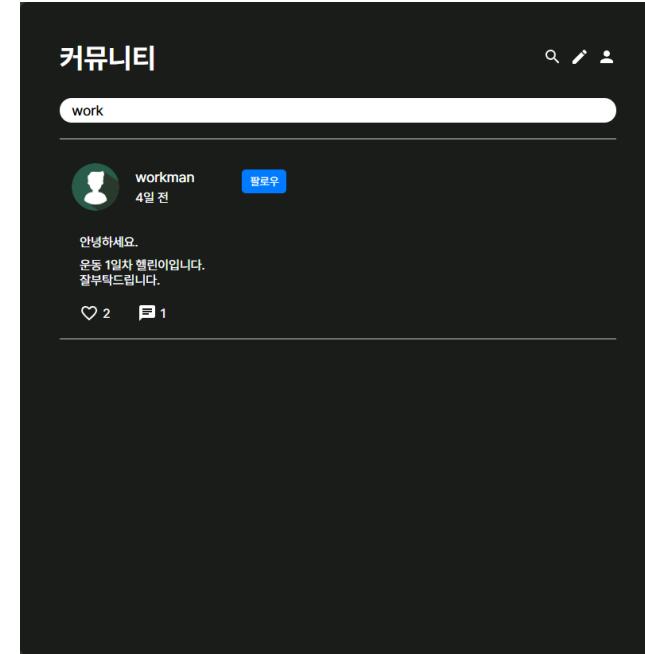
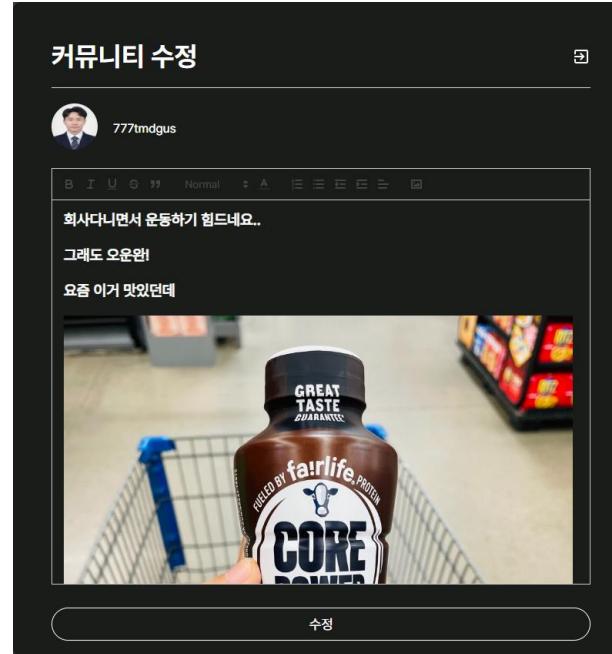
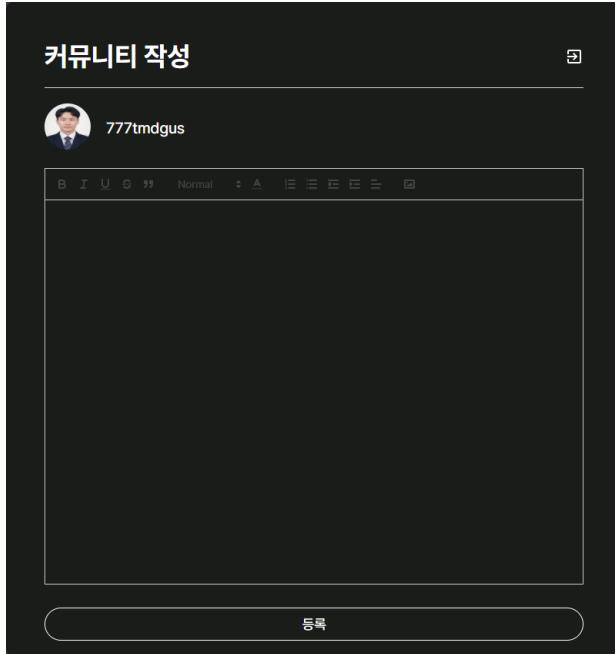


커뮤니티 피드 & 상세보기

게시판 형식으로 다양한 운동 후기를 열람하고,
게시글 클릭 시 이미지, 내용, 댓글이 포함된 상세화면 제공

14. 실제 구현 화면

서비스 동작 화면



커뮤니티 게시글 CRUD

운동 경험과 팁을 자유롭게 공유할 수 있으며
게시글 수정 및 삭제, 키워드 검색을 통한 빠른 정보 접근 가능

14. 실제 구현 화면

서비스 동작 화면



팔로잉 & 팔로우

관심 있는 사용자를 팔로우하여
해당 사용자의 커뮤니티 활동 및 게시글을 쉽게 확인 가능

서비스 동작 화면

FIT MARKET

AWAKEN YOUR MOTION.
GROGGY

모두 보조비 비단판 스포츠웨어(남) 스포츠웨어(여) 운동기구

최신순

파워온 에어 스텁퍼 천국의계단 99,000 원

이프노스 여성용 셀리나 레깅스 13,000 원

위즈아이 스포츠 트레이닝 세트 21,000 원

센트룸 / 펜 윈드스 29,000 원

뉴욕웨이 라이트 56,000 원

상품 이미지

상세설명

구매하기

파워온 에어 스텁퍼 천국의계단

수량: 1개

손잡이 유무: 손잡이 있음

운동 부위: 다리등

지압가능여부: 지압 가능

최대무게: 150kg

개당 가격 : 99,000원

- 1 +

장바구니

리뷰

배송/결제

아이보리

파워온 에어 스텁퍼 천국의계단

b451r591-23001

재질

크기, 중량

제품구성

장소 + 시간 에



마켓상품 목록 & 상세조회

운동 관련 상품을 한눈에 확인할 수 있는 마켓 리스트와
상품별 이미지, 설명, 가격 정보를 확인할 수 있는 상세페이지 제공

14. 실제 구현 화면

서비스 동작 화면

The image displays three screenshots of a web-based product management system. The first screenshot shows a camera icon and a form for inputting product details like name, category, and price. The second screenshot shows a woman using a step machine, its product details, and a 'Delete' confirmation dialog. The third screenshot shows a grid of products with a 'Delete' confirmation dialog overlaid.

Screenshot 1: Product Input Form

상품 제목을 입력하세요			
상품 설명			
Normal	B	I	U
카테고리를 선택하세요			
총 가격: <input type="text"/> 원			
상품 등록하기			

Screenshot 2: Product Details and Delete Confirmation

파워온 에어 스텝피 천국의계단			
상품 설명			
Normal	B	I	U
<ul style="list-style-type: none">수량: 1개손잡이 유무: 손잡이 있음운동 부위: 다리등지급가능여부: 지급 가능최대부피: 150kg			
운동기구			
개당 가격: <input type="text"/> 원			
상품 수정하기			

Screenshot 3: Product Grid and Delete Confirmation

최신순			
상품 이미지	상품명	가격	상세보기
	파워온 에어 스텝피 천국의계단	99,000 원	X
	이프로 헬스케어	12,000 원	X
	센트룸 면 음모스	29,000 원	X
	뉴욕웨이 라이트	58,000 원	X

삭제하시겠습니까?
이 작업은 되돌릴 수 없습니다.
예, 삭제합니다. **아니요, 취소합니다.**



마켓상품 CRUD

관리자는 상품 정보를 직접 등록하고,
가격이나 설명 등을 수정하거나 삭제할 수 있는 마켓 관리 기능 제공

14. 실제 구현 화면

서비스 동작 화면



FIT 주문/결제

구매자 정보

이름:	최승현
연락처:	010-5012-7769

받는 사람 정보

이름:	최승현
연락처:	010-5012-7769
주소:	서울 강서구 우장산로4길

배송 정보

상품명	위즈아이 스포츠 트레이닝 세트 / 수량: 1
-----	--------------------------

결제 정보(3만원 이상 구입 무효)

상품가격:	21000 원
배송비:	3000 원
결제 금액:	24000 원

결제하기

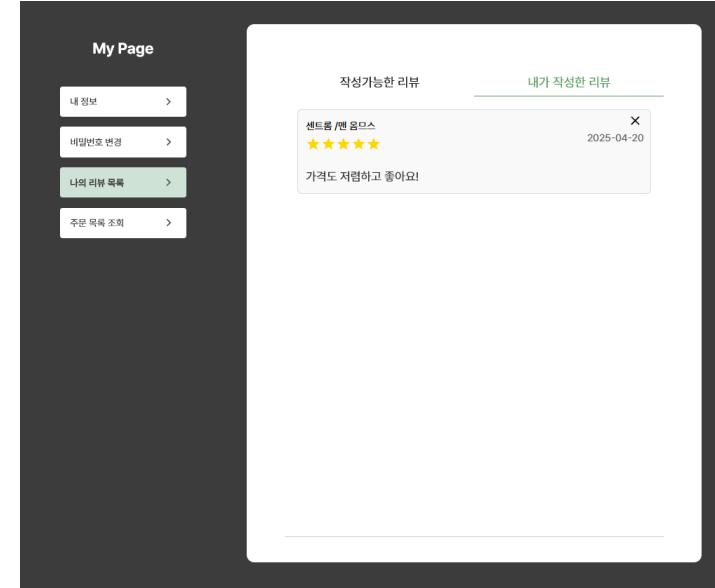
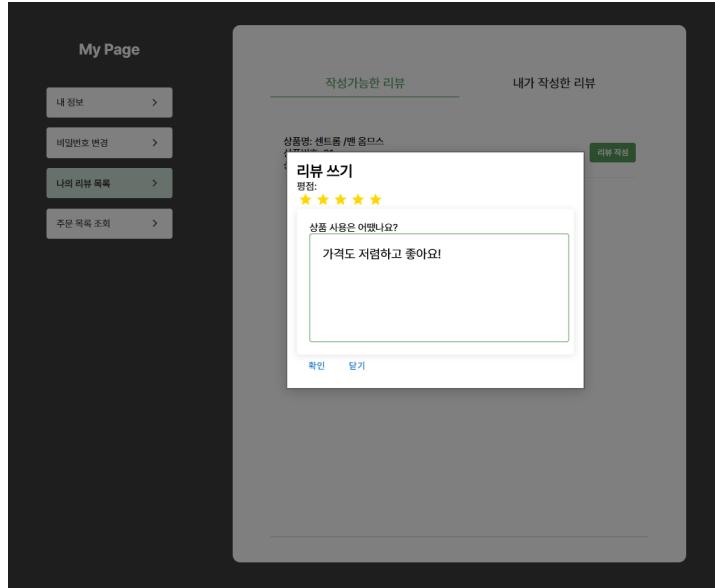
My Page 화면은 개인 정보 관리와 주문 조회 기능을 제공하는 메인 메뉴입니다. 주문/배송 리스트 화면은 최근 주문 내역과 배송 상태를 확인할 수 있는 목록입니다. 세트콤 / 맨 음마스라는 상점에서 1회 구매한 2개의 상품에 대한 정보가 표시됩니다. 각 상품마다 처리일시, 배송단계, 처리장소가 세부적으로 나열되어 있으며, 배송 조회 버튼이 있습니다.

장바구니 & 구매

원하는 상품을 장바구니에 담아 한번에 확인 및 결제 가능하며,
구매 내역은 마이페이지를 통해 조회 가능

14. 실제 구현 화면

서비스 동작 화면



This screenshot shows a product detail page for '이프노스 여성용 셀리나 레깅스'. It includes a large image of a person wearing the leggings, product details like size L, color black, and material spandex, and a price of 13,000원. A green '장바구니' (Cart) button and a green '구매하기' (Buy) button are at the bottom. Below the product info, there's a review section with a 5-star rating, the user 'user10', and the text '또 사고 싶어요' (I want to buy again). The footer shows tabs for '상품정보', '리뷰', '배송/결제', and '반품/교환'.



상품에 대한 별점 및 후기 작성 가능,
사용자는 자신의 리뷰를 자유롭게 수정 및 삭제 가능

14. 실제 구현 화면

서비스 동작 화면

회원 관리		게시글 관리					
프로필	아이디	가입일	이메일	전화번호	광고처리	회원등급	강퇴
	777tmdgus	2025-04-16	777tmdgus@naver.com	010-5012-7769	일반 ▾	관리자 ▾	
	earthCrash	2025-04-16	aithoodgang@gmail.com	010-0101-2929	일반 ▾	정회원 ▾	
	ComCrash_delete...	2025-04-16	aithoodgang@gmail.com	010-0101-2929	블랙 ▾	정회원 ▾	
	Allright24	2025-04-16	skgksql345@gmail.com	010-1111-1111	일반 ▾	관리자 ▾	
	kakaoofithub	2025-04-16	777tmdgus@kakao.com	010-5012-7769	일반 ▾	정회원 ▾	

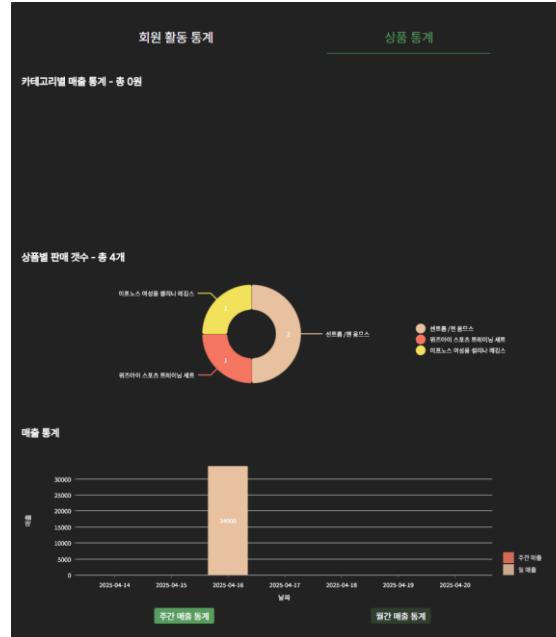
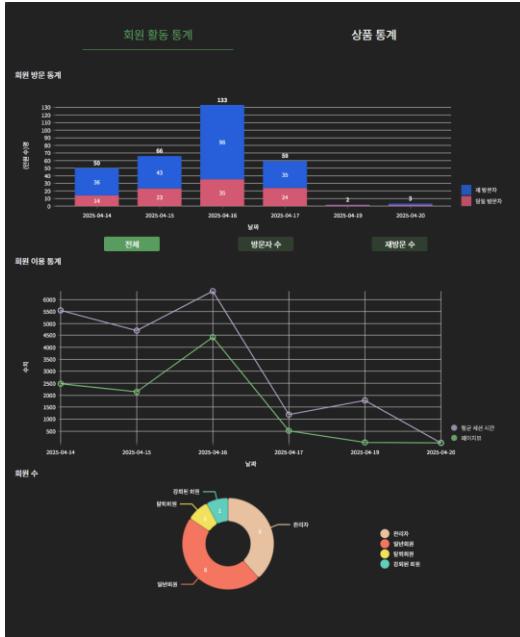


회원·게시글 관리

회원 등급 변경, 경고, 강퇴, 탈퇴 회원 조회 등 플랫폼 운영을 위한
기본 회원 관리 기능과 커뮤니티 게시글/댓글 모니터링 및 제재 기능 제공

14. 실제 구현 화면

서비스 동작 확인



현재 진행 중인 광고						
이름	이미지	주소	리스트	개시일	타입	삭제
マイプロテ린 광고	프로틴5 와이드.avif	https://www.myprotein.co.kr/		2025-04-16	w	<input checked="" type="checkbox"/>
マイプロテ린 광고 2	프로틴4 와이드.avif	https://www.myprotein.co.kr/		2025-04-16	w	<input checked="" type="checkbox"/>
뉴욕웨이 광고	프로틴1 세로.png	https://nutritionfactory.co.kr/prod...		2025-04-16	h	<input checked="" type="checkbox"/>
삼대오백 광고	프로틴2 세로.jpg	https://samdae500.com/?srstid=...		2025-04-16	h	<input checked="" type="checkbox"/>

광고 추가

 +이미지를 추가해주세요

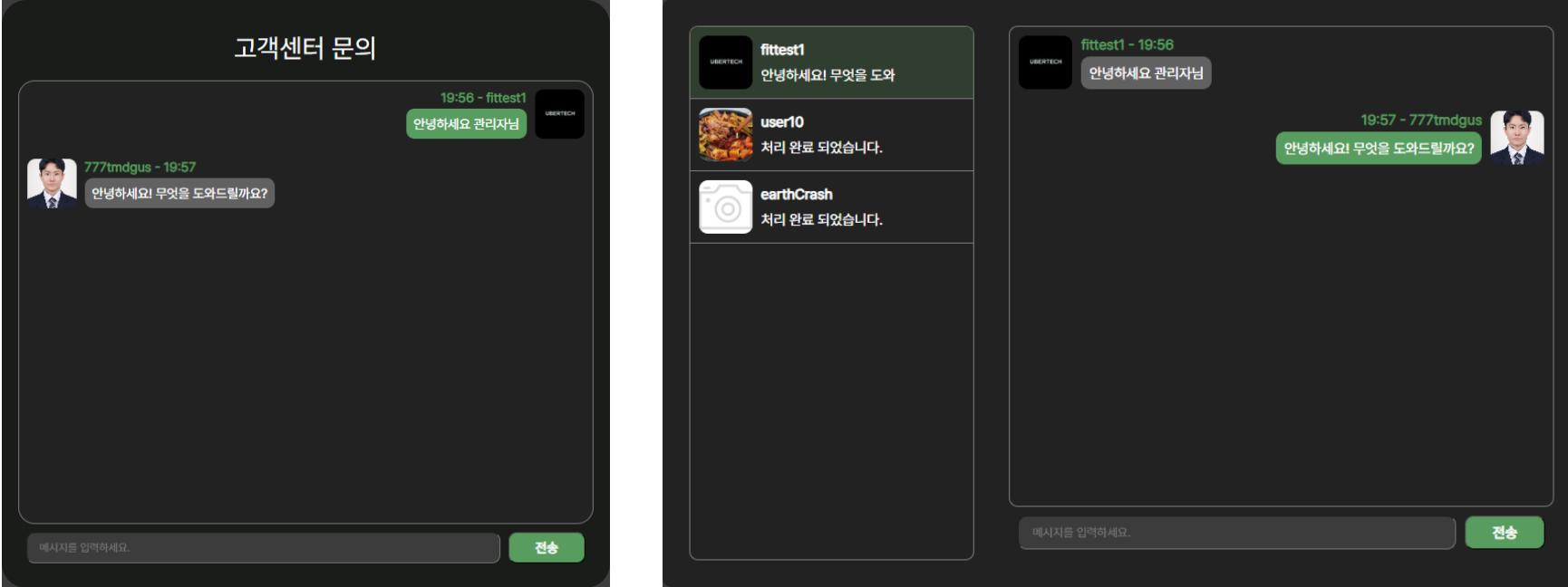


통계조회 & 광고 관리

회원활동, 운동루틴/기록, 마켓매출 등의 데이터를 기반으로 내부 DB 및 외부 도구를 활용한 시각화통계 제공
광고 이미지를 관리자페이지에서 직접 등록할 수 있으며, 등록된 광고는 랜덤 방식으로 사용자화면에 노출됨

14. 실제 구현 화면

서비스 동작 화면



실시간 문의 채팅

사용자 문의에 관리자 실시간 응답이 가능한
WebSocket 기반 1:1 문의 채팅 시스템 제공

프로젝트 회고



팀 회고

- 협업 방식

팀원별 브랜치를 만들어 기능 단위로 개발한 뒤,
main 브랜치에 직접 머지하여 빠른 피드백과
유연한 개발 흐름을 유지함

- 협력 경험

각자의 역할에 집중하면서도 GitHub, Notion을
통해 작업 상황을 공유하고 문제 발생 시 빠르게 조율함

- 잘한 점

명확한 역할 분담, 일정 관리의 유연성, 트러블 발생 시
유연한 대응

- 아쉬운 점

개발 후반부로 갈수록 일정이 다소 촉박해져 테스트와
UI 세부 조율 시간이 부족했음



개인 회고

- 담당 역할

로그인, 회원가입, JWT 인증 시스템, 소셜 로그인,
마이페이지, 공동 레이아웃 구성, 메인페이지 설계,
로고 디자인, 프로젝트 문서화

- 기술 성장

JWT 인증 흐름과 소셜 로그인 처리, FormData 기반
이미지 업로드, Axios 헤더 처리 등 실무에 가까운 경험
다수 확보

- 트러블 슈팅 경험

accessToken 재발급 문제, 이미지 삭제 처리 조건 등
주요 이슈들을 직접 해결하며 문제 분석 및 대응 능력 향상

- 느낀 점

사용자 입장에서의 설계 고민, 구조적인 프론트-백엔드
연동 경험을 통해 더 나은 사용자 흐름을 고려하는 개발
자로 한 걸음 성장할 수 있었던 프로젝트였음

**기능 하나하나에 고민을 담고,
사용자 경험을 생각하며 개발했습니다.**

fitHub 프로젝트는 단순한 구현을 넘어,
협업과 실무에 가까운 문제 해결을 통해
한단계 더 성장한 개발자로 나아갈 수 있었던 경험이었습니다.

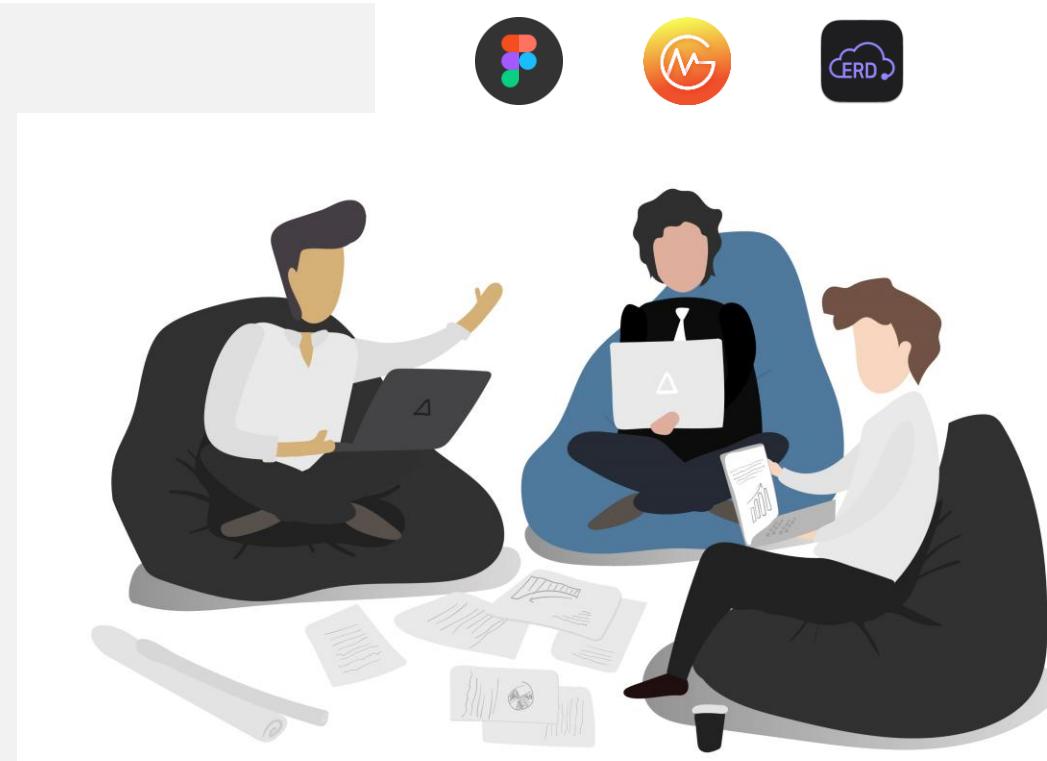
감사합니다.



[GitHub 저장소 바로가기](#)



[포트폴리오 보기](#)



777tmdgus@naver.com