

Fitness In Service

헬스장/필라테스/GX 센터 통합 관리 플랫폼

Spring Boot 멀티모듈 백엔드, 관리자 웹, 회원 앱으로 구성된 센터 운영 플랫폼입니다. 저는 백엔드 API(BO/MO)와 배치 개발, 그리고 기획 및 설계를 담당했습니다 (화면 개발은 지원).

팀 구성 — 백엔드 개발 1명(본인), 프론트엔드 개발 1명, 기획 및 운영지원 2명, 총 4명

총 4개 센터 이용 중, 이용자 약 70명, 회원 수 약 900명

beta.fitness-trainer-assistant.ai.kr

실제 배포된 베타 서비스, 직접 확인 가능

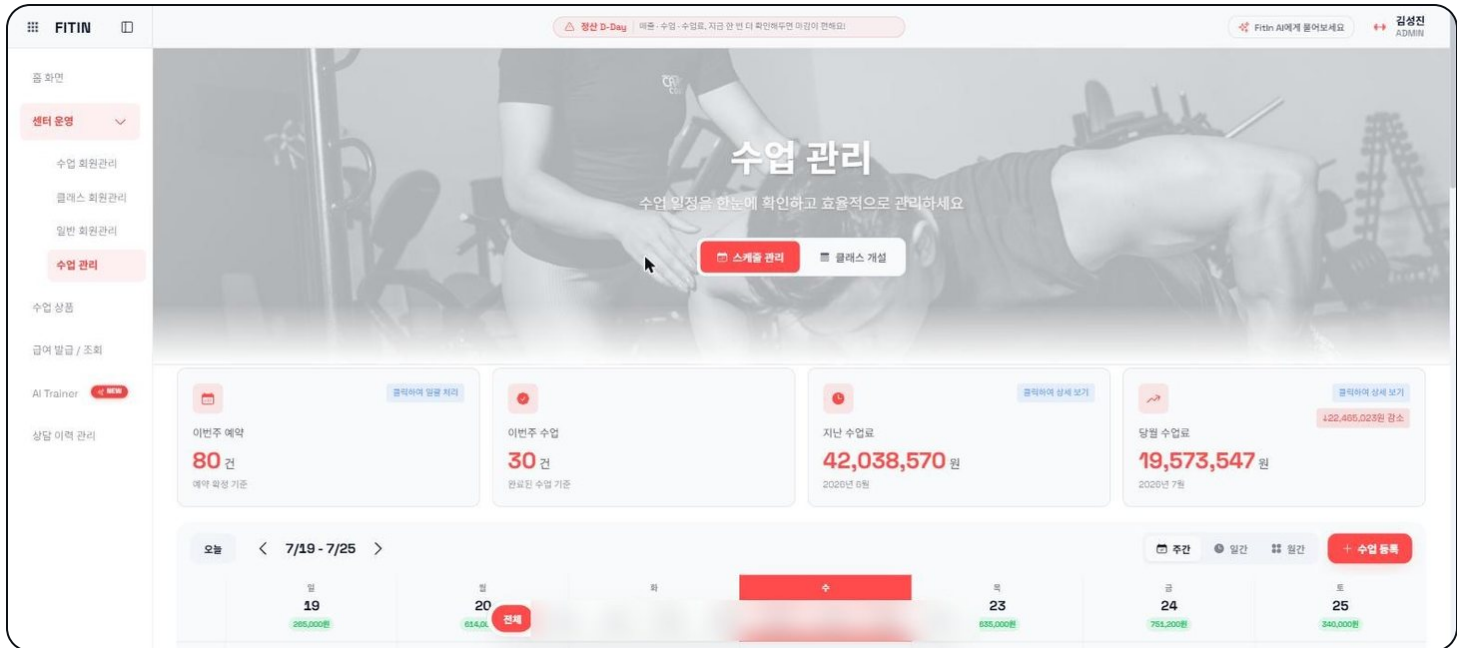
[서비스 열기 ↗](#)

Java 17, Spring Boot 4

MyBatis, MariaDB, Redis

AWS, Docker

Kakao OAuth, JWT, SSE



관리자 백오피스, 수업 관리 — 주간 달력 + 예약/수업/수업료 지표 (실제 배포 화면)

PROCESS

일하는 방식 — 요구사항 분석부터 운영까지

단계를 나눠 진행하고, 각 단계의 규칙과 결정을 문서로 남깁니다.

01

요구사항 분석

정책 해석, 케이스 분기, 데이터 모델링

02

설계

API 스펙, 레이어/경계, 트랜잭션 정책

03

개발

도메인별 구현, 사이드 이펙트 검토

04

QA/검증

모듈 빌드 게이트, 케이스 점검

05

배포/운영

Docker 배포, 로그 모니터링/대응

애매하면 확인 정책 애매하면 혼자 판단하지 않고 팀원들과 함께 내용 확인

변경 공유 API 스펙 바뀌면 API정의서 수정 및 프론트 개발자한테 수정사항 공유

배포 전 확인 배포 전엔 단위테스트 및 사이드이펙트 확인

원인부터 파악 장애 생기면 원인부터 잡고 재발하지 않도록 관리

DESIGN

설계 원칙

DDD 기반 멀티모듈

공통 도메인은 `Core`, 관리자(BO)/회원(MO)은 별도 모듈. 공통 코드는 반드시 Core에 두어 경계를 강제.

계층 & 경계 분리

Controller → Service → Mapper → Entity. `DTO` 는 API 계층, `VO` 는 Service↔Mapper — 도메인이 DTO에 의존하지 않게.

응답/예외 통일

모든 응답은 `ApiResponse<T>`, 예외는 `BusinessException` + 에러코드 enum으로 일관되게 처리.

정합성/추적성 우선

상수 중앙화로 매직리터럴 제거, TraceId 필터+AOP로 요청 전 구간 로깅해 장애를 추적 가능하게.

백엔드 설계 포인트 — 배경, 설계, 이유

주어진 정책을 백엔드에서 정합성 있게 풀어낸 설계입니다.

상품, 멤버십, 수업 3계층 — 도메인/DB 설계

배경	회원이 실제 소유하는 건 상품이 아니라 차감/예약의 기준이 되는 권리(멤버십) 이고, 개인+그룹이 섞인 패키지 상품 도 팔아야 했습니다.
설계	판매 단위(Product)/소유/차감 단위(Membership)/운영 단위(Class) 를 엔티티로 분리해 스키마에 반영 (상품 → 멤버십 1:N , 멤버십 → 수업 N:1), 결제 성공 시 상품을 unpack 해 멤버십을 N개 생성하도록 구현.
이유	패키지 상품을 단품 로직 변경 없이 확장할 수 있고, 화면별 표기 소스(스케줄=수업명, 보유함=멤버십명)를 분리해 혼선을 없앴습니다.

수업 취소/폐강의 차감 복구 정합성

배경	취소/폐강은 멤버십 차감 복구/정원/강사 급여 에 연쇄로 영향을 주고, 취소 권한이 수업 유형/마감기한에 따라 달라지는 정책이 있었습니다.
설계	차감 복구는 트랜잭션으로 즉시 반영 , 폐강은 배치 스케줄러가 최소 수강 인원 미달 시 자동 취소 → 전원 예약 취소 + 차감 복구 + 알림 . 권한 정책은 수업 유형/마감 전후로 백엔드에서 분기 처리.
이유	돈/자원에 직결되는 상태(차감/정원)는 정합성을 최우선으로 두고, 권한은 운영 실수를 구조로 막도록 분기했습니다.

PROBLEM SOLVING

문제 → 원인 → 해결 → 결과

운영하며 실제로 부딪힌 문제들. 코드 레벨 상세는 면접에서 설명 가능합니다.

알림/이력이 트랜잭션에 잘못 묶이는 문제 — 트랜잭션/배치

문제	알림 발송이나 실패 이력이 본 트랜잭션에 묶이면, 롤백 시 이력이 사라지거나 커밋 전에 알림이 먼저 나갔습니다. 배치에 서는 한 건 오류가 전체를 롤백시켰습니다.
원인	부가 작업(알림/이력)과 핵심 처리가 같은 트랜잭션 경계 안에 있었음.
해결	실패 이력/주문 만료 배치는 <code>REQUIRES_NEW</code> 로 건별 독립 트랜잭션 , 알림은 <code>@Async</code> + 커밋 후 콜백(<code>afterCommit</code>)으로 "커밋 성공 이후에만" 발송.
결과	부분 실패가 전체를 되돌리지 않고, 롤백된 거래의 알림이 잘못 나가는 일이 사라짐.

도메인 구분 한계 → 패키지 구조로 확장 — 확장성/운영안정

문제	수업을 클래스 유형으로 도메인만 구분해, 같은 도메인 내 개인/그룹/듀엣 형식을 나눌 수 없었고 개인+그룹 혼합 패키지 도 팔 수 없었습니다.
원인	판매(상품)/운영(수업) 단위가 한 축(클래스 유형)에 묶여 있어 확장이 막힘.
해결	상품유형이 '패키지'인 경우를 위한 구성항목 테이블을 신설해 결제 시 unpack하여 멤버십 N개 생성. 기존 상품 목록 쿼 리는 상태/전시여부 조건이라 패키지가 자동 포함 → 구매 플로우만 분기하고 나머지는 무변경.
결과	단품 로직에 영향 없이 패키지 판매 확장 . "운영 중 자산은 건드리지 않고 확장은 격리"라는 원칙과 일치.

아키텍처

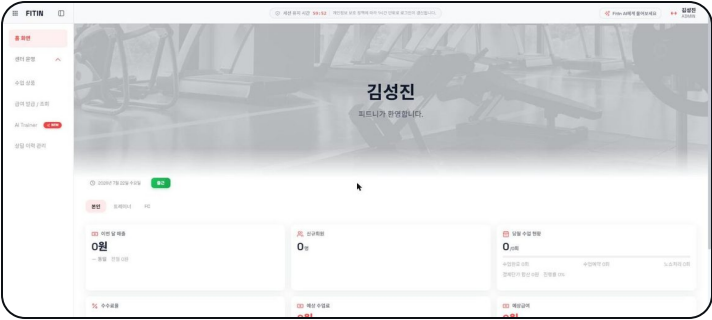


SCREENS

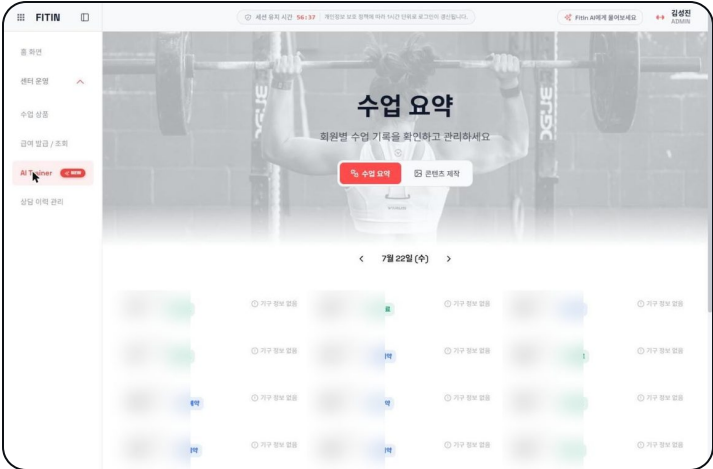
서비스 화면 — 실제 배포 4개 시스템

하나의 백엔드(BO/MO API/배치)가 아래 4개 웹을 서비스합니다.

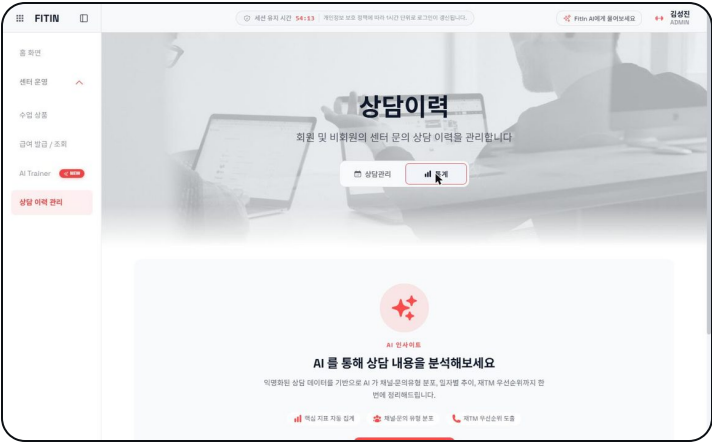
① 관리자 백오피스 데스크톱



대시보드 — 매출/회원/수업/급여 지표

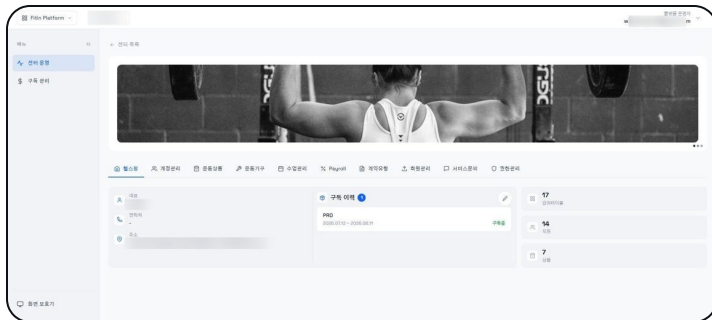


AI Trainer — 회원별 수업 요약/AI 콘텐츠

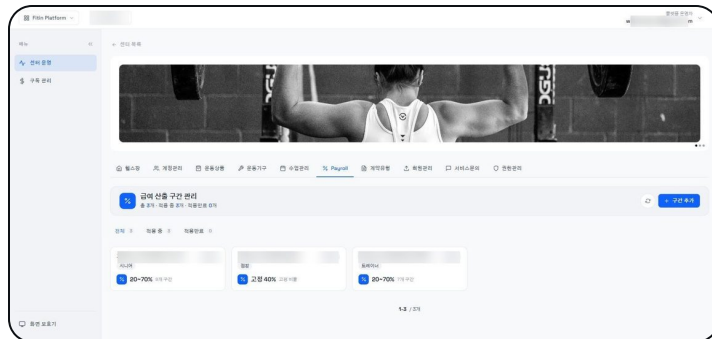


상담이력 AI 인사이트 — 채널/유형/재TM 분석

② 운영 어드민 데스크톱, 플랫폼 운영



센터 상세 운영 — 10개 관리 탭/구독/급여/직원

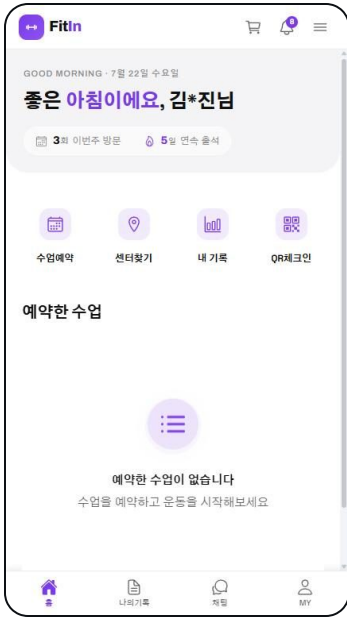


Payroll — 직급별 급여 산출 구간

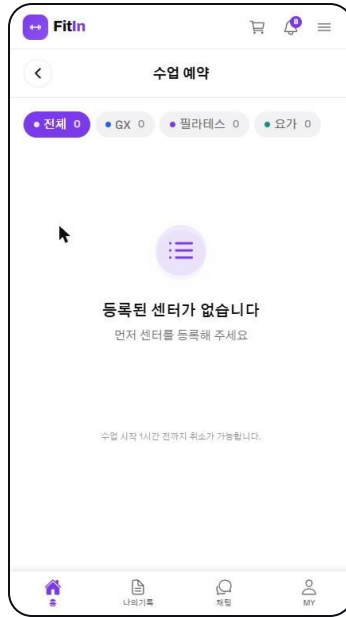


구독권 관리 — 플랫폼 구독권 + API 매핑

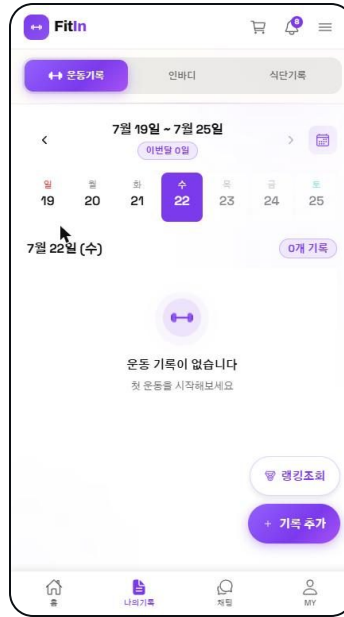
③ 회원 앱 모바일, 개발 중



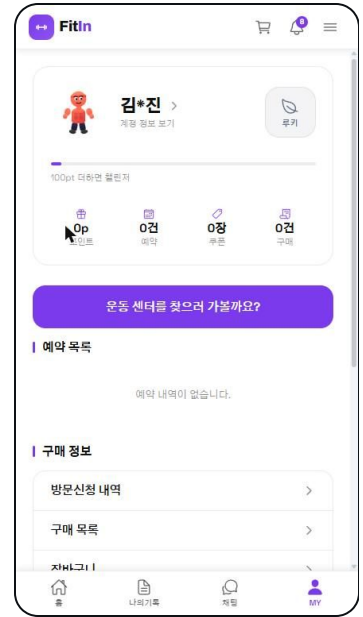
홈



수업 예약

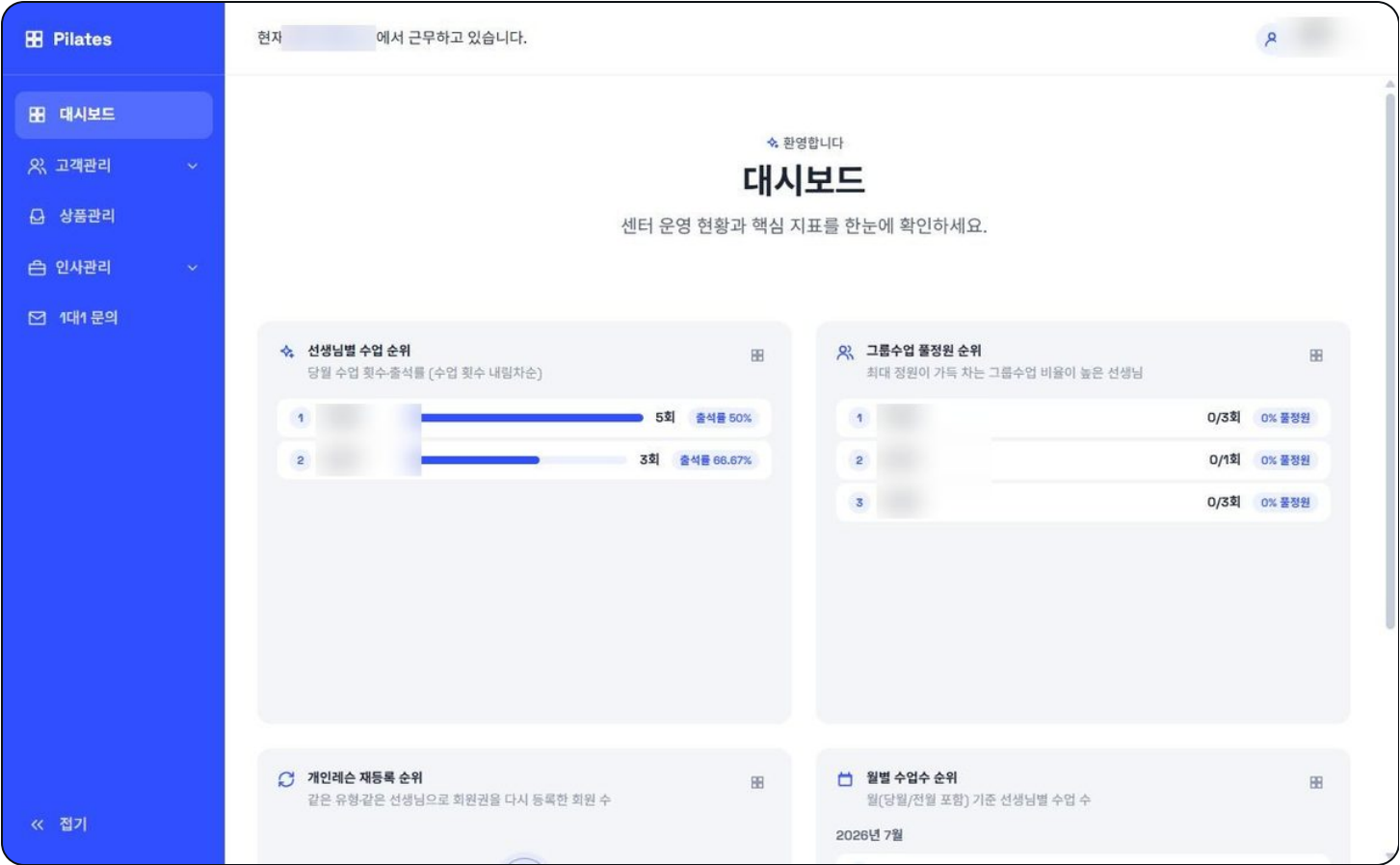


나의 기록



마이페이지

④ 필라테스 백오피스 데스크톱



대시보드 — 선생님별 수업 순위/풀정원/재등록 분석

STACK

기술 스택

Backend

Java 17

Spring Boot 4

MyBatis

Spring Batch

Data

MariaDB

AWS RDS

Redis

Pub/Sub

Storage

AWS S3

CloudFront

인증

JWT

Kakao OAuth (예정)

결제

Toss Payments (예정)

실시간/인프라

SSE

STOMP

FCM (예정)

Docker

AWS EC2

Nginx