

PRODUCT-MINDED FULL-STACK DEVELOPER

이기환

예약, 결제, CRM, 현장 POS를 하나의 제품으로 연결하는 제품형 풀스택 개발자입니다.

부산을 기반으로 활동하며 React, Node.js, Django, 클라우드 플랫폼을 활용해 반응형 웹 제품과 운영 시스템을 10년 이상 만들어 왔습니다. 웹 개발, 게임 개발, 인공지능, 로봇틱스에 관심이 있고, 아이디어를 빠르게 구체적인 PoC로 전환하는 프로토타이핑에 강점이 있습니다.

Django / DRF	React / TypeScript	Expo POS
Payment Flow	CRM Operations	GCP
PDF 다운로드	GitHub	Email



이기환

제품형 풀스택 개발자

Email kifhan@gmail.com

GitHub github.com/kifhan

Focus 제품 시스템

백엔드 계약, 고객 웹, 운영자 CRM, 현장 POS, 클라우드 운영을 하나의 제품 시스템으로 연결하는 데 집중합니다.

PRODUCT SUMMARY

고객, 운영자, 현장 흐름을 하나의 제품 상태로 연결

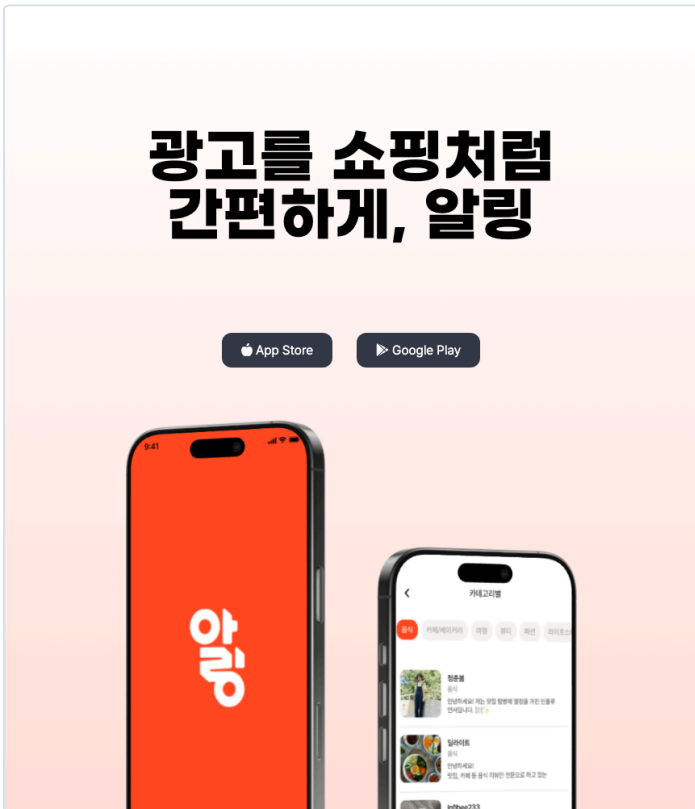
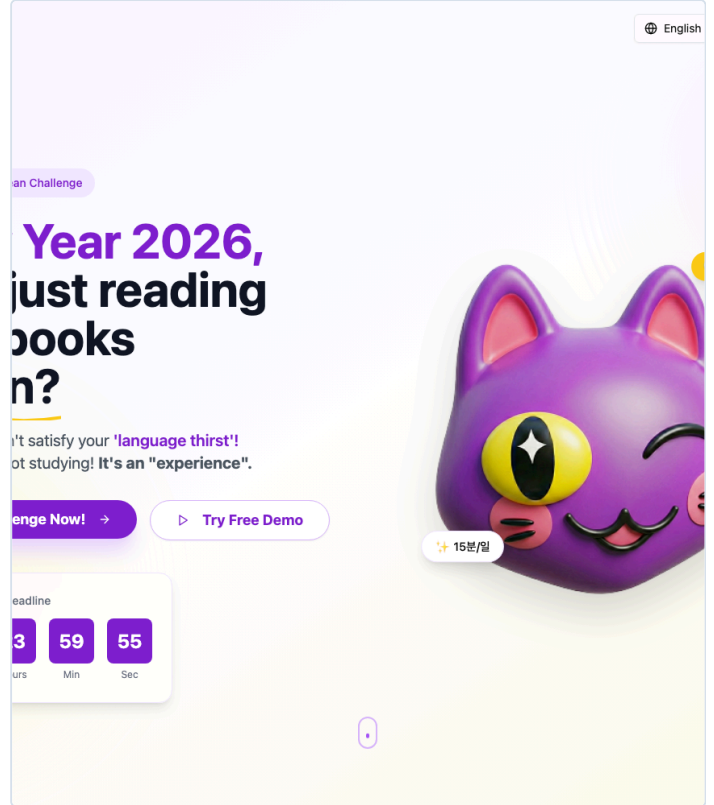
핵심은 화면의 개수가 아니라 서로 다른 제품 표면이 같은 주문, 예약, 결제, 권한, 사이트 단위 비즈니스 상태를 일관되게 공유하는 구조입니다.

<p>Customer Web</p> <p>구매와 예약</p> <p>고객이 상품을 탐색하고 결제, 예약, 구매 내역 확인까지 이어가는 흐름을 제공합니다.</p>	<p>CRM</p> <p>운영 관리</p> <p>예약, 주문, 결제 오류, 통계, 사이트별 권한을 운영자가 빠르게 처리할 수 있게 구성합니다.</p>	<p>POS</p> <p>현장 판매</p> <p>현장 결제, 주문 상태, 단말 설정, 빠른 복구 흐름을 태블릿/웹 UX에 맞춰 설계합니다.</p>	<p>Operations</p> <p>검증과 운영</p> <p>GCP, Cloud SQL, Storage, 배포 스크립트, 작업 문서를 통해 운영 리스크를 관리합니다.</p>
--	---	---	---

BUILT PROJECTS

직접 개발한 프로토타입과 서비스

제품 가설을 빠르게 검증하거나 실제 사용자 흐름을 보여주기 위해 만든 웹 서비스입니다.



PRODUCT SCREENS

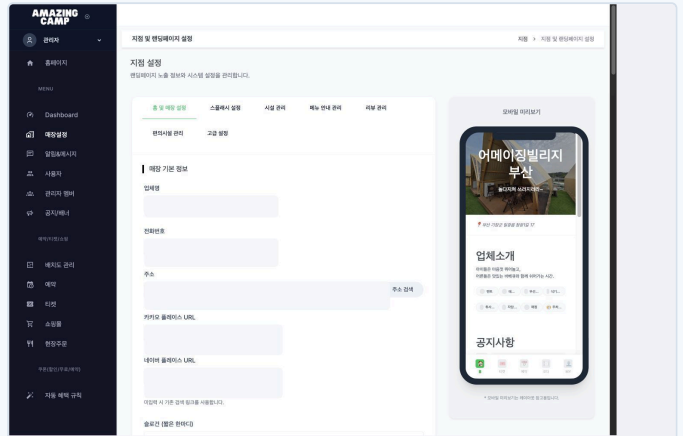
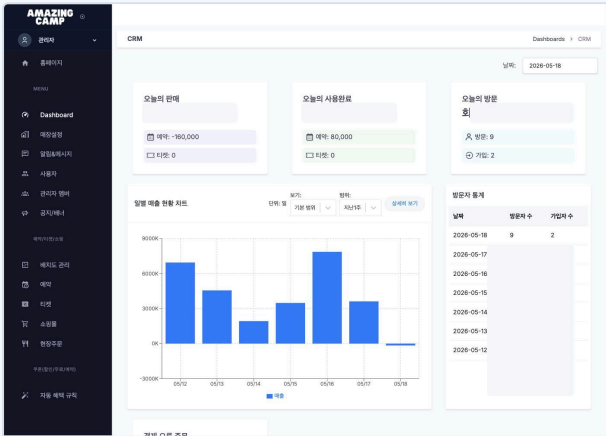
실제 서비스 화면으로 보는 제품 범위

고객 예약 화면, CRM 운영 화면, 현장 POS 화면이 같은 제품 흐름 안에서 이어집니다. 캡처는 staging 화면 기반이며 민감한 계정/예약 정보는 마스킹했습니다.

CRM 운영 보드: 대시보드와 매장 설정

CRM Board 1

운영 현황을 확인하고, 지점 랜딩/매장 정보를 같은 백오피스에서 관리합니다.

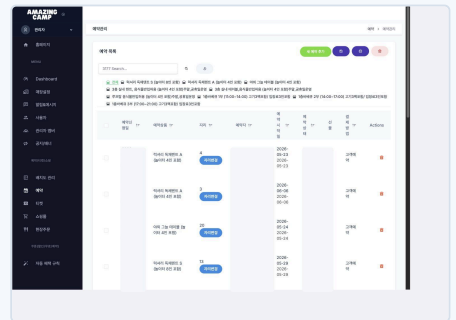
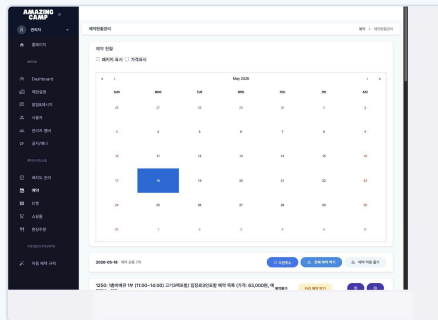
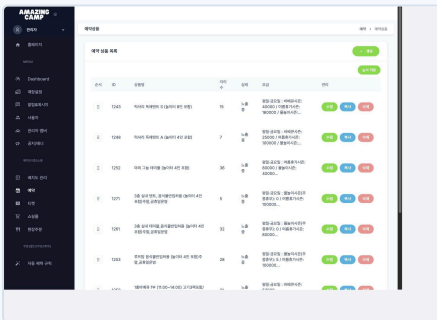


제품 가치 매출/방문/예약 지표와 매장 랜딩 설정이 분리되지 않아 운영자가 매일 필요한 상태를 빠르게 확인합니다.

CRM 예약 운영 보드: 상품, 캘린더, 예약 목록

CRM Board 2

예약 상품 구성, 일자별 현황, 예약 처리 목록이 하나의 운영 플로우로 이어집니다.

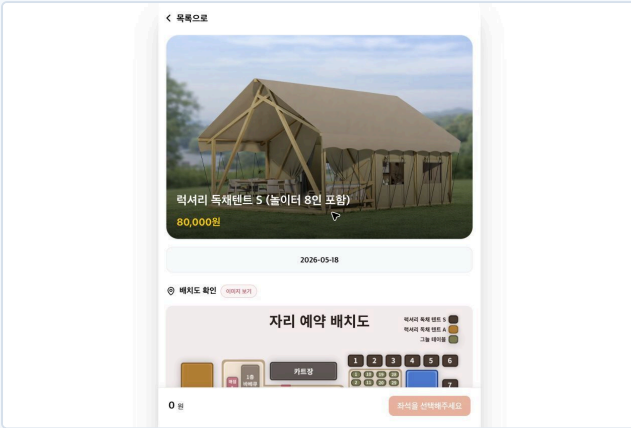


제품 가치 고객이 선택하는 상품과 운영자가 처리하는 예약 상태가 같은 API 계약 위에서 움직입니다.

고객 웹: 예약 상세

Web

날짜, 좌석, 상세 정보, 예약 CTA가 한 화면에서 이어지는 구매 전환 화면입니다.

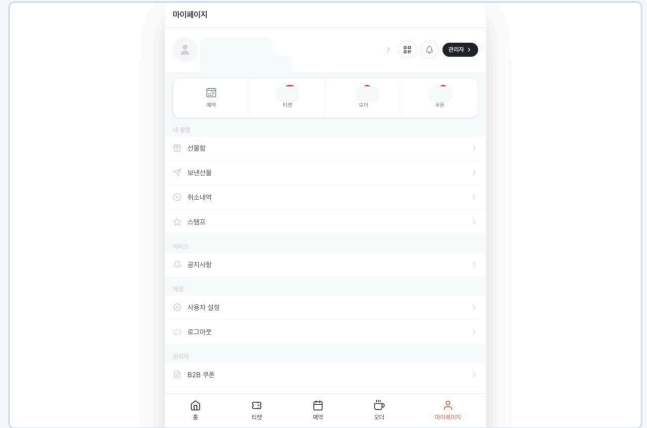


제품 가치 고객 선택 UI와 백오피스 상품/좌석 운영 기준이 같은 예약 기준을 공유합니다.

고객 웹: 마이페이지

Web

예약, 티켓, 오더, 쿠폰 상태를 고객이 직접 확인하는 self-service 화면입니다.

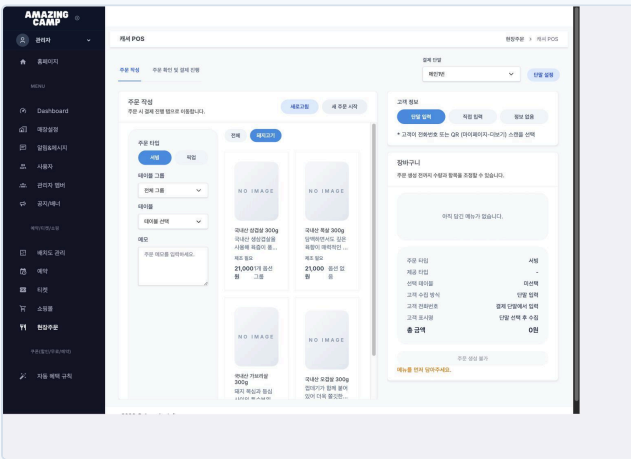


제품 가치 결제 이후 상태와 고객 보유 내역이 고객이 이해할 수 있는 제품 언어로 정리됩니다.

POS: 캐시 판매 화면

POS

테이블/픽업 선택, 메뉴 선택, 장바구니, 단말 결제 요청이 한 흐름으로 연결됩니다.

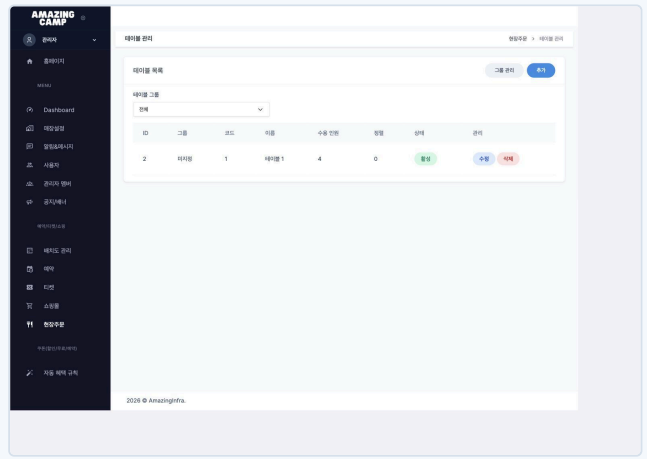


제품 가치 현장 직원의 빠른 조작을 위해 판매 흐름이 업무 단위로 나뉩니다.

POS: 테이블 관리

POS

현장 테이블과 주문 컨텍스트를 관리해 온라인 예약/현장 판매를 같은 운영 모델로 봅니다.



제품 가치 단순 결제 화면을 넘어 매장 운영 구조를 담은 POS 경험입니다.

PRODUCT CAPABILITIES

주요 제품 역량

고객, 운영자, 현장 직원이 같은 서비스 상태를 다룰 수 있도록 기능 경계와 데이터 계약을 맞춥니다.

예약/주문/결제 상태 흐름

고객 웹, CRM, POS가 같은 주문 상태를 해석하도록 백엔드 계약을 중심에 둡니다.

기술 기반 Django/DRF API, 상태 전이 규칙, 결제 사 응답 정규화

제품 효과 결제 성공/실패/예외 처리가 고객 흐름과 운영 대응으로 이어짐

CRM 운영 워크플로우

운영자가 반복적으로 처리하는 예약 변경, 결제 오류, 통계 확인을 밀도 높은 화면으로 제공합니다.

기술 기반 React, Vite, 서버 페이지네이션, bulk action, CSV export

제품 효과 반복 처리 비용을 줄이는 업무 중심 운영 UI

사이트 스코프 권한 모델

운영 권한을 전역 관리자 여부만으로 판단하지 않고 사이트 단위 역할과 액션 단위 정책으로 해석합니다.

기술 기반 Django permission gate, site-scoped permission helper, targeted tests

제품 효과 보안과 운영 편의성을 동시에 고려한 권한 계약

현장 POS 판매 흐름

현장 판매자는 빠른 상품 선택, 결제 대기 잠금, 복구 가능한 주문 상태가 필요합니다.

기술 기반 Expo Router, React Native, React Query, local device context

제품 효과 현장 터치 흐름과 결제 안정성이 같은 UX 안에서 유지됨

고객 마이페이지와 구매 내역

고객은 구매 내역, 예약 정보, 쿠폰/티켓 상태를 사이트 기준으로 정확히 확인해야 합니다.

기술 기반 React customer web, summary endpoint, shared modal contract

제품 효과 프론트 화면과 백엔드 집계 기준의 정합성 유지

운영 비용과 미디어 관리

운영 비용은 기능 밖의 문제가 아니라 서비스 품질을 지속 가능하게 만드는 제품 요소입니다.

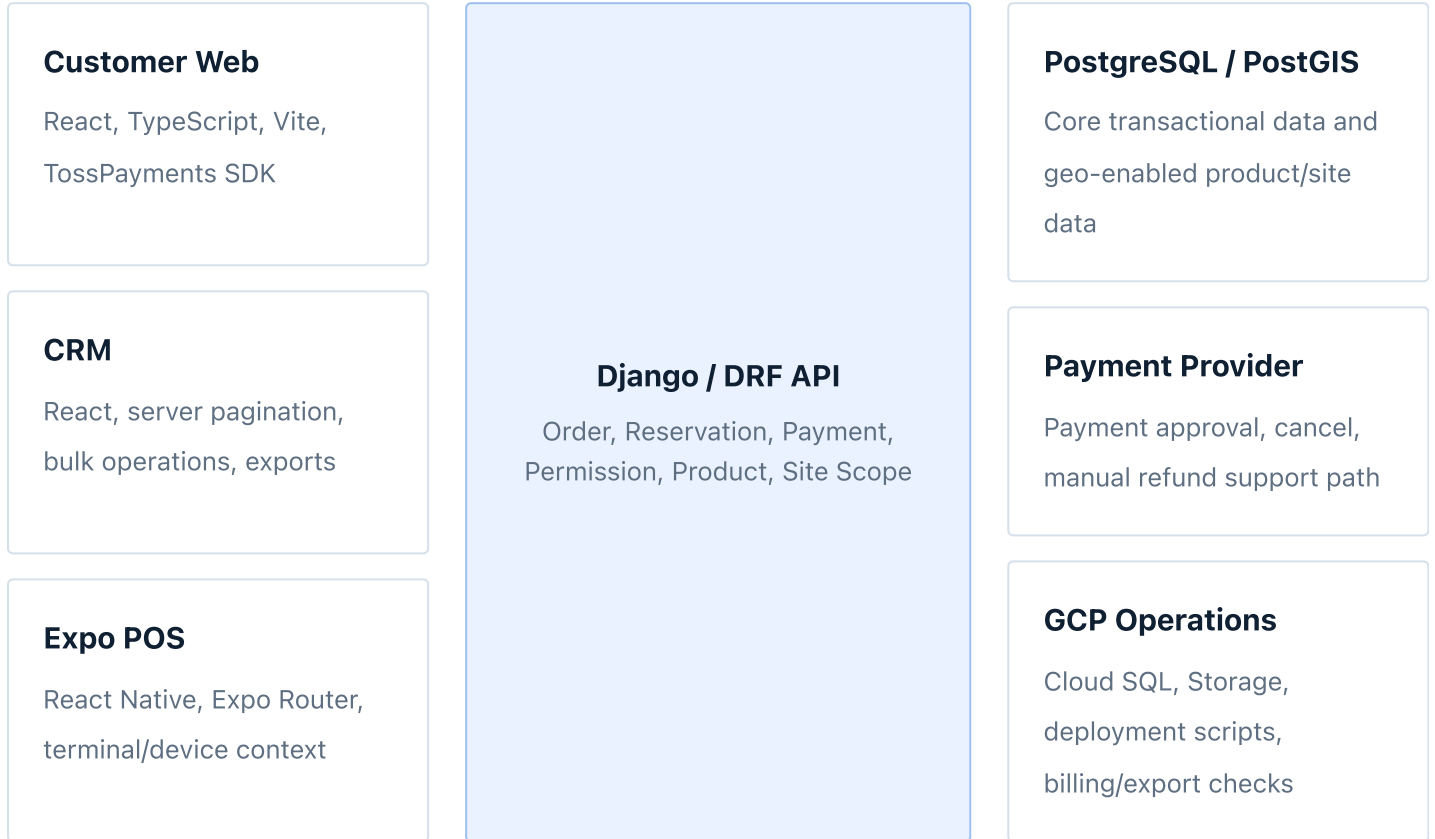
기술 기반 GCP Storage/CDN 점검, signed URL 구분, dry-run 기반 처리 스크립트

제품 효과 비용, 보안, 가시성을 함께 보는 운영 기준

ARCHITECTURE

서비스 아키텍처

중심은 Django API 계약입니다. 각 클라이언트는 사용 목적이 다르지만 같은 데이터 상태와 운영 규칙을 공유합니다.



Customer, operator, and cashier experiences are separated at the UI layer but converge on the same backend contract and operational evidence.

CASE STUDIES

대표 프로젝트 딥다이브

문제 정의, 계약 설계, 검증, 운영 반영까지 제품형 개발 역량이 드러나는 사례입니다.

CASE 01

예약 운영 플로우

CRM, reservation product, calendar

문제

고객이 보는 예약 상품과 운영자가 처리하는 예약 목록/캘린더 기준이 어긋나면 현장 운영이 흔들림.

제약

상품 노출, 좌석/일자 재고, 예약 상태, 취소/입장 처리가 서로 다른 화면에서 발생함.

접근

예약 상품, 캘린더, 예약 목록을 같은 Django API 계약과 site scope 기준으로 연결.

효과

상품 관리에서 일별 운영, 개별 예약 처리까지 한 흐름으로 이어지는 CRM 구조.

CASE 02

매장 설정과 고객 화면 연결

Site settings, customer web, product detail

문제

운영자가 수정한 매장 정보와 고객이 보는 랜딩/예약 상세가 일관되게 반영되어야 함.

제약

브랜드 정보, 상품 설명, 날짜, 좌석, CTA가 운영자 화면과 고객 화면 사이에서 분리되어 보일 수 있음.

접근

사이트 설정, 상품 상세, 좌석 선택, 마이페이지 요약을 동일한 site-aware 데이터 흐름으로 연결.

효과

CRM 설정 변경이 고객 경험으로 이어지는 제품 구조.

CASE 03

Standalone POS 현장 흐름

Expo POS, cashier workflow

문제

현장 판매는 화면 전환보다 빠른 선택, 결제 잠금, 실패 후 복구가 중요함.

제약

단말 설정, 사이트 선택, 주문 상태, 결제 대기 상태가 사용 중간에 흔들리면 안 됨.

접근

로그인 시점 site context, device-local settings, cart/payment state machine으로 분리.

효과

태블릿/웹 모두에서 현장 조작 흐름이 단순해지고 상태 복구 가능성이 높아짐.

TECHNICAL STACK

기술 스택과 선택 이유

스택은 제품 흐름, 운영 안정성, 검증 가능성을 기준으로 선택합니다.

Backend / API

- Django 5, Django REST Framework, Simple JWT
- PostgreSQL/PostGIS, Redis, Celery
- drf-spectacular, drf-yasg 기반 API 문서화
- Payment helper, permission gate, site-scoped business rules

Frontend / CRM / Web

- React 19, TypeScript, Vite, pnpm
- TanStack Query, Zustand, React Router
- Radix UI, Headless UI, Tailwind CSS, Bootstrap migration surface
- Server pagination, modal contract, toast/dialog UX, export flows

Mobile / POS

- Expo 55, React Native 0.83, Expo Router
- React Query, Secure Store, SVG/QR flows
- Device-local settings, terminal identity, payment state locks
- Android/Web shared cashier interaction model

Infra / Verification

- GCP, Cloud SQL, Cloud Storage, deployment scripts
- Targeted Django tests, Vitest, TypeScript checks
- Billing export and resource state validation
- Progress docs and task snapshots for operational continuity

UI / UX

사용자별 UX 원칙

고객, 운영자, 현장 판매자는 같은 제품을 보지만 속도와 오류 허용 범위가 다릅니다.

Customer Web

- 구매와 예약 흐름을 짧고 명확하게 유지
- 결제 전후 상태와 취소/환불 정보를 제품 언어로 표시
- 마이페이지에서 사이트 기준 요약과 구매 내역 정합성 유지

CRM

- 반복 업무를 빠르게 처리하는 고밀도 테이블과 bulk action
- 권한 실패, 결제 예외, 통계 기준을 운영자가 이해 가능한 상태로 표현
- 필터, export, 상태 변경을 업무 단위로 연결

POS

- 현장 터치 조작에 맞춘 큰 타겟과 빠른 상품 선택
- 결제 대기 중 이동/수정 잠금으로 상태 손상 방지
- 실패 주문 복구와 단말 설정을 업무 흐름 안에서 처리

WORKING STYLE

일하는 방식

문제 재현, 계약 확인, 작은 검증, 문서 동기화를 개발의 일부로 둡니다.

"실패 경로와 계약을 먼저 확인하고, 가장 작은 안전한 단위로 제품 품질을 높입니다."

- 코드베이스와 기존 계약을 먼저 읽고 변경 범위를 좁힙니다.
- 백엔드, 고객 웹, CRM, POS가 공유하는 상태 의미를 함께 맞춥니다.
- 결제, 권한, 예약처럼 리스크가 큰 흐름은 실제 caller, endpoint, permission gate를 추적합니다.
- 테스트와 수동 검증은 변경 위험에 맞춰 targeted하게 수행합니다.
- 진행 상태와 후속 작업은 문서에 남겨 다음 작업자가 이어받을 수 있게 합니다.

이기환

Product-minded Full-stack Developer

kifhan@gmail.com

<https://github.com/kifhan>