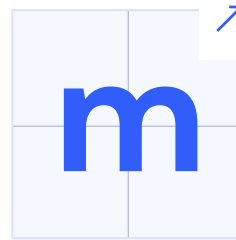


FULL-STACK & AI APPLICATIONS

김민혁

KIM MINHYUK



서비스를 만들고, 운영까지 연결합니다.

React·Next.js와 Python으로 웹 서비스를 구현하는 개발자입니다. 프론트엔드부터 API·데이터베이스·배포까지 연결하고, AI 기능에 근거·추적·실패 대응을 더합니다.

github.com/minhyuk35 minhyuk6512@gmail.com 010-7567-9674

01 핵심 역량

01 서비스 전체 흐름 구현

화면·인증·도메인 API·데이터 모델을 함께 설계하고 배포까지 수행했습니다.

[ASTRA](#) · [Study Flow](#)

02 운영을 고려한 AI 기능

원격 프롬프트, 응답 추적, 근거 출력과 실패 재시도를 실제 기능으로 구현했습니다.

[AI Ops Studio](#) · [DocMind](#)

03 시스템 간 연결과 협업

Discord·웹·게임을 API로 연결하고, 센서 팀과 데이터 계약을 문서로 합의했습니다.

[SIGN Voice](#) · [MajuBom](#)

02 주요 기술

FRONTEND

TypeScript · React · Next.js · Three.js

BACKEND / DATA

Python · FastAPI · Flask · PostgreSQL · Prisma · Drizzle

AI / INTEGRATION

OpenRouter · Gemini · Langfuse · SSE · Discord API

DELIVERY / QUALITY

Vercel · Render · pytest · ruff · TypeScript 검사

03 대표 프로젝트

01 AI Ops Studio

AI 응답 이후의 운영까지 설계



02 ASTRA

3D 경험과 서비스 기능을 하나로 연결



03 DocMind

답변의 근거를 확인할 수 있는 AI 경험



AI Ops Studio

커머스의 고객 문의를 AI로 응대하고, 프롬프트와 응답 이력을 운영자가 추적할 수 있도록 만든 통합 개발 프로젝트입니다.

기간·구성

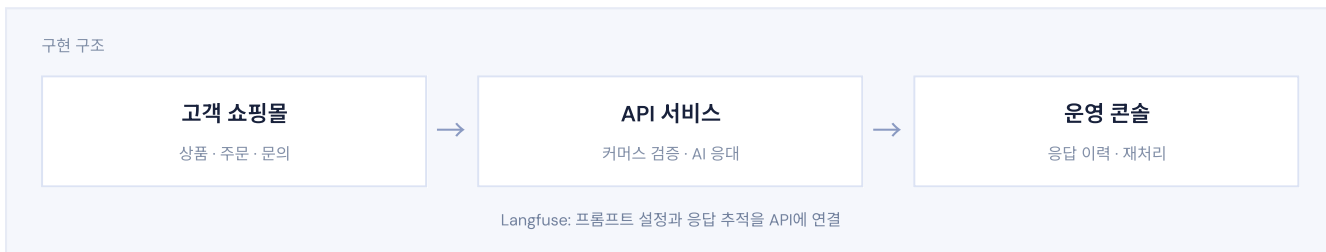
2026.07 / 개인 프로젝트

담당 역할

기획·설계·프론트엔드·API 전체 구현

해결하려던 문제

고객 문의에 AI가 답하더라도 어떤 프롬프트로 응답했는지, 실패한 작업을 어떻게 다시 처리할지 확인할 수 있어야 했습니다.



핵심 설계와 구현

01 프롬프트 변경과 배포 분리

Langfuse에서 프롬프트·모델 설정을 읽고, 연동 실패 시 환경변수 설정으로 폴백하도록 구현했습니다.

02 문의 처리 과정을 추적 가능하게

trace·session·user를 연결하고, 민감정보 마스크·담당자 전환·감사 로그·실패 작업 재시도를 콘솔에 구성했습니다.

03 주문 확인은 서버에서 검증

클라이언트 가격을 그대로 사용하지 않고, 상품 가격·재고·쿠폰을 주문 확정 직전에 서버에서 다시 계산했습니다.

구현 결과

고객 쇼핑물, 운영 콘솔, AI API와 커머스 API를 하나의 모노레포로 구성했습니다. 상품 탐색부터 모의 결제·반품, 문의 응대와 추적까지 연결했습니다.

확인할 수 있는 근거

주문 전체 흐름 검증 스크립트와 Langfuse 트레이스 전송 검증 스크립트를 작성했습니다. pytest·ruff·타입 검사·빌드 명령을 실행 절차에 포함했습니다.

구현 범위 결제는 로컬 모의 결제입니다. 실제 PG 웹훅 검증과 인증·인가 강화는 후속 과제로 문서화했습니다.

React

TypeScript

FastAPI

OpenRouter

Langfuse

SQLite

github.com/minhyuk35/AI-Ops-Studio

FULL-STACK / 3D COMMUNITY

ASTRA

3D 태양계를 탐색하며 길드·게시판·포인트 기능을 사용하는 커뮤니티 웹입니다. 기존 커뮤니티의 기능을 Next.js 기반으로 재설계했습니다.

기간 · 구성

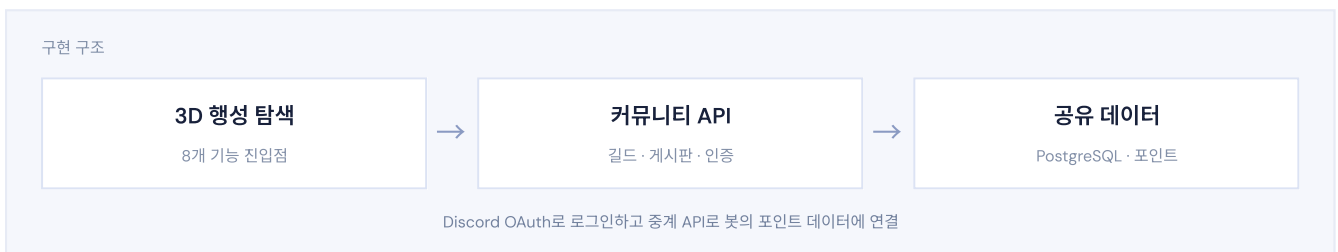
2026.05-06 / 개인 프로젝트

담당 역할

기획 · 디자인 · 3D · 백엔드 · 배포

해결하려던 문제

기존의 게시판·길드·포인트 기능을 유지하면서, 커뮤니티의 정체성을 탐색 경험에 반영하는 새로운 웹 구조가 필요했습니다.



핵심 설계와 구현

01 행성을 실제 내비게이션으로

8개 행성을 커뮤니티 기능과 연결하고, 관리자가 행성 노드의 매핑을 편집할 수 있도록 구현했습니다.

02 서비스 API를 단일 배포로 구성

길드·채팅·출석·공지 API를 Next.js에 모으고, Prisma 모델과 Discord OAuth 인증을 연결했습니다.

03 포인트 공유에 중계 API 사용

Discord 포인트를 조회하는 중계 엔드포인트를 두어, 웹에 필요한 값을 전달하도록 구성했습니다.

구현 결과

3D 탐색 화면에 길드·게시판·포인트·인증·관리 기능을 통합하고 자체 도메인에 배포했습니다. 기존 커뮤니티 웹의 기능을 새 도메인 모델로 이관했습니다.

확인할 수 있는 근거

공개 사이트와 저장소에서 3D 탐색 구조 및 구현 코드를 확인할 수 있습니다. 로그인 기반 커뮤니티 기능은 Discord 인증을 사용합니다.

Next.js

TypeScript

Three.js

Prisma

PostgreSQL

NextAuth

github.com/minhyuk35/ASTRAsign-company.co.kr

DocMind

PDF 질문에 답변과 근거 페이지·인용문을 제공하는 문서 분석 AI입니다. 원문 확인과 대화 복원까지 구현했습니다.

기간·구성

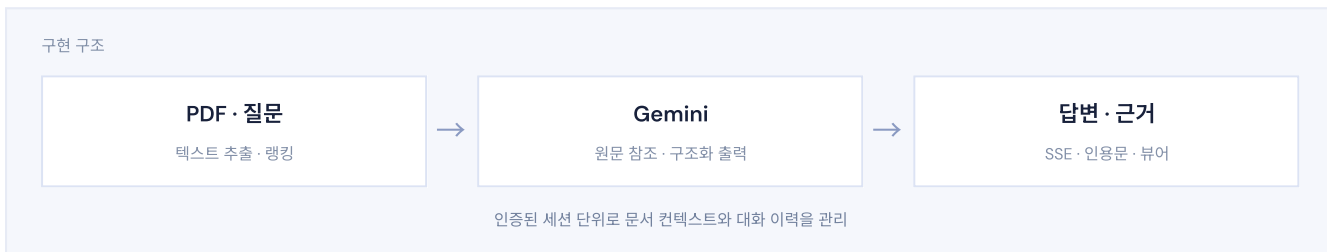
2026.05 / 개인 프로젝트

담당 역할

기획 · 프론트엔드 · AI 연동 · 데이터 처리

해결하려던 문제

AI 답변의 출처를 검토하기 어려워, 답변 옆에서 PDF 근거를 직접 확인할 수 있도록 설계했습니다.



핵심 설계와 구현

01 답변과 근거를 응답 계약으로

JSON Schema에 답변·페이지·인용문·근거 이유를 정의하고, PDF 정보가 부족한 경우 웹 출처를 구분해 제공했습니다.

02 관련 텍스트와 원문을 함께 사용

관련 페이지 청크를 랭킹하고 Gemini File API로 원본을 함께 참조했습니다. SSE로 답변과 근거 메타데이터를 전달했습니다.

03 세션 복원 전에 소유권 확인

메모리 캐시에 문서가 없으면 세션 소유자를 인증한 뒤 DB에서 복원했습니다. PDF 뷰어와 근거 영역 크롭도 구현했습니다.

구현 결과

PDF 업로드·질의응답·근거 뷰어·대화 복원을 연결했습니다. 답변과 원문 인용을 같은 화면에서 검토할 수 있습니다.

확인할 수 있는 근거

응답 스키마·페이지 랭킹·SSE·PDF 렌더링·세션 접근 제어 구현을 저장소에서 확인할 수 있습니다.

구현 범위 신뢰도는 모델 자기 평가이며, 정답률·환각 감소율을 실측한 값이 아닙니다.

Next.js

TypeScript

Gemini

SSE

Neon

pdfjs

github.com/minhyuk35/DocMind

경험의 확장.

협업, 서비스 운영, 플랫폼 연동과 인터랙티브 웹까지.
프로젝트별로 직접 기여한 구현을 정리했습니다.

팀 · 캡스톤 디자인

2026.06

MajuBom

팀에서 Flask 수집 서버·분석·대시보드를 담당했습니다. 센서 데이터 계약을 합의하고 분석 로직 분리와 원자적 저장을 구현했습니다.

Python · Flask · SQLite · HTTP JSON

[UI 기획 ↗](#) [시스템 구현 \(팀\) ↗](#)

개인 프로젝트

2025.12 개발 시작 · 2026.05 저장소 분리

SIGN Bot

Cog 자동 로딩·하리로드, 비동기 API 호출과 이원 DB를 설계한 Discord 봇. 전체 구현과 배포·운영을 담당했습니다.

Python · discord.py · OpenAI · SQLite

[GitHub ↗](#)

개인 프로젝트

2026.04

Study Flow

복습 스케줄 재생성의 먹등성과 Gemini 실패 재시도를 구현했습니다. Server Actions·Drizzle·Neon을 사용해 배포했습니다.

Next.js · Gemini · Drizzle · Neon

[GitHub ↗](#) [사이트 방문 ↗](#)

개인 프로젝트

2026.05

SIGN Voice

Roblox와 Discord 포인트를 서버리스 API로 연결했습니다. API Key·계정 연동 코드를 구현하고 DB 접근 범위를 제한했습니다.

Luau · Next.js · Neon · Vercel

[GitHub ↗](#)

개인 프로젝트

2026.06

Cosmosea

고래·나비의 지오메트리와 이동 경로를 직접 생성하고, 스크롤 카메라·GLSL·후처리를 결합했습니다.

Three.js · React Three Fiber · GLSL · Postprocessing

[GitHub ↗](#)

학과 프로젝트

학과 프로젝트

HATRIO

JSP·Servlet·JDBC로 쇼핑몰과 DB를 구현했습니다. WAR·SQL 초기화·설치 가이드까지 배포 세트로 구성했습니다.

Java · JSP · Servlet · JDBC

WAR · SQL · 설치 가이드 · 발표 자료

LET'S BUILD SOMETHING USEFUL.

[minhyuk6512@gmail.com ↗](mailto:minhyuk6512@gmail.com)