

빠른 UI, 견고한 API, 실용적인 시스템을 설계합니다.

API 설계와 실시간 협업에 집중합니다.



ABOUT ME

나에 대하여

6+

프로젝트

30+

기술 스택

100%

Full-Stack

소개

AI/RAG에 집중하는 풀스택 엔지니어

실시간 협업 도구 구축을 즐김

신뢰성과 UX를 중요시

핵심 역량

End-to-end 책임감 (UI → 인프라)

API 설계 + 데이터 모델링

Docker/AWS/Nginx로 DevOps

주요 성과

CRDT 기반 라이브 리뷰 도구 구축

AI 기반 글쓰기 플랫폼 출시

AWS에 풀스택 앱 배포



기술 스택

분야별로 그룹화된 기술들입니다. 각 기술은 실제 프로젝트에서 사용한 경험을 반영합니다.

Frontend

TypeScript

React

Vite

Tailwind CSS

Zustand

React Query

React Router

MUI

WebRTC

Backend

Spring Boot

FastAPI

REST APIs

OpenAPI/Swagger

WebSocket

PostgreSQL

MySQL

Redis

DevOps & Cloud

Docker

Docker Compose

Nginx

AWS EC2/S3

GitHub Actions

CI/CD

AI & Data

OpenAI API

LangChain

RAG

Vector DB

Pinecone

🤝 Collaboration

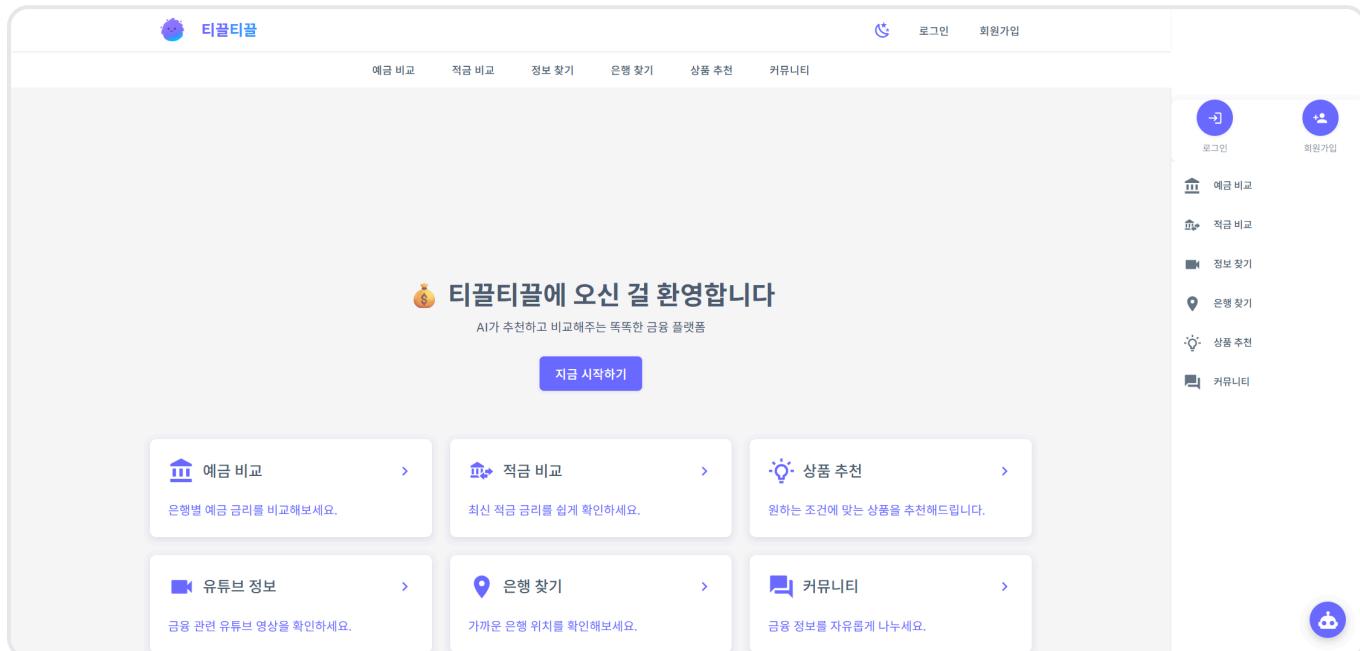
Notion

Slack

Figma

Jira

예·적금 추천, 자산·지도, 커뮤니티, AI 챗봇까지 뮤은 재테크 파트너 서비스.



프로젝트 개요

“금융감독원 API를 통한 금융상품 추천, Kakao 지도, 자산·커뮤니티, OpenAI/LangChain 챗봇 연동. EC2 + Docker + Nginx로 배포.”

기술 스택

Django

React

Docker

Nginx

AWS EC2

OpenAI

Kakao Maps

✨ 주요 기능

- 금융상품 API 기반 추천/검색
- 지도 기반 매장/지점 보기
- 커뮤니티 게시판/댓글
- AI 챗봇로 금융 Q&A

👤 담당 역할

- 금융 추천 로직 및 외부 API 연동
- 커뮤니티/지도 기능 구현
- AI 챗봇 프롬프트·플로우 구성
- 배포 파이프라인 일부 구성 및 CORS/환경변수 정리

🔧 트러블슈팅

1. 모델 입력 스케일 불일치

문제 예측치 분산 과대

원인 전처리 불일관

해결 입력 스케일러 고정·검증 단계 추가

근거: final_pjt_back/requirements.txt(ML 스택), 금융 입력 변수 특성

2. 로컬/배포 환경 차이

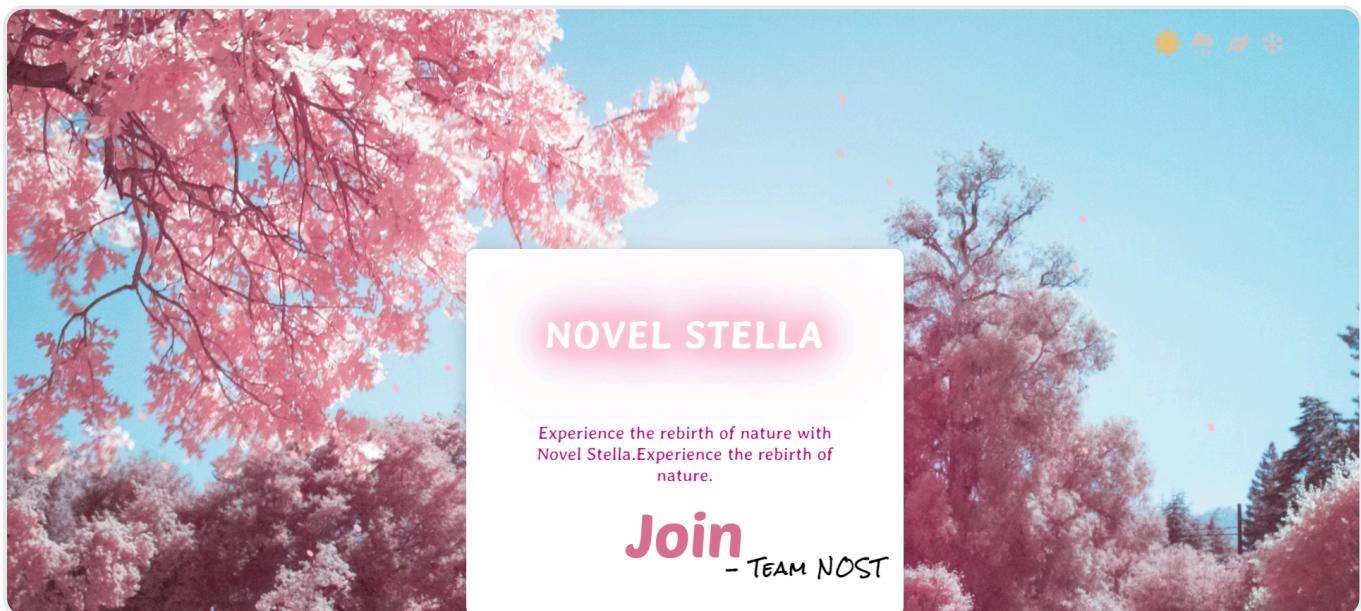
문제 브랜치별 동작 차이

원인 master(배포) vs release/local(로컬) 분기

해결 .env 프로필 분리·CI 환경 행렬

근거: README.md(브랜치 안내)

AI가 제안한 3가지 후보 중 선택/수정해 소설을 완성하는 생성형 커뮤니티.



프로젝트 개요



React + Django + PostgreSQL. LangChain/ChatGPT/DALL-E/DeepL을 활용해 다국어/이미지 생성 파이프라인 구성.



기술 스택

React

Zustand

Django

PostgreSQL

AWS

LangChain

OpenAI

DALL-E

✨ 주요 기능

- 3가지 후보안 생성·선택
- 문장 단위 편집/재생성
- 이미지 생성(표지/일러스트)
- 커뮤니티 공유·댓글



담당 역할

- 생성 파이프라인 설계 및 프롬프트 엔지니어링
- 피드/댓글 등 커뮤니티 흐름 설계
- 다국어(번역) 유필/컴포넌트 정리



트러블슈팅

1. 대용량 텍스트 저장 성능

문제	본문 저장 시 응답 지연
원인	단일 모델 직접 저장
해결	본문 외부 스토리지 분리(S3)-메타데이터 DB 저장 제안

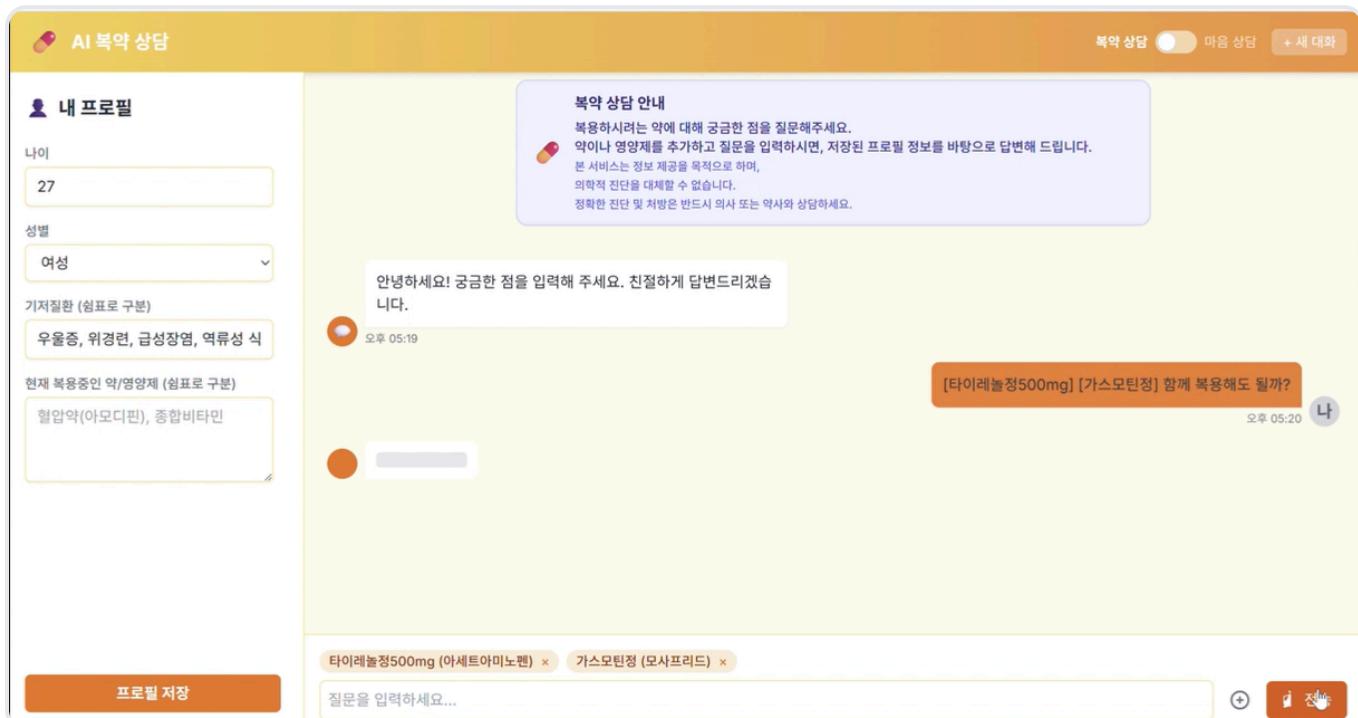
근거: backend/README.md 구조, 창문 생성 서비스 특성

2. 이미지 생성 실패율

문제	이미지 실패/429
원인	생성 API 쿼터 초과
해결	큐-콜다운·리트라이, 사용자별 레이트리밋

근거: frontend 비동기 호출 구조, 생성형 워크플로우

LLM + RAG로 정확하고 안전한 복약 정보를 제공하는 챗봇.



☰ 프로젝트 개요

FastAPI + LangChain + OpenAI. 의약품 데이터 파싱, 벡터 DB 인덱싱, RAG 파이프라인 구성.

🛠️ 기술 스택

FastAPI

LangChain

OpenAI

Pinecone

PostgreSQL

Python

✨ 주요 기능

- 의약품 검색/QA
- 상호작용/금기 자동 안내
- 출처 근거(References) 함께 제시

👤 담당 역할

- RAG 파이프라인 설계(인덱싱·검색·답변 체인)
- 의약품 데이터 파싱/정규화
- 안전성/금기/상호작용 가이드 출력 포맷

🔧 트러블슈팅

1. 한글 PDF 파싱 인코딩

문제 EUC-KR 문서 파싱 오류

원인 전처리 미흡

해결 UTF-8 강제 변환·예외 처리 파이프라인

근거: backend/requirements.txt(PDF 관련), 데이터 소스 comprehensive_drug_safety_docs.csv

2. RAG 검색 지연

문제 질의 응답 시간 과다

원인 벡터 검색 후 LLM 호출 순차 처리

해결 비동기 파이프라인 구성, 캐싱 레이어 추가

근거: backend/rag_pipeline.py, 성능 최적화

WebSocket 기반 실시간 협업으로 API 명세서를 작성하고 테스트하는 도구.

The screenshot shows the Ouroboros API tool interface. On the left, there's a sidebar with tabs for REST and WebSocket, a search bar, and filters for All, Mock, and Completed. The main area has tabs for API Spec and API Test, with API Spec selected. Under API Spec, there's a section for Method & URL (POST /api/test), Tags (test-controller), Request and Response sections, and a Body section with fields: email (string), password (string), name (string), age (integer), and phone (string). The body is defined with application/json type. At the bottom right are Import and Export buttons, and at the bottom center are Edit and Delete buttons.



프로젝트 개요



“React + TypeScript + Spring Boot. CRDT 기반 동시 편집, OpenAPI 스페지fi션, 코드 스니펫 생성.”



기술 스택

React

TypeScript

Spring Boot

WebSocket

CRDT

OpenAPI

✨ 주요 기능

- 실시간 동시 편집
- API 엔드포인트 테스트
- 코드 스니펫 자동 생성
- 버전 관리 및 히스토리



담당 역할

- CRDT 기반 실시간 협업 로직 구현
- OpenAPI 스펙 파싱/검증 시스템
- 코드 스니펫 생성 엔진



트러블슈팅

1. FormData 무한 루프

문제 useEffect 무한 루프 발생

원인 상태 업데이트 사이클

해결 useRef로 이전 값 추적, 조건부 업데이트

근거: front/src/features/testing/components/RequestBodyForm.tsx, FormDataBodyForm 컴포넌트

2. Node.js 전용 라이브러리 브라우저 호환성

문제 openapi-snippet 브라우저 에러

원인 Node.js 전용 모듈 의존

해결 vite-plugin-node-polyfills 추가

근거: front/vite.config.ts, front/src/features/spec/components/CodeSnippetPanel.tsx

3. Basic Auth 다국어 문자 Base64 인코딩

문제 InvalidCharacterError 발생

원인 btoa() ASCII만 처리

해결 UTF-8 안전 인코딩 함수 구현

근거: front/src/utils/encoding.ts, safeBase64 함수

4. JSON 편집기 사용자 경험 개선

문제 들여쓰기/구문 강조 부재

원인 기본 textarea 제한

해결 react-ace 에디터 통합

근거: front/src/components/JsonEditor.tsx, react-ace 통합

5. WebSocket 작업 완료 토글 반응성 최적화

문제 토글 클릭 후 지연

원인 비동기 작업 순차 실행

해결 isUpdatingProgressRef 플래그 추가

근거: front/src/features/spec/components/ApiEditorLayout.tsx, progress 토글 핸들러



연락하기

이메일: hy1x1mn@gmail.com

GitHub: github.com/hy2min