

김찬혁 Backend Developer

FastAPI · NestJS · PostgreSQL 기반 API 설계 · 트래픽 대응 · 실사용자 서비스 운영

449명 1,016개 6회

실사용 회원 자동화 테스트 수상 실적



김 찬 혁

연락처

010-3738-1882
dissolve1882@naver.com

아카이빙

github.com/inhadissolve
blog.naver.com/dissolve_chan
포트폴리오 별첨(PDF)

학력 · 자격

인하대학교 재학
정보처리기사 취득 예정

지원 동기

실사용자가 매일 쓰는 서비스를 **혼자 설계하고, 배포하고, 장애까지 감당해본** 백엔드 개발자입니다. 서버리스 환경에서 커넥션 풀과 풀링 부하를 직접 계산해 피크 트래픽을 대비했고, AWS 운영 중 발생한 다운타임의 원인을 추적해 자동 복구가 되도록 구조를 잡았습니다. 기능을 만드는 사람은 많지만 **그 기능이 트래픽 아래에서 버티는지까지 책임지는 사람**은 드물다고 생각합니다. 서비스 규모가 커지며 성능·비용·안정성을 함께 고민해야 하는 팀에서, 이미 그 문제를 작게라도 겪어본 개발자로 기여하고 싶어 지원합니다.

기본기

API 명세 · ERD · 트러블슈팅을 문서로 남겨 팀이 같은 기준에서 움직이게 만듭니다. 백엔드 pytest 685개와 프론트 테스트 331개를 유지하며 **한 번 겪은 장애를 다시 겪지 않도록** 회귀 테스트로 고정하고, 팀 프로젝트에서는 API 계약과 PR 우선순위를 조율하는 리드 역할을 맡아왔습니다.

핵심 프로젝트

[개인 서비스 · 1인 기획·개발·운영] 메모라이즈 — 암기 학습 웹서비스



메모라이즈 (Memorize)

v3.78 · React 19.2 · TypeScript 5.9 · FastAPI 0.140 · PostgreSQL(Neon) · OpenAI SDK 2.53

일정 : 2025.11 ~ 현재 · **참여 인력** : 1인 (기획 · FE · BE · AI 연동 · 운영)

서비스 : 학습 · 오답노트 · 저장형 시험 · 학습 기록으로 이어지는 암기 훈련 웹서비스, 실사용자 대상 운영 중

예상 트래픽 및 인프라 설계 기준 : 시험 기간 피크 동시접속 100~300명을 가정. 로그인 사용자는 화면을 열어두기만 해도 배지 30초 · 알림 30초 · heartbeat 2분 폴링으로 **1인당 시간당 약 270회** 요청이 발생하므로, 동점자 수가 아닌 체류 시간을 부하 기준으로 삼음. Vercel 서버리스 + Neon PostgreSQL(pooled) 구성, 인스턴스당 커넥션 3개 기준으로 선정.

449명

전체 회원

16,496회

14일 학습 완료

36명

최고 동시 접속

1,016개

자동화 테스트

- 서버리스 커넥션 고갈 방지** — 인스턴스 하나가 요청 하나를 처리하는 서버리스 특성상 풀을 크게 잡으면 인스턴스 수만큼 커넥션이 급절로 늘어 DB 한도에 부딪히는 구조를 파악, **pool_size 1 / max_overflow 2**로 인스턴스당 3개로 제한해 오토스케일 상한에 서도 커넥션이 병목이 되지 않도록 설계
- 컴퓨터 예산 소진으로 인한 전면 다운 예방** — 풀링 부하를 근거로 피크 기간 소모량을 계산한 결과 **하루 8시간 × 6일이면 월 예산 100 CU-hour**를 소진해 시험 도중 서비스 전체가 정지하는 시나리오를 사전에 확인, 피크 주간 전 플랜을 선제 전환해 다운타임 0으로 운영
- 지역 간 왕복 지연 제거** — API 함수 배포 리전이 기본값(미 동부)이면 싱가포르 DB까지 왕복으로 요청마다 수백 ms가 추가되는 문제를 확인, 함수 리전을 DB와 동일한 **sin1**로 고정해 지연을 구조적으로 제거
- AI 비용 가드레일** — 출제·채점의 추론 강도를 분리하고 예산 · 주간 생성 한도를 적용해 **14일간 시험지 170개를 추정 비용 \$0.24**로 운영, 관리자 화면에서 호출 실패와 누적 비용을 상시 확인
- 조회 성능 및 회귀 방지** — 시험 목록 조회 API의 N+1 쿼리를 배치 조회로 제거하고 페이지네이션 적용, 화면이 실제로 보내는 파라미터 형태까지 회귀 테스트에 포함해 **pytest 685개 · 프론트 331개** 유지
- 운영 관측성 확보** — 관리자 대시보드에서 **시간대별 최고 동시접속**과 기능별 이용 회원, AI 호출 실패 건수와 누적 추정 비용을 상시 확인하도록 구성해, 감이 아니라 지표를 근거로 개선 우선순위를 판단
- 운영 · 개발 환경 분리** — 로컬은 SQLite, 운영은 PostgreSQL로 분리해 실제 데이터가 운영 DB에 섞이지 않도록 하고, 버전 변경마다 변경 내역 · README · 앱 메타데이터를 함께 갱신하는 릴리스 규칙으로 운영

기술 스택

Backend

FastAPI 0.140 NestJS 11 Node.js 22 Python 3.10+ TypeScript 5.9

Database

PostgreSQL 18 SQLAlchemy 2.0 Prisma 6 Alembic SQLite

Cloud · Infra

AWS (EC2 · ALB · RDS · ECR · S3) Terraform 1.8 Docker GitHub Actions CloudWatch Vercel

Test

pytest 8.3 Jest 29 Supertest Testcontainers 12

AI · Data

OpenAI SDK 2.53 PyTorch 2.8 torchvision 0.23 TensorFlow 2.13 MediaPipe 0.10 Pandas 2.2

Frontend

React 19.2 Next.js 16.2 Vite 7.2 TailwindCSS

핵심 프로젝트 (계속)

[UMC 10th 팀 프로젝트 · Backend Team Lead] HomeFit — 청년 주거 · 금융 의사결정 서비스



HomeFit

NestJS 11 · Node.js 22 · Prisma 6 · PostgreSQL 18.4(RDS) · Terraform 1.8 · AWS

일정 : 2026.06 ~ 2026.08 · 부산 BEXCO 데모데이 운영 종료

참여 인력 : 백엔드 팀장(본인) 포함 10인 팀 프로젝트 · 기획 · 디자인 · Android 협업

Backend(Node.js) 파트 베스트 챌린저 개인 선정

예상 트래픽 및 인프라 설계 기준 : 데모데이 현장 동시 체험 트래픽을 가정에 ALB + EC2 Auto Scaling(min 1 / max 3) 구성. 헬스체크 30초 간격 · 연속 3회 실패 시 교체, ASG grace period 300초 기준으로 단일 인스턴스 장애 시 약 5분 내 자동 복구되도록 설계.

6개

검증 단위

7건

연동 광고

4개

금융상품

5분 내

자동 복구

- 1 운영 중 서버 다운 원인 파악 및 재발 방지 — 운영 중 약 5분간 접속이 끊긴 뒤 자동 복구되는 현상이 발생, CloudWatch 지표를 추적해 **t3 버스터미널** 인스턴스의 **CPU 크레딧 소진**으로 응답이 지연되며 헬스체크에 연속 실패한 것이 원인을 확인. ASG를 **ELB 헬스체크** 기반으로 두어 크레딧이 고갈된 인스턴스가 자동 교체되도록 하고, **CPUCreditBalance** 알람을 추가해 고갈 전에 감지하도록 조치
- 2 역동 데이터 수집 — 공식 출처 기반 6개 단위 · 7건 광고를 **Stable Hash + Prisma Upsert**로 재실행해도 중복되지 않게 Seed화하고, 주석형 · 자격조건 · 첨부파일을 트랜잭션으로 일괄 저장
- 3 API 계약 · 통합 테스트 주도 — 공통 응답과 오류 규칙을 동결하고 적격 · 탈락 · 프로필 미입력 계정의 예상 결과를 도메인 담당자와 정리해 Android 최종 스모크 테스트를 주도, Docker · Testcontainers 기반 E2E로 검증
- 4 배포 파이프라인 및 데이터 이관 — Railway 1차 배포를 **AWS Terraform IaC** 구조로 확장하고 GitHub Actions OIDC 배포를 구성, PostgreSQL 전체 데이터 이관 후 건수 교차 검증까지 완료

[KSEB 팀 프로젝트 · 7인 팀 팀장] 수담(手談) — 실시간 양방향 수어 인식 · 번역 서비스



수담(手談)

Python 3.10 · MediaPipe 0.10 · TensorFlow 2.13 · FastAPI 0.110 · Docker

일정 : 2025.07 ~ 2025.09 · 참여 인력 : 7인 팀 팀장 (AI · FE · BE · Unity 파트 조율)

KSEB 팀 프로젝트 우수상 (3위)

오픈소스 SW 페스티벌 최우수상 (인하대 총장상)

- 1 FastAPI · Docker 기반 AI 추론 API 구현 — JPEG 프레임을 MediaPipe로 처리해 **10프레임 × 194차원 텐서**로 변환하고, 라벨 · 신뢰도 · 상위 후보를 반환하는 API 구성
- 2 실시간 예측의 전처리 불일치 수정 — 학습에 없던 샘플별 정규화를 실시간 예측 스크립트에서 제거해 학습 · 추론 입력 스케일 일치
- 3 비수어 동작 오탐 · 중복 출력 제어 — 타이핑 · 휴대전화 사용을 none으로 묶는 학습 로직 추가. 예측 스크립트에 **신뢰도 임계값 · 3회 연속 일치 · 중복 누적 방지** 적용

업무 경험

UMC

UMC 10th — Backend(Node.js) 파트 챌린저

2026.03 ~ 2026.08

- 주차별 Node.js/Express 미션을 완주하고 **HomeFit 백엔드 팀장**으로 도메인별 API 계약 · PR · Android 연동 일정을 조율
- Railway 배포에서 AWS Terraform · GitHub Actions OIDC · RDS 데이터 이관까지 **서비스 생명주기 전 과정**을 경험
- 2026.08.22 부산 BEXCO 데모데이 현장 운영, **Backend(Node.js) 파트 베스트 챌린저 개인 선정**

탄소

탄소중립 Innovation Academy 5기 · 기업연계 프로젝트

2025.09 ~ 2025.12

- 산업 설비 진동 데이터를 Orbit 이미지로 변환, **PyTorch · torchvision ResNet18**로 정상 · 비정상进行分类하는 이상탐지 AI를 AI/HW 파트로 개발
- Grad-CAM 판단 근거 시각화와 --json 출력 옵션으로 백엔드 연동용 추론 모듈 구성 — **대상(1위)** 수상

KSEB

KSEB 부트캠프 4기 수료

2025.01 ~ 2025.11

- 1년간 백엔드 개발 역량을 집중 학습하며 미니프로젝트 대상 · 팀 프로젝트 우수상 수상, **전체 수료생 중 8명**의 개인 우수 교육생으로 선정

수상 내역

UMC 10th Backend(Node.js) 파트 베스트 챌린저

역할 : HomeFit 백엔드 팀장

2026.08

탄소중립 Innovation Academy 5기 기업연계 프로젝트 대상 (1위)

역할 : AI/HW 파트

2025.09 ~ 2025.12

오픈소스 SW 페스티벌 최우수상 · 인하대학교 총장상

역할 : 팀장

2025.09 ~ 2025.10

KSEB 팀 프로젝트 우수상 (3위)

역할 : 7인 팀 팀장

2025.07 ~ 2025.09

KSEB 미니프로젝트 대상 (1위)

역할 : AI/BE 담당

2025.03 ~ 2025.06

KSEB 부트캠프 4기 개인 우수 교육생

전체 수료생 중 8명 선정

2025.01 ~ 2025.11