

김찬호 이력서

Backend Developer

Email: kimmchanho@gmail.com
GitHub: github.com/kimchanho97
Website: kimchanho.site
Portfolio: [PDF](#)

자기소개

장애 상황이나 데이터 불일치가 생길 지점을 먼저 살피는 **Java/Spring 백엔드 개발자**입니다. Kafka 기반 채팅 시스템·Toss Payments 결제 시스템에서 메시지 전달과 외부 연동의 정합성을 다뤘고, 부하 테스트와 모니터링으로 성능 병목을 찾아 개선했습니다.

최근에는 AI 에이전트에 구현을 맡겨 운동 기록 서비스를 개발해 직접 사용하며 운영하고 있습니다. **해결할 문제와 완료 기준은 사람이 정하고, 위험한 변경은 코드로 제한하며, 결과는 검증으로 확인합니다.**

프로젝트

순수웨딩 2.0 | Fullstack

2025.03 ~ 2025.05

웨딩플래너와 예비부부를 투명한 가격으로 연결하는 웨딩 매칭 플랫폼으로, 카카오테크캠퍼스 팀 프로젝트를 **1인 프로젝트로 리빌딩**하여 백엔드 아키텍처와 인프라를 새롭게 구축한 서비스입니다.

Java, Spring Boot, JPA, JDBC, QueryDSL, Kafka, Redis, MongoDB, MySQL, React, JavaScript, Docker, Nginx, AWS (EC2, RDS, S3, ACM, CloudFront, Route 53), GitHub Actions, Prometheus, Grafana, k6

- **Kafka로 채팅·전송 서버를 분리**하고 저장소(MySQL·MongoDB·Redis)를 용도별로 구성해 수평 확장 가능한 채팅 아키텍처를 구현 → 컴포넌트별 저사양 t2.micro 환경의 k6 부하 테스트에서 SLO(p95 응답 시간 1초 이내·유실 0%)를 만족하는 **최대 동시 커넥션 5,200개, 메시지 처리량 1,050 TPS**를 달성
- Toss Payments 결제 승인은 성공했지만 내부 DB 반영이 실패하는 상황에 대비해 **PG 상태를 기준으로 내부 데이터를 복구**하는 흐름을 구현. 일시적 오류는 재시도하고, 승인 결과를 알 수 없는 타임아웃은 상태를 재조회하며, 웹훅과 실패 로그 기반 스케줄러로 비동기 복구
- 클라이언트의 금액 위변조에 대응해 **결제 금액·주문 ID를 서버에서 생성·저장**하고 승인 시 원본과 대조하도록 설계하고, 외부 PG API 호출은 인터페이스로 분리 → 결제 로직의 각 계층에 **단위 테스트 46개**를 작성해 예외 처리 동작을 검증
- 플래너 포트폴리오 목록 조회 API에서 **공통 데이터는 Redis에 캐싱**하고 사용자별 찜 여부는 조회 시 병합. DTO 프로텍션·커서 페이징과 Tomcat 스레드·DB 커넥션 풀 조정을 함께 적용 → k6 부하 테스트에서 **처리량 626 → 1,500 TPS(2.4배), 응답 시간 47%·CPU 사용률 30% 감소**

EasyShift | Backend

2025.02 ~ 2025.03

교대 근무 스케줄 자동 생성과 실시간 변경 관리를 지원하는 스마트 스케줄링 플랫폼으로, 구름톤 트레이닝 팀 프로젝트에서 **기획·설계부터 백엔드 개발 전반을 주도**한 서비스입니다.

Java, Spring Boot, JPA, JDBC, MySQL, H2, JUnit5, JMH

- 교대 근무 배정에서 휴무를 우선 반영하고 연속 근무를 피하도록, 우선순위 큐로 배정 순서를 정하는 **라운드 로빈 알고리즘**을 직접 설계·구현
- JPA 변경 감지로 근무 배정 시 발생하던 수백 건의 개별 업데이트 쿼리를 JDBC batchUpdate 일괄 처리로 전환해 **처리 속도를 5배(102ms → 18ms) 향상**

프로젝트

Notch [서비스 보기](#) | Fullstack · AI 에이전트 활용

2026.05 ~ 진행 중

- 중상급자를 위한 웨이트 트레이닝 기록 PWA입니다. Claude Code·Codex에 구현을 위임하고 기획·설계·검증을 맡아 혼자 개발했으며, **4개월째 직접 사용하며 운영 중**입니다.
- Next.js, TypeScript, PostgreSQL(Supabase·RLS), Drizzle, Vitest, Playwright, GitHub Actions, Claude Code, Codex CLI
- AI 산출물을 변경 성격별로 검증(도메인 로직 TDD, Postgres 통합 테스트, DB·동시성 변경은 독립·교차 리뷰)해 **결함 수십 건을 리뷰에서 적발**. 대표 사례로 컬럼 삭제 시 **ORM의 잔여 참조로 구버전 앱의 운동 기록 시작이 실패할 위험을 발견**, 앱 선택포·컬럼 삭제 순으로 나누고 새 앱을 삭제 전·후 DB에서 통합 테스트로 검증
 - AI가 구축한 CI의 '성공' 표시를 직접 확인해 **인증 준비 누락으로 E2E 전체가 건너뛰어진 채 통과하던 문제를 발견**. 준비가 안 되면 실패하도록 바꾸고 테스트별 계정으로 격리해 가려져 있던 **실패 6건(18개 중)**을 수정, 개발 서버 대신 프로덕션 빌드로 실행해 E2E 시간 5.6분 → 18.7초
 - 완료 조건·수정 범위·검증 방법을 정한 뒤 구현을 위임. **스키마 직접 편집·원격 DB 조작은 흑과 거부 규칙으로 차단**하고 원격 마이그레이션은 main 머지 시 단일 워크플로로만 적용하며, 합의한 파일 수의 2배 이상을 수정하거나 같은 오류가 3회 반복되면 **멈추고 보고하도록 규칙으로 정의**

※ 각 프로젝트의 설계 근거·측정 과정·트레이드오프는 [포트폴리오](#)에 정리했습니다.

학력

- 부산대학교 정보컴퓨터공학부 졸업
(3.54 / 4.5) | 2018.03 ~ 2024.08

자격

- 정보처리기사 | 2026.06
- SQLD | 2026.03

대외활동 및 수상

- 구름톤 트레이닝 | 2024.08 ~ 2025.03
 - 팀프로젝트 우수상
- 카카오테크캠퍼스 1기 FE 과정 | 2023.04 ~ 2023.11
 - 2단계 개인 프로젝트 우수상
 - 아이디어톤 장려상