성진혁

신입 백엔드 개발자 · Java · Spring

jinhyuk9714@gmail.com
github.com/sjh9714
포트폴리오 사이트

소개

동시성 제어와 데이터 정합성을 직접 재현하고 측정해 온 신입 백엔드 개발자입니다.

좌석 예약 시스템에서 락 전략 3종을 같은 조건으로 실측 비교해 중복 판매 0건을 검증했고, 혼합 부하에서 Redis 재고 선차감으로 쓰기 p95를 37ms에서 6ms로 줄였습니다.

Testcontainers 통합 테스트와 k6 부하 테스트로 주장에 근거를 붙이고, 측정하지 못한 항목은 측정하지 못했다고 문서에 남기는 방식으로 일합니다.

프로젝트

좌석 예약 시스템 2026.02 – 2026.05 · 개인 프로젝트

공연 좌석 예약 — 대기열 · 락 전략 · 이벤트 전달 · 설계 · 구현 · 측정 전체

`Java 21, Spring Boot 3.4.1, PostgreSQL 16, Redis 7 · Redisson 3.40.2, Apache Kafka (cp 7.6.0), JPA, Flyway, Testcontainers, k6 v1.5.0`

- 같은 좌석에 100명이 몰릴 때 나던 중복 판매를 락 3종 비교로 막아 oversell 0건, p95는 106–215ms로 실측
- 다른 좌석인데 예매가 서로 실패하던 원인이 잔여석 카운터 한 줄이었음을 밝혀 성공률 40% → 100%
- 매진된 좌석 요청까지 DB를 잡던 것을 Redis에서 미리 걸러 쓰기 p95 37ms → 6ms, 총 RPS 969 → 1,005
- 결제·만료 race와 멱등 replay, 대기열 토큰 우회를 시나리오로 재현해 체크 594/594 통과
- JWT 인증(가입·로그인)과 내 정보 조회, 데모 계정 원클릭 로그인
- 콘서트 목록·회차·좌석 배치 조회와 예매 내역 조회·취소, 대기열 순번은 SSE로 전달
- Transactional Outbox로 커밋과 발행을 분리하고, 재고 정합성 대조와 Kafka DLT 수동 replay를 운영용 엔드포인트로 제공

실시간 채팅 서버 2026.02 — 2026.05 · 개인 프로젝트

다중 인스턴스 채팅 — 메시지 영속화 · fan-out · 전달 검증 · 설계 · 구현 · 측정 전체

`Java 21, Spring Boot 3.4.3, WebSocket · STOMP, Apache Kafka 3.9.0 (KRaft), Redis 7 Pub/Sub, PostgreSQL 16, JPA, Testcontainers, k6 v1.5.0`

- 채팅방 목록이 방 개수만큼 쿼리를 날리던 것을 JPQL 프로젝션과 IN 배치로 모아 방 50개 기준 101회 → 3회 고정
- 커서 페이지네이션·멱등성·unread 등 핵심 쿼리를 EXPLAIN ANALYZE로 분석해 인덱스 5개 설계, 이미 커버되는 3개는 근거를 적고 추가하지 않음
- DB 커밋이 끝난 뒤에만 브로드캐스트하도록 순서를 강제 — 50명이 두 인스턴스에 나뉜 3회 반복에서 기대 4,900건 전부 도착, 누락·중복·순서 위반 0건
- 현재 커밋에서 200 VU 조회 부하를 3회 반복 재측정해 RPS 1,806–1,940 · p95 129–133ms · 39.8만 요청 중 HTTP 실패 0건 확인
- JWT 인증(가입·로그인)과 사용자 검색, 1:1·그룹 방 생성과 참여
- 커서 기반 메시지 조회와 읽음 처리, 재접속 시 마지막 수신 ID 기준으로 놓친 구간 보충 조회
- Redis 패턴 구독이 수신 채널명을 목적지로 쓰던 것을 payload 기준으로 고쳐 다른 방으로 새던 메시지를 막고 단위 테스트로 고정

FinMate — 청년 금융 온보딩 2026.04 — 2026.07 · 4명 — 기획 2 / 풀스택 1(본인) / 데이터 1

소셜 핀테크 — 원장 파생 지표 · 또래 비교 · 미션 · 풀스택 — 앱의 프론트·백엔드 단독

`React 19, TypeScript strict, Vite, Tailwind CSS v4, Motion v12, React Router 7, html-to-image, Java 21 · Spring Boot 3.5, PostgreSQL 16, Flyway · Testcontainers, fal.ai (FLUX)`

- 또래 비교가 매 요청 원장 88만 행을 다시 세던 것을 사람×월 사전 집계로 접어 p50 32.5ms → 0.72ms
- 고치기 전에 먼저 재서 개인 화면은 p95 0.96ms로 문제가 아님을 확인하고, 인덱스(50MB·24.0ms) 대신 사전 집계(1.9MB·0.72ms)를 선택
- 외부 이미지 생성 API가 3~6초 걸리는 동안 행 잠금이 풀리던 것을 상태 값으로 선점하게 바꿔 두 인스턴스가 같은 작업을 가져가지 못하게 함
- 합성 데이터셋 2,000명 887,002행을 JDBC 배치로 적재(34,490행/초)하고 원장·프로필 정합성을 통합 테스트로 고정
- 하루의 거래에서 그날의 주인공(소비·저축·투자)을 뽑아 AI 그림일기 한 장으로 기록 — 하루 한 장 멱등성, 3회 재시도, 응답 없는 작업 회수
- 예산 챌린지 판정을 저장하지 않고 매번 원장에서 세고, 포인트는 잔액이 아니라 원장으로 쌓아 이중 지급 차단 — 또래 비교는 금액을 구간으로만 내보내고 20명 미만 그룹은 만들지 않는다
- 화면의 모든 수치를 거래 원장에서 파생하는 순수 셀렉터로 통일하고, 차트는 라이브러리 없이 SVG로 직접 구현해 번들에 차트 의존성 0개

배리어프리 길찾기 (My ETA) 2026 · 해커톤 · 7명 — 개발 4명 / 기획·리서치 3명 (24시간 해커톤)

교통약자 대상 경로 안내 — 개인화 ETA · 접근성 데이터 통합 · 백엔드 · 프론트엔드 — 커밋 기준 최다 기여 (전체 코드의 91%)

`Python 3.12, FastAPI, Pydantic, HTTPX, TMAP API, 서울 공공데이터, Kakao Local, pytest, OpenAPI 3.1`

- 경로에 접근성 정보를 붙일 때 구간마다 외부를 차례로 묻던 것을 같이 물어 대기 3,097ms → 52ms, 외부 호출 60회 → 20회
- 표준 보행속도로 낸 ETA가 교통약자에게 항상 틀리던 것을 이동 유형·보조기구 프로필과 안내 중 모은 실제 속도로 보정
- 확인 못 한 시설 정보를 '있음'으로 채우던 것을 '확인 안 됨'으로 그대로 응답하도록 상태를 정의
- 경로 이탈과 대중교통 놓침을 감지해 현재 위치를 새 출발점으로 경로와 ETA를 다시 계산
- TMAP 경로·서울 버스·지하철 실시간 도착·엘리베이터·장소 검색 등 외부 API 5종을 provider 어댑터 계층으로 통합
- 이동 프로필 조회·수정과 목적지 장소 검색, 대중교통·도보·택시 유형별 경로 검색과 상세 조회
- 안내 세션 시작·위치 갱신·재탐색·완료 — 완료 시 실제 속도를 프로필에 반영하고, 원본 좌표는 안내 중 계산에만 쓰고 영구 저장하지 않는다

활동

하나금융그룹 × SK텔레콤 Tech4Good 2026 — 해커톤 — 교통약자 내비게이션 My ETA 개발 (15조 피프틴피프틴)

하나금융그룹 청년 금융인재 양성 과정 (하나 파워온) — 금융·데이터 교육 과정 수료 활동

학력

가톨릭대학교 성심교정 컴퓨터정보공학부 · 2021.03 – 2028.03 (졸업 예정)