

임기호 | **Software Engineer**

Full-Stack · Platform · Integration

lasid84@gmail.com | 010-5496-0663 | 이력서: limgiho.github.io

경력 기술서

2022.01 ~ 현재 | (주)케이더블유이코리아 | 전략개발부 | 차장 | 국제 종합 물류 기업

[담당업무] 사내 시스템 개발 및 운영 리드 (Lead Software Engineer)

[주요 프로젝트]

1. Docker Swarm 기반 CI/CD 표준 배포 체계 구축 | 2025.11 ~ 2025.12

- GitLab CI/CD, Docker, Docker Swarm, Portainer
- 배경
 - 기존 PM2 중심의 수동 배포는 서버 상태와 환경 차이에 따라 결과 재현이 어려웠고, 테스트 누락 방지와 배포 이력 추적·롤백 기준도 미흡했습니다.
- 성과
 - 테스트 강제: 푸시 트리거 테스트 -> 빌드 -> 배포 3단계 파이프라인에서 테스트 실패시 후속 단계 자동 차단
 - 불변 이미지·재현성: 브랜치-SHA/디제스트 태깅으로 서버 상태 의존 제거, 동일 이미지 재배포 가능
 - 이력·추적성: 빌드/배포 로그와 아티팩트로 변경 원인 신속 파악
 - 운영 가시성(Portainer): 서비스/태스크 상태·로그를 UI에서 실시간 확인(재시작 루프·헬스 실패 즉시 파악)

2. Backend 서비스 리팩토링(Express → NestJS) | 2025.06 ~ 2025.10

- [Node.js](https://nodejs.org), NestJS, Prisma, PostgreSQL, Redis, BullMQ
- 배경
 - 기존 Express 기반 서비스는 자유도가 높아 기능 확장 시 일관된 코드 구조 유지가 어렵고 테스트 코드 도입도 제한적이었습니다. 이러한 문제를 해결하고 유지보수성과 서비스 안정성을 근본적으로 높이기 위해 리팩토링을 시작했으며, NestJS의 구조화된 아키텍처 도입을 통해 팀 코드 컨벤션을 확립하고 TDD 기반 개발 환경을 구축했습니다.
- 성과
 - 코드베이스를 도메인별 모듈로 분리하여 신규 기능 개발 및 유지보수 효율성 증대
 - Redis와 BullMQ를 도입해 무거운 작업을 비동기 배치 파이프라인으로 분리하고, 인증 데이터 캐싱으로 DB 부하를 절감
 - DI와 데코레이터 등 NestJS 구조를 도입, 인증·로깅 등 공통 로직 재사용성 및 확장성 강화
 - 단위·통합 테스트를 도입해 변경 영향 범위를 사전에 검증할 수 있는 구조를 마련

3. 웹 기반 통합 사내 시스템 구축 | 2023.09 ~ 2025.05

- Express, Next.js, Turborepo, PostgreSQL, PL/pgSQL, PM2, Worker Threads
- 배경
 - 파편화된 사내 시스템을 하나로 통합하여 중복 업무를 감소, 업무 효율을 향상시키고, 장기적으로 유지보수성을 확보할 수 있는 회사의 메인 시스템 구축을 목표로 프로젝트를 시작했습니다.
- 성과
 - 15종의 파편화된 레거시를 5개 핵심 비즈니스 도메인(항공/해운 수출입 및 재무) 중심의 단일 웹 플랫폼으로 통합
 - 여러 레거시 시스템(C#, Oracle Forms, React, VB)을 Next.js & Express 기반 웹 시스템으로 통합 전환
 - Monorepo 기반 공용 라이브러리 사용으로 유지보수성 향상
 - 관세청 API·스크래핑 자동화로 수작업 10분 → 5초 처리
 - 공통 배포 기준과 개발 구조를 정립해 협업 시 변경 관리와 운영 방식의 일관성을 확보

4. RPA 프로젝트 (삼성 NERP 연동) | 2024.08 ~ 2024.12

- C#(.NET 4.5), WinForms, OCR, MySQL
- 배경
 - 삼성 시스템의 데이터 입력 작업에 과도한 시간이 소요되어 업무팀의 야근이 빈번하게 발생하던 문제를 해소하기 위해, 자동화를 목표로 프로젝트를 시작했습니다.
- 성과
 - [문서 검증·입력 자동화로 월 1.200시간 절감](#)
 - 실시간 처리 도입으로 입력 누락과 오입력 가능성을 줄임

5. 세금계산서 프로그램 리뉴얼 | 2023.01 ~ 2023.07

- C#(.NET 6), gRPC, WinForms, PostgreSQL, PL/pgSQL
- 배경
 - Oracle Forms 기반 레거시 시스템은 기술지원 종료로 유지보수가 어려워 운영 리스크가 누적되었고 Oracle 의존으로 비용 부담과 확장 제약이 있어 .NET 6 기반으로 리뉴얼과 DB 전환을 추진했습니다.
- 성과
 - 레거시 시스템 리뉴얼로 유지보수성·확장성 강화
 - Oracle → PostgreSQL 전환으로 라이선스 비용 절감
 - 1,000만 건 데이터 마이그레이션 자동화 및 정합성 검증
 - 스마트빌 연동으로 국세청 즉시 신고 및 매입 데이터 실시간 처리

[퇴직 또는 이직사유]

- 사내 시스템 개발과 운영 자동화 경험을 바탕으로, 더 큰 규모의 서비스 환경에서 역량을 확장하고자 이직을 결심했습니다.

[담당업무] MES시스템 개발 및 운영

[주요 프로젝트]

1. 운반비 전산화 구축 | 2020.04 ~ 2020.07

- C#(.NET 4.5), ASP.NET, WinForms, WCF, Oracle, PL/SQL
- 배경
 - 수기 작성·수작업 중심의 운반비 관리로 인해 관리 리스크와 비용 투명성 저하 문제가 발생하여, 이를 해결하고 프로세스 표준화 및 전산화 구축을 목표로 프로젝트를 시작했습니다.
- 성과
 - 8개 공장의 운반비 프로세스를 전산화해 수작업을 줄이고 처리 표준화를 확보
 - SAP 연동 도입으로 실시간 운반비 처리 및 비용 투명성 확보
 - 최우수사원 표창

2. MES시스템 - SAP 연동 인터페이스 개발 | 2019.01 ~ 2019.12

- C#(.NET 4.5), ASP.NET, WinForms, WCF, Oracle, PL/SQL
- 배경
 - 회사의 핵심 ERP로 SAP를 새롭게 도입하면서, **MES**와 **SAP** 간의 생산·납품·재고 정보 연계를 자동화하여 운영 효율을 높일 필요가 발생했습니다. 기존에는 수작업 전송과 중복 입력으로 인해 지연·오류 가능성이 컸기 때문에, 이를 제거하고 실적 데이터를 실시간으로 연동하는 인터페이스 구축을 목표로 프로젝트를 수행했습니다.
- 성과
 - ERP(SAP) ↔ MES 간 생산·납품·재고 이동 실시간 데이터 연동 구현
 - 기준정보 자동 마이그레이션 적용으로 운영 오류 및 수작업 감소

3. 제천·슬로바키아 공장 MES 업그레이드 | 2016.04 ~ 2017.12

- VB6 -> C#(.NET 4.5), ASP.NET, WinForms, WCF, Oracle(9i -> 11g), PL/SQL
- 배경
 - 기존 MES가 VB6 기반으로 구축되어 기술 지원 종료, 컴포넌트 제한, 개발 환경이 **Windows 7** 이하로 고정되는 등 유지보수와 기능 확장이 어려운 상황이 되었고, 국내외 공장의 운영 효율성을 높이기 위해 **C#** 기반에서의 리뉴얼 및 데이터베이스 업그레이드가 필요해 프로젝트를 시작했습니다.
- 성과
 - 제천·슬로바키아 공장의 MES 리뉴얼 및 운영 안정성 확보
 - LOT 단위 추적 시스템 구축으로 품질 관리 프로세스 고도화
 - 과장 특진

4. SPC 이상현상 알림 시스템 개발 | 2015.09

- VB6, Oracle PL/SQL, TCP Socket 통신
- 배경
 - 생산 품질 이슈를 사후 대응이 아닌 실시간·선제적으로 관리하기 위해, 공정 데이터를 기반으로 이상 징후를 자동 감지하고 즉시 알림을 제공하는 SPC(Statistical Process Control) 알림 시스템을 구축했습니다. 공정 설비와의 실시간 연계를 위해 TCP 소켓 기반 알림 모듈을 단독으로 설계·개발했습니다.
- 성과
 - 일평균 100만 건 생산 데이터 실시간 분석 및 7가지 이상 징후 감지 구현
 - SMS 및 현황판 알림 연동으로 품질 이슈 사전 대응 체계 마련

[퇴직 또는 이직사유]

- 수도권 근무를 희망하여 이직

개인 프로젝트

1. 댓글 필터 (Android · Web)

- Android, Expo, React Native, TypeScript, WebView
- (웹) Next.js, Transformers.js, WebLLM, Web Worker
- 1인 개발로 앱 기획, 구현, 배포까지 수행했습니다. 네이버 블로그 글을 **WebView**로 열어 온디바이스에서 댓글 가시성을 조정하고, 좋아요 기준 필터와 차단 사용자 숨김 기능으로 유익한 댓글을 우선 노출할 수 있도록 했습니다.
- Next.js 웹 버전을 신규 개발해, Transformers.js + WebLLM을 Web Worker로 구동해 브라우저 온디바이스에서 댓글 분류와 본문 요약을 수행하도록 구현했습니다.
- [댓글 필터 \(Play Store\)](#) | [개발기](#)

2. HealthDog

- Android, Expo, React Native, TypeScript, Google OAuth, Firebase, Supabase, AdMob, RevenueCat
- 1인 개발로 운동 기록·조화·통계 기능 중심의 앱을 구현했습니다. Google OAuth, 데이터 저장 및 동기화(Supabase, Firebase), 광고(AdMob), 구독 결제(RevenueCat) 등 운영에 필요한 외부 연동을 적용했습니다.
- [HealthDog 앱 \(Play Store\)](#)

3. 액션독: 3라인 퍼즐 (Android)

- React Native, Expo, TypeScript, Google OAuth, Supabase, AdMob
- 1인 기획·개발 및 스토어 론칭한 3라인 매치 퍼즐 게임으로, 블록 하강·매치 판정·콤보 연쇄·게임오버 등 게임

코어 엔진을 도메인 주도 아키텍처(**4계층 분리**) 로 직접 설계·구현했습니다.

- Google OAuth 로그인, Supabase 기반 글로벌 점수 랭킹, AdMob 광고 연동을 적용했습니다.
- [액션독 \(Play Store\)](#)