

# 기술의 경계를 넘어 조직의 문제를 직접 해결하는 전종현입니다



연락처.

hoogom88@gmail.com

학력.

서울과학기술대학교 (2016.03 ~ 2023.02)

ITM 전공

전공학점 4.02 / 4.5

개인 사이트.

[Github ↗](#)

[Notion ↗](#)

## Introduce

- Flutter 기반 모바일 앱을 분석–설계–개발–배포–운영까지 전 과정을 단독으로 수행하며, 출시 이후 운영 병목과 이슈까지 해결해 온 경험을 보유하고 있습니다.
- 앱 개발 과정에서 백엔드·데이터·QA 영역의 리소스 부족으로 개발 흐름이 지연되는 상황을 경험했고, 이를 해소하기 위해 필요한 영역의 기술을 학습하며 역할 범위를 확장해 왔습니다.
- 조직과 서비스의 문제를 먼저 발견하고, 제안에 그치지 않고 직접 구현까지 책임지는 방식으로 일해 왔습니다.
- 반복적인 업무와 수작업 의존도를 줄이기 위해 다양한 방식으로 AI를 활용하여, 개발·QA·데이터 처리 프로세스를 자동화하고 팀 생산성을 개선했습니다.

## Career

### Enkorwithus (2024.08 ~ 재직중)

외국인 대상 단기 임대 플랫폼 'EnkoStay' 모바일 앱 전체를 단독으로 설계/개발하고, 백엔드 API 개발, 백오피스 서비스 개발을 병행

### EnkoStay App 설계 및 개발

- 앱 서비스의 설계 단계부터 개발, 출시 이후 운영까지 전 과정을 단독으로 책임지며 실 사용 서비스로 안정화
- 2025년 6월 24일 iOS/Android 동시 출시, 누적 다운로드 30K +, MAU 7K +
- FeatureFlag / ServerDriven UI 설계로 심사-배포 없이 UI 및 로직을 긴급 변경 가능한 구조 확립
- 예외 전파와 처리를 2계층으로 분리·모듈화하여 통일된 에러 처리 패턴 확립
- Sentry 기반 에러 트래킹 고도화로 빠른 에러 파악 및 해결이 가능한 환경 구축
- 다국어 지원, 네트워크 상태 및 에러 대응 등 다양한 환경의 사용자를 고려한 사용성/안정성 확보
- Riverpod 상태 관리를 기반으로, 과도한 계층 분리를 피하고 AI 친화적인 아키텍처 설계
- GitHub Action, Fastlane을 활용한 CI/CD 파이프라인 구축
- Amplitude, GA, Braze 등 다양한 이벤트 트래킹 및 CRM 툴 연동
- 캐싱, 프리로딩 등을 적용하여 빠르고 부드러운 UX 구현
- 주요 로직 모듈화로 재사용성 및 유지보수성 개선

## 백엔드(NestJS) API 개발 병행

- 조직 내 백엔드 리소스 부족으로 인한 **개발 병목을 해소**하기 위해, 자발적으로 백엔드 스택(NestJs)을 단기간에 학습 후 즉시 실무 투입
- 주요 페이지 개선 업무에서 백엔드 파트를 담당하며, **6개월간 40개 이상의 API 개발 병행**
- 랭킹 시스템 구축을 담당하여 **전 과정**을 주도적으로 수행
  - 데이터 분석 → 기획 → DB 설계 → API 개발 → A/B 테스트
- 운영 중 발생하는 백엔드 이슈·버그 대응 및 데이터 관련 요청을 상시 처리하며 서비스 안정성 유지에 기여

## 자연어 기반 데이터 분석 시스템 구축

- 전 직원이 자연어로 데이터를 조회·분석할 수 있는 시스템을 **자발적으로 제안하고, 설계부터 개발·운영까지 엔드 투 엔드로 단독 구축**
- 사용자와의 자연어 대화를 기반으로 '의도 파악 → 조건 명확화 → SQL 자동 생성 → 결과 해석 → 대시보드 생성'의 **5단계 분석 파이프라인 설계·구현**
- 테이블 간 제약사항과 비즈니스 정책 문서가 불완전한 환경에서, 백엔드 **코드 기반 프롬프트 자동 업데이트 파이프라인**을 구축하여 **스키마 변경 사항을 자동 반영**
- Redash 연동을 통해 **SQL·테이블 스키마 지식 없이도 누구나 팀별 맞춤형 대시보드를 생성할 수 있는 환경 확립**
- Slack 알림, 자동 버전 관리, 에러 대응, 로그 자동 저장 등 운영·유지보수 체계까지 포함한 프로덕션 수준 시스템 완성
- 전 직원이 즉각적으로 데이터를 확인할 수 있는 데이터 기반 의사결정 환경 구축
- 그 결과, 개발자가 직접 처리하던 데이터 확인 요청 **80% 이상 감소, 주 평균 30건** 이상의 데이터 조회가 해당 시스템을 통해 처리됨

## 프로세스 개선 및 생산성 향상

- AI 기반 '코드 -> QA 계획서/테스트 시나리오 자동 생성'으로 QA 준비 리소스의 **90% 이상 대체**
- AI 기반 '코드 ->이벤트 택소노미 문서 동기화 및 불일치 감지' 시스템 구축
- 'AI를 활용한 하우스 추천 기능' 사내 해커톤 기획 및 실제 기능으로 도입

## 소만사 인턴 (2022.07 ~ 2022.08)

데이터보호 전문 기업 소만사에서, **가상 데스크톱 인프라(VDI)** 접속 앱 프로토타입 개발 업무를 담당

- RDP 관련 오픈소스 조사 및 기능 분석
- 'FreeRDP' 오픈소스 프로젝트 구조 및 동작 방식 분석
- 사내 기술 문서 검토 및 서버 요청 절차 확인
- 사내 서버의 프로토콜에 맞춰 'Socks5 Proxy 연결' 기능 구현
- RSA 암호화를 통한 로그인 및 로그인 계정으로 접속 가능한 VD 목록 표시 구현
- 오픈소스의 '클립보드 Redirection', '가상 키보드 표시' 버그 해결
- 패킷 분석, 메모리 덤프 분석을 통한 오류 해결
- VDI 사업부 월말 회의에서 최종 프로토타입 공유 및 검토

# Experience / Project

## 오름 캠프 Flutter 2기 (2024.03 ~ 2024.05)

- Dart, Flutter 기초 학습
- 다른 참가생을 대상으로 매주 기술 발표 세션 자발적 진행
- 팀 프로젝트 - Weaco (2024.04 ~ 2024.05) | [GitHub](#) / [Google PlayStore](#)
  - 날씨 기반 외출복(OOTD) 공유 서비스
  - 캐싱/리사이징/프리로딩 세 가지 전략을 조합하여 이미지 로딩을 3 → 1초로 단축
  - 클린 아키텍처를 적용해보았으나, 빠른 기능 변경이 잦은 환경에서는 오히려 비효율적임을 경험하고 프로젝트 상황에 맞는 아키텍처 선택의 중요성을 체득
  - 외부 라이브러리 호출을 전용 클래스로 격리하여 의존성을 한 곳으로 모으고, 교체 및 제거가 용이한 구조 설계
  - 의존 객체를 **Mock**으로 주입할 수 있도록 설계하여, 테스트 가능한 구조를 확보하고 객체 간 결합도를 낮춤
  - 기술 스택: Flutter, Firebase, Provider, Get It, Hive

## 네이버 부스트 캠프 (2021.07 ~ 2021.11)

- CS 지식 및 Kotlin, Android 기초 학습
- 자기 주도적 학습 및 지속 가능한 학습 습관 형성
- 팀 프로젝트 - 그물잠(2021.10 ~ 2021.11) | [GitHub](#)
  - 낚시 관련 정보 제공 및 기록 공유 서비스
  - 낚시 환경의 네트워크 불안정성을 고려하여 오프라인 캐싱 설계
  - 여유 메모리 확인/네트워크 상태 체크 등 세밀한 분기 처리로 오류 발생을 최소화
  - BindingAdapter와 양방향 DataBinding을 활용하여 MVVM 패턴의 장점을 살리고 View의 UI 로직을 최소화
  - 콜백 전달 패턴의 가독성 저하와 메모리 누수 위험을 분석하고, Response 래퍼 클래스를 통한 결과 전달 방식을 대안으로 도출
  - 스톱워치의 시간 누적 방식이 Foreground Service 전환 시 오차를 유발함을 파악하고, 차이 계산 방식으로 변경하여 해결
  - 기술 스택: Android, Kotlin, Hilt, Firebase(FCM), Room, Coroutine

# Qualifications

- SQLD (2024.03.09)
- 정보처리기사 (2021.06.02)