

주로 헬스케어 분야에서 커리어를 쌓은 개발자로, 의료영상·생체신호 AI 제품의 프론트엔드부터 백엔드·배포까지, 규제 대상 의료기기 소프트웨어가 실제 병원에서 동작하는 전 과정을 경험했습니다.

눈앞의 버그를 고치는 것보다 같은 문제가 다시 생기지 않는 구조를 만드는 일에 관심이 많습니다. 제품별로 흩어져 있던 프론트엔드를 모노레포와 디자인 시스템으로 통합하고, 수동으로 반복하던 검증을 CI 프로세스로 옮기고, AI 에이전트를 팀의 개발 규범 안으로 편입시키는 작업을 해왔습니다.

낮선 문제는 깊이 파고들고, 판단의 근거는 문서로 남겨 동료와 같은 기준을 공유합니다.

기술 스택

LANGUAGES	TypeScript · Python · JavaScript · SQL
FRONT END	React · React Query · Zustand / Recoil · React Hook Form · styled-components · Vite · pnpm
BACKEND	FastAPI · Node.js · GraphQL · WebSocket · SQLAlchemy · Celery
DATA / INFRA	PostgreSQL · Redis · Docker · nginx · GitHub Actions
AI TOOLING	Claude Code (CLAUDE.md · Subagents · Skills) · Cursor · Codex

경력

뷰노 2020 ~ 현재

- AI Native 개발 워크플로우 및 팀 개발 규범 정립
- DeepECG Univ (with Backend Monorepo) 개발 및 운영
- Frontend Monorepo 개발 및 운영
- LungCT AI 개발 및 운영
- 소프트웨어 현장 설치 지원 및 유지보수

Performance By TBWA 2019 ~ 2020

- 마케팅 자동화 툴 개발 및 운영
- 웹 로그 수집 및 분석
- 이벤트 웹 페이지 제작

네오젠소프트 2016 ~ 2019

- 헬스케어 관련 웹 어플리케이션 개발 및 운영
- HL7 FHIR 기반 진료정보 연동 및 건강 데이터 관리 서비스 개발

○ 뷰노

2020 ~ 현재

AI Native 개발 워크플로우 및 팀 개발 규범 정립

2026 ~

AI 에이전트를 팀 개발 프로세스에 편입시키기 위한 개발 규범·문서 체계·서브에이전트 파이프라인 설계

- 프로젝트 전반의 개발 규범을 CLAUDE.md로 단일화하여 정립. harness 설정, 아키텍처 문서 참조 규칙, 팀 작업용 work 문서 작성 규칙을 명문화해 AI 에이전트가 항상 동일한 컨텍스트에서 작업하도록 구성
- 팀 내 프론트엔드 담당과의 협업 비용을 줄이는 Claude Code 스킬을 직접 제작. 구현한 백엔드 코드로부터 API 명세를 작성하는 스킬, Custom Error Code를 마크다운 문서로 정리하는 스킬을 만들어 명세·에러 규격을 코드와 동기화된 상태로 공유
- 완료 작업을 work 단위로 진행 내역·follow-up 항목을 남기는 규칙을 세워, 팀원 각자가 사용하는 AI 에이전트가 동일한 이력을 참조하도록 구성. 개인별 AI 사용으로 갈라지기 쉬운 구현 컨벤션과 판단 기준을 팀 차원에서 수렴
- PM·Architect·Engineer·QA를 서브에이전트로 역할 분리하고, 대화 → outline → 상세 설계 문서 → 사람의 검토 → 서브에이전트 개발 위임 → 결과 코드 검토로 이어지는 개발 파이프라인을 설계. 코드 생성 이전 단계에 사람의 검토 지점을 두어 코드의 품질을 통제
- Cursor·Codex를 동일 룰셋으로 비교 평가한 경험을 바탕으로 Claude Code 기반 개발 규범을 정립

DeepECG Univ (with Backend Monorepo) 개발 및 운영

2026 ~

의료진이 업로드한 ECG(심전도)를 파싱·비동기 채점해 결과를 제공하는 FastAPI Clean Architecture 백엔드(uv workspace 모노레포)를 설계·구축

- ECG 장비 데이터를 파싱·저장하고, 파형 데이터를 분석 엔진에 전달해 AMI·LVSD·Kidney·HK 등 소견별 예측 결과를 score와 함께 제공하는 시스템의 백엔드를 주도적으로 설계·구축
- FastAPI Clean/Hexagonal Architecture(Domain·Adapter·Core 4계층, Protocol 기반 Repository)로 시스템 구조를 설계하고, uv workspace 모노레포·공통 패키지·Database와 신규 App 확장용 boilerplate·Docker 배포 기반을 구성
- 8개 벤더·약 12종 장비의 ECG 포맷을 파싱하는 파서를 Registry·Factory 패턴으로 설계하여, 벤더별 포맷을 표준 데이터 구조로 정규화하고 신규 장비를 최소 변경으로 확장 가능하도록 구성
- 기존 4개 엔진에 순차적으로 분석을 요청하던 구조를 Redis broker·Celery 기반 병렬 처리로 개선하여 채점 응답 시간을 기존 대비 1/4 수준으로 단축
- 외부 장비 연동용 System API(ECG 파서·업로드, 분석 score·결과 전달)와 환자·통계·설정 등 다수 도메인 API를 직접 구현
- JWT 기반 인증/보안 기능 개발 (토큰 버전 회전, 로그인 잠금, 이용약관 최초 동의 처리 등)

- PHI(환자정보) 컬럼 암호화(pgcrypto)·감사 로그·사이버보안 테스트 스위트, SBOM(CycloneDX) 공급망 보안 및 OSS 라이선스 리스팅(Dependency-Track 연동) 등 의료 데이터 보안·컴플라이언스 요구사항을 구현
- 엔진 병렬화 방식·이벤트 루프 선택 등 주요 아키텍처 의사결정의 비교 검토 과정을 문서로 남겨 팀의 판단 근거로 공유

Frontend Monorepo 개발 및 운영 2022 ~ 2025

여러 제품의 Frontend를 pnpm workspace 모노레포로 통합하고, 사내 디자인 시스템 개발 및 운영

- 기존 제품별로 Backend+Frontend가 함께 묶여 있던 구조에서 Frontend 팀이 분리됨에 따라, 여러 의료영상 AI 제품(Fundus AI·DeepBrain·Chest X-ray·LungCT 등)의 Frontend를 pnpm workspace·Vite 기반 모노레포로 통합
- 여러 제품에서 공통으로 사용하는 사내 패키지를 개발하여 재사용성 확보. HTTP 통신·토큰 관리(fetch-manager), WebSocket 관리(websocket-manager), 날짜 유틸(date-manager), 공용 커스텀 Hook 등
- 디자인팀과 협업하여 사내 디자인 시스템(Atomic Design 기반)을 구축하고, React 공용 컴포넌트를 개발하여 제품 간 UI 일관성 확보
- 각 제품은 React·React Query·Zustand(구 Recoil에서 전환)·React Hook Form·Cornerstone.js를 공통 스택으로 사용하여 개발 방식을 표준화
- GitHub Actions 기반 CI/CD 프로세스를 정립. PR 시 변경된 앱·공용 패키지만 감지해 lint·typecheck를 수행하고, 개별적으로 수동 실행해왔던 검증 스크립트를 파이프라인에 편입해 자동 실행되도록 전환. release 태그를 기준으로 제품을 식별해 빌드·라이선스 검사·릴리즈까지 이어지는 배포 흐름 구성
- SonarQube 정적 분석의 Bug·Code Smell을 처리 가능한 단위로 묶어 개선(React Hooks 규칙 위반 등 런타임 리스크 우선)
- 차트 라이브러리 도입 시 후보 라이브러리로 프로토타입을 구현해 장단점을 비교하고 최종 선정에 기여
- 상시 코드 리뷰에 리뷰어로 참여하여 여러 제품 간 구현 방식의 편차를 줄임

Lung CT AI 개발 및 운영 2022 ~ 2025

흉부 CT 폐결절 검출 및 분석 AI 웹 어플리케이션 개발 및 운영

- React·Cornerstone.js 기반 흉부 CT DICOM 판독 뷰어를 개발하여, 전처리한 DICOM 영상을 VUNO Lung CAD 딥러닝 모델에 전달하고 반환된 폐결절 검출 결과(결절 목록·영상 내 위치·Segmentation)를 시각화
- Cornerstone3D 출시에 맞춰, 2D Cornerstone에서 픽셀 조작으로 3축(axial·coronal·sagittal) 단면을 구현하던 기존 뷰어를 Cornerstone3D 기반으로 마이그레이션
- 기존 GraphQL(Apollo Client) 기반 프론트-백엔드 통신을 REST API 구조로 전환
- WebSocket 이벤트별 갱신 정책을 설계. 데이터 변경 이벤트를 페이지·발신 사용자에게 따라 전체 refetch / 상태값만 갱신 / 사용자 확인 후 갱신으로 분기해, 불필요한 재조회 없이 화면 일관성 확보

- 동시 판독 시 발생하는 판독 결과 충돌을 막기 위해 백엔드와 Lock 획득·해제 규약 (WebSocket 메시지 및 Release API)을 협의하고, 프론트엔드 측 처리를 구현
- 장시간·연속 판독 시 뷰어가 무한 로딩되거나 강제 종료되던 문제를 추적해 두 가지 원인을 해소. 영상 렌더링용 볼륨과 세그멘테이션용 볼륨이 각각 수백 MB 규모의 캐시를 이중으로 점유하던 것을 확인해 세그멘테이션을 표시하지 않는 배포 환경에서는 해당 볼륨을 생성하지 않도록 조건화하고, 뷰어 언마운트 시 revokeObjectURL 미호출과 DICOM 상태 미초기화로 메모리가 누적되던 부분은 cleanup 처리로 제거
- Segmentation Threshold 변경 시 매 조작마다 좌표를 전역 상태에 기록하던 방식을 viewport의 present/reference를 저장한 뒤 camera reset 시점에 복원하는 방식으로 재설계하여, 임계값 변경 후에도 zoom·pan 등 판독 컨텍스트가 유지되도록 개선
- PSP 파트너 연동을 위한 Longitudinal(추적검사) 신규 API를 설계하여, baseline·follow-up CT 간 결절 매칭과 시계열 변화 추적을 지원

소프트웨어 현장 설치 지원 및 유지보수 2020 ~ 2022

- Fundus AI·Chest X-ray·DeepBrain·LungCT 등 다양한 뷰노 의료 AI 제품을 국내외 여러 병원에 데모·설치·배포하고, 운영 중 발생하는 장애에 대응
- 병원마다 상이한 DICOM 태그 규격에 대응하여 제품의 영상 데이터 인터페이스를 정합하고, 각 병원 PACS 운영환경과의 안정적 연동을 확보

○ Performance By TBWA

2019 ~ 2020

마케팅 자동화 툴 개발 및 운영 2019 ~ 2020

- 마케터들의 업무 효율 상상을 위해, 고객사에게 제공하는 리포트 작성 및 포털 사이트 키워드 입찰 등의 업무 자동화 툴 개발
- Excel 자동화 업무를 위한 Excel 관련 자체 프레임워크 제작 및 활용

웹 로그 수집 및 분석 2019 ~ 2020

- Google Analytics, Facebook Pixel 등 다양한 Web Tracker를 활용해, 웹 사이트 이용자의 로그를 수집하며, 수집한 데이터를 유의미한 마케팅 전략을 세울 수 있도록 데이터를 가공하여 제공

이벤트 웹 페이지 제작 2019 ~ 2020

- 고객사의 이벤트 및 프로모션 웹 페이지 제작

개인혈당기기 Accu-Chek QC Solution 2018

혈당 측정기기의 QC 측정결과를 저장하고 모니터링할 수 있는 웹 어플리케이션

- QC Solution Database 설계 및 Stored Procedure 개발
- QC Solution 모니터링 페이지 기획, 개발 및 유지보수

Health4U 리뉴얼 2016 ~ 2018

진료정보를 조회하고 건강관리 서비스를 제공받을 수 있도록 제공하는 웹 어플리케이션

- Hybrid App을 위한 반응형 웹 어플리케이션 개발
- HL7 FHIR 기반으로 병원 EMR의 진료기록·검사결과·투약 처방 내역을 조회하는 기능을 개발
- Samsung S-Health 및 iOS 건강 앱과 연동하여 개인 건강 데이터를 수집 및 저장할 수 있는 웹서비스 개발
- Health4U 내 건강수첩 기능을 위한 Database 테이블 설계 및 Stored Procedure 개발

건강지킴이 연구과제 2016

환자 별 건강 데이터를 수집하여 차트로 시각화하고, 목표 수치를 처방하는 웹 어플리케이션

- Hybrid App을 위한 모바일 웹 어플리케이션 개발
- 환자별 건강 데이터를 차트로 시각화하는 모니터링 대시보드 웹 어플리케이션 개발
- Samsung S-Health 데이터를 병원 EMR과 연동하기 위한 웹 서비스 개발