

# 강태욱 경력기술서

## AI 자동화 엔지니어(Agent Engineer) 지원용

Email: woogi.dev@gmail.com Phone: +82 10-9884-3376 Portfolio: <https://woogi.is-a.dev>

GitHub: <https://github.com/woogi-kang>

## Executive Summary

저는 2025년 Claude Code 공개 이후 Claude/Codex/Gemini CLI 기반 환경에서 누적 약 100억 토큰 규모의 실험과 실무 적용을 반복하며, AI가 개인 생산성 도구를 넘어 조직의 업무 방식 자체를 바꾸려면 어떤 실행 환경이 필요한지 설계해 온 Product Engineer입니다. 제품 엔지니어로서는 Flutter 기반 B2C/B2B 앱 출시와 운영을 6년간 경험했고, 최근에는 회사 운영 업무를 AI Agent와 데이터 파이프라인으로 자동화하는 일에 집중하고 있습니다.

와이어트 전략팀의 AI 자동화 엔지니어 역할과 관련해 가장 강한 경험은 네 가지입니다. 첫째, 사내 업무 자동화 톨과 agent/skill 자산 저장소, 메신저 기반 Agent gateway를 통해 Claude/Codex/Gemini CLI와 agent/skill 자산을 비개발 직군도 함께 사용할 수 있는 업무 접점으로 적용한 경험입니다. 둘째, 병원 상세, 피부 시술, 이벤트·프로모션, 뷰티 제품, 리뷰, 채널·성과 데이터처럼 업무 도메인에 필요한 외부 데이터를 수집 대상으로 정의하고 checkpoint, retry, validation, export까지 닫는 데이터 파이프라인을 구축한 경험입니다. 셋째, LLM 프롬프트를 운영 자산으로 관리하고 A/B 평가와 품질 검증 루프를 만든 경험입니다. 넷째, CheckYourHospital 같은 사이드 프로젝트를 포함해 AI 제품을 데모가 아니라 리포트, Admin, 운영 runbook, 후속 자동화까지 포함한 서비스 흐름으로 만든 경험입니다.

이 문서는 와이어트 공고의 담당 업무인 사내 업무 자동화 도구 및 AI Agent 개발, 데이터 파이프라인/대시보드 구축, LLM Test & Learn 운영, 외부 AI 파트너 협업 및 기술 통합에 맞춰 관련 경력을 재구성한 것입니다.

## Wyatt 적용 시나리오

- 전략 회의체 액션 아이템, 리포트 요청, 후속 실행 상태를 task board와 메신저 Agent로 구조화
- 닥터포헤어/어노브/탱글엔젤의 상품, 리뷰, 프로모션, 채널 데이터를 수집·검증해 전략팀 리포트와 대시보드에 연결
- 브랜드 콘텐츠, 상품 상세, FAQ, 고객 응대 문구의 LLM variant를 만들고 성과·사실성·톤을 Test & Learn으로 비교
- 민감한 데이터 조회, 외부 발송, 고객 노출 산출물은 approval gate와 human review를 거쳐 운영

## Case 1. 사내 AI 업무 자동화 환경

### Context

AI CLI 도구는 개인 생산성을 높이는 데는 빠르게 효과가 나지만, 팀 단위 업무로 확장하려면 문제가 생깁니다. 사람마다 프롬프트 작성 방식, 검증 기준, 결과물 형식, 재사용 방식이 달라지고, 결과적으로 같은 문제를 반복 설명하거나 품질을 다시 확인해야 합니다. 특히 비개발 직군은 CLI나 개발 환경에 직접 접근하기 어렵기 때문에, 메신저처럼 익숙한 업무 접점에서 도메인 특화 AI Agent를 사용할 수 있어야 했습니다.

이 문제를 해결하기 위해 사내 업무 자동화 톨, agent/skill 자산 저장소, 메신저 기반 Agent gateway, 실행 상태 추적 task board를 조합했습니다. 사내 업무 자동화 톨은 회사 업무 요청, 실행, 승인, 산출물, 감사 로그를 task/run 단위로 관리하는 AI 업무 운영 시스템입니다. agent/skill 자산 저장소는 Claude Code, Codex CLI, Gemini CLI가 같은 규칙과 agent/skill/workflow 자산을 공유하도록 만든 작업 자산 저장소입니다. 메신저 기반 Agent gateway는 Slack/Discord 같은 업무 채널에서 모든 회사 구성원이 도메인 특화 AI Agent를 호출할 수 있도록 적용한 접점입니다. task board는 Agent 작업의 요청, 상태, 결과, 이슈를 추적하는 역할로 연결했습니다.

구현은 Next.js 운영 UI, FastAPI API, Claude/Codex/Gemini CLI worker, Redis queue, PostgreSQL, file/output storage를 연결하는 방식으로 설계했습니다. 업무 요청과 실행 상태는 issue/task 및 메신저 관점에서 확인할 수 있게 하고, 민감한 작업은 승인과 human review를 거치도록 경계를 두었습니다.

### What I Built

- Claude Code, Codex CLI, Gemini CLI가 같은 규칙과 skill 자산을 참조할 수 있는 CLAUDE.md, AGENTS.md, GEMINI.md, .agents/skills, .claude/skills 구조 정리
- 도메인 Agent, skill catalog, slash command, team orchestration template, lifecycle hook을 운영/후보 자산으로 분리해 관리하는 workspace 구성
- 자연어 또는 template 기반 작업 분해 흐름을 만들고, plan/design/develop/QA/deploy/review 단계를 Agent workflow로 분리
- 메신저 기반 Agent gateway를 Slack/Discord와 연결해 사내 규칙, 서비스 위키, 운영 정책, 리포트 업무 등 도메인 특화 AI Agent를 메신저에서 호출할 수 있는 접점 설계
- 사내 업무 자동화 톨에서 업무 요청, 실행, 승인, 산출물, 감사 로그, worker 상태를 운영 테이블과 dashboard에서 확인할 수 있도록 설계
- 민감한 외부 제출/발송은 직접 자동 실행하지 않고 approval gate와 human review 이후 수동 제출로 경계를 둬
- 사이드 프로젝트인 CheckYourHospital과 Memoriz에는 task board 기반 개발 원칙을 문서화해 목표, 범위, 영향 표면, 검증 기준, 배포/롤백 노트를 이슈에 남기는 작업 흐름 정착

## Operating Details

Agent workflow를 단순히 "AI에게 일을 시키는 방식"으로 설계하지 않았습니다. 실제 운영에서는 요청자, 실행 환경, 작업 범위, 검증 기준, 실패 기록, 승인 여부가 남아야 합니다. 그래서 사내 업무 자동화 툴의 CLI adapter는 민감한 작업을 기본 dry-run으로 실행하고, 실제 실행에는 approval과 작업 범위 확인을 요구하도록 구성했습니다.

또한 API가 provider CLI를 직접 실행하지 않고 worker가 queue를 소비하도록 분리했습니다. worker는 실행 상태 보고, 민감 로그 redaction, timeout, 출력 제한, 변경 파일 보고, 작업 범위 위반 감지를 담당합니다. 이 구조는 AI를 도입할 때 자주 발생하는 문제인 "누가 어떤 지시를 했고, 어떤 파일이나 데이터에 영향을 주었고, 실패했을 때 어디서 멈췄는지 알 수 없음"을 줄이기 위한 장치입니다.

## Impact

- AI 도구 사용을 개인별 prompt 작성 수준에서 비개발 직군도 함께 사용할 수 있는 업무 운영 자산으로 전환
- 개발, 기획, 디자인, 콘텐츠, 마케팅, 리뷰, 법무, 재무 등 여러 도메인의 반복 업무를 agent/skill/workflow 단위로 구조화
- 업무 요청, 승인, 실행, 검증, 기록이 이어지는 AI 업무 운영 환경을 설계
- 비개발 직군이 Slack/Discord, Admin, issue workflow 같은 익숙한 접점에서 도메인 특화 AI Agent를 사용할 수 있는 확장 기반 확보

## Wyatt Application

와이어트 전략팀에서는 사업계획, P&L 분석, 전략 회의체, 전사 핵심 과제, 내부 프로세스 개선처럼 반복성과 문맥 의존성이 큰 업무가 많습니다. 이 경험을 활용하면 다음과 같은 업무 자동화가 가능합니다.

- 전략 회의체 액션 아이템을 issue/task로 자동 구조화하고 후속 실행 상태를 추적
- 사내 문서, 운영 정책, 회의록, 리포트를 Agent가 검색하고 요약하는 업무 점점 구축
- 자사물, 올리브영, 해외 채널, 광고/콘텐츠 성과 리포트를 정기 task로 관리
- 민감한 데이터나 외부 발송 작업은 approval gate와 human review를 거치도록 설계

## Case 2. Domain Data Collection and Verification Pipeline

### Context

업무 자동화의 성패는 데이터 품질에 크게 의존합니다. LLM이 아무리 좋아도 입력 데이터가 불완전하거나 출력이 불분명하면 현업 의사결정에는 쓰기 어렵습니다. 저는 네이버 플레이스 병원 데이터뿐 아니라 병원 상세, 피부 시술, 이벤트·프로모션, 뷰티 제품, 리뷰, 채널·성과 데이터처럼 회사 업무 도메인에 필요한 데이터를 대상으로 수집, 검증, 보정, 병합, export까지 담은 파이프라인을 여러 차례 구축했습니다.

대표 사례는 naver-place-clinic-crawler, CheckYourHospital의 event pipeline, 그리고 회사 업무 도메인에 맞춰 수집 대상을 바꾸며 운영한 데이터 수집/검증 workflow입니다.

### Naver Place Clinic Crawler

네이버 플레이스에서 피부과 병원 상세 데이터를 수집하는 HTTP-only crawler입니다. Headless browser 없이 httpx 요청과 \_\_APOLLO\_STATE\_\_ SSR JSON 파싱을 활용해 동작하도록 설계했습니다.

### Key Design

- 4,255건 피부과 입력 데이터를 대상으로 지역별/구역별 크롤링 실행
- 서울 1,723건을 6개 구역으로 분할하고 split DB를 병렬 실행한 뒤 병합하는 구조 설계
- 동명 병원 문제를 해결하기 위해 이름 검색, 이름+지역 검색, 후보별 좌표 비교, 도로명 검색 폴백으로 이어지는 3-tier place id matching 구현
- 병원별 pipeline을 search place id, home, information, photos, validation, export skill로 분리
- SQLite checkpoint DB, WAL mode, per-record commit, 실패건 재시도, 점진적 backoff, CSV/JSON export 구조 구현
- 수집 데이터는 진료시간, 휴무일, 편의시설, 지하철, 찾아오는 길, 예약/톡톡 URL, 홈페이지, SNS, 사진 수 등 운영에 필요한 필드로 정리

### Operating Point

이 파이프라인의 핵심은 "한 번 돌려서 데이터를 긁는 것"이 아니라, 중단 후 재개, 실패 재시도, 병합, validation status, export를 고려한 운영 구조입니다. 데이터 수집 업무는 항상 일부 실패와 변경된 DOM, 중복, 동명 객체 문제가 발생하기 때문에, 처음부터 보정 가능한 단위로 나누는 것이 중요했습니다.

### CheckYourHospital Event Pipeline

CheckYourHospital은 병원 홈페이지의 AI SEO/AEO 준비도와 이벤트·프로모션 데이터를 진단하기 위해 만든 개인 사이트 프로젝트입니다. 그중 event pipeline은 병원 이벤트·프로모션 정보를 수집하는 worker pipeline이며, Playwright, Gemini/Codex CLI, Cloud SQL, R2를 활용하고 월간 로컬 운영 runbook을 별도로 작성했습니다.

### Pipeline Flow

- locate: 대상 병원의 이벤트·프로모션 후보 페이지 탐색
- collect: 텍스트/이미지 기반 이벤트 후보 수집
- crosscheck: 수집 결과를 원문, 이미지, 가격, 시술명 기준으로 검증
- R2 upload: 실행 산출물, screenshot, raw response를 run 단위로 저장

- DB insert: Cloud SQL에 구조화 결과 저장

### Operating Point

운영자가 같은 파이프라인을 반복 실행할 수 있도록 Cloud SQL 접속, R2 설정, Gemini/Codex CLI 인증, Playwright 설치, 단일 병원 실행, 이어받기, 검증 절차를 runbook으로 남겼습니다. 자동화는 코드뿐 아니라 운영자가 실패 없이 반복 실행할 수 있는 문서와 체크리스트까지 포함해야 한다고 보고 접근했습니다.

### Impact

- 병원 상세, 피부 시술, 이벤트·프로모션, 뷰티 제품, 리뷰, 채널·성과 데이터처럼 업무 도메인에 필요한 외부 데이터를 수집 가능한 skill 단위로 분리하고, checkpoint, retry, validation, export까지 닫는 구조 구축
- HTTP-only 수집, browser automation, LLM extraction, structured validation을 상황에 맞게 조합
- 데이터가 실패하거나 품질 이슈가 있을 때 원인 추적, 재시도, 보정, 병합이 가능한 운영 단위 확보
- 운영 runbook을 통해 특정 개발자에게만 의존하지 않는 반복 실행 구조 마련

### Wyatt Application

와이어트에서는 브랜드별 상품 정보, 고객 리뷰, 올리브영/자사몰/글로벌 채널 판매 지표, 경쟁사 상품/가격/프로모션, SNS/커뮤니티 반응 등을 정기적으로 수집하고 분석할 수 있습니다.

제가 구축한 방식은 다음 업무에 적용할 수 있습니다.

- 닥터포헤어/어노브/탱글엔젤 관련 리뷰, 상품 상세, 프로모션, 채널 데이터를 주기적으로 수집
- 고객 VOC와 상품/성분/가격 정보를 연결해 브랜드별 개선 포인트 도출
- 데이터 수집 실패와 중복을 checkpoint/retry/validation으로 관리
- 전략팀 리포트나 대시보드에서 바로 쓸 수 있도록 CSV/DB/API 형태로 정리

## Case 3. LLM Test & Learn and Content Automation

### Context

LLM 기반 자동화는 처음 한두 번 잘 동작하는 것보다, 운영 중 품질을 측정하고 개선하는 루프가 중요합니다. 특히 브랜드 콘텐츠, 고객 응대, 리포트 생성처럼 외부로 노출되거나 의사결정에 쓰이는 영역에서는 사실성, 톤, drift, 성과 지표를 함께 관리해야 합니다.

social-outreach 프로젝트에서는 멀티 플랫폼 콘텐츠/아웃리치 파이프라인을 설계하며, 검색, 수집, 분석, 콘텐츠 생성, 성과 측정, 프롬프트 개선, lesson 저장으로 이어지는 Test & Learn 구조를 만들었습니다.

### What I Built

- X, Instagram, Threads, Xiaohongshu 등 플랫폼별 worker adapter를 둘 수 있도록 contracts, shared, core, app 계층을 분리
- 키워드 기반 prefilter와 LLM classification을 결합해 후보 콘텐츠의 의도와 품질을 분류
- 전문가 기반 콘텐츠 생성 후 persona adaptation을 적용하는 2단계 콘텐츠 생성 구조 설계
- prompt versioning, prompt selector, active/testing 상태, A/B evaluator, quality scorer, bandit, drift monitor, competitor monitor, lesson store 구조 구현
- 성과 데이터 수집 후 analyze -> generate -> vote -> debate -> adopt -> lesson으로 이어지는 self-evolution loop 설계
- 콘텐츠 품질과 브랜드 리스크를 줄이기 위해 factual guard, persona guard, drift guard, human-in-the-loop review, rate limit, emergency halt를 적용

### Operating Details

이 프로젝트를 지원 문서에서 표현할 때 핵심은 "자동 발송"이 아닙니다. 핵심은 브랜드 콘텐츠 운영에서 LLM을 안전하게 실험하고 개선하는 방법입니다. 그래서 운영 구조는 무작위 자동화가 아니라 다음 원칙으로 설계했습니다.

- 브랜드 톤과 사실성을 분리해 검증한다.
- 프롬프트는 version, status, metrics를 가진 운영 자산으로 관리한다.
- 개선안은 단일 LLM 판단에 의존하지 않고 voting/debate, A/B evaluation, quality score를 거친다.
- 플랫폼 정책과 브랜드 리스크를 고려해 승인 흐름과 레이트 리밋을 둔다.
- 성과가 좋았던 lesson을 다음 콘텐츠와 프롬프트 개선에 반영한다.

### Impact

- 콘텐츠 생성 자동화를 단발성 생산량 증가가 아니라 성과 기반 Test & Learn 루프로 전환
- prompt drift와 사실성 오류를 검증 가능한 품질 이슈로 다룰 수 있는 구조 마련
- 멀티 플랫폼 운영에서 공통 계약, 공유 로직, 플랫폼별 adapter를 분리해 확장 가능성 확보
- 브랜드/콘텐츠/성과 데이터가 다시 프롬프트 개선으로 이어지는 학습 루프 설계

## Wyatt Application

와이어트는 닥터포헤어, 어노브, 탱글엔젤 등 브랜드별 고객군과 채널이 다르고, 국내외 확장에 따라 콘텐츠 톤과 고객 반응도 달라집니다. 이 경험은 다음 업무에 연결할 수 있습니다.

- 고객 리뷰와 SNS 반응을 분석해 브랜드별 VOC theme와 상품 개선 포인트 추출
- 콘텐츠/광고/상품 상세 문구의 LLM variant를 만들고 성과 기반으로 비교
- 해외 시장별 일본어/영어/중국어 콘텐츠의 브랜드 톤, 사실성, 금칙 표현을 검증
- 마케팅/영업/상품팀이 반복적으로 만드는 리포트, 카피, FAQ, 비교표를 semi-automation으로 전환

## Case 4. AI Product Quality and Report Automation

### Context

AI 기능은 출시 후 품질이 흔들리기 쉽습니다. 모델, 프롬프트, 데이터, 검색 인덱스, threshold 중 하나만 바뀌어도 결과 품질이 달라질 수 있기 때문입니다. 저는 AI 기능을 구현할 때 평가 데이터, 리포트, 테스트, 운영 스냅샷이 함께 갱신되도록 품질 검증 체계를 설계했습니다.

대표 사례는 개인 사이드 프로젝트 CheckYourHospital과 Memoriz입니다.

### CheckYourHospital

CheckYourHospital은 병원 홈페이지를 7개 카테고리 48개 항목으로 진단하는 AI SEO/AEO 사이드 프로젝트입니다. 외국인환자 유치 사이트 특화 진단, 의료법 컴플라이언스, 다국어 hreflang, AI 검색 노출, Naver Place까지 함께 분석합니다.

#### Key Functions

- Technical SEO, Content, International, Authority, AI/AEO, Machine Readability, Medical Compliance 7개 카테고리 스코어카드
- 의료법 컴플라이언스 critical issue가 있으면 전체 점수를 cap하는 gating rule
- ChatGPT, Gemini, Claude, Perplexity 질의 기반 AI 추천 시뮬레이션
- 언어x페이지 매트릭스, Princeton GEO 기반 콘텐츠 점수, Naver Place 연동, 경쟁사 radar chart
- Jinja2 HTML -> Playwright PDF -> output storage로 이어지는 리포트 생성
- 리드 캡처 후 후속 follow-up 자동화 흐름

### Memoriz

커플 기록 앱으로 Flutter app, Next.js landing/admin, FastAPI API, Playwright E2E, ops dashboard, AI album/search 기능을 포함합니다.

AI/search 기능에서는 ai\_change\_guard.py 를 통해 구현 변경과 평가/리포트 산출물 갱신을 연결했습니다. 예를 들어 album AI나 album search 구현 파일이 변경되면 관련 eval report script, eval set template, sample report, targeted test 변경 여부를 검사합니다.

이 접근은 AI 기능의 품질을 "개발자가 감으로 확인하는 것"에서 벗어나, 기능 변경 시 평가 근거와 리포트가 함께 남도록 만드는 품질 장치입니다.

### Impact

- AI 제품을 단순 상담/검색 기능이 아니라 진단, 평가, 리포트, 리드 캡처, 후속 자동화까지 연결된 운영 흐름으로 설계
- AI 기능 변경 시 평가/리포트 산출물과 targeted test를 요구해 품질 저하를 프로세스 안에서 감지
- Admin, dashboard, runbook, smoke test, E2E를 포함한 운영 표면을 함께 관리
- 고객에게 노출되는 AI 결과물의 신뢰도와 내부 운영자의 검증 가능성을 함께 고려

## Wyatt Application

와이어트에서 LLM을 도입할 때도 프롬프트가 좋아 보이지만 볼 것이 아니라, 업무별 평가 기준과 운영 리포트가 필요합니다.

적용 가능한 예시는 다음과 같습니다.

- 상품 리뷰 요약 Agent의 정확도, 누락률, sentiment 분류 품질을 eval set으로 관리
- 브랜드별 FAQ/상담 답변 생성 Agent의 금칙 표현, 사실성, 톤을 자동 검증
- 월간/주간 리포트 생성 Agent의 숫자 출처, SQL 결과, chart input을 structured output으로 검증
- 프롬프트 변경 시 이전 버전과 비교하고, 성과가 낮으면 rollback할 수 있는 PromptOps 체계 운영

## Case 5. Product Operations and Device Automation

### Context

엑스퍼트아이엔씨에서는 AI 스마트 안경 제품군인 OWL, C-Biz, C-Sound의 모바일, 데스크톱, 스마트 글래스 앱을 함께 관리했습니다. 이 경험은 AI Agent 직무의 중심 사례는 아니지만, 자동화 결과물이 실제 현장에서 쓰일 때 필요한 배포, 권한, 장애 대응, 운영자 경험을 다룬 경력입니다.

### What I Built

- Flutter 기반 Mobile/Desktop/Smart Glasses 제품군 구조 설계 및 BLE 디바이스 연동 구현
- Android Device Owner 기반 Kiosk Mode, OTA update, ADB Factory 웹앱으로 현장 기기 설치/삭제/업데이트 자동화

- Claude Code를 사내에 도입하고 Figma-to-Flutter, 자동화 Agent, 서버에이전트 기반 개발 분업을 실험

## Impact

- 3개 서비스와 각 플랫폼을 소수 인원으로 유지보수할 수 있는 구조로 정리
- 현장 설치, 업데이트, 장애 대응 같은 반복 운영 업무를 자동화
- 분석이나 개발에서 끝나지 않고 배포, 현장 운영, 모니터링까지 닫는 실행 경험 축적

## Wyatt Application

와이어트 전략팀에서도 분석에서 끝나지 않고 실행까지 이어지는 역량이 중요합니다. 이 경험은 자동화 결과물이 실제 현장에서 쓰일 때 발생하는 설치, 배포, 권한, 예외, 장애, 운영자 경험까지 고려하는 태도로 연결됩니다.

## Core Skill Set

---

### AI Agent and Automation

- 사내 업무 자동화 툴, agent/skill 자산 저장소, 메신저 기반 Agent gateway, task board
- Claude Code, Codex CLI, Gemini CLI, Slack/Discord Agent 운영 접점
- Agent/Skill/Workflow/Slash Command/Hook/Template 설계
- MCP, Tool Calling, Claude Agent SDK, LangChain, LangGraph, ReAct Agent
- Messenger-based domain agents, internal chatbot, Admin automation

### LLM Test & Learn

- Prompt versioning, active/testing/archive 상태 관리
- A/B evaluation, LLM-as-Judge, voting/debate, quality score
- factual guard, drift monitor, persona guard, human review
- 평가 데이터, 리포트 산출물, targeted test, rollback flow

### Data Pipeline

- Python, SQL, PostgreSQL, SQLite, Redis
- Playwright, httpx, BeautifulSoup, APOLLO\_STATE parsing
- checkpoint, retry, split DB, merge, validation, CSV/JSON export
- Cloud SQL, GCS/R2, MinIO, Docker, Docker Compose

### Product Engineering

- Next.js, React, TypeScript, FastAPI, Pydantic
- Flutter, Dart, Riverpod, Melos, Freezed
- Sentry, Crashlytics, Analytics, Playwright E2E, Vitest
- GCP Cloud Run, Secret Manager, Cloud Storage, Cloud SQL

## Closing

---

제가 와이어트에서 기여할 수 있는 부분은 AI 자동화 도구를 도입하는 데서 끝나지 않습니다. 전략팀의 업무 흐름을 이해하고, 반복 업무와 의사결정 병목을 데이터와 Agent workflow로 구조화하며, LLM 실험을 운영 가능한 Test & Learn 체계로 만들 수 있습니다.

특히 와이어트가 고객과 제품 품질 기준을 높게 두는 브랜드 회사라는 점에서, AI 자동화 역시 빠른 생성보다 정확한 데이터, 검증 가능한 결과, 현업이 반복 사용할 수 있는 프로세스로 설계되어야 한다고 생각합니다. 저는 지금까지 만든 Agent workflow, 데이터 파이프라인, 품질 검증 체계 경험을 바탕으로, 와이어트의 전사 실행 속도와 업무 생산성을 높이는 AI 자동화 엔지니어로 기여하고 싶습니다.