

SUMMARY

RAG 근거 선택: 근거 충족률 66.25% → 91.25%, 답이 없는 질문에 근거를 붙인 경우 20 → 0건

에이전트 실행 환경: 사용자 정의 Skill과 승인 기반 실행, 편집 품질 82.5% → 91.2%(114건 평가)

데이터 파이프라인: PDF 변환 통과율 45.17% → 89.89%, 약품 변환 비용 예상 \$200 대비 \$11.18

PROJECTS & EXPERIENCE

Fruition

2026.04 — 현재

LLM Wiki 기반 AI 워크스페이스 · AI SW 마에스트로 17기 · AWS EKS 배포 완료, QA 진행 중
팀 프로젝트 (3명 + 디자이너) / AI 파트 리더

시작 계기: 팀원이 제기한 문서 관리의 불편함을 디자인 싱킹으로 구체화. 저장만 해 둔 문서를 필요할 때 찾지 못하는 문제를 풀고자 했습니다.
팀 구성: 본인(AI 전체), 백엔드 1명(Spring 백엔드 전체), 프론트엔드·DevOps 1명(프론트엔드, MSA 기반 AWS 배포), 외주 디자이너 · 멘토 3명

GitHub · AI 분리 저장소

RAG 과정의 검색 정확도 개선

저장된 문서에서 AI 답변에 필요한 근거를 찾고, 답이 없는 질문에는 근거를 붙이지 않게 하는 기능입니다.

- 답변 근거 선택 개선

문제: 문서 근거로 답하는 서비스에서 기존 검색이 근거를 놓치거나 답 없는 질문에도 근거를 붙여 답변 신뢰도가 흔들림.

판단: GPT 선택은 비용(약 2.5배)·시간 부담, 가중치·RRF 조정은 여러 값을 실험해도 점수가 비슷해, 요청 분류에서 먼저 검증한 Jev(정답 77 → 81/98건) 채택.

해결: 벤치마크상 한국어 의미 검색이 가장 좋은 BGE-M3와 정확한 용어를 잡는 BM25로 후보를 모으고 Jev가 최종 근거를 고르는 2단계 구조. 동일 후보·블라인드 교차 채점으로 비교한 뒤 후보를 300개로 줄이고 병렬 4·6·8·10을 비교해 정확도 차이 없이 가장 효율적인 8병렬로 조정.

성과: 답이 있는 80문항의 근거 충족률 66.25% → 91.25%. 선택 시간 중앙값 9.090초 → 1.137초(기존 0.431초보다 느림). 근거 신뢰도를 높여 사용자가 답변을 믿고 계속 쓸 이유를 만들.

사용자 정의 Agent Skill과 승인 기반 실행

반복 작업을 Skill로 저장·재사용하고, AI가 만든 변경은 사용자가 승인한 뒤에만 반영되도록 설계했습니다.

- 사용자 정의 작업 지침(Skill)

문제: 사용자마다 반복 작업이 달라 모두 기능으로 만들 수 없고, AI 편집안이 요청을 빠뜨린 채 전달돼 같은 결과를 원하는 작업을 운에 맡기게 됨.

판단: 지침은 AI, 권한·실행 범위 검사는 코드, 게시와 실제 반영은 사용자가 말도록 책임 분리. 코드의 형식 검사만으로는 요청 누락을 못 잡아 LLM 의미 평가를 추가.

해결: 자연어 지침 작성·검토·게시와 권한·승인 우회 검사, 미리보기 후 별도 승인. 편집 결과는 의미 평가 후 최대 1회 재작성. 생성·평가·재작성은 단계별로 추적하도록 LangGraph 노드로 나누고, 게시한 Skill은 PostgreSQL에 불변 버전으로 저장.

성과: 편집 품질 94 → 104/114건(82.5% → 91.2%). 승인 전에는 문서가 바뀌지 않음을 확인. 결과를 안정시켜 선결제 크레딧이 재작업에 낭비되지 않게 함.

PDF 문서 변환

사용자가 올린 논문 PDF를 검색과 답변에 쓸 수 있도록, 표·수식까지 보존하는 Markdown으로 변환했습니다.

- PDF → Markdown 변환

문제: 논문 PDF의 표·수식이 깨지면 Wiki와 답변의 근거가 되는 수치를 쓸 수 없음.

판단: 깨진 곳만 고치는 방식은 탐지가 오류를 놓쳤고, 전체 AI 변환은 표·수식 정확도와 비용이 부족해 내용 유형별 처리로 전환. 본문은 추출이 빠른 AnyDoc, 복원은 GPU 운영비가 드는 sLLM 대신 상용 LLM API.

해결: 본문은 AnyDoc으로 추출해 깨진 부분만 AI로 보완, 표·수식은 원본 영역을 따로 판독, 그림은 원본 보존.

성과: 같은 30쪽·445단위 블라인드 평가에서 통과율 45.17% → 89.89%. 핵심 수치가 Wiki에 온전히 편집되게 해 품질 저하로 인한 이탈 요인을 줄임.

MSA: 기능별로만 나누면 DB 부하는 그대로이고 서비스 간 호출만 늘어난다고 보고 데이터 소유권 기준 분리를 제안, 팀원·멘토와 경계·통신 방식을 공동 설계.

수치는 모두 내부 평가 기준입니다.

문제 해결 과정과 실험 결과는 상세 포트폴리오에서 볼 수 있습니다.

RAG 근거 선택 에이전트 실행 환경 PDF 변환

Pilltip

2025.03 — 2025.12

개인 맞춤 AI 안심 복약 솔루션 · 부산대학교 졸업과제 · APK 내부 배포 사용자 6명
팀 프로젝트 (3명 + 디자이너) / 핵심 백엔드 · 데이터 정제

시작 계기: 약품 정보가 흩어져 있어, 지금 건강 상태에서 특정 약을 먹어도 되는지 확인하기 어렵다는 문제에서 출발했습니다.
팀 구성: 본인(백엔드·AI), Android 1명(Kotlin 앱 전반), 백엔드·보안 1명(사용자 DB, 로그인, 문진표, AES-GCM), 디자이너. 아래 목록은 제가 구현한 기능입니다.

[GitHub 저장소](#)

데이터 정제 파이프라인 구축

약 4만 4천 건의 의약품 원문을 쉬운 설명으로 바꾸는 작업입니다. 같은 문장은 한 번만 변환해 약품별로 재사용했습니다.

- 문제:** 약 4만 4천 건을 그대로 쉬운 설명으로 바꾸면 API 비용이 약 \$200으로, 예산 안에서 모든 약의 설명을 제공할 수 없었음.
판단: 약 수를 줄이는 대신 같은 문장은 한 번만 변환. 실시간 응답이 필요 없는 작업이라 Batch API로 대기 시간을 받아들이고 비용을 낮춤.
해결: 문장 블록으로 나눠 완전 일치 문장을 통합하고 GPT Batch API로 변환한 뒤 약품별로 재연결. 2026년 코어 병렬 처리와 확장이 가능한 PySpark로 재현해 단계 별 효과 측정.
성과: 전체 원문 변환 예상 비용 약 \$200 대비 실제 API 지출 \$11.18. 재현 실험에서 완전 일치 중복 제거로 텍스트 744MB → 44.1MB(-94.1%).

비용은 예상치와 실제출의 비교이며 전처리 인건비는 제외했습니다.

개인 건강 정보 기반 답변 제공 AI 복약 챗봇 개발

사용자가 증상을 자연어로 물으면 관련 약품과 복약 정보를 찾는 기능입니다. 건강 프로파일은 내부에서 판단해 개인화에 활용하고, 외부 AI에 보내는 정보와 분리했습니다.

- 문제:** 약을 먹어도 되는지 확인하려는 사용자의 증상 표현과 약 효능 용어가 달라 검색이 어렵고, 개인화에 필요한 건강 정보는 외부 AI로 보낼 수 없었음.
판단: 동의어 사전은 증상이 많고 서로 겹쳐 의미 검색으로, 프로필 필터링은 걸려야 할 경우가 너무 많아 내부 판단으로 전환. 벡터 DB는 Milvus보다 가벼운 Weaviate, 오케스트레이션은 백엔드가 Spring이라 별도 Python 컨테이너 부하를 피해 Spring AI 사용.
해결: Weaviate 의미 검색과 AI 오케스트레이션으로 약품 탐색·복약 위험·섭취량별 처리 경로 구성. 임신 여부·복용약·기저질환은 내부 코드로 판단.
성과: 자연어 증상으로 약품 정보를 탐색하고, 해당 건강 프로필을 외부 LLM 입력에서 제외한 개인화 흐름 구현.

문제 해결 과정과 실험 결과는 상세 포트폴리오에서 볼 수 있습니다.

[데이터 파이프라인 사례](#) [개인화 설계 사례](#)

기술

AI & Search	LangChain · LangGraph · Spring AI · RAG LangSmith · LLM Evaluation · LLM Guardrail
Backend	Java · Spring Boot · Python
Data & Infra	MySQL · PostgreSQL · Redis · Elasticsearch · Weaviate Docker · Docker Compose · GitHub Actions · Kafka

활동 · 기여

2026.07 — 2026.09	200 SIGNAL · TQQQ 200일선 신호 앱 (개인 프로젝트) 커뮤니티의 TQQQ 200일선 매매법을 전략 명세로 구조화하고, 실시간 시세로 신호를 알리는 Android 앱으로 만들어 가족 2명이 사용했습니다. 원문이 주장한 성과는 미검증으로 표시하고, 45개 사이클을 재현해 승률이 19%가 아닌 40.00%임을 확인했습니다. 부분익절을 거래로 세면 승률이 72.73%로 오르는 것을 보여, 원문의 두 승률이 서로 다른 단위를 센다는 점까지 밝혔습니다. 전략 변형 5종도 같은 가정으로 백테스트해 CAGR-MDD를 구간별로 비교했습니다. 사용자 요청으로 QQQ 기반 두 번째 전략과 대기 자금의 SGOV·SPYM 운용을 추가했습니다. 앱은 배포하지 않았고, 계좌 정보는 기기 안에만 저장합니다. GitHub 저장소
2026.06 — 2026.06	Rhwp 오픈소스 기여 Claude Code에 작업 계획과 기여 규칙을 주고 이슈 분석부터 PR까지 자동화해, HWP/HWPX 저장 오류 수정 등 PR 7건 병합·Contributor로 등재됐습니다. GitHub 저장소 PR #1213 · 병합 PR #1351 · 제출
2025.03 — 2026.01	부산대학교 APPTIVE Backend 멘티 및 멘토. 멘티 12명 대상 6회 멘토링과 코드 리뷰. HTTP·Servlet·REST API·DB 기초를 보강하는 커리큘럼 개편에 참여했습니다. APPTIVE 백엔드 멘토 공로상 · 2026.01.09 
2025.11 / 2025.07	SK AI SUMMIT · K-ICT WEEK in Busan 부산대학교 대표 전시팀으로 Pilltip 부스를 운영하며 방문자 100명 이상에게 시연하고 기술 질의응답을 진행했습니다. 받은 피드백 10건 중 대부분이 건강기능식품 확장 요청이었습니다. 
2022.02 — 2025.07	백준 945일 연속 문제 해결 solved.ac 기준 최장 945일 연속. 누적 1,659문제, Platinum IV. BaekJoon 풀이 저장소

학력 · 자격

2022.02 — 2026.02 (편입)	부산대학교 평균 학점 4.12 / 4.5
2019.03 — 2022.02 (중퇴)	대구대학교 평균 학점 4.2 / 4.5
2025.09	정보처리기사 한국산업인력공단

수상

2025.10		캡스톤디자인 금상 소프트웨어·인공지능 분과 / 부산대학교 정보의생명공학대학
2025.09		부산 DATA WEEK 최우수상 데이터 활용 우수사례 공모전 / 부산테크노파크
2025.09		AI LAUNCH 커리어스쿨 창업톤 장려상 KRYPTON X x Root Impact x Google.org
2025.08		SW중심대학 디지털 경진대회 후원기업상 SW중심대학협의회