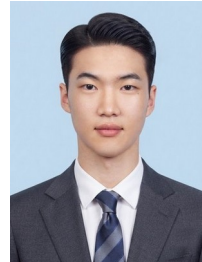


## 지원 직무: AI 서비스 · 백엔드 개발자

FastAPI · RAG 멀티 에이전트 AI 서비스 백엔드 개발자 류준입니다. LLM 에이전트와 RAG 백엔드를 설계하고, 직접 배포하고, 숫자로 검증합니다. (Build → Deploy → Evaluate)

|       |                                                                                             |                                                                                     |
|-------|---------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| 이름    | 류준                                                                                          |  |
| 생년월일  | 비공개                                                                                         |                                                                                     |
| 주소    | 비공개                                                                                         |                                                                                     |
| 연락처   | (E-mail) fbwns1234@gmail.com                                                                |                                                                                     |
| 포트폴리오 | <a href="https://whoareryu.cloud/?from=jobkorea">https://whoareryu.cloud/?from=jobkorea</a> |                                                                                     |

### 경력 및 강점 요약

개인 프로젝트 1개와 팀 프로젝트 7개를 설계·구현·배포·평가까지 수행하며, LLM 서비스를 '동작하는 데모'가 아니라 '믿을 수 있는 서비스'로 만드는 방법을 익혔습니다. 모든 프로젝트는 개인 도메인(whoareryu.cloud)에서 24 시간 공개 운영 중이며, "측정하지 않은 수치는 적지 않는다"를 원칙으로 일합니다.

**[RAG · 에이전트 이해]** 권한별로 검색 결과가 달라지는 사내보안 RAG(Secu-Agent, 개인)와 다산콜센터 상담원 어시스트 RAG(CallGuard, 팀 · ai/ 담당)를 만들며 청킹·임베딩·하이브리드 검색(BM25 + dense, RRF 융합)을 직접 구현했습니다. LLM 에이전트 게임 REMAKE DAY(팀장 겸 AI Agent Engineer)에서 역할별 모델 분리와 API 계약 설계를 맡아, 에이전트 서비스는 성공 경로보다 실패 경로 설계가 품질을 결정한다는 것을 배웠습니다.

**[평가 · 하네스 능력]** LLM을 채점자로 쓰지 않는 골든셋 기반 평가 하네스(규칙 기반 자동채점 · 3회 최저치 고정 · CI 기준선 게이트)를 구축할 수 있습니다. CallGuard에서 골든셋 295건으로 BM25 기준선(Recall@5 0.853 · MRR 0.686)을 재고 하이브리드 검색으로 0.971 · 0.875까지 개선을 증명했고, Secu-Agent에서는 흔들리던 지표의 원인(의존성 잠금 누락)을 고정한 뒤에야 수치를 공개했습니다. LLM이 실패해도 서비스가 무너지지 않는 Agent Harness(스키마 검증 → 사실 검사 → 재생성 → 폴백)을 코드로 구현합니다.

**[백엔드 · 배포 능력]** FastAPI + PostgreSQL(pgvector) 백엔드와 Next.js 프론트를 Vercel + Supabase로 배포해 비용 0원으로 24시간 운영하고, 근거 문턱(0.67, 실측 결정)·호출 상한을 코드로 강제해 환각과 비용을 함께 제어합니다. 생성 모델 교체로 응답 11.5초 → 3초 단축, Vercel 방화벽·CSP·HSTS 보안 헤더 적용 경험이 있습니다.

**[협업 · 문제 정의]** 해커톤 4회(JB 금융그룹 Fin:AI Challenge, 원티드 AI Championship, 2026 AI Blockchain Challenge in Daegu, K-디지털 트레이닝 해커톤)에서 팀장 2회를 맡으며 API 계약을 먼저 고정해 병렬 개발하는 방식, '데이터로 답할 수 있는 질문'만 기능으로 남기는 범위 결정 방식을 익혔습니다. 이전 사회 경험에서 짧은 대화로 상대의 요점을 잡아내는 습관을 길렀고, 이를 팀 회의와 요구사항 정리에 그대로 씁니다.

### Skill Set

|               |                                                                                                                                                                                                       |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 백엔드           | Python / FastAPI — API 서버 설계, API 계약 정의(REMAKE DAY), 레이트 리밋·비용 상한 구현(허브 채팅), PostgreSQL / pgvector / Supabase, 공공데이터 CSV 정규화 적재(CallGuard · 발자국)                                                      |
| RAG · 검색      | 청킹·임베딩 파이프라인, BM25 + dense 하이브리드 검색(CallGuard: nori + KoE5), 벡터+키워드 RRF 융합(Secu-Agent), 권한 기반 사전 필터링 RAG(Secu-Agent), 근거 문턱 기반 생성 억제(허브 채팅). 임베딩: multilingual-e5-small, KoE5, gemini-embedding-001   |
| LLM · 멀티 에이전트 | LangChain / LangGraph(Secu-Agent), 역할별 LLM 에이전트 설계 및 역할 프롬프트(REMAKE DAY: Core claude-sonnet-5 · NPC claude-haiku-4-5), LLM 폴백 체인(JB Silver Connect: Gemini → Groq → Mock), Gemini · Claude · Groq API |
| 평가            | 골든셋 기반 Recall@K · MRR 측정(CallGuard 295건 · Secu-Agent 30건), 규칙 기반 자동채점, 3회 최저치 고정, "측정 불가" 구분, CI 기준선 게이트(CallGuard), 경계 강제·권한 누출 테스트·변이 검사(Secu-Agent)                                                |

|            |                                                                                                                                                                                                         |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 설계         | 헥사고날·클린 아키텍처, SOLID, GoF 디자인 패턴 — 의존성은 안쪽으로, 경계는 테스트가 지킨다. 계산·판정은 결정론 코드, LLM은 설명만(localhost:daegu · 발자국)                                                                                               |
| 배포 · 프론트   | Vercel(방화벽·보안 헤더), Supabase, Cloudflare Tunnel, AWS EC2 · Docker<br>Compose(CallGuard, 팀 공동 구성), Linux(WSL2 Ubuntu), Git/GitHub, Next.js / TypeScript / React(Secu-Agent · 허브 · localhost:daegu 프론트 배포) |
| AI Tooling | Claude Code 기반 하네스·루프 엔지니어링 — 설계→계획→태스크별 구현·서브에이전트 리뷰→수정의 개발 루프, TDD·systematic-debugging-verification-before-completion 스킬, 세션 로그·배포 파일 편집 전 런북 강제 혹은 Ollama 로컬 모델 운영 및 QLoRA 파인튜닝 가능(가이드 문서화 경험)      |

학력사항

|                 |                      |                         |
|-----------------|----------------------|-------------------------|
| '18.03 ~ '20.02 | 한국호텔관광전문학교<br>호텔경영학과 | 2018.03 입학 · 2020.02 졸업 |
|-----------------|----------------------|-------------------------|

경력사항 (총 약 5 년)

|                              |                                                               |                                                                                                                                                                                                                                                                                                              |
|------------------------------|---------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (약 3 년)                      | <b>F&amp;B 매장</b><br>지점장 (매니저 → 실장 → 지점장)<br>퇴사사유: 경영악화       | 매장 운영 총괄 — 매출·원가·인력·고객 응대 관리 <ul style="list-style-type: none"><li>일 매출 마감·원가율 관리·발주 운영, 월 손익 기준으로 메뉴 구성과 프로모션 결정</li><li>직원 채용·교육·스케줄 운영, 피크타임 동선·역할 재배치로 회전을 개선</li><li>리뷰·컴플레인 1차 대응 및 재방문 고객 관리 — '불만을 응대'하기보다 '불만이 생기는 구조'를 고치는 방식으로 운영</li><li>매니저에서 지점장까지 3년 내 승진 — 운영 전반을 숫자로 보고하고 결정하는 역할</li></ul> |
| '21.04 ~ '22.04<br>(1년)      | <b>엠블던 호텔 (現 엠에스산업)</b><br>프런트 데스크 · 실습 교육 / 주임<br>퇴사사유: 경영악화 | 모교(한국호텔관광전문학교)가 설립한 호텔에서 프런트 고객 응대와 재학생 현장 교육을 병행 <ul style="list-style-type: none"><li>체크인·아웃, 예약·객실 배정, 컴플레인 1차 대응 및 하우스키핑 ·F&amp;B 부서 간 요청 조율</li><li>재학생 대상 프런트 업무(PMS·응대 절차) 교육 — 업무를 '남이 따라 할 수 있는 절차'로 정리하는 습관을 익힘</li></ul>                                                                          |
| '19.11 ~ '20.12<br>(1년 2 개월) | <b>코트야드 메리어트 수원</b><br>프런트 데스크 / 사원<br>퇴사사유: 건강상의 문제          | 글로벌 체인 호텔 프런트에서 체크인·아웃, 예약·객실 데이터 관리, 외국인 고객 응대 <ul style="list-style-type: none"><li>노쇼·오버부킹 대응 등 표준 운영 절차(SOP) 기반 응대 경험</li></ul>                                                                                                                                                                          |

교육사항

|                                  |                                         |                                                                                                                                                            |
|----------------------------------|-----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| '26.04 ~ '26.10<br>(10.27 수료 예정) | <b>대한상공회의소</b><br>IBM × Red Hat AX 아카데미 | AI Ops · AI 서비스 개발 과정, 960 시간 <ul style="list-style-type: none"><li>Python·FastAPI 백엔드, RAG·LLM 에이전트, 클라우드 (AWS)·Docker·Kubernetes 운영, 팀 프로젝트 수행</li></ul> |
|----------------------------------|-----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|

# 자기소개서

류준 · AI 서비스 · 백엔드 개발자 지원

## Q1. 성장과정에 대해 서술하십시오.

「만드는 사람이 아니라 검증하는 사람이 되고 싶었습니다」 — 2026 년 대한상공회의소 IBM × Red Hat AX 아카데미(960 시간)에 입과하면서 제가 어떤 개발자가 되고 싶은지가 분명해졌습니다.

과정 초반에는 LLM 을 붙이면 무엇이든 그럴듯하게 동작하는 것이 신기했습니다. 그러나 첫 팀 프로젝트에서 데모는 잘 되는데 "이게 정말 맞게 답한 건가"를 아무도 설명하지 못하는 순간을 겪었습니다. 이전 사회 경험에서 저는 늘 숫자로 보고하고 기준을 세워 사람을 설득하는 자리에 있었기 때문에, 근거 없이 '잘 됩니다'라고 말하는 것이 불편했습니다. 그때부터 제 관심은 '어떻게 만들까'에서 '어떻게 믿게 만들까'로 옮겨갔습니다.

그래서 모든 프로젝트에 네 가지 규칙을 적어 두고 지켰습니다. 측정하지 않은 수치는 적지 않는다, LLM 을 채점자로 쓰지 않는다, 실패를 지우지 않는다, 경계는 사람이 아니라 기계가 지킨다. 이 규칙으로 사내보안 RAG Secu-Agent(개인)에서는 골든셋 30 건으로 Recall@10 0.867 · MRR@10 0.543 을 측정해 공개했고, 다산콜센터 상담원 어시스트 RAG CallGuard(팀, ai/ 담당)에서는 골든셋 295 건과 CI 기준선 게이트로 검색 품질의 퇴행을 막았습니다. 팀장을 맡은 REMAKE DAY에서는 LLM 이 실패해도 게임이 무너지지 않는 Agent Harness 를 코드로 만들었습니다.

수업과 별개로 개인 프로젝트 1 개와 팀 프로젝트 7 개를 개인 도메인 whoareryu.cloud 에 올려 24 시간 공개 운영하고 있습니다. 저는 코드를 쓰는 능력만큼 "이 숫자를 믿어도 되는가"를 먼저 묻는 개발자로 성장하고 있습니다.

## Q2. 성격에 대해 서술하십시오.

「짧은 대화에서 요점을 잡아내고, 기준을 세워 결론을 내는 사람」입니다.

저는 사람을 상대하는 자리에서 사회생활을 시작했습니다. 호텔 프런트에서는 체크인 줄이 밀린 몇 분 안에 고객이 정말 원하는 것이 무엇인지 — 객실 변경인지, 사과인지, 그저 빨리 쉬고 싶은 것인지 — 를 가려내야 했고, 매장 운영에서는 클레임이 들어왔을 때 감정 뒤에 있는 실제 요점(음식인지, 대기 시간인지, 응대 태도인지)을 먼저 찾아야 구조를 고칠 수 있었습니다. 이 과정에서 상대의 말을 끝까지 듣고, 들은 내용을 한 문장으로 되돌려 확인하는 습관이 몸에 뱌습니다.

이 성향은 개발에서도 그대로 작동합니다. 팀 회의에서 요구사항이 길어질 때 "그러니까 지금 필요한 건 A 이고, B 는 다음에 해도 된다는 뜻이죠?"라고 되묻고 표로 정리해 공유합니다. localhost:daegu에서는 기획서와 보유 데이터가 어긋났을 때 '데이터로 답할 수 있는 질문 / 없는 질문' 표를 만들어 논의의 기준을 세웠고, REMAKE DAY에서는 팀장으로서 API 계약을 먼저 문서로 고정한 뒤 5 명이 병렬로 개발하게 했습니다.

또 저는 결론을 미루지 않는 편입니다. 다만 그 결론은 감이 아니라 측정에서 나와야 한다고 믿습니다. 의도를 정확히 파악하고, 기준을 세우고, 숫자로 결론을 내는 — 그래서 팀이 같은 것을 보고 같은 방향으로 움직이게 만드는 개발자가 되려 합니다.

## Q3. 갈등 해결 경험에 대해 서술하십시오.

갈등은 '취향'이 아니라 '기준 부재'에서 온다고 생각합니다. 기준을 표로 만들어 보여주는 방식으로 팀의 방향 갈등을 풀었습니다.

[상황] 2026 AI Blockchain Challenge in Daegu 출품작 localhost:daegu(팀 3 인)는 기존 프로젝트 코드를 재사용해 시작한 탓에 기능은 많은데 "무엇을 푸는 서비스인가"가 비어 있었습니다. 기획서에는 창업 입지 추천 같은 기능이 있었지만 실제 보유한 데이터는 인허가 이력과 행정동별 개폐업 집계뿐이었고, 입지 판단에 필요한 카드 매출·유동인구 데이터는 없었습니다. 기획서대로 가자는 의견과 데이터로 답할 수 있는 범위로 줄이자는 제 의견이 코드 프리즈 일주일 전에 갈렸습니다.

[행동] 감정이 아니라 기준으로 논의하기 위해 '현재 데이터로 답할 수 있는 질문 / 없는 질문' 표를 만들어 공유했습니다. 데이터가 없는 기능은 LLM 이 채우면 '그럴듯한 거짓말'이 된다는 점을 보여주고, 보유 데이터와 결정론 계산으로 답할 수 있는 흐름을 엔드포인트 단위로 정리했습니다. 상대 의견 중 "예비 창업자가 바로 쓸 수 있어야 한다"는 요구는 버리지 않고 창업자금 점검 동선을 그 대상에 맞춰 설계하는 방식으로 흡수했습니다. 계산·비교표는 LLM 없이 코드가 만들고 LLM 은 설명만 쓰게 하자는 합의도 이 표에서 나왔습니다.

[결과] 데이터가 없는 기능을 덜어내고 '대구 예비 창업자를 위한 창업자금 사전상담 도우미'로 범위를 좁혀 접수 마감 내 제출했습니다. 제출 전 배포 주소에서 가상 고객 10명 시나리오를 돌려 결함 7건을 찾아 고쳤고, 10명 전원이 목표에 도달하는 것을 확인했습니다.

[배운 점] 말로 설득하기보다 '비교할 수 있는 것'을 먼저 만들어 놓았을 때 합의가 빨랐습니다. 그리고 상대 안을 통째로 버리지 않고 한 조각을 흡수할 때 갈등이 협업으로 바뀐다는 것을 배웠습니다.

#### Q4. 핵심역량에 대해 서술하십시오.

AI 서비스 백엔드 개발자의 핵심역량은 「LLM을 검증 장치로 감싸는 하네스 설계」와 「숫자로 검증하는 평가」라고 생각합니다.

첫째, 하네스 엔지니어링. "하네스는 프롬프트가 아니라 코드"라는 원칙으로 일합니다. REMAKE DAY에서는 LLM 응답을 JSON 파싱 → 스키마 검증 → 사실 증 검사 → 재생성 2회 → 폴백으로 처리하는 Agent Harness를 만들어 LLM이 실패해도 게임이 무너지지 않게 했고, 모델이 스키마 상한을 어기면 하네스에서 절단하도록 구현했습니다. 포트폴리오 허브 채팅에서는 근거 유사도 문턱 0.67 아래면 생성 자체를 하지 않고, IP당 하루 20회·하루 생성 200회 상한을 코드로 강제해 환각과 비용을 함께 막았습니다. 둘째, 평가와 숫자. LLM을 채점자로 쓰지 않고 골든셋과 규칙 기반 채점으로 측정합니다. CallGuard에서는 골든셋 295건으로 BM25 기준선(Recall@5 0.853 · MRR 0.686)을 먼저 재고, 하이브리드 검색으로 0.971 · 0.875까지 올린 뒤 3회 최저치 고정과 CI 기준선 게이트로 퇴행을 막았습니다. Secu-Agent에서는 MRR이 0.526↔0.543으로 흔들리는 원인을 의존성 잠금 누락으로 찾아 고정했고, 데이터 정리 후 지표가 내려갔을 때도 그 값을 그대로 공개했습니다.

셋째, 범위를 정하는 문제 정의. 요구사항을 '우리 데이터로 답할 수 있는가'로 걸러 데이터가 없는 기능은 덜어내고, 계산은 결정론 코드·설명만 LLM으로 분리합니다(localhost:daegu · 발자국). 팀 프로젝트 7개와 해커톤 4회에서 이 세 역량을 반복해 검증했습니다.

#### Q5. 직무 수행 경험에 대해 서술하십시오.

CallGuard — 서울시 120 다산콜센터 실시간 상담원 어시스트 RAG(팀 4인, 2026.08~10)에서 ai/ 영역을 담당했습니다. 청킹 · BM25 · 하이브리드 검색 · 임베딩 · 평가 하네스가 제 몫이었고, 백엔드 구조와 개인정보 마스킹, 프론트 화면은 팀원이 맡았습니다.

[상황(S)] 상담원이 통화 중 민원 유형에 맞는 안내 문서를 몇 초 안에 받아야 하는 서비스였습니다. 공공데이터 CSV를 정규화해 적재한 뒤 처음 붙인 검색은 BM25 단독이었고, 골든셋 295건 기준 Recall@5 0.853 · MRR 0.686이었습니다. 정답 문서가 5위 안에 없으면 상담원은 통화 중에 그 안내를 할 수 없습니다.

[과제(T)] 검색 품질을 올리되, 올라갔다는 것을 팀원 누구나 재현할 수 있는 숫자로 증명해야 했습니다. 특히 LLM에게 "잘 찾았는지" 물어보는 방식은 쓰지 않기로 했습니다.

[행동(A)] 먼저 평가 하네스를 만들었습니다. 골든셋 295건, 규칙 기반 자동채점, 같은 구성을 3회 돌려 최저치를 기록값으로 고정, 재현되지 않는 항목은 "측정 불가"로 따로 표시, CI에서 기준선 아래로 떨어지면 병합을 막는 게이트. 그 위에서 청킹 단위를 조정하고, nori 형태소 분석 기반 BM25에 KoE5 dense\_vector를 결합한 하이브리드 검색을 구현해 구성별로 비교했습니다.

[결과(R)] 운영 구성 기준 Recall@5 0.971 · MRR 0.875(2026-09-22 실측)로 BM25 기준선 대비 Recall@5 +0.118, MRR +0.189를 재현 가능한 수치로 기록했습니다. 이후 팀원의 변경이 검색 품질을 떨어뜨리면 CI가 자동으로 막아, 품질이 사람의 주의가 아니라 기계에 의해 지켜지는 상태가 되었습니다.

[배운 점(L)] 개선보다 측정이 먼저입니다. 재현되지 않는 수치 위에서 한 개선은 개선이 아니고, 기준선이 있어야 팀이 같은 숫자를 보고 논의할 수 있습니다. [연결(C)] 입사 후에도 담당 서비스의 검색 정확도·응답 지연·토큰 비용을 먼저 골든셋과 기준선으로 정리해 "개선했다"를 숫자로 보고하는 방식으로 일하겠습니다.

#### Q6. 문제해결 경험에 대해 서술하십시오.

「같은 코드인데 지표가 흔들린다」 — 수치를 믿을 수 있게 만든 뒤에야 개선을 시작했습니다.

[상황] 개인 프로젝트 Secu-Agent(권한 기반 사내보안 RAG)의 검색 품질을 골든셋 30건으로 측정하던 중, 코드를 바꾸지 않았는데 MRR@10이 어떤 날은 0.526, 어떤 날은 0.543으로 나왔습니다. 이 상태로는 개선이 효과인지 우연인지 구분할 수 없었고, 포트폴리오에 적을 수치도 결정할 수 없었습니다.

[과제] 지표가 재현되는 상태를 먼저 만들고, 그 위에서 검색 개선과 데이터 정리의 효과를 검증해야 했습니다.

[행동] '한 번에 하나씩' 변수를 고정했습니다. ① 같은 입력·같은 골든셋으로 반복 측정해 흔들림이 코드가 아니라 환경에서 온다는 것을 확인. ② 의존성 버전을 하나씩 대조해 잠금 파일이 누락된 라이브러리가 원인임을 찾아 고정하자 측정값이 0.543으로 재현. ③ 재현되는 상태에서 PDF의 깨진 글자 2,062자를 정리했는데 지표가 오히려 내려갔습니다. 정리 전으로 되돌려 수치를 지키는 대신 내려간 값을 그대로 공개하고 원인을 문서에 남겼습니다. ④ 권한 경계가 정말 지켜지는지는 강제 테스트·권한 누출 테스트를 코드로 두고, 변이 검사로 테스트 자체의 유효성까지 확인했습니다.

[결과] Recall@10 0.867 · MRR@10 0.543을 재현 가능한 수치로 공개했고, 이 경험을 CallGuard 평가 하네스(3회 최저치 고정 · "측정 불가" 구분 · CI 게이트)로 옮겨 팀 프로젝트의 기본 규칙으로 만들었습니다. [배운 점] 흔들리는 지표로 한 개선은 개선이 아닙니다. 측정이 재현되는 상태를 먼저 만드는 것이 가장 빠른 길이고, 내려간 숫자를 지우지 않는 것이 다음 숫자를 믿게 만든다는 것을 배웠습니다.

## Q7. 지원 동기 및 입사 후 포부에 대해 서술하시오

「설계하고, 배포하고, 숫자로 검증하는 AI 백엔드 개발자」 — 제가 가진 세 가지 강점으로 귀사의 AI 서비스가 '동작하는' 수준을 넘어 '믿을 수 있는' 수준이 되는 데 기여하겠습니다.

[지원 동기 · 보유 강점] 개인 프로젝트 1개와 팀 프로젝트 7개, 해커톤 4회(팀장 2회)를 거치며 세 가지를 갖췄습니다. 첫째, LLM 실패를 흡수하는 하네스를 코드로 만드는 능력(REMAKE DAY Agent Harness, 허브 채팅의 근거 문턱·비용 상한). 둘째, 골든셋과 기준선으로 품질을 측정하고 CI 게이트로 퇴행을 막는 평가 루프(CallGuard Recall@5 0.853 → 0.971). 셋째, 요구사항을 '데이터로 답할 수 있는가'로 걸러 범위를 정하고 팀이 같은 기준을 보게 만드는 소통 방식(localhost:daegu, REMAKE DAY API 계약). 실제 사용자와 트래픽이 있는 귀사에서 이 세 가지를 더 큰 규모로 쓰고 싶습니다.

[입사 후 1년 — 이렇게 기여하겠습니다] 담당 서비스의 검색 정확도·응답 지연·토큰 비용을 골든셋과 기준선으로 먼저 정리해, 모든 변경을 "개선했다"가 아니라 "얼마나 개선했다"로 보고하겠습니다. LLM 호출 경로마다 스키마 검증·폴백·상한을 두는 하네스를 팀 코드베이스에 심고, 설계 → 계획 → 구현 → 리뷰 → 수정의 개발 루프와 인계서를 운영 문서로 남겨 제가 없어도 돌아가는 상태를 만들겠습니다.

[입사 후 3년] RAG·멀티 에이전트 서비스를 설계부터 배포·평가까지 책임지는 백엔드 개발자로, 의존성은 안쪽으로 두고 경계는 테스트가 지키는 구조를 팀의 기본값으로 만들고 싶습니다. 짧은 대화에서 요점을 잡아내는 습관으로 사용자와 동료가 실제로 막히는 지점을 찾고, 데이터가 없는 기능은 덜어내며 핵심을 더 잘 만드는 데 기여하겠습니다.

위 내용은 사실과 다름이 없습니다.

2026년 10월 15일

작성자 류 준 (인)

# 포트폴리오

개인 프로젝트 1 개 · 팀 프로젝트 7 개 — 전체는 <https://whoareryu.cloud/?from=jobkorea> 에서 소개 사이트 · 시연 영상 · GitHub 로 확인할 수 있습니다. 아래는 대표 4 건입니다.

## 포트폴리오 ① whoareryu.cloud — 포트폴리오 허브 · 프로젝트 통합 운영 · AI 채팅 (개인)

|             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |            |                                                                                                       |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|-------------------------------------------------------------------------------------------------------|
| 기간          | '26.10 ~ 현재 (운영 중)                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 참여인원       | 1명 (개인 · 기여도 100%)                                                                                    |
| 개요          | 개인 도메인 아래 프로젝트 8 개(개인 1 · 팀 7)를 서브도메인(secu · remakeday · daegu · jb · adapdog · rpg · research)으로 통합한 포트폴리오 허브. 각 프로젝트는 소개 사이트 + 시연 영상 + GitHub 로 24 시간 공개되고, 메인 화면은 '내 프로젝트·이력을 아는' AI 채팅과 프로젝트 격자로 구성                                                                                                                                                                                                                                                                                          |            |                                                                                                       |
| 링크          | <a href="https://whoareryu.cloud/?from=jobkorea">https://whoareryu.cloud/?from=jobkorea</a>                                                                                                                                                                                                                                                                                                                                                                                                      |            |                                                                                                       |
| 담당 역할       | 설계 · 백엔드 · RAG · 배포 · 보안 · 운영 전담                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Tech stack | Next.js(Vercel), Supabase(pgvector), gemini-embedding-001, Gemini flash-lite, Vercel 방화벽, Claude Code |
| 내용          | <ul style="list-style-type: none"><li>포트폴리오 AI 채팅: 프로젝트·이력 문서 2,000 여 조각을 Supabase pgvector 에 색인하고 벡터+ 키워드 하이브리드 검색으로 근거를 찾아 답변. 근거가 없으면 생성하지 않고 "관련 기록을 못 찾았습니다"로 응답</li><li>운영: 맥북에 의존하던 구조를 Vercel + Supabase 로 이전해 맥북을 꺼도 사이트·채팅이 24 시간 동작, 운영 비용 0 원</li><li>보안·계측: Vercel 방화벽(DDoS 자동 방어, 봇 차단·탐색 경로 기록 규칙), CSP·HSTS 보안 헤더, ?from= 유입 표시 방문 집계 직접 구현</li></ul>                                                                                                                            |            |                                                                                                       |
| 구현기능 및 개발과정 | <ul style="list-style-type: none"><li>① 근거 문턱을 '감'이 아니라 실측으로 결정 — 정상 질문의 유사도 최저 0.698, 엉뚱한 질문의 최고 0.655 를 측정해 그 사이인 0.67 을 문턱으로 고정. 문턱 아래면 LLM 을 호출하지 않아 환각과 비용을 함께 차단</li><li>② 비용 하네스 — IP 당 하루 20 회 · 하루 생성 200 회 상한을 코드로 강제하고, 한도 초과 시 생성 대신 근거 문서 목록을 반환해 서비스가 멈추지 않게 설계</li><li>③ 응답 지연 개선 — 생성 모델을 flash 에서 flash-lite 로 바꿔 응답 11.5 초 → 3 초. 같은 질문 세트로 품질을 비교한 뒤 교체</li><li>④ 인프라 이전 — 맥북 상시 가동이 필요하던 구조를 걷어내고 Vercel + Supabase 로 재배포. 배포 파일 편집 전 런북을 강제하는 훅을 두어 실수로 인한 장애를 예방</li></ul> |            |                                                                                                       |
| 성과          | <ul style="list-style-type: none"><li>프로젝트 8 개를 한 도메인·비용 0 원으로 24 시간 공개 운영 — 신규 프로젝트는 서브도메인 등록만으로 추가</li><li>근거 없는 질문에는 생성하지 않는 채팅(문턱 0.67 실측 결정), 응답 11.5 초 → 3 초</li><li>채용 담당자가 이력서 대신 AI 채팅으로 프로젝트를 질문할 수 있는 포트폴리오</li></ul>                                                                                                                                                                                                                                                                 |            |                                                                                                       |
| 느낀 점        | <ul style="list-style-type: none"><li>상황/과제: 만든 서비스를 꺼지지 않게, 비용 없이, 근거 없는 답을 하지 않게 운영하고 싶었음</li><li>행동/결과: 인프라 이전, 근거 문턱·비용 상한을 코드로 강제, 응답 지연 1/4 로 단축</li><li>배운 점(L): "하네스는 프롬프트가 아니라 코드"다. 프롬프트로 부탁한 규칙은 희망이고, 코드로 강제한 규칙만 지켜진다</li><li>연결(C): 배포·비용·품질 지표까지 숫자로 보고하는 백엔드 개발자로 일하겠음</li></ul>                                                                                                                                                                                                 |            |                                                                                                       |

## 포트폴리오 ② Secu-Agent — 권한 기반 사내보안 RAG (개인)

|    |                                                                                                                       |      |                    |
|----|-----------------------------------------------------------------------------------------------------------------------|------|--------------------|
| 기간 | '26.08.31 ~ '26.09.15                                                                                                 | 참여인원 | 1명 (개인 · 기여도 100%) |
| 개요 | 사용자 권한에 따라 검색 결과가 달라지는 사내 문서 RAG. 사후 필터링이 흘리는 "못 보는 문서가 있다"는 신호까지 사전 필터링으로 감추고, LLM 심판 없이 골든셋으로 검색 품질을 측정해 지표를 그대로 공개 |      |                    |
| 링크 | <a href="https://secu.whoareryu.cloud">https://secu.whoareryu.cloud</a>                                               |      |                    |

|             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |            |                                                                                         |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|-----------------------------------------------------------------------------------------|
| 담당 역할       | 기획 · 백엔드 · 프론트 · 평가 하네스 · 배포 · 문서 전부                                                                                                                                                                                                                                                                                                                                                                                                                                 | Tech stack | FastAPI, Next.js, pgvector, LangChain/LangGraph, multilingual-e5-small(임베딩), Gemini(생성) |
| 내용          | <ul style="list-style-type: none"> <li>검색: 벡터 검색 + 키워드 검색을 RRF 로 융합, 권한 조건을 검색 쿼리 단계에서 적용하는 사전 필터링</li> <li>평가: 골든셋 30 건 기준 Recall@10 0.867 · MRR@10 0.543 — LLM 을 채점자로 쓰지 않는 규칙 기반 채점</li> <li>검증: 경계 강제 테스트, 권한 누출 테스트, 서브에이전트 코드 리뷰, 변이 검사</li> </ul>                                                                                                                                                                                                             |            |                                                                                         |
| 구현기능 및 개발과정 | <p>① 사후 필터링 → 사전 필터링 — 검색 후 권한 밖 문서를 제거하면 결과 수·순위 변화로 "볼 수 없는 문서가 존재한다"는 사실이 새어 나감. 권한 조건을 검색 전에 적용해 신호 자체를 없애고, 권한 누출 테스트로 기계가 경계를 지키는지 확인</p> <p>② 흔들리는 지표의 원인 고정 — 같은 코드인데 MRR 이 0.526↔0.543 으로 바뀌는 현상을 추적해 의존성 잠금 파일 누락이 원인임을 찾아 고정. 이후 측정값이 재현되는 상태에서만 수치를 기록</p> <p>③ 실패를 지우지 않는다 — PDF 에서 깨진 글자 2,062 자를 정리하자 지표가 오히려 내려갔지만, 정리 전 수치로 되돌리지 않고 내려간 값을 그대로 공개하고 원인을 문서에 남김</p> <p>④ 검증 하네스 — 경계 강제·권한 누출 테스트를 코드로 두고, 서브에이전트 코드 리뷰와 변이 검사로 테스트 자체의 유효성까지 점검</p> |            |                                                                                         |
| 성과          | <ul style="list-style-type: none"> <li>권한 누출 0 건을 테스트로 보장하는 RAG — 사전 필터링으로 "못 보는 문서가 있다"는 신호까지 차단</li> <li>재현 가능한 평가 지표 공개: Recall@10 0.867 · MRR@10 0.543 (골든셋 30 건, LLM 심판 없음)</li> <li>기획부터 배포·문서까지 1 인으로 완결, 포트폴리오 허브에서 24 시간 시연 가능</li> </ul>                                                                                                                                                                                                                   |            |                                                                                         |
| 느낀 점        | <ul style="list-style-type: none"> <li>상황/과제: "권한을 지킨다"를 말미 아니라 테스트와 숫자로 증명해야 했음</li> <li>행동/결과: 사전 필터링 + 권한 누출 테스트, 흔들리는 지표의 원인 고정, 낮아진 지표도 그대로 공개</li> <li>배운 점(L): 측정값이 재현되지 않으면 그 수치는 아직 사실이 아니다. 경계는 사람의 주의가 아니라 기계가 지켜야 한다</li> <li>연결(C): 지표를 좋게 보이게 만드는 것이 아니라 믿을 수 있게 만드는 개발자가 되겠음</li> </ul>                                                                                                                                                               |            |                                                                                         |

**포트폴리오 ③ REMAKE DAY — LLM 에이전트 설득 게임 (팀 BeyondBob 5 인 · 원티드 AI Championship 2026 출품작)**

|       |                                                                                                                                                                                                                    |            |                                                                           |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|---------------------------------------------------------------------------|
| 기간    | '26.09.03 ~ '26.09.20                                                                                                                                                                                              | 참여인원       | 5 명 (팀 BeyondBob)                                                         |
| 개요    | 같은 하루를 반복하며 NPC 를 설득하는 LLM 에이전트 게임. 원티드 AI Championship 2026 출품(결과: 본선 미진출 · 33 위 · 103 표). 연구 사이트(research.whoareryu.cloud)에서는 AI 가 규칙을 제안하고 사람이 수락·거절·직접 작성하는 루프를 기록                                             |            |                                                                           |
| 링크    | <a href="https://remakeday.whoareryu.cloud">https://remakeday.whoareryu.cloud</a> · <a href="https://research.whoareryu.cloud">https://research.whoareryu.cloud</a>                                                |            |                                                                           |
| 담당 역할 | 팀장 겸 AI Agent Engineer — 아키텍처 · API 계약 · Agent Harness · LLM 역할 프롬프트 · 발표                                                                                                                                          | Tech stack | Core claude-sonnet-5, NPC claude-haiku-4-5, gemini-embedding-001, FastAPI |
| 내용    | <ul style="list-style-type: none"> <li>[팀이 한 일] 게임 루프·NPC·UI·연구 사이트 구현, 모델 평가 수치 산출(장민석)</li> <li>[내가 맡은 일] 멀티 에이전트 아키텍처와 API 계약 정의, LLM 출력이 실패해도 게임이 무너지지 않게 하는 Agent Harness, Core·NPC 역할 프롬프트, 최종 발표</li> </ul> |            |                                                                           |

|             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 구현기능 및 개발과정 | <p>① <b>Agent Harness</b> — LLM 응답을 JSON 파싱 → 스키마 검증 → 사실 총 검사 → 재생성 최대 2 회 → 폴백 순으로 처리하는 파이프라인을 코드로 구현. "LLM 이 실패해도 게임이 안 무너진다"를 설계 목표로 둠</p> <p>② <b>스키마 상한 절단</b> — 모델이 길이·개수 상한을 어기는 경우가 반복돼, 재생성에 기대지 않고 하네스에서 상한에 맞춰 절단하도록 구현해 응답 실패율과 지연을 함께 줄임</p> <p>③ <b>역할별 모델 분리</b> — 핵심 판정은 claude-sonnet-5, 대량 호출되는 NPC 대사는 claude-haiku-4-5 로 나눠 품질과 비용을 분리. 모델 간 계약은 API 스키마로 고정해 팀원이 병렬로 개발</p> <p>④ <b>제품 루프 설계</b> — 하루 반복 루프와 AI 제안→사람 승인 루프를 분리해 정의하고, 연구 사이트에서 사람 판 14 판 · 적용 규칙 60 건 중 AI 계열 48 건(탐색적 관찰)을 기록</p> |
| 성과          | <ul style="list-style-type: none"> <li>원티드 AI Championship 2026 출품 완료 — 함께 만든 후보작(에르덴 용병단) 중 검증 후 REMAKE DAY 를 선택해 제출</li> <li>LLM 출력 실패를 폴백으로 흡수하는 하네스로 시연 중 게임 중단 없이 발표 진행</li> <li>"측정하지 않은 것은 주장하지 않는다" — 연구 사이트에 관찰 결과를 탐색적 관찰로만 기록</li> </ul>                                                                                                                                                                                                                                                                       |
| 느낀 점        | <ul style="list-style-type: none"> <li>상황/과제: 5 인 팀의 방향·계약·발표를 맡으며 LLM 불확실성을 제품 품질로 넘기지 않아야 했음</li> <li>행동/결과: API 계약을 먼저 고정해 병렬 개발, 하네스로 실패를 흡수, 후보작 2 개 중 검증으로 선택</li> <li>배운 점(L): LLM 은 검증 장치로 감싸야 제품이 된다. 강제 수단 없는 규칙은 희망일 뿐이다</li> <li>연결(C): 에이전트 서비스에서 실패 경로까지 설계하는 백엔드 개발자가 되겠음</li> </ul>                                                                                                                                                                                                                       |

포트폴리오
④ localhost:daegu — 대구 예비 창업자 창업자금 사전상담 도우미 (팀 3 인 · 2026 AI Blockchain Challenge in Daegu 출품작)

|             |                                                                                                                                                                                                                                                                                                                                                                                |            |                                                |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|------------------------------------------------|
| 기간          | '26.09 ~ '26.10                                                                                                                                                                                                                                                                                                                                                                | 참여인원       | 3 명 (장민석 팀장 · 김충식 · 류준)                        |
| 개요          | iM BANK 주최 2026 AI Blockchain Challenge in Daegu 출품작(접수 완료 · 본선 미진출). 대구 예비 창업자가 창업자금을 점검하고 비교표를 받는 사전상담 도우미. 계산·비교표는 LLM 없이 결정론 코드로 만들고, LLM 은 설명만 쓰게 설계                                                                                                                                                                                                                      |            |                                                |
| 링크          | <a href="https://whoareryu.cloud/?from=jobkorea">https://whoareryu.cloud/?from=jobkorea</a> (허브 내 daegu 서브도메인)                                                                                                                                                                                                                                                                 |            |                                                |
| 담당 역할       | 기능 단위 풀스택 — 자금 점검 동선 · 계산 가정 표시 · 데이터 신뢰 표시 · 프론트엔드 배포                                                                                                                                                                                                                                                                                                                         | Tech stack | Next.js, FastAPI, PostgreSQL(pgvector), Gemini |
| 내용          | <ul style="list-style-type: none"> <li>[팀이 한 일] 서비스 전체 설계·구현, 결정론 계산 엔진, 임베딩 모델 비교(김충식)</li> <li>[내가 맡은 일] 자금 점검 동선(입력 → 점검 → 비교표) 구현, 계산에 쓰인 가정을 화면에 드러내는 '계산 가정 표시', 출처·갱신일 기준 '데이터 신뢰 표시', 프론트엔드 배포</li> </ul>                                                                                                                                                              |            |                                                |
| 구현기능 및 개발과정 | <p>① <b>데이터로 답할 수 있는 질문만 기능으로</b> — 보유 데이터(인허가 이력·개폐업 집계)로 답할 수 있는 질문과 없는 질문을 표로 정리해 공유하고, 데이터가 없는 기능은 떨어내는 쪽으로 범위를 정리</p> <p>② <b>계산은 코드, 설명은 LLM</b> — 금액·비교표가 LLM 생성에 섞이면 검증이 불가능해지므로 계산 경로에서 LLM 을 제거하고 설명 문장만 LLM 이 쓰게 분리. 계산 가정(기간·비율)을 화면에 그대로 표시해 사용자가 검증 가능하게 함</p> <p>③ <b>배포 주소에서 검증</b> — 가상 고객 10 명 시나리오로 배포 주소에서 동선을 끝까지 실행해 결함 7 건을 발견·수정, 10 명 전원 목표 도달 확인</p> |            |                                                |
| 성과          | <ul style="list-style-type: none"> <li>접수 마감 내 제출 완료 — 결정론 계산 + LLM 설명 분리 구조로 숫자가 재현되는 상담 도우미 구현</li> <li>배포 환경 가상 고객 10 명 검증에서 결함 7 건 발견·수정, 전원 목표 도달</li> </ul>                                                                                                                                                                                                              |            |                                                |
| 느낀 점        | <ul style="list-style-type: none"> <li>상황/과제: 숫자를 다루는 서비스에서 LLM 의 그럴듯한 답이 가장 큰 위험이었음</li> <li>행동/결과: 계산과 설명을 분리하고 가정을 드러내 검증 가능한 구조로 제출</li> <li>배운 점(L): 가장 비싼 작업은 코딩이 아니라 '지우기'다. 데이터가 없는 기능은 없는 편이 낫다</li> <li>연결(C): 요구사항을 '우리 데이터로 답할 수 있는가'로 먼저 걸러 불필요한 개발을 줄이겠음</li> </ul>                                                                                              |            |                                                |