임 태 환

가비아 AI전략개발팀에서 AI 채팅 서비스를 개발합니다.
프로덕트 관점에서 사고하며, 주인 의식을 갖고 일합니다.
사용자의 경험을 개선하고, 저의 노력이 팀과 제품에 기여하는 과정을 즐깁니다.

경력 4년 7개월

(주)가비아

2023.03 - 재직중 (3년 9개월) | 정규직 | AI전략개발팀 / 하이웍스개발팀 | 백엔드 개발

1. [AI채팅] SKILL 및 MCP 도구 개발

2026.10 - 2026.11 | AI전략개발팀

[MCP 서버]
하이웍스에는 메일, 전자결재, 게시판 등의 솔루션이 존재합니다.
AI 채팅에서 휴가 신청, 읽지 않은 메일 조회와 같은 기능을 사용하기 위해 기존 API를 호출하는 SDK를 구성하였습니다.
기존 API 호출은 SDK에서 관리하고, MCP 서버가 SDK를 사용하도록 역할을 나눴습니다.
AI 채팅에서는 MCP 서버만 호출하도록 하고, SDK는 별도로 버전을 관리할 수 있게 구성했습니다.

[한글 파일 생성]
공공기관에서 여전히 한글 파일을 많이 사용하는 점을 고려했습니다.
샌드박스 Docker 환경에서 rhwp 라이브러리를 검토하고, AI 채팅에서 한글 파일을 생성하는 스킬을 개발해 배포했습니다.

[신규 모델 관리 스킬]
GPT, Claude, Gemini 모델의 최신 버전이 빠르게 배포되고 있습니다.
모델을 추가할 때마다 정산 스케줄러와 API의 백엔드, 프론트엔드 코드 작업이 필요했습니다.
모델 정보는 백엔드에서 관리하도록 책임을 모았습니다.
벤치마크 점수가 높아도 내부 환경에서의 품질은 다를 수 있어, 별도 검증이 필요했습니다.
API와 스케줄러의 모델 코드 추가와 내부 환경의 정성 평가를 자동화하는 스킬을 개발했습니다.

https://www.hiworks.com/manual/hiworks/24224

2. [AI채팅] 플랫폼 스펙 통합

2026.09 - 2026.10 | AI전략개발팀

AI 채팅은 여러 프로바이더의 모델을 제공하고 있습니다.
LLM에 질문을 할 때는 추론, 이전 대화 이력, 도구 사용 이력 등이 담기게 됩니다.
사용자가 용도별로 모델을 전환할 수 있도록 플랫폼의 스펙을 통합하는 작업을 진행했습니다.

- 플랫폼마다 텍스트, 추론, 도구 호출을 다루는 형식이 달라 각 응답을 공통으로 사용할 block 이라는 개념을 두어 변환하고, 다음에

사용할 모델의 요청 형식으로 되돌리는 계층을 만들었습니다.
- 승인을 기다리는 도구 호출이 남아 있으면 모델 전환을 제한하도록 정책을 정했습니다.
- 요청 헤더로 기존 응답과 새 응답을 구분하는 버저닝을 지원했습니다. 프론트엔드에서 순차적으로 반영할 수 있도록 했으며, 현재는 전체 배포를 마치고 안정화된 상태입니다.

사용 기술: TypeScript, NestJS, MongoDB, LLM API, SSE, MCP

3. [AI채팅] 장기 기억 메모리

2026.07 - 2026.09 | AI전략개발팀

사용자의 습관과 반복되는 작업을 기억해 이후 답변에 반영하는 기능을 개발했습니다.
기존 LRU 방식만으로는 오래된 선호와 현재 필요한 정보를 구분하기 어려워, 기억의 저장부터 수정, 활용, 정리까지 기준을 설계했습니다.

[저장 기준과 갱신]
잘못 저장된 정보가 이후 답변에도 영향을 줄 수 있어, 어떤 정보를 기억할지 기준이 필요했습니다.
AI의 답변을 사용자 사실로 기억하거나 민감정보와 제3자 정보를 저장하는 오추출을 보완하기 위해 추출 프롬프트와 저장 필터를 수정했습니다.
자연어로 기억을 바꿀 때는 기존 기억을 먼저 찾아 추가/수정/삭제를 구분하도록 했습니다. 상충하는 선호가 함께 남지 않도록 기존 기억과 새 사실을 병합하는 흐름을 구성했습니다.
자동 추출과 자연어 도구가 같은 저장 로직을 사용하도록 통합했습니다.

[중요도와 생명주기]
검색으로 프롬프트에 노출된 횟수와 실제 답변에 활용된 횟수를 구분해 관리했습니다.
시간 반감기와 활용 빈도를 반영한 가중합 모델로 중요도를 계산하고, 활용도가 낮아진 기억을 아카이브하는 기준을 설계했습니다.
시간이 지나면 중요도가 달라지므로, 점수를 고정값으로 저장하지 않고 조회할 때 계산하도록 설계했습니다.

[대화 문맥을 반영한 검색]
후속 질문은 현재 질문만으로 대상을 특정하기 어려워, 같은 세션의 요약과 최근 대화를 검색 질의에 반영했습니다.
세션 요약은 장기 기억으로 다시 저장하지 않고 검색 문맥으로 사용했으며, 문맥이 없거나 조회에 실패한 경우 현재 질문으로 검색하도록 했습니다.
활용 이력과 같은 메타데이터는 검색 벡터와 분리해 관리하고, 기억의 본문이 바뀌면 Milvus의 벡터를 다시 생성하도록 구성했습니다.

[사용자 관리와 정책]
사용자가 저장된 기억을 확인하고 고칠 수 있도록 조회/수정/삭제 동작을 제안하고, 목업으로 협의했습니다.
관리 API와 자연어 도구를 개발하고, 오피스 관리자 정책과 사용자 설정의 지침이 서로 간섭하지 않도록 프롬프트를 정리했습니다.

[호출 비용]
메모리 추출, 기존 기억 병합, 활용 판정 단계의 호출 조건과 토큰 소모량을 분석했습니다.
짧은 인사와 단답은 추출 대상에서 제외하고, 기존 기억과 겹치지 않는 신규 사실이 하나면 병합 LLM 호출 없이 저장하는 조건을 정리했습니다.

사용 기술: TypeScript, NestJS, MongoDB, Milvus, BGE-M3, RAG

https://www.hiworks.com/manual/hiworks/24224/26420

4. [AI채팅] 예약 작업

2026.06 - 2026.07 | AI전략개발팀

사용자가 등록한 프롬프트를 지정한 시간에 실행할 수 있도록 AI 채팅 API와 예약 작업을 연동했습니다.
여러 배치 저장소에 분산되어 있던 스케줄링을 통합 스케줄러에서 관리하도록 구성했습니다.
MongoDB에서 작업을 선점하고, 예약 ID와 예정 실행 시각을 조합한 실행 ID에 고유 인덱스를 두어 여러 파드에서 같은 작업이 중복 실행되지 않도록 설계했습니다.
예약 작업은 병렬로 실행하도록 구현했습니다.
LLM의 응답은 SSE 방식으로 수신하도록 처리했습니다.
업무 시작 시간대에 집중되는 약 4천 건의 예약 작업(1시간 기준 약 67 RPM)을 처리하고 있습니다.

https://www.hiworks.com/manual/hiworks/24224/26141

5. [AI채팅] 보안 관리

2026.04 - 2026.06 | AI전략개발팀

기업용 AI 채팅에서는 민감정보가 LLM에 전달되는 것을 우려하는 고객이 많습니다.
Presidio 서버를 구성하여 NLP로 민감정보를 탐지하고, 처리된 데이터만 LLM에 전달하도록 했습니다.

이후 사용자 입력, 첨부파일, 도구 결과에 보안 정책이 적용되는 경로를 검토했습니다.
메모리 검색과 후속 요청에서도 민감정보 처리 정책이 유지되는지 점검했습니다.
로컬 부하 테스트로 긴 입력과 높은 동시성에서 발생하는 Presidio의 지연과 실패를 확인했습니다.

https://www.hiworks.com/manual/hiworks/24224/27000

6. [AI채팅] 일일 브리핑 PoC

2026.01 - 2026.03 | AI전략개발팀

메일과 게시글 등 업무 데이터를 개인별 브리핑으로 제공하는 PoC를 진행했습니다.
실제 고객 규모를 바탕으로 일 500만 건의 업무 데이터와 사용자 15만 명을 설계 기준으로 삼았습니다.
검증 데이터를 라벨링하고 LLM 모델별 성능을 평가했습니다.

- 피크 시간대의 처리 구조를 검토했습니다.
- 항목별 요약과 최종 브리핑 생성을 일괄 처리했을 때 월 예상 비용이 약 45% 낮아지는 것으로 산정했습니다.

검증 과정에서 범용 기준만으로는 사용자에게 의미 있는 항목을 고르기 어려웠습니다.
개인화 데이터가 먼저 필요하다고 판단해 메모리와 예약 작업을 제안하고, 작업 방향을 전환했습니다.

7. [하이웍스] 전자결재, 관리자 권한 작업

2024.12 - 2025.11 | 하이웍스개발팀

[전자결재 그룹사 기능 확장]
기존 전자결재는 하나의 회사 안에서 사용하는 구조였습니다.
지주사와 계열사가 함께 사용할 수 있도록 그룹사 기능을 확장했습니다.
회사별 조직과 권한을 나누되, 기존 고객이 사용하는 모바일 앱과 외부 연동 API는 계속 사용할 수 있어야 했습니다.

기존 API의 변경 범위를 줄이기 위해 포트-어댑터 패턴을 적용했습니다.
그룹사 기능에서 달라지는 부분을 분리하고 기존 연동 경로를 유지하였습니다.

[관리자 권한 개편 | 2025.08 ~ 2025.10]
기존 IngressAuth에서는 사용자 인증과 관리자 여부를 확인하고 있었습니다.
관리자별로 메뉴 접근 권한을 나누고, 그룹 단위로 권한을 부여할 수 있도록 구조를 개편했습니다.
공통 권한 체계의 변경으로 하이웍스 전체 저장소에 영향이 있어, 기존 고객의 인증과 연동을 유지하면서 저장소별 적용 범위와 전환 기준을 맞춰야 했습니다.

기존 IngressAuth와 JWT의 역할은 유지하고, 세부 권한은 별도의 인가 API로 확인하는 방식을 선택했습니다.
관리할 API가 늘어나는 부담은 있었지만, 기존 인증 흐름의 변경을 줄이고 서비스별로 전환할 수 있다는 점을 고려했습니다.

- 초기 권한 30개를 기능별로 정의하고, 관리자 그룹, 권한, 그룹별 관리자와 권한의 연결 관계를 데이터 모델로 설계했습니다.
- 관리자 그룹 설정과 소속 관리자 추가/삭제 API, 공통 권한 확인 API, 허용 메뉴 조회 API의 요청과 응답을 정의했습니다. 관리자 설정은 전체 관리자만 접근하고, 각 관리자에게는 허용된 메뉴만 노출하도록 정책을 정리했습니다.
- 기존 인증 정보만으로는 새 권한 구조의 적용 대상 회사를 구분할 수 없어, 캐시 API의 고객 환경 정보로 대상을 구분하도록 설계했습니다.
- 메뉴, 라이선스, 계정, GNB 등 연동 서비스 담당자와 영향 범위와 권한 조회 규격을 맞췄습니다. 기존 메뉴, 캐시, 계정 시스템과의 호환성을 유지하고 회귀 테스트를 반영하여, 각 저장소에서 권한 확인 방식을 단계적으로 전환할 수 있도록 구성했습니다.

사용 기술: Java, Spring Boot, JPA, MySQL, Bash, Hexagonal Architecture, Kafka, RBAC, OpenAPI

8. [경리회계] ERP 프로젝트 신규 개발

2023.03 - 2025.11 | 하이웍스개발팀

카드사별로 연동할 수 있는 인터페이스를 설계하고, 외부 스크래핑업체와 연동하였습니다.
경리회계에서 수집한 내역으로 경비를 청구하고, 하이웍스 전자결재에서 승인할 수 있도록 연동하였습니다.
실시간 수집과 일일 스크래핑을 함께 운영하고, 일일 작업에서는 최근 30일 내역을 다시 수집하여 신규 승인 내역과 기존 거래의 취소 상태를 반영하였습니다.

[법인카드 내역 수집]
외부 스크래핑 API에는 호출 제한이 있고, 같은 로그인 계정으로 동시에 요청하면 충돌하는 문제가 있었습니다.
같은 카드사 계정의 요청은 순차로 처리하고, 일일 수집은 계정 그룹 간에 워커 풀을 사용해 병렬로 처리했습니다.
처리량에 따라 대기 시간을 두고, 수집 중 예외로 실패한 건은 전체 수집이 끝난 뒤 순차적으로 재시도했습니다.
요청 접수와 데이터 수집을 분리하고 진행률과 수집 결과를 제공하여, 사용자가 수집 상태를 확인할 수 있도록 하였습니다.

[경비 청구와 전자결재 연동]
외부에서 수집한 원본 데이터와 사용자가 수정한 데이터를 분리했습니다.
반려, 취소, 재기안, 대리기안에 따른 상태 변화를 Kafka 이벤트로 분리하여 카드 내역과 경비 청구, 전자결재 결과의 상태를 동기화했습니다.

[데이터 연동과 이용료 정산]
고객에게 청구하는 데이터 연동 이용료를 일별로 집계하고, 카드/계좌/공동인증서의 연동 상태와 만기일을 관리했습니다.
이용료는 하이웍스 포인트로 청구하므로, 잔액 부족과 해지 예약 여부에 따라 연동을 유지할지 결정해야 했습니다.
정산 배치는 DB 연결 풀 크기에 맞춰 병렬 실행량을 제한하고, 강제 해지, 해지 예약 처리, 포인트 부족 재처리, 정상 연장 순서로 실행하도록 구성했습니다.
포인트 차감이 성공하면 만기일 변경을 확정하고, 실패하면 정상 연장을 롤백했습니다.

차감 실패 건은 포인트 부족 상태와 재처리 기한을 저장해 연장을 다시 시도하고, 기한까지 연장되지 않은 연동은 해제했습니다.

사용 기술: TypeScript, Node.js, Express, MySQL, Sequelize, CODEF, Kafka, Jest, Sentry

(주)게임스프링

2022.05 - 2023.02 (10개월) | 정규직 | 연구소 | 백엔드 개발

게임 운영 도구 및 사내 백오피스

2022.05 - 2023.02 | 백엔드 개발

게임 운영과 테스트에 필요한 내부 도구와 백오피스를 개발했습니다.

[실시간 게임 테스트 도구]
QA가 게임 데이터의 값을 바꿀 때마다 개발자에게 요청해야 했습니다.
Socket.IO를 이용해 QA가 게임 데이터와 화면을 실시간으로 직접 조작하며 테스트할 수 있는 도구를 개발했습니다.

[GraphQL 기반 운영 통계]
운영 지표를 확인하려면 매번 DB 조회가 필요했습니다.
GraphQL 기반으로 CCU, MAU, 매출 등 통계 데이터를 집계하고 조회하는 기능을 개발했습니다.
기획과 마케팅이 백오피스에서 필요한 지표를 직접 확인할 수 있도록 하였습니다.

[사내 업무 자동화]
주간 보고서를 작성할 때 일정이 누락되는 경우가 잦았습니다.
Google Calendar와 주간 보고서를 연결하여 일정 기록을 보고서에 반영하는 자동화를 구현했습니다.

사용 기술: TypeScript, Express, GraphQL, Socket.IO, Google Apps Script

학력

유한대학교

2016.02 - 2020.02 | 정보통신과 영상보안 전공 | 대학(2~3년제) | 전문학사 | 졸업

AI 활용 경험

GPT, Claude, Gemini 모델 관리 및 자동 검증 스킬 개발 경험

LLM API 활용한 AI채팅 플랫폼 스펙 통합 및 모델 전환 기능 개발

Milvus 벡터 DB와 BGE-M3 기반 장기 기억 메모리 시스템 구현 경험

스킬

AWS | MySQL | TypeScript | Node.js | JavaScript | Git | GitHub | Docker | HTML | SQL | Nest.js | GraphQL

수상/자격증/기타

유한대학교 창업동아리, 한이음 ICT 공모전
2016.07 | 기타

- 냉장고 반찬통의 위치와 유통기한을 관리하는 프로그램을 개발했습니다.
- 팀장으로 프로젝트 흐름과 역할 분담을 조율했습니다.

개인 서비스
기타

AI 심리 상담 서비스: 위로
AI에게 고민 상담을 진행하다, 컨텍스트가 가득 찰 때마다 불편함을 느꼈습니다.
온전히 이전 대화의 맥락을 유지할 수 없는 것을 해소하여 사용하고자 시작한 프로젝트입니다.

https://www.wirocare.com/

주식과 코인 정보 대시보드 핀텀
주식과 코인 시세, 관련 뉴스를 한곳에서 확인할 수 있는 대시보드를 직접 개발하고 운영합니다.
https://www.fin-term.com/

언어

한국어 | **유창함**

링크

- AI 심리 상담 서비스: 위로
  https://www.wirocare.com/

- 개인 깃허브
  https://github.com/fullth

- 개인 기술 블로그 - 개발자가 되고 싶은 개발자
  https://fullth.tistory.com/

- 주식과 코인 정보 대시보드 핀텀
  https://www.fin-term.com/