

제품 범위와 팀의 실행을 함께 관리하는 Product Manager

이세현

Product Manager

CONTACT

- Portfolio <https://portfolio.leeseh0806.com>
- E-mail leeseh0806@gmail.com
- GitHub <https://github.com/lee0806>
- Velog <https://velog.io/@ummm>



SUMMARY

3개 프로젝트에서 팀 리드를 맡아 요구사항 정의부터 배포까지 실행을 관리했습니다. 주간회의와 Linear 티켓으로 작업·의존성을 추적하고, 비용·보안·운영 제약에 따라 기능 범위와 기술 대안을 결정한 Product Manager 지원자입니다.

VALUES

근거를 제시하고 팀의 의사결정을 이끕니다.

CodeRun에서는 핵심 학습 흐름을 우선하기 위해 학습 자료 제외안을 제시하고, 멘토 피드백과 팀 회의로 파일 트리형 IDE를 단일 파일 구조로 축소했습니다. MOG에서는 토큰 비용으로 손익분기를 계산해 팀에 공유하고 종량제 도입을 합의했습니다.

팀의 작업을 명세와 티켓으로 연결합니다.

이전 프로젝트들에서 작업 추적의 어려움을 경험한 뒤 CodeRun과 MOG에 Linear를 도입했습니다. 주간회의·소회의와 프로젝트당 60~80개의 티켓으로 담당·마감·우선순위·의존 관계를 관리했습니다.

SKILLS

- Product Management
 - | 문제 정의·요구사항 문서화·대안 비교·기능 범위 조정
- Team Operations
 - | 주간회의·티켓·일정·의존성 관리·직군 간 조율
- Metrics & Validation
 - | Before/After 측정·비용 구조 분석·결과 검증
- Collaboration Tools
 - | Linear, Notion, Figma, GitHub, Postman, Swagger
- Technical Background
 - | Java, Spring Boot, React, TypeScript
 - | PostgreSQL, Redis, Docker, Jenkins, Nginx
 - | AWS EC2·S3, REST API, SSE, React Query
 - | Claude Code, Codex - 기술 검토·구현 범위 분석 보조

PROJECT 01

[구름 x 인프런] 자바 스프링 & 리액트 풀스택 개발자 성장 과정

CodeRun - 실시간 코드 실행 • 채점 및 라인 단위 피드백을 제공하는 웹 기반 IDE

설명	브라우저에서 코드 작성·실행·채점을 수행하고 강사가 라인별 피드백을 남기는 교육용 IDE로, 작성→실행→채점→피드백 흐름을 구현했습니다.
GitHub	https://github.com/deepdive-01/goorm-ide-back
기간	2026.05.19 ~ 2026.06.05
기술·도구	Java 21, Spring Boot 3.3, PostgreSQL, Redis, JPA, Spring Security, JWT, Judge0, AWS EC2·S3, Docker, Jenkins, Nginx
프로젝트 인원	6명 - BE 3, FE 3
역할	팀 리드·BE·Infra - 회의·티켓·기능 범위 관리
주요 의사결정 및 실행	한 달 일정에 맞춘 IDE 핵심 기능 범위 결정 문제 정의 파일 트리형 IDE와 학습 자료 업로드까지 구현하기에는 일정이 부족했고, 핵심인 코드 실행·채점·피드백 흐름이 지연될 위험이 있었음. 판단 및 실행 멘토 피드백을 바탕으로 파일 트리 구조에서 단일 파일 구조로 검토하고, 핵심 흐름과 거리가 먼 학습 자료 제외안을 제시했음. 두 안은 팀 회의에서 논의해 최종 범위로 확정했음. 성과 파일 트리·학습 자료를 제외한 범위를 기준으로 한 달 내 코드 작성→실행→채점→라인별 피드백 전 구간을 구현했음. API 명세와 MSW를 활용한 FE·BE 병렬 개발 문제 정의 FE가 실제 API 완성을 기다리면 화면 개발과 연동 일정이 순차적으로 밀릴 수 있었음. 판단 및 실행 BE 응답 구조를 먼저 합의하고 FE가 해당 명세로 MSW 응답을 구성해 실제 API 완성 전부터 개발하도록 작업 순서를 조정했음. 성과 실제 API 완성 전에 FE 화면 개발을 시작했고, 연동 단계에서 발견한 응답 차이를 명세와 구현에 반영했음. 회의와 Linear 티켓으로 6인 팀의 작업·의존성 관리 문제 정의 BE·FE 각 3명이 병렬로 작업해 진행 상황과 연동 의존성을 한눈에 추적하기 어려웠음. 판단 및 실행 매주 수요일 30~60분 회의와 필요시 소회의를 운영하고, 60~80개 티켓에 담당·마감·우선순위·의존 관계를 기록했음. 결과는 Notion 회의록에 정리했음. 성과 담당·마감·우선순위·의존 관계를 Linear에서 공유해 팀 전체가 작업 상태와 연동 순서를 같은 기준으로 확인했음.

PROJECT 02

[구름 x 인프런] 자바 스프링 & 리액트 풀스택 개발자 성장 과정

MOG - 멀티모달 AI 영수증 인식과 이동시간 기반 중간지점 추천 모임 관리 플랫폼

설명	약속 날짜 조율부터 중간지점 추천, 영수증 기반 정산, 모임 기록 요약까지 모임의 전 과정을 하나의 서비스로 제공하는 플랫폼입니다.
GitHub	https://github.com/goorm-mog/mog-backend
기간	2026.06.01 ~ 2026.07.06
기술·도구	Java 21, Spring Boot 3.5, PostgreSQL(RDS), Redis, Spring Security, JWT, OAuth 2.0, SSE, Gemini API, Kakao Mobility API, S3, Docker Compose, Cloudflare DDNS
프로젝트 인원	6명 - BE 3, FE 3
역할	팀 리드·BE·Infra - 요구사항·API 명세·티켓 관리
주요 의사결정 및 실행	초기 사용량에 맞춘 영수증 인식 비용 구조 결정 문제 정의 고정요금 OCR은 월 18,000원이 발생하고, 사용량이 불확실한 초기 서비스에서는 비용 효율을 판단하기 어려웠음. 판단 및 실행 입·출력 토큰 평균으로 Gemini 비용을 건당 약 0.8원으로 계산하고, 고정요금과 손익분기 약 22,500건을 비교해 팀에 공유했음. 성과 팀 합의로 종량제를 선택하고, 고정 JSON과 null 반환 규칙으로 응답 형식을 통제했음. API 명세를 먼저 확정해 FE·BE 병렬 개발 문제 정의 Notion의 초기 명세와 실제 API가 달라지며 FE Mock 응답과 BE 구현 사이에 차이가 발생했음. 판단 및 실행 정산·알림·모임 기록·요약 카드 명세를 작성하고, 개발 단계부터 Swagger를 단일 기준으로 사용했음. FE는 명세에 맞춰 MSW 응답을 구성했음. 성과 실제 API 완성 전 FE 개발을 시작하고, 연동 중 발견한 응답 차이를 명세와 구현에 함께 반영했음. 50명 규모 시나리오에서 조회 성능 한계 검증 문제 정의 모임 인원 증가 시 동작을 확인하기 위해 50명 규모 테스트 데이터로 요약 카드 조회를 측정한 결과, 51개 쿼리와 10초 이상의 지연이 확인됐음. 판단 및 실행 SQL 로그로 반복 조회 지점을 특정하고 @EntityGraph를 적용해 연관 데이터를 단일 쿼리로 조회하도록 변경했음. 성과 동일한 50명 테스트 조건에서 쿼리를 51개→1개, 연관 데이터 조회 시간을 10,280ms→229ms로 단축했음.

PROJECT 03

K-디지털 챌린지: NET 챌린지 캠프 시즌 12

HELIOS - CCTV 영상 데이터를 이용한 실시간 도로 노후화 탐지 시스템

설명	AI 기반 실시간 도로 파손 탐지 시스템으로 CCTV 영상 데이터를 분석해 파손 위험도를 평가하고 효율적인 유지 보수 체계를 지원하는 서비스입니다.
GitHub	https://github.com/Helios-CCTV/Helios-web
기간	2025.07 ~ 2025.11 (5개월)
기술·도구	React, TypeScript, Tailwind CSS, Zustand, React Query, Axios, Python, Ultralytics YOLOv8, OpenCV
프로젝트 인원	2명 - FE·AI 1, BE·DevOps 1
역할	팀 리드·FE·AI 담당 - 문서·일정·역할 분담 관리
주요 의사결정 및 실행	화면별 요청·상태 처리 기준 통합 문제 정의 동일 데이터가 반복 요청되고 로딩·오류·성공 상태 처리 방식이 화면마다 달라 개발 기준이 일관되지 않았음. 판단 및 실행 React Query와 staleTime으로 중복 refetch를 제한하고 요청 상태별 UI를 분기했음. 성과 중복 요청을 줄이고 요청 상태 처리 규칙을 통일해 이후 화면에도 동일한 기준을 적용했음. BE와 조회 범위를 합의해 지도 사용자의 초기 대기 시간 개선 문제 정의 전국 마커를 한 번에 렌더링해 초기 로딩에 약 2,300ms가 걸렸음. 판단 및 실행 BE에 뷰포트 좌표 범위 조회 API를 요청하고 연동 규격을 함께 조율했음. 프론트에서는 줌 레벨 8 미만 조회를 제한했음. 성과 초기 렌더링을 2,300ms에서 340ms로 85% 단축하고 지도 이동 시 끊김을 줄였음. 학습 데이터 기준을 수립하고 탐지 품질을 단계적으로 검증 문제 정의 원본 CCTV 이미지에는 비도로 영역이 포함되고 파손 유형별 데이터가 불균형해 학습에 바로 사용하기 어려웠음. 판단 및 실행 77,307장 중 약 5,500장을 전처리·13종 라벨링하여 YOLOv8n을 학습했음. 탐지 결과는 멀티모달 AI로 2차 검증했음. 성과 mAP@0.5를 58.8%→92%로 높였고, 탐지 결과 100장 수동 검수에서 2차 검증 결과의 90% 이상이 실제 파손 데이터와 일치했음.

OTHERS

프로젝트	CodeRun - 실시간 코드 실행 • 채점 및 라인 단위 피드백을 제공하는 웹 기반 IDE MOG - 멀티모달 AI 영수증 인식과 이동시간 기반 중간지점 추천 모임 관리 플랫폼 HELIOS - CCTV 영상 데이터를 이용한 실시간 도로 노후화 탐지 시스템
직무 교육	[구름 x 인프런] 자바 스프링 & 리액트 풀스택 개발자 성장 과정 2025.12.19 ~ 2026.07.13
학력	경기대학교 졸업예정 (3.65/4.5) 토목공학과 전공, 컴퓨터공학과 복수전공 2020 ~ 2026
자격증	SQLD (SQL Developer) 2026.03.27
수상 경력	2025 산학협력 캡스톤 디자인 경진대회: 심화캡스톤 디자인 동상 2025 산학협력 캡스톤 디자인 경진대회: 기초캡스톤 디자인 은상 한국정보기술학회 우수논문상 K-디지털 챌린지: NET 챌린지 캠프 시즌 12 국가보안기술연구소 소장상 [구름 x 인프런] 자바 스프링 & 리액트 풀스택 개발자 성장 과정 우수 수료생