

주다훤 (Dahwon Ju)

Technical PM

전화번호

010-6518-8253

이메일

hwondaa@gmail.com

포트폴리오

<https://www.hwonda.com/pm>

GitHub

<https://github.com/hwonda>

업무 경험

(주) 한국해양기상기술

Technical Project Manager & Frontend

2023.01 ~ 재직중 (3년 2개월~)

전사 프로세스 관리

프로젝트 관리도구 및 AI 에이전트 도입

역할

TPM

수행 기간

2024.03 ~ 2026.01 (1년 10개월)

Tech Stack

Jira, Figma, Storybook, Claude Code, Cursor, Github

• Situation

- 엑셀과 Slack 중심의 수동 관리로 인해 프로젝트 일정과 마일스톤을 실시간으로 추적하기 어려움
- 기획 · 디자인 · 개발 간 커뮤니케이션도 파편화되어 요구사항 누락과 납기 지연 리스크가 상시 존재
- 브랜치 전략, 커밋 규칙, 디자인 협업 방식 등 엔지니어링 표준이 부재로 인한 유지보수 효율성 저하

• Task

- 전사 차원의 협업 프로세스를 표준화 및 일정 가시성과 리스크 대응력 향상
- AI 도구를 포함한 실행 가능한 개발 체계를 정착

• Action

- Jira 도입:** Jira를 도입해 WBS 기반 일정 관리 체계를 설계하고, 태스크 단위로 마일스톤을 추적할 수 있는 운영 방식으로 전환
- 협업 도구 표준화:** Figma와 Storybook을 도입해 기획 · 디자인 · 개발 간 협업 체계 구축 및 UI/UX 관련 커뮤니케이션 오버헤드 해소
- 형상관리 기준 정립:** 1기능 1브랜치, Squash Merge, Commit Convention 등 도입해 일관된 형상관리 및 배포 운영 기준을 수립
- AI 에이전트 도입:** Cursor, Claude Code 등 AI 에이전트 활용 가이드를 문서화 해 반복 업무 자동화와 조직 내 AI 활용 확산을 주도

• Result

- 프로젝트 일정 가시성이 확보되며 데이터 기반 의사결정 문화가 정착
- 유지보수와 리팩토링 효율이 약 50~60% 향상
- 커뮤니케이션 손실이 줄어들어 납기 지연 리스크를 선제적으로 통제할 수 있는 구조 마련

스마트 양식장 2025

스마트 양식 연구 지원 통합 플랫폼 개발

Link | [소개](#)

역할

PM · UI/UX · Frontend

수행 기간

2025.06 ~ 2025.12 (6개월)

Tech Stack

React, TypeScript, TailwindCSS, Figma, Jira, Github

• Situation

- 양식장 데이터와 시뮬레이션 기능이 분산되어 있어 **연구진의 분석 및 의사결정에 과도한 시간 소요**
- 사내 리소스 부족**으로 일정 리스크가 존재
- Vue.js 2로 제작된 기존 양식장 서비스의 공식 지원이 종료돼 보안 취약점 이슈가 존재

• Task

- 산재한 연구 데이터를 단일 서비스에서 탐색·분석할 수 있는 통합 플랫폼 기획 및 개발
- 짧은 기간 내 기존 양식장 서비스 기능 마이그레이션 및 신규 기능 개발

• Action

- 서비스 기획, UI/UX 설계, 개발 **전 과정을 주도**하며 통합 플랫폼 구체화
- 연구 현장 방문 및 정기 커뮤니케이션**을 통해 고객 요구사항을 수집·구조화하고 기능 명세로 연결
- Jira 기반 WBS와 태스크 단위 운영 체계를 구축**해 협업 효율화 및 일정 리스크 사전 통제
- 기존 Vue로 제작된 35개 기능 React로 마이그레이션

• Result

- 연구 데이터 분석 및 의사결정 시간을 **40분에서 2분으로 단축**
- 복잡한 요구사항 및 제한된 리소스에도 불구하고 **데드라인 내 런칭** 달성

@koast/ui

사내 React + Tailwind UI 라이브러리

Link | [소개](#) [NPM](#) [Storybook](#)

역할

1인 개발(PM · Frontend)

수행 기간

2025.03 ~ 2025.06 (3개월)

Tech Stack

React, TypeScript, TailwindCSS, Vite, Storybook, Github Actions

• Situation

- 프론트엔드 작업이 특정 인력에게 집중
- 전체 프로젝트에서 **UI 구현 병목**이 발생으로 전체 일정 지연
- JSP, Vue 등 이질적 스택 혼재

• Task

- 프론트엔드 의존도 완화
- UI 개발 생산성 개선
- 기술 스택 표준화

• Action

- 공통 프레임워크 도입**: React, TailwindCSS 기반 표준 스택 제안 및 도입
- 도메인 특화 디자인 시스템 구축**: 해양·기상 데이터 시각화 요소를 포함한 공통 컴포넌트를 TailwindCSS 기반으로 자산화해 전사 UI 가이드라인을 정립
- Storybook 기반 컴포넌트 문서화**: Storybook과 배포 체계를 연계해 백엔드 개발자도 독립적으로 UI를 구현·검증할 수 있는 개발 환경을 구축

• Result

- 프론트엔드 의존도를 낮춰 **개발 병목을 해소**
- 복잡한 특정 컴포넌트(Time Slider) 구현 시간을 **8시간에서 10분 이내로 단축**

기상청 UI/UX 개선

기상청 대국민 서비스 UI/UX 현대화 및 시스템 효율화

Link | [소개](#) [프로젝트](#)

역할

PM · Frontend Lead

수행 기간

2025.06 ~ 2025.08 (3개월)

Tech Stack

Vue, TypeScript, Axios, Figma, Jira, Gitlab

• Situation

- 대국민 서비스의 **UI 노후화로 사용자 불만이 누적**되고 있었고, 2,000라인 이상의 레거시 구조로 인해 **단순 수정에도 과도한 시간이 소요**

• Task

- 서비스 중단 없이 UI/UX를 개선하고, 고객사 요구사항을 반영하면서도 유지보수와 기능 개선이 가능한 구조로 시스템을 재정비

• Action

- 이해관계자 요구사항 정렬:** 기상청 실무진과 직접 커뮤니케이션하며 핵심 Pain Point와 개선 우선순위를 정의
- 단계적 개선 로드맵 수립:** 서비스 연속성을 유지할 수 있도록 점진적 배포 기반의 UI/UX 개선 로드맵을 설계
- 모듈화 기반 리팩토링:** 대형 메서드와 레거시 로직을 기능별 컴포넌트로 분리해 재사용 가능하고 유지보수 가능한 구조로 리팩토링

• Result

- 개선 후 사용자 설문(672명) 기준 **UI 만족도를 4점에서 8점으로 높이고 평균 체류 시간을 2배 향상**
- 신규 메뉴 대응 및 유지보수 시간을 60분에서 5분 수준으로 단축해 **운영 효율을 약 90% 개선**

AI Code Reviewer Bot

오픈 LLM 기반 사내 코드 리뷰 시스템 구축

Link | [소개](#)

역할

PM · AI Ops

수행 기간

2025.04 ~ 2025.05 (1개월)

Tech Stack

Python, Systemd, Gitlab, Ollama, Pipeline

• Situation

- 보안 정책상 외부 클라우드 AI를 사용할 수 없는 **온프레미스 환경**
- 단순 오탈자나 컨벤션 위반까지 시니어 엔지니어가 수동으로 검수해 **코드 리뷰 병목 발생**

• Task

- 보안 제약을 준수하면서도 반복적인 1차 코드 리뷰를 자동화해 리뷰 리소스 부담을 줄이고, 지속 가능한 품질 관리 체계를 구축

• Action

- 온프레미스 환경에 최적화된 AI 리뷰 아키텍처 수립:** 외부망 통신이 차단된 내부 NAS 환경에서 Ollama 기반 로컬 LLM 구조를 설계해 보안 제약 내에서 AI 리뷰 도입 기반을 마련함
- 코드 리뷰 자동화 워크플로우 내재화:** Systemd 기반 백그라운드 워커와 GitLab 파이프라인을 연동해 MR 발생 시 자동으로 1차 리뷰가 수행되는 체계를 구축함
- 지속 가능한 품질 운영체계 정립:** 수동 개입 없이 운영 가능한 자동화 구조를 설계해 반복 리뷰 업무를 줄이고 코드 품질 관리의 일관성을 높임

• Result

- 기상청 프로젝트의 **MR당 코드 리뷰 시간을 30분에서 15분으로 단축**해 리뷰 효율을 50% 높임
- 수동 확인 중심의 리뷰 프로세스를 자동화해 **지속 가능한 코드 품질 관리 체계를 확보**

기상청 서비스

기상청 대국민 서비스 및 기상청 내 솔루션 시스템

Link | [소개\(2023\)](#) [소개\(2024\)](#)

역할

PM · Frontend

수행 기간

2023.04 ~ 2025.03 (2년)

Tech Stack

Vue, Vuex, TypeScript, Axios, OpenLayers

• Situation

- 프론트엔드 외주 운영으로 인해 **지난 2년간 고객사 요구사항이 누적**
- 노후화된 코드 구조로 인해 **신규 기능 확장성과 시스템 안정성이 저하됨**

• Task

- 누적된 기술 부채를 해소 및 기존 운영 안정성을 유지
- 고객사 요구사항과 신규 기능을 지속적으로 반영할 수 있는 구조로 개선

• Action

- **기술 부채 구조 진단**: 기존 로직 전반을 분석해 누적된 기술 부채와 확장성 저해 요인을 식별하고, 우선순위 기반의 개선 과제를 정의
- **정적 타이핑 기반 안정성 강화**: TypeScript를 전면 도입해 런타임 에러를 사전에 방지할 수 있는 정적 타입 체계를 구축
- **운영·확장을 고려한 리팩토링 추진**: 유지보수성과 신규 기능 대응력을 높일 수 있도록 리팩토링 로드맵을 수립하고 단계적으로 적용

• Result

- **2년간 누적된 기술 부채를 해소**하고 타입 안정성을 강화해 서비스 운영 신뢰도와 유지보수성을 높임
- 기존 운영 안정성을 유지하면서도 고객사 요구사항과 신규 기능을 **보다 안정적으로 반영할 수 있는 개발 기반을 마련**

수색구조 시스템 UI/UX 개선

해양 사고 시각화 플랫폼 UI/UX 개선

Link | [소개](#)

역할

PM · UI/UX · Frontend

수행 기간

2024.03 ~ 2024.05 (2개월)

Tech Stack

HTML, JavaScript, CSS, Figma, Git

• Situation

- 20여 개 이상의 색상이 혼재된 화면 구성으로 긴급 상황에서 **핵심 데이터의 시인성이 떨어졌고**, 비대한 단일 JavaScript 파일 구조로 인해 **기능 수정과 유지보수 생산성이 낮음**

• Task

- 긴급 상황에서 핵심 정보를 빠르게 식별할 수 있도록 시각화 체계를 재정비하는 동시에, 레거시 프론트엔드 구조를 개선해 유지보수와 신규 기능 대응이 가능한 운영 기반을 마련

• Action

- **정보 우선순위 기반 시각화 체계 재설계**: 무분별하게 사용되던 20여 개의 색상 테마를 데이터 중요도와 사용 맥락에 따라 7개의 핵심 Color Set으로 재정의해 정보 인지성과 시인성을 높임
- **레거시 프론트엔드 구조 모듈화**: 3,000라인 이상의 단일 JavaScript 파일을 기능 단위로 분리해 코드 가독성과 유지보수성을 높이고, 기능 개선과 버그 대응이 가능한 구조로 재구성

• Result

- 긴급 상황에서 데이터 우선순위가 더 명확하게 전달되도록 **UI/UX를 개선해 정보 시인성을 높임**
- 신규 기능 추가 및 버그 수정 시간을 **4시간에서 30분 내외로 단축**해 유지보수 생산성을 **약 85% 향상**
- 레거시 구조를 모듈화해 **안정적인 운영과 후속 개선이 가능한 프론트엔드 기반을 마련**

개인 프로젝트

디키 - Diki

AI 초심자를 위한 데이터 용어사전

Link | [소개](#) [프로젝트](#) [Github](#)

역할

UI/UX · Frontend

수행 기간

2024.10 ~ 2025.12 (1년 2개월)

Tech Stack

Next, TypeScript, TailwindCSS, Firebase, Redux, Vercel

• Situation

- AI-데이터 분석 분야의 전문 용어가 파편화되어 있어 초심자의 정보 탐색 효율이 낮음
- 실시간 데이터 요청 구조는 Firestore·Vercel의 쿼리 제한과 운영 비용 부담을 유발할 수 있어, 검색 성능과 비용 효율을 함께 고려한 서비스 설계가 필요

• Task

- 초심자 친화적인 용어사전을 구축하는 동시에, 운영 비용을 최소화하면서도 빠르고 정확한 검색 경험을 제공할 수 있는 구조 마련

• Action

- 비용 효율 중심 아키텍처 설계: Next.js SSG를 적용해 용어 페이지를 사전 렌더링하고 API 호출 비용 절감 및 페이지 로딩 속도 개선
- 검색 정확도 중심 탐색 경험 설계: 한/영 제목, 태그 등 필드별 가중치를 반영하고, 정확한 구문 매칭부터 접두사/접미사 검색, Fuzzy 검색(오타 1자 허용)을 지원하는 등 다각도 쿼리 엔진을 구축
- 운영 자동화 파이프라인 구축: GitHub Actions와 Vercel을 연동해 데이터 수집부터 정기 빌드·배포까지 자동화하고 수동 운영 공수 최소화

• Result

- SSG 기반 구조를 통해 서버 비용 제로에 가까운 지속 가능한 운영 환경 확보
- 구조화된 데이터와 메타 태그 최적화를 통해 검색 엔진 노출 가능성과 정보 전달 효율 향상
- 자동화된 배포 파이프라인을 구축해 수동 개입 없이 콘텐츠를 지속 운영·관리할 수 있는 기반 확보

지지집 - Zizip

부동산 위치 정보 시각화 서비스

Link | [소개](#) [프로젝트](#) [Github](#)

역할

PM · UI/UX · Full-stack

수행 기간

2025.10 ~ 2025.12 (2개월)

Tech Stack

Next, TypeScript, TailwindCSS, Zustand, React-Query, Vercel

• Situation

- LH 등 공공임대 공고 데이터가 엑셀 형태로 제공되어 사용자가 주택 위치를 확인하려면 주소를 개별적으로 다시 검색해야 하는 불편이 있음
- 공고마다 컬럼명이 달라 비정형 데이터를 일반 사용자가 직접 통합·분석하거나 시각화하기 어려운 구조

• Task

- 비정형 공고 데이터를 표준화
- 지도 중심 인터페이스로 전환해 주택 탐색 경험 개선

• Action

- 데이터 표준화 인터페이스 설계: 최근 3년간의 공고 데이터를 분석해 주요 변수를 추출하고, 주택명·주소 등 필수 항목 기준의 매핑 로직을 구축해 비정형 데이터를 단일 규격으로 변환함
- 핵심 기능 기반 MVP 기획: '엑셀 업로드 시 지도 즉시 표출'이라는 핵심 사용자 가치에 집중해 공고 목록 전체를 한눈에 조망할 수 있는 지도 인터페이스를 설계·개발함
- 성능 최적화 및 안정성 확보: 대량 마커 렌더링 시 발생하는 UI 프리징을 줄이기 위해 Web Worker를 도입하고, 줌 레벨 기반 클러스터링 전략을 적용함

• Result

- 별도 마케팅 없이도 부동산 커뮤니티에서 평균 30개 이상의 댓글과 70개 이상의 좋아요를 얻으며 제품 반응을 확인
- 산재된 텍스트 기반 공고 데이터를 시각화 자산으로 전환해 사용자의 주택 탐색 비용과 의사결정 부담을 낮추고, 최대 활성 사용자 578명을 달성

기타

블로그

프론트엔드 기술 블로그

[링크](#) | [소개](#) [서비스](#) [Github](#)

개요

MDX 기반 커스텀 블로그 제작

기술 스택

Next, TypeScript, TailwindCSS, Vercel

개발 기간

2024.07 ~ 2025.01 (6개월)

• MDX 기반 커스텀 콘텐츠 관리 시스템 구축

- gray-matter와 next-mdx-remote를 활용해 마크다운 내 JSX 컴포넌트 삽입이 가능한 유연한 작성 환경 구현
- rehype-pretty-code, rehype-slug 등 플러그인 도입으로 코드 하이라이팅 및 문서 구조화(Heading ID) 자동화

• 검색 엔진 최적화 및 배포 자동화

- 빌드 시점에 Sitemap, RSS/Atom/JSON Feed를 자동 생성하는 커스텀 스크립트를 구현하여 검색 엔진 노출 및 배포 파이프라인 강화
- Next Metadata API를 활용한 동적 OpenGraph 및 Twitter Card 설정으로 소셜 공유 시 시인성 증대

React-multi-email

OpenSource Contribute

[링크](#) | [소개](#) [NPM](#) [Github](#)

개요

React 기반 다중 이메일 작성 라이브러리 기여

github stars: 300+

weekly downloads: 100,000+

기술 스택

Github, React, Jest

개발 기간

2023.07 ~ 2024.01 (6개월)

• 테스트 자동화 및 코드 안정성 강화

- Jest를 활용하여 주요 로직에 대한 검증 코드를 작성하여 잠재적 런타임 에러 사전 차단

• 핵심 로직 버그 수정 및 코드 리뷰

- 실제 사용자들이 겪는 버그 이슈를 분석하여 수정 로직을 제안
- PR 및 Issue Review

• 개발자 경험 개선을 위한 문서화

- Docusaurus/MDX 기반의 공식 문서 시스템을 구축하여 사용자가 라이브러리를 더 쉽고 빠르게 도입할 수 있는 가이드 제공

스킬

협업

Jira, Confluence, Slack, Notion, Github/Gitlab

디자인

Figma, Storybook

AI

Claude Code, Cursor, ChatGPT, Ollama, NotebookLM

프론트엔드

React, Next, Vue, TypeScript, TailwindCSS, SCSS, Git, Firebase, Swagger, Docker