

Portfolio

Contact

010-6366-9357

jxeong@gmail.com

눈에 띄게 성장하고
싶은
개발자 김소정입니다.

CONTENTS

01 About Me

02 Projects

02-1 TalenTree: 우리 학교 학생들을 위한 재능 거래 플랫폼

02-2 Inside Mirror: 거울 앞, 당신의 하루가 스마트해진다

02-3 Find-My: UWB 보드를 사용한 실내 측위 프로그램

02-4 졸업전시 웹사이트: 덕성여대 소프트웨어전공 졸업전시: 공명(共鳴)

About Me

01

포기하지 않는 개발자

1년 9개월간 학부 연구생으로 다양한 프로젝트에 참여하였고, 그 과정에서 쉽게 포기하지 않는 마음가짐이 얼마나 중요한지 배웠습니다. 문제를 끝까지 해결하려는 끈기와 스스로 방향을 찾아가는 자세를 몸에 익힐 수 있었고, 이 경험은 새로운 환경이나 복잡한 문제를 마주했을 때도 주도적으로 해결해 나갈 수 있는 단단한 개발자로 성장하는 밑바탕이 되었습니다.

02

책임감과 리더십을 가진 개발자

부학생회장으로서 학생회와 함께 다양한 행사를 기획하고 운영했습니다. 특히 이번 졸업전시회는 기획, 운영, 웹사이트 개발까지 전 과정을 직접 주도하며 리더십과 실행력을 키울 수 있는 값진 기회였습니다. 여러 사람의 의견을 조율하고 일정을 관리하는 과정에서 다른 협업 환경에서도 주도적으로 역할을 수행할 수 있는 기반을 쌓을 수 있었습니다.

03

의사소통 능력이 좋은 개발자

1년간 동기들과 졸업 프로젝트를 진행하며 다양한 의견 충돌과 갈등을 겪었습니다. 그 과정에서 감정을 앞세우기보다 서로의 입장을 조율하고 상황을 해결해 나가며 자연스럽게 의사소통 능력과 갈등 대처 능력을 키웠습니다. 덕분에 협업이 중요한 개발 환경에서도 팀 분위기를 조율하고, 유연하게 소통하며 문제를 해결할 수 있는 강점을 가질 수 있게 되었습니다.



About Me



김소정

2003.05.09

010-6366-9357

jxeong@gmail.com



Github

<https://github.com/jxeong>

Education

2022.03
~ 재학 중

- 덕성여자대학교 소프트웨어전공
전체 4.25/4.5 · 전공 4.32/4.5
2026.02 졸업 예정

Activities

2023.03
~ 진행 중

- 덕성여자대학교 소프트웨어전공
부학생회장

2023.09
~ 진행 중

- 소프트웨어전공 이형규 교수
학부연구생 활동

Award

2024.12

- 교내 창업경진대회 동상

2025.05

- 한국산업정보학회 우수논문상

Certificate – 어학, 자격증

2025.05

- TOEIC 910

2025.06

- SQLD 개발자 자격증

Projects

2024.09
~ 2024.10

- 학교 학생들을 위한 재능 거래 플랫폼
: **TalenTree**

2024.10
~ 2025.06

- 거울 앞, 당신의 하루가 스마트해진다
: **Inside Mirror**

2024.12
~ 2025.06

- UWB를 사용한 실내 측위 프로그램
: **Find-My**

2025.06
~ 진행 중

- 덕성여대 소프트웨어전공
졸업전시회 웹사이트 개발

Tool & Skills

Language Python, C/C++, JAVA, JavaScript

Tool React, Spring Boot, JSP, FastAPI

DB MySQL

ETC Git, Figma, Notion

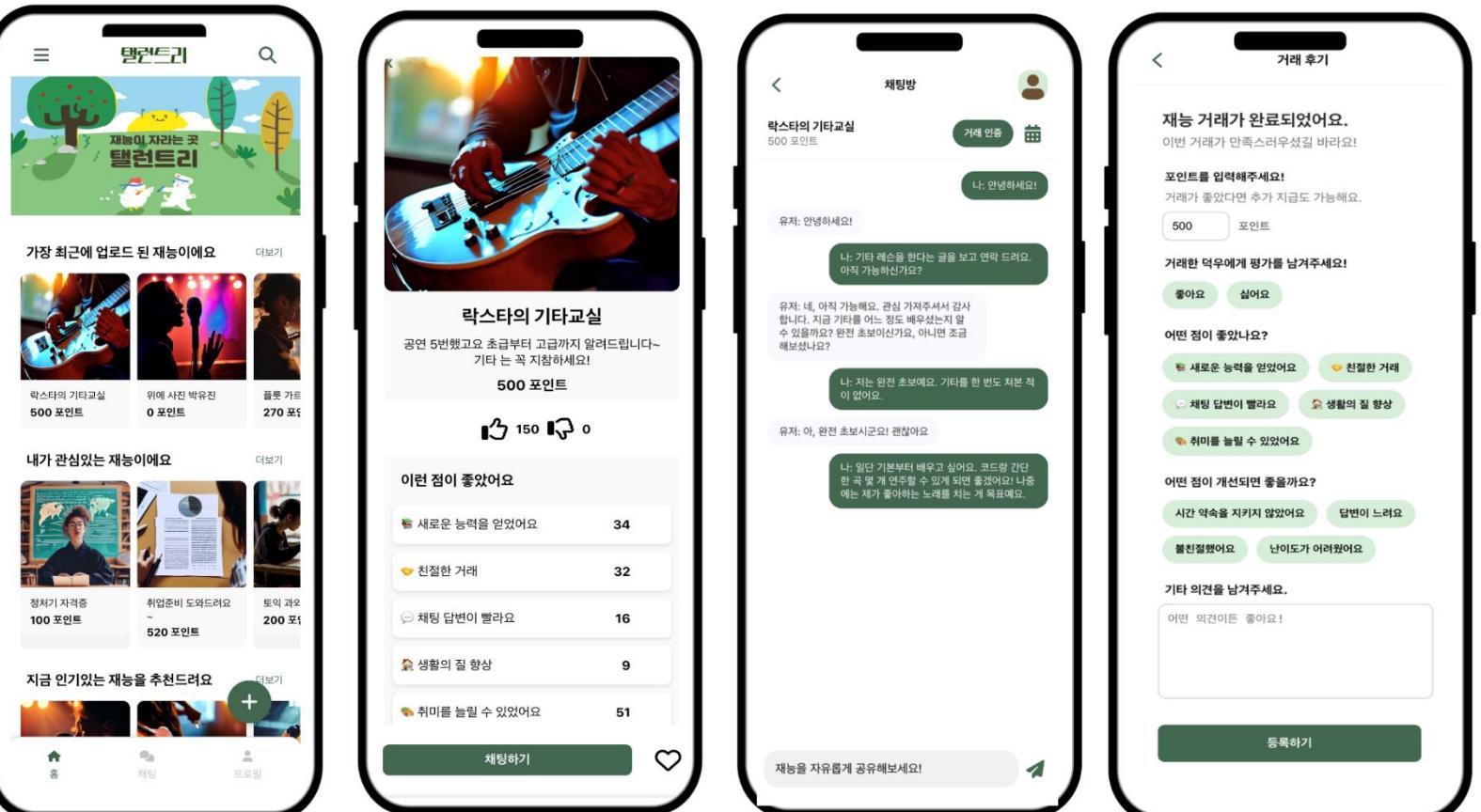
TalenTree Inside Mirror Find-My

우리 학교 학생들을 위한 재능 거래 플랫폼

TalenTree

팀프로젝트 웹 개발

덕성여대 학생들이 각자의 재능을 자유롭게 거래할 수 있는 React 기반 웹 플랫폼을 개발했습니다. 실제 금전 거래 대신 웹 내 포인트 시스템을 도입하여 학우라면 누구나 부담 없이 참여할 수 있도록 구현했습니다.



기간/인원 2024.09 ~ 2024.10 - 2개월 / 4명

기술 스택

- 프론트엔드 React JavaScript
- 백엔드 / DB SpringBoot MySQL
- 기타 툴 Git Notion Figma

담당 기능

- 로그인, 채팅, 거래 인증, 거래 후기 페이지 UI 개발
- 게시물 카테고리별 조회 기능 구현
- 서버 데이터베이스와 프론트 화면 연결
(프론트엔드-백엔드 연동)

배운 점

- React를 활용하여 프론트엔드 UI/UX를 개발하며 반응형 동적 UI와 사용자 경험을 개선하는 방법을 익혔습니다.
- React에서 Axios를 사용해 Spring Boot 백엔드와 API 통신을 구현하는 방법을 배우고, 조회/저장 등 데이터를 효율적으로 관리하는 방법을 알게 되었습니다.



TalenTree

트러블슈팅 1

1. 초기 개발 시 DB 설계 단계 누락

웹 프로젝트가 처음이라 명확한 DB 설계 없이 바로 개발을 시작했고, 기능이 추가될 때마다 테이블과 컬럼을 반복적으로 수정해야 했습니다. 그때마다 백엔드(Spring Boot) 코드도 함께 수정해야 해 비효율적이었습니다.

시도한 방법

중간에 전체 데이터 흐름을 다시 정리하고, 재정비 시간을 가졌습니다. 필요한 테이블 구조와 컬럼을 명확히 정의한 뒤 ERD를 기반으로 이후에는 새로운 기능도 이 구조에 맞춰 추가하며 안정적으로 개발할 수 있었습니다.

알게 된 점

개발 초기 단계에서 DB 설계의 중요성을 체감했고, 명확한 데이터 흐름과 구조를 먼저 정의하면 전체 개발 효율과 협업 속도가 크게 향상된다는 점을 배웠습니다.

트러블슈팅 2

2. 프론트엔드에서의 필터링 기능

게시물 목록에서 특정 카테고리만 필터링해서 보여주려 했지만, 프론트엔드에서 필터링을 처리하다 보니 성능 저하와 데이터 일관성 문제가 발생했습니다.

시도한 방법

프론트엔드 필터링 대신, 백엔드에서 카테고리 값을 기준으로 쿼리를 분기하여 필요한 데이터만 전달하도록 개선했습니다. React에서는 Axios를 통해 값을 넘기고, 선택한 카테고리에 따라 데이터가 갱신되도록 구현했습니다.

알게 된 점

데이터 필터링은 백엔드에서 처리해야 성능과 일관성을 유지할 수 있다는 것을 배웠고, 프론트와 백엔드 간의 역할 분담에 대한 이해가 한층 깊어졌습니다.

The screenshot shows a web-based user interface for the Inside Mirror project. At the top, there are three tabs: 'TalenTree' (selected), 'Inside Mirror', and 'Find-My'. Below the tabs, a large title '거울 앞, 당신의 하루가 스마트해진다' (Your day in front of the mirror becomes smart) is displayed. The main heading 'Inside Mirror' is prominently shown in large blue letters. To the right of the heading are two blue rounded rectangles containing the text '팀프로젝트' and 'IoT'. The central content area contains a detailed description of the project, mentioning facial recognition login, voice control, and AI integration. Below the text are four images: a smartphone showing the app's splash screen, a second smartphone displaying a calendar and note-taking interface, a tablet showing a media player and weather forecast, and a small rectangular device representing a smart mirror.

기간/인원	2024.10 ~ 2025.06 - 8개월 / 5명
기술 스택	<ul style="list-style-type: none">▪ 프론트엔드 React JavaScript▪ 백엔드 SpringBoot MySQL AWS FastAPI▪ 기타 툴 Git Notion WebSocket
담당 기능	<ul style="list-style-type: none">▪ FaceNet 모델 기반의 사용자 식별 FastAPI 구현▪ 로컬에서 처리한 시선 추적 좌표를 WebSocket을 통해 서버로 전달하는 UI 상호작용 기능 구현▪ 스마트미러와 연동되는 모바일 웹 프론트엔드 디자인
배운 점	<ul style="list-style-type: none">▪ FaceNet 기반 얼굴 인식 코드를 FastAPI 형태로 변환하며, Python 모델을 API 서버로 분리해 활용하는 방법을 익혔습니다.▪ 로컬에서 추론한 시선 좌표를 WebSocket으로 서버에 전송하고, 이를 프론트엔드에서 반영하는 전체 흐름을 이해하였습니다.▪ 모바일 웹에서도 얼굴 인식 기능을 동일하게 적용하여, 기기와 상관없이 개인화된 사용자 경험을 제공할 수 있음을 배웠습니다.

Inside Mirror

트러블슈팅 1

1. WebSocket handshake 문제

시선 추적 모델은 로컬에서 구동되었고, 추론된 좌표 데이터를 실시간으로 서버에 전달하기 위해 WebSocket을 사용했으나, 초기 핸드셰이크 단계에서 연결 오류가 발생해 서버와의 통신이 이루어지지 않았습니다.

시도한 방법

AWS 서버 환경에서 CORS를 확인하고, WebSocket 연결 시 사용하는 포트가 열려 있는지 점검하였습니다. 또한 클라이언트 측 WebSocket 연결 시점과 URL 프로토콜(ws/wss), 포트 일치를 확인하며 문제를 해결하였습니다.

알게 된 점

실시간 통신 환경에서는 단순한 연결 이상의 요소(CORS, 포트, 핸드셰이크 절차 등)를 종합적으로 고려해야 하며, 서버-클라이언트 간 명확한 프로토콜 정립이 중요하다는 점을 배웠습니다.

트러블슈팅 2

2. 커뮤니케이션 부족으로 인한 일정 딜레이

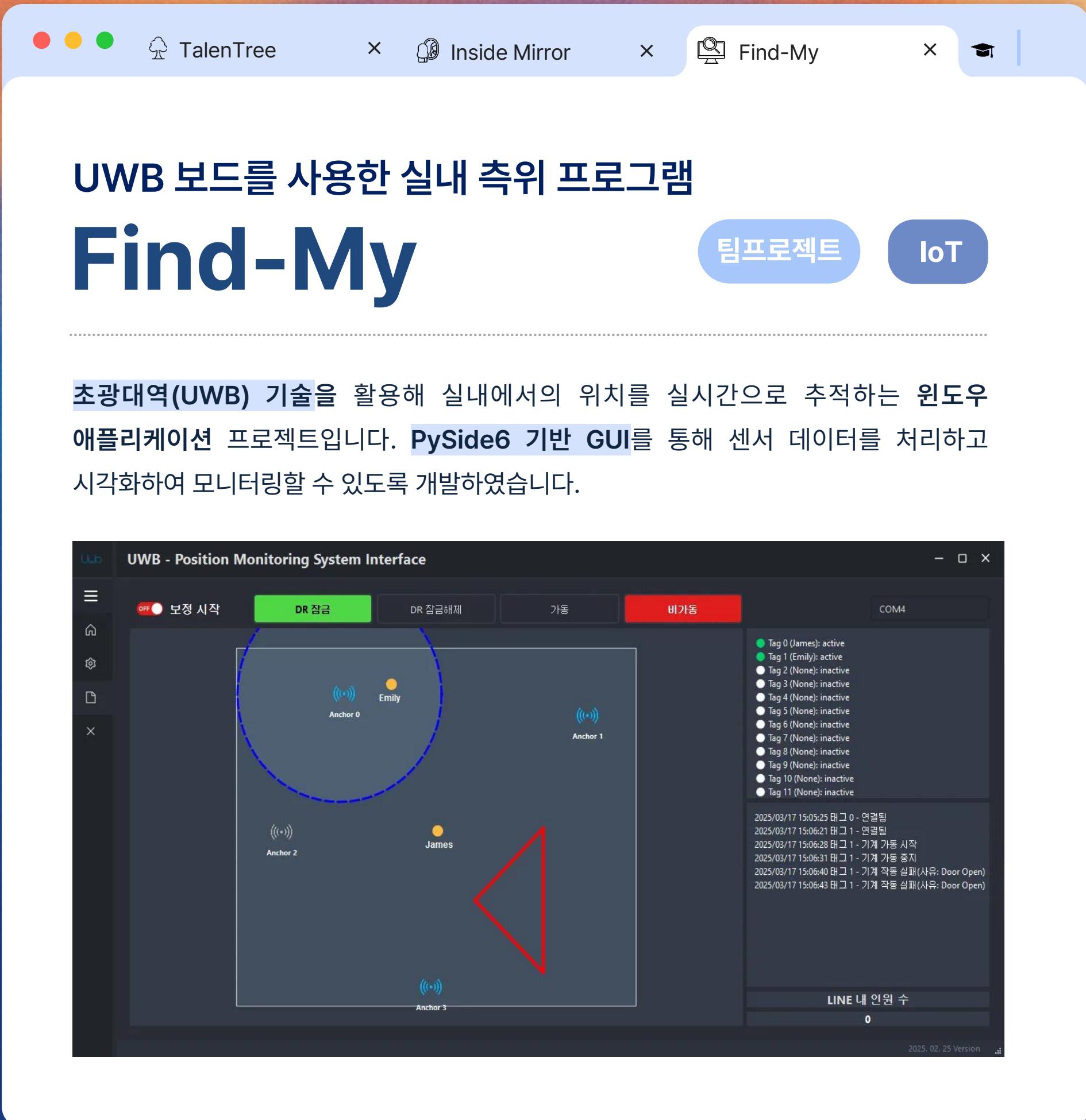
팀원 5명이 각각 기능을 개발했지만 서로의 작업 방식과 구조를 잘 공유하지 않아, 기능 통합 시 연동 방식이 맞지 않거나 작업 방식이 겹치면서 일정이 지연되고 충돌이 잦았습니다.

시도한 방법

각자의 구현 방식과 API 흐름을 정리하고, 코드 리뷰 및 주기적인 회의를 통해 구조를 통일했습니다. 연동 방식은 역할 분담을 명확히 하고 계획을 기반으로 재정리하여 각 파트가 원활하게 연결되도록 조정했습니다.

알게 된 점

기능 개발도 중요하지만, 팀 프로젝트에서는 공유와 소통이 그 이상으로 중요하다는 것을 느꼈습니다. 초기에 전체 구조와 연동 방식을 충분히 공유하는 것이 협업 효율성과 일정 관리의 핵심이라는 점을 깨달았습니다.



기간/인원 2024.12 ~ 2025.06 - 6개월 / 2명

기술 스택

- 사용 언어 Python C++
- 사용 프로그램 PySide6 Qt Designer
- 기타 툴 Git Notion Figma

담당 기능

- PySide6로 실시간 실내 측위 GUI 인터페이스 제작
- UWB, RFID 시리얼 데이터를 파이썬 수신 및 처리
- 실제 실내 공간에서 시스템 테스트 및 정확도 검증

배운 점

- PySide6로 UI 디자인과 기능 연동을 진행하며 앱 개발 전반의 흐름을 이해했습니다. 또한 UWB, RFID 센서에서 시리얼 데이터를 Python에서 어떻게 수신하고 처리할 수 있는지 알게 되었습니다.
- 넓은 실내 공간에서 실제로 시스템을 테스트하면서 센서 데이터에 다양한 오차가 발생할 수 있음을 경험했고, 정확도를 높이기 위한 피드백 공유와 개선 작업이 얼마나 중요한지 체감할 수 있었습니다.

Find-My

트러블슈팅 1

1. UWB와 RFID 데이터 동시 처리 문제

UWB 기반 거리 데이터와 ATmega128에서 전달되는 RFID 데이터를 하나의 코드에서 동시에 처리해야 했고, 데이터 흐름과 조건 분기가 얹히면서 코드가 복잡하고 비효율적으로 작동하는 문제가 발생했습니다.

시도한 방법

UWB와 RFID 데이터를 각각의 클래스로 분리하고, 메인 로직에서 필요한 정보만 받아 사용할 수 있도록 구조를 개선했습니다. 센서별로 데이터를 처리하는 쓰레드나 이벤트 흐름을 구분하여 충돌 없이 안정적으로 작동하도록 설계했습니다.

알게 된 점

다양한 하드웨어 장치에서 실시간으로 데이터를 수신·처리할 때는 각 장치의 흐름을 명확히 분리하고, 전체 시스템 구조를 먼저 설계하는 것이 안정성과 효율성 확보에 매우 중요하다는 것을 배웠습니다.

트러블슈팅 2

2. PySide6로 GUI를 구현하며 코드 구조가 복잡해진 문제

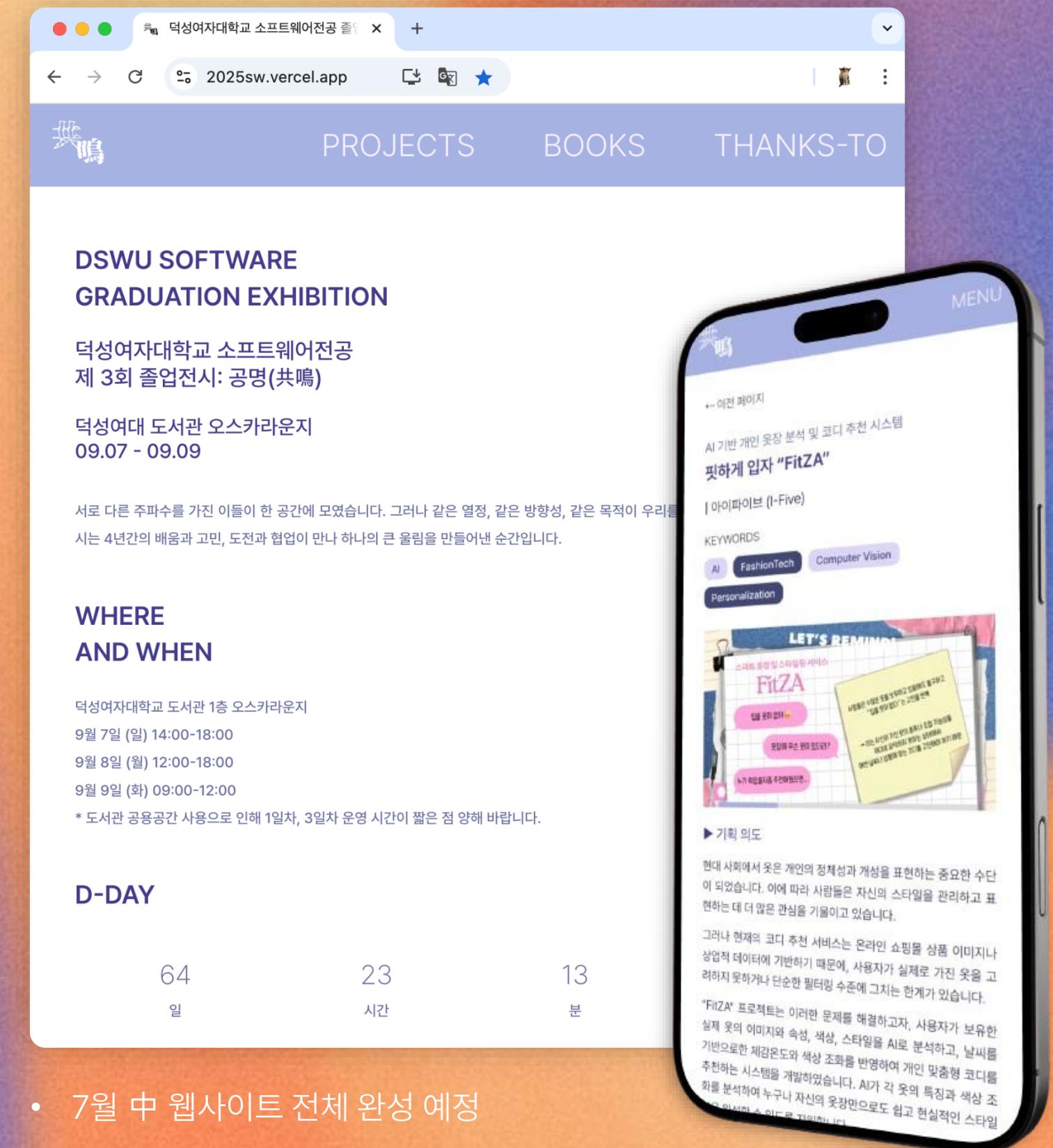
실시간 위치 계산, 시각화, 센서 통신, 이미지 처리 등 다양한 기능을 한 화면에 통합하다 보니 코드가 지나치게 길고 복잡해졌으며, 유지 보수와 디버깅이 어려워졌습니다.

시도한 방법

기능별 클래스로 구조를 재정리하고, 시각화, 센서 처리, 이벤트 관리 등을 각각의 모듈로 분리하여 코드의 가독성과 확장성을 높였습니다. 또한 상태 관리와 UI 업데이트를 클래스 내부 메서드로 일관되게 관리하도록 개선했습니다.

알게 된 점

실시간 동작이 많은 GUI 시스템에서는 클래스를 활용한 기능 분리와 메서드 정리가 유지 보수성과 기능 확장에 직접적인 영향을 준다는 점을 체감했습니다.





졸업전시 웹사이트 트러블슈팅

발생한 문제

- 졸업전시 참여 팀마다 소개 항목과 작성한 내용이 모두 달라, 각각 별도의 페이지를 만들기엔 유지 보수와 작업 효율 측면에서 비효율적인 문제가 있었습니다.
- 로컬 환경에서 카카오톡 공유 기능이 작동하지 않아 원인을 파악하던 중, API가 공인 IP 또는 등록된 도메인에서만 정상적으로 동작한다는 제약을 확인했습니다.

시도한 방법

- JavaScript로 템플릿 구조를 통일하고, 각 팀의 데이터에서 작성된 항목만 조건부 렌더링 되도록 처리하여 하나의 컴포넌트로 모든 팀 페이지를 동적으로 구성했습니다.
- Vercel을 사용해 프론트엔드 페이지를 배포한 뒤, 해당 URL을 카카오 개발자 콘솔에 등록하여 공유 기능을 다시 테스트했습니다.

알게 된 점

작은 기능 하나를 구현할 때도 환경과 구조를 고려한 설계가 중요하며, 외부 API나 다양한 데이터 구조에 유연하게 대응하려면 기술적인 이해뿐 아니라 실제 서비스 환경에서의 테스트와 경험이 함께 필요하다는 것을 배웠습니다.

읽어주셔서 감사합니다.

Portfolio

Contact

김소정

010-6366-9357

jxeong@gmail.com