

DESIGNER & DEVELOPER

---



# Portfolio

Frontend Developer

---

JINA JEONG



## 정진아

디자인하는  
프론트엔드 개발자

☎ 010.5720.5770

✉ jina092400@gmail.com

🐱 [github.com/jina0924](https://github.com/jina0924)

## 자기소개

### 기술 역량

🌸 React



Redux를 사용하여 데이터를 저장하고 사용할 수 있음

🟢 Vue



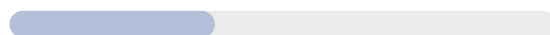
라이프사이클 혹은 이해하고 활용할 수 있음

🟡 Javascript



바닐라 자바스크립트를 이해할 수 있음

🐍 Python



알고리즘 문제를 풀 수 있음

### 프로젝트

#### ■ BOM

메디컬 웨어러블 디바이스를 통한 건강 정보 수집  
BMS를 통해 배터리 정보 수집 및 사고 예방  
웹, 모바일 모니터링 서비스

★ 삼성 SDI 기업 연계 프로젝트

★ 삼성 청년 SW 아카데미 최우수상

#### ■ FindIt!

Object Detection과 특징점 검출을 활용한  
온오프라인 연계 보물찾기

#### ■ PLANTinum

IoT 기반 식물 자동 관리 서비스 및 식물 거래 플랫폼

★ 삼성 청년 SW 아카데미 우수상

### 협업 방법



GitLab



Jira



Mattermost



Notion



Figma

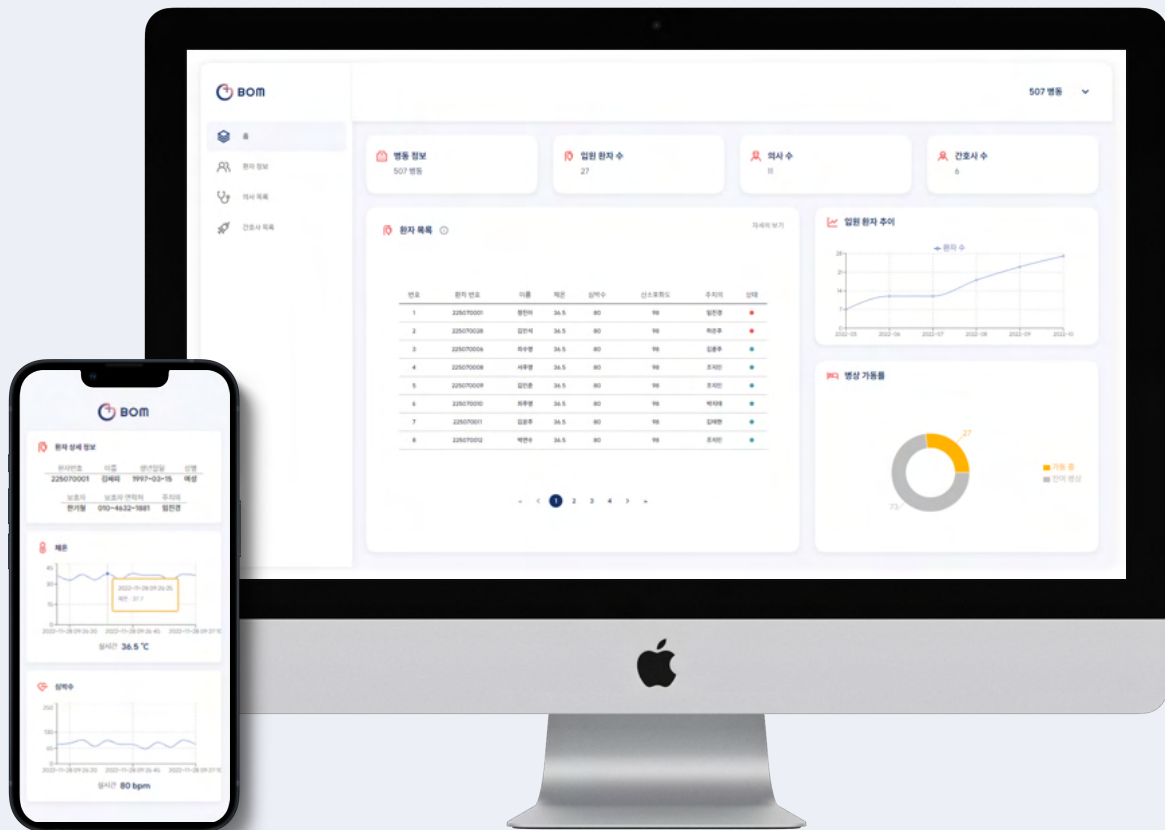
### 교육과정

2022.01 ~ 2022.12 삼성 청년 SW 아카데미

2012.03 ~ 2018.02 연세대학교 생활디자인, 심리학

# BOM

환자 및 BMS 모니터링 서비스



인원 5명

기간 2022.10.11 ~ 2022.11.30

기술 스택 React, Django, Raspberry Pi 등

코드 정보 [GitHub](#)

# BOM

## 프로젝트 개요



**BOM**은 메디컬 웨어러블 서비스로, 보다는 의미와 환자의 몸을 이중적으로 표현하고자 하였습니다. 본 프로젝트는 환자의 **건강 정보를 모니터링**해 실시간으로 위험을 감지할 수 있게 합니다. 또한 웨어러블 디바이스에서 필수적인 **배터리 관리**를 제공하며, 이 모든 정보를 웹을 통해 통합 관리할 수 있는 **웹 IoT 서비스**입니다.

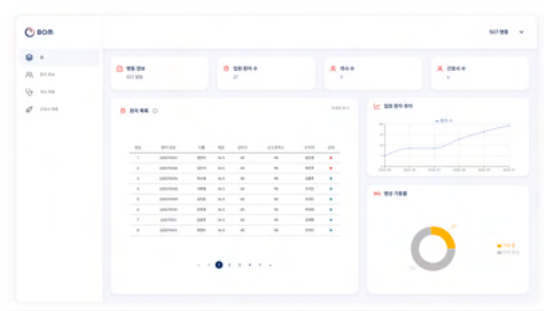
## 상세 설명

- **담당 역할**    프론트엔드 엔지니어
- **담당 기능**    로그인, 환자 정보
- **기여도**        20%
- **기술 스택**    React, tailwindCSS, HTML, CSS, VSCode
- **라이브러리**    axios, daisyui, recharts, react-js-pagination, react-responsive-carousel
- **특이 사항**    삼성 SDI 기업 연계 프로젝트

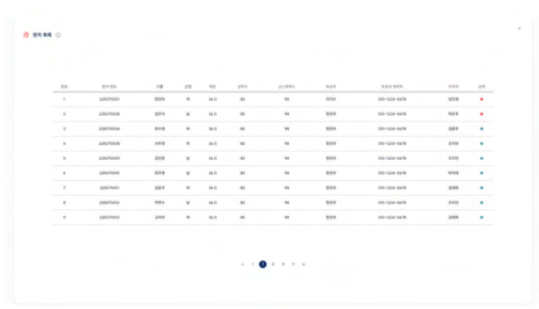
## 예시 이미지



로그인 페이지



메인 페이지



환자 목록 확대 페이지



의사 목록 페이지

# Findit!

시 기반 온오프라인 연계 보물찾기 게임



인원 6명

기간 2022.09.05 ~ 2022.10.07

기술 스택 React, Spring Boot, MySQL, stompJS, YOLOv5, Fast API 등

코드 정보 [GitHub](#)

# FindIt!

## 프로젝트 개요



**FindIt!**

FindIt!은 Object Detection과 특징점 검출을 활용한 온오프라인 연계 보물찾기 실시간 웹 게임입니다. 동심을 찾아 떠나고 싶은 어른들, 야외 활동이 필요한 어린이들 모두가 즐길 수 있는 게임으로 기획하였습니다. 숨겨진 보물을 **핸드폰 카메라로 인식**하여 점수를 획득하고 대결하는 **실시간 웹 게임**입니다.

## 상세 설명

- **담당 역할**    프론트엔드 엔지니어
- **담당 기능**    인트로, 로그인, 방장 메인, 게임 설정
- **기여도**        17%
- **기술 스택**    React, Redux, MUI, HTML, CSS, stompJS, VSCode
- **라이브러리**   axios, redux-persist, react-camera-pro, react-responsive-carousel
- **특이 사항**    로그인 사용자와 비로그인 사용자 분리 / 소켓 통신

## 예시 이미지



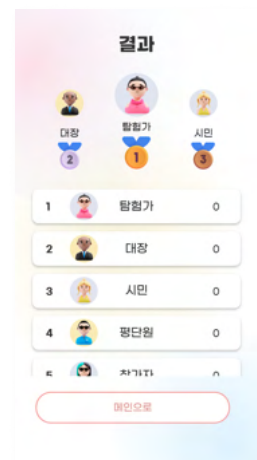
메인 페이지



게임 생성 페이지



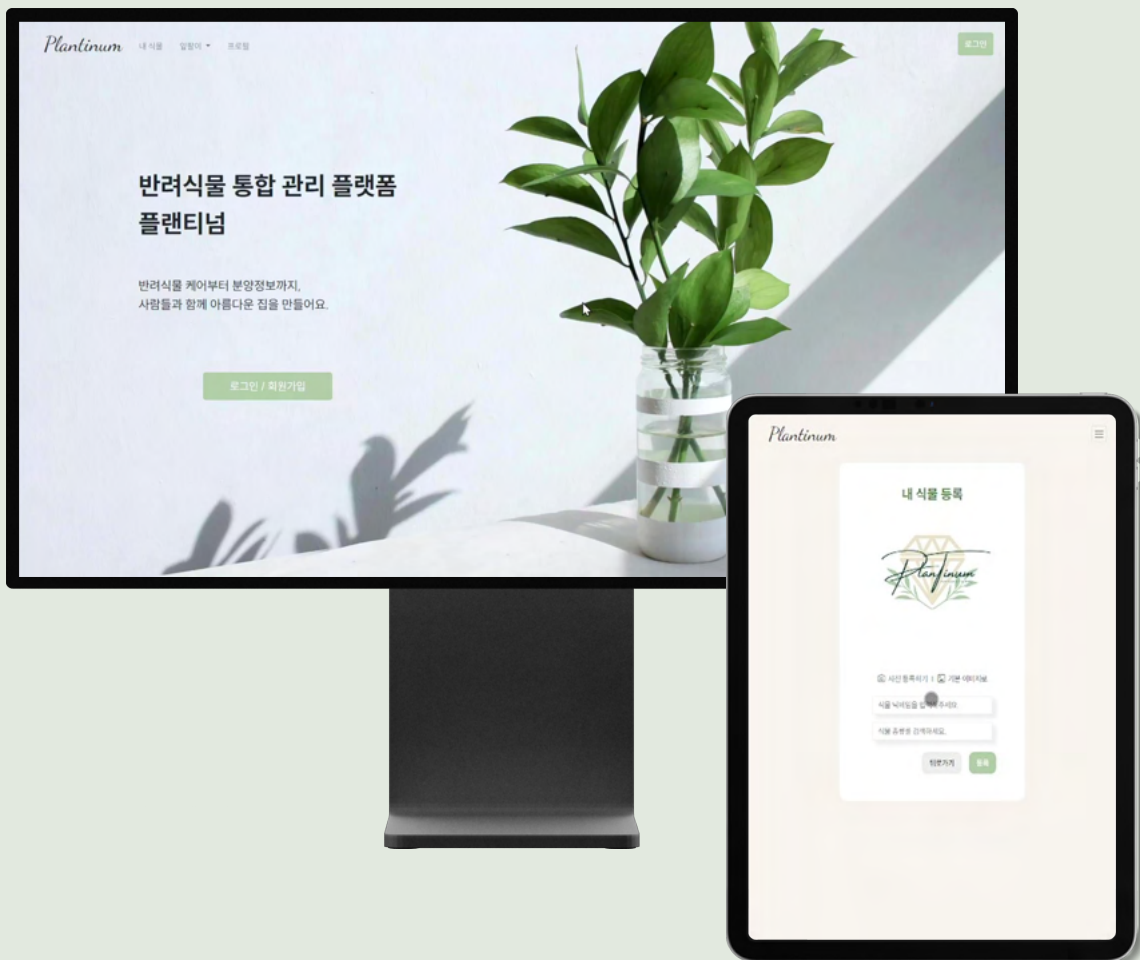
게임 진행 페이지



게임 결과 페이지

# PLANTinum

식물 자동 관리 및 식물 거래 플랫폼



인원 5명

기간 2022.07.11 ~ 2022.09.04

기술 스택 Vue, Django, node.js, socket.io, Raspberry Pi 등

코드 정보 [GitHub](#)



# PLANTinum

## 프로젝트 개요



PLANTinum 이란 Plant + Platinum 의 합성어입니다. 또한, plan을 의미하기도 해서 계획적으로 식물을 관리하고 보살핀다는 의미를 담고 있습니다. 본 프로젝트는 반려식물을 자동으로 케어해주는 **내 식물** 서비스와 웹을 통하여 이렇게 기른 반려식물을 거래할 수 있게 만드는 **오픈마켓**으로 구성된 IoT 플랫폼 서비스 개발을 목표로 합니다.

## 상세 설명

- **담당 역할** 프론트엔드 엔지니어
- **담당 기능** 로그인, 내 식물 조회, 내 식물 등록, 오픈마켓 채팅
- **기여도** 20%
- **기술 스택** Vue, Vuex, HTML, CSS, VSCode
- **라이브러리** axios, vuex-persistedstate, socket.io-client
- **특이 사항** 화분에 장착된 센서값을 웹에 반영하여 정보 전달

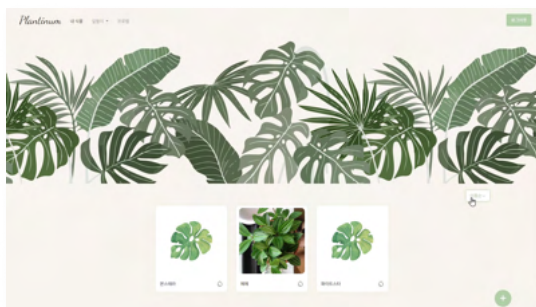
## 예시 이미지



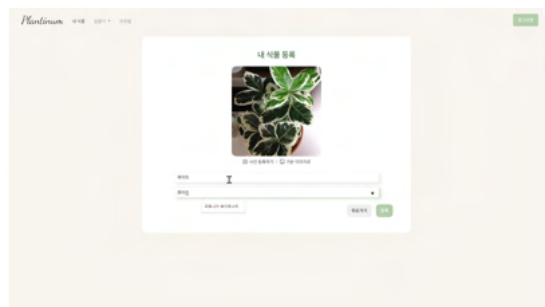
메인 페이지



로그인 페이지



내 식물 페이지



내 식물 등록 페이지



# 자기소개서

## ■ 기록을 통한 협업 능력 향상

식물 자동 관리 및 식물 거래 웹 IoT 서비스인 PLANTinum에서 협업을 통해 프로젝트를 성공적으로 완성한 경험이 있습니다. 프론트엔드 담당자인 저를 비롯하여 백엔드, 임베디드 팀원들이 함께 모였습니다. 서로 다른 배경의 6명이 모여 협업하기 위해 다음과 같은 노력을 기울였습니다.

첫 번째로 commit 메시지, branch 전략 등의 컨벤션을 마련했습니다. 서로 공유된 환경에서 진행상황을 쉽게 파악하기 위해서는 개발에 앞서 규칙을 정하는 것이 중요했습니다. 따라서 모두가 모여 합의된 컨벤션을 마련하는 시간을 가졌고, 확정된 컨벤션을 GitLab wiki에 기록해두어 필요할 때마다 편하게 조회할 수 있도록 했습니다. 두 번째로는 일정 관리를 위해 Jira를 활용하였습니다. 매일 아침 데일리 스크럼을 진행했고, 이를 토대로 각자 할 일과 해결한 일을 Jira에 상세히 기록해 두었습니다. 해당 기록을 보면서 쉽게 개발 상황을 파악하고 개발 속도를 맞춰나갈 수 있었습니다. 마지막으로, 회의 내용이나 함께 문제를 고민한 내용을 Notion 등에 문서화해서 명확한 의사소통에 도움이 되고자 했습니다. 이외에도 mattermost, Figma 등 다양한 협업툴을 사용하여 협업의 폭을 넓히는 것을 경험할 수 있었습니다.

## ■ 요구사항과 품질을 모두 충족시킨 개발 역량

명세사항을 준수하면서도 품질을 고려하여 개발할 수 있는 역량이 제가 가진 차별화된 강점입니다. BOM 프로젝트는 삼성 SDI와 기업 연계로 진행되었고, 배터리 관리 시스템(BMS)을 모니터링할 수 있는 웹 서비스를 구현하는 것이 명세로 주어졌습니다. 저를 비롯한 프론트엔드 팀은 React를 사용하여 BMS 데이터를 시각화한 웹 페이지를 구현하는 역할을 맡았습니다. 프론트엔드 담당자로서 웹 품질을 높이는 것이 모니터링하는 사용자 경험을 위해 중요하다고 생각했습니다. 따라서 웹 품질을 수치화한 Lighthouse 점수를 모든 항목에서 90점 이상 받는 것을 목표로 삼았습니다.

먼저, 코드의 재사용성을 높여 중복 코드를 줄이고자 코드 작성 전에 프론트엔드 팀원들과 함께 컴포넌트를 구조화하였습니다. 컴포넌트를 모아 페이지를 구성했기에 디자인적 통일성은 물론, 전체 코드량을 축소시켜 페이지 조회 속도를 단축하는 등의 이점을 가질 수 있었습니다. 두 번째로, API 요청을 setTimeout 순환 구조로 작성하여 데이터의 중복을 막고자 했습니다. 소켓을 사용하지 않고 실시간성을 확보해야 했기 때문에 일정 간격으로 백엔드에 데이터 요청을 해야했습니다. 따라서 응답 성공 여부에 따라 타이머함수를 작동시켜 안정적으로 데이터를 조회할 수 있는 페이지를 구현할 수 있었습니다. 이러한 노력의 결과로 목표했던 Lighthouse 점수를 90점 이상 기록하며 실시간 배터리 모니터링 웹 서비스를 높은 퀄리티로 구현할 수 있었고, 자율 프로젝트 전국 130개 팀 중 우수 8개 팀으로 선정될 수 있었습니다.