프로젝트 명세서

웹 IoT 프로젝트

“ PLANTinum, IoT 플랫폼 ”

A109팀
팀장 임진경
팀원 김유민
팀원 문요성
팀원 이대현
팀원 이지수
팀원 정진아

목차

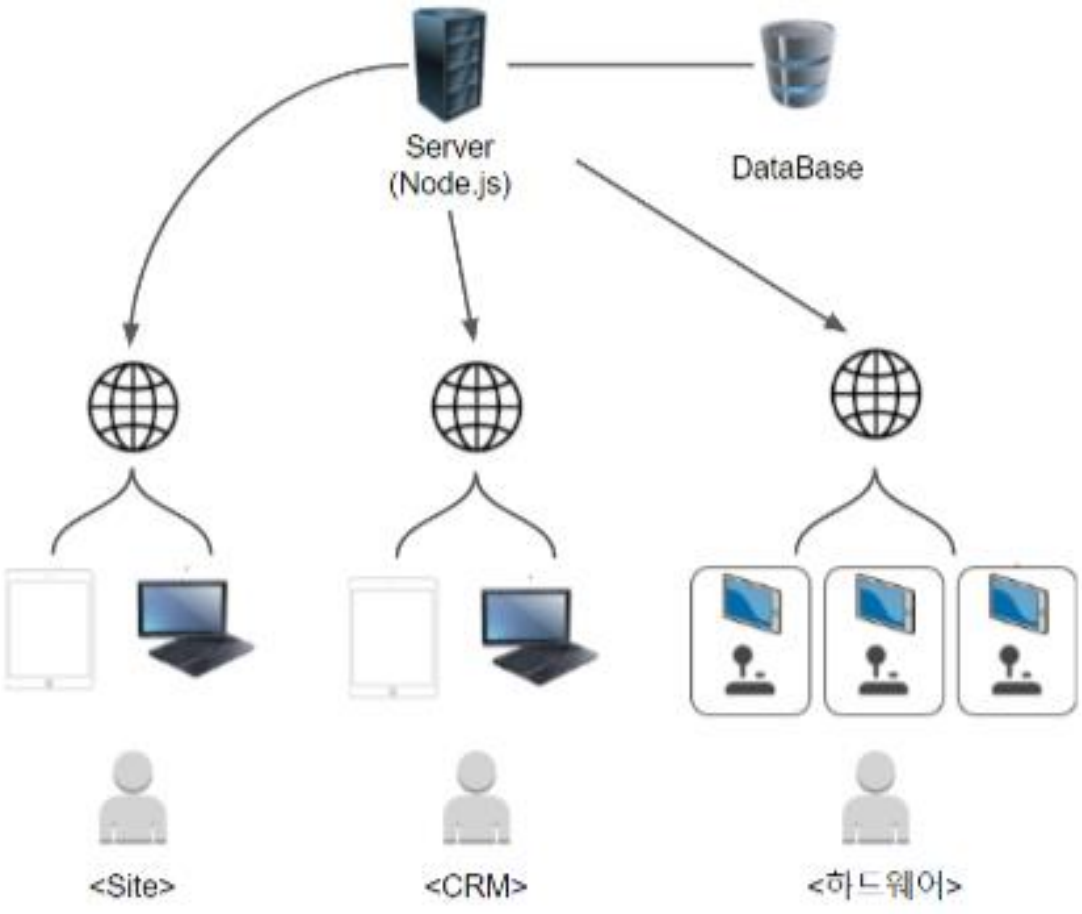
1. 프로젝트 개요

소개

PLANTinum이란 Plant(식물)와 Platinum(백금)의 합성어를 의미합니다. 또한, 플랫폼을 통해 계획적(Plan)으로 식물을 관리하고 보살핀다는 의미를 담고 있기도 합니다. 본 프로젝트에서는 반려 식물을 케어해주는 자동화된 하드웨어 시스템과 웹 커뮤니티를 통합한 플랫폼 서비스를 제공하는 『IoT 플랫폼』 구현을 목표로 합니다.

"플랫폼 연동 IoT 서비스" PLANTinum은 사용자가 화면을 통해 화분 내의 토양 상태를 확인하고, 자동화된 케어 서비스를 제공하도록 하는 몇 가지의 선택 항목을 가집니다. 또한 연동된 웹 플랫폼에서 식물 상태와 서비스 기록을 확인할 수 있으며 비슷한 다수의 사용자가 경험과 소통을 나눌 수 있는 커뮤니티를 제공하는 IoT 플랫폼입니다.

전체 서비스의 흐름은 다음과 같습니다.



프로젝트의 진행은 3개의 Sub PJT로 진행되며, 각 Sub PJT의 진행은 아래 표와 다음 각 단계를 설명한 내용을 참고하시기 바랍니다.

Sub PJT 1. BE	Sub PJT 2. FE	Sub PJT 3. HW
- Django로 백엔드 구성 - MySQL로 DB 구성 - 게시글 관리	- Window에 기반한 환경 구성 - Vue로 온라인 웹 플랫폼 제작 - 반응형 웹	- Raspberry Pi로 하드웨어 구성 - API 제작 - BE 서버 연결

Sub PJT 1. Back-End

Sub PJT 1에서는 Django를 사용하여 서비스의 백엔드 영역을 구현합니다. 주 기능으로는 회원관리, 식물 페이지, 커뮤니티, 프로필 관리, 관리자 페이지 등이 있습니다.

Sub PJT 2. Front-End

Sub PJT 2에서는 Vue를 사용하여 서비스의 UI를 구현합니다. 화면은 메인과 개인 프로필, 커뮤니티로 나뉘며 각 화면은 반응형으로 웹과 모바일에 차이가 있습니다. 관리자용 페이지가 따로 존재합니다. 구현이 완료되었다면 부가 기능으로 식물 거래를 돕는 당근 페이지를 구현합니다.

* BE/FE 고려 사항

1. 메인페이지: 물주기, 식물관리, 식물선택(식물 클릭 시 해당 식물의 다이어리로)
2. 다이어리: 식물 상세페이지, 게시글 형태
   - 식물 사진(등록을 하되, 없으면 default image를 넣는데 이미지 구현안되면 그냥 글로만), 식물 정보, 내용(기록), 최근 물주기 시간
   - ex) 인스타그램 마이페이지
3. 마이페이지: 회원 등급 ⇒ 등급 기준, 혜택 (부가기능), 비밀번호 변경? (고려중), 회원가입할 때 입력했던 정보, 정보 수정
4. Nav 바: 마이페이지, 메인페이지, Logo, 커뮤니티
5. 커뮤니티: 게시글 형식, 질문답변 게시판(1순위), 자유게시판(2순위), 검색 기능(부가기능)
6. 식물 정보: 계절에 따라 정보를 갱신(온도 센서를 통한 데이터로 기준 판별 (ex. 온도가 10도 이하면 겨울) ⇒ 기상청 등에서 자세한 데이터 확인 후 기준 정할 것. 기준에 이유가 있을 것.)
7. 보안 토큰은 basic token
8. id는 기본 email만 입력(id가 email 형태) / 구현이 어려울 시 id와 email 구분해서 입력 받기

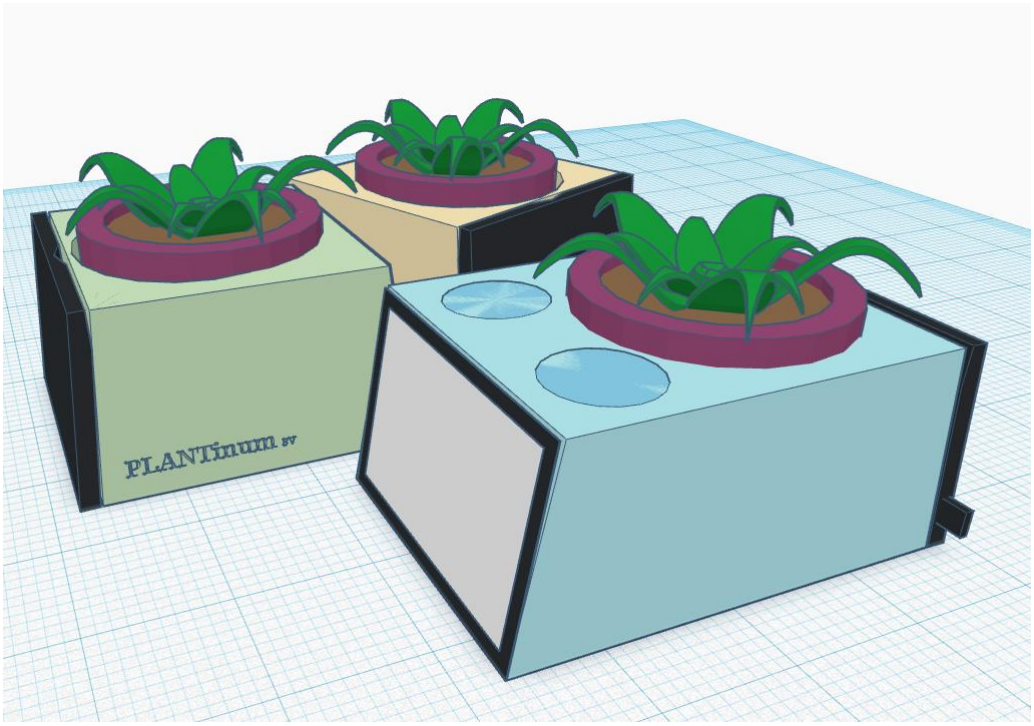
9. 게시판에서 글쓴이 클릭 시 글쓴이가 적은 글 리스트 보이도록(타인에게는 개인정보 비공개)

Sub PJT 3. Hardware

Sub PJT 3에서는 라즈베리파이와 ESP32를 사용하여 서비스의 하드웨어를 구현합니다. LCD 화면에 식물의 토양 습도와 온도, 기존 기록 등의 정보를 출력하며 웹과 연동하여 식물의 상태를 기록할 수 있습니다. 토양습도 센서, 온도 센서, 워터 필터 등 다양한 센서와 라즈베리파이를 조합하여 뼈대를 제작한 후, 3D 프린터로 프레임을 출력합니다. 각 센서들이 수집한 데이터들은 정해진 시기에 맞춰 서버로 전송됩니다.

*** HW 고려 사항**

1. 보드는 하나 ⇒ 라즈베리파이
2. 3D 프린터 재질(단열, 방수, 무독성)
3. 발열이슈 ⇒ 발열을 잡을 수 있다면 디자인적 요소를 가미해서 수정
   만약 불가능하다면
   1) 사용자가 직접 급수
   2) 보드를 외부로 빼냄
   3) 열차단 소재 (방열판)
   4) 보드 변경(STM32)
   5) 화분 개수 늘려서 열 분산
4. 3D 모델링 크기: 20*20 한계? ⇒ 어렵다면 조립형
5. 서보모터 대신 워터 펌프 ⇒ 위에 3D 모델링으로 조립형 상판 제작
6. LCD 화면 구성 - 절전모드(Watchdog)
7. 수분 잔량 측정: 비접촉식 수위 센서 2개 연결 ⇒ 50%, 20% 안내



2. 프로젝트 목표

---

1) Django와 Vue를 활용한 IoT 플랫폼 구축
2) 임베디드 KIT를 이용한 IoT 서비스 구현

3. 필수 지식 학습

프로젝트 진행을 위한 자기주도 학습에 실마리가 될 수 있는 필수 학습 대상 키워드와 유용한 참고 자료 링크 목록이 아래 표에 제시되어 있습니다. 아래 목록에 제시된 내용은 기초적인 내용에 해당하는 예이며, 이외 추가로 필요한 항목들을 직접 찾아 학습해 나가시기 바랍니다. *

주제		참고 링크
반응형 SPA	ES6	JavaScript의 Array 객체에 대한 소개 https://developer.mozilla.org/ko/docs/Web/JavaScript/Reference/Global_Objects/Array
	React	React 공식 문서(React 기본 예시, CRA, JSX, Hook, Context 등) https://ko.reactjs.org/docs/hello-world.html
	SPA	SPA 개념 설명 https://linked2ev.github.io/devlog/2018/08/01/WEB-What-is-SPA/
	반응형 웹	반응형 웹 디자인 패턴 https://developers.google.com/web/fundamentals/design-and-ux/responsive/patterns?hl=ko
	MariaDB	Maria DB 공식 문서 https://mariadb.org/documentation/
임베디드 키오스크 제작	Raspbian	Raspbian OS 설치 가이드 https://geeksvoyage.com/raspberry%20pi4/installing-os-for-pi4/
	Arduino	아두이노 기초 http://blog.naver.com/PostView.nhn?blogId=jamduino&logNo=220812436210
	Arduino 연결	Node.js에서 시리얼 포트를 활용한 아두이노 연결 https://linked2ev.github.io/devlog/2018/08/01/WEB-What-is-SPA/
	Express.js	Express.js의 이해 및 설치 http://webframeworks.kr/getstarted/express.js
	Material-UI	Material UI Theme 사용법 https://bitly.kr/1LmbXZH

	pm2	프로세스 관리툴 중 하나인 pm2의 사용법 소개 https://blog.outsider.ne.kr/1197
	crypto	암호화 모듈 crypto 사용 https://nodejs.org/api/crypto.html
임베디드 키오스크 완성	인증	인증과 권한의 개념 https://hanee24.github.io/2018/04/21/authentication-authorization/
	JWT	JWT 장단점 및 토큰 구성 개념 http://www.opennaru.com/opennaru-blog/jwt-json-web-token/
	Socket	Socket의 정의 및 사용법 이해 https://en.wikipedia.org/wiki/Network_socket
	SSR	SPA에서의 SSR과 CSR https://bit.ly/2NK7bfp
	Next.js	Next.js 소개 https://nextjs.org/docs/getting-started
	CSS Animation	CSS Animation 소개 https://developer.mozilla.org/ko/docs/Web/CSS/CSS_Animations
	Infinite Scroll	React에서의 Infinite Scroll 구현 https://reactgo.com/react-infinite-scroll/

4. 과제 명세

Sub PJT 1. Back-End

Sub PJT 2. Front-End

Sub PJT 3. Hardware

- 20220712 - 설계

Req. 8. 센서 Pin

이름	필요 Pin
LCD monitor	• 리본 케이블 1 • GND • VCC
모터드라이브 (L9110S 모델 사용예정) 소형 모터드라이브	• GND • VCC • Pin 2개
워터펌프	• 모터드라이브에 +, - 핀 연결
DHT11(온습도)	• GND • VCC • Data Pin 1
LED	• GND (저항 연결) • VCC
수위센서(접촉)	• GND • VCC • mcp3008에 Data Pin 1 연결
토양수분센서	• GND • VCC • mcp3008에 Data Pin 1 연결
mcp3008(디지털 컨버터)	• GND 2 • VCC 2

	• Data Pin 4

- 가격

이름	가격
모터드라이브	• 약 600원
워터 펌프	• 약 5500원
토양수분센서	• 220원 ~ 2000원
수위센서(비접촉)	• 22000원
수위센서(접촉)	• 1000~2000원
브레드보드	• 기존에 받는게 사이즈가 클수도 있어서 • 소형 5000원 예상
mcp3008	• 4~5000원 예상

5. 프로젝트 실행 및 배포

PM2를 이용한 프로세스 관리

애플리케이션의 배포 작업은 많은 경우 백그라운드 작업으로 진행되며, 이를 위해 별도의 도구를 사용하는 것이 일반적입니다. 또한 싱글 스레드 방식의 동작을 기본으로 하는 Node.js 런타임 환경의 작업 효율을 높이기 위해, 컴퓨터 내 모든 자원의 동시 활용을 지원하는 도구 역시 사용이 강제되는 경우가 많습니다.

Node.js의 프로세스 관리자 PM2는 이러한 두 가지 필요성을 모두 충족시켜주는 편리하고 유용한 도구입니다. PM2는 Node.js 환경의 백그라운드 프로그램 실행과 더불어 멀티 코어 시스템 CPU의 모든 코어를 활용한 프로그램 실행을 지원합니다.

플랫폼 자동화 설정

Raspberry Pi에서 UI를 활용하기 위해 크롬 브라우저의 키오스크 모드를 사용합니다. (위 내용은 "[명세서] 웹 서버 및 임베디드 기반 키오스크 제작"에 자세히 안내되어 있습니다.) 안타깝게도 저희는 해당 사항이 없으니 다른 방향을 생각해보고자 합니다. 실행 및 배포에는 aws 서버를 이용하기로 계획하고 있습니다.

로깅 설정

서버의 동작에 영향을 주지는 않지만, 실제로 서버를 운영하는 데 있어 가장 중요한 것 중 하나가 로그의 관리입니다. 서버에 문제나 장애가 발생했을 경우, 가장 빠르고 효과적인 해결책이 로그를 추적하는 것이기 때문입니다. express에서 자주 쓰이는 로깅 라이브러리로는 morgan과 winston이 있습니다.

프로그램 배포

예전으로 거슬러 올라가면 서버를 점검하고 수정할 때마다 서버를 정지시키고 재실행해서 변경사항을 적용하곤 했습니다. 이건 당연히 거대한 서비스상의 결함입니다. 서버가 재실행되는 동안은 사용자들이 접속을 할 수 없고, 처리 중에 있던 서비스가 중간에 중단될 수 있기 때문입니다.

Node.js로 배포할 경우

다행히도 PM2 설정과 코드에 신경을 쓴다면 이러한 문제를 없애고, 지속적인 서비스를 고려한 '무중단 배포'를 가능케 할 수 있습니다. 소스를 수정해서 저장하고, pm2 restart 대신 pm2 reload로 재시작을 하면 됩니다. restart는 프로세스를 무조건 kill한 후에 시작하기 때문에 사용해서는 안 됩니다.

단순한 상황에서는 이것만으로 중단이 사라지지만 이때 몇 가지 구멍에서 끊김 현상이 발생할 수 있습니다.

1. new_app이 완전히 동작하기 전에 요청을 받을 경우
2. old_app이 요청 처리 도중에 SIGKILL로 죽을 경우
3. HTTP Keep-Alive 가 사용될 경우 *

⇒ 실질적으로는 AWS로 배포할 예정입니다. 현재 의논 중!

* 별첨

각 명세서에 첨부된 별첨 목록을 첨부합니다.

웹 IoT 명세서 1. 반응형 SPA 제작

- Chocolatey 설치
- Node.js 설치
- VS Code 설치
- 프록시 서버 NginX 설치

웹 IoT 명세서 2. 웹 서버 및 임베디드 기반 키오스크 제작

- Raspbian OS 설치
- Raspbian OS 환경설정
- Maria DB 설치
- 미들웨어 활용