


## 설계 프로젝트 제안서

0. 작품(과제)명	KYK IoT (IoT 플랫폼)
1. 개발 동기 및 목적, 필요성	<p><b>1. 개발 동기</b> 대부분의 일상적인 디바이스들은 인터넷과 연결되어 있다. 센서를 사용하여 데이터를 수집하고 사용자에게 유용한 정보를 제공한다. 또한 기기 간에 상호작용을 가능하게 한다. 이러한 IoT 즉, 사물 인터넷 덕분에 우리들의 삶은 더욱 편리해졌다. KOREA IT TIMES에 따르면 2022년 전 세계 IoT 매출이 195억 달러를 기록하여 작년 대비 13%로 증가한 것으로 나타났다. 더 나아가 IoT를 스스로 개발하고 통제하고 싶은 사람들도 많아지고 있다. 이에 따라 IoT플랫폼에 대한 수요가 높아지고 있다. 하지만 현재 제공되는 IoT플랫폼은 유료인 경우가 대부분이다. 간혹 무료인 경우에는 데이터가 공개되거나 기기 연결 제한이 있다. 우리 팀은 이러한 현재 상황을 인식하여 새로운 IoT플랫폼을 만들기로 하였다. 이때 다른 IoT플랫폼과 다른 우리 IoT플랫폼만의 특징을 나타내는 기능을 추가하고자 생각하였다. 이에 가상현실에서 서로 소통할 수 있는 메타버스로 구성된 커뮤니티를 제공하기로 하였다. 무료이면서 기기 제한이 없고 데이터 보안과 메타버스 커뮤니티를 제공하는 IoT플랫폼을 개발하고자 한다.</p> <p><b>2. 개발 목적</b> 무료이고 데이터 보안이 철저하면서 기기 연결 제한이 없고 메타버스 커뮤니티를 제공하는 IoT 플랫폼을 개발하는 것이 목적이다. 학생들이 개발하는 경우나 대형 프로젝트를 개발하는 경우에 유료인 IoT플랫폼은 사용하기에 금전적으로 부담스럽다. 자신의 정보가 다른 사람에게 노출되는 일을 염려하는 사용자도 있다. 기기를 마음껏 여러 개를 연결하고 싶은 경우도 있다. 다른 사용자들과 소통하고 아이디어를 더 발전하고 싶은 경우도 있다. 이런 사항을 적극 수용하여 사용자가 편하게 사용할 수 있는 IoT플랫폼을 설계하고자 한다.</p> <p><b>3. KYK IoT</b> 새로운 IoT 플랫폼의 이름은 팀원들의 성을 합쳐서 'KYK IoT'로 정하였다. 로고는 K와 Y를 조합하여 디자인하였고 기술적이면서도 현대적인 느낌을 표현하였다. 노란색과 흰색의 밝은 색상을 이용하여 따뜻함과 편안함을 느끼게 함으로써 사용자들이 어렵지 않고 편하게 사용할 수 있는 IoT 플랫폼임을 표현하였다.</p> 
2. 과제 해결 방안 및 과정	<p><b>1. 기존 IoT 플랫폼 특징 소개 및 장단점 비교 분석</b></p> <p><b>(1) AWS IoT</b>  <b>◎특징:</b> AWS IoT는 아마존 웹 서비스에서 제공하는 IoT플랫폼이다. 디바이스 연결, 메시징, 보안, 규모 조정, 데이터 분석을 지원한다.  <b>◎장점</b></p>

- 대규모 IoT 디바이스와 애플리케이션을 위한 엔드 투 엔드(end-to-end) 서비스를 지원한다. 이때 엔드 투 엔드는 통신에서 송신자와 수신자 사이의 모든 노드를 거쳐 데이터 전송이 완료되는 것이다. 데이터 전송에 참여하는 모든 노드는 데이터가 암호화 되어 전송된다. 데이터의 안정성과 보안성이 보장된다.
- 신뢰성이 높은 연결성과 보안성을 제공한다.
- 디바이스, 애플리케이션, 클라우드 간 통신을 지원하는 다양한 프로토콜을 제공한다.
- 메시지 라우팅, 상태 추적, 규칙 기반 처리, 장애 조치, 데이터 분석 등 IoT업무에 필요한 다양한 기능을 제공한다.
- 다양한 AWS서비스와 통합이 가능하다.
- 대규모 IoT 업무에 대한 확장성이 뛰어나다.
- 개발 및 운영 비용이 절감된다.

#### ◎단점

- 소규모 애플리케이션의 경우 비용이 효율적이지만 많은 디바이스와 데이터 전송 및 다운이 자주 일어나는 대규모 애플리케이션의 경우 비용이 빠르게 증가한다.
- MQTT와 HTTP, WebSocket 프로토콜은 지원하지만 다른 프로토콜은 지원하지 않는다.

### (2) Naver Cloud Platform IoT

◎특징: Naver Cloud Platform은 Naver을 비롯하여 수많은 서비스들을 대상으로 한 신속하고 안정적인 IT인프라 운영 경험을 바탕으로 다양한 클라우드 서비스를 제공한다.

#### ◎장점

- 안정성이 뛰어나다. 서버가 다운되거나 장애가 발생할 경우 백업 서버를 통해 서비스를 제공한다.
- 여러 개의 서버를 운영하더라도 CPU 사용률과 메모리 사용률을 한 화면에 쉽게 확인할 수 있다.
- 즉각적인 데이터 처리와 대규모 데이터 저장이 가능하다.
- 기본적으로 웹 기반 대시보드를 제공하고 모바일 앱에서 사용 가능하다.

#### ◎단점

- AWS보다 서비스 제공 지역이 적어 다른 지역에서 운영하는데에 제한이 있다.
- 다른 클라우드 플랫폼에 비해 사용자 지정 옵션이 부족한 부분이 있다.
- 다른 플랫폼에 비해 연동 서비스가 부족하다.

### (3) Blynk

◎특징: IoT 애플리케이션 개발을 목적으로 제공된 클라우드 기반 오픈소스 플랫폼이다. 모바일 앱과 연동되어 IoT 디바이스를 제어할 있다.

#### ◎장점

- 가상의 디바이스를 만들어 IoT디바이스와 상호작용할 수 있다.
- 다양한 하드웨어 플랫폼과 호환이 된다.
- 오픈 소스로 제공되어 개발자들이 기능을 확장하고 수정할 수 있다.

#### ◎단점

- 오픈 소스로 제공되므로 보안에 취약할 수 있다.
- Blynk 클라우드 서버를 사용하면 추가 비용이 발생한다.
- 사용은 무료이지만 기능제한이 있다. 큰 프로젝트를 개발하기 위해서 유료 결제가 필요하다.

### (4) Adafruit

◎특징: 하드웨어 제조 및 소프트웨어 개발 회사이다. 개발 보드, 여러 종류 센서, 모터 등의 하드웨어 제품을 판매한다. 또한 대규모 커뮤니티를 운영하여 IoT 개발에 유용하다.

#### ◎장점

- 높은 품질의 하드웨어 제품을 판매한다. 제품이 안정적이고 효율적으로 작동한다.
- 세계적인 규모의 하드웨어 커뮤니티를 운영한다.
- IoT프로젝트를 위한 자습서 및 문서를 제공한다.

	<p>◎단점</p> <ul style="list-style-type: none"> <li>- 다른 저렴한 오픈 소스 옵션과 비교할 때 가격이 비싸다.</li> <li>- Adafruit 제품은 Adafruit에서 직접 판매되기 때문에 지역적으로 접근하기 어렵다.</li> </ul> <p><b>(5) Microsoft Azure IoT</b></p> <p>◎특징: Microsoft의 클라우드 기반 IoT 플랫폼이다. 다양한 기능과 서비스를 제공하여 IoT 솔루션을 구축하고 관리할 수 있다.</p> <p>◎장점</p> <ul style="list-style-type: none"> <li>- 대규모 디바이스 네트워크를 지원하여 필요에 따라 수천 대의 디바이스를 연결하고 관리할 수 있다.</li> <li>- 다양한 통신 프로토콜을 지원하여 다양한 종류의 디바이스와 통신할 수 있다.</li> <li>- 강력한 보안 및 인증기능을 제공하여 디바이스 및 데이터 보안을 강화할 수 있다.</li> <li>- 디바이스에서 생성된 데이터를 처리하는 기능을 제공한다.</li> </ul> <p>◎단점</p> <ul style="list-style-type: none"> <li>- 사용한 서비스, 리소스에 따라 비용이 발생한다.</li> <li>- 다양한 기능과 서비스를 제공하여 초기 설정 및 구성이 복잡하다.</li> </ul> <p><b>2. KYK IoT 기능 설명 및 구현 방식</b></p> <p><b>(1) 자체 서버 운영</b></p> <p>◎기능 설명: 서버 호스팅 종류로는 여러 가지가 있다. 공유 호스팅, 가상 서버 호스팅, 전용 서버 호스팅, 클라우드 서버 호스팅, 자체 서버 운영 등이 있다.</p> <ul style="list-style-type: none"> <li>- <b>공유 호스팅:</b> 여러 사용자들이 하나의 서버를 공유하여 이용하는 호스팅 방식이다. 비용이 저렴하지만 리소스 공유로 인해 다른 사용자들의 활동에 영향을 받는다.</li> <li>- <b>가상 서버 호스팅:</b> 하나의 물리적 서버를 여러 개의 가상 서버로 분할하는 방식이다. 사용자는 각자의 가상 서버를 독립적으로 사용할 수 있고, 자원 할당이 유연하다.</li> <li>- <b>전용 서버 호스팅:</b> 사용자가 전용 서버를 렌트하여 이용하는 방식이다. 높은 성능을 가지고 다른 사용자의 영향을 받지 않아 안정적인 서비스를 제공한다. 하지만 비용이 높은 편이다.</li> <li>- <b>클라우드 서버 호스팅:</b> 인터넷을 통해 가상 서버를 이용하는 방식이다. 서버의 자원을 필요에 따라 증감할 수 있다. 또한 비용이 유연하게 부과되므로 필요한 만큼만 지불할 수 있다.</li> <li>- <b>자체 서버 운영:</b> 사용자가 직접 서버를 구매하거나 기존의 서버를 사용하여 운영하는 방식이다. 자유도가 높다. 하지만 관리비용이 높은 편이다.</li> </ul> <p>IoT 플랫폼에서 클라우드 서버 호스팅을 할 경우 일정한 돈을 지급해야 하므로 무료로 IoT 플랫폼을 사용할 수 없다. 또한 다른 호스팅은 리소스 공유 문제가 있거나 다른 서버를 렌탈 또는 구매해야 하는 문제가 있다. 이러한 문제점을 해결하기 위해 자체 서버를 이용하기로 하였다. 개인이 가지고 있는 노트북이나, 컴퓨터를 이용하여 서버를 대체한다면 무료로 사용할 수 있다. 직접 관리가 가능하고 다른 서버를 사용하는 것에 비해 보안성도 우수하다.</p> <p>◎구현 방식: (기존 os를 바꾸지 못하는 경우 가상머신을 사용한다.) os는 리눅스를 사용한다. 그 이유는 다음과 같다.</p> <ul style="list-style-type: none"> <li>- 오픈소스이며 무료이다. 코드를 직접 확인할 수 있는 장점이 있고 경제적이다.</li> <li>- 안정적이다. 리눅스는 앞서 설명한 것처럼 오픈소스이므로 수많은 개발자들이 참여하여 개발하고 유지보수를 한다. 버그나 취약점을 빠르게 발견하여 해결되어 안정적이다.</li> <li>- 보안에 강력하다. 기본적으로 보안에 강화된 설계를 하였고 보안 기능을 제공한다.</li> <li>- 정보를 얻기 쉽다. 대부분의 개발자들이 리눅스에서 개발을 하므로 다른 os에 비해 상대적으로 정보를 얻기 쉽다.</li> </ul> <p>이후 웹서버는 Apache를 사용하였다. C언어 기반으로 만들어진 서버 사이드 스크립트 언어인 PHP를 사용하여 다양한 웹 페이지를 제공하는 동적 웹 페이지를 생성할 수 있도록 하였다.</p> <p><b>(2) 데이터 관리</b></p>
--	---

#### ◎기능 설명 및 구현 방법

- **데이터 수집:** IoT기기의 센서에서 데이터를 수집한다. 이때 프로토콜이 필요하다. 프로토콜은 원거리 통신 장비나 컴퓨터에서 서로 메시지를 주고 받는 통신 규약이다. 여러 프로토콜 중 MQTT를 사용하여 데이터를 수집하고 IoT플랫폼으로 전송한다. MQTT는 Message Queuing Telemetry Transport의 약자로 텔레메트리(telemetry) 장치나 모바일 기기에 적합한 프로토콜이다. 경량 프로토콜이면서 대역폭을 제한할 수 있고 전송 계층 보안(TLS)과 보안 소켓 레이어(SSL)를 이용해 암호화할 수 있어 보안적이다.
- **데이터 저장:** 수집된 데이터는 저장하기 위해 데이터베이스(DB)가 필요하다. 관계형 데이터베이스 모델인 MySQL를 사용하여 PHP와 연동하여 사용한다.
- **데이터 가공:** MySQL에 저장된 데이터를 가공한다. 필요한 데이터만 추출하여 유용하게 사용할 수 있다. 데이터 정제, 전처리 변환, 통합 등이 있다.
- **데이터 보안:** MySQL 암호화를 이용하여 데이터의 보안성을 높인다.

#### (3) 디바이스 관리

##### ◎기능 설명 및 구현 방식:

- **디바이스 등록:** IoT 디바이스를 플랫폼에 연결한다. MQTT 프로토콜을 사용한다. keep-alive 메시지를 사용하여 연결을 유지한다. 디바이스 ID도 지정한다.
- **데이터 수집:** 연결된 디바이스를 MQTT 프로토콜을 사용하여 수집한다.
- **상태 모니터링:** 디바이스 상태 정보를 주기적으로 수집한다. 플랫폼에서 디바이스 센서 데이터, 배터리 수명, 연결 상태 등을 보여준다.
- **제어 및 설정:** 디바이스의 고유 ID를 이용하여 디바이스를 식별한다. 이후 플랫폼에서 명령을 디바이스로 전송하여 제어 및 설정을 변경한다. 디바이스는 해당 명령을 처리하고 응답을 반환한다. 플랫폼은 응답을 처리하여 결과를 반환한다.

#### (4) 대시보드

◎기능 설명: 여러 IoT 디바이스와 센서에서 수집한 데이터를 사용자가 쉽게 볼 수 있도록 한 시각적 인터페이스이다. 일반적으로 그래프, 표, 맵 등 시각화 도구를 사용하여 데이터를 표시한다. IoT 시스템에서 현재 상태와 동향을 파악하기에 유용하다. 대시보드에서도 디바이스를 제어하거나 설정할 수 있는 기능을 제공한다.

##### ◎구현 방식

- **대시보드 구성요소 선정:** 사용자가 필요로 하는 정보를 파악하고 이를 반영할 구성요소를 선택한다. 장비의 위치, 디바이스 연결 상태, 디바이스 배터리 수명, 특정 센서의 데이터 등을 선택하였다.
- **데이터 수집 및 저장:** 센서를 통해 수집한 데이터를 데이터베이스에 저장한다.
- **대시보드 디자인:** 사용자가 읽기 쉽고 직관적인 디자인을 한다. 웹 프레임워크, 데이터베이스, 그래프 라이브러리 등을 활용하여 구현한다. 이때 대시보드를 구성하는 각 요소들을 데이터와 연동시킨다.

#### (5) 보안

##### ◎기능 설명 및 구현 방법

- **업데이트 관리:** 보안 취약점을 관리하기 위해 정기적으로 업데이트를 한다. 이때 업데이트된 버전을 제공하여 사용자들이 업데이트된 버전을 설치하도록 권고한다.
- **데이터 보안:** MySQL 암호화를 이용하여 데이터의 보안성을 높인다. MySQL에는 두가지 암호화 알고리즘이 있다. 대칭키 암호화 알고리즘은 암호화 키와 복호화 키가 동일한 것을 말하며 AES, DES가 있다. 일방향 암호화 알고리즘은 해쉬 함수를 이용하여 암호화하는 알고리즘이다. 하지만 복호화는 불가능하다는 특징을 가지고 있으며 MD5, SHA1, SHA2가 있다. 보안을 더 강화하기 위해 복호화가 불가능한 일방향 암호화 알고리즘을 사용하여 데이터를 보안한다.
- **접근 제어:** IoT 디바이스에 접근할 수 있는 사용자를 제한한다. 접근 제어 모델은 강제적 접근 제어 모델(MAC), 임의적 접근 제어 모델(DAC), 역할 기반 접근 제어 모델(RBAC), 속성 기반 접근 제어 모델(ABAC)이 있다. 보안을 위해 사용자에게 최소 권한을 부여하고 파

의 소유자가 그 파일에 대한 접근 제어를 설정할 수 있도록 하기 위하여 임의적 접근 제어 모델(DAC)을 선택하였다.

#### (6) 엣지 컴퓨팅

◎기능 설명: 엣지 컴퓨팅(Edge Computing)은 클라우드 서비스로 처리하기 어려운 처리 작업을 로컬 디바이스에서 처리하는 분산 컴퓨팅 모델이다. 많은 양의 IoT 디바이스를 사용하면 데이터양의 급증으로 인하여 네트워크 대역폭과 지연시간이 문제가 될 수 있다. 이 점을 해결하기 위하여 엣지 컴퓨팅은 로컬 디바이스에서 데이터를 처리한다. 이를 통하여 네트워크 대역폭과 지연 시간을 줄일 수 있다. 또한 실시간 데이터 처리를 할 수 있고 더 나은 보안을 제공할 수 있다. 네트워크 대역폭이 많으면 빠르게 데이터를 전송할 수 있지만 일정 수준 이상에서는 네트워크 병목 현상이 발생하여 오히려 전송 속도가 느려질 수 있다. 이때 병목 현상이란 전체 시스템의 성능이 하나의 구성요소로 인하여 제한받는 현상이다. 이러한 상황을 방지하기 위하여 엣지 컴퓨팅을 사용하여 네트워크 대역폭을 줄인다.

◎구현 방식: IoT 디바이스에서 생성된 데이터를 데이터베이스에 저장하기 위하여 MySQL를 설치한다. 로컬 데이터베이스를 생성한 후 사용자를 생성하고 권한을 부여한다.

#### (7) 도커 이미지 배포

◎기능 설명 및 구현 방법: 새롭게 만든 IoT플랫폼을 도커 이미지를 통해 배포한다. 도커는 컨테이너 기반의 오픈소스 가상화 플랫폼이다. 도커 파일을 작성 후 도커 이미지를 빌드하여 KYK IoT 홈페이지에 도커 이미지를 올려둔다. 또한 도커 이미지 다운받아 컨테이너를 실행하여 KYK IoT 플랫폼을 사용하는 방법도 PDF파일에 정리하여 홈페이지에 올린다. 도커이미지를 활용하여 IoT플랫폼을 설치하면 사용자의 자체 서버에서 IoT플랫폼을 실행할 수 있고 비교적 쉽게 설치가 가능하다.

#### (8) 메타버스(Metaverse)

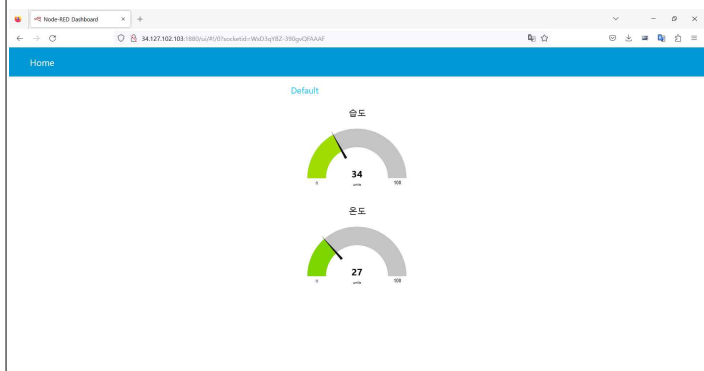
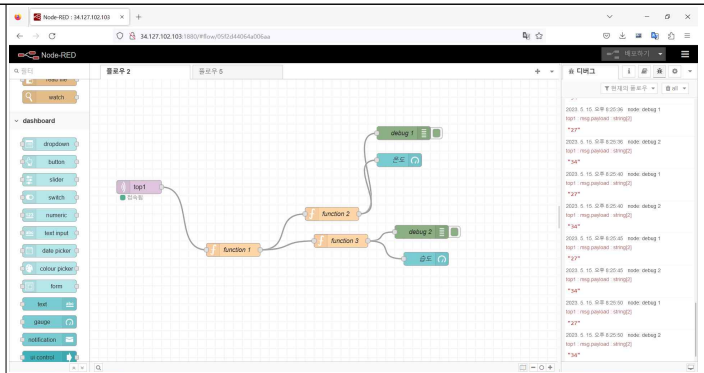
◎기능 설명: 메타버스는 현실세계와 같은 사회적·경제적 활동이 통용되는 3차원 가상공간이다. KYK IoT 플랫폼에 메타버스 환경으로 만들어진 커뮤니티로 통하는 위젯을 추가한다. 메타버스로 구성된 커뮤니티에서 사용자들은 서로의 IoT디바이스를 소개, 구경할 수 있고 다른 사용자와 의견을 공유할 수 있다.

##### ◎구현 방법

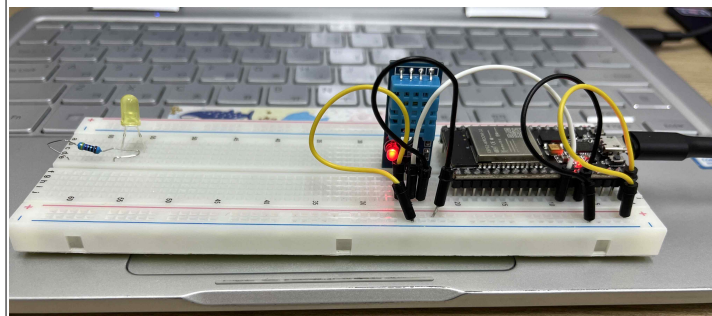
- **3D 모델링:** 웹 기반 가상 공간을 위하여 3D 모델링과 디자인 작업을 수행한다. 3D 그래픽 도구와 Unity 게임 엔진을 사용하여 가상공간의 모델과 환경을 디자인한다.
- **WebGL 사용:** WebGL은 Web Graphics Library의 약자로 웹상에서 3D 그래픽을 웹 브라우저에 렌더링한다. WebGL을 사용하여 3D 객체와 텍스처를 렌더링하고 사용자 입력을 처리한다.
- **사용자 인터페이스 개발:** 사용자가 웹 기반 가상 공간에 접속하고 다른 사용자와 상호작용을 할 수 있도록 사용자 인터페이스를 개발한다. HTML, CSS, JavaScript 등을 사용하여 사용자가 가상 공간에 접속하고 필요한 조작을 할 수 있는 UI를 제공한다.
- **WebSocket 사용:** 웹 소켓을 사용하여 클라이언트와 서버 간에 지속적인 연결을 유지할 수 있고 실시간 양방향 통신이 가능하다. 또한 낮은 지연 시간을 가지고 있어 실시간 통신에 적합하다.

#### 3. 모의 구현

- IoT 플랫폼은 Apache, MySQL, PHP, Node.js, Node-Red, MQTT를 이용하여 간단하게 구현하였다.
- IoT 디바이스는 ESP32와 DHT11을 사용하여 온습도를 측정하도록 만들었다.



```
moisture.ino
60 } else {
61   Serial.print("failed with state ");
62   Serial.print(mqttclient.state());
63   delay(2000);
64 }
65 }
66
67 mqttclient.subscribe(topic_sub);
68 mqttclient.publish(topic_pub, "ESP32 logged in");
69 prevupdateTime = millis();
70 }
71
72 void loop() {
73   mqttclient.loop();
74   sensors_event_t event;
75
76   unsigned long currentTime = millis();
77   if (currentTime > prevupdateTime + UPDATE_INTERVAL) {
78     dht.temperature().getEvent(&event);
79     float h = event.temperature;
80     dht.humidity().getEvent(&event);
81     float t = event.relative_humidity;
82     std::stringstream st;
83     st << h;
84     st << " ";
85     st << t;
```



	<p><b>4. 차별성과 우수성</b></p> <ul style="list-style-type: none"> <li>- AWS IoT와 Naver Cloud Platform IoT 등 다른 IoT 플랫폼은 유료이다. 하지만 KYK IoT는 자체 서버에서 구축하여 무료로 사용할 수 있다.</li> <li>- 엣지 컴퓨팅 기술을 사용하여 네트워크 대역폭과 지연 시간을 줄일 수 있다. 따라서 실시간 처리를 할 수 있으며 더 나은 보안성을 제공할 수 있다.</li> <li>- 도커 이미지를 제공하였다. 자체 서버의 버전 문제에서 벗어나 도커 이미지를 사용하여 간편하게 설치가 가능하다.</li> <li>- 메타버스 커뮤니티를 제공하여 사용자 간의 소통이 가능하다. 가상의 공간에서 서로의 IoT 디바이스를 볼 수 있으며 자신만의 공간으로 꾸밀 수 있다. 메타버스 환경이므로 다른 커뮤니티에 비해 시각적으로 즐기며 소통할 수 있다.</li> </ul>
<p><b>3. 과제 결과 및 결론</b></p>	<p><b>1. 제품 활용성 및 기대효과</b></p> <ul style="list-style-type: none"> <li>- 효율적으로 자원 관리를 할 수 있다. 센서와 장치를 통해 실시간 데이터를 수집하고 분석함으로써 자원의 사용량을 모니터링하고 최적화하며 효율적인 관리를 할 수 있다.</li> <li>- 가정이나 도시와 같은 환경에서 스마트 기기와 네트워크를 통합하여 스마트 홈이나 스마트 시티 솔루션을 구현할 수 있다. 스마트 홈은 조명, 난방, 보안 시스템 등을 자동으로 제어하여 사용자의 편의를 높일 수 있다. 스마트 시티는 도시 인프라 모니터링, 주차 관리, 쓰레기 수거 등을 효율적으로 관리할 수 있다.</li> <li>- 산업 자동화를 하여 생산성을 향상시킬 수 있다. 제조, 농업, 운송 등 다양한 산업 분야에서 자동화를 하여 생산성 향상을 도모할 수 있다. 생산 라인을 모니터링하고 제어하면서 불량률을 줄이고 생산 효율을 높일 수 있다.</li> </ul> <p><b>2. 향후 계획</b></p> <p>KYK IoT 플랫폼이 신뢰성을 얻고 많은 사용자가 사용을 한다면 사용자의 서버 성능에 따른 업그레이드 버전과 유료 버전을 출시할 예정이다. 업그레이드 버전은 데이터 마이닝 기능을 추가할 것이다. 이 기능은 사용자의 서버에 GPU가 필요하다. 유료 버전은 ChatGPT 기능을 추가하여 개발의 편의를 제공할 것이다.</p> <ul style="list-style-type: none"> <li>- <b>데이터 마이닝:</b> 대규모의 데이터에서 지식을 추출하는 과정을 말한다. 데이터 전처리에서 데이터를 수집하고 정제한다. 패턴 인식에서 데이터의 패턴을 인식하고 규칙을 찾는다. 모델링에서 추출된 패턴과 규칙을 사용해 예측모델을 구축하고, 평가에서 구축된 모델이 얼마나 정확한지 평가한다. 데이터 마이닝을 사용한다면 기업의 경우 고객 선호도, 상품 추천, 예측 분석 등 알 수 있다. 또한 새로운 비즈니스 모델을 발견하고 경쟁 우위를 확보할 수 있다.</li> <li>- <b>ChatGPT:</b> OpenAI는 사용자에게 API를 통한 GPT 모델 액세스를 제공한다. ChatGPT는 API의 호출 횟수, 사용량 및 기타 요소에 따라 금액이 추가된다. 이로 인해 ChatGPT를 사용한 KYK IoT는 유료화가 된다. 하지만 ChatGPT의 기능이 추가된다면 자동화와 자율성이 증진된다. 또한 대화기능을 사용하여 적절한 응답을 얻을 수 있다. 이로 인해 개발 속도가 빨라지게 되는 장점이 있다.</li> </ul>