

BỘ KHOA HỌC VÀ CÔNG NGHỆ
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



BÁO CÁO BÀI TẬP LỚN

HỌC PHẦN: IOT VÀ ỨNG DỤNG

**Đề tài: Hệ thống tưới cây thông minh ứng dụng
AI để lên lịch hằng tuần**

Giảng viên: TS. Kim Ngọc Bách

Nhóm 6

B22DCCN482 Trịnh Quang Lâm

B22DCCN434 Vũ Nhân Kiên

B22DCCN889 Vũ Thế Văn

B22DCCN422 Cao Thị Thu Hương

Hà Nội, 11/2025

Phân chia nhiệm vụ

Mã sinh viên	Họ tên	Công việc
B22DCCN482	Trịnh Quang Lâm (Nhóm trưởng)	<ol style="list-style-type: none">1. Lên kế hoạch, phân chia nhiệm vụ2. Thuyết trình3. Viết báo cáo4. Code Frontend
B22DCCN434	Vũ Nhân Kiên	<ol style="list-style-type: none">1. Làm Slide2. Thuyết trình3. Code Backend
B22DCCN889	Vũ Thế Văn	<ol style="list-style-type: none">1. Thuyết trình2. Code UI3. Báo cáo
B22DCCN422	Cao Thị Thu Hương	<ol style="list-style-type: none">1. Code phần cứng2. Thuyết trình3. Báo cáo

Danh sách bảng biểu và từ khóa viết tắt

Số thứ tự	Hình	Chú thích
1	Hình 1	Sơ đồ tổng quan hệ thống
2	Hình 2	Sơ đồ chân ESP32 DEVKIT ver 36 chân
3	Hình 3	Sơ đồ khối hoạt động cơ bản của DevKit 32D
4	Hình 4	Sơ đồ mạch nguyên lý tổng
5	Hình 5	Sơ đồ Use Case tổng quan

MỤC LỤC

I. Giới thiệu đề tài

1. Mô tả dự án

Dự án này nhằm xây dựng một hệ thống tưới cây tự động sử dụng vi điều khiển ESP32 kết hợp AI phân tích dữ liệu thời tiết. Dự án được thiết kế nhằm mang lại giải pháp nông nghiệp thông minh, hiện đại và hiệu quả. Hệ thống sử dụng ESP32 làm bộ xử lý trung tâm, kết nối với các cảm biến môi trường như cảm biến độ ẩm đất, cảm biến nhiệt độ và độ ẩm không khí để thu thập dữ liệu theo thời gian thực. Các dữ liệu này sẽ được truyền về hệ thống Backend để lưu trữ, phân tích và đưa ra quyết định tưới tiêu phù hợp.

Điểm đặc biệt của dự án là việc tích hợp AI phân tích dữ liệu thời tiết. AI sẽ dự đoán điều kiện thời tiết trong tương lai từ đó xây dựng lịch tưới tự động tối ưu cho từng tuần. Điều này giúp hạn chế tình trạng tưới thừa khi sắp có mưa hoặc thiếu nước khi thời tiết khô hạn kéo dài.

Ngoài ra, hệ thống còn cung cấp giao diện web trực quan phát triển bằng ReactJS, cho phép người dùng theo dõi trạng thái thiết bị, lịch tưới, dữ liệu cảm biến và can thiệp thủ công khi cần thiết. Phần Backend được xây dựng trên NodeJS với ExpressJS, kết nối với cơ sở dữ liệu MongoDB Atlas để lưu trữ thông tin. Nhờ đó, toàn bộ hệ thống hoạt động dựa trên mô hình IoT kết hợp AI, vừa tự động vừa cho phép giám sát và điều khiển từ xa thông qua Internet.

2. Mục tiêu và phạm vi của hệ thống

2.1. Mục tiêu hệ thống

- **Vấn đề thực tế cần giải quyết:** Trong canh tác truyền thống, người nông dân phải kiểm tra thủ công các yếu tố môi trường như độ ẩm đất, nhiệt độ. Công việc này không chỉ tốn công sức mà còn thiếu chính xác, thường dẫn đến tình trạng tưới nước không hợp lý: tưới quá ít làm cây khô héo, hoặc tưới quá nhiều gây lãng phí nước và làm úng rễ cây.
- **Mục tiêu của hệ thống IoT:** Dự án được xây dựng nhằm mang lại một giải pháp nông nghiệp thông minh, hiện đại và hiệu quả thông qua các mục tiêu cụ thể sau:
 - Tự động hóa giám sát và điều khiển: Hệ thống sẽ tự động thu thập dữ liệu môi trường theo thời gian thực (nhiệt độ, độ ẩm không khí, áp suất, độ ẩm đất). Dựa trên các ngưỡng được cài đặt, hệ thống sẽ tự động bật/tắt bơm tưới.
 - Tối ưu hóa việc tưới tiêu bằng AI: Điểm đặc biệt của dự án là tích hợp trí tuệ nhân tạo (AI) để phân tích và dự đoán điều kiện thời tiết, cụ thể là khả năng mưa trong 60 phút tới. Dựa vào dự báo này, hệ thống sẽ tự động điều chỉnh lịch tưới để tránh tưới thừa khi sắp có mưa, giúp tiết kiệm nước và bảo vệ cây trồng.
 - Giám sát và tương tác từ xa: Cung cấp một giao diện web trực quan, cho phép người dùng theo dõi trạng thái hệ thống, xem dữ

liệu lịch sử và can thiệp điều khiển thủ công từ bất kỳ đâu có kết nối internet.

2.2. Phạm vi hệ thống

- **Phạm vi triển khai:** Trong giai đoạn hiện tại, hệ thống được thiết kế và triển khai cho quy mô nhỏ như vườn gia đình, ban công hoặc một khu vực canh tác thử nghiệm
- **Phạm vi phần cứng:**
 - o Bộ điều khiển trung tâm: Sử dụng 1 vi điều khiển ESP32
 - o Cảm biến: Hệ thống tích hợp các cảm biến môi trường bao gồm DHT22 (nhiệt độ, độ ẩm không khí), BME280 (áp suất), và cảm biến độ ẩm đất.
 - o Thiết bị: Sử dụng 1 bơm nước mini 12V được điều khiển thông qua module MOSFET, cho phép không chỉ bật/tắt mà còn có khả năng điều chỉnh lưu lượng tưới (PWM).
- **Ngoài phạm vi hệ thống:** Ngoài phạm vi hệ thống: Hệ thống hiện tại chưa bao gồm các chức năng mở rộng như bón phân tự động hay giám sát sâu bệnh. Việc tối ưu hóa cho các khu vực canh tác quy mô lớn cũng sẽ được xem xét trong các giai đoạn phát triển sau.

3. Thu thập yêu cầu của các bên liên quan

3.1. Đối với người dùng cuối (người làm vườn, người quản lý hệ thống):

- **Yêu cầu về chức năng:** Cần một giao diện web đơn giản để có thể xem tất cả các thông số môi trường dưới dạng số liệu và biểu đồ trực quan. Yêu cầu chức năng điều khiển bật/tắt bơm tưới công khi cần thiết, ghi đè lên lịch tự động. Cần có chức năng xem lại nhật ký và lịch sử các lần tưới để đánh giá.
- **Yêu cầu về trải nghiệm:** Giao diện phải thân thiện, dễ thao tác, kể cả với người không có nhiều kiến thức về công thức

3.2. Đối với nhóm phát triển

- **Yêu cầu kỹ thuật:** Hệ thống phải là một sản phẩm hoàn chỉnh, tích hợp liền mạch giữa các khối: Phần cứng (ESP32, cảm biến), Backend(NodeJS, ExpressJS), Frontend(ReactJS) và AI(mô hình XGBoost).
- **Yêu cầu về quy trình:** Cần một hệ thống quản lý mã nguồn (Git/Github) để các thành viên có thể cộng tác, theo dõi thay đổi và quản lý phiên bản một cách hiệu quả.
- **Ràng buộc hệ thống:** Phải đảm bảo các giả định và ràng buộc kỹ thuật được tuân thủ, ví dụ như AI cần ít nhất 60 ngày dữ liệu để huấn luyện, và hệ thống phải có chế độ hoạt động dự phòng khi mất kết nối Internet

4. Tiêu chí thành công(KPIs)

Để đánh giá mức độ thành công của dự án, các tiêu chí định lượng sau được đặt ra:

- Hiệu quả và tối ưu hóa:
 - Giảm tưới trùng mưa: Giảm tối thiểu từ 25%-40% số lần tưới không cần thiết khi trời sắp mưa
 - Tiết kiệm tài nguyên: Tiết kiệm từ 15%-30% lượng nước sử dụng so với

phương pháp tưới thủ công hoặc hẹn giờ thông thường

- Độ chính xác và độ tin cậy:
 - Độ chính xác cảm biến: Sai số của cảm biến độ ẩm đất phải nằm trong khoảng tưới dưới 5%
 - Độ tin cậy truyền dữ liệu: Tỷ lệ các gói tin dữ liệu từ cảm biến gửi lên server phải lớn hơn 98%
- Hiệu năng và độ trễ:
 - Thời gian cập nhật dữ liệu: Dữ liệu từ cảm biến phải được cập nhật lên server trong vòng dưới 5 giây
 - Thời gian phản hồi AI: Thời gian để AI xử lý và trả về kết quả dự báo cho mỗi yêu cầu phải nhỏ hơn 300ms
- Khả năng mở rộng:
 - Kiến trúc phần mềm và phần cứng phải được thiết kế để có thể dễ dàng hỗ trợ thêm các cảm biến mới hoặc các khu vực tưới mới trong tương lai mà không cần thay đổi lớn về hạ tầng

5. Kết quả mong đợi

Dựa trên các phân tích trên, dự án khi hoàn thành được kỳ vọng sẽ đạt được kết quả sau:

- **Một hệ thống tưới hoàn toàn tự động:** Hệ thống có khả năng tự vận hành một cách thông minh, từ việc thu thập dữ liệu, phân tích, dự báo cho đến ra quyết định tưới, giúp giải phóng sức lao động và tối ưu hóa quy trình chăm sóc cây trồng
- **Giao diện quản lý trực quan và hiệu quả:** Người dùng sẽ có một công cụ mạnh mẽ để giám sát khu vườn của mình từ xa, đảm bảo họ luôn nắm được tình hình và có thể can thiệp kịp thời khi cần thiết
- **Minh chứng về hiệu quả tiết kiệm:** Các chỉ số về tiết kiệm nước và năng lượng phải được ghi nhận và chứng minh thông qua dữ liệu lịch sử, khẳng định giá trị thực mà công nghệ IoT và AI mang lại.
- **Một nền tảng dữ liệu có giá trị:** Hệ thống sẽ liên tục thu thập và lưu trữ dữ liệu về môi trường và các hoạt động tưới. Nguồn dữ liệu này là quan trọng, có thể được dùng để tiếp tục phân tích, cải tiến mô hình AI và đưa ra các quyết định canh tác tốt hơn trong tương lai

II. Mô tả tổng quan hệ thống

1. Mô tả tổng quan

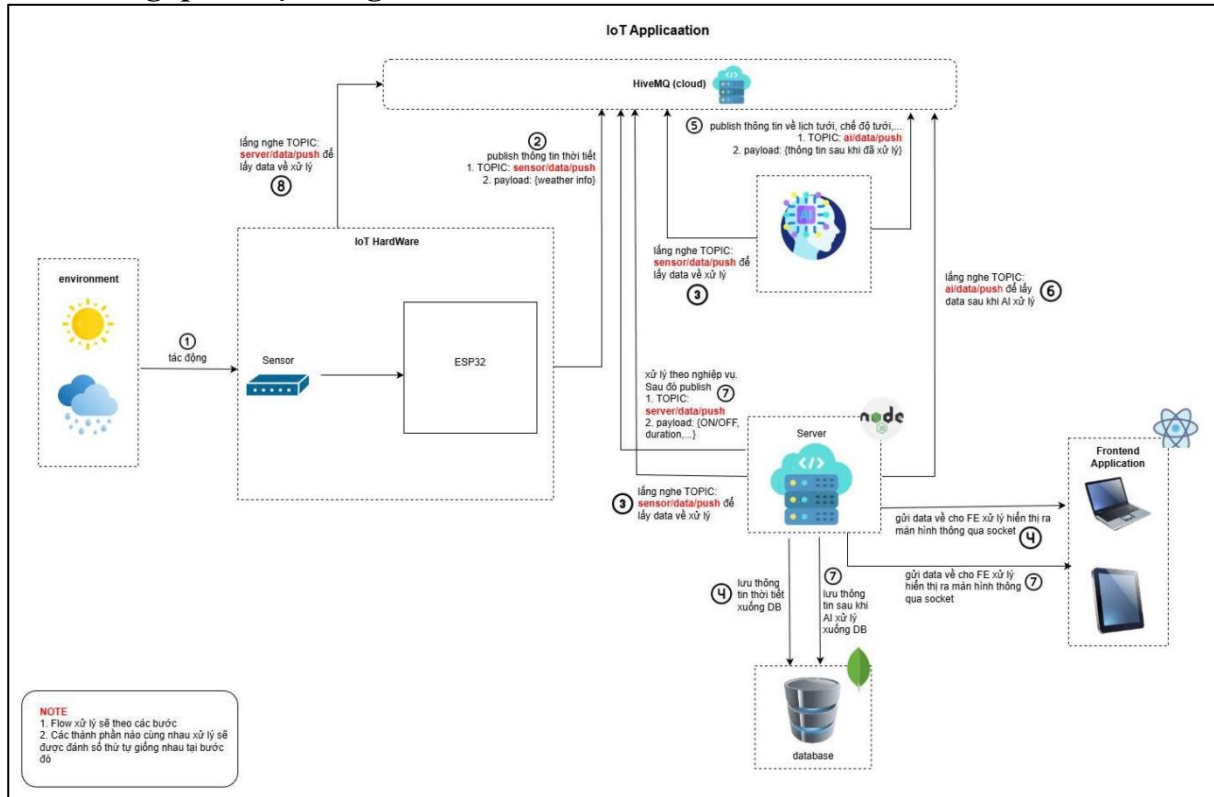
Hệ thống tưới cây thông minh bao gồm 3 phần chính:

- **Phần cảm biến và điều khiển:** ESP32 kết nối với các cảm biến môi trường (DHT22, BME280, cảm biến độ ẩm đất) để thu thập dữ liệu.
- **Phần xử lý dữ liệu và trí tuệ nhân tạo:** Backend (NodeJS + ExpressJS) xử lý dữ liệu cảm biến, AI dự đoán mưa bằng mô hình XGBoost.
- **Phần giao diện người dùng:** Ứng dụng web (ReactJS) hiển thị dữ liệu, lịch

tưới và cho phép điều khiển bơm từ xa

Sơ đồ tổng quan hệ thống

Sơ đồ tổng quan hệ thống:



Hình 1: Sơ đồ tổng quan hệ thống

Chi tiết: Xem tại [đây](#)

Giải thích:

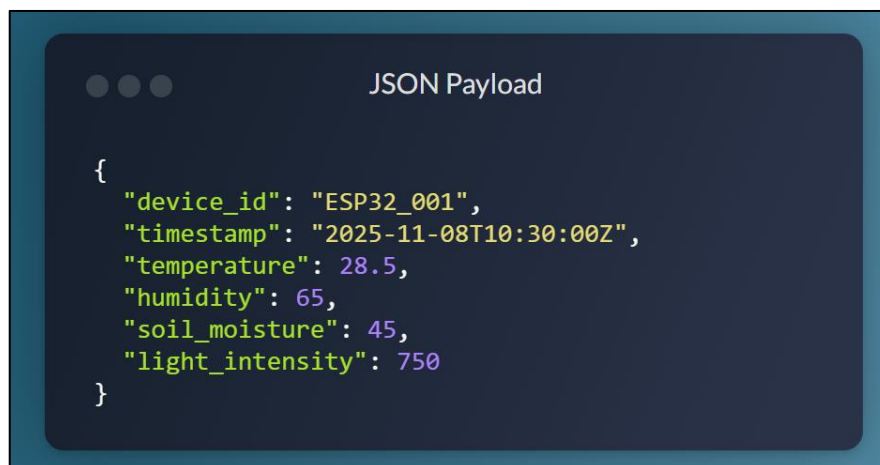
Bước 1: Thu thập dữ liệu từ môi trường

Các cảm biến được lắp đặt trong môi trường sẽ đo đạc và thu thập thông tin:

- Nhiệt độ không khí
- Độ ẩm không khí
- Độ ẩm đất
- Cường độ ánh sáng

Bước 2: ESP32 gửi dữ liệu lên Cloud

1. ESP32 đọc giá trị từ các cảm biến
2. Định dạng dữ liệu thành JSON payload



3. Kết nối tới HiveMQ Cloud qua giao thức MQTT
4. Publish dữ liệu lên TOPIC: `sensor/data/push`

Bước 3: Server và AI đồng thời lắng nghe dữ liệu

Server Backend (NodeJS):

- Subscribe vào TOPIC: `sensor/data/push`
- Vai trò chính:
 - Nhận dữ liệu thời gian thực từ sensor
 - Validate và chuẩn hóa dữ liệu
 - Xử lý logic nghiệp vụ cơ bản
 - Chuẩn bị lưu vào database

AI Modules:

- Subscribe vào TOPIC: `sensor/data/push`
- Vai trò chính:
 - Nhận cùng dữ liệu với Server
 - Phân tích và dự đoán nhu cầu tưới nước
 - Áp dụng các thuật toán machine learning hoặc rule-based system

Bước 4: Server xử lý và phân phối dữ liệu

Server thực hiện 2 nhiệm vụ song song:

A. Lưu trữ dữ liệu

```
Save data

await WeatherData.create({
  deviceId: payload.device_id,
  timestamp: payload.timestamp,
  temperature: payload.temperature,
  humidity: payload.humidity,
  soilMoisture: payload.soil_moisture,
  lightIntensity: payload.light_intensity
});
```

B. Gửi real-time cho Frontend

```
Send data to FrontEnd

// Sử dụng Socket.IO
io.emit('sensor-update', {
  type: 'weather_data',
  data: payload
});
```

Frontend nhận được:

- Cập nhật dashboard real-time
- Hiển thị biểu đồ nhiệt độ, nhiệt độ theo thời gian
- Cảnh báo nếu các thông số vượt ngưỡng

Bước 5: AI xử lý

AI: Sau khi xử lý, publish thông tin về lịch tưới, chế độ tưới,..

1. TOPIC: ai/data/push
2. Payload: {thông tin su khi đã xử lý}

Bước 6: Server nhận data sau khi AI xử lý

Server subscribe TOPIC: ai/data/push để nhận:

1. Lịch tưới tự động
2. Thời lượng tưới
3. Độ tin cậy của AI

Bước 7: Server xử lý và điều khiển

A. Lưu dữ liệu AI vào database

B. Gửi cho Frontend. Frontend sẽ hiển thị

- a. Thông báo lịch tưới sắp tới
- b. Biểu đồ thống kê lượng nước đã dùng
- c. Lịch sử quyết định của AI

C. Xử lý logic nghiệp vụ và điều khiển

Sau đó publish thông tin đã xử lý

1. TOPIC: server/data/push
2. Payload: {thông tin sau khi đã xử lý}

Bước 8: ESP32 nhận lệnh và thực thi

ESP32: Lắng nghe TOPIC: server/data/push để bắt tín hiệu từ server để xử lý bật máy bơm nước,...

2. Môi trường phát triển và hoạt động

a. Môi trường phát triển

- Arduino IDE: Lập trình và nạp chương trình cho ESP32.
- Visual Studio Code: Phát triển frontend (ReactJS) và backend (NodeJS).
- MongoDB Atlas: Cơ sở dữ liệu NoSQL lưu trữ dữ liệu cảm biến.
- Git/Github: Quản lý mã nguồn và làm việc nhóm.

b. Môi trường hoạt động

- Kết nối Wi-Fi nội bộ.
- Trình duyệt web hiển thị giao diện người dùng.
- Máy chủ NodeJS hoặc Cloud lưu trữ và xử lý dữ liệu.

1. Ràng buộc giả định

- Cảm biến hoạt động ổn định với chu kỳ đo 5 phút.
- Hệ thống yêu cầu kết nối Internet liên tục.
- AI cần ít nhất 60 ngày dữ liệu để học và dự đoán chính xác.
- ESP32 có giới hạn tài nguyên nên chỉ đảm nhận nhiệm vụ thu thập và gửi dữ liệu, không chạy mô hình AI trực tiếp.
- Nếu mất kết nối, hệ thống tự chuyển sang chế độ tưới thủ công theo lịch mặc định, hoặc cho phép người dùng điều khiển thủ công/điều khiển từ xa.

2. Yêu cầu chức năng

- Thu thập và gửi dữ liệu cảm biến (nhiệt độ, độ ẩm, áp suất, độ ẩm đất).
- Lưu trữ dữ liệu và phân tích trên server.
- AI dự đoán khả năng mưa và lượng mưa trong 60 phút tới.
- Điều khiển tự động bơm tưới theo ngưỡng độ ẩm và kết quả dự báo.
- Cho phép người dùng theo dõi và điều khiển thủ công qua giao diện web.

- Ghi lại lịch sử hoạt động và nhật ký tưới cây

3. Yêu cầu phi chức năng

- Thời gian phản hồi < 300 ms cho mỗi yêu cầu dự đoán.
- Hệ thống ổn định, tự phục hồi khi kết nối lại.
- Có khả năng mở rộng, thêm cảm biến hoặc khu vực tưới.
- Giao diện thân thiện, dễ sử dụng.
- Đảm bảo bảo mật cơ bản (xác thực truy cập web).

4. Công nghệ và phần cứng sử dụng

4.1. Phần cứng

a. Vi điều khiển ESP32

ESP32-WROOM-32D là một module Wi-Fi + Bluetooth tích hợp, phát triển bởi Espressif Systems, được sử dụng rộng rãi trong IoT (Internet of Things), tự động hóa, các thiết bị nhúng thông minh nhờ khả năng xử lý mạnh mẽ và kết nối không dây. Trên thực tế, người ta thường sử dụng bo mạch phát triển ESP32 DevKit V1 tích hợp sẵn module ESP32-WROOM-32D để dễ dàng lập trình và kết nối.

Các thông số chính:

- CPU: Xtensa® dual-core 32-bit LX6 (tốc độ tối đa 240 MHz).
- Bộ nhớ: 448 KB ROM, 520 KB SRAM, 4 MB Flash (trên module).
- Kết nối: Wi-Fi 802.11 b/g/n, Bluetooth v4.2 (Classic + BLE).
- GPIO: 34 chân I/O, hỗ trợ PWM, ADC (12-bit), DAC (8-bit), SPI, I2C, UART.
- Điện áp hoạt động: 3.0V – 3.6V (DevKit hỗ trợ cấp từ 5V qua cổng USB).

Các linh kiện chính trên bo mạch ESP32 DevKit 32D:

Module ESP32-WROOM-32D:

- Thành phần trung tâm (SoC ESP32 + Flash 4 MB)
- Chứa bộ xử lý, bộ nhớ, Wifi/Bluetooth, và mạch RF

USB-to-UART Bridge (CP2102 hoặc CH340):

- Cho phép nạp chương trình và giao tiếp với máy tính thông qua cổng USB
- Chuyển đổi tín hiệu USB ↔ UART

Ổn áp AMS1117-3.3V:

- Giúp chuyển đổi điện áp từ 5V (USB) hoặc Vin xuống 3.3V để nuôi ESP32

Cổng Micro-USB:

- Dùng để cấp nguồn (5V) và nạp code.

Nút nhấn:

- EN (Reset): Reset lại vi điều khiển
- BOOT (IO0): Giữ để đưa ESP32 vào chế độ nạp chương trình

Thạch anh (Crystal 40 MHz):

- Tạo xung clock ổn định cho chip ESP32

LED chỉ thị (thường nối với GPIO2):

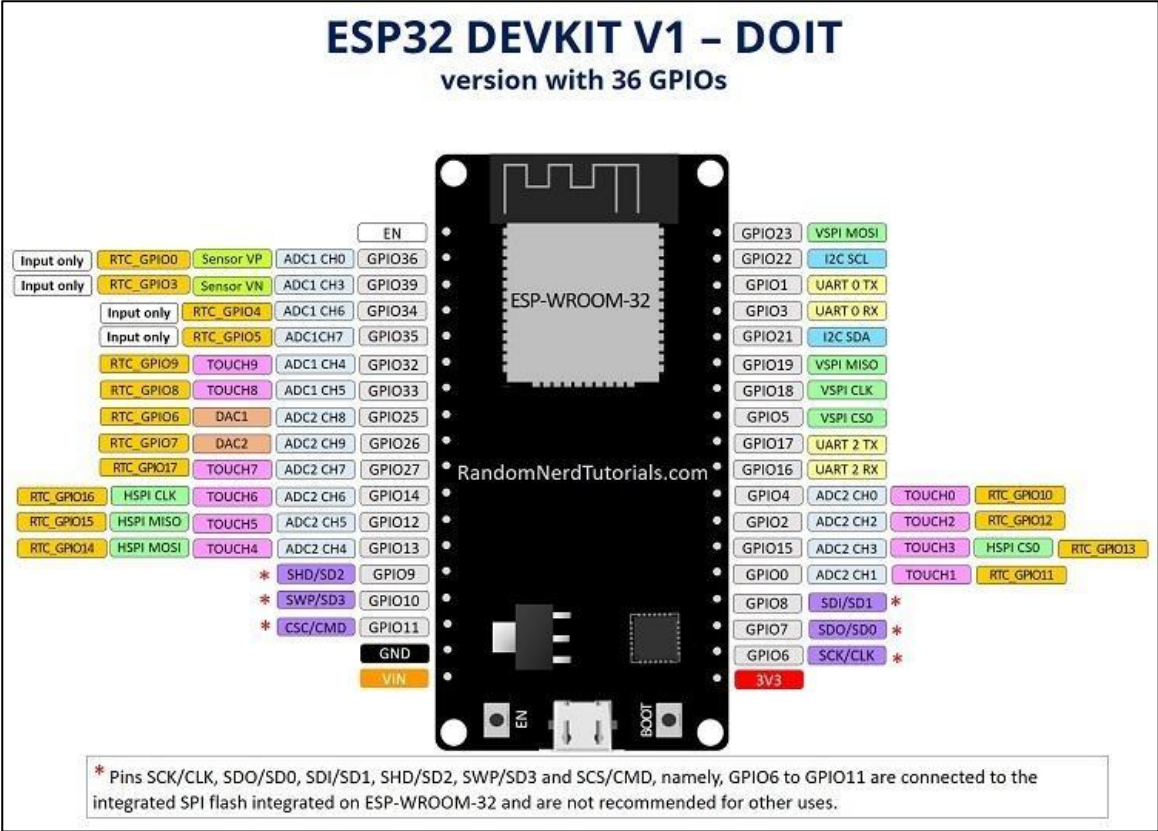
- Báo hiệu nguồn hoặc được dùng test xuất tín hiệu.

Hàng chân header (Male header pins):

- Xuất các chân GPIO, nguồn (3.3V, 5V), GND để dễ dàng kết nối với breadboard hoặc module khác.

Các tụ điện, điện trở dán (SMD):

- Dùng để lọc nhiễu, ổn định nguồn, kéo lên/kéo xuống cho các chân tín hiệu.

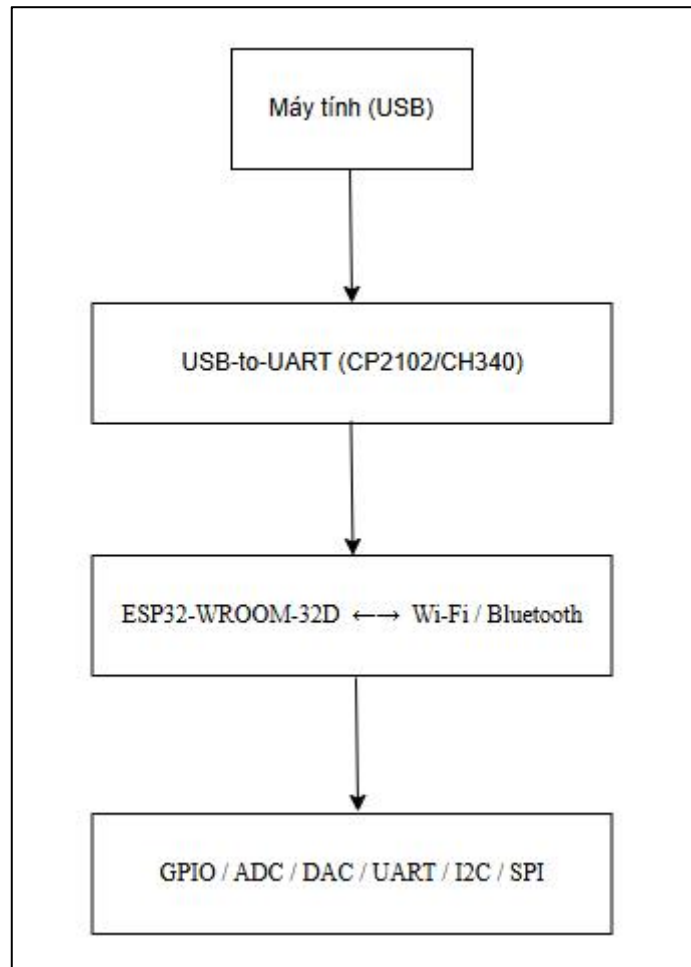


Hình 2: Sơ đồ chân ESP32 DEVKIT ver 36 chân

Chân GPIO	Chức năng chính	Ghi chú đặc biệt
VIN (5V)	Cấp nguồn 5V từ USB hoặc ngoài	
3V3	Nguồn 3.3V từ ổn áp	
GND	Mass (Ground)	
EN	Reset (Enable)	Kéo xuống GND để reset chip
GPIO0	ADC2_CH1, Touch0	Dùng BOOT mode, cẩn thận khi sử dụng
GPIO1 (TX0)	UART0 TX	Dùng cho USB nạp code
GPIO2	ADC2_CH2, Touch2	Ảnh hưởng boot, nên tránh kéo

		xuống GND
GPIO3 (RX0)	UART0 RX	Dùng cho USB nạp code
GPIO4	ADC2_CH0, Touch4	GPIO đa năng
GPIO5	VSPI CS	GPIO đa năng
GPIO6 – 11	Kết nối Flash	Không dùng
GPIO12	ADC2_CH5, Touch5, HSPI MISO	Ảnh hưởng boot, nên tránh khi khởi động
GPIO13	ADC2_CH4, Touch4, HSPI MOSI	
GPIO14	ADC2_CH6, Touch6, HSPI CLK	
GPIO15	ADC_CH3, Touch3, HSPI CS	Ảnh hưởng boot
GPIO16	UART 2 RX	GPIO đa năng
GPIO17	VSPI CLK	GPIO đa năng
GPIO18	VSPI MISO	GPIO đa năng
GPIO19	VSPI MISO	GPIO đa năng
GPIO21	I2C SDA	GPIO đa năng
GPIO22	I2C SCL	GPIO đa năng
GPIO23	VSPI MOSI	GPIO đa năng
GPIO25	ADC2_CH8, DAC1	GPIO đa năng
GPIO26	ADC2_CH9, DAC2	GPIO đa năng
GPIO27	ADC2_CH7, Touch7	GPIO đa năng
GPIO32	ADC1_CH4, Touch9	GPIO đa năng
GPIO33	ADC1_CH5, Touch8	GPIO đa năng
GPIO34	ADC1_CH6	Chỉ Input
GPIO35	ADC1_CH7	Chỉ Input
GPIO36 (VP)	ADC1_CH0	Chỉ Input
GPIO39 (VN)	ADC1_CH3	Chỉ Input

Sơ đồ khối hoạt động cơ bản của DevKit 32D



Hình 3: Sơ đồ khối hoạt động cơ bản của DevKit 32D

b. Cảm biến độ ẩm đất

Giới thiệu chung:

Cảm biến độ ẩm đất là loại cảm biến dùng để đo hàm lượng nước trong đất, được ứng dụng trong nông nghiệp thông minh, hệ thống tưới tự động, giám sát cây trồng. Nguyên lý hoạt động dựa trên việc thay đổi điện trở hoặc điện dung của đất khi hàm lượng nước thay đổi.

Có 2 loại phổ biến:

- Cảm biến độ ẩm đất loại điện trở (Soil Moisture Sensor – Resistive):
 - Đo sự thay đổi điện trở của đất.
 - Rẻ, đơn giản nhưng dễ bị ăn mòn điện cực.
 - Cảm biến độ ẩm đất loại điện dung (Capacitive Soil Moisture Sensor):
 - Đo sự thay đổi hằng số điện môi của đất.
 - Bền hơn, ít bị ăn mòn, kết quả ổn định hơn.
- ⇒ Ở đây sử dụng loại dựa trên điện dung.

Nguyên lý hoạt động:

Khác với các cảm biến độ ẩm đất thông thường dùng phương pháp đo điện trở, cảm biến này hoạt động dựa trên nguyên lý đo điện dung.

- Khi độ ẩm trong đất thay đổi, hằng số điện môi của đất cũng thay đổi.
- Sự thay đổi này làm biến thiên giá trị điện dung, từ đó mạch cảm biến chuyển đổi thành tín hiệu điện áp analog để vi điều khiển có thể đọc và xử lý.

Phương pháp đo điện dung giúp tránh ăn mòn điện cực, nâng cao độ bền và độ ổn định so với loại cảm biến đo điện trở truyền thống.

Sơ đồ chân

Chân	Ký hiệu	Chức năng
VCC	3.3V – 5V	Nguồn cấp
GND	GND	Mass
AO	Analog Output	Xuất tín hiệu analog

Nguyên lý đo lường:

- Module hoạt động theo nguyên lý đo điện dung thay vì đo điện trở như các cảm biến thông thường.
- Vi điều khiển đọc tín hiệu từ chân AO (Analog Output) của cảm biến.
 - Khi đất khô → điện dung giảm → điện áp AO cao.
 - Khi đất ẩm → điện dung tăng → điện áp AO thấp.

Ứng dụng thực tế:

- Hệ thống tưới cây tự động (khi đất khô → bật bơm, khi ẩm → tắt bơm).
- Đo lường và giám sát độ ẩm đất trong nông nghiệp thông minh.
- Kết hợp với ESP32 để đưa dữ liệu lên IoT Cloud (Blynk, MQTT, Firebase...).

Ưu điểm và nhược điểm:

- Sử dụng nguyên lý điện dung, điện cực phủ sơn chống ăn mòn, độ bền cao hơn loại điện trở.
- Giá rẻ, dễ tìm, dễ sử dụng, tương thích với nhiều hệ thống nhúng (3.3–5V).
- Cần hiệu chỉnh ban đầu để có kết quả chính xác.
- Dù bền hơn loại điện trở nhưng vẫn có thể hư hỏng theo thời gian nếu đặt lâu trong môi trường đất ẩm liên tục.

c. Cảm biến môi trường BME280

Giới thiệu chung:

BME280 là cảm biến môi trường tích hợp của hãng Bosch Sensortec, được thiết kế để đo nhiệt độ, độ ẩm và áp suất khí quyển với độ chính xác cao. Đây là phiên bản nâng cấp của BMP280 (chỉ đo nhiệt độ và áp suất).

Ứng dụng trong:

- Trạm thời tiết mini
- Hệ thống IoT giám sát môi trường
- Thiết bị đeo thông minh (smartwatch, fitness tracker)
- Nông nghiệp thông minh, nhà thông minh

Thông số kỹ thuật chính:

Điện áp hoạt động: 1.8V – 3.6V (module breakout thường hỗ trợ 3.3V và 5V).

Dòng tiêu thụ: cực thấp (~3.6 μ A khi đo, ~0.1 μ A ở chế độ sleep).

Đo nhiệt độ:

- Dải: -40°C → +85°C
- Sai số: $\pm 1.0^\circ\text{C}$

Đo độ ẩm:

- Dải: 0% → 100% RH
- Sai số: $\pm 3\%$ RH

Đo áp suất khí quyển:

- Dải: 300 → 1100 hPa

- Sai số: ± 1 hPa
- Giao tiếp: I²C (tối đa 3.4 MHz) hoặc SPI (tối đa 10 MHz).

Cấu tạo và nguyên lý hoạt động:

BME280 tích hợp 3 cảm biến:

- Cảm biến nhiệt độ (Temperature sensor): Dùng để bù sai số cho độ ẩm và áp suất.
- Cảm biến độ ẩm (Humidity sensor – điện dung): Đo sự thay đổi hằng số điện môi của vật liệu nhạy ẩm.
- Cảm biến áp suất (Pressure sensor – áp điện trở màng mỏng): Dựa trên sự thay đổi điện trở khi màng cảm biến biến dạng do áp suất khí quyển.
- Chip BME280 có bộ xử lý tín hiệu tích hợp, dữ liệu được hiệu chỉnh sẵn và xuất ra dạng số (digital) qua giao tiếp I²C hoặc SPI.

Sơ đồ chân:

Chân	Ký hiệu	Chức năng
VCC	3.3V – 5V	Nguồn cấp
GND	GND	Mass
SDA	I ² C data	Dữ liệu (có thể dùng MISO trong SPI)
SCL	I ² C clock	Xung clock (có thể dùng SCK trong SPI)
CS	Chip Select (SPI)	Kéo lên 1 = I ² C, kéo xuống 0 = SPI
SDO	Địa chỉ I ² C (0/1) hoặc MOSI (SPI)	

Thông thường khi dùng ESP32 qua I²C:

- SDA (BME280) → GPIO21 (ESP32)
- SCL (BME280) → GPIO22 (ESP32)

Ưu điểm và nhược điểm:

Ưu điểm	Nhược điểm
<ul style="list-style-type: none"> • Đo được cả nhiệt độ, độ ẩm và áp suất → tiện lợi, tích hợp cao. • Kích thước nhỏ, tiêu thụ điện năng thấp. • Giao tiếp linh hoạt: I²C hoặc SPI. • Độ chính xác cao hơn so với DHT22 và BMP280 	<ul style="list-style-type: none"> • Giá thành cao hơn so với cảm biến đơn chức năng. • Cần thư viện để xử lý dữ liệu (ví dụ Adafruit BME280, SparkFun BME280 cho Arduino/ESP32).

d. Bơm nước tự môi 12V sử dụng động cơ 365/385

Thông số kỹ thuật chính:

- Điện áp hoạt động 9-12V
- Dòng điện:
 - Không tải: ~0.23A
 - Tải làm việc: 0.5-0.7A
- Công suất tiêu thụ: ~6-8W
- Lưu lượng nước tối đa: 2-3 lít/phút
- Áp suất tối đa: 1-2.5kg/cm

- Chiều cao đẩy tối đa: 1-2.5m
- Chiều cao hút tối đa: 1-2m
- Tuổi thọ trung bình: 2-3 năm (trong điều kiện sử dụng bình thường)
- Kích thước tổng thể: 90x40x35 mm
- Đường kính ống nước: 8mm (ngoài)
- Khối lượng: 111g

Nguyên lý hoạt động:

Bơm sử dụng động cơ DC 365/385 gắn với cơ cấu cánh quạt/guồng bơm. Khi cấp điện 9–12V, động cơ quay tạo lực hút chất lỏng từ đường ống vào (IN) và đẩy ra đường ống ra (OUT). Do có khả năng tự mồi, bơm có thể hút nước từ mực thấp hơn (tối đa 1–2 mét) mà không cần đổ nước vào ống trước khi khởi động.\

Ứng dụng trong mạch:

- Tưới cây thông minh: Kết hợp với ESP32 + cảm biến độ ẩm đất, bơm sẽ được điều khiển bật/tắt thông qua MOSFET/Relay tùy vào ngưỡng độ ẩm cài đặt.
- Hệ thống IoT: Bơm có thể kết hợp với cảm biến DHT22, BME280 để kiểm soát môi trường (nhiệt độ, độ ẩm, áp suất).
- Hệ thống làm mát hoặc bơm dung dịch cho các mô hình thí nghiệm.

Ưu điểm và hạn chế:

Ưu điểm	Hạn chế
<ul style="list-style-type: none"> • Kích thước nhỏ gọn, giá thành rẻ. • Lắp đặt dễ dàng, nguồn cấp đơn giản. • Lưu lượng nước đủ cho các mô hình nhỏ và hệ thống tưới cây mini. • Có khả năng tự mồi, tiện lợi khi hút nước từ bể chứa thấp hơn. 	<ul style="list-style-type: none"> • Không phù hợp với chất lỏng đặc hoặc chứa nhiều tạp chất. • Tuổi thọ động cơ có giới hạn (2-3 năm). • Lưu lượng nhỏ, chỉ thích hợp cho quy mô mini/mô hình.

e. Relay 5V 1 kênh

Giới thiệu chung:

Module Relay 5V 1 kênh là mạch đóng/ngắt thiết bị điện bằng tín hiệu điều khiển từ vi điều khiển như ESP32, Arduino, STM32,... Relay hoạt động như một công tắc điện tử, cho phép điều khiển tải AC hoặc DC công suất cao thông qua tín hiệu logic điện áp thấp.

Module thường được sử dụng trong các hệ thống tự động hóa, IoT, điều khiển bơm, quạt, đèn,...

Cấu tạo và thành phần chính:

- Relay 5V: linh kiện cơ điện tử chính, thực hiện đóng/ngắt mạch điện tải.
- Opto cách ly: tách biệt mạch điều khiển và mạch công suất, giúp chống nhiễu và bảo vệ vi điều khiển.
- Transistor điều khiển: khuếch đại dòng để kích relay hoạt động khi nhận tín hiệu logic.
- Điốt bảo vệ (flyback diode): chống xung ngược khi cuộn dây relay nhả, bảo vệ mạch điều khiển.
- LED báo trạng thái: hiển thị khi relay đang bật (ON).

- Chân kết nối IN/OUT: gồm IN (chân điều khiển), VCC, GND và chân COM, NO, NC (kết nối tải).

Thông số kỹ thuật:

- Điện áp hoạt động: 5V DC
- Tín hiệu điều khiển: 3.3V – 5V (tương thích với ESP32, Arduino).
- Dòng kích relay: khoảng 70–90 mA.
- Điện áp tải:
 - AC: 250V – 10A
 - DC: 30V – 10A

Chức năng và nguyên lý hoạt động:

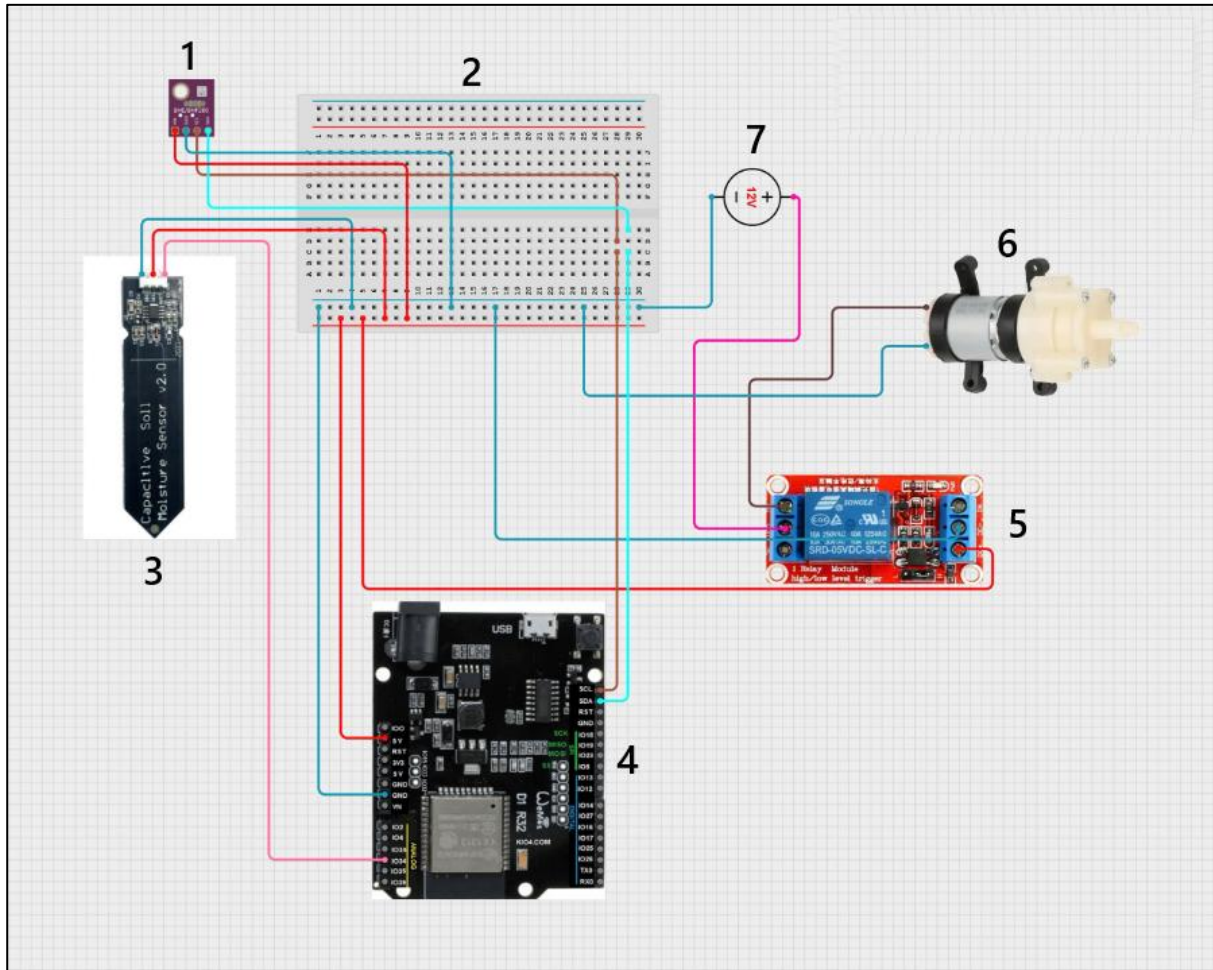
Khi vi điều khiển xuất tín hiệu HIGH (3.3V hoặc 5V) đến chân IN, transistor dẫn → cuộn dây relay được cấp điện → relay đóng tiếp điểm NO → tải được cấp nguồn. Khi tín hiệu LOW (0V), cuộn dây mất điện → relay ngắt mạch, tải ngừng hoạt động. Nhờ opto cách ly, tín hiệu điều khiển được tách biệt khỏi dòng điện tải, giúp mạch điều khiển an toàn và chống nhiễu.

Ứng dụng:

- Hệ thống tưới cây tự động: điều khiển bật/tắt bơm nước.
- Điều khiển thiết bị điện AC/DC: quạt, đèn, motor, đèn LED, máy bơm.
- Dự án IoT: điều khiển từ xa qua ESP32, Blynk, MQTT, Firebase,...
- Tự động hóa gia đình: bật tắt thiết bị điện thông minh theo cảm biến.

Ưu điểm	Hạn chế
<ul style="list-style-type: none"> • Dễ sử dụng, phổ biến, giá rẻ. • Điều khiển được cả tải AC và DC. • Cách ly tốt giữa mạch điều khiển và mạch tải. • Tương thích với tín hiệu 3.3V/5V từ vi điều khiển. 	<ul style="list-style-type: none"> • Đóng/ngắt chậm hơn MOSFET, có tiếng “tạch” khi hoạt động. • Tuổi thọ cơ học giới hạn (do tiếp điểm mòn theo thời gian). • Tiêu thụ dòng kích lớn (~70–90 mA). • Không phù hợp cho ứng dụng cần đóng/ngắt nhanh hoặc PWM.

Sơ đồ nối dây:



Hình 4: Sơ đồ mạch nguyên lý tổng

Chú thích:

1. Cảm biến môi trường BME280
2. Breadboard (Bo mạch test)
3. Cảm biến độ ẩm đất
4. ESP32 Devkit
5. Module Relay 1 kênh (Công tắc điện tử)
6. Máy bơm nước Mini
7. Nguồn 12V

4.2. Phần mềm

a. Server/Client:

- Môi trường phát triển:

- **Arduino IDE:** Được sử dụng để lập trình và nạp chương trình điều khiển và vi điều khiển (ESP32/Arduino). Đây là công cụ chính để viết mã, biên dịch và quản lý kết nối với phần cứng IoT.
- **Visual Studio Code:** Trình soạn thảo mã nguồn hiện đại, hỗ trợ nhiều ngôn ngữ lập trình và có nhiều tiện ích (extension). VS Code được sử dụng để phát triển cả frontend(ReactJS) và backend(NodeJS), đồng thời dễ dàng quản lý dự án với Git.

- Frontend:

- **ReactJS:** Dùng để xây dựng giao diện người dùng (UI), cho phép hiển thị trực quan trạng thái của các thiết bị IoT, đồng thời cung cấp các nút điều khiển để tương tác trực tiếp với hệ thống.
- **Ngôn ngữ:** TypeScript - Ngôn ngữ chính để viết Logic cho giao diện, xử

lý sự kiện và gọi API đến Backend Server.

- **Backend:**

- **NodeJS(22.17.1) với ExpressJS:** NodeJS cung cấp môi trường chạy JavaScript phía Server. ExpressJS là framework nhẹ trên NodeJS, được sử dụng để xây dựng các API RESTful, giúp kết nối frontend với cơ sở dữ liệu và các thiết bị.
- **Ngôn ngữ:** TypeScript: Dùng để viết các dịch vụ Backend, xử lý yêu cầu từ Frontend, giao tiếp với cơ sở dữ liệu, và quản lý luồng dữ liệu từ các thiết bị IoT.

- **Database:**

- **MongoDB Atlas:** Dịch vụ cơ sở dữ liệu NoSQL trên nền tảng đám mây. MongoDB Atlas giúp lưu trữ dữ liệu từ các thiết bị IoT (ví dụ: nhiệt độ, độ ẩm, trạng thái thiết bị), đồng bộ với Backend và hỗ trợ truy xuất nhanh chóng để hiển thị trên frontend.

- **Version Control:**

- **Git (Github Server):** Dùng để quản lý mã nguồn, theo dõi thay đổi và hỗ trợ làm việc nhóm. Github Server đóng vai trò là kho lưu trữ trung tâm, giúp các thành viên dễ dàng chia sẻ, kiểm soát phiên bản, và cộng tác hiệu quả

- **Commucation / Cloud Service:**

- **MQTT Protocol:** Được sử dụng làm giao thức truyền thông giữa các thiết bị IoT và Server. MQTT giúp gửi và nhận dữ liệu cảm biến theo thời gian thực với độ trễ thấp, tiết kiệm băng thông, và hoạt động ổn định trong môi trường mạng yếu
- **HiveMQ Cloud:** Nền tảng MQTT broker trên đám mây, dùng để kết nối và quản lý các thiết bị IoT thông qua giao thức MQTT. HiveMQ Cloud giúp hệ thống có thể mở rộng dễ dàng, đảm bảo tính ổn định và bảo mật khi truyền dữ liệu giữa thiết bị, server và frontend

4.3. AI

1. Bài toán, giải định và tiêu chí đánh giá

1.1. Bài toán

- Đầu vào: Chuỗi đo cảm biến tại vườn theo bước 5': nhiệt độ không khí `temp_c`, độ ẩm không khí `rh-pct`, áp suất khí quyển `pressure-hpa`, độ ẩm đất `soil-moist-pct`, mưa 5 phút `rain_mm_5min` (nếu có); thông tin vận hành tưới (`irrigation_events`); đặc trưng thời gian (giờ, thứ); tùy chọn hỗ trợ từ API thời tiết khu vực.
- Đầu ra:
 - Nowcast nhị phân: xác suất mưa trong 60' tới:
 $p = P(\text{rain_next_60}=1 \mid x) \rightarrow$ quy ra label 0/1 theo ngưỡng.
 - Hồi quy lượng mưa: ước lượng `rain_amount_next_60_mm` (mm/60').

1.2. Giả định

- Tần suất đo ổn định 5'/bản ghi; mỗi device/zone là một chuỗi thời gian riêng.
- Có tối thiểu 30–60 phút lịch sử gần nhất để tạo đặc trưng trễ/rolling.
- Nhấn mưa được suy ra từ `rain_mm_5min` hoặc nguồn ngoại lực đã chuẩn hóa về mm.

1.3. Chỉ số đánh giá (KPI)

- Nowcast: AUC-ROC, PR-AUC, Precision/Recall/F1 tại ngưỡng lựa chọn; ưu tiên Recall (hạn chế bỏ lỡ mưa).

- Hồi quy: RMSE/MAE; ràng buộc đầu ra ≥ 0 .
- Vận hành: giảm $\geq 25-40\%$ lần tưới trùng mưa; độ trễ suy luận < 300 ms/yêu cầu.

2. Thiết kế dữ liệu & gán nhãn

2.1. Lược đồ dữ liệu huấn luyện (tối thiểu)

- sensor_raw_60d.csv : ts(UTC), device_id, temp_c, rh_pct, pressure_hpa, soil_moist_pct, rain_mm_5min
- irrigation_events_60d.csv : device_id, start_ts, end_ts
- labels_rain_60d.csv : ts, device_id, rain_next_60 (0/1), tùy chọn rain_amount_next_60_mm
- external_weather_60d.csv (tùy chọn) : các cột như api_rain_prob_60, api_rain_mm_60, api_cloud, ...

2.2. Quy tắc gán nhãn

- Nhãn nhị phân 60'

$$rain_next_60(t) = 1 \left(\sum_{k=1}^{12} rain_mm_5min(t + 5k') > \tau_{mm} \right)$$

Trong đó τ_{mm} là ngưỡng mm (ví dụ 0.1–0.5 mm), chọn theo thực nghiệm để cân bằng PR-AUC/Recall.

- Hồi quy lượng mưa 60':

$$rain_amount_next_60_mm(t) = \sum_{k=1}^{12} rain_mm_5min(t + 5k')$$

2.3. Làm sạch & đồng bộ thời gian:

- Chuẩn UTC, sort theo (deviceId, ts)
- Kiểm tra trùng (device_id, ts); khử/ghép bản ghi; nội suy bảo thủ nếu đứt nhịp ngắn ($\leq 1-2$ bước)
- Chuẩn kiểu số float32, giá trị bất thường (áp suất/độ ẩm) \rightarrow winsorize/clamp theo IQR.

3. Thiết kế đặc trưng (Feature Engineering)

Được áp dụng riêng theo từng thiết bị/zone theo trật tự thời gian:

- Trễ & trung bình trượt:
 - o *_lag15 = shift(3) với temp-c, rh-pct, pressure-hpa, soil-moist-pct
 - o *_mean30 = rolling(6).mean() (30').
- Biến thiên/gradient:
 - o Pressure-delta15 = pressure-hpa - pressure_hpa.shift(3)
 - o temp_delta15, rh_delta15.
- Dấu hiệu mưa gần: rain_in_last_15m = rolling(3).sum().gt(0)
- Đặc trưng thời gian: hour_of_day, day_of_week.
- Đặc trưng tưới:
 - o is_irrigating_now (1 nếu ts \in [start, end] một sự kiện tưới)
 - o irrig_total_min_last_3h/6h = rolling tổng phút tưới 36/72 bước (5'/bước).
- Ngoại lực (tùy chọn): api_rain_prob_60, api_rain_mm_60, mây gió... sau khi đồng bộ thời gian.

- **Nguyên tắc quan trọng:** Bộ tạo đặc trưng FeatureBuilder dùng chung cho train và infer. Thứ tự cột được cố định, lưu vào models/metadata.json để chống “trôi đặc trưng”.

4. Kiến trúc mô hình

4.1. Lựa chọn thuật toán

XGBoost cho cả nhị phân và hồi quy: ổn định với dữ liệu dạng bảng, tận dụng tốt đặc trưng FE, suy luận nhanh & giải thích được (feature importance). So với LSTM/CNN: XGBoost phù hợp hơn khi mỗi bước thời gian đã được “mã hoá” bởi các cửa sổ trễ/rolling; giảm yêu cầu dữ liệu dài & tài nguyên.

4.2. Hậu đầu mô hình

- Đầu phân loại – xgb-nowcast:
 - o objective = binary:logistic, đầu ra là $p \in (0,1)$.
 - o Mất cân bằng lớp: scale-pos-weight = neg/pos hoặc reweighting.
 - o Tham số khởi tạo khuyến nghị:
 - eta 0.03–0.05, max_depth 5–7, subsample 0.8–0.9, colsample_bytree 0.8–0.9,
 - n-estimators 400–1200, early_stopping_rounds 50–100
- Đầu hồi quy – xgb-amount: Dự đoán mm/60’; tiêu chí RMSE/MAE; clamp đầu ra ≥ 0 để tránh giá trị âm

5. Quy trình huấn luyện & đánh giá

5.1. Chia tập & giữ thuật tự thời gian

Không shuffle; tách train:val/test = 85:15 theo mốc thời gian (hoặc TimeSeriesSplit k-fold). Nếu nhiều thiết bị, đảm bảo mỗi device có mặt ở cả train và test theo dòng thời gian để phản ánh vận hành thật.

5.2. Huấn luyện

- Khởi tạo tham số như 4.2; tính scale_pos_weight từ tỉ lệ lớp.
- Sử dụng early stopping trên tập validation để tránh overfit; lưu best_iteration.
- Lưu mô hình .pkl (joblib) và metadata.json (phiên bản mô hình, danh sách FEATURES, ngưỡng mặc định threshold_default, thông tin chuẩn hoá).

5.3. Tối ưu ngưỡng phân loại

- Theo F1: quét ngưỡng $\in [0,1]$ chọn thr_f1* tối đa F1.
- Ưu tiên Recall: chọn ngưỡng nhỏ nhất sao cho Recall \geq mục tiêu (ví dụ 0.80), rồi trong miền đó lấy ngưỡng có Precision cao nhất.
- Ghi threshold_default vào metadata.json; cho phép chỉnh trên UI.

5.4. Chỉ số & báo cáo

- Nowcast: AUC-ROC, PR-AUC, bảng Precision/Recall/F1 tại các ngưỡng (0.5, thr_f1*, thr_recall*), confusion matrix.
- Hồi quy: RMSE, MAE, biểu đồ residuals; phần trăm dự báo âm (sau clamp = 0).
- Giải thích mô hình: feature importance theo gain/cover; ablation (loại nhóm đặc trưng để đo mức giảm hiệu năng).

6. Suy luận (infer) & ra quyết định tưới

6.1. Suy luận

- Lấy bản ghi gần nhất (hoặc timestamp người dùng chọn) cho từng device_id
- Dùng FeatureBuilder dựng đặc trưng từ lịch sử gần (≥ 30 phút)
- Nowcast: trả $p = \text{predict_proba}(x)$
- Hồi quy (tùy chọn): $mm = \max(0, \text{predict}(x))$

6.2. Chính sách quyết định (mặc định – tinh chỉnh theo vườn)

- Nếu $p \geq 0.70 \rightarrow$ hoãn tưới 60–90 phút
- Nếu $0.50 \leq p < 0.70$ và $\text{soil_moist_pct} \geq$ mục tiêu \rightarrow giảm 30–60% thời lượng/lưu lượng
- Nếu $p < 0.50 \rightarrow$ tưới theo lịch
- Nếu $mm \geq 3$ mm dù p thấp \rightarrow cảnh báo xem xét (mưa giông cục bộ)
- Ghi log: device, ts, p, thr, mm, action để hậu kiểm

7. MLOps cho phần AI (tối thiểu)

- Huấn luyện theo lịch: Cron mỗi tuần; điều kiện kích hoạt nếu có $\geq N$ ngày dữ liệu mới hoặc drift
- Theo dõi drift:
 - o Data drift: phân phối pressure_hpa, rh_pct, temp_c, soil_moist_pct theo tuần/tháng
 - o Prediction drift: trung bình p , tỉ lệ cảnh báo; lệch so với lịch sử
- Hậu kiểm: khi có mưa thật, tính Precision/Recall theo tháng/season
- Quản lý phiên bản: model_version, dataset_hash, feature_schema; giữ ≥ 3 bản để rollback
- Độ tin cậy: timeout suy luận < 300 ms; nếu không đủ lịch sử để dựng FE \rightarrow dùng rule-based an toàn (tưới theo lịch + cảnh báo)

8. Kế hoạch thí nghiệm & đường cơ sở

- Thiết lập baseline:
 - o Đặc trưng: chỉ dùng temp_c, rh_pct, pressure_hpa, soil_moist_pct, rain_mm_5min
 - o Mô hình: Logistic Regression (chuẩn hoá z-score)
 - o Mục tiêu: làm mốc AUC/PR-AUC để so sánh cải tiến
- Bản cải tiến (FE cơ bản)
 - o Thêm lag15, mean30, delta15, rain_in_last_15m, thời gian
 - o Huấn luyện XGBoost; so sánh PR-AUC, Recall@0.5
- Bản cải tiến 2 (tích hợp tưới + ngoại lực):
 - o Thêm is_irrigating_now, rolling tưới 3/6h; api_rain_prob_60
 - o Đo mức tăng điểm và phân tích trọng số đặc trưng
- Ablation & chọn ngưỡng:
 - o So sánh khi bỏ nhóm áp suất/độ ẩm đất/ngoại lực
 - o Chọn threshold-default theo mục tiêu vận hành (Recall cao).

9. Phân tích rủi ro & cách giảm thiểu (riêng phần AI)

- Thiếu nhãn mưa chính xác: giai đoạn đầu có thể dựa vào tổng rain_mm_5min và API; khuyến nghị lắp cảm biến mưa để nâng độ tin cậy.
- Sai lệch FE giữa train và infer: đóng gói FeatureBuilder thành module dùng chung; kiểm thử bằng “golden samples”.
- Mất cân bằng lớp (ít mưa): dùng scale_pos_weight, oversample có kiểm soát, tối ưu ngưỡng theo Recall.
- Chuỗi dữ liệu đứt mạch: rolling với min_periods và cảnh báo mất dữ liệu; bỏ

suy luận nếu thiếu lịch sử.

- Khí hậu theo mùa: tính chỉnh ngưỡng theo tháng/mùa; huấn luyện mô hình theo mùa hoặc thêm one-hot season.

10. Kết quả mong đợi & định hướng mở rộng

- Ngắn hạn: đạt PR-AUC ≥ 0.50 , Recall@thr ≥ 0.75 ; dashboard hiển thị xác suất & quyết định tưới minh bạch; tiết kiệm $\geq 15\text{--}30\%$ nước.
- Trung gian:
 - o Thêm nowcast đa mốc (15/30/60') để ra quyết định linh hoạt theo khung giờ
 - o Tự động lịch tưới tuần dựa trên dự báo lặn, ưu tiên giờ mát/ít gió
 - o Đóng gói ONNX để suy luận nhẹ trên edge (Raspberry Pi/mini-PC)
- Dài hạn: thử nghiệm mô hình lai: XGBoost + encoder chuỗi ngắn (TabNet/Temporal Convolution) khi dữ liệu đủ dài; học chuyển giao giữa các khu vườn

11. Kết luận

Phần AI của hệ thống tập trung vào nowcasting mưa cục bộ với XGBoost và bộ đặc trưng thời gian/hoạt động tưới được thiết kế cẩn thận. Cách tiếp cận này phù hợp bối cảnh IoT nông nghiệp: nhẹ – nhanh – dễ giải thích, đồng thời đủ linh hoạt để mở rộng. Mấu chốt bảo đảm chất lượng là: (i) FE nhất quán giữa train và infer, (ii) tối ưu ngưỡng theo mục tiêu vận hành (ưu tiên Recall), và (iii) MLOps tối thiểu để giám sát drift, tái huấn luyện định kỳ.

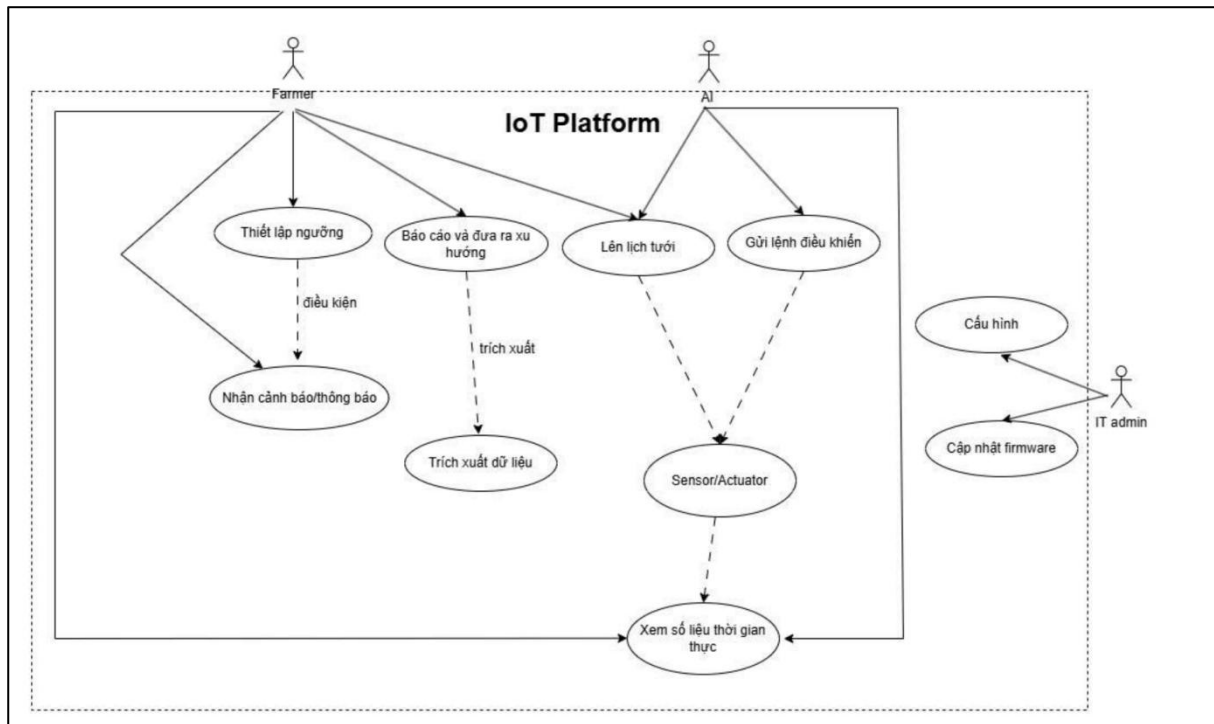
Khi triển khai thực địa vài tuần, mô hình sẽ học “dấu vân tay vi khí hậu” của khu vườn, từ đó cải thiện khả năng phát hiện mưa sớm và tự động hoá quyết định tưới một cách an toàn, tiết kiệm và minh bạch.

III. Biểu diễn chức năng

1. Các tác nhân trong hệ thống

- Người dùng (người nông dân, người quản lý hệ thống)
- Hệ thống tưới thông minh (IoT + AI)

2. Use case tổng quan



Hình 5: Use Case tổng quan

Mô tả Use Case:

1. Các Actos trong hệ thống:

- Farmer:
 - Là người dùng cuối, trực tiếp quản lý và vận hành trang trại
 - Tương tác với hệ thống để theo dõi và đưa ra các quyết định canh tác, bao gồm:
 - Thiết lập ngưỡng hoạt động cho hệ thống (ví dụ: ngưỡng độ ẩm, nhiệt độ)
 - Có thể lên lịch tưới thủ công
 - Xem báo cáo, xu hướng và trích xuất dữ liệu để phân tích
 - Nhận các cảnh báo và thông báo khi có sự kiện bất thường
 - Xem số liệu thời gian thực của các cảm biến gửi lên, qua đó có thể biết được tình trạng của khu vườn
- AI:
 - Là tác nhân logic tự động, chịu trách nhiệm đưa ra các quyết định thông minh
 - Tương tác với các thành phần khác để tối ưu hóa vận hành:
 - Tự động lên lịch tưới dựa trên phân tích dữ liệu
 - Gửi lệnh điều khiển trực tiếp đến các thiết bị chấp hành
 - Xem số liệu thời gian thực để làm đầu vào cho các mô hình dự đoán
- IT Admin:
 - Là người chịu trách nhiệm quản trị, bảo trì và đảm bảo kỹ thuật cho toàn bộ hệ thống
 - Có các quyền cao nhất liên quan đến cấu hình và cập nhật:
 - Cấu hình các thông số hệ thống, tài khoản người dùng
 - Cập nhật firmware cho các thiết bị IoT từ xa

2. Mô tả Use Case

- Thiết lập ngưỡng:
 - Cho phép Farmer định nghĩa các giới hạn hoạt động (ví dụ: độ ẩm

đất quá thấp, nhiệt độ quá cao) để hệ thống tự động cảnh báo

- Báo cáo và đưa ra xu hướng:
 - Hệ thống tự động tổng hợp dữ liệu đã được trích xuất để tạo ra các báo cáo trực quan và phân tích xu hướng, giúp Farmer nắm bắt được tình hình canh tác
- Nhận cảnh báo/thông báo:
 - Hệ thống gửi thông báo tự động đến Farmer khi các chỉ số đo được vượt qua ngưỡng đã thiết lập
- Trích xuất dữ liệu:
 - Cung cấp khả năng lấy dữ liệu lịch sử từ hệ thống để phục vụ cho việc xem báo cáo hoặc mục đích phân tích khác
- Lên lịch tưới:
 - Tác nhân AI phân tích dữ liệu và tự động tạo một lịch trình tưới tiêu tối ưu cho cây trồng
 - Tác nhân Farmer cũng có thể tự lên lịch thủ công tưới tiêu theo ý muốn của mình
- Gửi lệnh điều khiển:
 - Tác nhân AI trực tiếp gửi các lệnh thành công đến các thiết bị như máy bơm, van nước.
 - Tác nhân Farmer cũng có thể điều khiển thiết bị bơm tưới tiêu thủ công
- Cấu hình:
 - Cho phép IT Admin thiết lập các tham số nền tảng, quản lý thiết bị và phân quyền người dùng
- Cập nhật firmware:
 - Cung cấp chức năng cho IT Admin để nâng cấp phần mềm điều khiển trên các thiết bị phần cứng từ xa
- Xem số lượng thời gian thực:
 - Hiển thị các thông số đo được từ cảm biến một cách trực tiếp, phục vụ cho việc giám sát của Farmer hoặc cho AI thu thập dữ liệu để train, tăng khả năng dự đoán đúng trong tương lai

IV.Kế hoạch triển khai

1. Phân chia công việc

a. Công việc chính:

- Làm báo cáo giữa kỳ
- Kiểm tra báo cáo cuối kỳ
- Test sản phẩm sau toàn bộ quá trình xây dựng

b. Phân chia công việc

Mã sinh viên	Họ tên	Nhiệm vụ
B22DCCN482	Trịnh Quang Lâm (Nhóm trưởng)	1. Lập kế hoạch, phân chia nhiệm vụ 2. Thuyết trình 3. Làm báo cáo 4. Code Frontend
B22DCCN434	Vũ Nhân Kiên	1. Làm Slide 2. Thuyết trình 3. Code Backend
B22DCCN422	Cao Thị Thu Hương	1. Làm báo cáo

		2. Code phần cứng 3. Thuyết trình
B22DCCN889	Vũ Thế Văn	1. Code AI 2. Làm báo cáo 3. Thuyết trình

2. Kế hoạch triển khai

Tuần	Nội dung chính	Kết quả dự kiến
Tuần 1 (17/09 – 23/09)	<ul style="list-style-type: none"> Xác định đề tài, mục tiêu, phạm vi hệ thống. Phân công nhiệm vụ trong nhóm. Tìm hiểu các thành phần phần cứng (ESP32, cảm biến DHT22, BME280, Soil Sensor, bơm nước). 	Hoàn thiện đề cương ý tưởng, chọn mô hình cảm biến và bơm
Tuần 2 (24/09 – 01/10)	<ul style="list-style-type: none"> Nghiên cứu nguyên lý hoạt động các cảm biến. Lập sơ đồ nguyên lý và sơ đồ kết nối phần cứng. Chuẩn bị linh kiện, kiểm tra mạch cơ bản 	Sơ đồ kết nối ESP32 với cảm biến và bơm hoạt động ổn định.
Tuần 3 (30/09 – 06/10)	<ul style="list-style-type: none"> Viết chương trình ESP32 đọc dữ liệu cảm biến Gửi dữ liệu qua Serial / Wi-Fi đến server thử nghiệm Kiểm tra tín hiệu từ cảm biến 	ESP32 đọc và gửi được dữ liệu nhiệt độ, độ ẩm, độ ẩm đất.
Tuần 4 (07/10 – 13/10)	<ul style="list-style-type: none"> Xây dựng backend (NodeJS + ExpressJS) Thiết kế API nhận dữ liệu từ ESP32 và lưu MongoDB Kiểm thử API với dữ liệu mẫu 	Backend hoạt động, lưu dữ liệu cảm biến thành công
Tuần 5 (14/10 – 20/10)	<ul style="list-style-type: none"> Phát triển frontend (ReactJS) Hiển thị dữ liệu cảm biến theo thời gian thực Thiết kế giao diện hiển thị độ ẩm, nhiệt độ, trạng thái bơm 	Giao diện Web cơ bản hiển thị dữ liệu IoT
Tuần 6 (21/10 – 27/10)	<ul style="list-style-type: none"> Thiết kế và huấn luyện mô hình AI (XGBoost) dự báo mưa Chuẩn bị dữ liệu huấn luyện, xử lý Feature Mô hình AI huấn luyện được, có thể dự đoán xác suất mưa 	Mô hình AI huấn luyện được, có thể dự đoán xác suất mưa
Tuần 7 (28/10 – 03/11)	<ul style="list-style-type: none"> Tích hợp AI với backend Xây dựng API cho dự đoán thời tiết Thử nghiệm điều kiện tưới tự động dựa trên kết quả AI 	Hệ thống tự động tưới theo dự báo mưa hoạt động
Tuần 8 (04/11 – 10/11)	<ul style="list-style-type: none"> Hoàn thiện giao diện web (hiển thị lịch tưới, biểu đồ, điều khiển thủ công) 	Toàn bộ luồng dữ liệu hoạt động ổn định, giao diện hoàn chỉnh

	- Kiểm thử toàn hệ thống (phần cứng – backend – frontend – AI)	
Tuần 9 (11/11 – 17/11)	- Viết báo cáo cuối kỳ (theo mục P1–P2) - Tổng hợp hình ảnh, kết quả kiểm thử - Chuẩn bị slide thuyết trình	Báo cáo cuối kỳ hoàn chỉnh, slide trình bày sẵn sàng
Tuần 10 (18/11 – 24/11)	- Nộp và trình bày báo cáo cuối kỳ - Thu thập phản hồi từ giảng viên - Lên kế hoạch cải tiến giai đoạn cuối kỳ (bổ sung AI hoặc mở rộng cảm biến)	Báo cáo cuối kỳ được chấm và phản biện xong

V. Kết luận

Hệ thống Tưới cây thông minh ứng dụng AI để lên lịch tưới hàng tuần là một bước tiến nhỏ nhưng có ý nghĩa trong việc ứng dụng IoT và Trí tuệ nhân tạo vào nông nghiệp hiện đại.

Bằng cách kết hợp vi điều khiển ESP32, các cảm biến môi trường (DHT22, BME280, cảm biến độ ẩm đất), bộ điều khiển bơm tự động và mô hình AI dự báo mưa (XGBoost), nhóm đã xây dựng được một giải pháp có khả năng tự động hóa việc tưới tiêu, tiết kiệm nước, và tăng hiệu quả chăm sóc cây trồng