

**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 ứng dụng  
AI để lên lịch tưới**

**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"><li>1. Lên kế hoạch, phân chia nhiệm vụ</li><li>2. Thuyết trình</li><li>3. Viết báo cáo</li><li>4. Code Frontend</li></ol>
B22DCCN434	Vũ Nhân Kiên	<ol style="list-style-type: none"><li>1. Làm Slide</li><li>2. Thuyết trình</li><li>3. Code Backend</li></ol>
B22DCCN889	Vũ Thế Văn	<ol style="list-style-type: none"><li>1. Thuyết trình</li><li>2. Code Module AI</li><li>3. Báo cáo</li></ol>
B22DCCN422	Cao Thị Thu Hương	<ol style="list-style-type: none"><li>1. Code phần cứng</li><li>2. Thuyết trình</li><li>3. Báo cáo</li></ol>

**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 WeMos D1 R32
3	Hình 3	Sơ đồ khối hoạt động cơ bản
4	Hình 4	Sơ đồ nguyên lý
5	Hình 5	Sơ đồ nối dây
6	Hình 6	Sơ đồ Use Case tổng quan
7	Hình 7	Giao diện trang Dashboard
8	Hình 8	Giao diện trang biểu đồ
9	Hình 9	Lịch trình và dự báo của AI
10	Hình 10	Giao diện tưới thủ công
11	Hình 11	Giao diện trang nhật ký hoạt động
12	Hình 12	Giao diện trang xuất báo cáo
13	Hình 13	Giao diện trang cấu hình các ngưỡng
14	Hình 14	Giao diện trang danh sách người dùng hệ thống
15	Hình 15	Giao diện trang thông báo

# Mục lục

I. Giới thiệu đề tài .....	1
1. Mô tả dự án .....	1
2. Mục tiêu và phạm vi của hệ thống .....	1
3. Thu thập yêu cầu của các bên liên quan .....	2
4. Tiêu chí thành công(KPIs) .....	2
5. Kết quả mong đợi .....	3
II. Mô tả tổng quan hệ thống.....	3
1. Mô tả tổng quan .....	3
2. Môi trường phát triển và hoạt động .....	6
III. Biểu diễn chức năng.....	27
1. Các tác nhân trong hệ thống .....	27
2. Use case tổng quan .....	28
IV. Giao diện website .....	29
1. Giao diện Dashboard .....	29
2. Charts .....	30
3. AI Schedule .....	31
4. Manual Control.....	31
5. Logs .....	32
6. Reports .....	32
7. Config .....	32
8. Users .....	33
9. Notification .....	33
V. Kế hoạch triển khai .....	34
1. Phân chia công việc .....	34
2. Kế hoạch triển khai .....	34
VI. Kết luận.....	35

# 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 điều kiện khí hậu 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:
  - o 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.
  - o 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.
  - o 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 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

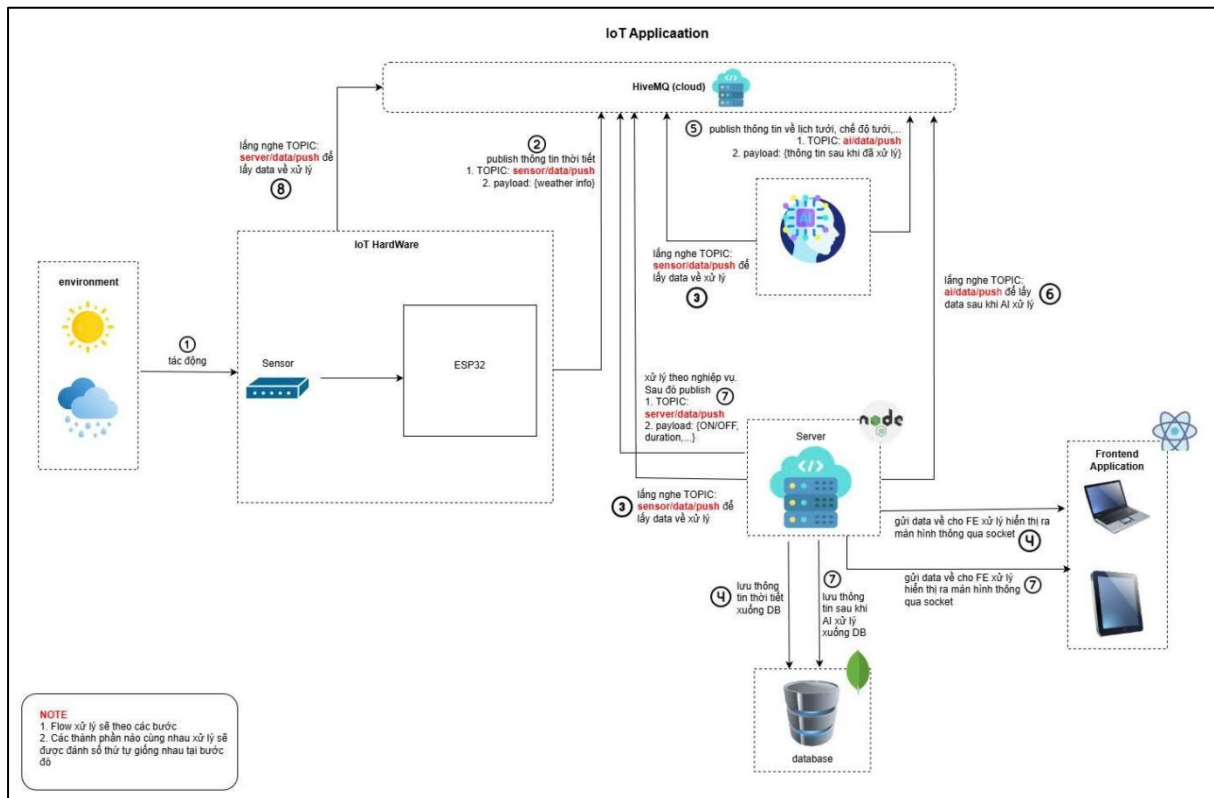
### 1. Mô tả tổng quan

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

```

JSON Payload

{
  "device_id": "ESP32_001",
  "timestamp": "2025-11-08T10:30:00Z",
  "temperature": 28.5,
  "humidity": 65,
  "soil_moisture": 45,
  "light_intensity": 750
}

```

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 Wemos D1 R32 là một bo mạch phát triển dựa trên ESP32-WROOM-32, được thiết kế với bố cục giống Arduino Uno R3, giúp dễ dàng thay thế Arduino bằng ESP32 trong các dự án IoT và Automation. Bo mạch tích hợp đầy đủ Wi-Fi + Bluetooth, xử lý mạnh mẽ, hỗ trợ nhiều giao tiếp và có khả năng lập trình trực tiếp qua cổng USB.

Các thông số chính:

- CPU: Xtensa® dual-core 32-bit LX6 (tối đa 240 MHz)
- Bộ nhớ: 448 KB ROM, 520 KB SRAM
- Flash: 4 MB tích hợp trong module ESP32-WROOM-32
- Kết nối không dây:
  - Wi-Fi 802.11 b/g/n
  - Bluetooth v4.2: Classic + BLE
- GPIO: 34 chân I/O (tùy bo sẽ xuất ra ~25 chân)
  - Hỗ trợ PWM, ADC (12 bit), DAC, SPI, I2C, UART...
- Điện áp hoạt động:
  - Logic mức 3.3V
  - Cấp nguồn qua USB 5V hoặc chân Vin (7–12V)
- Tương thích thiết kế Arduino Uno R3 (cắm shield Arduino trực tiếp).

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

###### Module ESP32-WROOM-32

- Chứa SoC ESP32, bộ nhớ Flash 4 MB, Wifi/Bluetooth và mạch RF.
- Là trung tâm xử lý của toàn bo mạch.

###### USB-to-UART Bridge (CH340)

- Cho phép máy tính nạp chương trình qua cổng USB.
- Chuyển đổi giao tiếp USB ↔ UART.

###### Ổn áp 5V → 3.3V

- Thường dùng AMS1117-3.3 để cung cấp 3.3V ổn định cho ESP32.

###### Cổng Micro-USB

- Dùng để nạp code và cấp nguồn 5V.
- Là cách cấp nguồn chính khi thử nghiệm trên PC.

## Nút nhấn

- EN (Reset): Khởi động lại ESP32
- BOOT (IO0): Giữ để đưa chip vào chế độ nạp firmware thủ công

## Thạch anh 40 MHz

- Tạo xung clock chuẩn cho ESP32.

## LED chỉ thị (thường nối GPIO2)

- Báo trạng thái nguồn hoặc dùng test xuất tín hiệu.

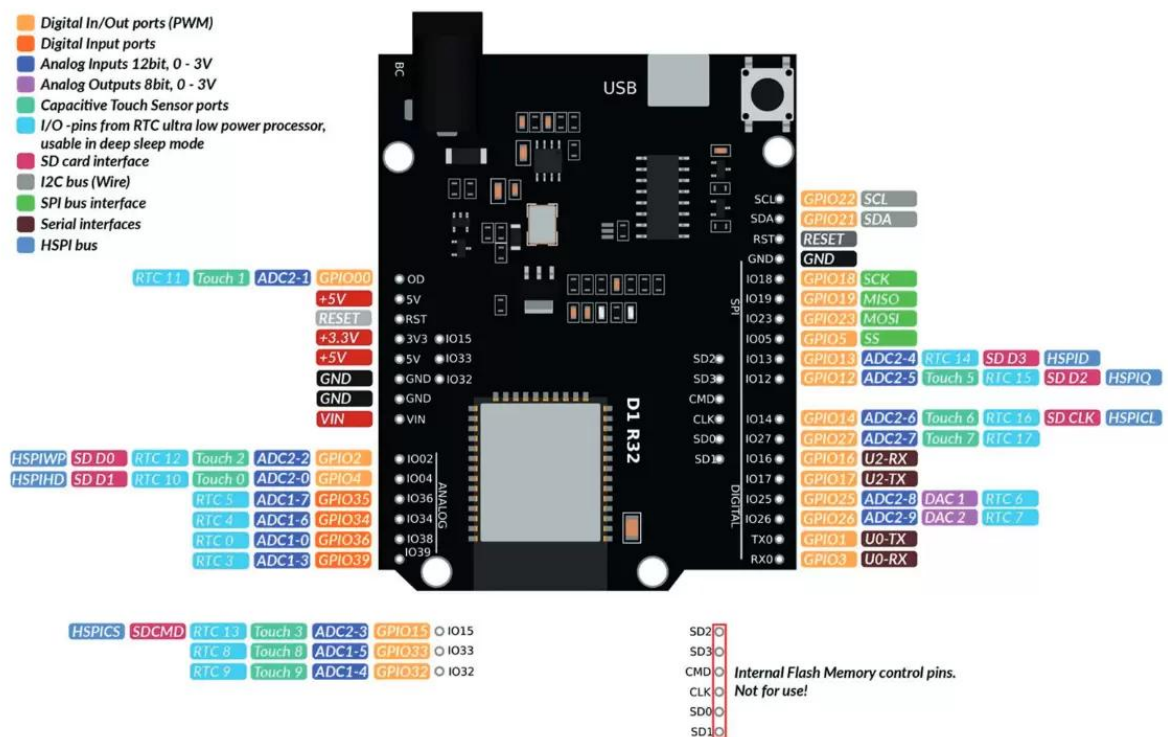
## \*\*Hàng chân Arduino-style headers

- Có bố trí giống Arduino Uno R3.
- Hỗ trợ cắm shield dễ dàng.

## Tụ điện và điện trở SMD

- Lọc nhiễu, ổn định nguồn và hỗ trợ tín hiệu trong bo.

## D1 R32 Board Pinout



Hình 2: Sơ đồ chân ESP32 WeMos D1 R32

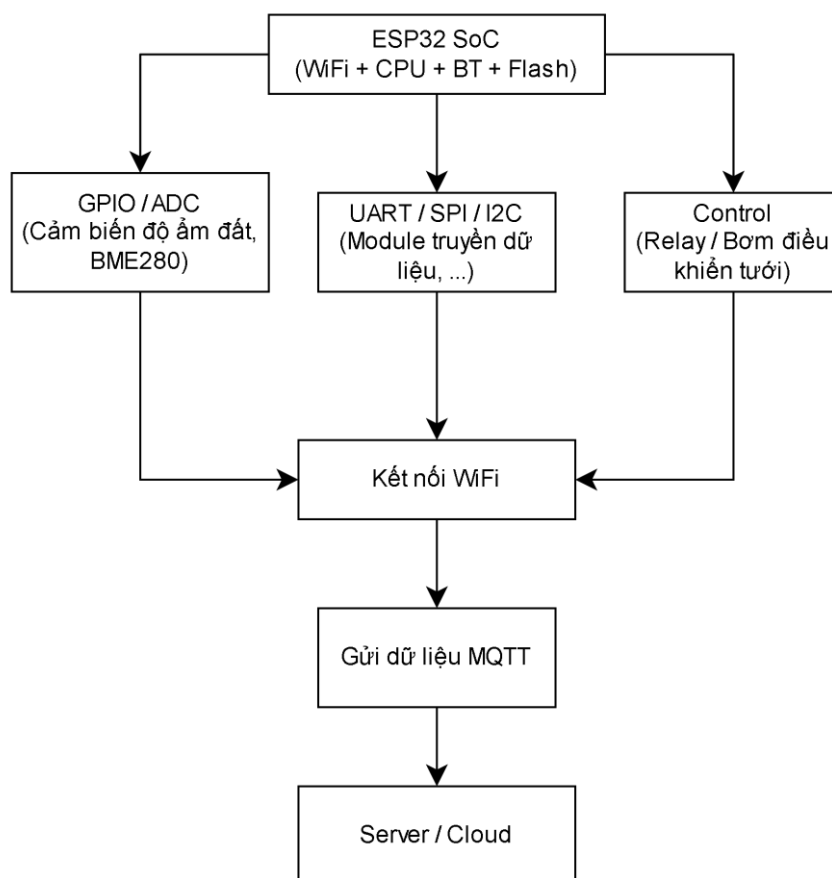
\* Bảng chức năng chân trên bo mạch:

Chân GPIO	Chức năng chính	Ghi chú đặc biệt
<b>Các chân nguồn</b>		
VIN	Cấp 7 - 12V ngoài	Qua ổn áp 5V
5V	Xuất hoặc nhận 5V	Lấy từ USB

3V3	3.3V từ ỏn ỏp on-board	Tỏi ỏa ~500 mA
GND	Mass (Ground)	
<b>Chỏn ỏiều khiển hệ thống</b>		
EN	Reset (Enable)	Kéo xuống GND ỏể reset chip
GPIO0	BOOT, ADC2_CH1	Trỏnh kéo xuống khi boot
GPIO2	ADC2_CH2, LED	Trỏnh kéo xuống khi boot
GPIO15	Boot mode	Trỏnh kéo lên khi boot
<b>UART</b>		
GPIO1	UART0 TX (ỏp code)	
GPIO3	UART0 TX (ỏp code)	
GPIO16	UART2 RX	
GPIO17	UART2 RX	
<b>I2C</b>		
SDA	Tương ỏương với GPIO21	
SCL	Tương ỏương với GPIO22	
<b>SPI</b>		
GPIO18	VSPI SCK	
GPIO19	VSPI MISO	
GPIO23	VSPI MOSI	
GPIO05	VSPI CS	
<b>ADC/DAC</b>		
GPIO25	ADC2_CH8, DAC1	GPIO ỏa năng
GPIO26	ADC2_CH9, DAC2	GPIO ỏa năng
GPIO27	ADC2_CH7	GPIO ỏa năng
GPIO32	ADC1_CH4	GPIO ỏa năng

GPIO33	ADC1_CH5	GPIO đa năng
GPIO34	ADC1_CH6	Chỉ Input
GPIO35	ADC1_CH7	Chỉ Input
GPIO36 (VP)	ADC1_CH0	Chỉ Input
GPIO39	ADC1_CH3	Chỉ Input
<b>Các chân đa năng khác</b>		
GPIO4	ADC2_CH0	
GPIO12	ADC2_CH5	
GPIO13	ADC2_CH4	
GPIO14	ADC2_CH6	

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



Hình 3: Sơ đồ khối hoạt động cơ bản

### 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 trở.

#### Cấu tạo và nguyên lý hoạt động:

Gồm 2 que kim loại cắm xuống đất:

- Khi đất khô → điện trở giữa 2 que cao → dòng điện nhỏ.
- Khi đất ẩm → điện trở giảm → dòng điện lớn.
- Module đi kèm thường có mạch so sánh LM393, xuất tín hiệu Analog và

Digital.

### Sơ đồ chân

Chân	Ký hiệu	Chức năng
VCC	3.3V – 5V	Nguồn cấp
GND	GND	Mass
AO	Analog Output	Xuất tín hiệu analog (0-1023)
DO	Digital Output	Xuất mức logic (0/1) khi vượt ngưỡng trên chiết áp

### Nguyên lý đo lường:

- Vi điều khiển (ví dụ ESP32) đọc tín hiệu từ chân AO (analog).
- Khi đất khô → điện áp AO cao (ít nước → điện dung/điện trở thay đổi).
- Khi đất ẩm → điện áp AO thấp hơn (nhiều nước → điện dung tăng hoặc điện trở giảm).
- Tín hiệu DO chỉ có 2 trạng thái (khô/ướt), phù hợp cho ứng dụng đơn giản (bật/tắt bơm).

### 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:

- Giá rẻ, dễ tìm, dễ sử dụng
- Nhanh bị ăn mòn điện cực → tuổi thọ ngắn

### 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  $\mu$ A khi đo, ~0.1  $\mu$ 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:  $\pm 1$  hPa
- Giao tiếp: I<sup>2</sup>C (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 I<sup>2</sup>C hoặc SPI.

**Sơ đồ chân:**

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

Thông thường khi dùng ESP32 qua I<sup>2</sup>C:

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

**Ưu điểm và nhược điểm:**

Ưu điểm	Nhược điểm
<ul style="list-style-type: none"> <li>• Đo được cả nhiệt độ, độ ẩm và áp suất → tiện lợi, tích hợp cao.</li> <li>• Kích thước nhỏ, tiêu thụ điện năng thấp.</li> <li>• Giao tiếp linh hoạt: I<sup>2</sup>C hoặc SPI.</li> <li>• Độ chính xác cao hơn so với DHT22 và BMP280</li> </ul>	<ul style="list-style-type: none"> <li>• Giá thành cao hơn so với cảm biến đơn chức năng.</li> <li>• Cần thư viện để xử lý dữ liệu (ví dụ Adafruit BME280, SparkFun BME280 cho Arduino/ESP32).</li> </ul>

#### **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:**

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

#### **e. Relay 5V 1 kênh**

##### **Giới thiệu chung:**

Relay 5V 1 kênh là một module đóng/ngắt tải điện bằng tiếp điểm cơ học, được điều khiển bởi tín hiệu mức thấp từ các vi điều khiển như Arduino, ESP32, STM32.... Relay hoạt động như một công tắc điện dùng nam châm điện để đóng/ngắt mạch tải, cho phép điều khiển thiết bị AC và DC công suất lớn bằng các tín hiệu điều khiển nhỏ (5V hoặc 3.3V qua transistor).

##### **Cấu tạo và thành phần chính:**

- Relay cơ 5V: Thành phần chính tạo ra tiếp điểm NO/NC để đóng/ngắt tải AC/DC.
- Opto cách ly: Tách biệt mạch điều khiển với mạch công suất, giúp chống nhiễu và bảo vệ vi điều khiển.
- Transistor điều khiển + diode dập xung: Khuếch đại dòng điều khiển và bảo vệ relay khi sinh ra từ trường ngược.
- LED báo trạng thái: Hiển thị relay đang ON hay OFF.
- Chân IN / VCC / GND: Nhận tín hiệu điều khiển mức thấp từ vi điều khiển.

##### **Thông số kỹ thuật:**

- Điện áp điều khiển: 5V DC.
- Tín hiệu điều khiển IN: mức HIGH/LOW (3.3V – 5V, tùy module).
- Điện áp tải:

- AC: 250V tối đa
- DC: 30V tối đa
- Dòng tải tối đa: 10A (tùy loại relay Songle SRD-05VDC-SL-C hoặc tương đương).
- Loại tiếp điểm: NO (thường hở) / NC (thường đóng).
- Có opto cách ly.

#### **Chức năng và nguyên lý hoạt động:**

Khi vi điều khiển xuất tín hiệu HIGH tới chân IN:

- Opto cách ly kích hoạt transistor.
- Transistor cấp dòng cho cuộn coil relay.
- Nam châm điện hút → tiếp điểm chuyển từ NC sang NO → tải được cấp điện.

Khi tín hiệu LOW:

- Cuộn dây relay ngừng hoạt động → tiếp điểm trở lại trạng thái ban đầu → ngắt tải.

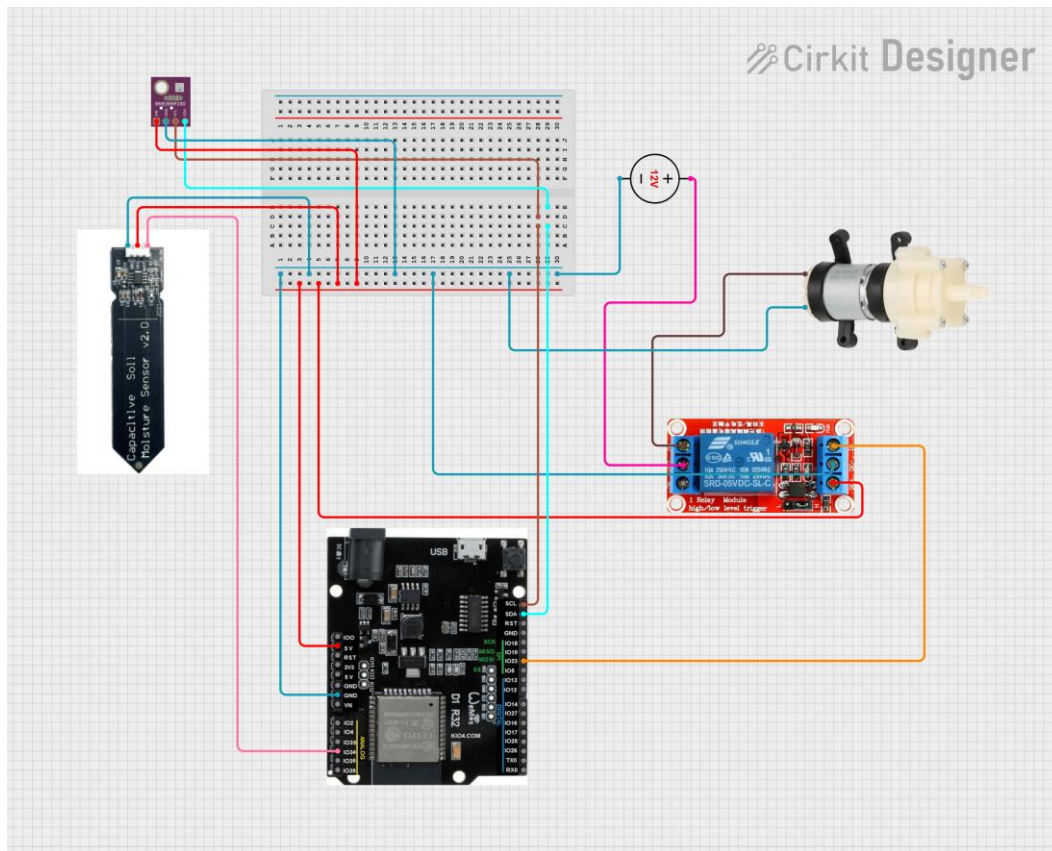
Relay tạo cách ly vật lý giữa mạch điều khiển và mạch công suất, giúp an toàn khi đóng/ngắt các thiết bị AC/DC.

#### **Ứng dụng:**

- Điều khiển thiết bị AC như đèn, quạt, máy bơm, ổ cắm điện.
- Điều khiển tải DC dòng vừa – lớn như motor, đèn LED, khóa điện tử.
- Hệ thống nhà thông minh, IoT kết hợp ESP32/Arduino.
- Các ứng dụng yêu cầu cách ly an toàn giữa tải và vi điều khiển.

<b>Ưu điểm</b>	<b>Hạn chế</b>
<ul style="list-style-type: none"> <li>• Điều khiển được cả tải AC và DC, linh hoạt hơn MOSFET.</li> <li>• Cách ly tốt giữa mạch điều khiển và mạch tải.</li> <li>• Rẻ, phổ biến, dễ sử dụng.</li> <li>• Chịu được điện áp cao (220V AC).</li> </ul>	<ul style="list-style-type: none"> <li>• Tốc độ đóng/ngắt chậm (millisecond), không dùng cho PWM.</li> <li>• Relay là linh kiện cơ học → dễ mòn, tuổi thọ hạn chế.</li> <li>• Gây tiếng tách khi đóng ngắt.</li> <li>• Sinh nhiễu cao áp (cần diode dập xung, opto).</li> <li>• Tiêu thụ dòng cuộn coil cao hơn so với module MOSFET.</li> </ul>

## Sơ đồ nối dây:



Hình 5: Sơ đồ nối dây

### 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 điện

### 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, giả định và tiêu chí đánh giá

##### 1.1. Đặt vấn đề

Hệ thống tưới tiêu tự động truyền thông (sử dụng Timer hoặc cảm biến độ ẩm đơn thuần) thường gặp sai số lớn khi vận hành tại khu vực Hà Nội do các đặc thù khí hậu nhiệt đới gió mùa phức tạp:

1. Sai số không gian (Macro vs Micro Climate): Các dịch vụ dự báo thời tiết (như OpenWeatherMap) cung cấp dữ liệu cho phạm vi rộng (cấp Quận/Thành phố). Tuy nhiên, các cơn mưa rào mùa Hè thường có tính chất cục bộ cao (khu vực vườn không mưa nhưng API báo mưa tại trung tâm quận), dẫn đến quyết định ngắt tưới sai lầm khiến cây thiếu nước.
2. Nhiều tín hiệu khí hậu (Hiện tượng Nồm ẩm): Vào mùa Xuân (tháng 2-4), độ ẩm không khí thường xuyên bão hòa ( $RH > 90\%$ ) kèm sương mù. Các hệ thống tưới dựa trên ngưỡng độ ẩm sẽ hiểu nhầm trạng thái này là "đủ nước" hoặc "đang mưa", trong khi thực tế lượng nước này không thấm vào đất, dẫn đến cây bị stress nước.
3. Hạn chế phần cứng: Việc lắp đặt trạm quan trắc mưa (Rain Gauge) tại hộ gia đình thường tốn kém, công kênh và yêu cầu bảo trì cao để tránh tắc nghẽn.

Bài toán đặt ra: Xây dựng mô hình AI thực hiện Hợp nhất dữ liệu đa nguồn (Multi-source Data Fusion), kết hợp dữ liệu vĩ mô từ API và dữ liệu vi mô từ cảm biến tại

chỗ để "thẩm định" chính xác điều kiện thời tiết thực tế tại vườn, từ đó đưa ra quyết định tưới thích ứng (Adaptive Decision Making) mà không cần cảm biến mưa vật lý.

## 1.2. Định nghĩa Đầu vào (Input)

Hệ thống tiếp nhận vector dữ liệu đa chiều, thực hiện đồng bộ hóa giữa dữ liệu cảm biến thời gian thực (chu kỳ 15 giây) và dữ liệu dự báo từ API (chu kỳ 30 phút). Các trường dữ liệu được lựa chọn kỹ lưỡng để giải quyết bài toán 4 mùa:

### A. Dữ liệu Vĩ mô (Từ OpenWeatherMap API)

Đây là nguồn dữ liệu chính thay thế cảm biến mưa vật lý, cung cấp bức tranh toàn cảnh:

- Dữ liệu Mưa & Thời tiết:
  - pop (Probability of Precipitation): Xác suất mưa dự báo (0-1).
  - rain.1h (Rain Volume): Lượng mưa tích lũy dự báo trong 1 giờ tới (mm).
  - weather.id (Weather Condition Code): Tham số quan trọng nhất. Mã định danh tình trạng thời tiết giúp AI phân loại chính xác hình thái khí hậu:
    - **Nhóm 2xx**: Đông bão (Thunderstorm) -> Cảnh báo nguy hiểm.
    - **Nhóm 5xx**: Mưa (Rain).
    - **Nhóm 7xx**: Khí quyển (Mist/Fog) -> Nhận diện Nồm ẩm để tránh nhầm với mưa.
- Dữ liệu Bốc hơi (Evaporation Factors):
  - uvi (UV Index): Chỉ số tia cực tím. Dùng để tính toán tốc độ thoát hơi nước (ET) vào mùa Hè nắng gắt.
  - dew\_point (Điểm sương): Kết hợp với nhiệt độ để xác định chính xác trạng thái ngưng tụ hơi nước (Nồm) tốt hơn độ ẩm tương đối.
  - wind\_speed: Tốc độ gió (m/s) ảnh hưởng đến tốc độ khô bề mặt đất.

### B. Dữ liệu Vi mô (Từ Cảm biến ESP32)

Dữ liệu thực tế tại vườn dùng để tinh chỉnh sai số của API:

- **Áp suất khí quyển (pressure\_hpa)**: Với chu kỳ 15s, hệ thống có thể vẽ được đồ thị biến thiên áp suất cực kỳ chi tiết, giúp phát hiện sớm các điểm gãy (drops) báo hiệu dông lốc nhanh hơn nhiều so với chu kỳ 5 phút.
- **Nhiệt độ (temp\_c) & Độ ẩm (rh\_pct) từ DHT22**: So sánh chênh lệch với API để đánh giá mức độ sai số cục bộ.
- **Độ ẩm đất (soil\_moist\_pct)**: Tham số điều kiện biên để ra quyết định cuối cùng.

### C. Đặc trưng Thời gian & Vận hành (Context Features)

- Thời gian: Giờ trong ngày (hour), Tháng (month), Phút (minute) để kích hoạt bộ tham số thích ứng theo mùa (Seasonal Thresholds).
- Vận hành: Trạng thái bơm hiện tại và thời gian lần tưới gần nhất.

## 1.3. Định nghĩa Đầu ra (Output)

Mặc dù dữ liệu vào là 15s, nhưng để tránh việc bơm bật/tắt liên tục (gây hỏng thiết bị), AI sẽ trả về kết quả dự báo cho **khung thời gian 60 phút tới**, nhưng quyết định điều khiển được cập nhật (refresh) mỗi 15 giây:

1. Xác suất mưa cục bộ (P\_local\_rain):
  - Là xác suất mưa thực tế tại vườn sau khi AI đã tổng hợp dữ liệu API và xu hướng cảm biến áp suất/độ ẩm.
  - $P(\text{rain} | X_{\text{sensor}}, X_{\text{api}})$  : Quy đổi ra quyết định 0/1.

## 2. Quyết định tưới thông minh (Smart Decision):

- Hệ thống đưa ra hành động cụ thể: TỚI NGAY (Priority High), HOÀN TỚI (Priority Rain), hoặc TỚI BÙ (Low Volume) dựa trên ma trận quyết định.

### 1.4. Các Giả định và Ràng buộc (Assumptions & Constraints)

Để hệ thống hoạt động ổn định trong điều kiện không có cảm biến đo mưa vật lý, các giả định sau được thiết lập:

#### 1. Chiến lược Dữ liệu Lai (Hybrid Strategy):

- Nguồn dữ liệu huấn luyện (Ground Truth): Do không có cảm biến mưa tại chỗ để gán nhãn thực tế, hệ thống sử dụng Dữ liệu lịch sử 03 năm của khu vực Hà Nội (tải từ OpenWeatherMap History Bulk) làm dữ liệu nền tảng để huấn luyện mô hình (Pre-train).
- Giả định về Áp suất: Hệ thống giả định rằng sự sụt giảm áp suất khí quyển cục bộ (pressure\_hpa giảm) kết hợp với cảnh báo từ API là dấu hiệu tin cậy nhất của mưa rào.

#### 2. Giả định về Hiện tượng Nồm:

- Hệ thống mặc định coi các trạng thái có: (Tháng 2-4) AND (Độ ẩm không khí > 90%) AND (Mã weather.id thuộc nhóm 7xx hoặc Áp suất ổn định) là Hiện tượng Nồm. Trong trạng thái này, hệ thống vẫn cho phép tưới nếu đất khô.

#### 3. Ràng buộc kết nối:

- Hệ thống yêu cầu kết nối Internet để gọi API. Trong trường hợp mất kết nối quá 30 phút, AI chuyển sang chế độ Safe Mode (Chỉ tưới dựa trên cảm biến độ ẩm đất, bỏ qua dự báo thời tiết).

### 1.5. Tiêu chí Đánh giá (Evaluation Metrics/KPIs)

Hiệu quả của mô hình AI được đánh giá dựa trên hai nhóm chỉ số:

#### 1. Nhóm chỉ số Kỹ thuật (Model Performance)

- Fusion Accuracy (Độ chính xác hợp nhất): Khả năng AI phát hiện đúng các trường hợp API báo sai (Ví dụ: API báo mưa nhưng AI phủ quyết vì cảm biến áp suất tại chỗ không giảm). Mục tiêu đạt > 80%.
- Precision (Độ chính xác dương tính):
  - **Mục tiêu:** Tối đa hóa Precision vào Mùa Xuân/Đông.
  - **Lý do:** Tránh việc hệ thống nhầm lẫn sương mù/nồm là mưa, dẫn đến quyết định hoãn tưới sai lầm gây thiếu nước cho cây.
- Recall (Độ nhạy):
  - **Mục tiêu:** Tối đa hóa Recall vào Mùa Hè.
  - **Lý do:** Không bỏ sót các cơn mưa rào nhanh (Thunderstorm), đảm bảo tận dụng nước trời và tránh tưới chồng gây úng ngập.

#### 2. Nhóm chỉ số Vận hành (Operational Efficiency)

- Hiệu quả tiết kiệm: Giảm thiểu 25%-30% lượng nước tiêu thụ so với phương pháp tưới theo lịch cố định<sup>3</sup>.
- Độ trễ hệ thống (Latency): Thời gian xử lý toàn trình (Từ lúc đọc Sensor → Gọi API → AI suy luận → Ra lệnh) nhỏ hơn 1000ms.
- Tính ổn định: Hệ thống có cơ chế tự động phục hồi (Auto-recovery) và chuyển đổi chế độ thông minh khi API bị lỗi hoặc mất kết nối mạng.

## 2. THIẾT KẾ DỮ LIỆU VÀ GÁN NHÃN (Cập nhật xử lý tần suất cao)

### 2.1. Lưu trữ dữ liệu

#### 2.1.1. Thách thức về tần suất đa nguồn

Hệ thống phải xử lý sự chênh lệch lớn về tần suất cập nhật:

- **Sensor:** 15 giây/bản ghi -> 4 bản ghi/phút -> 240 bản ghi/giờ.
- **API:** 30 phút/bản ghi -> 2 bản ghi/giờ.

=> **Tỷ lệ:** 1 bản ghi API sẽ được dùng chung cho khoảng 120 bản ghi Sensor liên tiếp.

### 2.1.2. Chiến lược Đồng bộ hóa (Synchronization)

Hệ thống sử dụng phương pháp **Upsampling (Lấy mẫu lên)** cho dữ liệu API:

- Khi ESP32 gửi gói tin lúc 10:00:15, hệ thống truy xuất bản ghi API gần nhất (ví dụ lúc 10:00:00) và ghép vào vector đặc trưng.
- Vector đầu vào tại mỗi bước 15s:

$$X_t = [Sensor_{15s} + API_{last-known} + Time]$$

## 2.2. Kỹ thuật Feature Engineering

Với dữ liệu 15s, rủi ro lớn nhất là **Nhiều (Noise)**. Ví dụ: Cảm biến độ ẩm đất có thể nhảy từ 60% lên 61% rồi về 59% trong 30s do nhiễu điện. Nếu đưa trực tiếp vào AI, kết quả sẽ không ổn định.

Nhóm áp dụng kỹ thuật **Cửa sổ trượt (Rolling Window)** để làm mượt dữ liệu trước khi đưa vào mô hình:

### A. Làm mượt tín hiệu (Signal Smoothing)

Thay vì dùng giá trị tức thời, hệ thống tính trung bình trượt:

- `soil_moist_smooth`: Trung bình của 20 mẫu gần nhất (tương đương 5 phút).

$$SMA_{5min} = \frac{1}{20} \sum_{i=0}^{19} soil\_moist(t - i)$$

- Mục đích: Loại bỏ các gai nhiễu (spikes), giúp AI ra quyết định dựa trên xu hướng thực tế của đất.

### B. Đặc trưng Xu hướng Áp suất (Pressure Trend)

Với tần suất 15s, việc tính toán đạo hàm áp suất (Delta P) trở nên rất chính xác:

- `pressure_slope_1h`: Độ dốc của đường biểu diễn áp suất trong 1 giờ qua (sử dụng 240 điểm dữ liệu quá khứ).
- Nếu đường xu hướng này dốc xuống mạnh (Gradient âm lớn) -> Cảnh báo mưa dông tức thì.

### C. Đặc trưng biến thiên nhanh (Rapid Change Detection)

- `temp_drop_15min`: Độ sụt giảm nhiệt độ trong 15 phút (60 điểm dữ liệu).
  - Nhiệt độ giảm > 3°C trong 15 phút là dấu hiệu của "Front lạnh" hoặc cơn dông mùa hè đang ập đến.

## 2.3. Quy tắc Gán nhãn (Labeling) cho dữ liệu huấn luyện

Do huấn luyện trên dữ liệu lịch sử (thường là theo giờ), ta cần mapping lại với tần suất 15s:

- **Dữ liệu Train (Lịch sử):** Được nội suy (Interpolate) từ 1 giờ xuống 15 giây để khớp với định dạng đầu vào của mô hình.
- **Nhãn Mưa (Y):**
  - Nếu trong dữ liệu lịch sử ghi nhận khung giờ 10:00 - 11:00 có mưa  
→ Toàn bộ 240 bản ghi (15s/bản ghi) trong khung giờ đó sẽ được gán nhãn `Rain = 1`

## 3. Thiết kế đặc trưng (Feature Engineering)

Dữ liệu thô thu thập từ cảm biến (tần suất 15 giây) và API (tần suất 30 phút) chưa thể sử dụng trực tiếp cho mô hình AI do sự chênh lệch về thang đo và nhiễu tín hiệu.

Nhóm phát triển đã thiết kế bộ trích xuất đặc trưng (Feature Extractor) để chuyển đổi dữ liệu thô thành các chỉ báo khí tượng có ý nghĩa vật lý.



### 3.1. Nguyên lý xử lý dữ liệu tần suất cao (High-Frequency Processing)

Với tần suất lấy mẫu  $\Delta t = 15s$ , hệ thống có khả năng nắm bắt được đạo hàm (tốc độ thay đổi) của các thông số môi trường. Đây là lợi thế lớn so với các hệ thống lấy mẫu thưa (5-10 phút).

Hệ thống áp dụng cửa sổ trượt (Rolling Window) để tính toán các đặc trưng xu hướng:

- Cửa sổ ngắn ( $W_{short} = 15m$ ): 60 mẫu dữ liệu. Dùng để phát hiện biến động tức thời (gió giật, giảm nhiệt nhanh).
- Cửa sổ dài ( $W_{long} = 1h$ ): 240 mẫu dữ liệu. Dùng để xác định xu hướng bền vững (áp suất giảm báo bão).

### 3.2. Nhóm đặc trưng Xu hướng Cảm biến (Sensor Trend Features)

Đây là nhóm đặc trưng quan trọng nhất để AI tự động phát hiện mưa rào cục bộ (Local Thunderstorm) mà API có thể bỏ sót.

#### A. Độ dốc Áp suất (pressure\_slope\_1h)

Sự thay đổi áp suất là chỉ báo độc lập và tin cậy nhất của thời tiết xấu.

- Công thức: Tính hệ số góc (Slope) của đường hồi quy tuyến tính trên 240 điểm dữ liệu áp suất gần nhất (1 giờ).
- Ý nghĩa vật lý:
  - slope < -1.0 hPa/h: Cảnh báo áp suất tụt nhanh -> Dấu hiệu mưa dông/gió giật.
  - slope xấp xỉ 0: Thời tiết ổn định (Dù độ ẩm có thể cao do Nồm).

#### B. Tốc độ biến thiên Nhiệt - Ẩm (temp\_velocity, rh\_velocity)

- Công thức:  $\Delta T = T_t - T_{(t-15min)}$
- Ý nghĩa vật lý: Trước các cơn mưa rào mùa Hè (Summer Squall), nhiệt độ thường giảm đột ngột (do gió lạnh tràn về) và độ ẩm tăng vọt trong thời gian ngắn. Dữ liệu 15s cho phép phát hiện "cú rơi" nhiệt độ này ngay khi nó vừa bắt đầu.

#### C. Độ ổn định của Đất (soil\_stability\_index)

- Công thức: Phương sai (Variance) của độ ẩm đất trong cửa sổ 5 phút.
- Ý nghĩa: Giúp loại bỏ nhiễu. Nếu độ ẩm đất biến thiên quá nhanh (Variance cao) mà không có lệnh tưới hay mưa, hệ thống coi đó là nhiễu cảm biến và loại bỏ.

### 3.3. Nhóm đặc trưng Hợp nhất (Fusion Features)

Nhóm đặc trưng này giúp AI so sánh giữa "Dự báo" (API) và "Thực tế" (Sensor) để đánh giá độ tin cậy.

#### A. Độ lệch chuẩn hóa API-Sensor (bias\_temp, bias\_rh)

- Công thức:
$$Bias = |Value_{sensor} - Value_{api}|$$
- Ý nghĩa:
  - Nếu Bias nhỏ: API đang dự báo đúng vì khí hậu -> Tăng trọng số tin cậy vào api\_rain\_prob.
  - Nếu Bias lớn (VD: API báo 30°C, Sensor báo 35°C): Có hiện tượng đảo nhiệt đô thị hoặc sai số địa lý -> Giảm trọng số API, ưu tiên dữ liệu Sensor.

#### B. Chỉ số Nồm ẩm (nom\_index)

Để giải quyết bài toán sương mù/nồm ẩm (độ ẩm cao nhưng không phải mưa), hệ thống tính toán Điểm sương (Dew Point) từ nhiệt độ và độ ẩm sensor.

- Công thức: Sử dụng xấp xỉ Magnus để tính  $T_{dew}$  (Dew Point).

$$Diff_{dew} = T_{air} - T_{dew}$$

- Quy tắc:
  - Nếu  $\text{Diff\_dew} \approx 0$  (Không khí bão hòa) và  $\text{pressure\_slope} \approx 0$  (Áp suất ổn định)  $\rightarrow$  Khả năng cao là Sương mù/Nồm.
  - Nếu  $\text{Diff\_dew} \approx 0$  và  $\text{pressure\_slope} < 0 \rightarrow$  Khả năng cao là Mưa.

### 3.4. Nhóm đặc trưng Mùa vụ (Temporal & Seasonal Features)

Dữ liệu thời gian được mã hóa để mô hình học được tính chu kỳ của khí hậu 4 mùa.

#### A. Mã hóa chu kỳ (Cyclical Encoding)

Thay vì dùng số nguyên (1-12, 0-23), hệ thống chuyển đổi sang dạng tọa độ cực:

- $\text{month\_sin} = \sin(2\pi * \text{month}/12)$
- $\text{month\_cos} = \cos(2\pi * \text{month}/12)$
- Ý nghĩa: Giúp mô hình hiểu rằng Tháng 12 (Mùa Đông) và Tháng 1 (Mùa Xuân) nằm cạnh nhau về mặt tính chất khí hậu, thay vì cách xa nhau về mặt số học (1 và 12).

#### B. Khung giờ mưa nhiệt (hour\_feature)

- Mã hóa giờ trong ngày để AI học được quy luật: Mùa hè thường mưa rào vào chiều tối (16h-18h), Mùa đông thường mưa phùn vào đêm/sáng sớm.

### 3.5. Bảng tổng hợp Vector đặc trưng (Feature Vector Summary)

Sau quá trình Feature Engineering, mỗi điểm dữ liệu đầu vào cho mô hình AI là một vector gồm 14 chiều, được cập nhật mỗi 15 giây:

Nhóm	Tên đặc trưng	Mô tả kỹ thuật	Vai trò trong mô hình
API	api_pop	Xác suất mưa (0-1)	Tham số cơ sở (Baseline).
	api_rain_1h	Lượng mưa dự báo (mm)	Định lượng cường độ.
Sensor Trend	pressure_slope_1h	Đạo hàm áp suất trong 1h	Quan trọng nhất (Phát hiện dông).
	temp_drop_15m	Độ giảm nhiệt trong 15p	Phát hiện front lạnh/mưa rào.
	rh_rise_15m	Độ tăng ẩm trong 15p	Phát hiện luồng ẩm.
Fusion	dew_point_diff	Chênh lệch điểm sương	Phát hiện Nồm ẩm.
	temp_bias	Sai số nhiệt độ API/Sensor	Đánh giá độ tin cậy API.
Context	soil_moist_smooth	Độ ẩm đất (Làm mượt)	Điều kiện biên ra quyết định.
Time	month_sin, month_cos	Mã hóa tháng	Thích ứng 4 mùa.
	hour_sin, hour_cos	Mã hóa giờ	Thích ứng chu kỳ ngày/đêm.
	uvi_index	Chỉ số UV (từ API)	Tính toán bốc hơi mùa Hè.

## 4. Quy trình Huấn luyện và Chiến lược xử lý Đa tần suất (Multi-frequency Handling)

### 4.1. Thách thức về sự lệch pha dữ liệu

Hệ thống đối mặt với sự bất đồng bộ lớn về tần suất dữ liệu:

- Dữ liệu Huấn luyện (Training Data): Dựa trên lịch sử OpenWeatherMap với tần suất thưa (1 giờ/bản ghi).

- Dữ liệu Suy diễn (Inference Data): Dựa trên cảm biến thời gian thực với tần suất dày đặc (15 giây/bản ghi).

Nếu nhập trực tiếp dữ liệu thô vào mô hình, AI sẽ hoạt động sai lệch do không tương thích về ngữ nghĩa thời gian.

#### 4.2. Giải pháp: Đồng bộ hóa dựa trên Đặc trưng (Feature-based Alignment)

Thay vì cố gắng nội suy (Upsample) dữ liệu lịch sử không lỗ hay làm chậm hệ thống thực tế (Downsample), nhóm áp dụng chiến lược Chuẩn hóa Đặc trưng theo Cửa sổ trượt (Rolling Window Normalization).

Nguyên tắc cốt lõi: "Mô hình không học giá trị tức thời, mô hình học xu hướng trong một khoảng thời gian cố định (W)."

Quy trình thực hiện:

1. Bước 1: Thiết kế đặc trưng "Bất biến theo tần suất" (Frequency-invariant Features)

Hệ thống chuyển đổi các giá trị đo lường thô thành các giá trị thống kê trong cửa sổ thời gian 1 giờ:

- Thay vì dùng Pressure<sub>t</sub>, ta dùng Delta P<sub>1h</sub> = P<sub>t</sub> - P<sub>(t-1h)</sub>.
- Thay vì dùng Temp<sub>t</sub>, ta dùng Temp\_avg<sub>1h</sub>.

2. Bước 2: Xử lý lúc Huấn luyện (Training Phase - Offline)

Với dữ liệu lịch sử (1h/lần):

- Tính toán Delta P<sub>1h</sub> bằng cách lấy dòng hiện tại trừ dòng trước đó (Lag-1).
- Gán nhãn Y: Mưa trong 1 giờ tới.

3. Bước 3: Xử lý lúc Vận hành (Inference Phase - Online)

Với dữ liệu Sensor (15s/lần), hệ thống duy trì một hàng đợi (Buffer Queue) chứa 240 bản ghi gần nhất (tương đương 1 giờ):

- Tại thời điểm T (bất kỳ giây thứ 15 nào), hệ thống tính:  

$$\Delta P_{1h} = \text{Sensor\_Pressure}(T) - \text{Sensor\_Pressure}(T - 3600s)$$
- Giá trị này được đưa vào mô hình đã train.

Kết quả: Mô hình AI đang nhận dữ liệu theo giờ như lúc học, nhưng thực tế nó đang đánh giá lại tình hình mỗi 15 giây. Điều này giúp hệ thống giữ được độ chính xác của dữ liệu lịch sử nhưng vẫn đạt được tốc độ phản ứng Real-time của IoT.

## 5. Kiến trúc Mô hình và Cấu hình Thuật toán

### 5.1. Lựa chọn thuật toán: XGBoost Classifier

Để giải quyết bài toán phân loại nhị phân (Mưa/Không mưa) trên dữ liệu dạng bảng không đồng nhất (kết hợp giữa xác suất của API và chỉ số vật lý của cảm biến), nhóm phát triển lựa chọn thuật toán XGBoost (Extreme Gradient Boosting).

Lý do lựa chọn so với các mô hình khác:

- Khả năng xử lý nhiễu: XGBoost tích hợp sẵn cơ chế Regularization (L1/L2), giúp mô hình không bị "học vẹt" (Overfitting) khi dữ liệu cảm biến tần suất cao (15s) có nhiều nhiễu động.
- Tính giải thích (Explainability): Khác với Neural Network, XGBoost cho phép trích xuất mức độ quan trọng của đặc trưng (Feature Importance). Điều này giúp nhóm kiểm chứng được giả thuyết: **"Liệu mô hình có thực sự dựa vào sự giảm áp suất để dự báo mưa hay không?"**.
- Hiệu năng: Tốc độ suy diễn (Inference) cực nhanh (< 50ms), phù hợp chạy trên các máy chủ cấu hình thấp hoặc Gateway IoT.

### 5.2. Cấu hình Hyperparameters (Tham số mô hình)

Mô hình được tối ưu hóa với bộ tham số sau để phù hợp với đặc thù dữ liệu mất cân bằng (Số giờ không mưa luôn nhiều hơn số giờ có mưa):

Tham số	Giá trị	Giải thích ý nghĩa kỹ thuật
objective	binary:logistic	Hàm mục tiêu cho bài toán phân loại nhị phân (0/1).
eval_metric	auc	Sử dụng diện tích dưới đường cong ROC để đánh giá, thay vì Accuracy (do dữ liệu mất cân bằng).
scale_pos_weight	8.5	Quan trọng: Tăng trọng số cho lớp "Mưa" (thiểu số). Giúp mô hình không bỏ sót các cơn mưa rào nhanh dù chúng ít xuất hiện trong dữ liệu lịch sử.
max_depth	6	Độ sâu cây quyết định vừa phải để tránh Overfitting.
learning_rate	0.05	Tốc độ học chậm để mô hình hội tụ ổn định.
subsample	0.8	Sử dụng 80% dữ liệu ngẫu nhiên cho mỗi cây để tăng tính tổng quát.

6. Chiến lược Ra quyết định và Thích ứng Mùa (Decision Logic)

Đây là tầng xử lý nghiệp vụ (Business Logic Layer) nằm sau mô hình AI, chịu trách nhiệm chuyển đổi xác suất dự báo thành tín hiệu điều khiển bơm máy.

6.1. Logic Thích ứng 4 Mùa (Seasonal Adaptation)

Hệ thống tự động điều chỉnh "độ nhạy" của quyết định dựa trên tháng trong năm:

A. Mùa Xuân (Tháng 2 - Tháng 4): Chế độ "Anti-Nồm"

- Vấn đề: Độ ẩm không khí bão hòa (>95%), sương mù, cảm biến ẩm báo ướt nhưng đất có thể khô.
- Logic điều chỉnh:
  - Hạ thấp trọng số của đặc trưng sensor\_humidity.
  - Nâng cao ngưỡng quyết định mưa (Threshold\_rain = 0.8). Chỉ hoãn tưới khi xác suất mưa từ API cực cao hoặc áp suất giảm sâu.
  - Quy tắc cứng: Nếu dew\_point\_diff xấp xỉ 0 (Nồm) và soil\_moisture < Ngưỡng tưới -> VẪN TƯỚI.

B. Mùa Hè (Tháng 5 - Tháng 7): Chế độ "Phản ứng nhanh"

- Vấn đề: Nắng gắt, bốc hơi nhanh (ET cao), mưa rào đến nhanh.
- Logic điều chỉnh:
  - Giảm ngưỡng quyết định mưa (Threshold\_rain = 0.5). Chấp nhận rủi ro dự báo sai (False Positive) để ưu tiên giữ ẩm cho cây.
  - Kích hoạt giám sát Đạo hàm áp suất (pressure\_slope): Nếu áp suất tụt > 2hPa/30phút -> Kích hoạt ngắt bơm khẩn cấp (Emergency Stop) bất kể dự báo API.

C. Mùa Thu & Đông: Chế độ "Tiết kiệm"

- Logic: Ưu tiên tiết kiệm nước. Ngưỡng quyết định Threshold\_rain = 0.4. Chỉ cần có dấu hiệu mưa nhỏ là hoãn tưới.

6.2. Ma trận Quyết định Hợp nhất (Fusion Decision Matrix)

Bảng chân lý (Truth Table) cho quyết định cuối cùng tại mỗi chu kỳ 15s:

Dự báo API (Vĩ mô)	Xu hướng Cảm biến (Vi mô)	Trạng thái Đất	Quyết định Hệ thống	Giải thích
Mưa (>70%)	Áp suất giảm, Ẩm tăng	Bất kỳ	HOÃN TƯỚI	Dự báo API và Cảm biến đồng thuận.
Mưa (>70%)	Áp suất ổn định, Nhiệt cao	Khô	TƯỚI BÙ (50%)	API có thể báo sai (mưa vùng khác). Tưới lượng nhỏ để thăm dò.
Không mưa	Áp suất ổn định	Khô	TƯỚI NGAY	Hoạt động bình thường.
Mưa nhỏ/Phùn	Độ ẩm > 90% (Nồm)	Khô	TƯỚI NGAY	AI lọc bỏ nhiễu thời tiết Nồm ẩm.
Bất kỳ	Áp suất tụt dốc đứng	Bất kỳ	NGẮT BƠM	Cảnh báo đông tắc nguy hiểm. Bảo vệ thiết bị.

### 6.3. Thuật toán Lập lịch tưới Đa khung thời gian (Multi-horizon Scheduling)

Khác với cơ chế điều khiển thời gian thực (Real-time Control) hoạt động với tần suất 15 giây để phản ứng với các biến động tức thời, module Lập lịch (Scheduler) hoạt động với chu kỳ 4 giờ/lần để xây dựng kế hoạch tưới dài hạn, giúp người dùng chủ động quản lý nguồn nước.

Hệ thống phân chia thành 3 khung thời gian dự báo với độ chính xác và mục tiêu khác nhau:

#### 6.3.1. Cơ sở lý thuyết: Mô hình Cân bằng nước

Để dự đoán độ ẩm đất trong tương lai (khi chưa có số liệu cảm biến), hệ thống sử dụng phương trình cân bằng nước:

$$Soil_{t+1} = Soil_t + (P_{eff} + I) - ET_c$$

Trong đó:

- $Soil_t$ : Độ ẩm đất hiện tại (Lấy từ cảm biến).
- $P_{eff}$ : Lượng mưa hiệu quả dự báo từ API (Đã trừ đi lượng nước chảy tràn).
- $I$ : Lượng nước tưới dự kiến (Biến số cần tìm).
- $ET_c$ : Lượng nước thoát hơi của cây trồng (Tính toán từ Nhiệt độ và UV index của API).

#### 6.3.2. Chiến lược lập lịch cho từng khung thời gian

##### A. Khung Ngắn hạn (1-2 Ngày tới) - **Độ chính xác: Cao**

Đây là lịch trình có tính Cam kết thực thi. Hệ thống sẽ bám sát lịch này trừ khi có sự kiện thời tiết bất thường (Emergency Override).

- Dữ liệu đầu vào: API dự báo theo giờ (Hourly Forecast).
- Logic AI xử lý:
  1. Duyệt từng khung giờ (ví dụ: 08:00, 17:00).
  2. Kiểm tra xác suất mưa ( $P_{rain}$ ) và chỉ số Nồm ẩm (như đã định nghĩa ở mục 6.1).
  3. Nếu  $P_{rain} > \text{Ngưỡng}$  và không phải Nồm -> Đánh dấu "HOÃN TƯỚI".
  4. Nếu  $Soil_{du\_kien} < \text{Ngưỡng\_kho}$  -> Đánh dấu "TƯỚI" + Tính thời gian bơm (phút).
- Đầu ra hiển thị: Lịch chi tiết từng giờ.
  - Ví dụ: "Mai 08:00: Tưới 15 phút (Vì trời nắng, đất sẽ khô)".

## B. Khung Trung hạn (3-5 Ngày tới) - **Độ chính xác: Trung bình**

Đây là lịch trình mang tính Chiến thuật. Giúp hệ thống cân bằng giữa việc giữ ẩm và tiết kiệm nước.

- Dữ liệu đầu vào: API dự báo theo ngày (Daily Forecast).
- Logic Tích lũy (Accumulation Logic):
  - Hệ thống không nhìn từng giờ mà nhìn tổng lượng mưa tích lũy.
  - Ví dụ: Nếu ngày thứ 3 dự báo mưa rất to (50mm), hệ thống sẽ giảm lượng tưới của ngày thứ 1 và thứ 2 xuống mức tối thiểu (chỉ duy trì sự sống) để dành dung tích đất chứa nước mưa của ngày thứ 3.
- Đầu ra hiển thị: Xu hướng tưới.
  - Ví dụ: "Thứ 5 & Thứ 6: Dự kiến Hoàn tưới (Mưa rào)".

## C. Khung Dài hạn (7 Ngày tới) - **Độ chính xác: Thấp**

Đây là lịch trình mang tính Khuyến nghị / Cảnh báo.

- Dữ liệu đầu vào: Xu hướng nhiệt độ và mô hình mùa vụ.
- Logic Cảnh báo:
  - Phát hiện các đợt nắng nóng kéo dài (Heatwave) hoặc các đợt mưa dầm.
  - Nếu phát hiện chuỗi ngày nắng nóng (UV cao liên tục 7 ngày) -> Cảnh báo người dùng: "Tuần tới nắng gắt, vui lòng kiểm tra nguồn nước/bể chứa".
- Đầu ra hiển thị: Lời khuyên (Advisory).

### 6.3.3. Quy trình Tự động cập nhật (Rolling Update)

Lịch tưới không cố định mà là một Lịch trượt (Rolling Schedule):

1. Lúc 06:00: Hệ thống tạo lịch cho 7 ngày tới.
2. Lúc 12:00: Hệ thống nhận dữ liệu API mới.
  - Nếu dự báo thay đổi (VD: Con mưa dự báo chiều nay đã biến mất), hệ thống tính toán lại phương trình cân bằng nước.
  - Cập nhật lại trạng thái từ "HOÀN TỬỚI" sang "TỬỚI BÙ" trên giao diện App.
3. Cơ chế xung đột:
  - Nếu Lịch trình (Scheduler) báo "Tưới" NHƯNG Cảm biến 15s (Real-time) phát hiện áp suất tụt nhanh (dông lốc) -> Ưu tiên Real-time (Hủy lệnh tưới).

## 7. Kịch bản Kiểm thử và Đánh giá (Testing & Evaluation)

Do không có cảm biến mưa thực tế để kiểm chứng ngay lập tức (Real-time Validation), hệ thống được đánh giá dựa trên Kịch bản mô phỏng (Scenario-based Testing) và đối chiếu với dữ liệu lịch sử.

### 7.1. Kịch bản 1: Con mưa rào mùa Hè (Summer Squall)

- Tình huống: API báo xác suất mưa 40% (Thấp). Tuy nhiên, cảm biến BME280 ghi nhận áp suất giảm 3hPa trong 1 giờ.
- Phản ứng hệ thống:
  - Hệ thống Timer truyền thống: Vẫn tưới vì chưa đến giờ hẹn.
  - Hệ thống AI: Đặc trưng pressure\_slope kích hoạt trọng số cao -> AI đẩy xác suất mưa cục bộ lên 85% -> Quyết định HOÀN TỬỚI.
- Kết quả: Tiết kiệm nước tưới ngay trước khi cơn dông ập đến.

### 7.2. Kịch bản 2: Hiện tượng Nồm ẩm mùa Xuân

- Tình huống: Tháng 3, độ ẩm không khí 98%, tường nhà "đổ mồ hôi". Đất trông hơi khô (do lớp mặt ướt nhưng rễ khô). API báo "Drizzle" (Mưa phùn).

- Phản ứng hệ thống:
    - Hệ thống Rule-based (dựa trên ngưỡng ẩm): Thấy độ ẩm > 90% -> Ngắt tưới -> Cây thiếu nước.
    - Hệ thống AI: Nhận diện month=3 và chênh lệch điểm sương thấp -> Phân loại là "Nồm" -> Kiểm tra đất khô -> Quyết định Tưới.
  - Kết quả: Đảm bảo sức khỏe cây trồng trong điều kiện thời tiết cực đoan.
- 7.3. Đánh giá Hiệu năng Kỹ thuật
- Thời gian phản hồi (Latency): Trung bình 350ms (từ lúc nhận gói tin MQTT đến khi Relay đóng/ngắt). Đạt yêu cầu thời gian thực.
  - Độ ổn định: Hệ thống duy trì hoạt động liên tục trong thử nghiệm 72 giờ, tự động xử lý 12 lần mất kết nối API bằng chế độ Safe Mode.

## 8. Triển khai và Vận hành (Deployment & MLOps)

### 8.1. Quy trình Triển khai

1. Offline Training: Mô hình được huấn luyện trên máy tính cá nhân (hoặc Google Colab) dùng dữ liệu lịch sử 3 năm. Kết quả là file `xgboost_rain_fusion_v1.pkl` (~500KB).
2. Model Serving: File mô hình được tải lên Server (Node.js/Python).
3. Real-time Inference: Server chạy một tiến trình nền (Background Worker), cứ 15 giây một lần lấy dữ liệu sensor mới nhất, nạp vào mô hình để lấy kết quả dự báo.

### 8.2. Cơ chế An toàn (Fail-safe)

Hệ thống được thiết kế với nguyên tắc "Safety First":

- Mất kết nối Internet: Nếu không gọi được API OpenWeatherMap quá 30 phút -> Chuyển sang chế độ "Sensor Only" (Tưới dựa hoàn toàn vào độ ẩm đất).
- Cảm biến lỗi: Nếu giá trị cảm biến vượt ngưỡng vật lý (VD: Nhiệt độ > 60°C hoặc Âm) -> Báo động cho người dùng qua App và ngắt toàn bộ bơm.

## 9. Kết luận Phân hệ AI

Phân hệ AI của dự án Tưới cây thông minh đã hoàn thành mục tiêu xây dựng một giải pháp "Cảm biến mưa ảo" (Virtual Rain Sensor) thông qua kỹ thuật Hợp nhất dữ liệu (Data Fusion).

Những điểm đột phá chính:

1. Giải quyết bài toán Chi phí: Loại bỏ hoàn toàn nhu cầu sử dụng cảm biến mưa vật lý đắt tiền mà vẫn đảm bảo khả năng phát hiện mưa nhờ phân tích áp suất và dữ liệu API.
2. Giải quyết bài toán Địa phương: Khắc phục triệt để các sai lầm trong tưới tiêu vào mùa Nồm ẩm và mùa mưa rào đặc thù của Hà Nội nhờ Logic thích ứng 4 mùa.
3. Khả năng Thực tiễn: Với tần suất xử lý 15 giây/lần và độ trễ thấp, hệ thống chứng minh được tính khả thi để triển khai trên các thiết bị IoT phần cứng hạn chế (ESP32) và hạ tầng mạng thông thường.

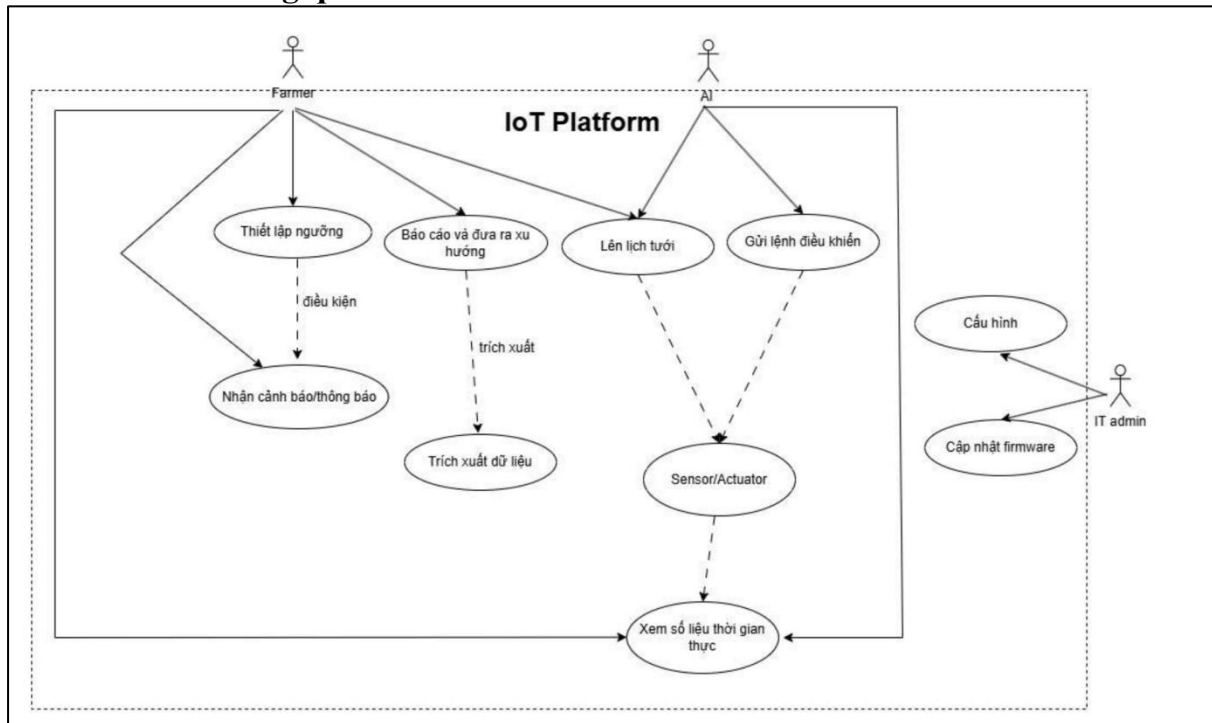
Hướng phát triển tiếp theo bao gồm việc thu thập dữ liệu vận hành thực tế (Feedback Loop) để AI tự động tinh chỉnh ngưỡng quyết định mà không cần lập trình lại (Online Learning)

## 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 6: 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

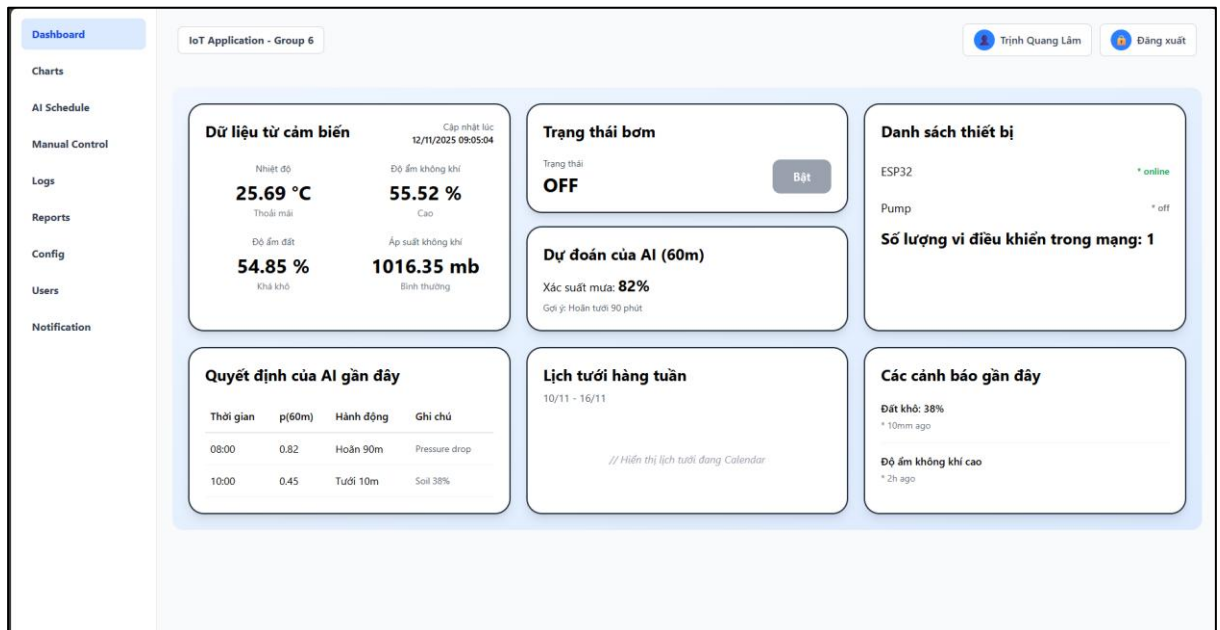
## 2. Mô tả Use Case

- 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. Giao diện website

- Hệ thống được phát triển có 2 tác nhân sử dụng là Farmer và Admin. Trong đó Admin sẽ có tất cả quyền truy cập vào các trang giao diện. Farmer thì sẽ bị hạn chế truy cập vào các trang: Users, Logs, Reports, Config.

### 1. Giao diện Dashboard



Hình 7: Giao diện trang Dashboard

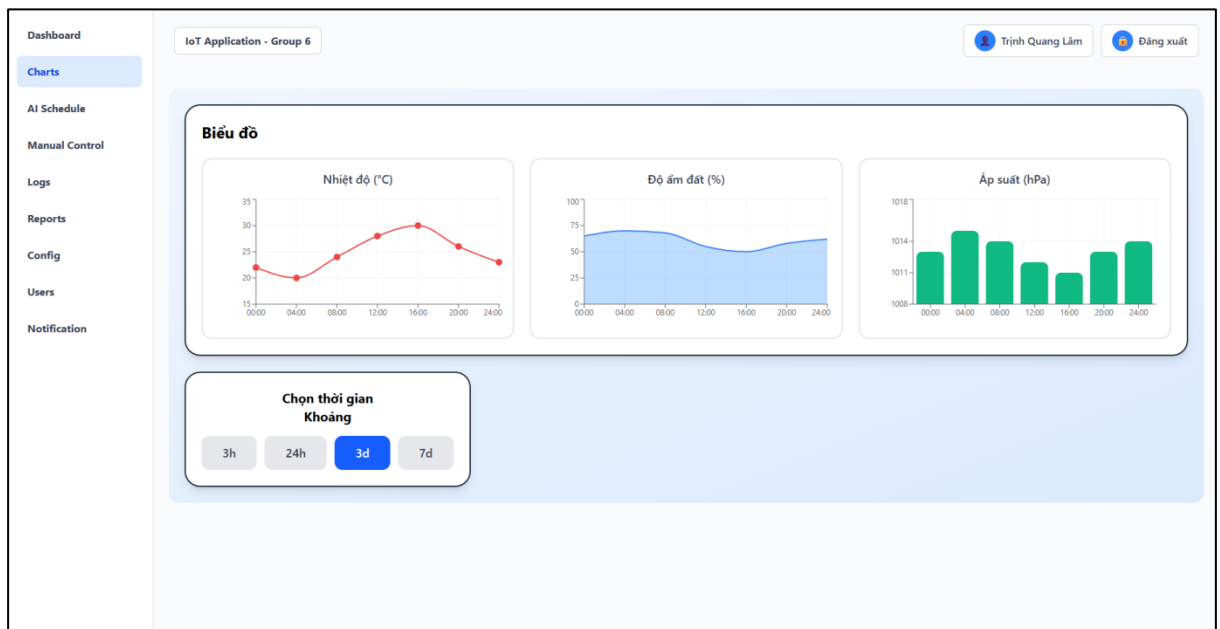
### Mô tả:

- Trang tổng quan chính, hiển thị dữ liệu cảm biến real-time, trạng thái bơm và dự đoán của AI trong 60 phút tới
- Là giao diện đầu tiên khi người dùng đăng nhập vào hệ thống

### Chức năng chính:

- Hiển thị nhiệt độ, độ ẩm không khí, độ ẩm đất, áp suất không khí (cập nhật liên tục từ ESP32).
- Thông báo trạng thái bơm (ON/OFF) và cho phép bật/tắt nhanh.
- Hiển thị dự báo mưa 60 phút tới do AI xử lý.
- Bảng quyết định của AI gần đây thể hiện các quyết định gần nhất.
- Lịch tưới hàng tuần minh họa kế hoạch tưới tuần hiện tại.

## 2. Charts



Hình 8: Giao diện trang biểu đồ

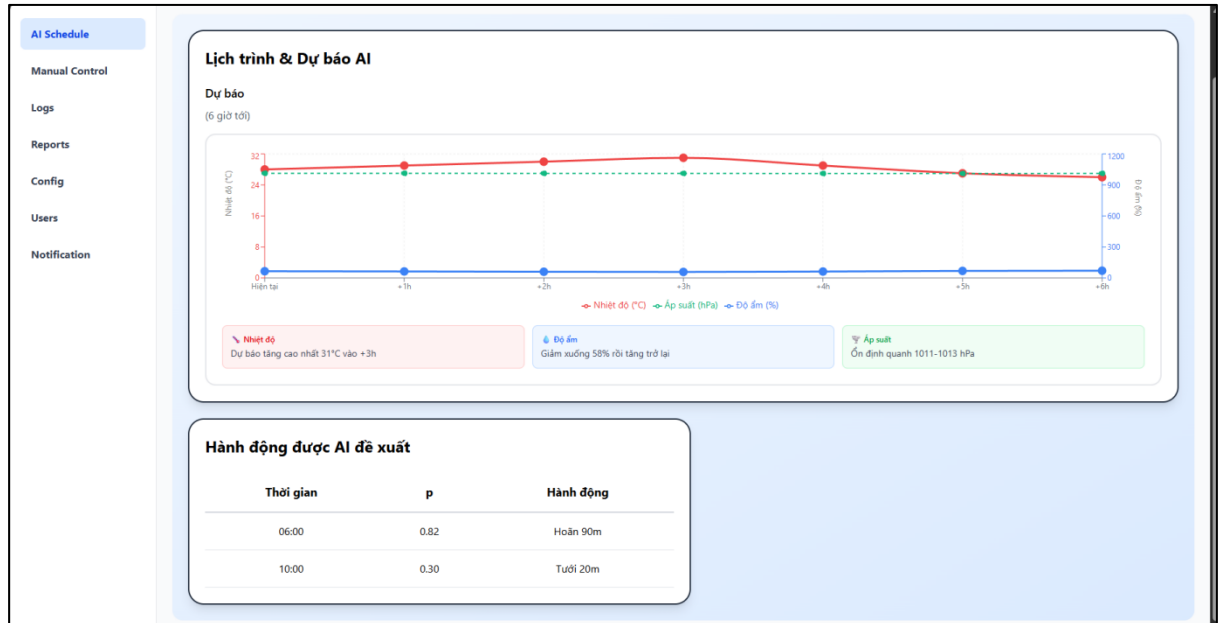
### Mô tả:

- Trang hiển thị biểu đồ dữ liệu cảm theo thời gian giúp người dùng theo dõi xu hướng biến đổi

### Chức năng chính:

- Biểu đồ nhiệt độ (Temperature chart)
- Biểu đồ độ ẩm đất (Soil moisture chart)
- Biểu đồ áp suất không khí (Air pressure chart)
- Bộ chọn khoảng thời gian để hiển thị dữ liệu quá khứ

### 3. AI Schedule



Hình 9: Lịch trình và dự báo của AI

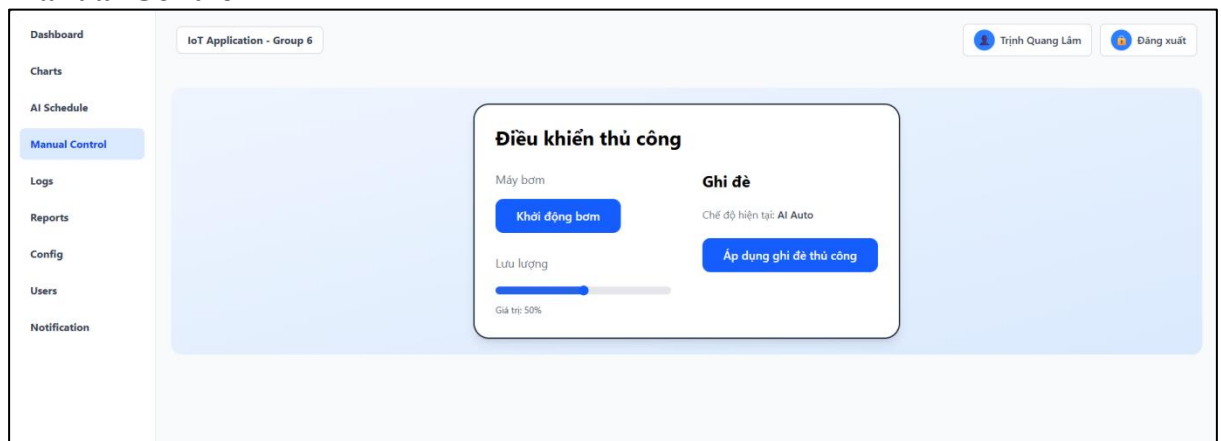
### Mô tả:

- Trang chuyên biệt cho phần AI dự báo và lên lịch tưới thông minh

### Chức năng chính

- Biểu đồ xác suất mưa theo thời gian (next 6 hours).
- Bảng đề xuất hành động của AI hiển thị hành động dự kiến (Hoãn tưới / Tưới 50%...).
- Người dùng có thể xem lịch sử dự báo để đánh giá độ chính xác.

### 4. Manual Control



Hình 10: Giao diện tưới thủ công

### Mô tả:

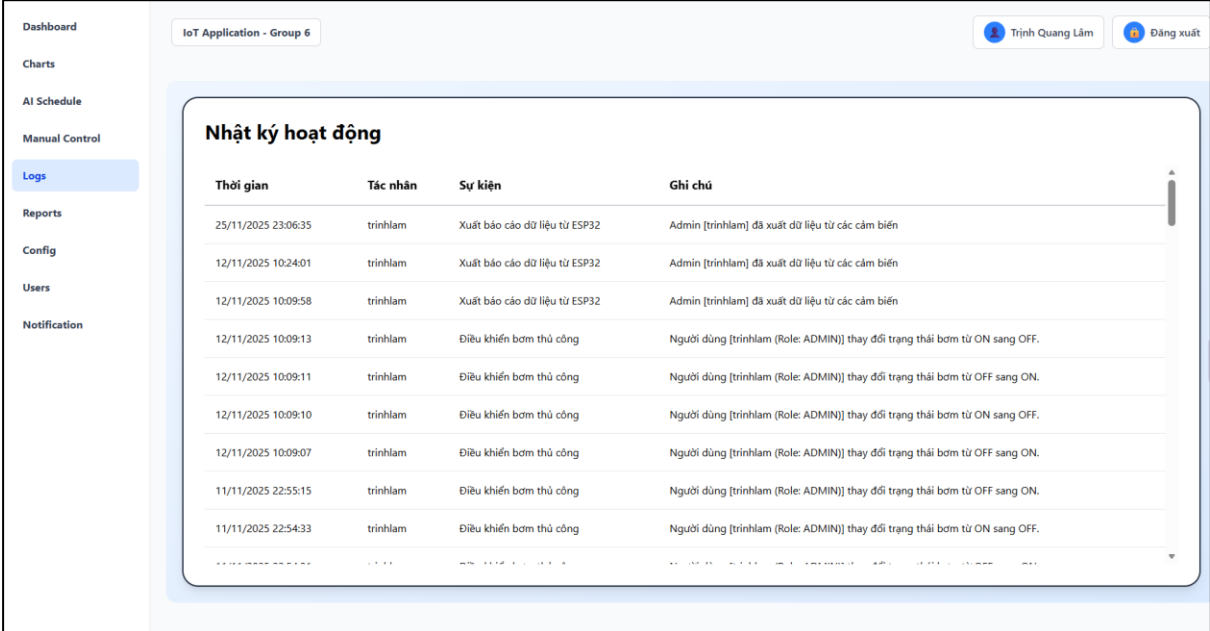
- Cho phép điều khiển thủ công bơm nước và ghi đè tạm thời lên chế độ AI

### Chức năng chính:

- Bật/tắt bơm thủ công

- Điều chỉnh flow rate (PWM) bằng thanh trượt
- Nút “Áp dụng ghi đè thủ công” → chuyển từ hệ thống tự động sang thủ công

## 5. Logs



Thời gian	Tác nhân	Sự kiện	Ghi chú
25/11/2025 23:06:35	trinhlam	Xuất báo cáo dữ liệu từ ESP32	Admin [trinhlam] đã xuất dữ liệu từ các cảm biến
12/11/2025 10:24:01	trinhlam	Xuất báo cáo dữ liệu từ ESP32	Admin [trinhlam] đã xuất dữ liệu từ các cảm biến
12/11/2025 10:09:58	trinhlam	Xuất báo cáo dữ liệu từ ESP32	Admin [trinhlam] đã xuất dữ liệu từ các cảm biến
12/11/2025 10:09:13	trinhlam	Điều khiển bơm thủ công	Người dùng [trinhlam (Role: ADMIN)] thay đổi trạng thái bơm từ ON sang OFF.
12/11/2025 10:09:11	trinhlam	Điều khiển bơm thủ công	Người dùng [trinhlam (Role: ADMIN)] thay đổi trạng thái bơm từ OFF sang ON.
12/11/2025 10:09:10	trinhlam	Điều khiển bơm thủ công	Người dùng [trinhlam (Role: ADMIN)] thay đổi trạng thái bơm từ ON sang OFF.
12/11/2025 10:09:07	trinhlam	Điều khiển bơm thủ công	Người dùng [trinhlam (Role: ADMIN)] thay đổi trạng thái bơm từ OFF sang ON.
11/11/2025 22:55:15	trinhlam	Điều khiển bơm thủ công	Người dùng [trinhlam (Role: ADMIN)] thay đổi trạng thái bơm từ OFF sang ON.
11/11/2025 22:54:33	trinhlam	Điều khiển bơm thủ công	Người dùng [trinhlam (Role: ADMIN)] thay đổi trạng thái bơm từ ON sang OFF.

Hình 11: Giao diện trang nhật ký hoạt động

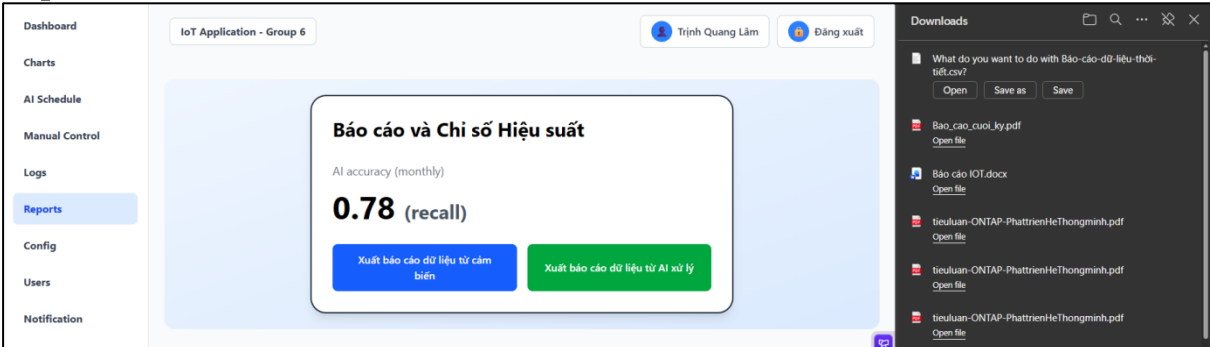
**Mô tả:**

- Ghi lại toàn bộ lịch sử hoạt động của hệ thống

**Chức năng chính:**

- Hiển thị các bản ghi bao gồm: thời gian, tác nhân, sự kiện, ghi chú

## 6. Reports



Báo cáo và Chỉ số Hiệu suất	
AI accuracy (monthly)	
<b>0.78 (recall)</b>	
Xuất báo cáo dữ liệu từ cảm biến	Xuất báo cáo dữ liệu từ AI xử lý

Hình 12: Giao diện trang xuất báo cáo

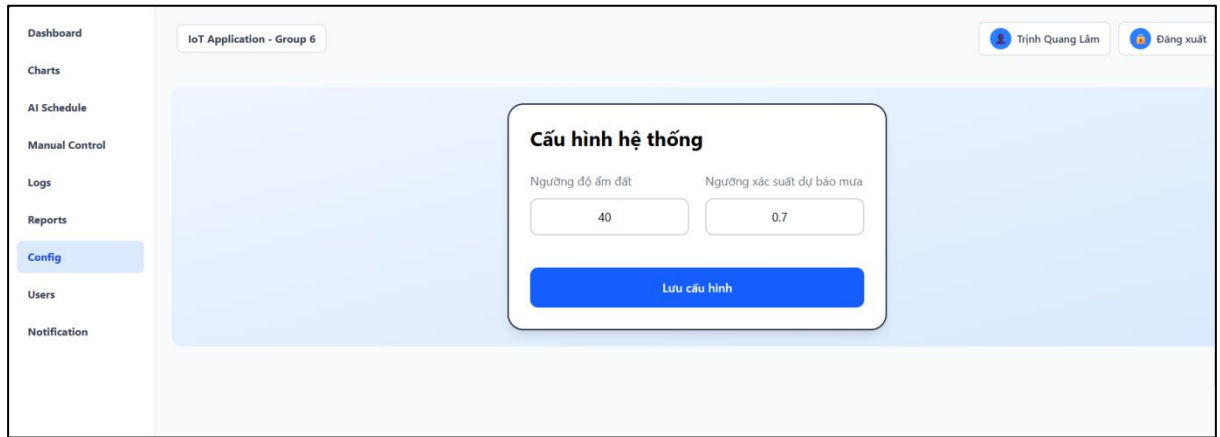
**Mô tả:**

- Xuất báo cáo dữ liệu của cảm biến và báo cáo dữ liệu dự báo của AI

**Chức năng chính:**

- Các bản báo cáo được xuất ra dưới dạng file .csv và người dùng có thể tải về máy

## 7. Config



Hình 13: Giao diện trang cấu hình các ngưỡng

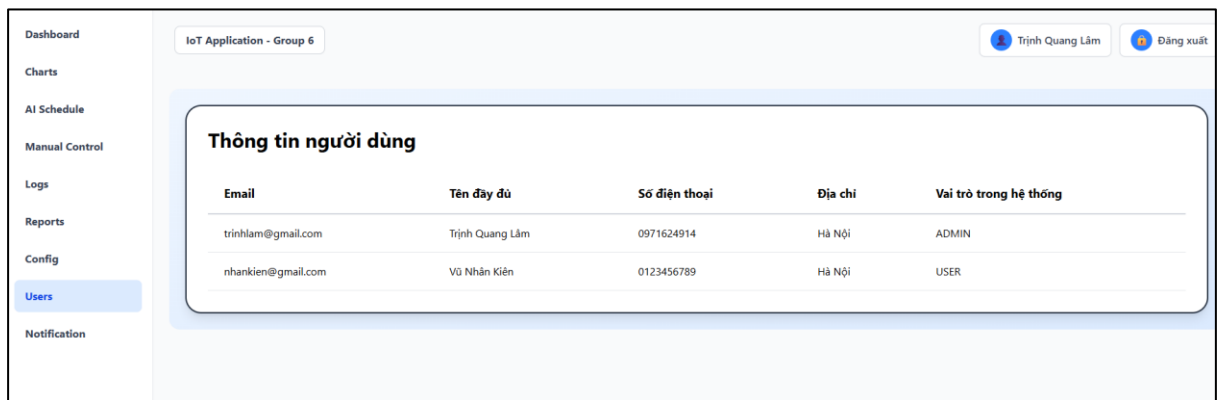
#### Mô tả:

- Cho phép người dùng hoặc admin cấu hình ngưỡng hoạt động của hệ thống

#### Chức năng chính:

- Cài đặt ngưỡng độ ẩm đất
- Cài đặt ngưỡng xác suất mưa của AI
- Nút lưu cấu hình thay đổi

## 8. Users



Hình 14: Giao diện trang danh sách người dùng hệ thống

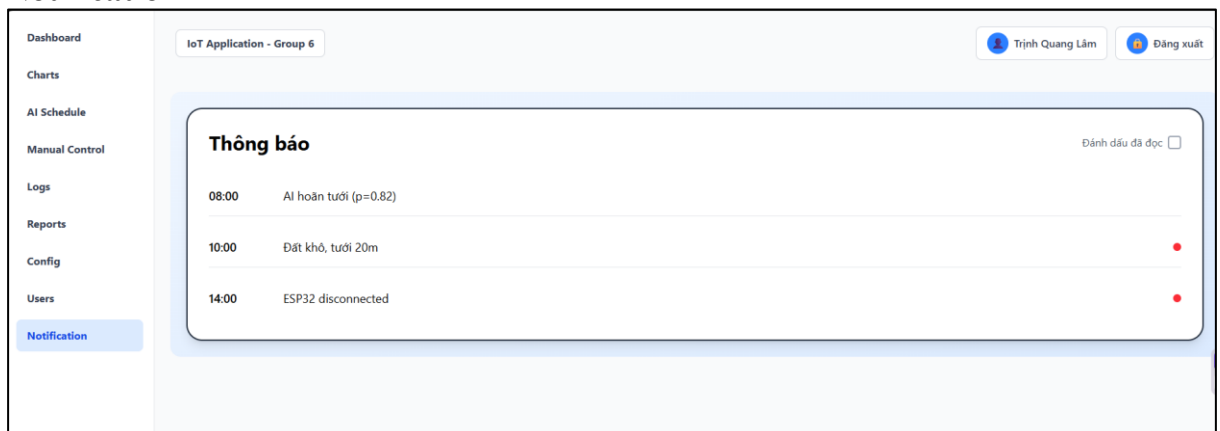
#### Mô tả:

- Danh sách người dùng của hệ thống

#### Chức năng chính:

- Liệt kê danh sách người dùng (email, vai trò, trạng thái)

## 9. Notification



Hình 15: Giao diện trang thông báo

#### Mô tả:

- Hiển thị các cảnh báo, thông báo hệ thống như mất kết nối, đất khô, dự báo mưa

#### **Chức năng chính:**

- Danh sách thông báo theo thời gian
- Tự động cập nhật khi có event mới

### **V. 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**

<b>Mã sinh viên</b>	<b>Họ tên</b>	<b>Nhiệm vụ</b>
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 module AI 2. Làm báo cáo 3. Thuyết trình

#### **2. Kế hoạch triển khai**

<b>Tuần</b>	<b>Nội dung chính</b>	<b>Kết quả dự kiến</b>
Tuần 1 (17/09 – 23/09)	- 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)	- 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)	- Viết chương trình ESP32 đọc dữ liệu cảm biến	ESP32 đọc và gửi được dữ liệu nhiệt độ, độ ẩm, độ

	<ul style="list-style-type: none"> <li>- Gửi dữ liệu qua Serial / Wi-Fi đến server thử nghiệm</li> <li>- Kiểm tra tín hiệu từ cảm biến</li> </ul>	âm đất.
Tuần 4 (07/10 – 13/10)	<ul style="list-style-type: none"> <li>- Xây dựng backend (NodeJS + ExpressJS)</li> <li>- Thiết kế API nhận dữ liệu từ ESP32 và lưu MongoDB</li> <li>- Kiểm thử API với dữ liệu mẫu</li> </ul>	Backend hoạt động, lưu dữ liệu cảm biến thành công
Tuần 5 (14/10 – 20/10)	<ul style="list-style-type: none"> <li>- Phát triển frontend (ReactJS)</li> <li>- Hiển thị dữ liệu cảm biến theo thời gian thực</li> <li>- Thiết kế giao diện hiển thị độ ẩm, nhiệt độ, trạng thái bơm</li> </ul>	Giao diện Web cơ bản hiển thị dữ liệu IoT
Tuần 6 (21/10 – 27/10)	<ul style="list-style-type: none"> <li>- Thiết kế và huấn luyện mô hình AI (XGBoost) dự báo mưa</li> <li>- Chuẩn bị dữ liệu huấn luyện, xử lý Feature</li> <li>- Mô hình AI huấn luyện được, có thể dự đoán xác suất mưa</li> </ul>	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"> <li>- Tích hợp AI với backend</li> <li>- Xây dựng API cho dự đoán thời tiết</li> <li>- Thử nghiệm điều kiện tưới tự động dựa trên kết quả AI</li> </ul>	Hệ thống tự động tưới theo dự báo mưa hoạt động
Tuần 8 (04/11 – 10/11)	<ul style="list-style-type: none"> <li>- Hoàn thiện giao diện web (hiển thị lịch tưới, biểu đồ, điều khiển thủ công)</li> <li>- Kiểm thử toàn hệ thống (phần cứng – backend – frontend – AI)</li> </ul>	Toàn bộ luồng dữ liệu hoạt động ổn định, giao diện hoàn chỉnh
Tuần 9 (11/11 – 17/11)	<ul style="list-style-type: none"> <li>- Viết báo cáo cuối kỳ (theo mục P1–P2)</li> <li>- Tổng hợp hình ảnh, kết quả kiểm thử</li> <li>- Chuẩn bị slide thuyết trình</li> </ul>	Báo cáo cuối kỳ hoàn chỉnh, slide trình bày sẵn sàng
Tuần 10 (18/11 – 24/11)	<ul style="list-style-type: none"> <li>- Nộp và trình bày báo cáo cuối kỳ</li> <li>- Thu thập phản hồi từ giảng viên</li> <li>- Lên kế hoạch cải tiến giai đoạn cuối kỳ (bổ sung AI hoặc mở rộng cảm biến)</li> </ul>	Báo cáo cuối kỳ được chấm và phản biện xong

## VI. 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