

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Khoa: Công nghệ thông tin 1



BÁO CÁO GIỮA KỲ

Đề tài: HỆ THỐNG TƯỚI CÂY THÔNG MINH

GVHD : TS. Kim Ngọc Bách

SV thực hiện : Nguyễn Đức Đạt

Lưu Minh Hiền

Nguyễn Đức

Khởi

Vũ Hồng Linh

Nhóm môn học : 06

Môn học : IoT và Ứng dụng

Hà Nội - 10/2025

Chương 1: TỔNG QUAN

1.1. Đặt vấn đề

Ngày nay, công nghệ kết nối đầu tiên cần nhắc đến hiển nhiên là Wifi – công nghệ kết nối không dây phổ biến nhất hiện nay. Cũng vì tính phổ biến của dạng kết nối này mà cái tên Wifi thường bị lạm dụng để chỉ kết nối không dây nói chung.

Lí do mà kết nối Wifi được ưa chuộng như vậy đơn giản là vì khả năng hoạt động hiệu quả trong phạm vi vài chục đến vài trăm mét của các mạng WLAN.

Và trong thời đại công nghiệp hóa hiện đại hóa hiện nay, việc phát minh và chế tạo ra các thiết bị thông minh có khả năng điều khiển từ xa đang và sẽ rất được quan tâm và rất hữu ích cho cuộc sống hàng ngày.

Trong bối cảnh nền nông nghiệp Việt Nam đang hướng tới hiện đại hóa và ứng dụng công nghệ cao để gia tăng năng suất, việc tối ưu hóa nguồn tài nguyên nước và nhân công là một yêu cầu cấp thiết. Nhận thấy những thách thức trong phương pháp tưới tiêu truyền thống, chúng tôi đã quyết định thực hiện đề án “Hệ thống IoT tưới tiêu thông minh”.

Đề tài này không chỉ dừng lại ở việc bật/tắt máy bơm từ xa, mà còn xây dựng một giải pháp nông nghiệp chính xác. Cốt lõi của hệ thống là việc giám sát các cảm biến được đặt trực tiếp tại nông trại, như cảm biến độ ẩm đất, nhiệt độ, và độ ẩm không khí. Dữ liệu này sẽ được thu thập và phân tích liên tục để đưa ra quyết định tưới tiêu tối ưu. Khi dự án hoàn thành, hệ thống có thể cảnh báo khi đất quá khô hoặc có sự cố ở máy bơm, tự động tưới theo độ ẩm thực tế của đất thay vì lịch trình cố định, và cho phép người dùng điều khiển toàn bộ hệ thống van, máy bơm... thông qua các nút nhấn hiển thị trạng thái hoạt động trên giao diện web hoặc ứng dụng di động (Android/iOS).

Như vậy, người nông dân dù ở bất cứ nơi nào, chỉ cần có kết nối internet, đều có thể giám sát và điều khiển toàn bộ quá trình tưới tiêu cho nông trại của mình. Điều này không chỉ giúp tiết kiệm thời gian, công sức di chuyển mà còn đảm bảo cây trồng luôn nhận được lượng nước phù hợp, tránh lãng phí.

Khi dự án thành công và được áp dụng rộng rãi, mô hình này sẽ mang lại hiệu quả kinh tế vượt trội, giúp người nông dân tiết kiệm tới 50% lượng nước tưới và tăng năng suất cây trồng. Đây là một giải pháp thiết thực, góp phần vào sự phát triển bền vững của nền nông nghiệp nước nhà, đưa Việt Nam tiến gần hơn tới một nền nông nghiệp thông minh, hiện đại.

1.2. Phạm vi triển khai

Mục tiêu của nhóm là triển khai hệ thống tưới tiêu thông minh trong phạm vi khu vườn của một hộ gia đình, khu vườn này có diện tích bề mặt $\leq 40m^2$.

a) Số lượng thiết bị

Hệ thống sẽ được giữ ở mức tối giản để đảm bảo hiệu quả và tiết kiệm chi phí:

- Cảm biến (Sensor): Cần từ 1 đến 2 cảm biến độ ẩm đất và một cảm biến mưa. Chúng sẽ được cắm ở các khu vực trọng yếu của khu vườn.
- Cơ cấu chấp hành (Actuator): 1 van nước điện từ hoặc 1 máy bơm mini. Thiết bị này sẽ được kết nối với nguồn nước chính của gia đình (vòi nước) hoặc kết nối thẳng tới téc chứa nước của gia đình.
- Bộ điều khiển trung tâm (Gateway/Controller): Sử dụng 1 vi điều khiển có kết nối Wi-Fi ESP32. Đây là bộ não trung tâm, vừa thu thập dữ liệu từ cảm biến, vừa điều khiển van/bơm, và kết nối vào mạng internet của gia đình.

b) Quy mô mạng

- Cục bộ: Hệ thống chỉ hoạt động trong phạm vi khu vườn của một hộ gia đình. Do đó, quy mô mạng là hoàn toàn cục bộ và nhỏ lẻ.
- Công nghệ kết nối: Mạng Wi-Fi gia đình là đủ và phù hợp nhất. Tất cả các thiết bị sẽ kết nối vào mạng Wi-Fi sẵn có.

c) Môi trường hoạt động

- Nông nghiệp quy mô nhỏ (ngoài trời): Các thiết bị như cảm biến và van nước sẽ được đặt ngoài trời, phải chịu ảnh hưởng trực tiếp của thời tiết (nắng, mưa, độ ẩm cao).
 - Giải pháp: Cần đặt bộ vi điều khiển trong một hộp nhựa chống nước. Cảm biến và van nước cũng cần chọn loại có độ bền cao, được thiết kế để sử dụng ngoài trời.
- Yêu cầu khác: Đảm bảo tính bảo mật an toàn thông tin cho thiết bị IOT.

1.3. Tiêu chí của nhóm

Để đánh giá một cách khách quan xem hệ thống tưới cây thông minh có hoạt động tốt và đạt được mục tiêu đề ra hay không, nhóm chúng tôi xây dựng các chỉ số hiệu suất chính (KPIs) có thể định lượng được. Các KPIs này giúp đo lường và kiểm chứng mức độ thành công của dự án. Các KPIs cụ thể cho hệ thống này bao gồm:

a) Độ chính xác (Accuracy):

- Mục tiêu: Đảm bảo dữ liệu từ cảm biến phản ánh đúng tình trạng của đất.
- Chỉ số đo lường: Sai số của giá trị độ ẩm đất mà cảm biến đo được không được vượt quá $\pm 5\%$ so với kết quả đo bằng thiết bị chuyên dụng.

b) Độ trễ (Latency):

- Mục tiêu: Hệ thống phải có khả năng phản hồi nhanh chóng với các lệnh từ người dùng.

- Chỉ số đo lường: Thời gian từ khi người dùng nhấn nút bật/tắt trên ứng dụng di động đến khi van nước thực sự thay đổi trạng thái (mở hoặc đóng) phải dưới 5 giây.

c) Độ tin cậy (Reliability):

- Mục tiêu: Hệ thống phải hoạt động ổn định và liên tục trong điều kiện thực tế.
- Chỉ số đo lường:
 - Tỷ lệ thời gian hoạt động (uptime) của hệ thống phải đạt trên 98% trong suốt quá trình thử nghiệm.
 - Hệ thống phải có khả năng tự động kết nối lại mạng Wi-Fi trong vòng 1 phút khi có sự cố mất kết nối tạm thời.

d) Chi phí (Cost):

- Mục tiêu: Xây dựng một hệ thống hiệu quả với chi phí hợp lý, phù hợp cho quy mô hộ gia đình.
- Chỉ số đo lường:
 - Chi phí đầu tư (CAPEX): Tổng chi phí để mua sắm toàn bộ linh kiện và vật tư phụ không vượt quá ngân sách đề ra là 500,000 VNĐ.
 - Chi phí vận hành (OPEX): Chi phí vận hành bao gồm tiền điện duy trì hệ thống IOT, tiền nước tưới tiêu và chi phí bảo hành nếu có hỏng hóc.

e) Khả năng mở rộng và tiết kiệm tài nguyên:

- Hệ thống phải hỗ trợ nhân đôi số lượng cảm biến trong 3 năm tới mà không cần thay đổi hạ tầng mạng.
- Giảm ít nhất 30-50% lượng nước tưới so với phương pháp truyền thống.

1.4. Kết quả mong đợi

- Tự động hóa tưới tiêu để tiết kiệm nước và thời gian của gia đình.
- Người trong nhà có thể giám sát từ xa qua smartphone.
- Dữ liệu thu thập liên tục giúp đưa ra quyết định tưới tiêu tốt hơn (ví dụ: tưới bằng này nước cho loại cây A thì là hợp lý,...).

Chương 2: Thu thập yêu cầu từ các bên liên quan

2.1. Người dùng thiết bị (hộ gia đình)

Đối tượng: Chủ nhà và các thành viên trong gia đình, là những người trực tiếp tương tác và hưởng lợi từ hệ thống. Họ có thể không có chuyên môn sâu về kỹ thuật.

Các yêu cầu cụ thể:

- **Giao diện giám sát:** Phải đơn giản, trực quan trên ứng dụng điện thoại. Cần hiển thị rõ ràng các thông tin cốt lõi:
 - Chỉ số độ ẩm đất theo thời gian thực.
 - Trạng thái hoạt động của van/máy bơm (đang Bật hay Tắt).
- **Khả năng điều khiển:** Cần có một nút bấm lớn, dễ sử dụng trên ứng dụng để có thể Bật/Tắt chế độ tưới thủ công ngay lập tức khi cần.
- **Tính tự động:** Yêu cầu quan trọng nhất là hệ thống phải hoạt động ở chế độ "cài đặt và quên đi" (set and forget), tự động tưới khi đất khô mà không cần sự can thiệp hàng ngày.
- **Cảnh báo:** Nhận được thông báo đơn giản trên điện thoại khi hệ thống mất kết nối hoặc khi độ ẩm ở mức quá thấp trong thời gian dài.

2.2. Yêu cầu từ góc độ Quản lý & Đầu tư

Đối tượng: Chủ nhà, đóng vai trò là người quản lý dự án và nhà đầu tư.

Các yêu cầu cụ thể:

- **Hiệu quả chi phí (ROI):** Tổng chi phí đầu tư ban đầu (CAPEX) phải nằm trong ngân sách đã duyệt (dưới 500,000 VNĐ). Lợi ích thu lại ("Return") là tiết kiệm chi phí hóa đơn nước hàng tháng, tiết kiệm thời gian chăm sóc và đảm bảo cây cối trong vườn luôn khỏe mạnh, tránh chi phí mua cây mới.
- **Độ tin cậy và Chi phí vận hành (OPEX):** Hệ thống phải tiêu thụ ít năng lượng và không phát sinh thêm chi phí vận hành đáng kể nào khác. Yêu cầu hệ thống hoạt động bền bỉ, ít phải bảo trì, sửa chữa.
- **Khả năng mở rộng:** Thiết kế của hệ thống (cả phần cứng và phần mềm) cần cho phép khả năng mở rộng trong tương lai, ví dụ như thêm một cảm biến cho một khu vực vườn khác mà không cần thay đổi toàn bộ hệ thống.

2.3. Yêu cầu từ góc độ Kỹ thuật

Đối tượng: Người tự thực hiện dự án.

Các yêu cầu cụ thể:

- **Tích hợp hệ thống sẵn có:** Hệ thống IoT phải tích hợp và hoạt động trơn tru với hạ tầng mạng Wi-Fi hiện có của gia đình.
- **Giao thức truyền thông:** Lựa chọn nền tảng và giao thức phù hợp cho quy mô nhỏ. Wi-Fi kết hợp với giao thức MQTT (được hỗ trợ bởi các nền tảng như Blynk) là lựa chọn tối ưu.
- **Chuẩn bảo mật:** Phải tuân thủ các chuẩn bảo mật cơ bản cho mạng gia

đình, bao gồm sử dụng mật khẩu Wi-Fi mạnh (WPA2) và bảo vệ các mã truy cập (token) của dịch vụ IoT.

2.4. Phương pháp thu thập yêu cầu

Để hiểu đúng và đủ các nhu cầu trên, nhóm chúng tôi đề xuất các phương pháp sau:

- **Phỏng vấn:** Trao đổi trực tiếp với các thành viên trong gia đình để lắng nghe mong muốn và kỳ vọng của họ về một hệ thống tưới cây tự động.
- **Quan sát thực tế:** Khảo sát trực tiếp tại khu vực vườn cây để xác định các vấn đề thực tiễn: vị trí lắp đặt cảm biến tối ưu, vị trí nguồn điện/nguồn nước, các yếu tố môi trường có thể ảnh hưởng đến thiết bị. Việc này giúp tránh tình trạng thiết kế "trên lý thuyết" nhưng lại không phù hợp với thực tế.

Chương 3: Phân tích yêu cầu chức năng

3.1. Mô tả tổng quát các chức năng cơ bản và luồng hoạt động của hệ thống
Hệ thống tưới cây thông minh hoạt động dựa trên một luồng khép kín, kết hợp giữa việc thu thập dữ liệu từ môi trường thực tế và khả năng điều khiển từ xa của người dùng. Luồng hoạt động của hệ thống có thể được mô tả qua các bước chính như sau:

Bước 1: Thu thập dữ liệu độ ẩm từ Cảm biến

- **Cảm biến độ ẩm đất và cảm biến độ ẩm không khí** sẽ được lắp đặt để theo dõi trạng thái của cây trồng. Cảm biến sẽ liên tục đo độ ẩm và chuyển đổi thành tín hiệu điện (dưới dạng điện trở hoặc điện áp).

Bước 2: Xử lý và Gửi dữ liệu tại Vi điều khiển ESP

- Tín hiệu điện từ cảm biến được gửi đến **vi điều khiển ESP**.
- Mạch ESP sẽ đọc và chuyển đổi tín hiệu này thành một giá trị số cụ thể mà hệ thống có thể hiểu được (ví dụ: phần trăm độ ẩm).
- Sau đó, ESP sẽ đóng gói dữ liệu này và gửi lên **Cloud Server** thông qua mạng Wi-Fi của gia đình, sử dụng các giao thức mạng phổ biến như MQTT hoặc HTTP.

Bước 3: Xử lý, Lưu trữ và Phản hồi từ Cloud Server

- **Cloud Server** nhận dữ liệu từ ESP gửi lên.
- Dữ liệu này ngay lập tức được xử lý và lưu trữ vào **Cơ sở dữ liệu (Database)** để ghi nhận lịch sử và phục vụ cho việc phân tích sau này.
- Đồng thời, Cloud Server cung cấp các giao diện lập trình ứng dụng (API) hoặc Socket để **Ứng dụng di động (UI)** có thể truy vấn và lấy dữ liệu này về.

Bước 4: Hiển thị và Tương tác trên Giao diện người dùng (UI)

- Ứng dụng di động trên điện thoại của người dùng sẽ gọi tới API của Cloud Server để lấy dữ liệu mới nhất.
- Dữ liệu (độ ẩm, trạng thái máy bơm, v.v.) sẽ được **hiển thị một cách trực quan** trên giao diện ứng dụng.
- Người dùng có thể theo dõi các thông số và tương tác với hệ thống bằng cách nhấn các nút điều khiển (ví dụ: nút "Bật/Tắt máy bơm thủ công").

Bước 5: Luồng điều khiển ngược từ Người dùng đến Thiết bị

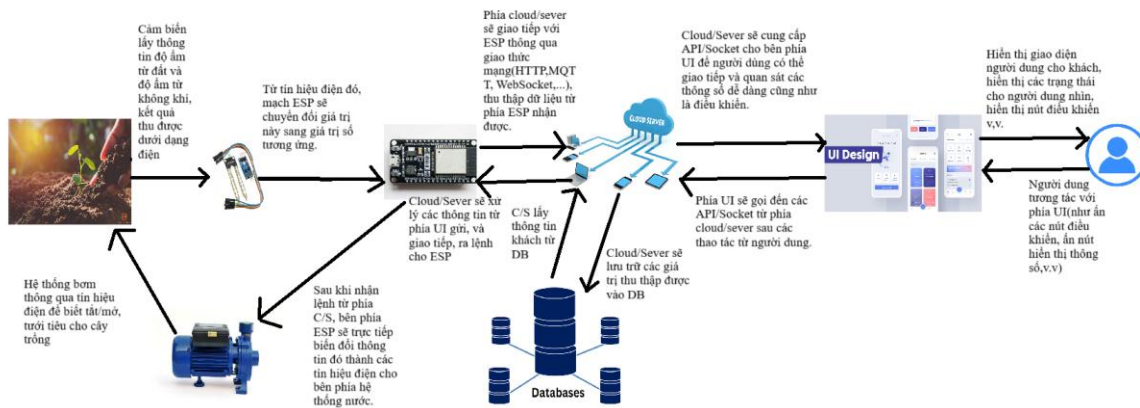
- Khi người dùng thực hiện một thao tác trên ứng dụng (ví dụ: nhấn nút "Bật"), một lệnh điều khiển sẽ được gửi từ ứng dụng đến **Cloud Server**.
- Cloud Server xác thực và xử lý lệnh này, sau đó gửi một thông điệp điều khiển xuống cho **vi điều khiển ESP** tương ứng.

Bước 6: Thực thi lệnh và Hoàn thành chu trình

- **Vi điều khiển ESP** nhận lệnh từ Cloud Server.
- ESP sẽ xử lý lệnh này và xuất ra tín hiệu điện để điều khiển **hệ thống bơm**

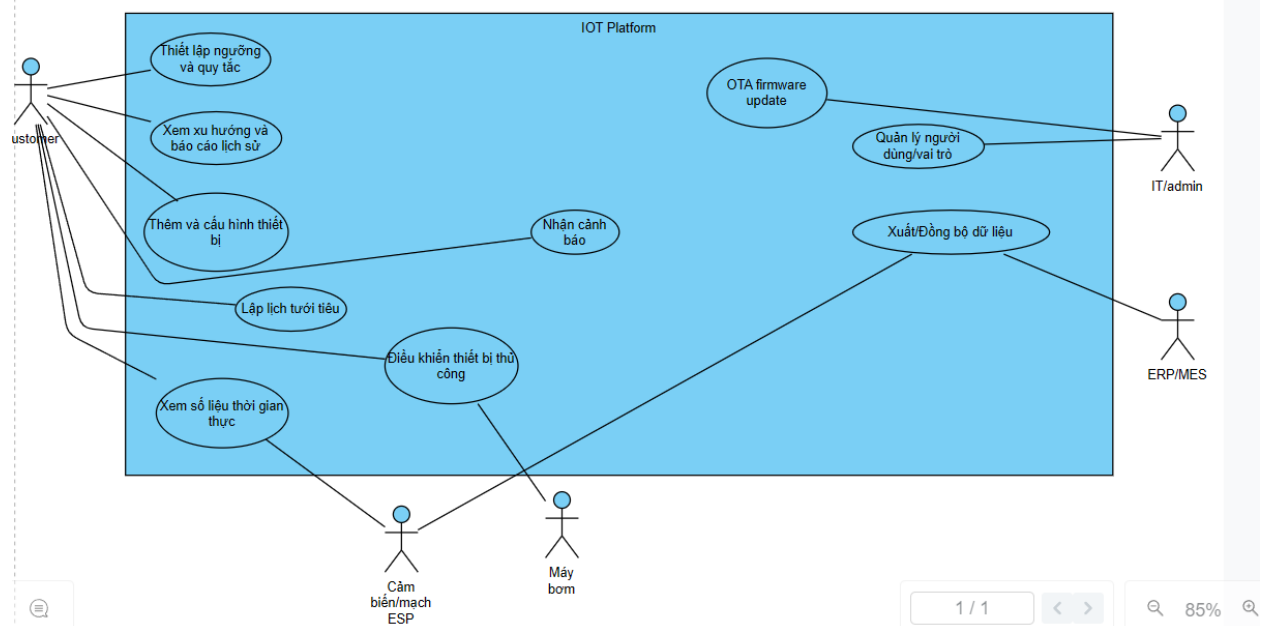
(thường thông qua một mạch relay).

- Máy bơm sẽ được bật/tắt theo đúng lệnh, thực hiện việc tưới tiêu cho cây trồng.
- Quá trình này làm thay đổi độ ẩm của đất, và cảm biến sẽ lại ghi nhận giá trị mới, bắt đầu lại một chu trình giám sát mới.



Hình 3.1.1: Mô tả sơ lược về luồng hoạt động của hệ thống

3.2. Đặc tả luồng công việc



Hình 3.2: Sơ đồ mô tả Use-case

Mô tả Use-case:

1. Các Actor trong hệ thống

- **Customer (Khách hàng/Chủ nhà):** Đây là người dùng chính và quan trọng

nhất của hệ thống. **Customer** là người sở hữu khu vườn, có nhu cầu giám sát và tự động hóa việc tưới tiêu. Họ tương tác với hệ thống chủ yếu thông qua giao diện ứng dụng di động để xem thông tin, điều khiển thủ công và thiết lập các quy tắc tự động.

- **IT/admin (Quản trị viên):** Đây là người dùng có vai trò kỹ thuật, chịu trách nhiệm quản lý và bảo trì hệ thống. **IT/admin** có các quyền hạn nâng cao như cập nhật phần mềm cho thiết bị từ xa (OTA firmware update) và quản lý tài khoản người dùng. Trong bối cảnh dự án hộ gia đình, vai trò này và vai trò Customer có thể là cùng một người, nhưng được tách ra để thể hiện sự phân chia về chức năng quản trị.
- **ERP/MES (Hệ thống Doanh nghiệp):** Đây là một tác nhân hệ thống (system actor), đại diện cho một phần mềm bên ngoài (ví dụ: một bảng tính Google Sheets, một hệ thống quản lý nhà thông minh khác). Vai trò của **ERP/MES** là truy cập vào nền tảng IoT thông qua API để **xuất và đồng bộ dữ liệu**, phục vụ cho các mục đích phân tích hoặc lưu trữ lớn hơn.
- **Tác nhân phụ:**
 - **Cảm biến/mạch ESP (Sensor/ESP board):** Đây là tác nhân phần cứng, đóng vai trò là "giác quan" của hệ thống. **Cảm biến** có nhiệm vụ đo đạc các thông số thực tế từ môi trường (cụ thể là độ ẩm đất) và gửi dữ liệu đó về cho nền tảng IoT xử lý. Nó là nguồn cung cấp dữ liệu đầu vào cho các chức năng như xem số liệu và tự động hóa.
 - **Máy bơm (Pump):** Đây là tác nhân phần cứng, đóng vai trò là "cơ bắp" của hệ thống. **Máy bơm** nhận lệnh trực tiếp từ hệ thống (do người dùng điều khiển hoặc do quy tắc tự động kích hoạt) và thực hiện hành động vật lý là bơm nước để tưới cây. Nó là đối tượng thực thi đầu ra của hệ thống.

2. Mô tả Use-case

- **Xem số liệu thời gian thực:** Cho phép khách hàng xem ngay lập tức các thông số hiện tại của khu vườn, chẳng hạn như độ ẩm đất và trạng thái bật/tắt của máy bơm, thông qua giao diện ứng dụng.
- **Xem xu hướng và báo cáo lịch sử:** Cung cấp cho khách hàng các biểu đồ hoặc báo cáo về dữ liệu đã được thu thập trong quá khứ (ví dụ: độ ẩm đất trong 7 ngày qua, lịch sử các lần tưới). Chức năng này giúp người dùng hiểu rõ hơn về nhu cầu nước của khu vườn.
- **Điều khiển thiết bị thủ công:** Cho phép khách hàng bật hoặc tắt máy bơm nước ngay lập tức từ xa chỉ bằng một nút nhấn trên ứng dụng, bỏ qua chế độ tự động khi cần thiết.
- **Thiết lập ngưỡng và quy tắc:** Đây là chức năng cốt lõi của chế độ tự động. Nó cho phép khách hàng cài đặt các quy tắc logic cho hệ thống, ví dụ: "Nếu

độ ẩm đất xuống dưới 40%, hãy tự động tưới trong 5 phút".

- **Lập lịch tưới tiêu:** Cho phép khách hàng thiết lập một lịch tưới cố định theo thời gian (ví dụ: tưới vào lúc 6 giờ sáng và 5 giờ chiều mỗi ngày).
- **Thêm và cấu hình thiết bị:** Cung cấp khả năng cho khách hàng tự thêm một thiết bị (cảm biến, máy bơm) mới vào hệ thống và cấu hình các thông số ban đầu cho nó.
- **Nhận cảnh báo:** Hệ thống sẽ tự động gửi thông báo (notification) đến điện thoại của khách hàng khi có sự kiện quan trọng xảy ra, chẳng hạn như độ ẩm đất quá thấp, máy bơm gặp sự cố, hoặc thiết bị mất kết nối.
- **Quản lý người dùng/vai trò:** Cho phép quản trị viên tạo, xóa hoặc chỉnh sửa tài khoản người dùng và phân quyền truy cập cho họ vào hệ thống.
- **OTA firmware update (Cập nhật phần mềm từ xa):** Cung cấp khả năng cho quản trị viên nâng cấp phiên bản phần mềm cho các thiết bị ESP từ xa thông qua mạng internet mà không cần phải can thiệp vật lý.
- **Xuất/Đồng bộ dữ liệu:** Cung cấp một giao diện lập trình ứng dụng (API) để các hệ thống phần mềm bên ngoài có thể tự động truy vấn và lấy dữ liệu (ví dụ: dữ liệu độ ẩm, lịch sử tưới) từ nền tảng IoT để phục vụ cho các mục đích phân tích hoặc lưu trữ riêng.

Chương 4: Các yêu cầu phi chức năng

Phần này đặc tả các ràng buộc về chất lượng, hiệu năng, độ tin cậy và các yếu tố khác để đảm bảo hệ thống tưới cây thông minh hoạt động ổn định và hiệu quả trong thực tế. Các yêu cầu này đóng vai trò quyết định đến trải nghiệm người dùng và sự bền vững của hệ thống.

1. Hiệu năng (Performance)

Hiệu năng của hệ thống đo lường tốc độ và khả năng đáp ứng của các chức năng, ảnh hưởng trực tiếp đến cảm nhận của người dùng.

- **Độ trễ điều khiển (Control Latency):**

- **Yêu cầu:** Thời gian từ lúc người dùng nhấn nút "Bật/Tắt" thủ công trên ứng dụng đến khi máy bơm thực sự thay đổi trạng thái phải **dưới 3 giây**.
- **Diễn giải:** Một độ trễ thấp là cực kỳ quan trọng để mang lại trải nghiệm tương tác tức thì. Nếu thời gian chờ quá lâu, người dùng sẽ có cảm giác hệ thống bị "treo" hoặc không nhận lệnh. Họ có thể nhấn nút nhiều lần, dẫn đến việc gửi đi các lệnh chồng chéo và gây ra hoạt động không mong muốn. Phản hồi nhanh chóng khẳng định với người dùng rằng hệ thống đang hoạt động ổn định và nghe theo lệnh của họ.

- **Tần suất cập nhật dữ liệu (Data Update Frequency):**

- **Yêu cầu:** Dữ liệu độ ẩm từ cảm biến phải được cập nhật lên hệ thống và hiển thị trên ứng dụng với tần suất không ít hơn **15 phút một lần**.
- **Diễn giải:** Tần suất này là một sự cân bằng hợp lý giữa việc cung cấp thông tin kịp thời và việc tối ưu tài nguyên. Cập nhật quá thường xuyên (ví dụ: mỗi giây) sẽ gây tốn pin không cần thiết cho thiết bị, tạo ra lưu lượng mạng lớn và làm đầy cơ sở dữ liệu một cách nhanh chóng. Ngược lại, cập nhật quá thưa (ví dụ: mỗi giờ) có thể khiến người dùng không nắm bắt được những thay đổi đột ngột của điều kiện đất, làm giảm tính hiệu quả của việc giám sát.

2. Bảo mật (Security)

Dù là một hệ thống tại nhà, các yếu tố bảo mật cơ bản là bắt buộc để bảo vệ dữ liệu và sự riêng tư của người dùng, cũng như an toàn cho mạng gia đình.

- **Mã hóa kết nối (Connection Encryption):**

- **Yêu cầu:** Toàn bộ dữ liệu trao đổi giữa thiết bị ESP và Nền tảng IoT (Cloud) phải được mã hóa bằng giao thức **TLS/SSL**. Kết nối Wi-Fi tại nhà phải sử dụng chuẩn bảo mật **WPA2** trở lên.
- **Diễn giải:** Việc mã hóa đảm bảo rằng dữ liệu (ví dụ: thông tin độ ẩm, lệnh điều khiển) không thể bị "đọc trộm" khi đang được truyền trên mạng. Điều này ngăn chặn nguy cơ kẻ xấu nghe lén thông tin hoặc,

nghiêm trọng hơn, lợi dụng lỗ hổng để tấn công vào các thiết bị khác trong cùng mạng Wi-Fi của gia đình.

- **Xác thực (Authentication & Authorization):**

- **Yêu cầu:** Người dùng phải **đăng nhập bằng tài khoản và mật khẩu** để sử dụng ứng dụng. Mỗi thiết bị ESP phải có một **mã xác thực (token) duy nhất** để kết nối với Nền tảng IoT.
- **Diễn giải:** Việc xác thực người dùng đảm bảo chỉ những thành viên được cho phép mới có quyền xem và điều khiển hệ thống. Trong khi đó, mã xác thực thiết bị đóng vai trò như "mật khẩu" của riêng phần cứng, ngăn chặn kẻ xấu tạo ra một thiết bị ảo để gửi dữ liệu giả mạo hoặc cố gắng chiếm quyền điều khiển thiết bị thật.

3. Độ tin cậy (Reliability)

Độ tin cậy đo lường khả năng hoạt động ổn định và tự phục hồi của hệ thống khi gặp sự cố.

- **Khả năng tự phục hồi kết nối (Auto-reconnect):**

- **Yêu cầu:** Khi bị mất kết nối Wi-Fi, thiết bị ESP phải có khả năng **tự động thử kết nối lại** sau mỗi 1-2 phút cho đến khi thành công.
- **Diễn giải:** Sóng Wi-Fi trong gia đình, đặc biệt là ở khu vực sân vườn, có thể không ổn định. Yêu cầu này đảm bảo hệ thống không bị "chết" hoàn toàn chỉ vì một lần rớt mạng tạm thời. Nếu không có cơ chế này, mỗi lần mạng chập chờn, người dùng sẽ phải khởi động lại thiết bị bằng tay, điều này làm mất đi hoàn toàn tính "tự động" và tiện lợi của hệ thống.

- **Khả năng phục hồi sau mất điện (Power Outage Recovery):**

- **Yêu cầu:** Sau khi có điện trở lại, thiết bị ESP phải **tự động khởi động lại, kết nối vào mạng và quay trở lại trạng thái hoạt động** trước đó mà không cần bất kỳ sự can thiệp nào từ người dùng.
- **Diễn giải:** Đây là yêu cầu bắt buộc đối với một hệ thống tự động hóa. Nó đảm bảo rằng sau một sự cố mất điện, hệ thống có thể tự mình tiếp tục nhiệm vụ tưới tiêu theo lịch hoặc theo quy tắc đã định sẵn, duy trì lợi ích "cài đặt và quên đi" (set-and-forget) cho người dùng.

4. Khả năng mở rộng (Scalability)

- **Kiến trúc module (Modular Architecture):**

- **Yêu cầu:** Cấu trúc phần mềm và thiết kế phần cứng phải cho phép **dễ dàng mở rộng** trong tương lai, cụ thể là có thể thêm ít nhất **2-3 cảm biến hoặc máy bơm mới** mà không cần phải thiết kế lại toàn bộ hệ thống.
- **Diễn giải:** Điều này có nghĩa là mã nguồn cần được viết theo dạng các

hàm hoặc lớp có thể tái sử dụng. Ví dụ, việc thêm một cảm biến mới chỉ đơn giản là khai báo một đối tượng cảm biến mới và cấu hình chân cắm, thay vì phải viết lại toàn bộ logic đọc dữ liệu. Việc suy nghĩ trước về khả năng mở rộng sẽ tiết kiệm rất nhiều công sức nếu người dùng muốn nâng cấp hệ thống sau này.

5. Chi phí và Năng lượng (Cost & Power)

- **Chi phí (Cost):**

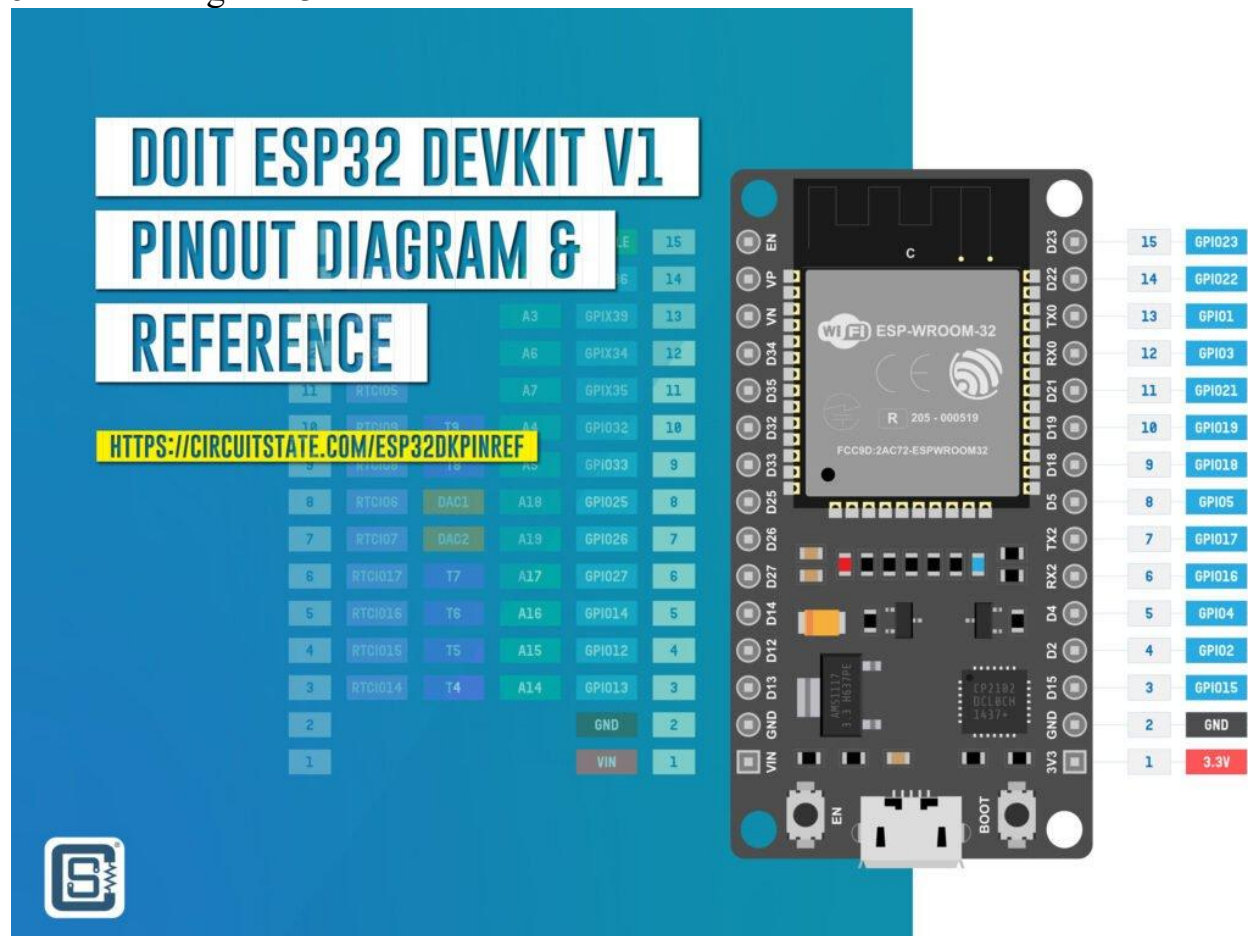
- **Yêu cầu:** Tổng chi phí đầu tư ban đầu (CAPEX) cho phần cứng phải **dưới 500,000 VNĐ**. Chi phí vận hành (OPEX) hàng tháng (tiền điện) phải ở mức không đáng kể.
- **Diễn giải:** Việc tuân thủ ngân sách chặt chẽ là một ràng buộc quan trọng, đảm bảo giải pháp này thực sự là một dự án "nhà làm" (DIY) thiết thực và có tính khả thi cao về mặt tài chính cho một hộ gia đình.

- **Tối ưu năng lượng (Power Optimization):**

- **Yêu cầu:** Phần mềm trên thiết bị ESP phải tận dụng **chế độ ngủ sâu (deep sleep)** giữa các lần đo và gửi dữ liệu.
- **Diễn giải:** Thiết bị không cần phải hoạt động với 100% công suất liên tục. Bằng cách cho vi điều khiển "ngủ" trong phần lớn thời gian và chỉ "thức dậy" vài giây để đo đạc rồi gửi dữ liệu, hệ thống có thể giảm mức tiêu thụ năng lượng xuống hàng chục, thậm chí hàng trăm lần. Điều này đặc biệt quan trọng nếu trong tương lai hệ thống được cấp nguồn bằng pin hoặc năng lượng mặt trời.

Chương 5: Công nghệ sử dụng

5.1 Phần cứng ESP32



+) Vi xử lý: "Bộ não" của hệ thống

- **Chip chính: ESP32-WROOM-32** Đây là module chứa chip ESP32 và các thành phần cần thiết như ăng-ten, bộ nhớ flash. Hầu hết các bo mạch phát triển ESP32 đều sử dụng module này.
- **Lõi xử lý: Dual-core (2 lõi) Tensilica LX6 32-bit** Đây là một trong những ưu điểm vượt trội nhất của ESP32. Việc có 2 lõi xử lý hoạt động song song mang lại lợi ích rất lớn:
 - **Phân luồng tác vụ:** Bạn có thể dành riêng **một lõi** để xử lý các tác vụ nặng và phức tạp về kết nối mạng (Wi-Fi, Bluetooth). **Lõi còn lại** sẽ hoàn toàn tự do để chạy code ứng dụng chính của bạn (đọc cảm biến, điều khiển màn hình, tính toán...).
 - **Hiệu suất cao:** Điều này giúp hệ thống của bạn hoạt động mượt mà, không bị "khựng" hay chậm trễ khi đang thực hiện các tác vụ mạng, một vấn đề thường gặp trên các vi điều khiển đơn lõi như Arduino.
- **Vi xử lý phụ ULP (Ultra Low Power Coprocessor)** Hãy nghĩ về nó như một "bộ não phụ" siêu tiết kiệm điện. Trong khi 2 lõi chính đang "ngủ sâu"

để tiết kiệm năng lượng, ULP có thể âm thầm hoạt động để theo dõi các cảm biến. Nếu có một sự kiện xảy ra (ví dụ: cảm biến phát hiện chuyển động), ULP sẽ "đánh thức" các lõi chính dậy để xử lý. Đây là chìa khóa để tạo ra các thiết bị IoT chạy bằng pin trong thời gian rất dài.

+) Bộ nhớ: Không gian làm việc và lưu trữ

- **SRAM: 520 KB** SRAM là bộ nhớ truy cập ngẫu nhiên, nơi chương trình lưu trữ các biến và dữ liệu tạm thời khi đang chạy. 520KB là một con số **rất lớn** so với các vi điều khiển 8-bit (ví dụ: Arduino Uno chỉ có 2KB). Lượng SRAM dồi dào này cho phép bạn:
 - Xử lý các tác vụ phức tạp, ví dụ như chạy một web server nhỏ ngay trên thiết bị.
 - Làm việc với các bộ dữ liệu lớn, như xử lý âm thanh hoặc thậm chí là hình ảnh từ camera (trên các phiên bản ESP32-CAM).
 - Sử dụng các thư viện lớn mà không lo bị tràn bộ nhớ.
- **Bộ nhớ Flash: 4 MB** Đây là nơi lưu trữ chương trình (code) của bạn. 4MB cung cấp không gian thoải mái cho những dự án lớn, tích hợp nhiều thư viện và cho phép bạn sử dụng tính năng **OTA (Over-the-Air) Update** – cập nhật phần mềm cho thiết bị từ xa qua Wi-Fi mà không cần kết nối cáp USB.

+) Kết nối không dây: Thế mạnh cốt lõi

- **Wi-Fi: 802.11 b/g/n (2.4 GHz)** ESP32 có thể hoạt động ở nhiều chế độ Wi-Fi khác nhau:
 - **Station (STA):** Kết nối vào một mạng Wi-Fi có sẵn (như mạng Wi-Fi nhà bạn) để truy cập internet.
 - **Access Point (AP):** Tự phát ra một mạng Wi-Fi riêng để các thiết bị khác (như điện thoại) có thể kết nối trực tiếp vào nó, rất hữu ích cho việc cấu hình ban đầu.
 - **STA + AP:** Hoạt động ở cả hai chế độ cùng lúc.
- **Bluetooth: v4.2 Classic và BLE** Việc hỗ trợ cả hai chế độ Bluetooth mang lại sự linh hoạt tối đa:
 - **Bluetooth Classic:** Dùng cho các ứng dụng truyền dữ liệu liên tục như loa không dây, tai nghe.
 - **Bluetooth Low Energy (BLE):** Siêu tiết kiệm năng lượng, được thiết kế cho các thiết bị đeo, cảm biến không dây, beacon... chỉ gửi các gói dữ liệu nhỏ và không thường xuyên.

+) GPIO và Giao tiếp Ngoại vi

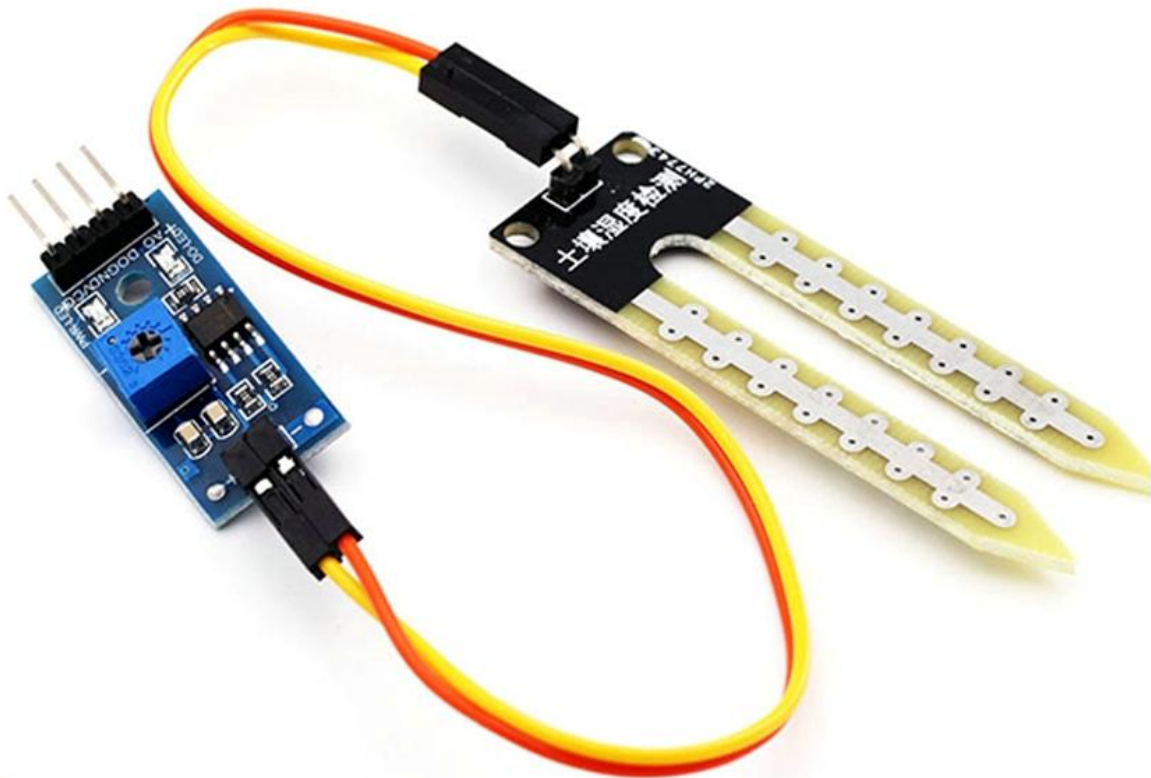
Đây là cách ESP32 tương tác với thế giới bên ngoài.

- **ADC (12-bit):** Có thể đọc tín hiệu analog với độ phân giải 4096 mức khác

nhau, cho phép đo đạc các cảm biến một cách rất chính xác.

- **DAC (8-bit):** Có thể xuất ra tín hiệu điện áp analog thực sự, hữu ích trong các ứng dụng âm thanh hoặc điều khiển analog.
- **Cảm ứng điện dung (Touch Pins):** Cho phép bạn tạo ra các nút bấm cảm ứng chỉ bằng cách nối một miếng kim loại vào các chân này mà không cần thêm bất kỳ linh kiện nào.
- **Các giao tiếp nối tiếp (UART, SPI, I2C):** Là các chuẩn giao tiếp phổ biến nhất để kết nối với hàng ngàn loại cảm biến, màn hình, và các module khác trên thị trường.

5.2. Cảm biến đo độ ẩm đất



Cảm biến phát hiện độ ẩm đất, bình thường đầu ra mức thấp, khi đất thiếu nước

đầu ra sẽ mức cao. Module có thể sử dụng để tưới nước tự động
Độ nhạy của cảm biến độ ẩm đất có thể điều chỉnh được (Bằng cách điều chỉnh
biến trở màu xanh trên board mạch)

Phần đầu dò được cắm vào đất để phát hiện độ ẩm, khi độ ẩm của đất đạt ngưỡng
thiết lập, đầu ra DO sẽ chuyển từ mức thấp lên mức cao.

Thông Số Kỹ Thuật Module Cảm Biến Độ Ẩm Đất:

- Điện áp làm việc 3.3V ~ 5V
- Có lỗ cố định để lắp đặt thuận tiện
- PCB có kích thước nhỏ 3.2 x 1.4 cm
- Sử dụng chip LM393 để so sánh, ổn định làm việc
- Đầu kết nối sử dụng 3 dây
- VCC 3.3V ~ 5V
- GND GND của nguồn ngoài
- DO Đầu ra tín hiệu số (mức cao hoặc mức thấp)
- AO Đầu ra tín hiệu tương tự (Analog)

5.3. Cảm biến đo độ ẩm không khí

Module DHT11

Bộ sản phẩm gồm:

+ Module DHT11

+ Dây nối



META

Linh Kiện Điện Tử

DHT11 là cảm biến nhiệt độ và độ ẩm giao tiếp với 1 chân dữ liệu, DHT11 đo được giá trị độ ẩm từ 20% đến 90%RH và nhiệt độ từ 0°C đến 50°C, độ chính xác: $\pm 5\%RH$ và $\pm 2^\circ C$.

Xin lưu ý chỉ sử dụng cảm biến độ ẩm nhiệt độ DHT11 trong môi trường độ ẩm thuần là hơi nước, các môi trường đặc biệt ủ kín như ủ tỏi đen, ủ yếm khí... sẽ sinh ra nấm và vi khuẩn bám lên bề mặt cảm biến làm hư hỏng cảm biến.

Thông số kỹ thuật:

- Điện áp hoạt động: 5VDC.
- Chuẩn giao tiếp: TTL, 1 wire.
- Khoảng đo độ ẩm: 20%-80%RH sai số $\pm 5\%RH$.

- Khoảng đo nhiệt độ: 0-50°C sai số $\pm 2^{\circ}\text{C}$.
- Tần số lấy mẫu tối đa 1Hz (1 giây / lần).
- Kích thước: 28mm x 12mm x 10mm.
- DHT11 có 4 chân: VCC, DATA, NC, GND
- Module DHT11 đã được gắn sẵn điện trở và led báo nguồn, nên có 3 chân.
- VCC: Nguồn 3.3 - 5.5VDC
- DATA: Chân dữ liệu
- GND: Nối đất, cực âm

5.4. Phần mềm

- 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: JavaScript - 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: JavaScript: 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.