POST AND TELECOMMUNICATIONS INSTITUTE OF TECHNOLOGY

FACULTY OF INFORMATION TECHNOLOGY 1

IOT Project

Hệ thống giám sát chất lượng nước ao & hỗ trợ ra quyết định can thiệp

Lớp	:	E22HTTT
Nhóm	:	5
Tên thành viên	:	Nguyễn Hồng Thăng
		Đỗ Văn Đức
		Tăng Minh Quang

Hanoi, 2025

I. Giới thiệu đề tài

1. Mô tả đề tài

Ngày nay cùng với sự phát triển vượt bậc của khoa học kỹ thuật, **Internet of Things (IoT)** được ứng dụng rộng rãi để giám sát, điều khiển và tự động hóa trong nhiều lĩnh vực của đời sống. Trong đó, **nuôi trồng thủy sản** đang dần số hóa nhằm nâng cao năng suất, giảm chi phí vận hành và hạn chế rủi ro môi trường. Chất lượng nước trong ao (bị ảnh hưởng bởi thời tiết, thức ăn, thói quen vận hành…) thường biến động khó lường; nếu không theo dõi sát sao, người nuôi có thể can thiệp chậm, dẫn đến nước xấu, cá bị stress và tăng chi phí khắc phục.

Từ thực tế đó, nhóm chúng em lựa chọn đề tài **" Hệ thống giám sát chất lượng nước ao & hỗ trợ ra quyết định can thiệp"**. Mục tiêu của đề tài là xây dựng một giải pháp **IoT chi phí thấp**, dễ triển khai cho sinh viên và hộ nuôi nhỏ, giúp **quan sát các chỉ số nước theo thời gian thực**, cảnh báo khi vượt ngưỡng và **hỗ trợ người vận hành ra quyết định** can thiệp kịp thời (như sục khí, đổi nước). Hệ thống hướng tới phục vụ các hộ nuôi gia đình, mô hình thí nghiệm trong trường học, cũng như các cơ sở nuôi thủy sản cần giải pháp giám sát cơ bản nhưng **hiệu quả và tiết kiệm**, góp phần nâng cao chất lượng vận hành và đời sống người nuôi.

2. Mục tiêu đề tài

a. Mục tiêu tổng quát

- **Xây dựng hệ thống IoT chi phí thấp** để **giám sát chất lượng nước ao** theo thời gian thực và **hỗ trợ ra quyết định can thiệp** (sục khí/đổi nước) kịp thời, phù hợp điều kiện triển khai của sinh viên và hộ nuôi nhỏ.
- **Nâng cao tính chủ động** trong vận hành ao nuôi bằng cách theo dõi xu hướng biến động (pH, độ đục, nhiệt độ), cảnh báo vượt ngưỡng và **đề xuất hành động** nhằm giảm rủi ro, tiết kiệm chi phí.

b. Mục tiêu cụ thể

- **Giám sát & cảnh báo**
  - Thu thập liên tục các chỉ số **pH, độ đục, nhiệt độ nước** với chu kỳ 1–5 phút; hiển thị trên **dashboard** thời gian thực.
    Thiết lập **ngưỡng an toàn** và **cảnh báo** khi vượt ngưỡng.
  - **KPI**: độ trễ cảm biến→dashboard ≤ **5–10 giây**; uptime truyền dữ liệu ≥ **95%** thời gian chạy demo.
  - Phát triển mô hình **AI** dự báo các chỉ số dựa trên đặc trưng chuỗi thời gian .
- **Khuyến nghị can thiệp**

- Xây dựng **quy tắc hoặc mô hình AI** đề xuất thời lượng, tự động **sục khí/đổi nước** khi dự báo vượt ngưỡng; có **hysteresis** và **cooldown** để tránh bật/tắt liên tục.

3. Phạm vi triển khai

Số lượng và loại thiết bị

- **01 nút IoT** :thu thập dữ liệu và điều khiển.
- **01 máy chủ cục bộ** chạy broker/DB/dashboard/AI service.

Loại thiết bị & cảm biến/chấp hành:

- **Vi điều khiển:** ESP32 DevKit (Wi-Fi tích hợp).
- **Cảm biến nước:**
  - pH (đầu dò + board khuếch đại loại Gravity-clone).
  - Độ đục (Turbidity – module SEN0189 clone).
  - Nhiệt độ nước (DS18B20 chống nước).
- **Chấp hành:** bơm nước DC 5–12V (mô phỏng đổi nước/sục khí) + relay 2 kênh.
- **Nguồn:** Adapter DC 12V (hoặc 5V cho bơm USB) + buck 5V cho ESP32.

4. Tổng quan phương hướng

Đề tài **"Hệ thống giám sát chất lượng nước ao & hỗ trợ ra quyết định can thiệp"** được nhóm triển khai theo hướng phát triển **một mô hình IoT hoàn chỉnh**, bao gồm cả **phần cứng** (thiết bị đo đạc và điều khiển) và **phần mềm** (ứng dụng giám sát, phân tích và khuyến nghị). Hệ thống hướng đến việc giải quyết bài toán thực tế: người nuôi khó theo dõi liên tục tình trạng nước và thường can thiệp theo lịch cố định, dễ dẫn đến pH/độ đục vượt ngưỡng, chi phí vận hành tăng.

Phương hướng tổng thể của nhóm được chia thành **4 giai đoạn chính:**

Giai đoạn 1 – Nghiên cứu và thiết kế hệ thống:

Nhóm tìm hiểu và đặc tả các thành phần theo **mô hình ba lớp**:

- **Lớp cảm nhận (Perception Layer):** Cảm biến **pH** (đầu dò + board khuếch đại), **Turbidity** (độ đục), **DS18B20** (nhiệt độ nước); **relay** và **bơm DC** mô phỏng can thiệp (đổi nước/sục khí).

- **Lớp mạng (Network Layer): ESP32** kết nối **Wi-Fi**, truyền dữ liệu qua **MQTT** đến máy chủ nội bộ.

- **Lớp ứng dụng (Application Layer):** Dashboard/web trên **ReactJs** để theo dõi chỉ số, nhận **khuyến nghị can thiệp** và thao tác **Áp dụng**.

Giai đoạn 2 – Xây dựng và tích hợp phần cứng:

ESP32 đảm nhiệm đọc cảm biến và điều khiển **relay/bơm**; các đầu dò được lắp vào **bể mô phỏng** (10–50L). Nhóm hiệu chuẩn **pH** (dung dịch 4.01/7.00/10.01), kiểm tra ổn định tín hiệu độ đục và nhiệt độ, bố trí an toàn điện (DC 5–12V, cầu chì, kill-switch). Các thành phần được tích hợp thành **mô hình thực tế** phục vụ thử nghiệm.

Giai đoạn 3 – Phát triển phần mềm và giao tiếp IoT:

Máy chủ cục bộ (laptop) triển khai **Mosquitto (MQTT)**, **InfluxDB/Timescale** để lưu trữ chuỗi thời gian và **ReactJS** làm giao diện. Thiết bị và server giao tiếp qua **MQTT** nhằm đảm bảo tốc độ, ổn định và thời gian thực. Giao diện cung cấp: hiển thị pH/độ đục/nhiệt độ, cấu hình ngưỡng cảnh báo, nhật ký can thiệp, chế độ **AUTO/ECO/MANUAL** và nút **Áp dụng khuyến nghị**.

Giai đoạn 4 – Mở rộng và tích hợp AI:

Hệ thống tích hợp **mô hình AI** huấn luyện trên **dữ liệu công khai** về chất lượng nước ao (có pH/độ đục/nhiệt độ). Mô hình khai thác **đặc trưng chuỗi thời gian** (lag, rolling, giờ/ngày) để **dự báo pH/độ đục ngắn hạn (t + 1–3–6 giờ)** và sinh **khuyến nghị thời lượng can thiệp** (bơm/sục khí). Kết quả được trả về ứng dụng để người vận hành theo dõi và quyết định áp dụng.

Định hướng phát triển:

- Hoàn thiện mô hình thử nghiệm, kiểm tra **độ ổn định truyền dữ liệu** và **độ trễ cảm biến → dashboard** trong mạng nội bộ.
- Tối ưu **giao diện người dùng (UI/UX)**, bổ sung biểu đồ xu hướng và lịch sử can thiệp.
- **Mở rộng đa thiết bị** trên cùng mạng LAN, cho phép quản lý nhiều ao/nút cảm biến đồng thời.
- Nâng cấp lớp AI: tinh chỉnh mô hình, bổ sung **giải thích SHAP**, và nghiên cứu tích hợp **điều khiển bằng giọng nói** hoặc ứng dụng di động trong giai đoạn sau.

Với phương hướng trên, nhóm hướng tới xây dựng một hệ thống **IoT thông minh, chi phí thấp, ổn định và có khả năng mở rộng**, góp phần đưa công nghệ IoT vào đời sống theo hướng **tiện ích và tự động hóa** trong lĩnh vực nuôi trồng thủy sản.

II. Các công nghệ, Lý thuyết áp dụng

A. Phần cứng

1. ESP32 DevKit

ESP32 là vi điều khiển tích hợp **Wi-Fi** (2.4 GHz) và **Bluetooth BLE**, phù hợp cho các hệ thống IoT cần thu thập dữ liệu, xử lý nhẹ tại biên (edge) và giao tiếp thời gian thực.

Thành phần/đặc điểm chính:

- **CPU dual-core 32-bit**, xung đến 240 MHz; **RAM ~520 KB**, **Flash** on-board.
- **Wi-Fi 802.11 b/g/n**, **BLE**; **GPIO đa dụng**, **ADC/DAC**, **UART/SPI/I²C**.
- **Nguồn 3.3 V**, nút **Boot/Reset**; lập trình qua **Arduino IDE** hoặc **ESP-IDF**.

Vai trò trong hệ thống:

- Thiết bị trung tâm của **lớp cảm nhận**: đọc cảm biến pH/độ đục/DS18B20, lọc tín hiệu (median/EMA), đóng/ngắt **relay bơm**, và **publish/subscribe MQTT** tới máy chủ nội bộ.

2. Cảm biến pH (đầu dò + board khuếch đại loại "Gravity-clone")

Cảm biến pH hoạt động dựa trên điện thế điện cực tỉ lệ với độ hoạt động ion H^+ trong nước. Tín hiệu đầu dò rất nhỏ (mV) nên cần **bo khuếch đại/điều hòa tín hiệu** trước khi vào ADC.

Thành phần:

- **Điện cực pH** chuẩn **BNC**; **board khuếch đại** xuất **analog** ổn định.
- **Dung dịch chuẩn pH 4.01/7.00/10.01** để **hiệu chuẩn 2–3 điểm**.

Vai trò:

- Chỉ số **cốt lõi** phản ánh môi trường nước; dùng để **cảnh báo vượt ngưỡng** và làm **đặc trưng** cho mô hình dự báo/khuyến nghị.

3. Cảm biến độ đục (Turbidity – module SEN0189 "clone")

Đo **độ tán xạ ánh sáng** do hạt lơ lửng tạo ra; xuất **điện áp analog** xấp xỉ mức **NTU**.

Thành phần: đầu dò quang + mạch so-khớp.

Vai trò:

- Nhạy với **thức ăn dư/cặn** → dùng để kích hoạt **đổi nước/sục khí** và làm biến quan trọng trong **dự báo đục t+h**.

4. Cảm biến nhiệt độ nước DS18B20 (chống nước)

Cảm biến số 1-Wire, sai số nhỏ, dễ đi dây dài.

Vai trò:

- Ảnh hưởng đến **hoà tan oxy** và động học hóa-sinh; là **đặc trưng** quan trọng khi dự báo pH/độ đục và đề xuất thời lượng can thiệp.

5. Khối chấp hành & nguồn

- **Relay 2 kênh** (5–12 V): điều khiển **bơm nước DC**/thiết bị mô phỏng sục khí.
- **Bơm DC 5–12 V + ống silicon**: mô phỏng **đổi nước** hoặc **sục khí**.
- **Nguồn DC 12 V** (cho bơm) + **buck 5 V** cho ESP32; **cầu chì/kill-switch** để an toàn.

Vai trò:

- Thực thi **khuyến nghị can thiệp** (bơm X phút); minh họa rõ ràng hiệu quả kiểm soát chất lượng nước.

B. Phần mềm

1) MQTT (Mosquitto – Publish/Subscribe)

Giao thức nhẹ cho IoT, độ trễ thấp, tối ưu LAN.
Vai trò:

- ESP32 publish telemetry: agrosense/{deviceId}/telemetry (pH, turb, temp, pump_state).
- ESP32 subscribe lệnh: agrosense/{deviceId}/cmd (PUMP_ON, dur_s) và phản hồi .../ack.
- Broker Mosquitto chạy trên laptop; có thể bật TLS/ mật khẩu broker.

2) Backend NodeJS (Express)

- Xây dựng RESTful API cho cấu hình ngưỡng, quản lý thiết bị, lịch can thiệp (nếu bật tính năng).
- MQTT client trong NodeJS để nhận telemetry, phát lệnh, và đẩy khuyến nghị tới topic .../recommendations.
- Xử lý nghiệp vụ: cảnh báo, rule cục bộ phía server, ghi log sự kiện; JWT cho xác thực API (nếu mở ra ngoài LAN).
- *(Tuỳ chọn)* Docker Compose gói Mosquitto + Node + Mongo/Influx cho dễ triển khai.

3) Cơ sở dữ liệu MongoDB

- Lưu **cấu hình**, **thiết bị**, **lịch**, **sự kiện/cảnh báo**, **khuyến nghị**; lưu **telemetry** có thể dùng **Mongo Timeseries** hoặc kết hợp InfluxDB nếu cần đồ thị nặng.

- **Mongoose** định nghĩa schema; **TTL/retention** cho dữ liệu thô; script **export CSV** phục vụ huấn luyện/đánh giá.

4) Frontend ReactJS

- Dashboard hiển thị realtime **pH/đục/nhiệt độ**, thẻ cảnh báo, nút **"Áp dụng khuyến nghị"** và **điều khiển thủ công**.

- Gọi **API Express** (GET/POST/PUT/DELETE) và/hoặc subscribe **MQTT qua bridge/WebSocket** để nhận cập nhật tức thời.

- UX: màu ngưỡng, biểu đồ zoom/pan, lịch sử can thiệp.

C. Tính năng AI

1) Mô hình & suy luận trong NodeJS

- **Bài toán:** dự báo ngắn hạn **pH_t+h**, **turbidity_t+h** (h = 1–3–6 giờ) từ đặc trưng **time-series**:

    - **Lag** (t−1, t−6, t−12…), **rolling mean/max/min/var** (3h/6h/24h), **hour-of-day**, **day-of-week**, **tổng phút bơm** gần đây.

- **Mô hình sử dụng trong Node:**

    - **ElasticNet/Linear Regression** (thư viện ML cho JS như ml-regression, scikit.js, hoặc **tự triển khai** từ hệ số huấn luyện).

    - **RandomForest/XGBoost**: có thể **xuất mô hình** (tree/coef) sang **JSON** và viết **trình suy luận** JS nhẹ, hoặc dùng các thư viện JS tương đương (nếu phù hợp).

- **Quy trình khuyến nghị:** nếu dự báo **vượt ngưỡng**, tính **phút bơm đề xuất X** với **min/max** và **cooldown**; trả {water_minutes, reason, confidence} → **MQTT**.

2) Huấn luyện mô hình

- Dùng **dữ liệu công khai** (có pH/đục/nhiệt) để **tiền huấn luyện** bằng Python/Notebook **ngoại tuyến** *hoặc* bằng NodeJS scripts.

- **Xuất tham số** (hệ số ElasticNet / cây RF) → **JSON** → **NodeJS** nạp để **suy luận online**.

- **Đánh giá:** chia **train/test theo thời gian**, báo cáo **RMSE/R^2**; **SHAP/Permutation Importance** (tính offline) để minh bạch biến ảnh hưởng.

3) Bảo đảm an toàn & minh bạch

- **Fallback rule** khi thiếu dữ liệu hoặc confidence thấp.

- Nhật ký: dự báo, ngưỡng, khuyến nghị, kết quả thực thi, **thời gian hồi về ngưỡng** để phục vụ A/B.

III.Các tính năng dự kiến triển khai

1. Yêu cầu chức năng

Chức năng giám sát thời gian thực

- Thu thập và hiển thị liên tục các chỉ số **pH**, **độ đục (turbidity)**, **nhiệt độ nước**.
- Dữ liệu đi theo luồng: **ESP32 → MQTT Broker → NodeJS (Express) → MongoDB → ReactJS Dashboard**.
- Xem **xu hướng theo giờ/ngày**, tải **CSV**, hiển thị **ngưỡng màu** và thẻ cảnh báo.

Chức năng điều khiển thủ công

- Người dùng bấm **"Bơm ngay / Sục khí ngay"** trên giao diện.
- Lệnh đi **Frontend → NodeJS → MQTT topic agrosense/{deviceId}/cmd**.
- Thiết bị thực thi qua **relay** và phản hồi **ack** (thành công/thất bại, thời lượng).

Chức năng khuyến nghị can thiệp (AI Baseline – không DL)

- Backend NodeJS dự báo **pH/đục tại t+h (1/3/6 giờ)** từ **đặc trưng time-series** (lag/rolling, giờ/ngày).
- Khi dự báo vượt ngưỡng, hệ thống sinh **đề xuất** {water_minutes: X, reason, confidence}.

- Người vận hành có thể **Áp dụng** ngay; kết quả được **log** để đánh giá.

Chức năng lịch can thiệp

- Thiết lập lịch **đổi nước/sục khí** định kỳ (giờ/phút, thời lượng).
- Lịch lưu trên **MongoDB** và **đẩy xuống thiết bị**; thiết bị vẫn chạy lịch **cục bộ** khi mất mạng.

Chức năng cảnh báo & thông báo

- Cảnh báo khi **vượt ngưỡng**, **cảm biến lỗi**, **bơm không phản hồi**, **mất kết nối**.
- Hiển thị banner trên dashboard; tùy chọn gửi **email/Web Push** trong LAN.

Chức năng cập nhật & giám sát thiết bị

- Hiển thị **tình trạng kết nối**, **điện áp nguồn** (nếu đo), số lần reconnect.
- **OTA** firmware ESP32 (nếu kịp); lưu **nhật ký hoạt động** (lệnh, lý do, kết quả).

2. Yêu cầu phi chức năng

Hiệu năng

- Thời gian cảm biến → dashboard: **≤ 5–10 giây** trong LAN.
- Chu kỳ cập nhật telemetry: **1–5 phút** (cấu hình được).

Độ tin cậy

- Uptime truyền dữ liệu mục tiêu **≥ 95%** trong phiên demo.
- Khi mất Wi-Fi, thiết bị chạy **rule cục bộ** (ngưỡng an toàn) và **buffer** dữ liệu ngắn để đồng bộ lại.

Bảo mật

- MQTT broker có **mật khẩu/ACL**; ưu tiên **TLS (WSS/HTTPS)** nếu kịp.
- Mỗi thiết bị có **Device ID** và **device token**; tài khoản dashboard dùng **JWT** (nếu mở API).

Khả năng mở rộng

- Quản lý **nhiều nút** (≥ 3–5) theo topic agrosense/{deviceId}/....

- Tách **Broker / NodeJS / MongoDB** thành dịch vụ riêng (Docker Compose) khi cần.

Tiết kiệm năng lượng

- ESP32 có **light sleep** (tùy tiến độ); bơm/relay chỉ bật khi cần.

Trải nghiệm người dùng

- Giao diện **trực quan**, biểu đồ mượt, ngưỡng màu, thẻ cảnh báo rõ ràng, nút **Áp dụng** một chạm.

Khả năng bảo trì

- Hỗ trợ **OTA** (nếu kịp), **log lỗi**, công cụ **chẩn đoán** (ping thiết bị, xem last seen).

3. Yêu cầu giao tiếp

Thiết bị ↔ Máy chủ (IoT – Server) qua MQTT

- **Broker:** Mosquitto trong LAN.
- **Topic & payload (JSON):**
    - agrosense/{deviceId}/telemetry → {device_id, ts, ph, turb, temp, pump_state}
    - agrosense/{deviceId}/cmd → {action:"PUMP_ON", dur_s:120}
    - agrosense/{deviceId}/ack → {cmd_id, status, ts_start, ts_end}
    - agrosense/{deviceId}/alerts → {type, detail, ts} (thiết bị/Server phát hiện lỗi)
    - agrosense/{deviceId}/recommendations (Server → UI) → {water_minutes, reason, confidence, ts}

- **QoS:** 1 cho lệnh và ack; 0/1 cho telemetry.
- **Reconnect:** tự động; hàng đợi ngắn khi gián đoạn.

Máy chủ ↔ Giao diện (Server – Frontend)

- **RESTful API (NodeJS/Express):** cấu hình ngưỡng, lịch, người dùng, thiết bị.
- **WebSocket/MQTT bridge:** đẩy khuyến nghị & cảnh báo realtime tới ReactJS.
- **Xác thực: JWT** cho API (nếu mở rộng ngoài LAN).
- **Định dạng:** JSON.

Máy chủ ↔ Cơ sở dữ liệu (Server – Database)

- **MongoDB** (có thể dùng **timeseries collection** cho telemetry).
- **Collections:** devices, configs, telemetry, commands, recommendations, alerts, users.
- Truy vấn/ghi **async/await**; **TTL/retention** cho dữ liệu thô; endpoint **export CSV**.

Bảo mật giao tiếp

- Dùng **HTTPS/WSS** (TLS) khi có thể; **ACL** theo deviceId; **rate-limit** API; băm mật khẩu.

4. Yêu cầu hoạt động

Mạng & điện

- Wi-Fi 2.4 GHz LAN; uplink tối thiểu **1 Mbps**.

- Thiết bị **tự tái kết nối**; nguồn **DC 5–12V**; **kill-switch/cầu chì** cho bơm.

Phần mềm

- **NodeJS ≥ 18**, **MongoDB ≥ 6**; Mosquitto broker; ReactJS trên trình duyệt hiện đại.

- (Tuỳ chọn) **Docker Compose** để khởi chạy các dịch vụ đồng bộ.

5. Yêu cầu về thử nghiệm

Thử nghiệm phần mềm (ReactJS + NodeJS + MongoDB)

- UI: hiển thị biểu đồ pH/đục/nhiệt độ, thẻ cảnh báo, nút **Áp dụng** hoạt động.
- API: CRUD cấu hình/ngưỡng/lịch; kiểm tra **quyền truy cập/JWT** (nếu bật).
- Tải nhẹ: mô phỏng nhiều bản ghi/thiết bị, đảm bảo phản hồi mượt.

Thử nghiệm giao tiếp IoT (MQTT)

- Lệnh **PUMP_ON** đến ESP32 **≤ 2 giây** trong LAN.
- Mất mạng: thiết bị chạy **rule cục bộ**, sau đó **đồng bộ lại**; broker từ chối **token sai**.

Thử nghiệm AI Baseline (không DL)

- Huấn luyện ngoại tuyến trên **dataset công khai**; nạp mô hình/ hệ số vào NodeJS.
- Báo cáo **RMSE/R²** cho h=1/3/6 giờ; kiểm tra **khuyến nghị hợp lý** (ví dụ: đục tăng nhanh → đề xuất tăng phút bơm).
- **A/B demo**: lịch cố định vs khuyến nghị → so sánh **% thời gian vượt ngưỡng**, **tổng phút bơm/ngày**.

IV.Nhiệm vụ và kế hoạch

1.Nhiệm vụ

Tăng Minh Quang: Hardware + IoT

- ESP32 firmware và MQTT
- Cảm biến (pH, độ đục, nhiệt độ)
- Relay và bơm điều khiển

Nguyễn Hồng Thăng: Backend + AI

- NodeJS API và MQTT broker
- MongoDB

- Mô hình AI dự báo và khuyến nghị

Đỗ Văn Đức: Frontend + Integration

- ReactJS dashboard
- Tích hợp API và MQTT
- UI/UX và testing

2.Kế hoạch

TUẦN 1: Nghiên cứu & Setup

- Thành viên 1: Nghiên cứu ESP32 và cảm biến, vẽ sơ đồ mạch, mua linh kiện
- Thành viên 2: Setup môi trường (NodeJS, MongoDB, Mosquitto), thiết kế database schema, tìm dataset
- Thành viên 3: Thiết kế mockup UI, setup project ReactJS

TUẦN 2: Phát triển Cơ bản

- Thành viên 1: Lắp ráp phần cứng, đọc cảm biến, test relay, bắt đầu MQTT
- Thành viên 2: Setup MQTT broker, tạo API cơ bản, kết nối MongoDB, xử lý dataset
- Thành viên 3: Tạo components UI cơ bản (hiển thị chỉ số, nút điều khiển)

TUẦN 3: Tích hợp IoT

- Thành viên 1: Hoàn thiện MQTT 2 chiều (publish telemetry, subscribe command)
- Thành viên 2: API CRUD hoàn chỉnh, logic cảnh báo, feature engineering cho AI
- Thành viên 3: Tích hợp API, hiển thị data thật, biểu đồ realtime

TUẦN 4: AI & Khuyến nghị

- Thành viên 1: Tối ưu firmware, test độ ổn định, thu thập data
- Thành viên 2: Huấn luyện mô hình dự báo, tích hợp AI vào backend, API recommendations
- Thành viên 3: UI khuyến nghị, nút "Áp dụng", cải thiện UX

TUẦN 5: Hoàn thiện & Testing

- Thành viên 1: Hiệu chuẩn cảm biến, xử lý edge cases, test reliability
- Thành viên 2: Tối ưu API, xử lý lỗi, logging, test AI accuracy

- Thành viên 3: Hoàn thiện UI/UX, test tích hợp end-to-end, fix bugs

TUẦN 6: Demo & Báo cáo

- Thành viên 1: Chuẩn bị hardware demo, kiểm tra kết nối
- Thành viên 2: Chuẩn bị slide phần backend/AI, test scenario demo
- Thành viên 3: Video demo, tài liệu hướng dẫn, slide thuyết trình, rehearsa**l**