

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH



**BÁO CÁO ĐỒ ÁN  
THIẾT KẾ LUẬN LÝ**

**HỆ THỐNG GIÁM SÁT MÔI TRƯỜNG**

Sinh viên thực hiện	MSSV
Phan Trần Nguyên Phúc	2212643
Nguyễn Ngọc Phú	2212588
Hà Minh Nguyên	2212288
Phan Châu Phong	2212563

TP. Hồ Chí Minh, 2024

# Mục lục

<b>1</b>	<b>GIỚI THIỆU CHỦ ĐỀ</b>	<b>3</b>
<b>2</b>	<b>YÊU CẦU VẬN HÀNH HỆ THỐNG</b>	<b>3</b>
2.1	Yêu Cầu Chức Năng	3
2.2	Yêu Cầu Phi Chức Năng	5
<b>3</b>	<b>THIẾT BỊ VÀ CẢM BIẾN SỬ DỤNG</b>	<b>6</b>
3.1	DHT-11	6
3.2	MQ-2	6
3.3	Buzzer	6
3.4	Cảm biến lửa	6
3.5	Quạt PC PWM	6
3.6	ESP32	7
3.7	Bảng mạch mở rộng	7
3.8	Mạch chuyển mức tín hiệu 4 kênh	7
3.9	Đèn Led	7
<b>4</b>	<b>USE CASE</b>	<b>8</b>
4.1	Hệ thống giám sát môi trường cho cây trồng mini	8
4.2	Hệ thống giám sát môi trường trong xưởng	9
4.3	Hệ thống giáo dục IOT	10
4.4	Hệ thống an toàn bếp thông minh	11
4.5	Hệ thống giám sát kho lạnh	12
<b>5</b>	<b>SƠ ĐỒ KỸ THUẬT PHẦN CỨNG:</b>	<b>13</b>
5.1	Tổng Quan Hệ Thống	13
5.2	Sơ Đồ Khối Phần Cứng	13
5.3	Sơ Đồ Cấu Hình Chân	14
5.4	Sơ Đồ Dòng Dữ Liệu	14
5.5	Sơ Đồ Logic Điều Khiển Quạt	15
5.6	Sơ Đồ Giao Tiếp MQTT	15
<b>6</b>	<b>KỸ THUẬT PHẦN MỀM</b>	<b>16</b>
6.1	Tổng quan	16
6.2	Thiết kế tổng quan của hệ thống Dashboard	16



6.3	Thiết kế tổng thể Website . . . . .	18
<b>7</b>	<b>TRỰC QUAN SẢN PHẨM . . . . .</b>	<b>19</b>
7.1	Node-red Dashboard . . . . .	19
7.2	Hệ thống Website . . . . .	20
<b>8</b>	<b>KHẢ NĂNG MỞ RỘNG . . . . .</b>	<b>23</b>
<b>9</b>	<b>KẾT LUẬN . . . . .</b>	<b>24</b>

# 1 GIỚI THIỆU CHỦ ĐỀ

Ngày nay, việc giám sát môi trường trong không gian sống và làm việc trở thành nhu cầu cấp thiết nhằm đảm bảo sức khỏe và an toàn cho con người. Hệ thống giám sát môi trường thông minh được đề xuất là một giải pháp toàn diện, phù hợp để theo dõi các chỉ số quan trọng như nhiệt độ, độ ẩm, và nồng độ khí gas. Với việc sử dụng vi điều khiển ESP32 làm trung tâm điều khiển, hệ thống không chỉ đo lường các thông số này mà còn có khả năng cho phép người dùng điều khiển các thiết bị từ xa và tự động phản ứng khi phát hiện các điều kiện nguy hiểm.

Cụ thể, hệ thống được trang bị cảm biến nhiệt độ, độ ẩm và khí gas để liên tục giám sát môi trường. Khi nhiệt độ vượt ngưỡng cài đặt hoặc phát hiện khí gas vượt mức an toàn, quạt PC điều khiển qua tín hiệu PWM sẽ tự động kích hoạt để loại bỏ khí độc. Đồng thời, thông báo cảnh báo được gửi trực tiếp đến người dùng qua các ứng dụng, giúp họ kịp thời có biện pháp xử lý. Hệ thống hỗ trợ hiển thị thông số trên màn hình LCD 16x2 (tùy chọn), mang lại sự tiện lợi trong việc theo dõi. Hệ thống còn cho phép người dùng điều khiển các thiết bị chiếu sáng, điều chỉnh nhiệt độ, ánh sáng từ xa.

Với các tính năng nổi bật và dễ dàng truy cập, hệ thống giám sát môi trường này không chỉ phù hợp cho các hộ gia đình mà còn có thể ứng dụng trong các nhà xưởng, văn phòng hay các không gian công nghiệp.

# 2 YÊU CẦU VẬN HÀNH HỆ THỐNG

## 2.1 Yêu Cầu Chức Năng

### 2.1.1 Giám Sát và Điều Khiển Cốt Lõi

Hệ thống cần thực hiện giám sát tự động các tham số môi trường như nhiệt độ, độ ẩm và nồng độ khí trong thời gian thực. Cụ thể:

- Giám sát nhiệt độ, độ ẩm và nồng độ khí liên tục.
- Kích hoạt quạt hút tự động khi:
  - Nồng độ khí vượt ngưỡng cho phép.
  - Nhiệt độ vượt ngưỡng cài đặt.
- Hỗ trợ điều khiển ánh sáng từ xa với chức năng bật/tắt.
- Cung cấp tính năng cấu hình ngưỡng nhiệt độ có thể tùy chỉnh.

### 2.1.2 AI và Các Tính Năng Nâng Cao

Hệ thống cần tích hợp các tính năng dựa trên trí tuệ nhân tạo (AI) và các công nghệ tiên tiến để tối ưu hóa hiệu suất và khả năng dự đoán:

- Áp dụng các thuật toán học máy để xây dựng hệ thống dự đoán thông minh:
- Thuật toán hồi quy tuyến tính.

- Thuật toán rừng ngẫu nhiên.
- Dự đoán xu hướng nhiệt độ và độ ẩm trong tương lai.
- Triển khai hệ thống phát hiện cháy bằng camera.
- Phân tích dữ liệu môi trường và dự đoán các xu hướng tiềm năng.

### 2.1.3 Quản lý Dữ liệu và Giao tiếp

Hệ thống sẽ quản lý và lưu trữ dữ liệu một cách hiệu quả, đảm bảo khả năng truy xuất, phân tích và giao tiếp giữa các thành phần.

#### Node-RED Dashboard:

- **Nhận và xử lý cảnh báo:**
  - Node-RED nhận dữ liệu được gửi từ ESP32 lên Hivemq và xử lý thông qua các node để gửi cảnh báo nếu cần.
  - Xử lý các cảnh báo như:
    - \* Gửi email khi phát hiện các sự kiện bất thường (nồng độ khí cao, cháy, nhiệt độ quá cao).
    - \* Thực hiện các hành động khác (nếu có) dựa trên cảnh báo.

#### Hệ thống Website:

- **Lưu trữ dữ liệu:**
  - Dữ liệu người dùng: SQLite3
  - Dữ liệu cảm biến: MongoDB
  - Dữ liệu lịch sử: MongoDB
- **Truy cập dữ liệu:**
  - Cung cấp giao diện web để người dùng tương tác và xem dữ liệu.
  - Sử dụng WebSocket để cập nhật dữ liệu thời gian thực từ các cảm biến.

#### Giao tiếp giữa các thành phần:

- **ESP32 → HiveMQ → Node-RED:** Sử dụng MQTT để truyền tải dữ liệu cảm biến và cảnh báo.
- **Node-RED → Hệ thống Website:** Sử dụng WebSocket để cập nhật dữ liệu thời gian thực lên giao diện web.

### 2.1.4 Tính Năng Giao Diện Người Dùng

Hệ thống cần cung cấp giao diện người dùng rõ ràng, dễ sử dụng, hỗ trợ các chức năng tương tác:

- Hiển thị các kết quả thời gian thực trên Node-RED DashBoard, bao gồm: Biểu đồ tương tác, Nhật ký dữ liệu.

- Triển khai ứng dụng web thông qua giao thức HTTP bao gồm: Hiển thị kết quả, biểu đồ, Phân quyền truy cập, Dự đoán thời tiết tương lai, Điều khiển thiết bị.

## 2.2 Yêu Cầu Phi Chức Năng

### 2.2.1 Hiệu Suất và Độ Tin Cậy

Hệ thống cần duy trì độ tin cậy cao và hoạt động ổn định trong suốt quá trình vận hành:

- Đảm bảo hệ thống hoạt động liên tục 24/7 với thời gian ngừng hoạt động tối thiểu.
- Đảm bảo độ trễ kích hoạt thiết bị dưới 5 giây.
- Kết nối với liên tục với máy chủ trong vòng 2s.
- Cập nhật dữ liệu trên bảng điều khiển trong vòng 500ms.
- Hỗ trợ kết nối socket hiệu quả cho việc cập nhật dữ liệu thời gian thực.

### 2.2.2 Lưu Trữ Dữ Liệu và An Ninh

Hệ thống cần đảm bảo lưu trữ dữ liệu an toàn và khả năng mở rộng:

- Sử dụng kiến trúc cơ sở dữ liệu kép:
  - SQLite3 cho quản lý người dùng.
  - MongoDB cho dữ liệu cảm biến.
- Lưu trữ ít nhất 180 ngày dữ liệu lịch sử.
- Đảm bảo khả năng mở rộng hệ thống cơ sở dữ liệu (tối thiểu 500MB dung lượng).
- Phân chia dữ liệu an toàn giữa các phần dành cho người dùng và quản trị viên.

### 2.2.3 Trải Nghiệm Người Dùng

Giao diện người dùng cần đơn giản, dễ sử dụng và cung cấp các tính năng tương tác trực quan:

- Giao diện bảng điều khiển và website trực quan với các tính năng:
  - Hiển thị tham số rõ ràng.
  - Biểu đồ và báo cáo tương tác.
  - Cảnh báo có thể tùy chỉnh.
- Hỗ trợ các lựa chọn tương tác vật lý:
  - Điều khiển từ xa.
  - Giao diện LCD.
  - Quạt gió tự động.

## 3 THIẾT BỊ VÀ CẢM BIẾN SỬ DỤNG

### 3.1 DHT-11

- **Ứng dụng:** Đo nhiệt độ và độ ẩm, gửi dữ liệu đến hệ thống IoT. Nhiệt độ và độ ẩm là các thông số quan trọng để theo dõi môi trường trong nhà xưởng hoặc không gian giám sát. Các dữ liệu này cũng được sử dụng để đưa ra cảnh báo hoặc điều chỉnh thiết bị (như quạt).
- **Đầu vào:** Nhiệt độ và độ ẩm không khí - cảm biến được lắp đặt ở vị trí thông thoáng, tránh tác động trực tiếp từ nguồn nhiệt.
- **Đầu ra:** Giá trị nhiệt độ và độ ẩm, gửi đến bộ điều khiển ESP32 để chuyển tiếp đến MQTT broker.

### 3.2 MQ-2

- **Ứng dụng:** Phát hiện nồng độ khí gas. Đây là chỉ số chính để kích hoạt quạt hút khí hoặc phát cảnh báo nếu vượt ngưỡng an toàn.
- **Đầu vào:** Nồng độ khí gas trong không khí - cảm biến được lắp đặt ở vị trí dễ dàng phát hiện khí rò rỉ hoặc khói.
- **Đầu ra:** Giá trị tương tự (analog) phản ánh nồng độ khí, được ESP32 xử lý và gửi đến server IoT.

### 3.3 Buzzer

- **Ứng dụng:** Tạo cảnh báo âm thanh khi có sự cố (như phát hiện khí gas, lửa, hoặc nhiệt độ cao bất thường).
- **Đầu vào:** Tín hiệu kích hoạt từ bộ điều khiển ESP32.
- **Đầu ra:** Âm thanh cảnh báo rõ ràng, báo hiệu cho người dùng ngay lập tức.

### 3.4 Cảm biến lửa

- **Ứng dụng:** Phát hiện tia lửa hoặc ánh sáng phát ra từ đám cháy. Đây là thành phần quan trọng để cảnh báo nguy cơ hỏa hoạn trong không gian giám sát.
- **Đầu vào:** Ánh sáng hoặc tia lửa - cảm biến được lắp ở vị trí thoáng, dễ quan sát khu vực cần giám sát.
- **Đầu ra:** Tín hiệu kỹ thuật số (HIGH hoặc LOW), được ESP32 sử dụng để gửi cảnh báo khẩn cấp.

### 3.5 Quạt PC PWM

- **Ứng dụng:** Hút khí độc ra ngoài khi phát hiện nồng độ khí gas vượt ngưỡng hoặc khi nhiệt độ trong không gian giám sát quá cao.

- **Đầu vào:** Tín hiệu điều khiển PWM từ ESP32.
- **Đầu ra:** Không có đầu ra điện tử - chỉ tạo luồng khí để thông gió.

### 3.6 ESP32

- **Ứng dụng:** Là trung tâm điều khiển chính, nhận dữ liệu từ các cảm biến (DHT11, MQ-2, cảm biến lửa), xử lý thông tin và gửi lên MQTT broker (HiveMQ). Đồng thời, kích hoạt thiết bị đầu ra (quạt, buzzer) khi có sự kiện cần xử lý.
- **Đầu vào:** Dữ liệu từ cảm biến và lệnh điều khiển từ Node-RED (qua giao thức MQTT).
- **Đầu ra:** Điều khiển quạt, buzzer, và gửi dữ liệu đến dashboard trên Node-RED.

### 3.7 Bảng mạch mở rộng

- Là xương sống kết nối giữa Microbit và các cảm biến khác.

### 3.8 Mạch chuyển mức tín hiệu 4 kênh

- Hệ thống này hoạt động liên tục và cố định, do đó, một nguồn cung cấp điện ổn định và liên tục được ưu tiên hơn so với việc sử dụng pin. Đây là mạch chuyển mức tín hiệu logic 3.3V  $\Leftrightarrow$  5V 2 chiều 4 kênh. Mạch chuyển tốc độ cao nên có thể dùng với giao tiếp UART, SPI, I2C.

### 3.9 Đèn Led

- Đại diện cho thiết bị chiếu sáng hoặc các thiết bị được điều khiển



## 4 USE CASE

### 4.1 Hệ thống giám sát môi trường cho cây trồng mini

Use case ID	1
Use case name	Hệ thống giám sát môi trường cho cây trồng mini
Actor	người dùng, ESP32 và thiết bị (DHT-11, MQ-2, Cảm biến lửa, Buzzer)
Mô tả	Hệ thống giám sát các chỉ số môi trường như nhiệt độ, độ ẩm, khí gas, và phát hiện lửa. Khi các chỉ số này vượt ngưỡng cảnh báo, hệ thống sẽ thông báo đến người dùng qua ứng dụng hoặc buzzer.
Điều kiện tiên quyết	<ul style="list-style-type: none"><li>- Hệ thống đang hoạt động bình thường.</li><li>- Các cảm biến đã được kết nối và cấu hình với ESP32.</li><li>- Người dùng đã cài đặt ứng dụng giám sát trên điện thoại hoặc hệ thống Node-RED đã được thiết lập để nhận thông báo.</li></ul>
Quy trình hoạt động	<ul style="list-style-type: none"><li>- Người dùng mở ứng dụng hoặc giao diện giám sát (Node-RED Dashboard).</li><li>- Hệ thống bắt đầu thu thập dữ liệu từ các cảm biến (nhiệt độ, độ ẩm, khí gas, cảm biến lửa).</li><li>- Dữ liệu từ các cảm biến được xử lý và phân tích.</li><li>- Nếu các chỉ số nhiệt độ hoặc độ ẩm vượt quá ngưỡng cài đặt, hoặc nếu phát hiện khí gas hoặc lửa, hệ thống gửi thông báo cảnh báo đến người dùng qua ứng dụng hoặc kích hoạt buzzer.</li><li>- Người dùng có thể nhận thông báo qua điện thoại hoặc trên giao diện Node-RED, theo dõi tình trạng môi trường trong thời gian thực.</li></ul>
Trường hợp ngoại lệ	<ul style="list-style-type: none"><li>- Nếu một trong các cảm biến không hoạt động hoặc có sự cố, hệ thống sẽ thông báo lỗi và yêu cầu người dùng kiểm tra cảm biến.</li><li>- Nếu hệ thống không kết nối được với WiFi hoặc Node-RED, thông báo sẽ không được gửi và hệ thống yêu cầu kết nối lại.</li></ul>
Luồng thay thế	<ul style="list-style-type: none"><li>- Người dùng có thể chọn để nhận thông báo qua các phương thức khác nhau (SMS, email) thay vì chỉ thông qua ứng dụng.</li><li>- Người dùng có thể cài đặt ngưỡng cảnh báo linh hoạt cho từng cảm biến (nhiệt độ, độ ẩm, khí gas, cảm biến lửa).</li></ul>
Điều kiện sau khi hoàn thành	<ul style="list-style-type: none"><li>- Các chỉ số môi trường (nhiệt độ, độ ẩm, khí gas, cảm biến lửa) được ghi lại trong hệ thống và có thể truy xuất trong lịch sử.</li><li>- Hệ thống gửi thông báo cảnh báo khi các chỉ số môi trường vượt ngưỡng an toàn đã được cài đặt.</li></ul>

## 4.2 Hệ thống giám sát môi trường trong xưởng

User case ID	2
User case name	hệ thống giám sát môi trường trong xưởng
Actor	người dùng, ESP32, DHT11, MQ-2, Cảm biến lửa, Buzzer, Node-RED
Mô tả	ESP32 sẽ thu thập các chỉ số nhiệt độ, độ ẩm, khí gas và tình trạng lửa từ các cảm biến. Nếu có bất kỳ cảm biến nào phát hiện giá trị vượt ngưỡng (ví dụ: nhiệt độ cao, khí gas cao, hoặc có lửa), hệ thống sẽ gửi cảnh báo qua buzzer và thông qua Node-RED cho người dùng.
Điều kiện tiên quyết	<ul style="list-style-type: none"><li>- Hệ thống đang hoạt động bình thường.</li><li>- Các cảm biến đã được kết nối và cấu hình với ESP32.</li><li>- Người dùng đã cài đặt ứng dụng giám sát trên điện thoại hoặc hệ thống Node-RED đã được thiết lập để nhận thông báo.</li></ul>
Quy trình hoạt động	<ul style="list-style-type: none"><li>- Bước 1: Hệ thống bắt đầu thu thập dữ liệu từ cảm biến DHT11, MQ-2 và cảm biến lửa.</li><li>- Bước 2: ESP32 kiểm tra các giá trị: nhiệt độ, khí gas, lửa</li><li>- Bước 3: Hệ thống gửi thông tin cảnh báo qua MQTT đến Node-RED.</li><li>- Bước 4: Node-RED nhận thông tin và hiển thị cảnh báo trên Dashboard.</li><li>- Bước 5: Buzzer phát ra âm thanh cảnh báo trong trường hợp có lửa hoặc khí gas vượt ngưỡng.</li></ul>
Trường hợp ngoại lệ	<ul style="list-style-type: none"><li>- Nếu một trong các cảm biến không hoạt động hoặc có sự cố, hệ thống sẽ thông báo lỗi và yêu cầu người dùng kiểm tra cảm biến.</li><li>- Nếu hệ thống không kết nối được với WiFi hoặc Node-RED, thông báo sẽ không được gửi và hệ thống yêu cầu kết nối lại</li></ul>
Luồng thay thế	<ul style="list-style-type: none"><li>- Người dùng có thể chọn để nhận thông báo qua các phương thức khác nhau (SMS, email) thay vì chỉ thông qua ứng dụng.</li></ul>
Điều kiện sau khi hoàn thành	<ul style="list-style-type: none"><li>- Các chỉ số môi trường (nhiệt độ, khí gas, cảm biến lửa) được ghi lại trong hệ thống và có thể truy xuất trong lịch sử.</li><li>- Hệ thống gửi thông báo cảnh báo khi các chỉ số môi trường vượt ngưỡng an toàn đã được cài đặt, đồng thời buzzer phải phát tín hiệu cảnh báo khi các chỉ số vượt ngưỡng.</li></ul>

### 4.3 Hệ thống giáo dục IOT

Use case ID	3
Use case name	Hệ thống giáo dục IoT
Actor	Học sinh, giảng viên, ESP32 và thiết bị (DHT-11, MQ-2, Cảm biến lửa, Buzzer, Node-RED Dashboard)
Mô tả	Hệ thống được thiết kế để giảng dạy các nguyên tắc cơ bản về IoT. Học sinh sẽ học cách kết nối và cấu hình các cảm biến, phân tích dữ liệu thời gian thực trên Node-RED Dashboard và xử lý các sự kiện dựa trên dữ liệu từ cảm biến.
Điều kiện tiên quyết	<ul style="list-style-type: none"><li>- Các cảm biến được kết nối với ESP32.</li><li>- Môi trường Node-RED đã được thiết lập và sẵn sàng nhận dữ liệu.</li><li>- Học sinh đã được hướng dẫn cơ bản về cách sử dụng ESP32 và Node-RED.</li></ul>
Quy trình hoạt động	<ul style="list-style-type: none"><li>- Giảng viên trình bày nguyên lý hoạt động của từng cảm biến (DHT-11, MQ-2, cảm biến lửa).</li><li>- Học sinh cấu hình ESP32 để đọc dữ liệu từ các cảm biến và gửi đến Node-RED.</li><li>- Dữ liệu hiển thị trên Node-RED Dashboard dưới dạng biểu đồ hoặc giá trị thời gian thực.</li><li>- Học sinh thiết lập các sự kiện như: bật buzzer khi phát hiện khí gas, kích hoạt cảnh báo khi nhiệt độ vượt ngưỡng an toàn.</li><li>- Học sinh thực hiện các bài thực hành kiểm tra hệ thống với các tình huống giả định như nhiệt độ cao, rò rỉ khí gas, hoặc phát hiện lửa.</li></ul>
Trường hợp ngoại lệ	<ul style="list-style-type: none"><li>- Nếu cảm biến không gửi dữ liệu, học sinh sẽ kiểm tra kết nối hoặc cấu hình lại hệ thống.</li><li>- Nếu Node-RED không nhận dữ liệu, học sinh phải kiểm tra kết nối WiFi và cài đặt MQTT.</li></ul>
Luồng thay thế	<ul style="list-style-type: none"><li>- Học sinh có thể sử dụng thiết bị di động để kiểm tra Node-RED Dashboard thay vì dùng máy tính.</li><li>- Giảng viên cung cấp các bài tập thực tế để học sinh tự thiết kế kịch bản ứng dụng IoT khác, như giám sát môi trường cho nhà ở.</li></ul>
Điều kiện sau khi hoàn thành	<ul style="list-style-type: none"><li>- Học sinh hiểu cách kết nối và cấu hình cảm biến, sử dụng Node-RED để phân tích dữ liệu thời gian thực.</li><li>- Hệ thống phản hồi chính xác với các tình huống giả định, và học sinh trình bày được cách cải thiện hệ thống.</li></ul>

#### 4.4 Hệ thống an toàn bếp thông minh

Use case ID	4
Use case name	Hệ thống an toàn bếp thông minh
Actor	Người dùng, ESP32 và thiết bị (MQ-2, Cảm biến lửa, Buzzer, Node-RED Dashboard, PC PWM Fan)
Mô tả	Hệ thống giúp giám sát và đảm bảo an toàn trong nhà bếp bằng cách phát hiện khí gas rò rỉ và các nguy cơ cháy nổ. Khi phát hiện sự cố, hệ thống sẽ cảnh báo người dùng qua buzzer và Node-RED Dashboard, đồng thời kích hoạt quạt thông gió để giảm thiểu rủi ro.
Điều kiện tiên quyết	<ul style="list-style-type: none"><li>- MQ-2 và cảm biến lửa được lắp đặt trong bếp và kết nối với ESP32.</li><li>- Buzzer và quạt thông gió đã được cấu hình và sẵn sàng hoạt động.</li><li>- Node-RED Dashboard đã thiết lập để hiển thị dữ liệu và gửi cảnh báo.</li></ul>
Quy trình hoạt động	<ul style="list-style-type: none"><li>- MQ-2 đo nồng độ khí gas trong không khí và gửi dữ liệu đến ESP32.</li><li>- Nếu nồng độ khí gas vượt ngưỡng an toàn, ESP32 kích hoạt buzzer và quạt thông gió, đồng thời gửi cảnh báo đến Node-RED Dashboard.</li><li>- Cảm biến lửa theo dõi ánh sáng từ ngọn lửa và gửi tín hiệu đến ESP32 khi phát hiện cháy.</li><li>- Người dùng nhận cảnh báo từ Node-RED Dashboard và thực hiện biện pháp xử lý cần thiết (tắt gas, kiểm tra bếp).</li></ul>
Trường hợp ngoại lệ	<ul style="list-style-type: none"><li>- Nếu MQ-2 hoặc cảm biến lửa không hoạt động, ESP32 sẽ gửi thông báo lỗi đến Node-RED Dashboard.</li><li>- Nếu hệ thống không kết nối WiFi, các cảnh báo sẽ chỉ phát qua buzzer thay vì gửi đến Node-RED.</li></ul>
Luồng thay thế	<ul style="list-style-type: none"><li>- Người dùng có thể cấu hình ngưỡng cảnh báo tùy thuộc vào môi trường (ví dụ: mức khí gas an toàn cao hơn trong nhà bếp thương mại).</li><li>- Hệ thống có thể gửi cảnh báo qua email hoặc SMS ngoài giao diện Node-RED.</li></ul>
Điều kiện sau khi hoàn thành	<ul style="list-style-type: none"><li>- Hệ thống phát hiện chính xác khí gas rò rỉ hoặc cháy nổ và đưa ra cảnh báo kịp thời.</li><li>- Dữ liệu môi trường bếp được lưu trữ và hiển thị trên Node-RED Dashboard.</li></ul>

## 4.5 Hệ thống giám sát kho lạnh

Use case ID	9
Use case name	Hệ thống giám sát kho lạnh
Actor	Người dùng, ESP32 và thiết bị (DHT-11, MQ-2, Buzzer, Node-RED Dashboard)
Mô tả	Hệ thống giám sát và đảm bảo môi trường kho lạnh, kiểm tra các chỉ số nhiệt độ, độ ẩm và nồng độ khí gas rò rỉ. Khi các chỉ số vượt ngưỡng an toàn, hệ thống sẽ phát cảnh báo qua buzzer và gửi thông tin đến Node-RED Dashboard.
Điều kiện tiên quyết	<ul style="list-style-type: none"><li>- Các cảm biến (DHT-11, MQ-2) được lắp đặt trong kho lạnh và kết nối với ESP32.</li><li>- Buzzer đã cấu hình và sẵn sàng hoạt động.</li><li>- Node-RED Dashboard đã thiết lập để hiển thị dữ liệu và gửi cảnh báo.</li></ul>
Quy trình hoạt động	<ul style="list-style-type: none"><li>- DHT-11 đo nhiệt độ và độ ẩm, MQ-2 giám sát nồng độ khí gas và gửi dữ liệu đến ESP32.</li><li>- Nếu nhiệt độ, độ ẩm hoặc khí gas vượt ngưỡng an toàn:<ul style="list-style-type: none"><li>+ ESP32 kích hoạt buzzer để phát cảnh báo tại chỗ.</li><li>+ Dữ liệu sự cố được gửi lên Node-RED Dashboard.</li></ul></li><li>- Người dùng nhận cảnh báo qua Node-RED và thực hiện các biện pháp xử lý cần thiết.</li></ul>
Trường hợp ngoại lệ	<ul style="list-style-type: none"><li>- Nếu DHT-11 hoặc MQ-2 không hoạt động, ESP32 sẽ gửi thông báo lỗi đến Node-RED Dashboard.</li><li>- Nếu hệ thống mất kết nối WiFi, cảnh báo sẽ chỉ phát qua buzzer thay vì gửi đến Node-RED.</li></ul>
Luồng thay thế	<ul style="list-style-type: none"><li>- Người dùng có thể tùy chỉnh ngưỡng cảnh báo trên Node-RED Dashboard.</li><li>- Hệ thống có thể gửi cảnh báo qua email hoặc SMS thay vì chỉ hiển thị trên Node-RED.</li></ul>
Điều kiện sau khi hoàn thành	<ul style="list-style-type: none"><li>- Hệ thống giám sát chính xác và gửi cảnh báo kịp thời khi môi trường kho lạnh có sự cố.</li><li>- Dữ liệu nhiệt độ, độ ẩm và khí gas được lưu trữ và hiển thị trên Node-RED Dashboard.</li></ul>

## 5 SƠ ĐỒ KỸ THUẬT PHẦN CỨNG:

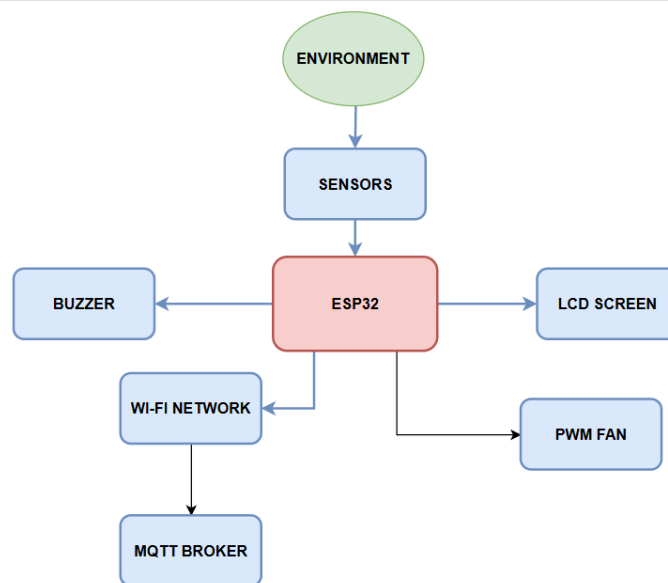
Trong phần này, chúng tôi cung cấp một phân tích chi tiết về kiến trúc hệ thống, mô tả cách các thành phần khác nhau tương tác trong hệ thống giám sát môi trường dựa trên ESP32.

### 5.1 Tổng Quan Hệ Thống

Hệ thống tích hợp nhiều cảm biến và thiết bị điều khiển với vi điều khiển ESP32 để giám sát và phản ứng với các điều kiện môi trường như nhiệt độ, độ ẩm, nồng độ khí, và sự xuất hiện của ngọn lửa. ESP32 đọc dữ liệu từ các cảm biến, điều khiển quạt và còi báo động, hiển thị dữ liệu trên màn hình LCD và giao tiếp với máy chủ từ xa qua giao thức MQTT.

### 5.2 Sơ Đồ Khối Phần Cứng

Dưới đây là sơ đồ khối mô tả sự kết nối giữa các thành phần trong hệ thống:



Hình 1: Sơ đồ khối phần cứng

#### Giải thích:

- Cảm biến **DHT11** được sử dụng để thu thập dữ liệu nhiệt độ và độ ẩm, sau đó dữ liệu này được xử lý bởi **ESP32**.
- Cảm biến **MQ Gas Sensor** phát hiện nồng độ các khí như LPG, CO và khói. Cảm biến này cung cấp tín hiệu analog cho ESP32.
- Cảm biến **Flame Sensor** theo dõi sự hiện diện của ngọn lửa. Nếu phát hiện ngọn lửa, ESP32 sẽ kích hoạt còi báo động.
- **PWM Fan** được điều khiển bởi ESP32 sử dụng điều chế độ rộng xung (PWM) dựa trên nồng độ khí. Nếu phát hiện mức khí nguy hiểm, tốc độ quạt sẽ được tăng lên để thông gió.

- **Màn hình LCD** hiển thị dữ liệu thời gian thực về nhiệt độ, độ ẩm, nồng độ khí và tình trạng ngọn lửa.
- **ESP32** giao tiếp với máy chủ từ xa qua giao thức **MQTT** trên mạng Wi-Fi để gửi dữ liệu cảm biến thời gian thực.

### 5.3 Sơ Đồ Cấu Hình Chân

Dưới đây là sơ đồ cấu hình chân sử dụng để kết nối các cảm biến và thiết bị điều khiển với ESP32:

Component	Pin Connection
DHT11 Sensor	GPIO 32
MQ Gas Sensor	GPIO 34 (Analog)
Flame Sensor	GPIO 27
PWM Fan	GPIO 9
Buzzer	GPIO 25
LCD Screen (I2C)	GPIO21 and GPIO 22

**Bảng 1:** Cấu Hình Kết Nối Chân Các Thành Phần

**Giải thích:**

- Cảm biến **DHT11** được kết nối với **GPIO 32** của ESP32 để đọc dữ liệu nhiệt độ và độ ẩm.
- Cảm biến **MQ Gas Sensor** phát ra tín hiệu analog tới **GPIO 34**, ESP32 sẽ đọc giá trị nồng độ khí.
- Cảm biến **Flame Sensor** gửi tín hiệu số đến **GPIO 27**, giúp ESP32 phát hiện sự xuất hiện của ngọn lửa.
- **PWM Fan** được kết nối với **GPIO 9**, nơi tốc độ quạt được điều khiển thông qua tín hiệu PWM.
- **Còi báo động** được kết nối với **GPIO 25**, và sẽ phát ra âm thanh khi phát hiện ngọn lửa.
- **Màn hình LCD** được kết nối qua giao thức I2C với **GPIO 21** (SDA) và **GPIO 22** (SCL) để hiển thị dữ liệu thời gian thực.

### 5.4 Sơ Đồ Dòng Dữ Liệu

Sơ đồ dòng dữ liệu mô tả cách thức thu thập, xử lý và truyền tải dữ liệu trong hệ thống:

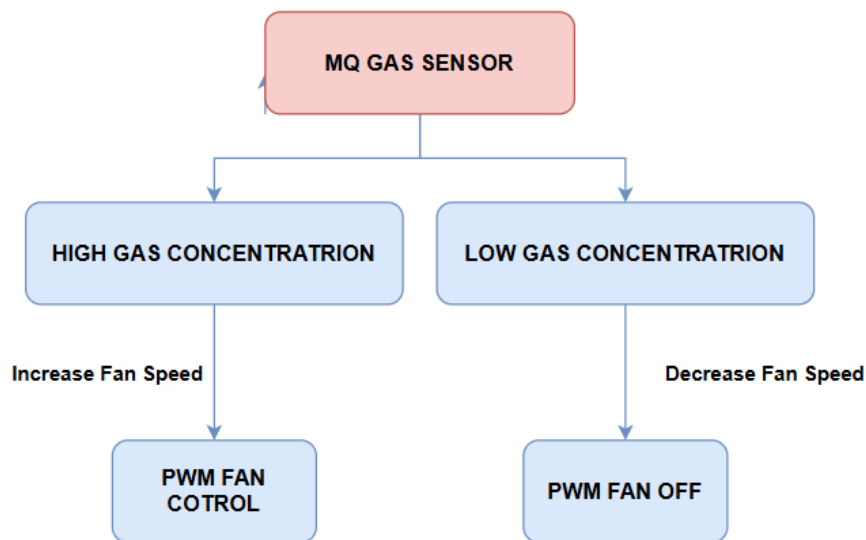
**Giải thích:**

- Các **Cảm biến (DHT11, MQ Gas Sensor, Flame Sensor)** liên tục gửi dữ liệu đến **ESP32 Microcontroller**, nơi dữ liệu được xử lý.
- **ESP32** sau đó gửi dữ liệu đã xử lý đến **Màn hình LCD** để hiển thị.
- Hệ thống điều khiển **PWM Fan** dựa trên nồng độ khí và kích hoạt **Còi báo động** khi phát hiện ngọn lửa.

- **ESP32** gửi dữ liệu cảm biến đến máy chủ từ xa qua giao thức **MQTT**, cung cấp giám sát từ xa các điều kiện môi trường.

## 5.5 Sơ Đồ Logic Điều Khiển Quạt

Dưới đây là sơ đồ logic mô tả cách thức điều khiển **PWM Fan** dựa trên dữ liệu từ **MQ Gas Sensor**:



Hình 2: Sơ đồ logic điều khiển quạt

**Giải thích:**

- Nếu nồng độ khí cao (vượt quá ngưỡng), tốc độ quạt sẽ được tăng lên để giúp thông gió khu vực.
- Nếu nồng độ khí trong giới hạn an toàn, tốc độ quạt sẽ giảm hoặc tắt.

## 5.6 Sơ Đồ Giao Tiếp MQTT

Sơ đồ giao tiếp MQTT mô tả cách thức ESP32 giao tiếp với máy chủ từ xa qua broker MQTT:

**Giải thích:**

- **ESP32** kết nối với broker MQTT qua Wi-Fi và **đăng tải** dữ liệu cảm biến (nhiệt độ, độ ẩm, nồng độ khí và tình trạng ngọn lửa) dưới dạng các thông điệp **JSON**.
- **Máy chủ từ xa** đăng ký (subscribe) vào chủ đề MQTT để nhận các cập nhật và giám sát hệ thống từ xa.

Những sơ đồ này mô tả chi tiết các thành phần phần cứng, dòng dữ liệu và logic của hệ thống, cung cấp một cái nhìn rõ ràng về cách các thành phần tương tác và hoạt động của hệ thống giám sát môi trường dựa trên ESP32.



## 6 KỸ THUẬT PHẦN MỀM

### 6.1 Tổng quan

Hệ thống thu thập và xử lý dữ liệu được thiết kế để giám sát và quản lý thông tin từ các cảm biến môi trường, cung cấp khả năng cập nhật dữ liệu theo thời gian thực. Dữ liệu được thu thập liên tục và xử lý để cung cấp các cảnh báo khi có sự thay đổi đáng chú ý, cho phép người dùng dễ dàng theo dõi và tương tác với các thông số môi trường qua một giao diện trực quan. Hệ thống này giúp người dùng nắm bắt và quản lý tình trạng môi trường một cách hiệu quả và thuận tiện.

Việc tích hợp các luồng công việc phức tạp giúp xử lý dữ liệu hiệu quả, đồng thời cung cấp một bảng điều khiển trực quan cho phép người dùng giám sát và nhận thông báo ngay khi có sự kiện quan trọng. Hệ thống này bảo đảm khả năng truyền tải dữ liệu ổn định, tạo ra một môi trường giám sát thông minh, tự động và đáng tin cậy, hỗ trợ việc ra quyết định kịp thời.

Hệ thống được thiết kế như một nền tảng IoT toàn diện, kết hợp từ việc thu thập dữ liệu cảm biến đến cung cấp giao diện web thân thiện với người dùng. Mục tiêu của hệ thống là giám sát và quản lý dữ liệu môi trường theo thời gian thực, đồng thời cung cấp các tính năng dự đoán và cảnh báo thông minh. Các thành phần chính của hệ thống bao gồm thiết bị IoT thu thập dữ liệu, xử lý backend quản lý và phân tích dữ liệu, và giao diện frontend cung cấp trải nghiệm người dùng trực quan và tương tác.

### 6.2 Thiết kế tổng quan của hệ thống Dashboard

#### 6.2.1 Node-RED

**Nhận dữ liệu:** Node MQTT được cấu hình để kết nối với bộ môi giới HiveMQ, nhận dữ liệu cảm biến bao gồm các thông số như nhiệt độ, độ ẩm và mức LPG. Những node này hoạt động như các điểm đầu vào, cho phép hệ thống thu thập dữ liệu cảm biến ở định dạng JSON có cấu trúc, đảm bảo tính nhất quán và dễ xử lý.

**Xử lý dữ liệu:** Các node chức năng trong Node-RED thực hiện các nhiệm vụ quan trọng như chuyển đổi đơn vị từ Celsius sang Fahrenheit, phân tích ngưỡng an toàn, và phát hiện các rủi ro tiềm tàng như rò rỉ khí gas hoặc cháy. Các node này sử dụng các hàm JavaScript tùy chỉnh, mang lại khả năng xử lý dữ liệu linh hoạt và mạnh mẽ, phù hợp với các yêu cầu cụ thể.

**Hiển thị và cảnh báo:** Hệ thống sử dụng bảng điều khiển Node-RED để trực quan hóa dữ liệu thông qua các biểu đồ và đồng hồ đo tương tác. Giao diện trực quan này cung cấp cái nhìn tổng quan theo thời gian thực về điều kiện môi trường, giúp người dùng giám sát và phản ứng kịp thời với các thay đổi. Trong trường hợp xảy ra bất thường, hệ thống sẽ kích hoạt thông báo qua email và cảnh báo để thông tin ngay lập tức cho người dùng.

#### 6.2.2 HiveMQ

HiveMQ đóng vai trò như một bộ môi giới MQTT, tạo điều kiện giao tiếp giữa vi điều khiển ESP32 và ứng dụng Node-RED. Bộ môi giới này hỗ trợ luồng dữ liệu theo thời gian thực, đảm bảo dữ liệu cảm biến được truyền đi hiệu quả và đáng tin cậy. Với kiến trúc

manh mẽ, HiveMQ xử lý lưu lượng dữ liệu lớn và duy trì tính toàn vẹn của dữ liệu, là lựa chọn lý tưởng cho các ứng dụng IoT yêu cầu kết nối liên tục và phản hồi nhanh.

### 6.2.3 Quy trình hoạt động của hệ thống

#### Nhận dữ liệu

Các node MQTT trong Node-RED được thiết lập để kết nối với bộ môi giới HiveMQ. Dữ liệu cảm biến về nhiệt độ, độ ẩm và mức LPG được nhận ở định dạng JSON, giúp thông tin có cấu trúc và dễ xử lý. Cách tiếp cận này đảm bảo việc thu thập dữ liệu liên mạch và khởi đầu cho quy trình xử lý dữ liệu.

#### Xử lý dữ liệu

**Chuyển đổi dữ liệu:** Các node chức năng trong Node-RED thực hiện các tính toán cần thiết, chẳng hạn như chuyển đổi nhiệt độ từ Celsius sang Fahrenheit và phân tích mức LPG. Những chuyển đổi này chuẩn hóa dữ liệu, giúp dễ dàng giải thích và sử dụng cho các bước xử lý và hiển thị tiếp theo.

#### Kiểm tra ngưỡng an toàn:

- **Node Kiểm tra mức khí gas:** Node này liên tục giám sát mức LPG và kích hoạt cảnh báo nếu giá trị vượt qua các ngưỡng an toàn được định trước, cho thấy nguy cơ rò rỉ khí gas tiềm tàng. Bằng cách so sánh dữ liệu cảm biến với các tiêu chuẩn an toàn, node này đảm bảo quản lý rủi ro chủ động.
- **Node Kiểm tra cháy:** Node này đánh giá dữ liệu để phát hiện nguy cơ cháy và gửi thông báo qua email cũng như cảnh báo khẩn cấp nếu phát hiện mối nguy hại. Logic của node tích hợp nhiều đầu vào cảm biến, cung cấp đánh giá toàn diện về an toàn cháy nổ.

#### Hiển thị dữ liệu

Bảng điều khiển Node-RED được trang bị:

- **Biểu đồ:** Hiển thị xu hướng theo thời gian thực của các thông số môi trường, cho phép người dùng quan sát các thay đổi theo thời gian. Biểu đồ cung cấp thông tin lịch sử, hỗ trợ phát hiện các mẫu hoặc bất thường trong dữ liệu.
- **Đồng hồ đo:** Hiển thị giá trị hiện tại của nhiệt độ, độ ẩm và mức LPG theo cách dễ hiểu. Đồng hồ đo cung cấp cái nhìn nhanh về các thông số chính, hỗ trợ đánh giá tức thì.
- **Thông báo tức thì:** Giao diện cung cấp các thông báo theo thời gian thực trong trường hợp khẩn cấp, đảm bảo thời gian phản hồi nhanh chóng. Bằng cách làm nổi bật các cảnh báo quan trọng, bảng điều khiển đảm bảo người dùng luôn được thông báo và có thể hành động ngay lập tức.

#### Cảnh báo và hành động

Trong trường hợp dữ liệu cảm biến vượt ngưỡng an toàn, hệ thống Node-RED sẽ kích hoạt cảnh báo qua email thông qua node mail và hiển thị thông báo cảnh báo trên bảng điều khiển. Những cảnh báo này cung cấp thông tin kịp thời, giúp người dùng xử lý các mối nguy hiểm ẩn nhanh chóng.

Một tính năng giới hạn tần suất (**limit**) được triển khai để ngăn hệ thống gửi nhiều cảnh báo trong thời gian ngắn, đảm bảo rằng các thông báo có ý nghĩa và không gây quá tải. Tính năng này cân bằng giữa việc đưa ra cảnh báo kịp thời và tránh gây mệt mỏi bởi cảnh báo, duy trì hiệu quả và khả năng sử dụng của hệ thống.

## 6.3 Thiết kế tổng thể Website

### 6.3.1 Frontend - ReactJS

**Hiển thị dữ liệu:** Frontend được xây dựng bằng ReactJS, cung cấp giao diện trực quan để hiển thị dữ liệu cảm biến. Các biểu đồ, đồng hồ đo và bảng dữ liệu được cập nhật thời gian thực thông qua kết nối WebSocket. Người dùng có thể dễ dàng theo dõi các thông số như nhiệt độ, độ ẩm và mức khí gas.

**Tương tác người dùng:** Hệ thống hỗ trợ các tính năng như cài đặt ngưỡng nhiệt độ, bật/tắt thiết bị từ xa, và nhận cảnh báo khi xảy ra sự cố. Giao diện được thiết kế tối ưu cho trải nghiệm người dùng, giúp họ điều khiển thiết bị IoT một cách trực quan.

**Kết nối Backend:** Frontend giao tiếp với Backend thông qua REST API và WebSocket, đảm bảo dữ liệu được truyền đi hiệu quả và cập nhật nhanh chóng.

### 6.3.2 Backend - Flask

**Nhận và lưu trữ dữ liệu:** Backend sử dụng Flask để xử lý dữ liệu từ ESP32. Dữ liệu môi trường, bao gồm nhiệt độ, độ ẩm và khí gas, được lưu trữ vào cơ sở dữ liệu MongoDB, đảm bảo tính toàn vẹn và khả năng truy xuất.

**Xử lý dữ liệu:** Các mô-đun trong Backend thực hiện phân tích dữ liệu, chẳng hạn như tính toán ngưỡng an toàn, dự đoán nhiệt độ bằng mô hình AI (Linear Regression và Random Forest), và phát hiện nguy cơ cháy thông qua dữ liệu camera.

**Giao tiếp với Frontend:** Backend cung cấp API để truyền dữ liệu tới giao diện ReactJS và gửi thông báo thời gian thực qua WebSocket khi phát hiện sự cố.

### 6.3.3 ESP32 và MongoDB

**ESP32 - Thu thập dữ liệu:** ESP32 đóng vai trò thiết bị IoT chính, tích hợp các cảm biến như:

- **DHT11/DHT22:** Đo nhiệt độ và độ ẩm.
- **MQ-2:** Phát hiện nồng độ khí gas.

Dữ liệu cảm biến được gửi đến Backend thông qua giao thức HTTP hoặc MQTT.

**MongoDB - Lưu trữ dữ liệu:** MongoDB được sử dụng để lưu trữ dữ liệu cảm biến với cấu trúc JSON. Điều này giúp hệ thống dễ dàng truy xuất và xử lý dữ liệu, hỗ trợ các tính năng như trực quan hóa và phân tích.

### 6.3.4 Kiến trúc tổng thể của hệ thống

**Nhận dữ liệu:** ESP32 thu thập dữ liệu từ các cảm biến và gửi đến Backend. Backend lưu trữ dữ liệu vào MongoDB và cung cấp các API để Frontend truy cập.

**Xử lý dữ liệu:** Backend thực hiện các tính toán như kiểm tra ngưỡng an toàn, dự đoán nhiệt độ, và phát hiện cháy.

**Hiển thị và điều khiển:** Frontend hiển thị dữ liệu theo thời gian thực qua các biểu đồ và đồng hồ đo. Người dùng có thể điều khiển thiết bị từ xa thông qua giao diện.

**Cảnh báo:** Hệ thống phát cảnh báo qua email và giao diện web khi phát hiện sự cố, đảm bảo người dùng nhận được thông tin kịp thời để xử lý.

## 7 TRỰC QUAN SẢN PHẨM

### 7.1 Node-red Dashboard

Giao diện dashboard sử dụng Node-RED được thiết kế để cung cấp một cái nhìn trực quan và dễ dàng giám sát các tình huống khẩn cấp trong môi trường. Các thông báo và cảnh báo sẽ được hiển thị rõ ràng để người sử dụng có thể phản ứng kịp thời.

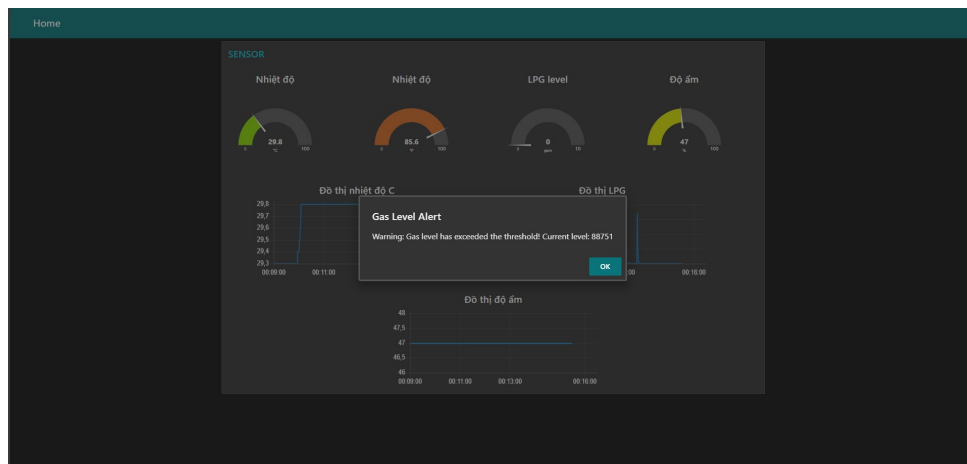
Giao diện chính của dashboard



Cảnh báo khi phát hiện lửa



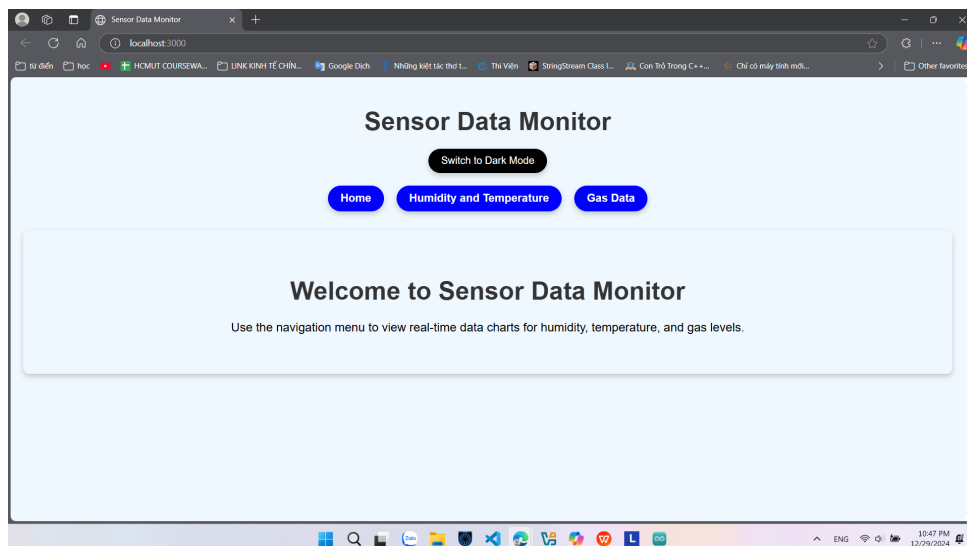
Cảnh báo khi phát hiện khí Gas bất thường



## 7.2 Hệ thống Website

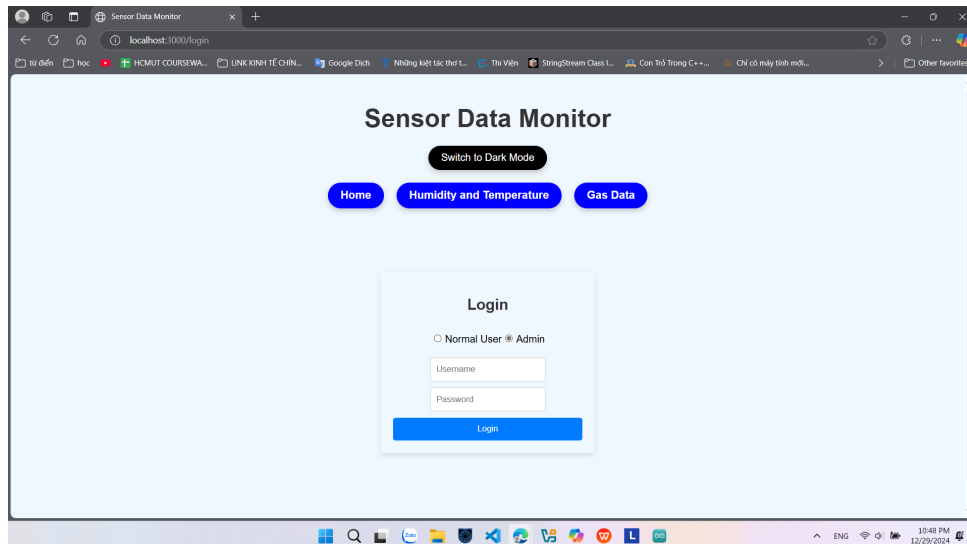
### 7.2.1 Tab Home

- **Mô tả:** Trang chủ cung cấp thông tin khái quát về hệ thống.
- **Nội dung chi tiết:**
  - **Mục tiêu:** Giám sát và cảnh báo an toàn trong các môi trường được giám sát.
  - **Phạm vi ứng dụng:** Đáp ứng nhu cầu quản lý dữ liệu môi trường trong nhà máy, gia đình hoặc văn phòng.
  - **Các thành phần chính:** Mô đun giao diện, mô đun xử lý dữ liệu, và mô đun cảnh báo.



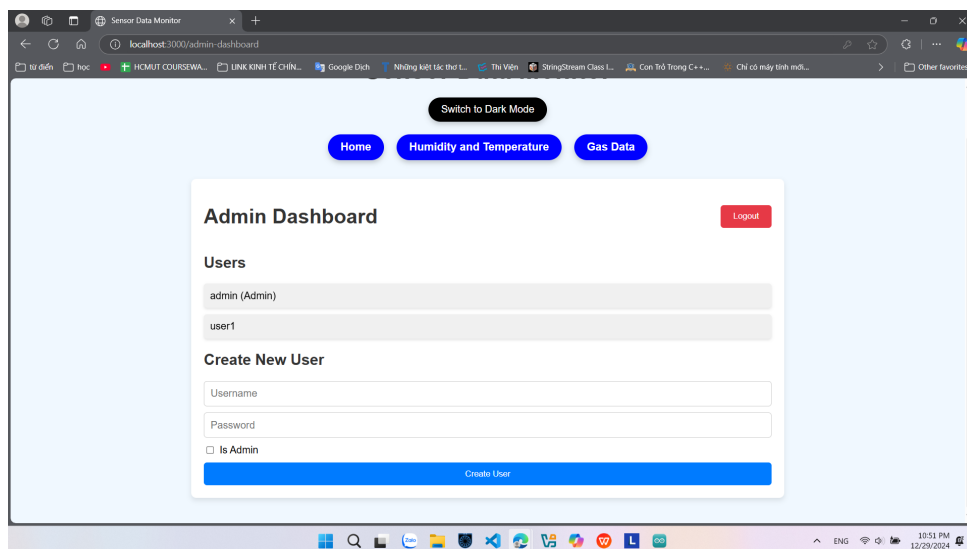
### 7.2.2 Chế độ đăng nhập (Admin và User)

- **Mô tả:** Hệ thống cung cấp giao diện đăng nhập cho hai loại người dùng: Admin và User.



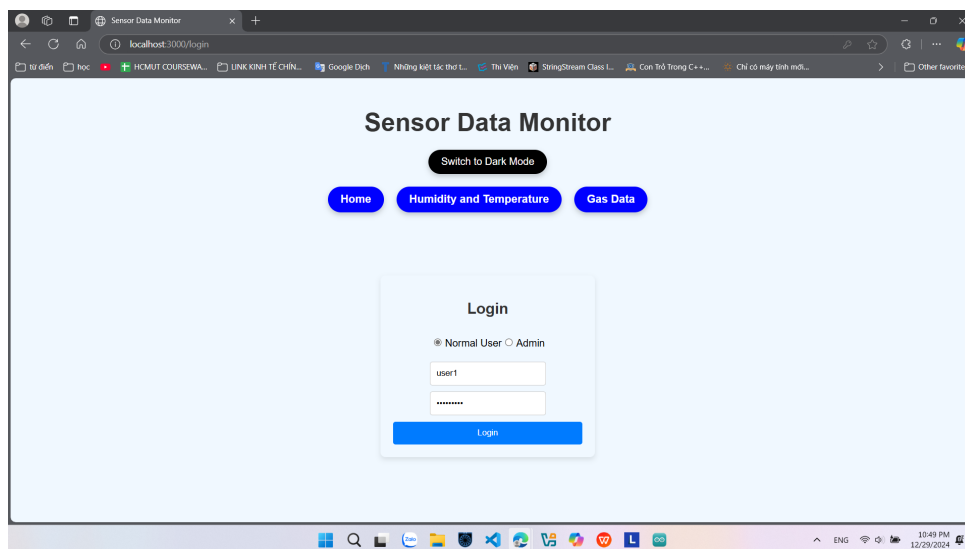
- **Chức năng:**

- **Admin:** Tạo, xóa, và quản lý User



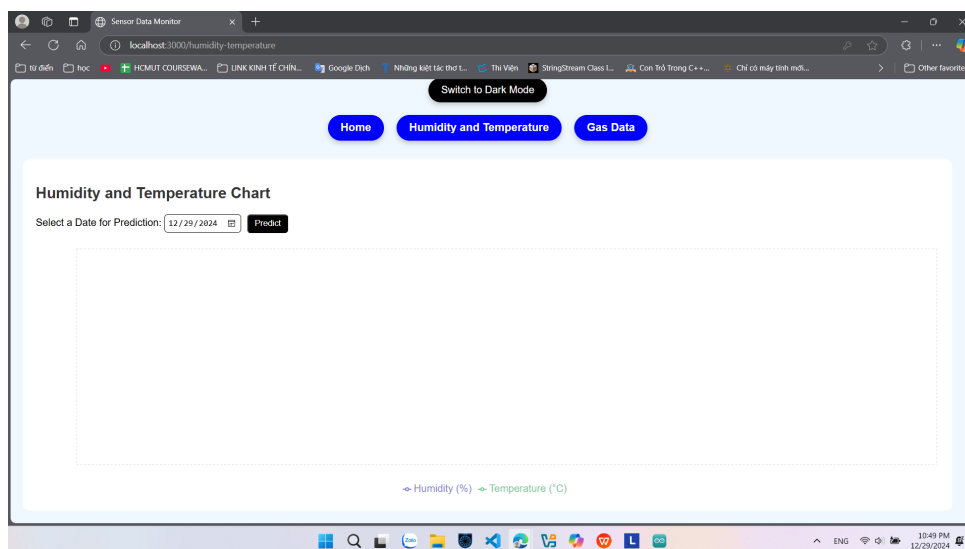
- **User:**

- \* Xem dữ liệu đã thu thập.
- \* Sử dụng các chức năng được phép truy cập.



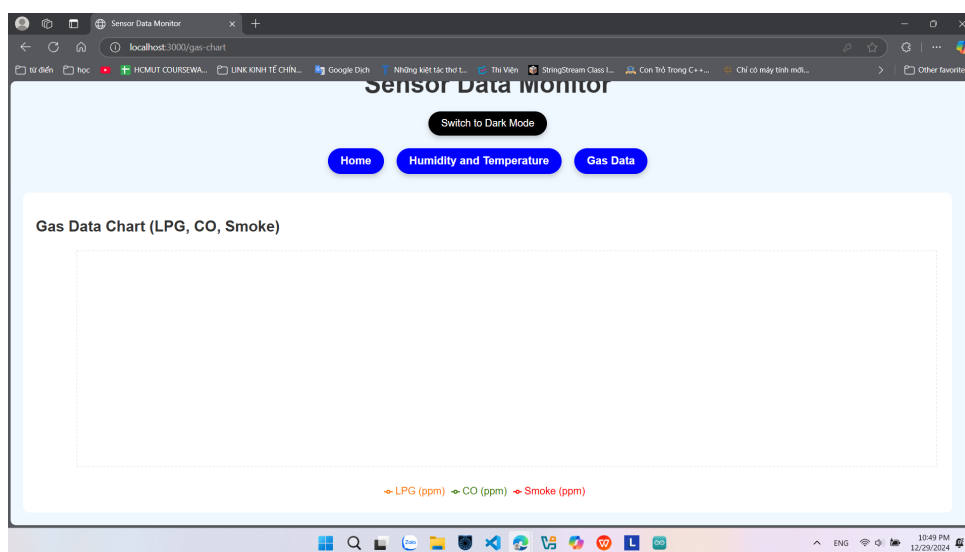
### 7.2.3 Tab Temperature and Humidity (Nhiệt độ và độ ẩm)

- **Mô tả:** Hiển thị dữ liệu nhiệt độ và độ ẩm theo thời gian thực.
- **Chức năng chi tiết:**
  - Thu thập dữ liệu từ các cảm biến nhiệt độ và độ ẩm.
  - Hiển thị dữ liệu dưới dạng biểu đồ.
  - Sử dụng AI phân tích dữ liệu lịch sử và dự báo xu hướng.



#### 7.2.4 Tab Gas Data

- **Mô tả:** Giám sát nồng độ khí gas trong không khí.
- **Chức năng chi tiết:**
  - Thu thập dữ liệu về khí gas từ các cảm biến.
  - Cảnh báo khi nồng độ khí gas vượt mức an toàn, bằng tin nhắn hoặc email.
  - Hiển thị dữ liệu trong thời gian thực.



## 8 KHẢ NĂNG MỞ RỘNG

Với nền tảng công nghệ vững chắc, hệ thống có khả năng mở rộng lớn. Các tính năng dựa trên trí tuệ nhân tạo, như dự đoán xu hướng nhiệt độ và độ ẩm, hay phát hiện cháy qua camera, có thể được tích hợp trong tương lai để gia tăng độ chính xác và tính tự động của hệ thống. Bên cạnh đó, khả năng mở rộng cơ sở dữ liệu với MongoDB và SQLite3 cũng tạo ra điều kiện thuận lợi cho việc lưu trữ và truy xuất dữ liệu trong môi trường lớn hơn, hỗ trợ thêm cho các ứng dụng công nghiệp, văn phòng hay các không gian công cộng.

Hệ thống còn có thể dễ dàng mở rộng về mặt giao diện người dùng với việc triển khai thêm các tính năng như phân quyền người dùng và điều khiển thiết bị thông qua các ứng dụng di động. Việc ứng dụng Node-RED và các giao thức MQTT/WebSocket cho phép mở rộng khả năng giao tiếp và kết nối với các hệ thống khác, mang lại tính linh hoạt cao cho các nhu cầu trong tương lai.

Với khả năng nâng cấp và tùy chỉnh này, hệ thống giám sát môi trường thông minh sẽ là giải pháp hiệu quả và bền vững, đáp ứng nhu cầu ngày càng cao về bảo vệ sức khỏe và an toàn trong mọi không gian.



## 9 KẾT LUẬN

Hệ thống giám sát môi trường thông minh mà chúng tôi đề xuất đã hoàn thành xuất sắc các mục tiêu đặt ra, cung cấp một giải pháp toàn diện và hiệu quả để theo dõi và điều khiển các yếu tố môi trường quan trọng như nhiệt độ, độ ẩm và nồng độ khí gas. Việc sử dụng vi điều khiển ESP32 kết hợp với các cảm biến chuyên dụng giúp hệ thống thu thập và phân tích các dữ liệu môi trường trong thời gian thực, đồng thời thực hiện các hành động tự động như kích hoạt quạt hút khi nồng độ khí độc vượt quá ngưỡng an toàn hoặc khi nhiệt độ vượt mức cài đặt. Điều này không chỉ giúp nâng cao chất lượng môi trường sống mà còn bảo vệ sức khỏe con người, đặc biệt trong các không gian có tiềm ẩn nguy cơ về khí độc hay điều kiện nhiệt độ không ổn định.

Bên cạnh đó, hệ thống giám sát còn cho phép người dùng có thể điều khiển từ xa các thiết bị như quạt, đèn chiếu sáng hay các thiết bị điện tử khác, giúp tiết kiệm năng lượng và tạo ra một không gian sống thông minh, tiện nghi hơn. Việc hiển thị các thông số trên màn hình LCD 16x2 và cung cấp các cảnh báo tức thì qua ứng dụng giúp người dùng luôn nắm bắt được tình trạng môi trường trong thời gian thực và có những biện pháp xử lý kịp thời.

Hệ thống này không chỉ phù hợp cho các hộ gia đình mà còn có tiềm năng ứng dụng rộng rãi trong các nhà xưởng, văn phòng hay các không gian công nghiệp, nơi mà yêu cầu về an toàn và bảo vệ sức khỏe là rất quan trọng. Hệ thống giúp nâng cao hiệu quả vận hành và cải thiện môi trường làm việc, giảm thiểu nguy cơ tai nạn liên quan đến các yếu tố môi trường như khí gas hay nhiệt độ cao.