

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**  
**KHOA CÔNG NGHỆ THÔNG TIN**

-----□□□□□-----



**BÁO CÁO BÀI TẬP LỚN**  
**IOT VÀ ỨNG DỤNG**

**Giảng viên hướng dẫn: Nguyễn Quốc Uy**

**Sinh viên thực hiện: Lê Duy Khánh**

**Mã sinh viên: B21DCCN451**

**Lớp: 05**

## **LỜI CẢM ƠN**

Để hoàn thành được báo cáo thực bài tập lớn môn học IOT và ứng dụng thì em xin cảm ơn tới thầy Nguyễn Quốc Uy đã luôn nhiệt tình chỉ dẫn, giảng dạy cho em những kiến thức về môn học để em có thể hoàn thiện tốt nhất .

Em xin chân thành cảm ơn!

Hà Nội, ngày 10 tháng 11 năm 2024

# CHƯƠNG 1: GIỚI THIỆU BÀI TOÁN

## 1.1. Đặt vấn đề

Trong những năm gần đây, sự phát triển vượt bậc của công nghệ đã tạo ra những bước tiến đáng kể trong việc cải thiện chất lượng cuộc sống con người, với sự hỗ trợ của các sản phẩm và ứng dụng thông minh. IoT (Internet of Things), hay Internet vạn vật, nổi lên như một trong những công nghệ chủ chốt, cho phép kết nối và giao tiếp giữa các thiết bị vật lý thông qua Internet. Các thiết bị IoT có khả năng thu thập, chia sẻ dữ liệu và tương tác với nhau mà không cần sự can thiệp của con người, từ đó tối ưu hóa quy trình và tự động hóa nhiều hoạt động trong cuộc sống hàng ngày. Một trong những ứng dụng nổi bật của IoT là trong lĩnh vực nhà thông minh, nơi người dùng có thể theo dõi và điều khiển các thiết bị điện tử từ xa thông qua các cảm biến. Điều này không chỉ mang lại sự tiện lợi mà còn giúp tiết kiệm năng lượng và cải thiện an ninh. Trong bối cảnh hiện tại, vai trò của IoT ngày càng trở nên quan trọng, đặc biệt với xu hướng tích hợp trí tuệ nhân tạo (AI) để phân tích dữ liệu và dự đoán hành vi người dùng, qua đó nâng cao trải nghiệm sử dụng.

Nắm bắt được vấn đề đó, bản thân em đã trang bị những kiến thức nền tảng cơ bản về các thiết bị IOT cũng như phần mềm để xây dựng một sản phẩm đơn giản là Smart Home.

## **CHƯƠNG 2 : CÁC CHỨC NĂNG**

### **2.1: Các chức năng cơ bản**

Phần mềm được xây dựng dưới dạng một ứng dụng di động trên nền tảng Android, sử dụng ngôn ngữ Kotlin và Jetpack Compose có các chức năng chính:

- + Xem các thông tin nhiệt độ, độ ẩm , ánh sáng realtime qua cảm biến
- + Điều khiển bật tắt 3 thiết bị: Đèn, Quạt, Điều hòa
- + Xem lịch sử data sensor: Tìm kiếm theo thời gian, tìm kiếm theo nhiệt độ, độ ẩm, ánh sáng
- + Xem lịch sử bật tắt thiết bị: Tìm kiếm theo tên thiết bị, theo trạng thái và có thể tìm kiếm theo thời gian, sắp xếp theo thời gian.
- + Hiện thị thông tin người phát triển

### **2.2: Các thiết bị sử dụng**

- + Phần cứng:
  - ESP32 devkit v1
  - Cảm biến đo nhiệt độ , độ ẩm DHT11
  - Cảm biến quang trở LDR
  - Led
- + Phần mềm:
  - Font-end : html, css, js
  - Back end : Node.js, Express.js
  - Database: Mysql
  - MQTT broker: Mosquitto

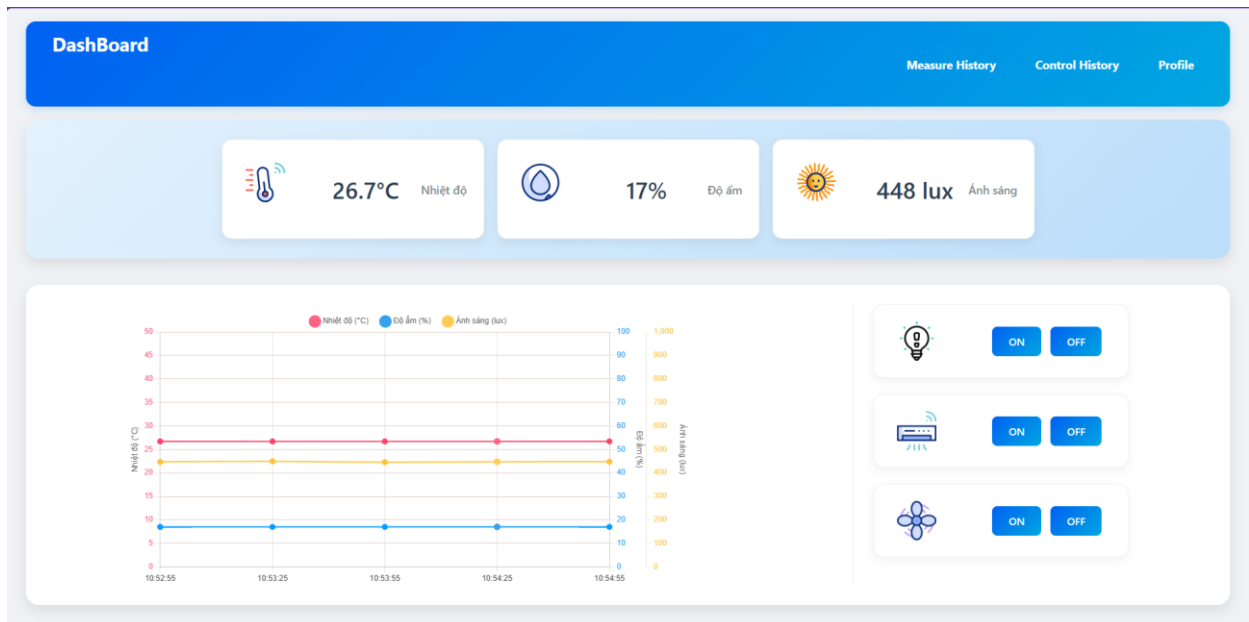
## CHƯƠNG 3 : GIAO DIỆN

Phần giao diện của trang được thiết kế với các tiêu chí dễ sử dụng, trực quan. Gồm 4 trang chính: Dashboard, Measure History, Control History và Profile.

### 3.1. Dashboard:

Giao diện trang hiển thị các thông tin liên quan đến thông tin độ đo trực tiếp từ các cảm biến và trạng thái, điều khiển các thiết bị cứng:

- + Phần hiển thị thông số 3 cảm biến lần lượt là ánh sáng, nhiệt độ , độ ẩm
- + Biểu đồ hiển thị theo 3 thông số là ánh sáng, nhiệt độ , độ ẩm
- + Phần điều khiển: cho phép điều khiển và hiển thị các trạng thái của 3 thiết bị ứng với 3 đèn led kết nối với esp32



Hình 1 : Home Page

### 3.2. Measure History Pages:

Hiển thị danh sách lịch sử dữ liệu của các cảm biến dưới dạng bảng với các độ đo liên quan và thời gian mà cảm biến thu thập dữ liệu.

Các dropdown options cho phép người dùng lựa chọn các chức năng để thực hiện sắp xếp dữ liệu theo ý muốn.( sort theo giá trị tăng dần hoặc giảm dần)

Dữ liệu cảm biến

Sắp xếp theo

Sắp xếp theo

10

Nhập thời gian cụ thể (M)

Tim Kiem

ID	Nhiệt độ	Độ ẩm	Ánh sáng	Thời gian
1	27.1	17	400.11	08/11/2024 21:59:57
2	27.1	17	401.32	08/11/2024 22:00:27
3	27.1	17	398.5	08/11/2024 22:00:57
4	27.1	17	395.27	08/11/2024 22:01:27
5	27.1	17	396.08	08/11/2024 22:01:57
6	27.1	17	397.29	08/11/2024 22:02:27
7	27.1	17	565.31	08/11/2024 22:02:57
8	27.1	17	402.53	08/11/2024 22:03:27
9	27.1	17	402.53	08/11/2024 22:03:57
10	27.1	17	397.29	08/11/2024 22:04:27

Trước

Trang 1 / 22

Tiếp theo

Hình 2 : Sensor Table

Phần Pagesize cho phép hiển thị 10 hoặc 20 hoặc 30 dữ liệu trên cùng 1 bảng:

Dữ liệu cảm biến

Sắp xếp theo

Sắp xếp theo

20

Nhập thời gian cụ thể (M)

Tìm kiếm

ID	Nhiệt độ	Độ ẩm	Ánh sáng	Thời gian
1	27.1	17	400.11	08/11/2024 21:59:57
2	27.1	17	401.32	08/11/2024 22:00:27
3	27.1	17	398.5	08/11/2024 22:00:57
4	27.1	17	395.27	08/11/2024 22:01:27
5	27.1	17	396.08	08/11/2024 22:01:57
6	27.1	17	397.29	08/11/2024 22:02:27
7	27.1	17	565.31	08/11/2024 22:02:57
8	27.1	17	402.53	08/11/2024 22:03:27
9	27.1	17	402.53	08/11/2024 22:03:57
10	27.1	17	397.29	08/11/2024 22:04:27
11	27.1	17	402.12	08/11/2024 22:04:57
11	27.1	17	402.12	08/11/2024 22:04:57
12	27.1	17	402.53	08/11/2024 22:05:27
13	27.1	17	403.33	08/11/2024 22:05:57
14	27.1	17	406.15	08/11/2024 22:06:27
15	27.1	17	400.11	08/11/2024 22:06:57
16	25.5	65	500	08/11/2024 22:07:26
17	27.1	17	402.53	08/11/2024 22:07:27
18	27.1	17	403.74	08/11/2024 22:07:57
19	27.1	17	401.32	08/11/2024 22:08:27
20	27.1	17	882.82	08/11/2024 22:08:57

Trước

Trang 1 / 11

Tiếp theo

Hình 3 : Hình ảnh Pagesize

Sắp xếp theo

Sắp xếp theo

10

Sắp xếp theo

Nhiệt độ

Độ ẩm

Ánh sáng

Nhập thời gian cụ thể (M)

Tìm kiếm

	Nhiệt độ	Độ ẩm	Ánh sáng	Thời gian
1	27.1	17	400.11	08/11/2024 21:59:57
2	27.1	17	401.32	08/11/2024 22:00:27
3	27.1	17	398.5	08/11/2024 22:00:57
4	27.1	17	395.27	08/11/2024 22:01:27
5	27.1	17	396.08	08/11/2024 22:01:57
6	27.1	17	397.29	08/11/2024 22:02:27
7	27.1	17	565.31	08/11/2024 22:02:57
8	27.1	17	402.53	08/11/2024 22:03:27
9	27.1	17	402.53	08/11/2024 22:03:57
10	27.1	17	397.29	08/11/2024 22:04:27

Trước

Trang 1 / 22

Tiếp theo

Sắp xếp theo

Sắp xếp theo

10

Sắp xếp theo

Tăng dần

Giảm dần

Nhập thời gian cụ thể (M)

Tìm kiếm

ID	Nhiệt độ	Độ ẩm	Ánh sáng	Thời gian
1	27.1	17	400.11	08/11/2024 21:59:57
2	27.1	17	401.32	08/11/2024 22:00:27
3	27.1	17	398.5	08/11/2024 22:00:57
4	27.1	17	395.27	08/11/2024 22:01:27
5	27.1	17	396.08	08/11/2024 22:01:57
6	27.1	17	397.29	08/11/2024 22:02:27
7	27.1	17	565.31	08/11/2024 22:02:57
8	27.1	17	402.53	08/11/2024 22:03:27
9	27.1	17	402.53	08/11/2024 22:03:57
10	27.1	17	397.29	08/11/2024 22:04:27

Trước

Trang 1 / 22

Tiếp theo

Hình 4 : Hiện thị dropdown sắp xếp theo các options

Thanh tìm kiếm cho phép người dùng tìm kiếm dữ liệu theo các trường có trong bảng.  
Có 3 kiểu tìm kiếm:

1. Tìm kiếm theo định dạng YYYY-MM-DD HH:mm:ss:

Dữ liệu cảm biến

Sắp xếp theo

Sắp xếp theo

10

08/11/2024 21:59:57

Tìm kiếm

ID	Nhiệt độ	Độ ẩm	Ánh sáng	Thời gian
1	27.1	17	400.11	08/11/2024 21:59:57

Trước

Trang 1 / 1

Tiếp theo

Hình 5: Tìm kiếm theo YYYY-MM-DD HH:mm:ss

2. Tìm kiếm theo định dạng YYYY-MM-DD:

Dữ liệu cảm biến				
Sắp xếp theo	Sắp xếp theo	10	08/11/2024	Tìm kiếm
ID	Nhiệt độ	Độ ẩm	Ánh sáng	Thời gian
1	27.1	17	400.11	08/11/2024 21:59:57
2	27.1	17	401.32	08/11/2024 22:00:27
3	27.1	17	398.5	08/11/2024 22:00:57
4	27.1	17	395.27	08/11/2024 22:01:27
5	27.1	17	396.08	08/11/2024 22:01:57
6	27.1	17	397.29	08/11/2024 22:02:27
7	27.1	17	565.31	08/11/2024 22:02:57
8	27.1	17	402.53	08/11/2024 22:03:27
9	27.1	17	402.53	08/11/2024 22:03:57
10	27.1	17	397.29	08/11/2024 22:04:27
<div> Trước Trang 1 / 14 Tiếp theo </div>				

Hình 6: Tìm kiếm theo định dạng YYYY-MM-DD

### 3. Tìm kiếm theo định dạng HH:mm:ss:

Dữ liệu cảm biến				
Sắp xếp theo	Sắp xếp theo	10	22:02:27	Tìm kiếm
ID	Nhiệt độ	Độ ẩm	Ánh sáng	Thời gian
6	27.1	17	397.29	08/11/2024 22:02:27
<div> Trước Trang 1 / 1 Tiếp theo </div>				

Hình 7 : Tìm kiếm theo định dạng HH:mm:ss:

### 3.3. Control History Page:

Hiển thị danh sách lịch sử bật tắt thiết bị dưới dạng bảng, bao gồm: ID thiết bị, tên, hành động và thời gian diễn ra hành động.

Lịch Sử Điều Khiển Thiết Bị			
Sắp xếp theo	10	Nhập thời gian cụ thể (M)	Tìm kiếm
ID	Thiết Bị	Trạng Thái	Thời gian
1	Đèn	ON	05/11/2024 22:36:28
2	Quạt	ON	05/11/2024 22:36:30
3	Quạt	OFF	05/11/2024 22:36:31
4	Điều Hòa	OFF	05/11/2024 22:36:32
5	Đèn	OFF	05/11/2024 22:36:33
6	Đèn	ON	05/11/2024 22:36:50
7	Điều Hòa	ON	05/11/2024 22:36:51
8	Quạt	ON	05/11/2024 22:36:53
9	Quạt	OFF	05/11/2024 22:36:54
10	Điều Hòa	OFF	05/11/2024 22:36:55
<div> Trước Trang 1 / 21 Tiếp theo </div>			

Hình 8: History Table Control



Các dropdown options cho phép người dùng lựa chọn các chức năng để thực hiện sắp xếp dữ liệu theo ý muốn.

Lịch Sử Điều Khiển Thiết Bị				
Sắp xếp theo	10	Nhập thời gian cụ thể (M)		Tìm Kiếm
Sắp xếp theo				
ON				
OFF				
ID	Thiết Bị	Trạng Thái	Thời gian	
1	Đèn	ON	05/11/2024 22:36:28	
2	Quạt	ON	05/11/2024 22:36:30	
3	Quạt	OFF	05/11/2024 22:36:31	
4	Điều Hòa	OFF	05/11/2024 22:36:32	
5	Đèn	OFF	05/11/2024 22:36:33	
6	Đèn	ON	05/11/2024 22:36:50	
7	Điều Hòa	ON	05/11/2024 22:36:51	
8	Quạt	ON	05/11/2024 22:36:53	
9	Quạt	OFF	05/11/2024 22:36:54	
10	Điều Hòa	OFF	05/11/2024 22:36:55	
Trước Trang 1 / 21 Tiếp theo				

Hình 9 : Dropdown Options thể hiện sắp xếp theo ON/OFF

Phần Pagesize thể hiện 10 hoặc 20 hoặc 30 giá trị lên cùng 1 bảng:

Lịch Sử Điều Khiển Thiết Bị				
Sắp xếp theo	20	Nhập thời gian cụ thể (M)		Tìm Kiếm
ID	Thiết Bị	Trạng Thái	Thời gian	
1	Đèn	ON	05/11/2024 22:36:28	
2	Quạt	ON	05/11/2024 22:36:30	
3	Quạt	OFF	05/11/2024 22:36:31	
4	Điều Hòa	OFF	05/11/2024 22:36:32	
5	Đèn	OFF	05/11/2024 22:36:33	
6	Đèn	ON	05/11/2024 22:36:50	
7	Điều Hòa	ON	05/11/2024 22:36:51	
8	Quạt	ON	05/11/2024 22:36:53	
9	Quạt	OFF	05/11/2024 22:36:54	
10	Điều Hòa	OFF	05/11/2024 22:36:55	
11	Đèn	OFF	05/11/2024 22:36:56	
12	Điều Hòa	ON	05/11/2024 22:36:57	
13	Quạt	ON	05/11/2024 22:36:58	
13	Quạt	ON	05/11/2024 22:36:58	
14	Đèn	ON	05/11/2024 22:37:00	
15	Đèn	OFF	05/11/2024 22:37:01	
16	Điều Hòa	OFF	05/11/2024 22:37:02	
17	Quạt	OFF	05/11/2024 22:37:03	
18	Đèn	ON	05/11/2024 22:37:05	
19	Đèn	OFF	05/11/2024 22:37:06	
20	Điều Hòa	ON	05/11/2024 22:37:06	
Trước Trang 1 / 11 Tiếp theo				

Hình 10 : Hiện thi PageSize

Thanh tìm kiếm cho phép người dùng tìm kiếm dữ liệu theo các trường có trong bảng,

nút lọc . Có 3 kiểu tìm kiếm :

1. *Tìm kiếm theo định dạng YYYY-MM-DD HH:mm:ss:*

Lịch Sử Điều Khiển Thiết Bị			
Sắp xếp theo	10	05/11/2024 22:36:28	Tìm kiếm
ID	Thiết Bị	Trạng Thái	Thời gian
1	Đèn	ON	05/11/2024 22:36:28
Trước Trang 1 / 1 Tiếp theo			

Hình 11: *Tìm kiếm theo định dạng YYYY-MM-DD HH:mm:ss:*

2. *Tìm kiếm theo định dạng YYYY-MM-DD:*

Lịch Sử Điều Khiển Thiết Bị			
Sắp xếp theo	10	05/11/2024	Tìm kiếm
ID	Thiết Bị	Trạng Thái	Thời gian
1	Đèn	ON	05/11/2024 22:36:28
2	Quạt	ON	05/11/2024 22:36:30
3	Quạt	OFF	05/11/2024 22:36:31
4	Điều Hòa	OFF	05/11/2024 22:36:32
5	Đèn	OFF	05/11/2024 22:36:33
6	Đèn	ON	05/11/2024 22:36:50
7	Điều Hòa	ON	05/11/2024 22:36:51
8	Quạt	ON	05/11/2024 22:36:53
9	Quạt	OFF	05/11/2024 22:36:54
10	Điều Hòa	OFF	05/11/2024 22:36:55
Trước Trang 1 / 7 Tiếp theo			

Hình 12: *Tìm kiếm theo định dạng YYYY-MM-DD:*

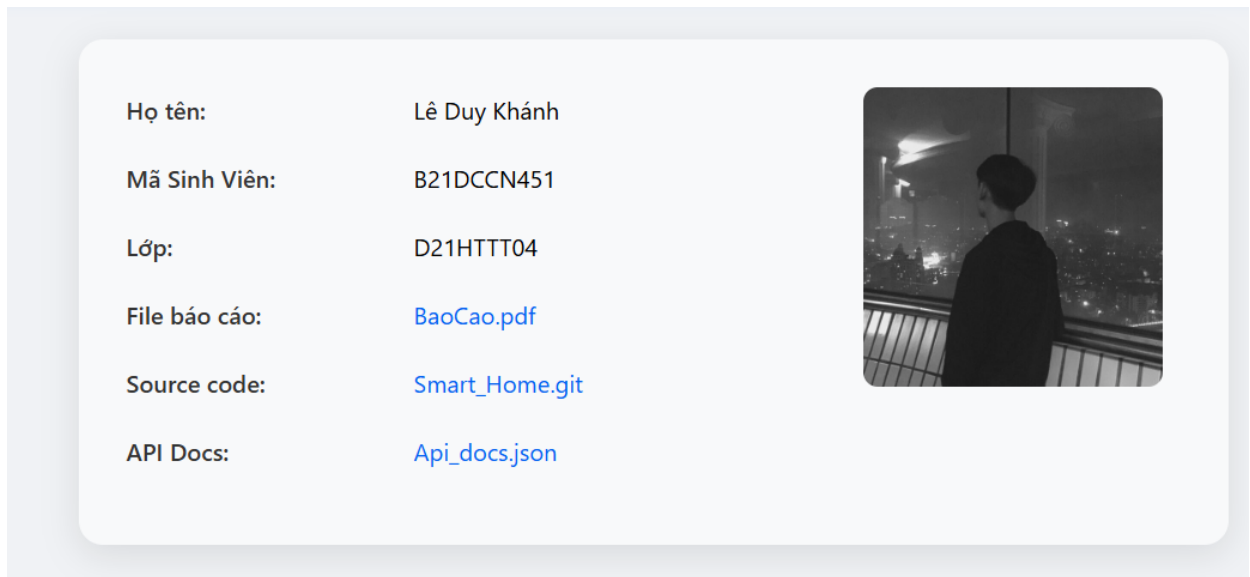
3. *Tìm kiếm theo định dạng HH:mm:ss:*

Lịch Sử Điều Khiển Thiết Bị			
Sắp xếp theo	10	22:37:17	Tìm kiếm
ID	Thiết Bị	Trạng Thái	Thời gian
30	Quạt	OFF	05/11/2024 22:37:17
Trước Trang 1 / 1 Tiếp theo			

Hình 13: *Tìm kiếm theo định dạng HH:mm:ss:*

### 3.4. Profile Pages:

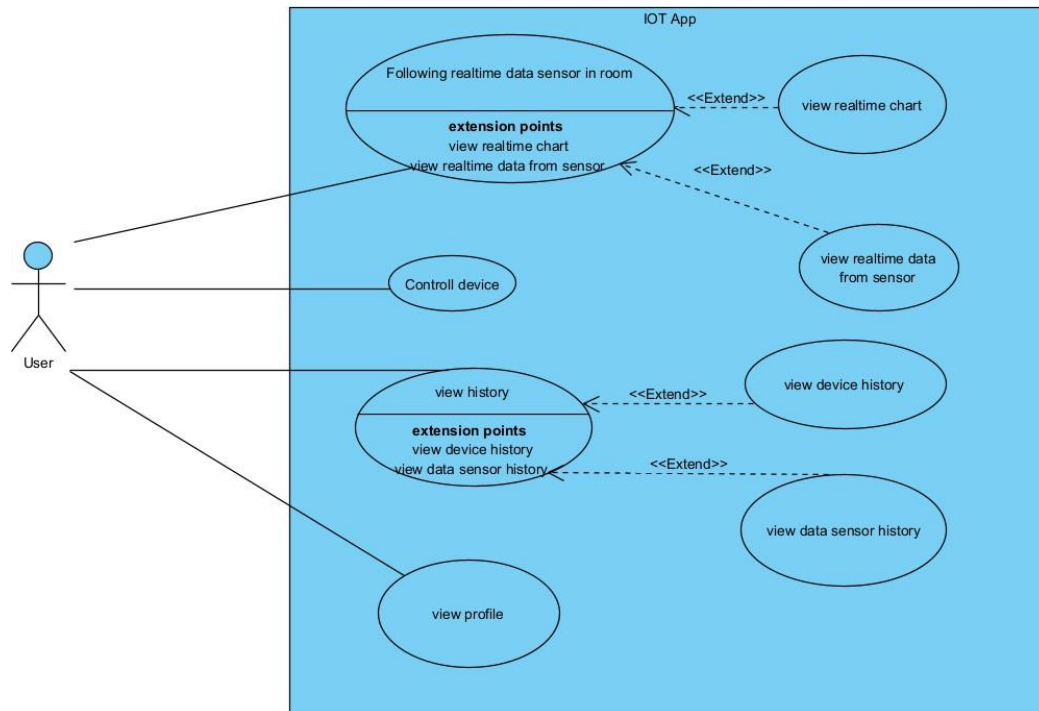
Giao diện hiển thị thông tin cá nhân và đường dẫn đến báo cáo.pdf, git project và api docs của dự án.



## CHƯƠNG 4 : THIẾT KẾ HỆ THỐNG

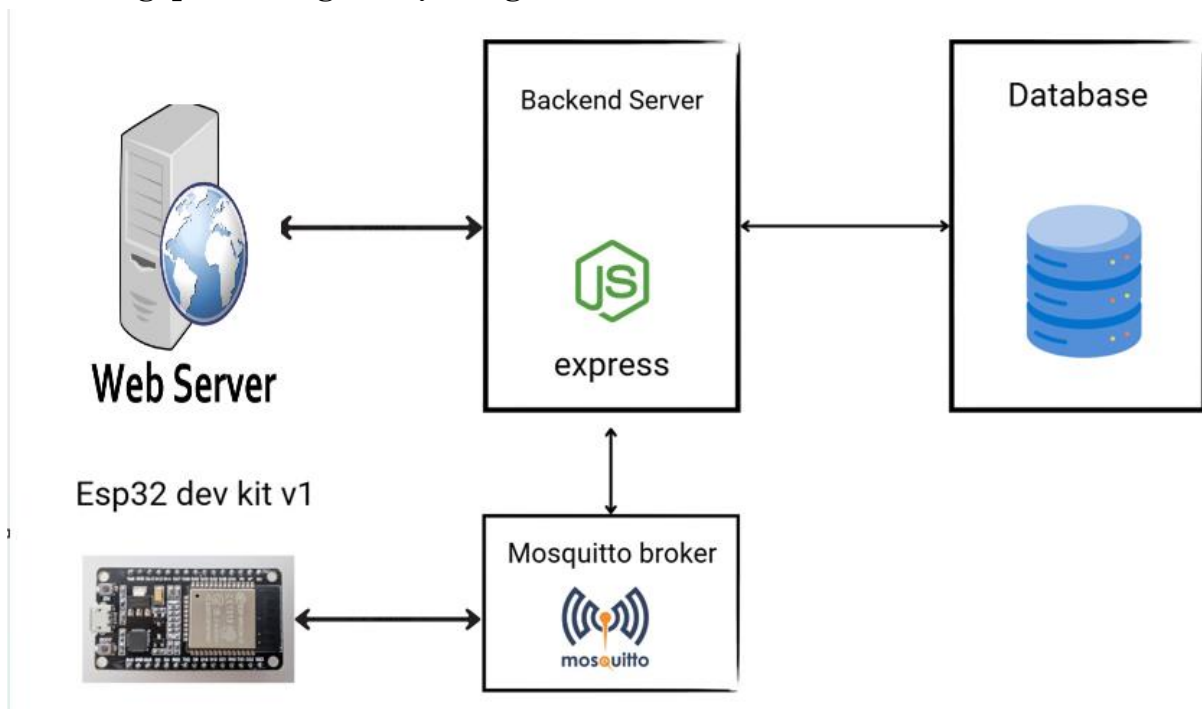
### 4.1: Sơ đồ usecase tổng quát

Sơ đồ Usecase tổng quát:



Hình 14: Sơ đồ usecase tổng quát

## 4.2: Tổng quan tương tác hệ thống:







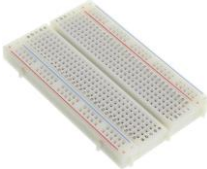

Hình 15: Tổng quan tương tác hệ thống

## 4.3. Phần cứng

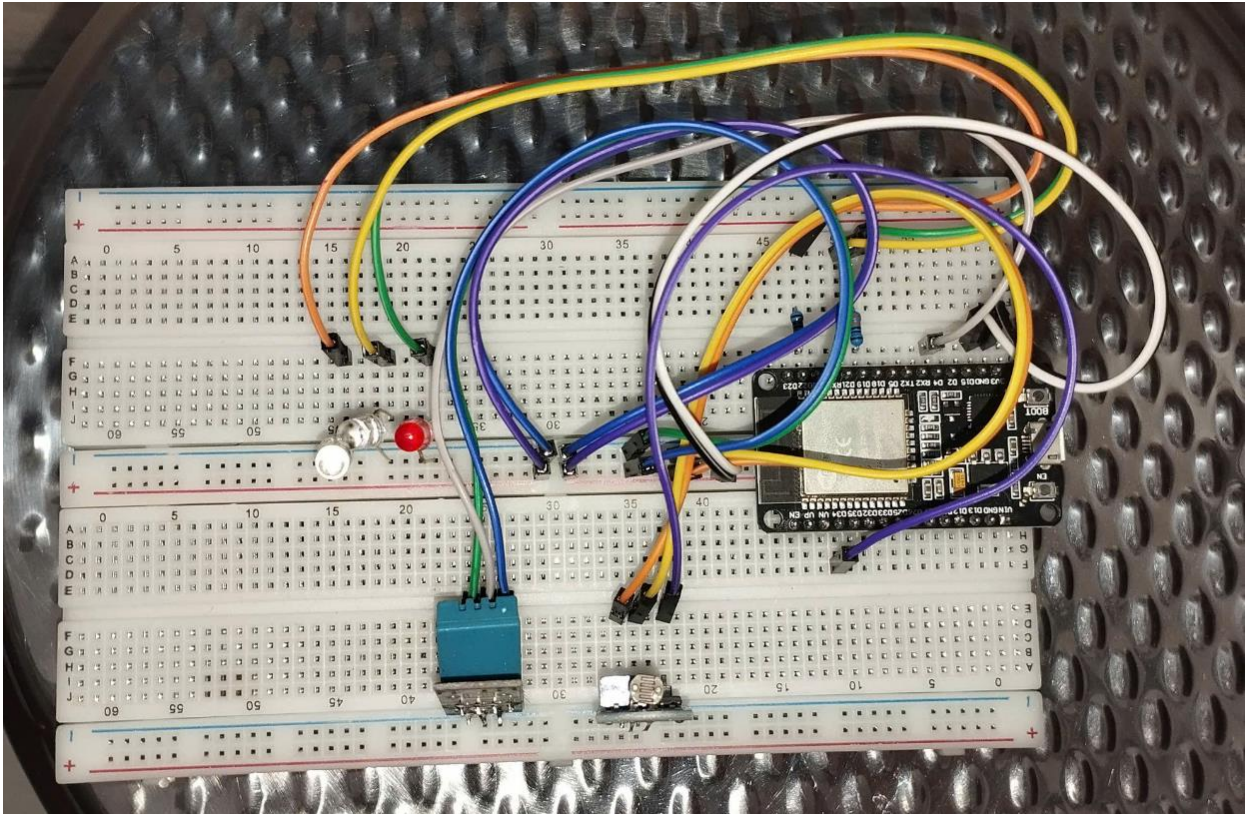
### 4.3.1 : Các thiết bị và các thành phần

Mạch bao gồm các thành phần

Thành phần	Hình ảnh
Mạch điều khiển Esp32	
Cảm biến nhiệt độ, độ ẩm DHT11	

Cảm biến ánh sáng: quang trở LDR	
Led 5mm	
Breadboard	
Dây nối	

#### 4.3.2: Thiết kế mạch:



*Hình 16: Thiết kế mạch*

#### **4.3.3 : Thực hiện nối mạch:**

- + DHT11
  - D4 (ESP32) - DATA (DHT11)
  - GND(ESP32) - GND(DHT11)
  - 3,3V(ESP32) - VCC(DHT11)
- + Led
  - Lần lượt các chân D19, D21, D22 với Anode Led1, Led2, Led3 đến led qua một điện trở

- Cathode của Led nối với GND(ESP32)
- + LDR
  - D34 (ESP32) - A0 (LDR)
  - Nối GND và VCC vào các chân tương ứng.

#### 4.3.4: Lập trình cho mạch điều khiển ESP32

Sử dụng Vscode với extension PlatformIO cho lập trình mạch điều khiển ESP32, với các thư viện cần thiết bao gồm Arduino, Wifi, DHT sensor Library, PubSubClient, ArduinoJson.

Các thành phần của mã nguồn:

##### 1. Khai báo thư viện và các biến

###### • Thư viện:

- WiFi.h: Kết nối WiFi.
- PubSubClient.h: Giao tiếp MQTT.
- DHT.h: Đọc dữ liệu từ cảm biến DHT.
- ArduinoJson.h: Phân tích và tạo dữ liệu JSON.

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <DHT.h>
#include <ArduinoJson.h>
```

Hình 17: Khai báo các thư viện

###### • Thông tin kết nối:

- ssid và password: Thông tin WiFi.
- mqttServer, mqttPort, mqttUser, mqttPassword: Thông tin kết nối MQTT.

```
6  const char* ssid = "khanh";
7  const char* password = "1234567899";
8  const char* mqttServer = "192.168.71.243";
9  const int mqttPort = 1884;
10 const char* mqttUser = "khanh";
11 const char* mqttPassword = "b21dccn451";
```

Hình 18: Các thông tin kết nối

###### • Khai báo cảm biến và chân kết nối:



- DHTPIN, DHTTYPE: Cấu hình chân và loại cảm biến DHT.
- LED1\_PIN, LED2\_PIN, LED3\_PIN: Chân điều khiển các thiết bị (Đèn, Điều Hòa, Quạt).
- Chân DHT11 được cấu hình trên chân số 4.
- Các chân điều khiển thiết bị (Đèn, Điều Hòa, Quạt) là 19, 21, 22.

```

13 #define DHTPIN 4
14 #define DHTTYPE DHT11
15 DHT dht(DHTPIN, DHTTYPE);
16
17 #define LED1_PIN 19
18 #define LED2_PIN 21
19 #define LED3_PIN 22

```

*Hình 19: Cấu hình dht11 và các chân đèn*

## 2. Cấu hình WiFi:

```

void setupWiFi() {
    delay(10);
    Serial.println();
    Serial.print("Kết nối đến WiFi");
    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    Serial.println("Đã kết nối đến WiFi");
}

```

*Hình 20: Cấu hình kết nối Wifi*

Hàm setupWiFi() kết nối ESP32 đến mạng WiFi:

- In thông báo kết nối và dấu . cho đến khi kết nối thành công.
- Khi kết nối thành công, in ra thông báo.

## 3. Kết nối MQTT

Hàm reconnect() đảm bảo ESP32 luôn kết nối đến máy chủ MQTT:

- Nếu chưa kết nối, in thông báo và thử kết nối lại.
- Nếu thành công, in ra thông báo và đăng ký chủ đề controldevice để nhận lệnh điều khiển.

```
void reconnect() {  
    while (!client.connected()) {  
        Serial.print("Kết nối đến MQTT...");  
        if (client.connect("ESP32Client", mqttUser, mqttPassword)) {  
            Serial.println("Đã kết nối");  
            client.subscribe("controldevice");  
        } else {  
            Serial.print("Lỗi, mã lỗi = ");  
            Serial.print(client.state());  
            delay(2000);  
        }  
    }  
}
```

*Hình 21: Hàm xử lý kết nối MQTT*

#### 4. Gửi dữ liệu cảm biến

Hàm sendSensorData() đọc dữ liệu từ các cảm biến và gửi qua MQTT:

- **Đọc nhiệt độ và độ ẩm** từ cảm biến DHT.
- **Đọc giá trị ánh sáng** từ cảm biến analog (chân 34) và chuyển đổi sang lux (công thức giả định).
- Kiểm tra dữ liệu đọc được có hợp lệ không.
- Tạo payload JSON chứa dữ liệu cảm biến và gửi lên chủ đề datasensor.
- In thông báo dữ liệu đã gửi.

```

void sendSensorData() {
    float temperature = dht.readTemperature();
    float humidity = dht.readHumidity();
    // Đọc giá trị analog từ chân cảm biến ánh sáng
    int lightAnalogValue = analogRead(34);

    // Chuyển đổi giá trị analog sang lux (công thức ví dụ, cần hiệu chỉnh theo cảm biến)
    float lightLux = lightAnalogValue * (3.3 / 4095.0); // Tính điện áp trên chân cảm biến
    lightLux = lightLux * 500; // Chuyển đổi điện áp sang lux (giả sử hệ số 500)

    if (isnan(temperature) || isnan(humidity)) {
        Serial.println("Lỗi đọc dữ liệu từ cảm biến!");
        return;
    }

    String payload = String("{\"temperature\":") + temperature + ", \"humidity\": " + humidity + ", \"light\": " + lightLux + "}";
    client.publish("datasensor", payload.c_str());
    Serial.println("Gửi dữ liệu cảm biến: " + payload);
}

```

*Hình 22: Hàm gửi dữ liệu cảm biến*

## 5. Xử lý lệnh điều khiển thiết bị

Hàm handleControlCommand() xử lý lệnh điều khiển từ chủ đề controldevice:

- Đọc dữ liệu lệnh và phân tích JSON.
- Kiểm tra tên thiết bị ("Đèn", "Điều Hòa", "Quạt") và điều khiển trạng thái (bật/tắt).
- Dựa trên giá trị "status" trong lệnh để bật/tắt chân tương ứng với thiết bị.

```

void handleControlCommand(char* topic, byte* payload, unsigned int length) {
    String message;
    for (int i = 0; i < length; i++) {
        message += (char)payload[i];
    }

    Serial.println("Nhận lệnh điều khiển: " + message);

    DynamicJsonDocument doc(1024);
    deserializeJson(doc, message);
    const char* device = doc["device"];
    bool status = doc["status"] == "ON";

    if (strcmp(device, "Đèn") == 0) {
        digitalWrite(LED1_PIN, status ? HIGH : LOW);
    } else if (strcmp(device, "Điều Hòa") == 0) {
        digitalWrite(LED2_PIN, status ? HIGH : LOW);
    } else if (strcmp(device, "Quạt") == 0) {
        digitalWrite(LED3_PIN, status ? HIGH : LOW);
    }
}

```

*Hình 23: Hàm xử lý điều khiển thiết bị*

## 6. Hàm setup()

Thiết lập ban đầu:

- Khởi động cổng Serial để debug.
- Khởi động cảm biến DHT.
- Cấu hình chân thiết bị là OUTPUT.
- Gọi hàm setupWiFi() để kết nối WiFi.
- Cấu hình MQTT server và callback để xử lý lệnh.

```
void setup() {
  Serial.begin(115200);
  dht.begin();

  pinMode(LED1_PIN, OUTPUT);
  pinMode(LED2_PIN, OUTPUT);
  pinMode(LED3_PIN, OUTPUT);

  setupWiFi();
  client.setServer(mqttServer, mqttPort);
  client.setCallback(handleControlCommand);
}
```

*Hình 23: Hàm xử lý*

## 7. Hàm loop()

Vòng lặp chính:

- Kiểm tra và kết nối lại MQTT nếu cần.
- Gọi client.loop() để xử lý các lệnh từ MQTT.
- Gửi dữ liệu cảm biến sau mỗi 30 giây bằng cách kiểm tra thời gian.

```
void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();

  static unsigned long lastSendTime = 0;
  if (millis() - lastSendTime > 30000) {
    sendSensorData();
    lastSendTime = millis();
  }
}
```

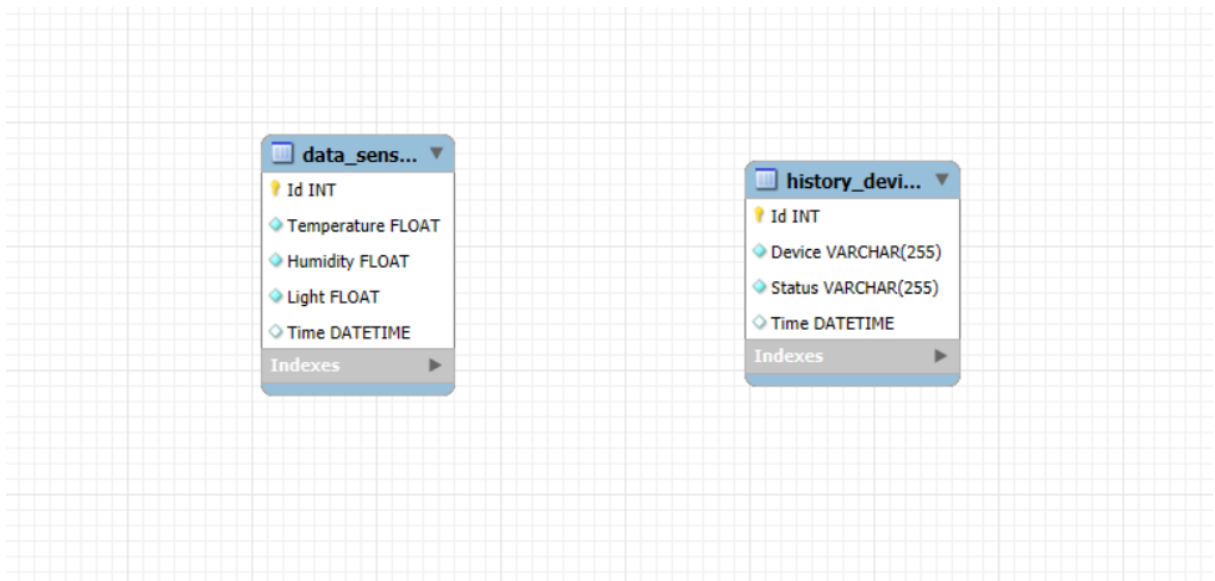
*Hình 24: Vòng lặp*

Kết nối:

Chương trình từ ESP32 được cài đặt để pub/sub tới Mosquitto Broker. Điều này giúp giao tiếp giữa Backend Server và ESP32 thông qua giao thức mqtt (một giao thức phù hợp cho các thiết bị IoT).

## 4.4. Backend

### 4.4.1 : Thiết kế cơ sở dữ liệu



Hình 12: Thiết kế cơ sở dữ liệu

Các bảng trong database:

- control\_history: chứa lịch sử bật tắt các Led

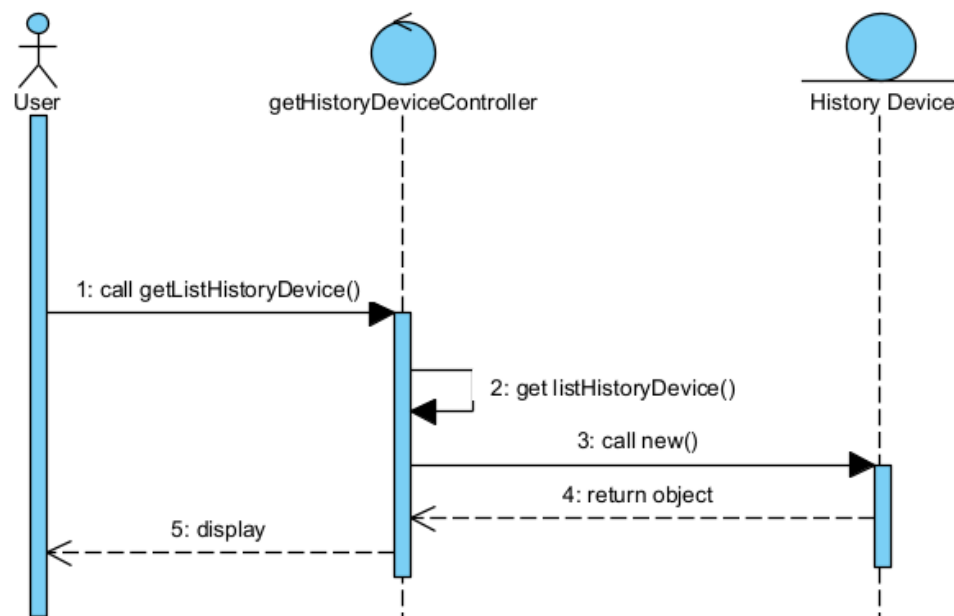
Cột	Kiểu dữ liệu	Mô tả
Id	INT (PK)	id
Device	VARCHAR(255)	tên của thiết bị
Status	VARCHAR(255)	ON, OFF
Time	DATETIME	

- sensordata: lịch sử dữ liệu từ 3 độ đo cảm biến

Cột	Kiểu dữ liệu	Mô tả
Id	INT (PK)	id
Temperature	FLOAT	nhiệt độ
Humidity	FLOAT	độ ẩm
Light	FLOAT	ánh sáng
Time	DATETIME	

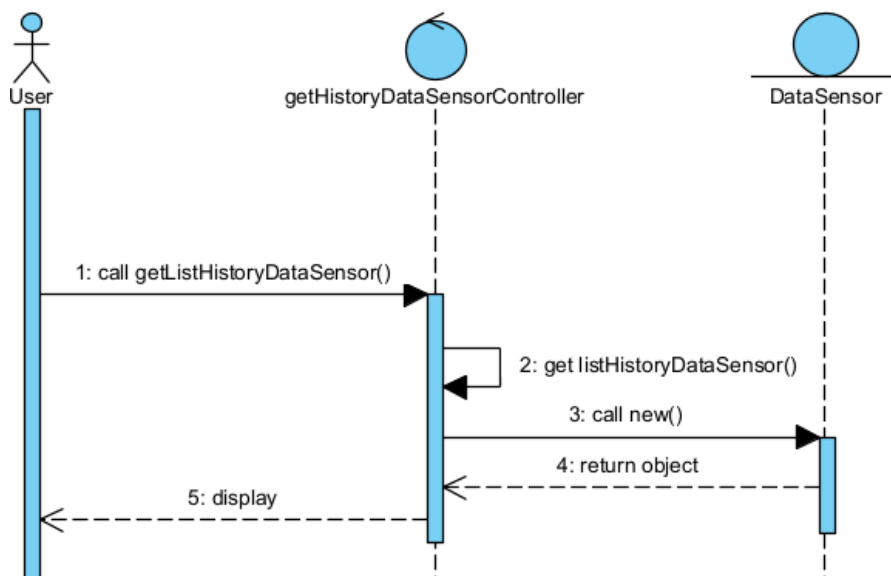
#### 4.4.2: Luồng hoạt động:

- + /api/get\_control\_history



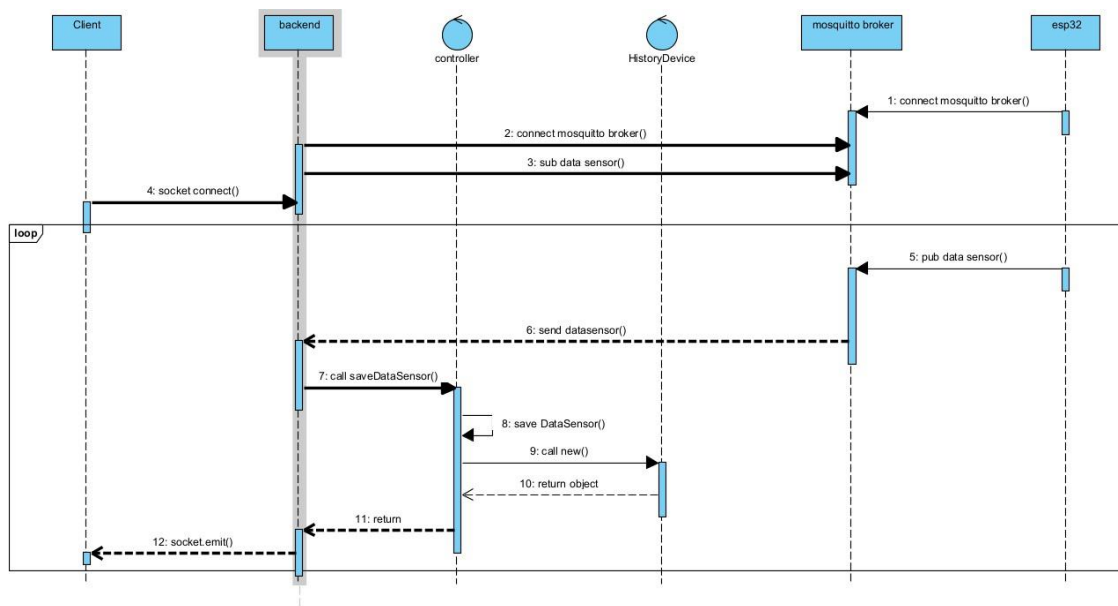
Hình 13:Luồng hoạt động

+ /api/get\_sensor\_data



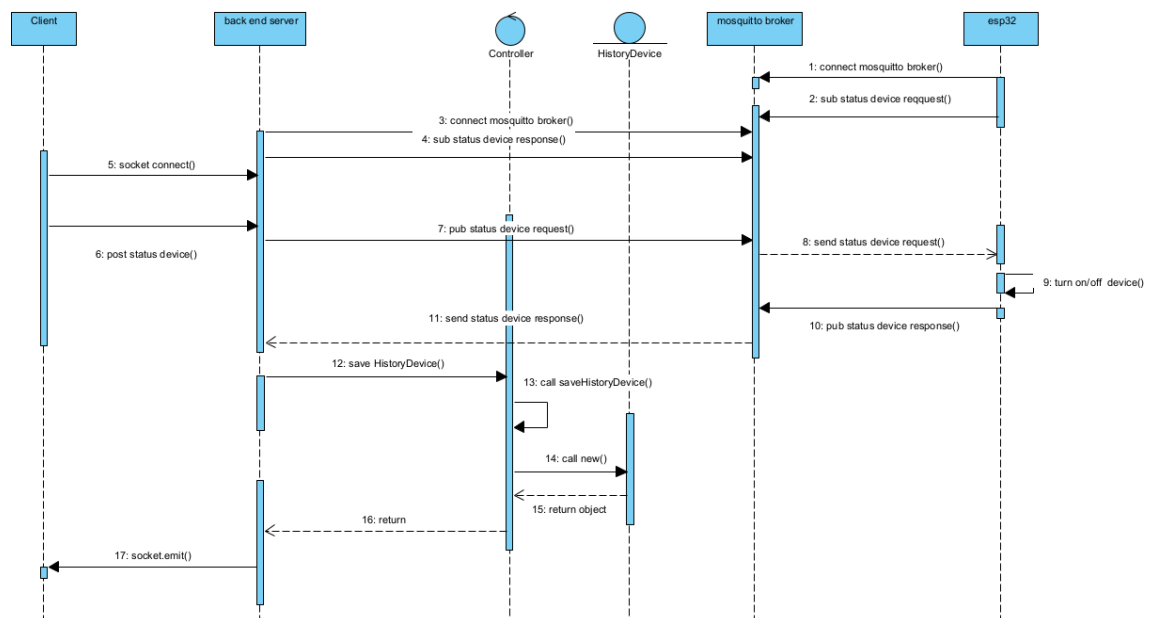
Hình 14: Luồng hoạt động

+ Dữ liệu realtime gửi từ cảm biến



Hình 15: Luồng hoạt động

+ Điều khiển các thiết bị:

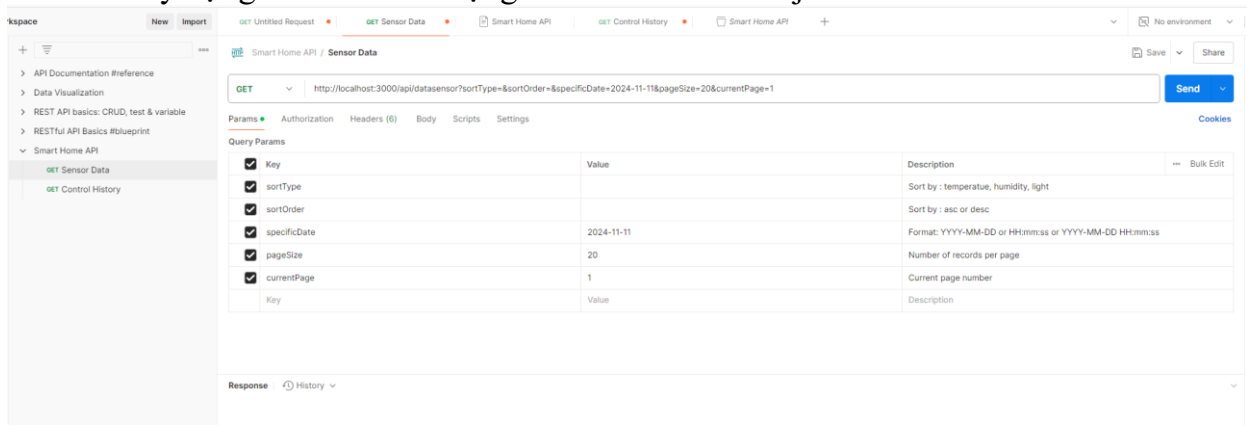


Hình 16: Luồng hoạt động



## 4.5: API Docs:

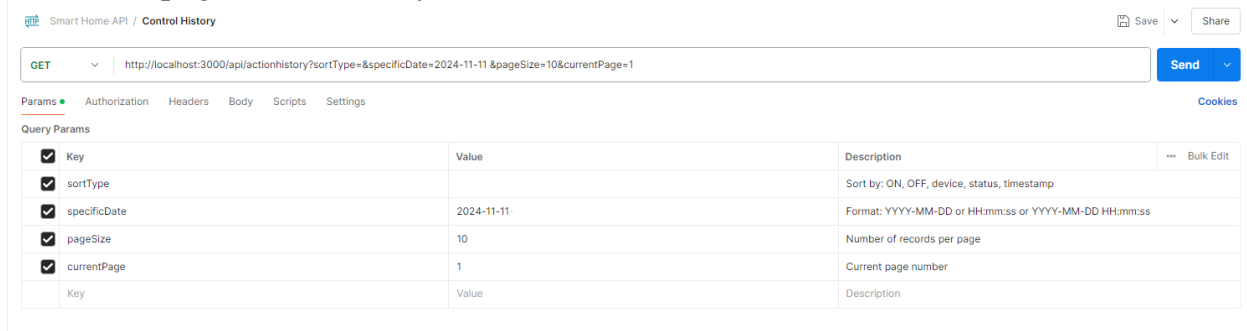
### Xây dựng API Docs sử dụng Postman cho Node.js:



Hình 17: API Docs Postman

### 4.5.1 : Api sử dụng để lấy dữ liệu liên quan đến lịch sử bật tắt các thiết bị

+ /api/get\_actionhistory



Hình 22: Api / api/get\_actionhistory

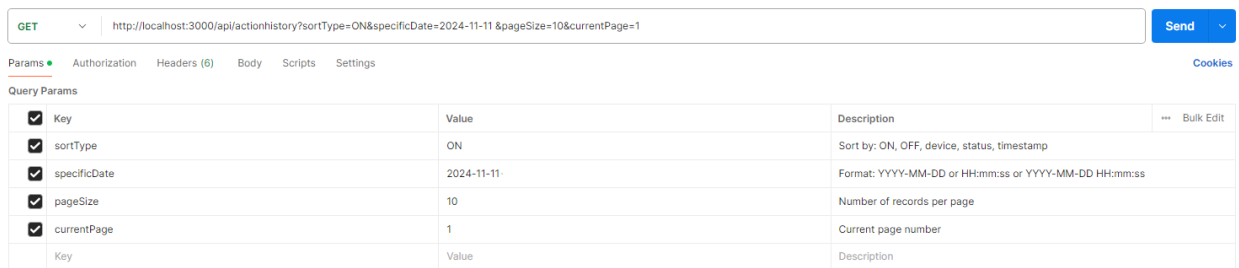
Api sẽ có 4 tham số truyền vào:

- + sortType: Xác định sắp xếp theo kiểu dữ liệu nào
- + specificDate: xác định tìm kiếm theo định dạng nào ( Format: YYYY-MM-DD or HH:mm:ss or YYYY-MM-DD HH:mm:ss )
- + Currentpage: trang hiện tại
- + pageSize: kích thước trang

Api sẽ trả về:

- + data: dữ liệu liên quan đến lịch sử bật tắt thiết bị
- + meta: dữ liệu liên quan đến phân trang

Demo:



The screenshot shows a REST client interface with a GET request to `http://localhost:3000/api/actionhistory?sortType=ON&specificDate=2024-11-11 &pageSize=10&currentPage=1`. Below the URL bar, the 'Query Params' section is expanded, displaying a table of the request's query parameters.

Key	Value	Description
sortType	ON	Sort by: ON, OFF, device, status, timestamp
specificDate	2024-11-11	Format: YYYY-MM-DD or HH:mm:ss or YYYY-MM-DD HH:mm:ss
pageSize	10	Number of records per page
currentPage	1	Current page number

*Hình 23: Chọn kiểu Sort*

```
Body Cookies Headers (8) Test Results ↺
Pretty Raw Preview Visualize JSON ↕
42     "status": "ON",
43     "timestamp": "2024-11-08T14:54:07.000Z"
44   },
45   {
46     "id": 74,
47     "device": "Đèn",
48     "status": "ON",
49     "timestamp": "2024-11-08T14:55:37.000Z"
50   },
51   {
52     "id": 75,
53     "device": "Điều Hòa",
54     "status": "ON",
55     "timestamp": "2024-11-08T14:55:38.000Z"
56   },
57   {
58     "id": 80,
59     "device": "Quạt",
60     "status": "ON",
61     "timestamp": "2024-11-08T14:55:43.000Z"
62   }
63 ],
64 "totalRecords": 82,
65 "totalPages": 9,
66 "currentPage": 1,
67 "pageSize": 10
68 }
```

Hình 24: Dữ liệu trả về

### 4.5.2: Api lấy dữ liệu về các thông số của cảm biến:

+ /api/get\_datasensor

Smart Home API / Get Sensor Data Save Share

GET ▼  Send ▼

Params • Authorization Headers (6) Body Scripts Settings Cookies

Query Params

<input checked="" type="checkbox"/> Key	Value	Description	⋮ Bulk Edit
<input checked="" type="checkbox"/> sortType	temperature	Field to sort by: temperature, humidity, light, timestamp	
<input checked="" type="checkbox"/> sortOrder	desc	Sort direction: asc, desc	
<input checked="" type="checkbox"/> specificDate	2024-11-08 22:00:57	Filter by date/time. Formats: YYYY-MM-DD, HH:mm:ss, YYYY-MM-DD HH:m...	
<input checked="" type="checkbox"/> pageSize	10	Number of records per page (default: 10)	
<input checked="" type="checkbox"/> currentPage	1	Page number (default: 1)	🗑
Key	Value	Description	

Hình 25: Api /api/get\_datasensor

Body Cookies Headers (8) Test Results 🔄

Pretty Raw Preview Visualize JSON ⌵ ⌵

```
51  },
52  {
53      "id": 63,
54      "temperature": 27.1,
55      "humidity": 17,
56      "light": 1114.91,
57      "timestamp": "2024-11-08T15:30:28.000Z"
58  },
59  {
60      "id": 57,
61      "temperature": 27.1,
62      "humidity": 17,
63      "light": 1114.1,
64      "timestamp": "2024-11-08T15:27:28.000Z"
65  },
66  {
67      "id": 58,
68      "temperature": 27.1,
69      "humidity": 17,
70      "light": 1113.3,
71      "timestamp": "2024-11-08T15:27:58.000Z"
72  }
73  ],
74  "totalRecords": 139,
75  "totalPages": 14,
76  "currentPage": 1,
77  "pageSize": 10
78  }
```

Hình 26: dữ liệu trả về

Api sẽ có 5 tham số truyền vào:

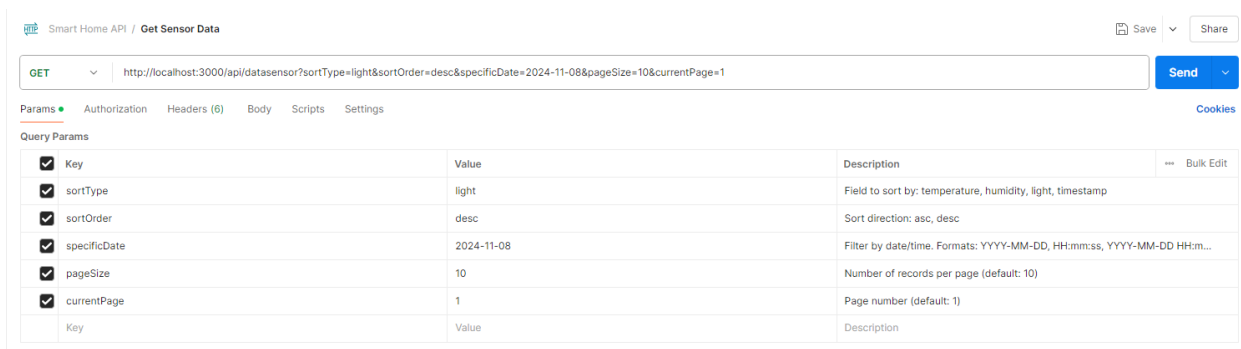
- + sortType: Xác định sắp xếp theo kiểu dữ liệu nào
- + sortOrder : Xác định sắp xếp theo tăng dần hay giảm dần
- + specificDate: xác định tìm kiếm theo định dạng nào ( Format: YYYY-MM-DD or HH:mm:ss or YYYY-MM-DD HH:mm:ss )
- + Currentpage: trang hiện tại
- + pageSize: kích thước trang

Api sẽ trả về:

- + data: dữ liệu liên quan đến lịch sử bật tắt thiết bị
- + meta: dữ liệu liên quan đến phân trang

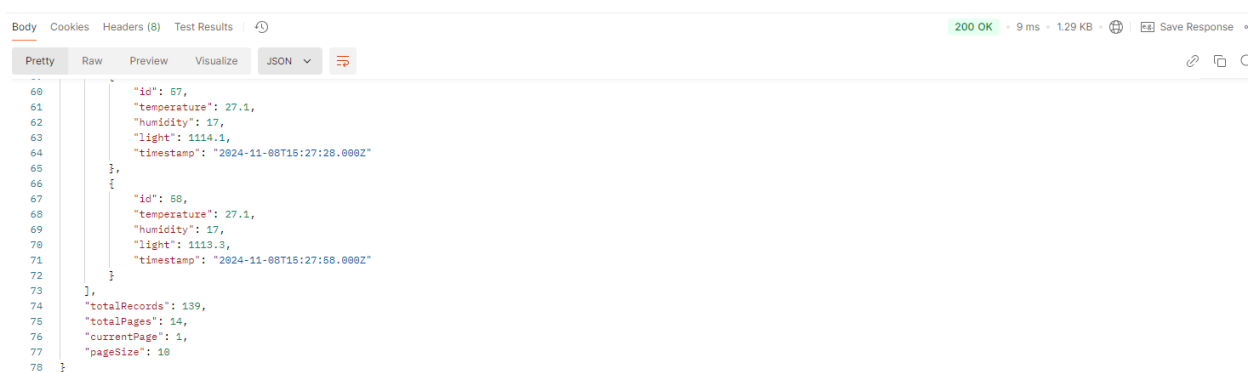
Demo

- + Thực hiện lấy dữ liệu



Hình 27: Chọn kiểu dữ liệu

- + Dữ liệu trả về



Hình 28: dữ liệu trả về cho api datasensor

### 4.5.3 : Api lấy dữ liệu hiển thị lên giá trị đo và biểu đồ:

+ /api/get\_display

Demo lấy dữ liệu hiển thị 3 giá trị đo mới nhất:

Smart Home API / Get Display Data

GET http://localhost:3000/api/display?type=values

Params Authorization Headers (6) Body Scripts Settings

Query Params

<input checked="" type="checkbox"/>	Key	Value
<input checked="" type="checkbox"/>	type	values
	Key	Value

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 272,
3   "temperature": 30.8,
4   "humidity": 15,
5   "light": 372.71,
6   "timestamp": "2024-11-12T15:00:20.000Z"
7 }
```

Demo lấy dữ liệu hiển thị 5 giá trị đo mới nhất cho biểu đồ:

Smart Home API / Get Display Data

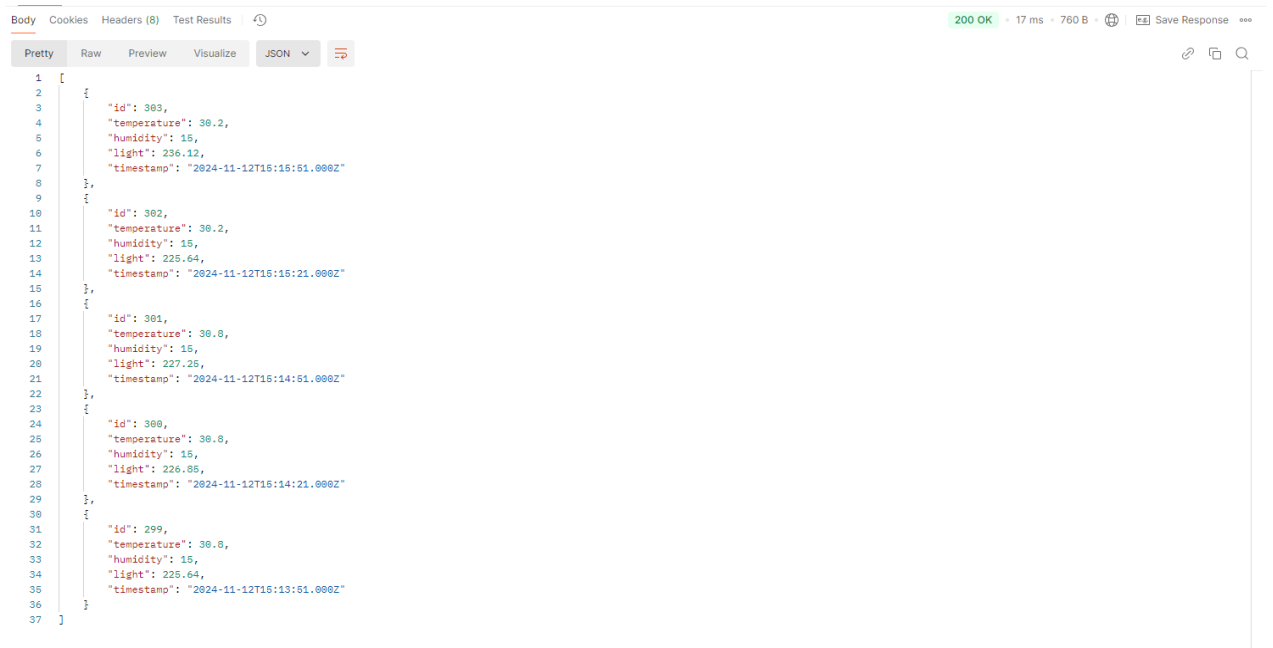
GET http://localhost:3000/api/display?type=chart

Params Authorization Headers (6) Body Scripts Settings

Query Params

<input checked="" type="checkbox"/>	Key	Value
<input checked="" type="checkbox"/>	type	chart
	Key	Value

## + Dữ liệu trả về

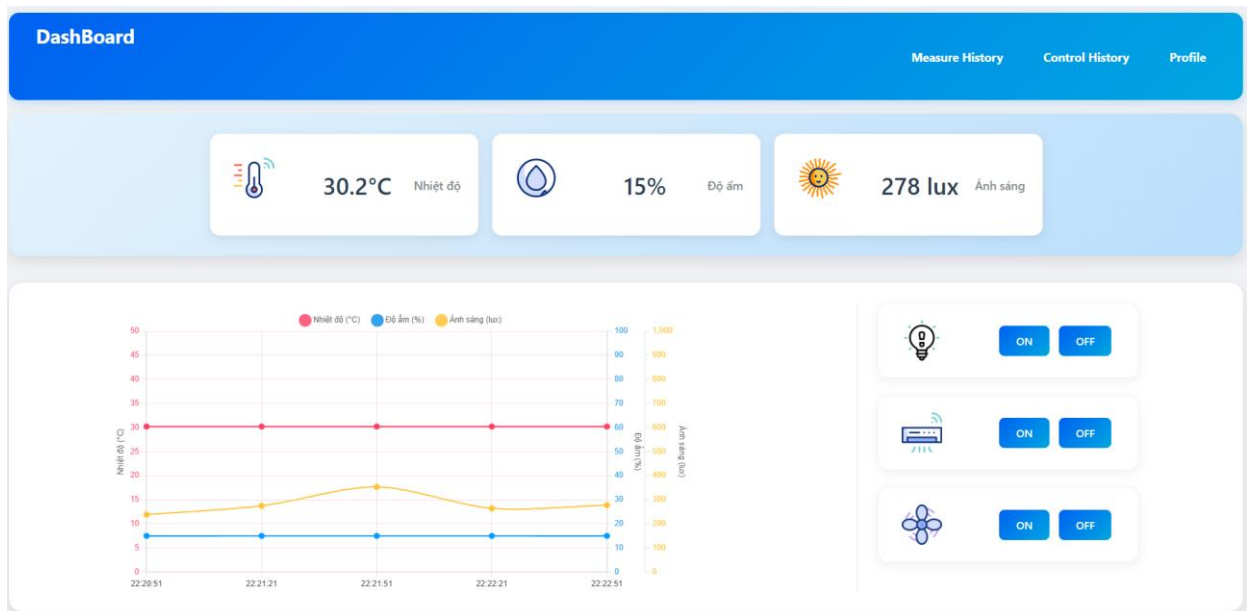


Hình 32: Dữ liệu trả về

#### 4.6: Frontend :

Trong dự án, phần front-end được phát triển sử dụng HTML, CSS và JavaScript nhằm xây dựng giao diện người dùng trực quan và dễ sử dụng. Cụ thể:

- HTML: Được sử dụng để cấu trúc nội dung của trang web, tạo các phần tử cơ bản như biểu mẫu, bảng dữ liệu, và các thẻ điều khiển khác.
- CSS: Đảm nhận vai trò định dạng giao diện, giúp tạo ra bố cục hấp dẫn, màu sắc hài hòa, và hiệu ứng hiển thị mượt mà để cải thiện trải nghiệm người dùng.
- JavaScript: Được triển khai để thêm tính tương tác động, xử lý sự kiện, cập nhật nội dung không cần tải lại trang, và thực hiện các chức năng như gọi API, cập nhật dữ liệu theo thời gian thực và điều khiển trạng thái thiết bị.



Hình 33:Fontend



#### 4.6.1. Lấy dữ liệu từ Server

Dự án Smart Home sử dụng kiến trúc Backend theo mô hình MVC (Model-View Controller) với Node.js, được tổ chức như sau:

Kiến trúc tổng quan:

- Áp dụng mô hình MVC để phân tách các thành phần
- RESTful API cho giao tiếp client-server
- Tích hợp MQTT cho giao tiếp real-time với thiết bị IoT
- Sử dụng MySQL làm cơ sở dữ liệu

```
back-end > api > JS apiRoutes.js > ...
1  'use strict';
2  const express = require('express');
3  const router = express.Router();
4  const asyncHandler = require('../middlewares/asyncHandler');
5  const Controller = require('../controllers/Controller');
6
7  // Sensor data routes
8  router.get('/datasensor', asyncHandler(Controller.handleGetLatestSensorData));
9  // Device control and history routes
10 router.get('/actionhistory', asyncHandler(Controller.handleGetActionHistory));
11 // Display data
12 router.get('/display', asyncHandler(Controller.handleGetDisplayData));
13
14 module.exports = router;
15 | Ctrl+L to chat, Ctrl+K to generate
```

Hình 34: Các api được định nghĩa trong source

## CHƯƠNG 5 : KẾT QUẢ

Github dự án:

- + Android App:<https://github.com/Maybetuandat/Smart-Home-PTIT.git>
- + Backend Server: [https://github.com/Maybetuandat/back\\_end\\_iot.git](https://github.com/Maybetuandat/back_end_iot.git)
- + Code esp32:[https://github.com/Maybetuandat/iot\\_ioplatform.git](https://github.com/Maybetuandat/iot_ioplatform.git)