

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN  
KHOA CÔNG NGHỆ THÔNG TIN



---

**ĐỒ ÁN CUỐI KÌ - NHÓM 10**  
VẬT LÝ CHO CNTT

---

TRIỆU NHẬT MINH - 21127112  
BÙI ĐỖ DUY QUÂN - 21127141  
HOÀNG ĐỨC VIỆT - 21127203

Giảng viên hướng dẫn:  
Cao Xuân Nam  
Đặng Hoài Thương

Ngày 21 tháng 8 năm 2023

# Mục lục

|     |   |    |
|-----|---|----|
| 1   | Thông tin nhóm . . . . .                                    | 2  |
| 2   | Sản phẩm . . . . .  | 2  |
| 2.1 | Tên sản phẩm . . . . .                                      | 2  |
| 2.2 | Mô tả toàn chức năng của hệ thống . . . . .                 | 2  |
| 2.3 | Sơ đồ truyền và nhận dữ liệu của các thiết bị IOT và server | 7  |
| 2.4 | Thiết kế 3D sản phẩm của nhóm . . . . .                     | 8  |
| 2.5 | Giao diện và chức năng của web . . . . .                    | 11 |
| 2.6 | Xây dựng flow trong NodeRed . . . . .                       | 15 |
| 3   | Phân công công việc . . . . .                               | 23 |
| 4   | Tài liệu tham khảo . . . . .                                | 24 |

## 1 Thông tin nhóm

Thứ tự nhóm: Nhóm 10

Danh sách thành viên:

- Triệu Nhật Minh - 21127112
- Bùi Đỗ Duy Quân - 21127141
- Hoàng Đức Việt - 21127203

## 2 Sản phẩm

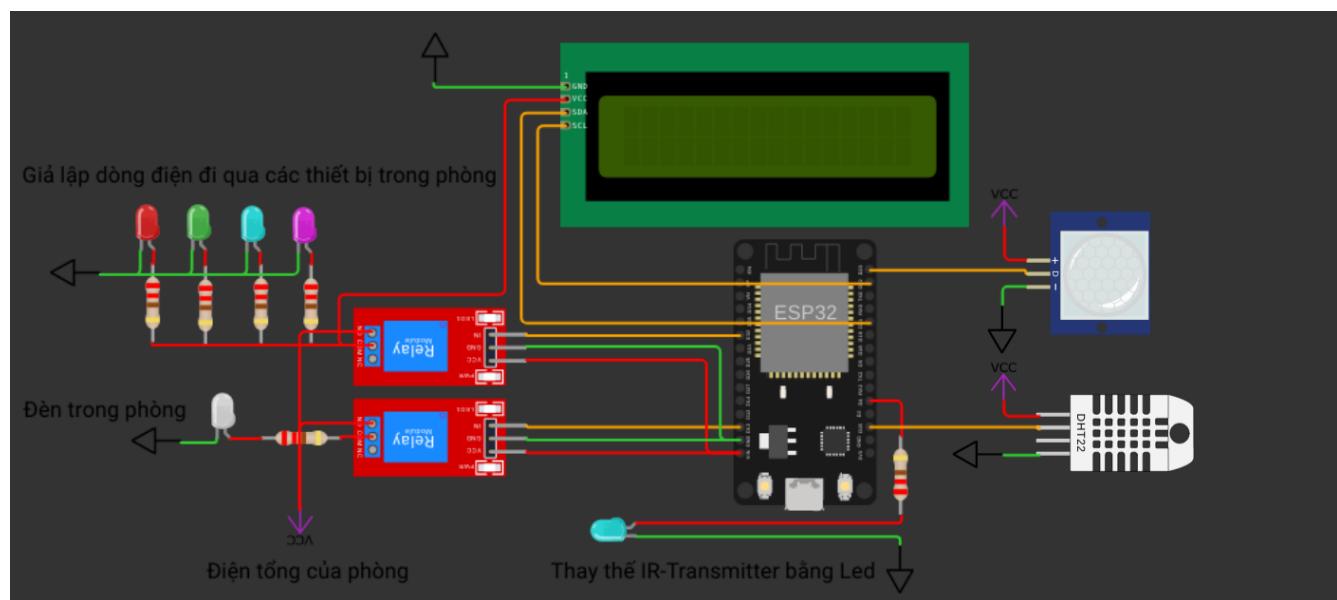
### 2.1 Tên sản phẩm

Thiết bị hỗ trợ sử dụng điện thông minh trong phòng

### 2.2 Mô tả toàn chức năng của hệ thống

#### a Hình ảnh sản phẩm

Trong đồ án này, nhóm đã chọn thực hiện giả lập sản phẩm trên nền tảng **Wokwi** và dưới đây là hình ảnh các thiết bị và mạch nhóm đã thiết kế.



Hình 1: Giả lập thiết bị hỗ trợ sử dụng điện thông minh trong phòng

#### b Các thiết bị sử dụng chính

Dựa trên yêu cầu và qui định của đồ án cuối kì thì nhóm đã kết hợp các thiết bị cho sản phẩm ở hình trên như sau:

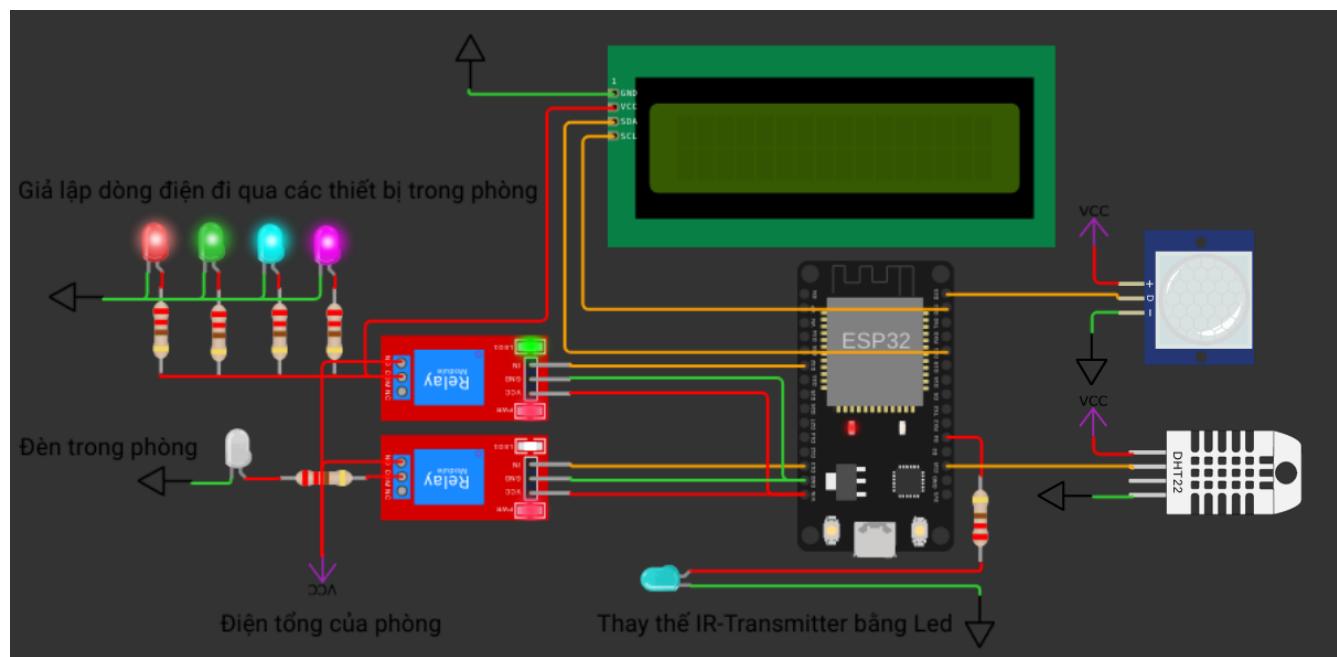
- Output: LCD, relay, IR-Transmitter
- Input: DHT22, cảm biến chuyển động (dạng hồng ngoại PIR).
- Board ESP32 có kết nối wifi.

#### c Các thiết bị khác trong hình ảnh của sản phẩm

- Như đã được chú thích trong hình, vì không thể giả lập thiết bị IR-Transmitter nên nhóm đã sử dụng 1 bóng đèn led màu xanh dương (được nối trực tiếp với board ESP32). Khi bóng đèn led này chớp tắt tức là thiết bị đã gửi tín hiệu hồng ngoại tới các thiết bị trong phòng.
- Bóng đèn led màu trắng chính là đèn chính trong phòng.
- 4 bóng đèn led còn lại được dùng để tương trưng cho các thiết bị trong phòng đang có dòng điện đi qua. Nếu có điện trong phòng, 4 bóng đèn led này sẽ phát sáng tương trưng cho việc các thiết bị điện trong phòng đang có điện.

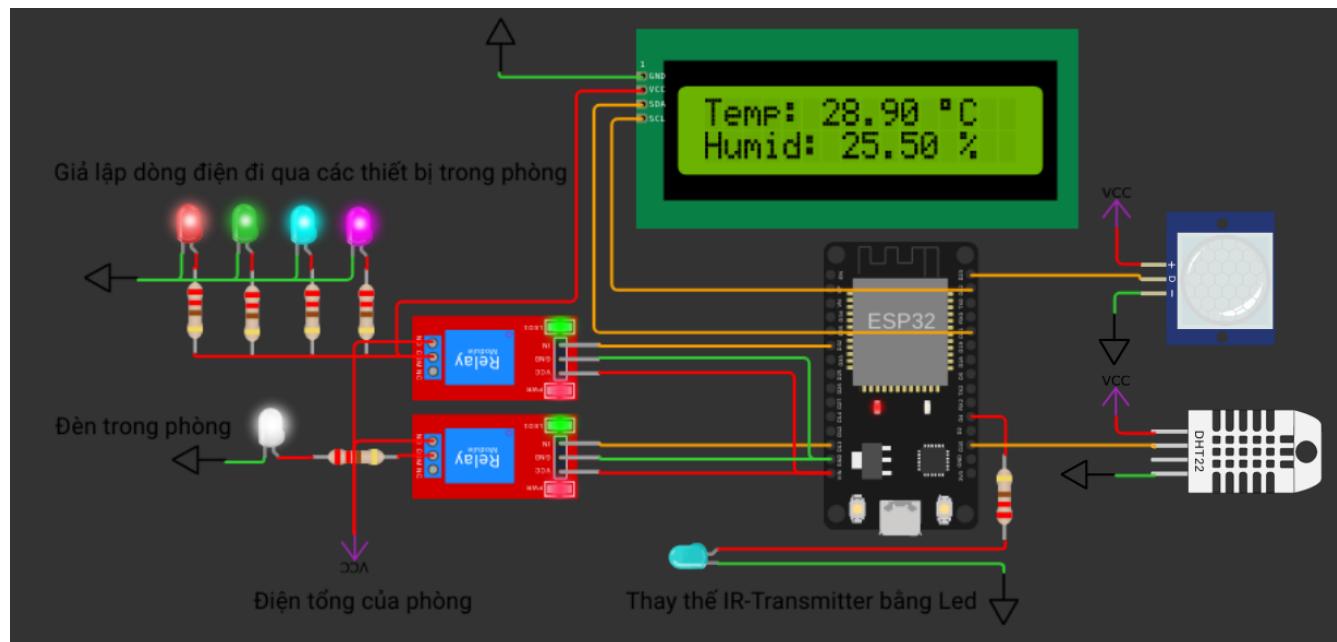
#### d Tổng quát cách thức hoạt động của sản phẩm

- Sản phẩm này của nhóm sẽ nhận biết rằng người trong phòng hay không để thực hiện mở các thiết bị điện cần thiết. Thêm vào đó người dùng có thể kiểm soát được 2 chế độ "hoạt động" và "bảo vệ", trong đó: chế độ "**hoạt động**" sẽ giúp người dùng bật tắt thiết bị điện khi người dùng ra hay vào phòng, và chế độ "**bảo vệ**" sẽ ngắt cung cấp điện cho các thiết bị đồng thời cảnh báo người dùng khi có người vào phòng.
- Ngay khi khởi tạo, thiết bị **ESP32** sẽ thực hiện kết nối wifi, nếu việc kết nối có vấn đề sẽ tiếp tục thực hiện kết nối **trong vòng 5 phút**. Nếu sau 5 phút không kết nối được thì các chức năng xem thông tin trên **trang web hay điện thoại** sẽ không thể sử dụng, và để thực hiện kết nối lại wifi sẽ thực hiện lại sau **5 phút từ khi việc kết nối cuối cùng thất bại**.
- Trong trường hợp kết nối được wifi thì **ESP32** sẽ tiến hành kết nối **MQTT server** và **cloud server (Thingspeak)**. Việc cố gắng thực hiện kết nối giữa 2 server cũng sẽ diễn ra trong 5 phút và sẽ thực hiện kết nối lại cũng sau 5 phút kể từ lần cuối kết nối thất bại trong trường hợp đã kết nối wifi.
- Ngay sau khi các kết nối thành công, thiết bị sẽ mặc định được cài đặt chế độ "**hoạt động**".



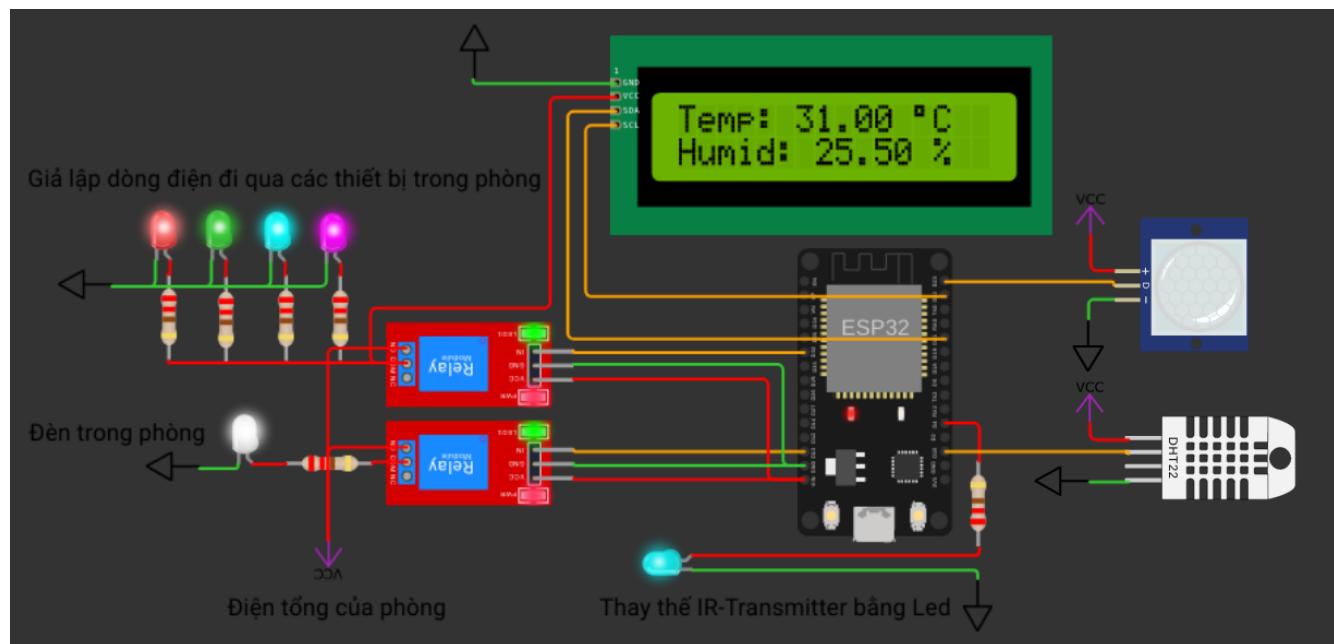
Hình 2: khi các kết nối thành công sẽ khởi tạo chế độ "hoạt động". 4 đèn led sáng tức là các thiết bị điện trong phòng đang được cung cấp điện

- Khi có người bước vào phòng, **cảm biến chuyển động** sẽ nhận tín hiệu hồng ngoại và đèn trong phòng sẽ sáng, đồng thời trên màn hình **LCD** sẽ hiện lên thông tin về nhiệt độ, độ ẩm trong phòng.



Hình 3: Đèn trong phòng sáng và thông tin được hiển thị trên màn hình LCD

- Ngoài ra khi người dùng trong phòng và nhiệt độ lớn hơn 30°C thì máy lạnh sẽ được tự động mở thông qua thiết bị **IR-Transmitter**.



Hình 4: Đèn led màu xanh đại diện cho IR-Transmitter sáng tức là đã phát tín hiệu hồng ngoại

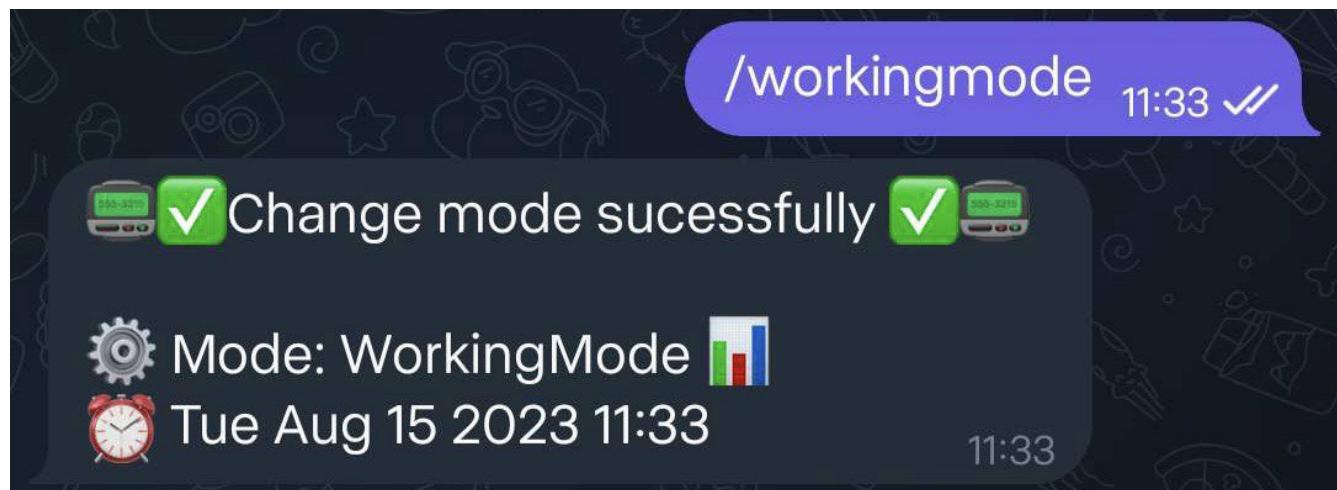
- Sau khi không còn ai trong phòng thì các thiết bị sẽ được tắt sau 1 khoảng thời gian nhất định, cụ thể **máy lạnh** sẽ được tắt sau khoảng 30 phút tính từ thời điểm không có người nào trong phòng, còn **các thiết bị khác** sẽ tắt sau khoảng 2 phút.
- Đối với trường hợp "**Bảo vệ**", các thiết bị trong phòng sẽ không được cung cấp nguồn điện, và khi có người vào trong phòng, sản phẩm sẽ cảnh báo về **chat bot telegram** rằng có người vào phòng.



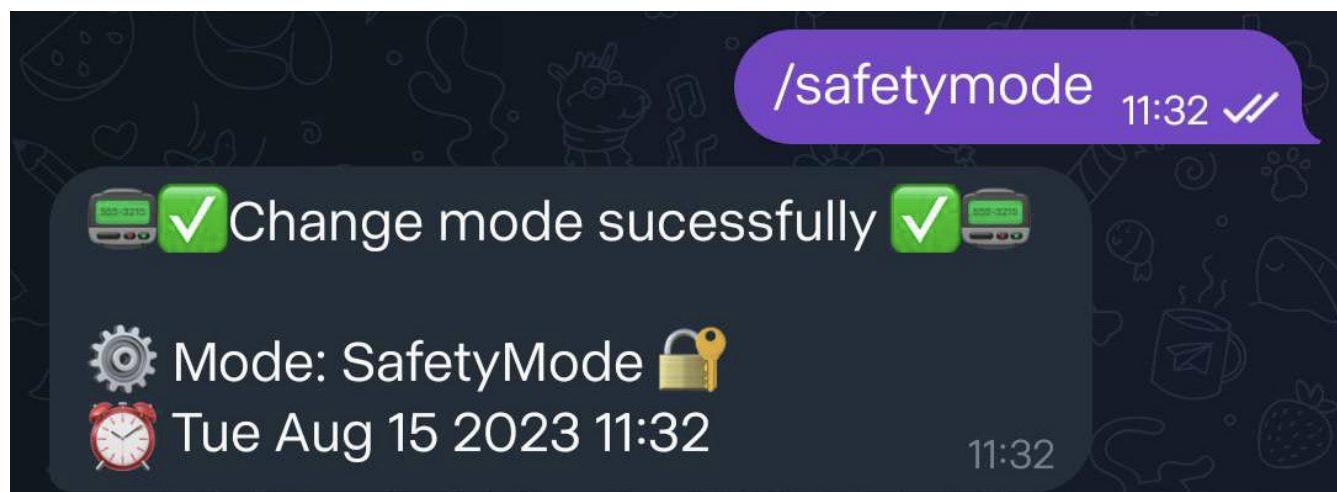
Hình 5: Tin nhắn cảnh báo từ chat bot telegram

- Ngoài ra, nhóm đã thiết lập cho phép thay đổi 2 chế độ này thông qua các

lệnh được cài đặt trong khung chat bot của telegram, giúp người dùng có thể thay đổi nhanh chóng. Lệnh để chuyển đổi sang chế độ "**hoạt động**" là "**/workingmode**" và chuyển sang chế độ "**bảo vệ**" là "**/safetymode**".

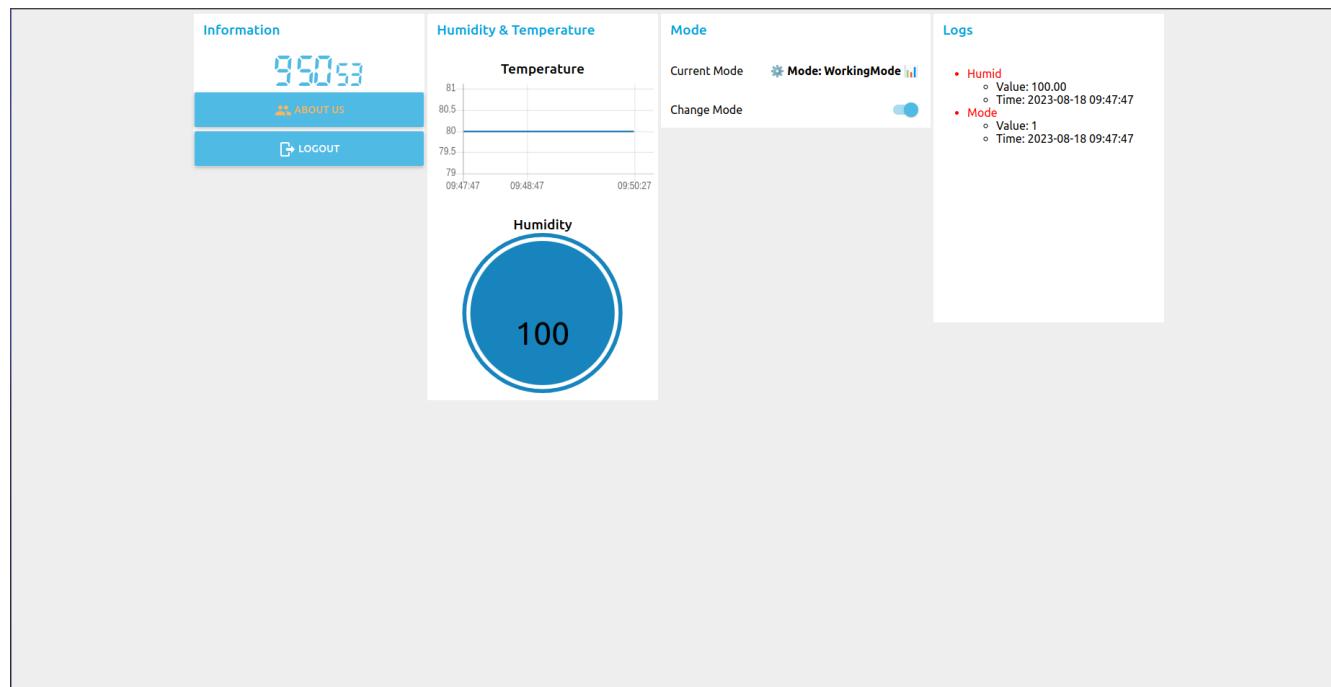


Hình 6: Lệnh chuyển đổi sang chế độ "hoạt động"



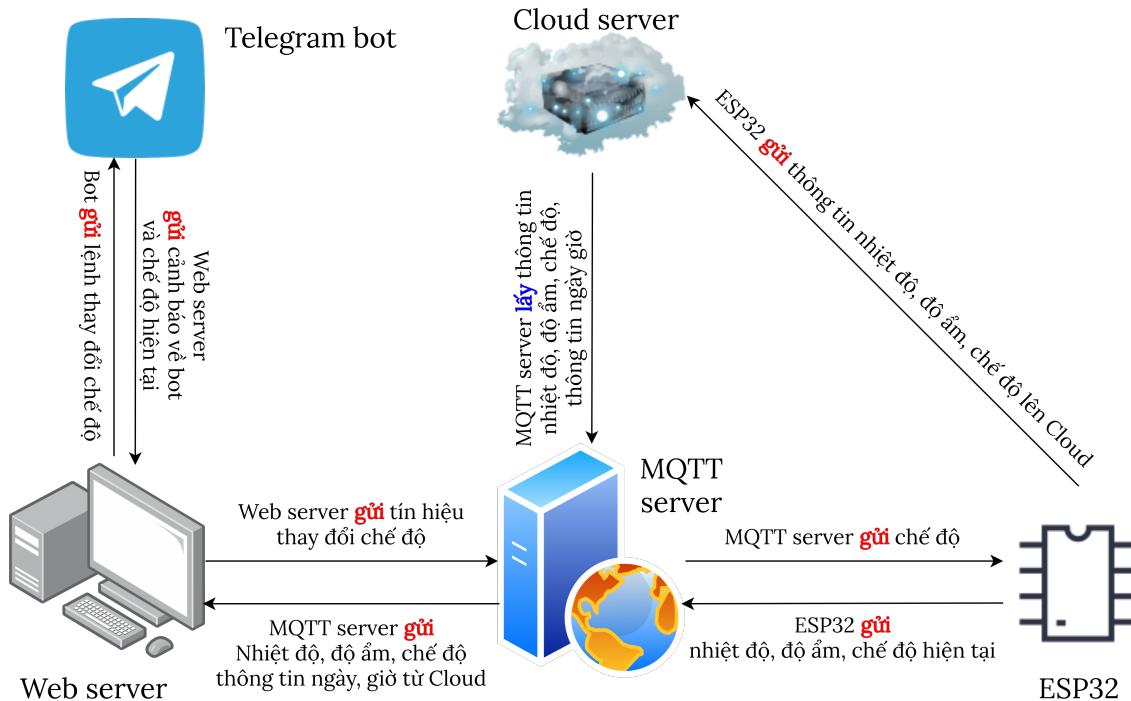
Hình 7: Lệnh chuyển đổi sang chế độ "bảo vệ"

- Thêm vào đó, người dùng có thể xem thông tin về nhiệt độ, độ ẩm, chuyển đổi chế độ trên **trang web** và cũng như lịch sử của các thông tin nhờ việc **lưu và lấy dữ liệu từ cloud**.



Hình 8: Hình ảnh tổng quát về trang web

### 2.3 Sơ đồ truyền và nhận dữ liệu của các thiết bị IOT và server



Hình 9: Sơ đồ truyền nhận dữ liệu

## 2.4 Thiết kế 3D sản phẩm của nhóm

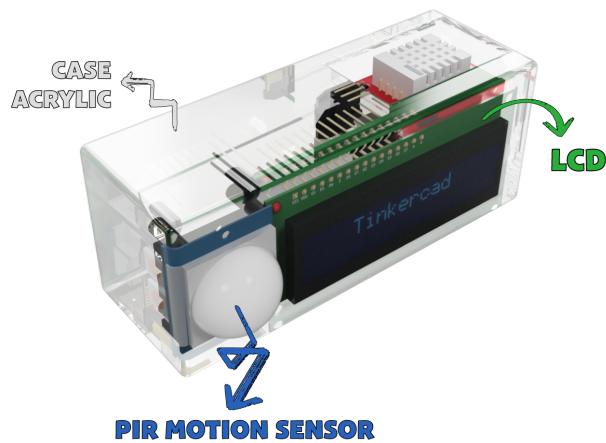
Dưới đây sẽ là chi tiết về hình ảnh của sản phẩm được thiết kế 3D ở nhiều góc độ.

- Đầu tiên phải nói tới vị trí đặt thiết bị ở trong phòng. Nhóm đã thiết kế bố trí sản phẩm ngay phía bên trái của cửa phòng để có thể nhận diện nhanh nhất khi có người vào phòng.



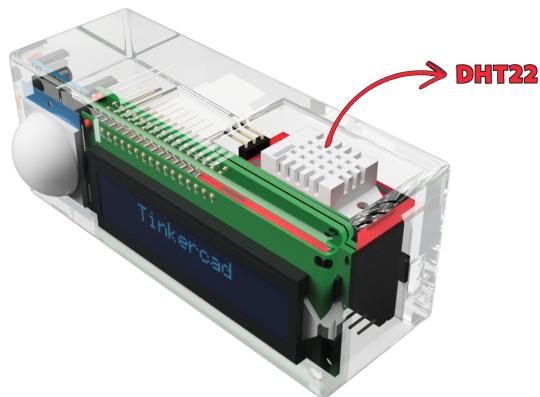
Hình 10: Vị trí đặt thiết bị ở trong phòng

- Đến với chi tiết của sản phẩm, trước hết các thiết bị của sản phẩm được liên kết với nhau và đặt trong hộp làm bằng hộp nhựa acrylic. Bên ngoài của hộp là 2 thiết bị **Cảm biến chuyển động** và **màn hình LCD** để người dùng dễ dàng nhìn thấy thông số trong phòng và giúp thiết bị cảm nhận người trong phòng tốt nhất.



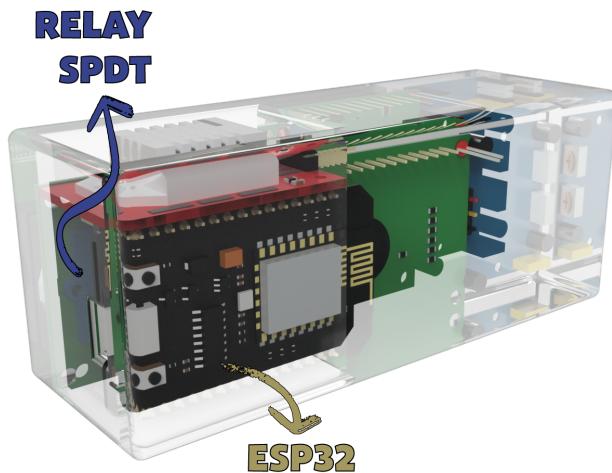
Hình 11: Mặt trước của sản phẩm

- Phía trên của sản phẩm chính là thiết bị **DHT22** dùng để đo nhiệt độ, độ ẩm. Thiết bị này sẽ kết nối với **LCD** và thông tin nhiệt độ, độ ẩm sẽ in ra màn hình.



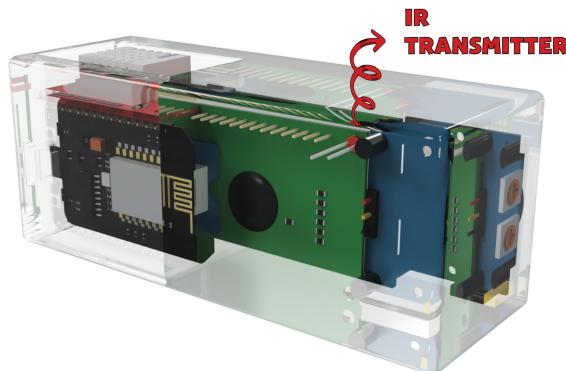
Hình 12: Mặt trên của sản phẩm

- Mặt sau của sản phẩm là thiết bị **ESP32** và vị trí giữa **ESP32** và LCD là thiết bị **Relay**.



Hình 13: Mặt sau của sản phẩm

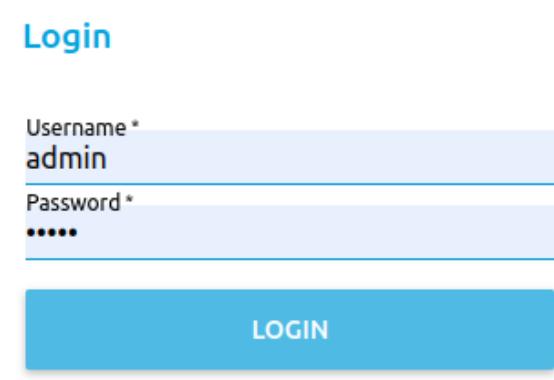
- Cuối cùng nhìn từ mặt sau chúng ta còn thấy được chân của thiết bị **IR-Transmitter**, phần đầu để phát tín hiệu hồng ngoại của thiết bị sẽ nằm phía mặt trước của sản phẩm.



Hình 14: Vị trí của IR-Transmitter

## 2.5 Giao diện và chức năng của web

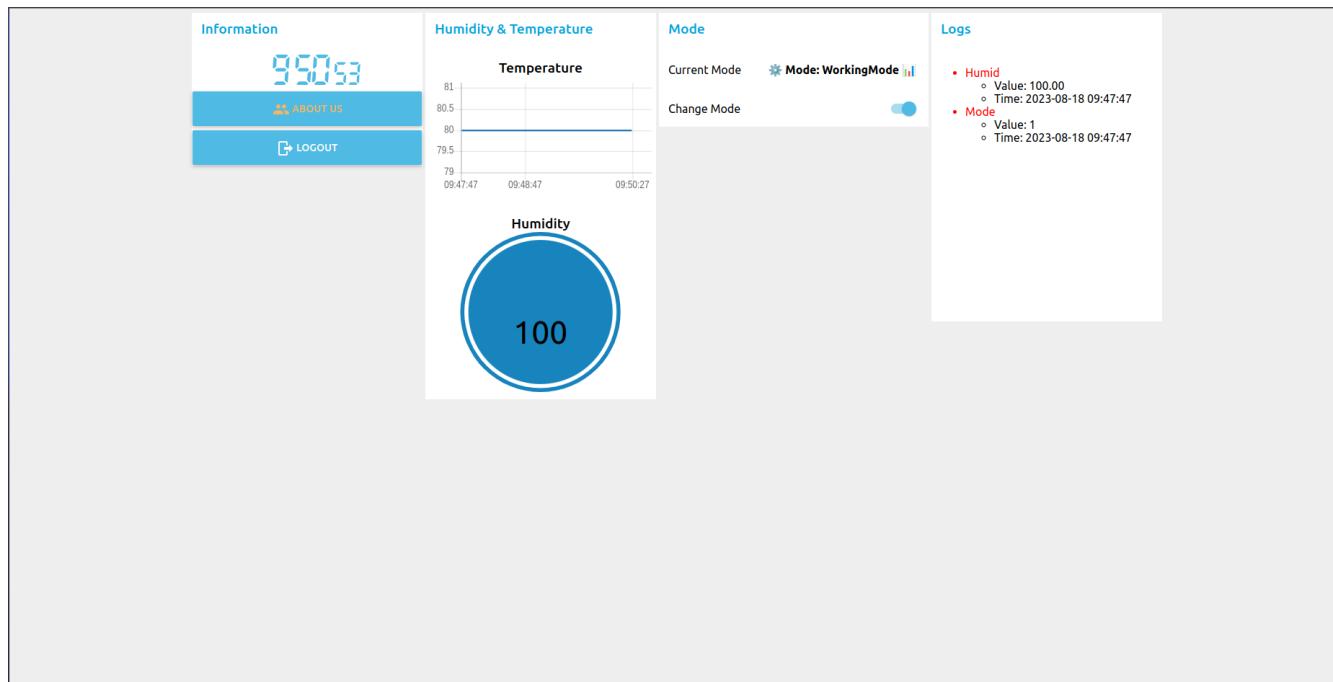
- Đầu tiên ngay khi truy cập vào trang web thì sẽ là màn hình đăng nhập, yêu cầu người dùng nhập **tên đăng nhập ("admin")** và **mật khẩu ("admin")**.



The screenshot shows a light gray background with a central white rectangular form. At the top left of the form, the word "Login" is written in blue. Below it, there are two input fields: one for "Username \*" containing "admin" and another for "Password \*" containing "\*\*\*\*\*". At the bottom of the form is a large, solid blue rectangular button with the word "LOGIN" in white capital letters.

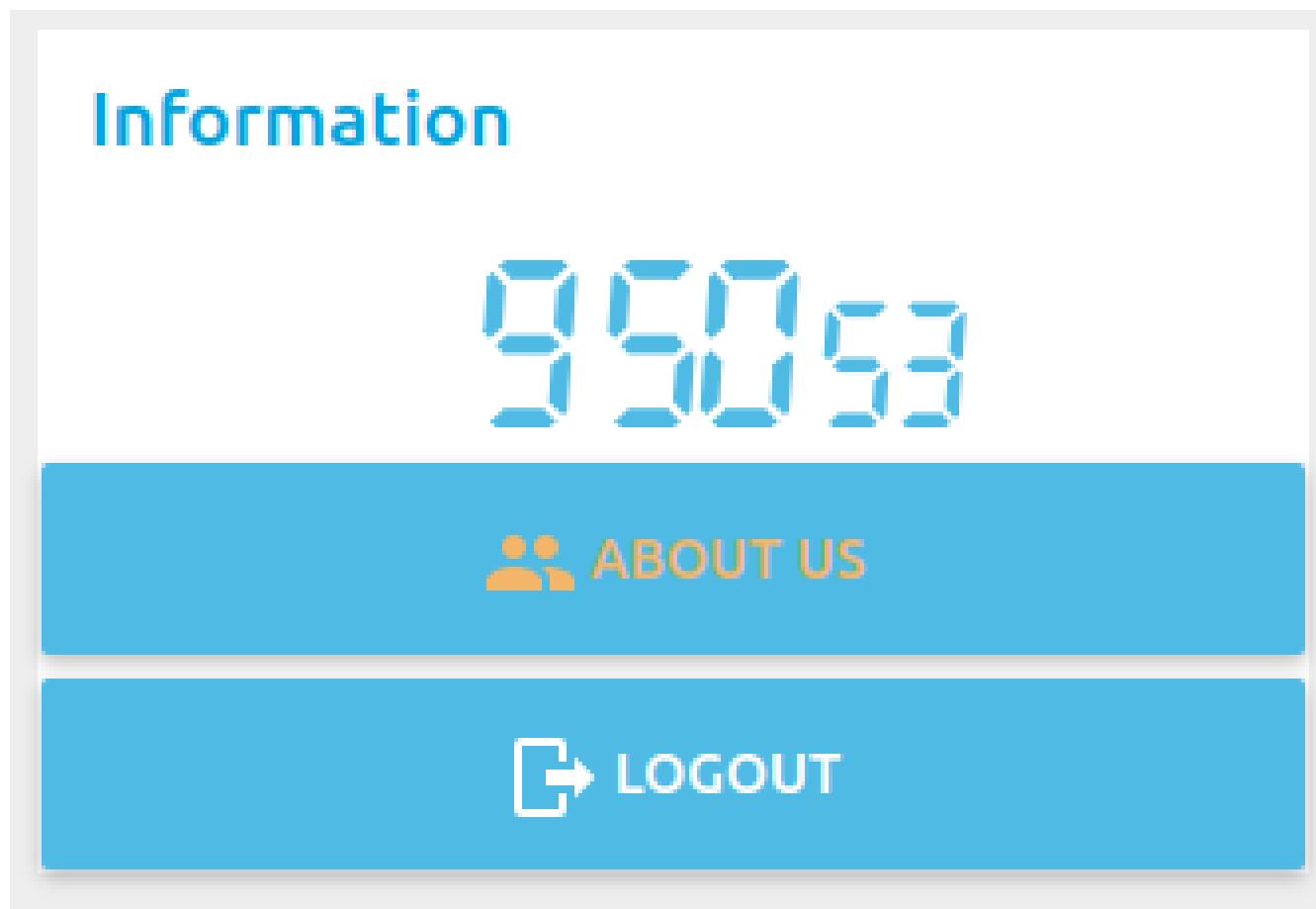
Hình 15: Giao diện trang đăng nhập của website

- Khi đăng nhập thành công, người dùng sẽ chuyển sang trang chính của website.



Hình 16: trang chính của website

- Nhóm đã thiết kế ở trang chính của trang web này mỗi chức năng sẽ được đặt ở 1 khung riêng biệt để người dùng dễ dàng nhận biết và sử dụng. Cụ thể như sau:
  - Đầu tiên ở bên trái ngoài cùng của trang web là khung để hiển thị **thời gian**, nút **chuyển tới thông tin của nhóm** và nút **đăng xuất khỏi trang web**.



Hình 17: Khung hiển thị thời gian, thông tin nhóm và nút đăng xuất

Ở trang **ABOUT US** sẽ hiển thị *Tên, mã số sinh viên, vai trò, và đường dẫn đến email* của mỗi thành viên.

**Group introduction**

**About Us**  
Đồ án cuối kì Vật lý dành cho CNTT (IOT)

**NHÓM 10 - LỚP 21CLC02**

**Triệu Nhật Minh**  
MSSV: 21127112  
Vai trò: Nhóm trưởng  
tnminh21@clc.fitus.edu.vn

[Contact](#)

**Bùi Đỗ Duy Quân**  
MSSV: 21127141  
Vai trò: Thành viên  
bddquan21@clc.fitus.edu.vn

[Contact](#)

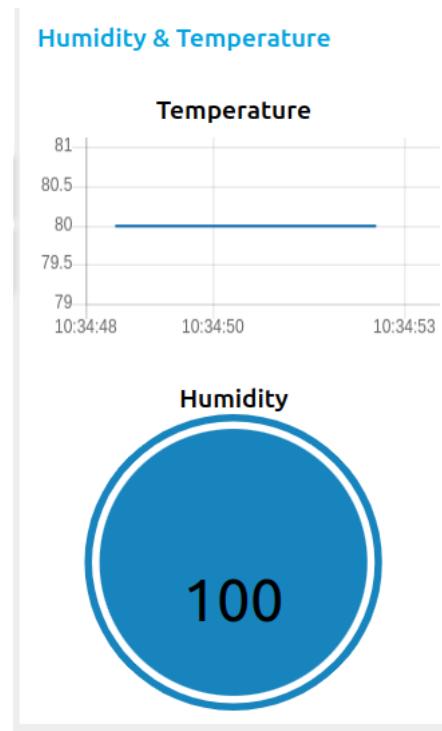
**Hoàng Đức Việt**  
MSSV: 21127203  
Vai trò: Thành viên  
hdviet21@clc.fitus.edu.vn

[Contact](#)

[BACK TO HOME](#)

Hình 18: Thông tin của nhóm

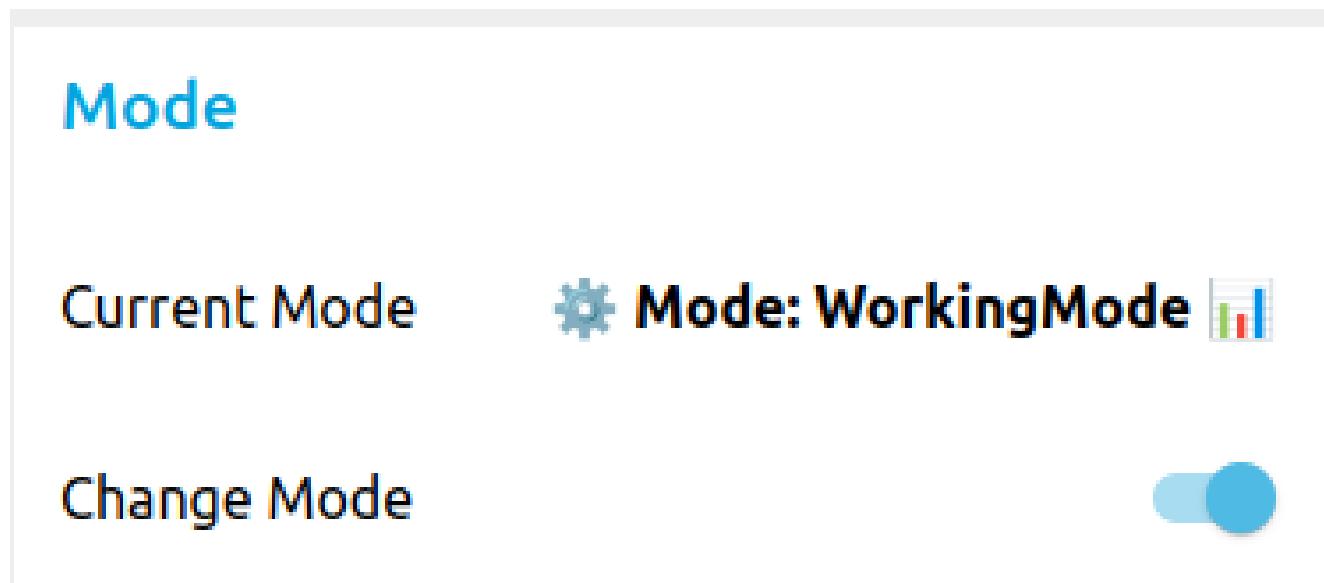
- Khung tiếp theo sẽ hiển thị biểu đồ đường về **nhiệt độ** và biểu đồ tròn về **độ ẩm** ở thời điểm hiện tại.



Hình 19: Biểu đồ nhiệt độ và độ ẩm

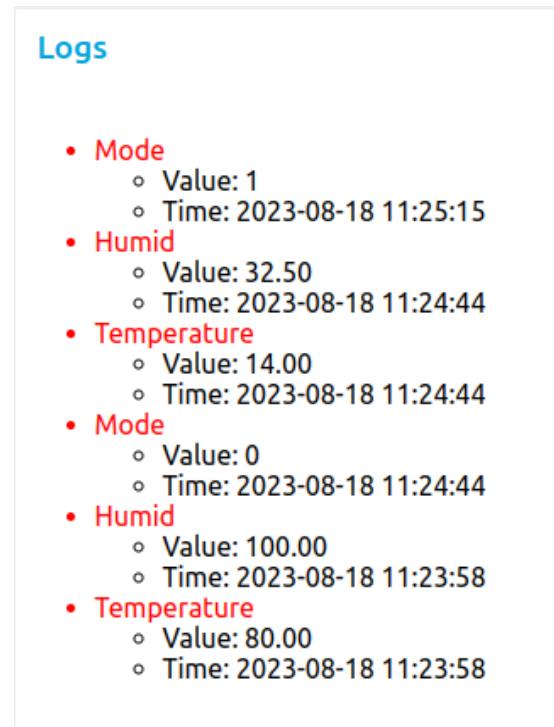
- Kế tiếp chính là khung **hiển thị chế độ hiện tại** và **nút thay đổi chế độ**. Với **nút thay đổi chế độ**, người dùng có thể thay đổi chế độ "hoạt

động" hay "bảo vệ" của thiết bị ở trong phòng.



Hình 20: Thông tin và chức năng chuyển chế độ

- Cuối cùng là khung **logs** để hiển thị lịch sử của những lần thay đổi chế độ và các thông tin về nhiệt độ, độ ẩm trong phòng.

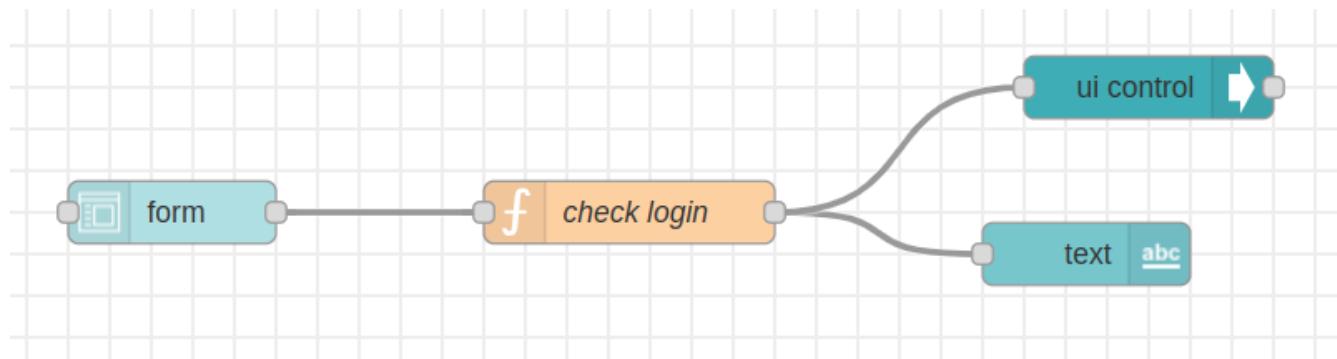


Hình 21: Lịch sử các thông tin

## 2.6 Xây dựng flow trong NodeRed

Giao diện và chức năng của web được xây dựng bằng NodeRed qua việc kết hợp các node chức năng để tạo thành flow cho trang web.

- Đầu tiên chính là phần node cho **đăng nhập vào trang web**. Node này sẽ có chức năng kiểm tra tài khoản và mật khẩu của người dùng, nếu đúng thì sẽ cho phép người dùng đăng nhập vào trang web, nếu sai thì sẽ hiện thông báo lỗi.



Hình 22: Các node về đăng nhập tài khoản

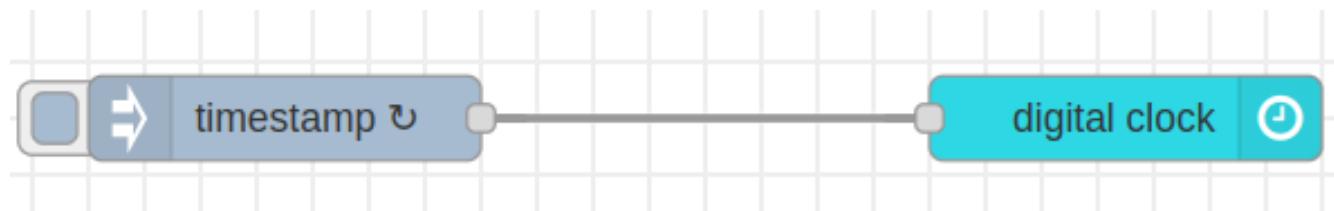
Sau khi người dùng điền tên đăng nhập và mật khẩu vào **form** thì hàm **check login** sẽ kiểm tra thông tin đăng nhập có hợp lệ hay không. Nhóm đã thống nhất rằng tên đăng nhập và mật khẩu để vào web đều là "**admin**". Đăng nhập thành công thì sẽ chuyển sang màn hình chính của web, và ngược lại thì sẽ đưa ra dòng thông báo đăng nhập thất bại.

```

var uname = msg.payload.username, pword = msg.payload.password
var validLogin = (uname == "admin" && pword == "admin")
if (validLogin == true)
    msg.payload = {"tab": "Home"}
else
{
    msg.payload = {"tab": "Login"}
    msg.announce = "Wrong username or password"
}
return msg;
  
```

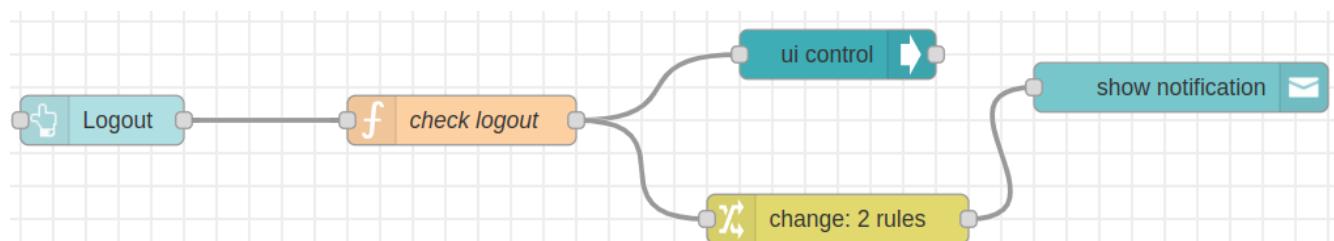
Hình 23: Hàm kiểm tra tài khoản và mật khẩu

2. Tiếp đó là các node cho giao diện về **đồng hồ và chức năng đăng xuất**. Để có thể lấy thông tin giờ/phút/giây hiện tại thì sẽ sử dụng node **digital clock** và 1 node **inject** để liên tục lấy thông tin thời gian liên tục sau 1 giây và hiển thị lên màn hình trang web.



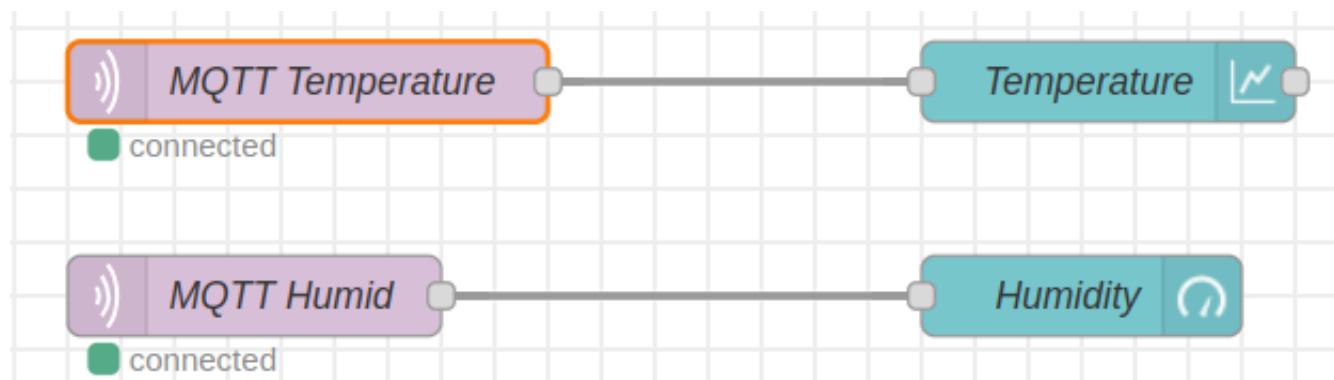
Hình 24: Node về đồng hồ

Nút **Log out** trên màn hình sẽ hỗ trợ chức năng thoát khỏi trang chính của web và quay trở lại trang đăng nhập trước đó.

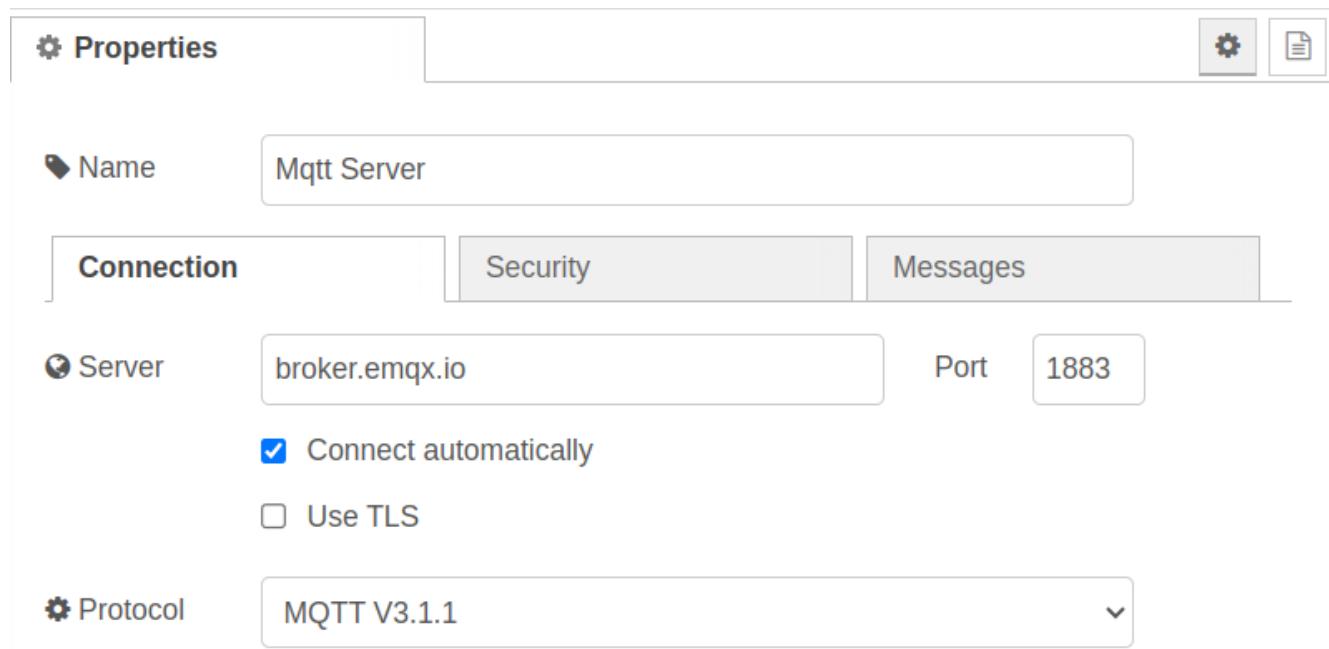


Hình 25: Node về đăng xuất

3. Kế đến đó chính là các node hỗ trợ chức năng chính của trang web. Đầu tiên để hiển thị thông tin nhiệt độ, độ ẩm trong phòng thì nhóm đã sử dụng node **MQTT in**-là node sử dụng giao thức **MQTT**. Node này được cài đặt sẽ kết nối với server **broker.emqx.io** thông qua giao thức MQTT và lấy thông tin được đưa lên về nhiệt độ, độ ẩm từ thiết bị **ESP32** và hiển thị lên trang web.

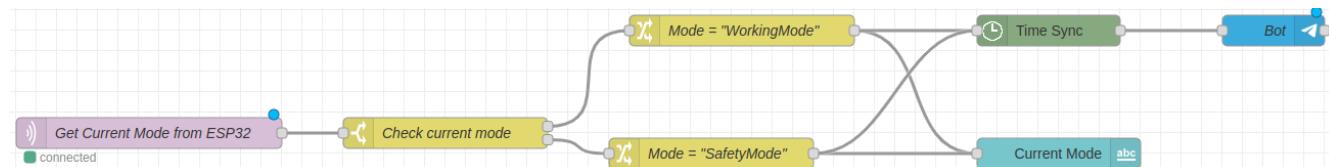


Hình 26: Node lấy thông tin nhiệt độ, độ ẩm

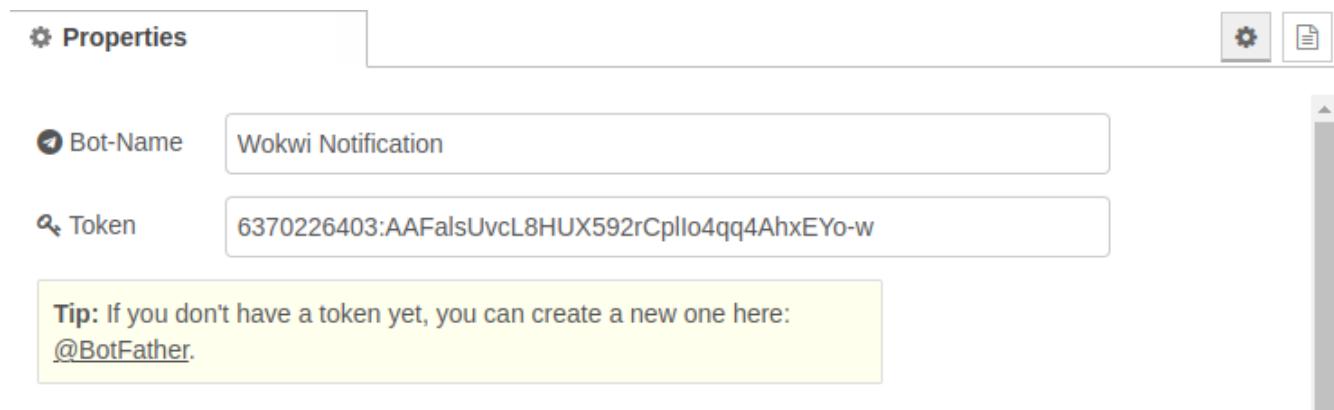


Hình 27: Thiết lập cho node MQTT in

4. Phần hiển thị thông tin đang ở chế độ nào cũng sẽ được thực hiện bằng cách lấy thông tin chế độ từ server đã được gửi từ **ESP32**. Hình ảnh giao diện về chế độ sẽ được cập nhật trên màn hình web theo đúng với thông tin chế độ của thiết bị. Đồng thời sản phẩm còn cho phép gửi thông tin chế độ và thời gian đã chuyển sang chế độ đó vào khung chat bot của telegram. Nhóm đã sử dụng node **sender** của **telegram bot** giúp truyền thông tin về chat box với bot của người dùng. Node này sẽ được khởi tạo bởi một mã token được cấp bởi **telegram bot father** và thông tin sẽ được gửi vào khung chat box của bot có mã token đó.



Hình 28: Node gửi thông tin chế độ



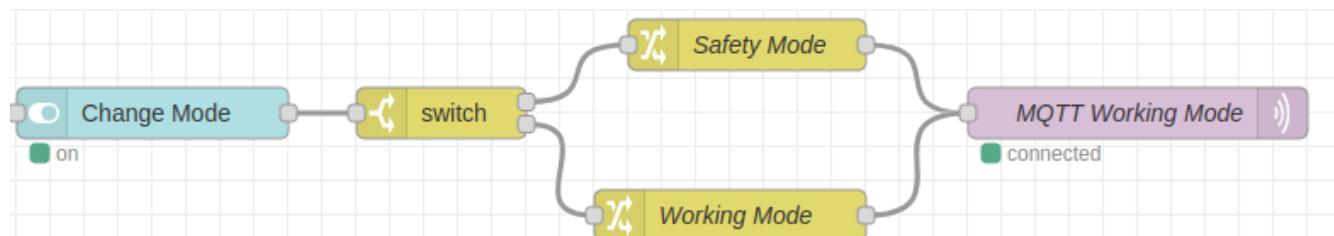
Hình 29: Thiết lập cho node sender

5. 1 chức năng cũng được cài đặt khá tương tự với chức năng trên đó là **gửi tin nhắn cảnh báo về người dùng**. Điểm tương đồng là cũng sẽ sử dụng node **MQTT in** để lấy dữ liệu khi có người vào phòng và cũng sẽ tạo tin nhắn và gửi qua chat box của **bot telegram**.



Hình 30: Node gửi tin nhắn cảnh báo

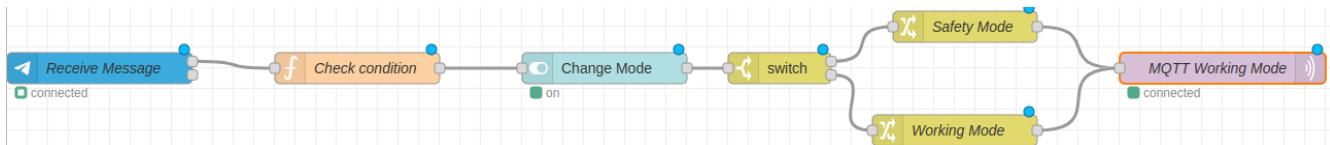
6. Chức năng tiếp theo được đề cập đến chính là **Chuyển đổi chế độ**. Nhóm đã thiết kế để cho người dùng có thể điều chỉnh giữa 2 chế độ thông qua 2 cách đó chính là **trên web và qua bot telegram**. Nút **bấm chuyển đổi** được cài đặt để khi chuyển sang chế độ nào sẽ gửi thông tin về lên server thông qua **MQTT out** và thông tin đó sẽ chuyển về **ESP32** để điều khiển các thiết bị trong phòng.



Hình 31: Node chuyển đổi chế độ

Việc gửi lệnh chuyển đổi thông qua **bot telegram** thì sẽ sử dụng node **telegram receiver** để nhận lệnh từ khung chat box của bot. Thông tin lệnh được

gửi tới sẽ được kiểm tra xem có phải là loại **message** hay không và nội dung phải là 1 trong 2 câu lệnh **safetymode** và **workingmode** thì mới được lưu lại để **nút chuyển đổi** cũng nhận thông tin và gửi lên server.



Hình 32: Node nhận lệnh chuyển đổi chế độ

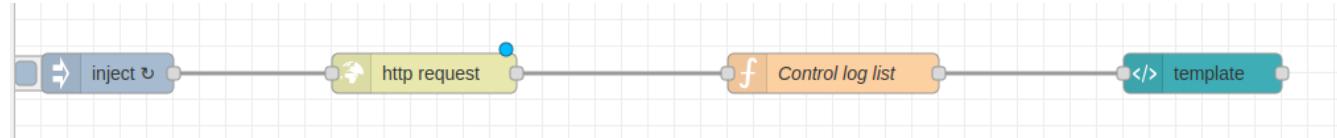
```

1 var content = msg.payload.content;
2 var id = msg.payload.chatId;
3 var type = msg.payload.type;
4
5 if (type == "message")
6 {
7     if(content == '/safetymode')
8         msg.payload = false;
9
10    if(content == '/workingmode')
11        msg.payload = true;
12 } else{
13     msg.payload = "not handle";
14 }
15 return msg;
  
```

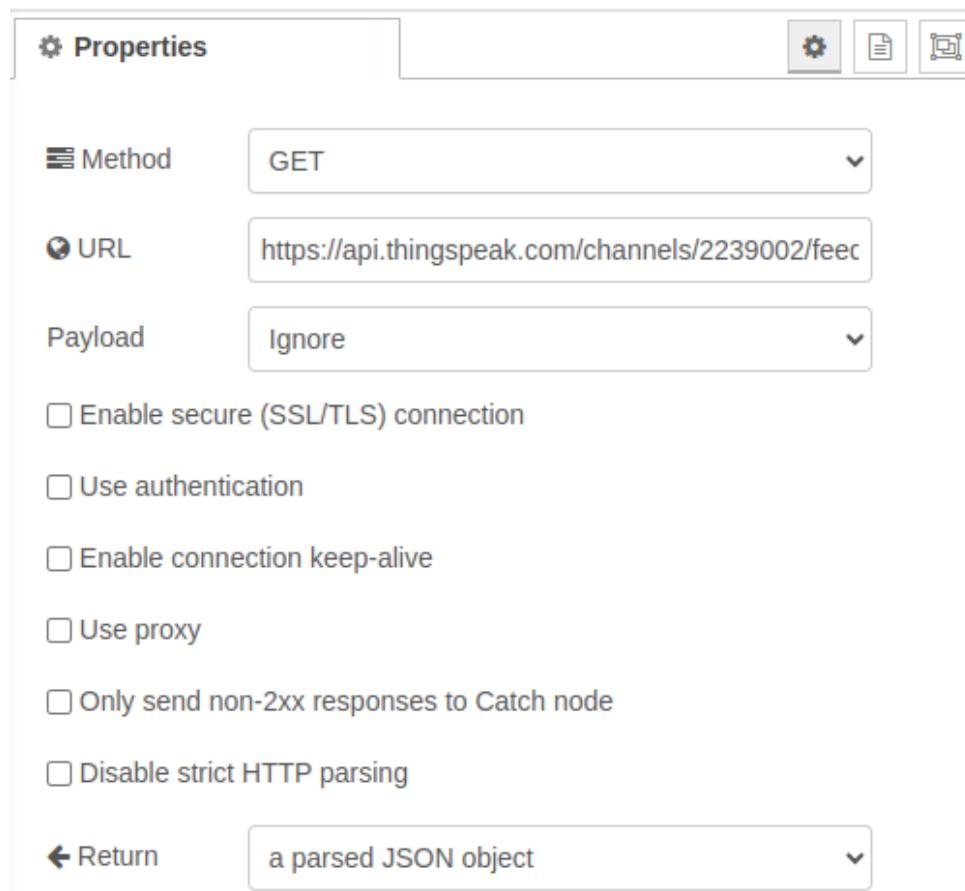
Hình 33: Hàm kiểm tra lệnh chuyển đổi chế độ

7. Cuối cùng là chức năng lấy dữ liệu từ trên cloud **thingspeak** về. Node mà nhóm đã sử dụng chính là **https request** để thực hiện yêu cầu tới đường dẫn **API key** của **thingspeak** để lấy thông số của gần nhất của 3 thông tin **nhiệt độ, độ ẩm và chế độ và thời điểm** của các thông số đó. Nút **inject** sẽ có nhiệm vụ khởi động **request** liên tục sau 1 giây và khi phía cloud có sự thay đổi thông số của bất kì thông tin nào thì thông số đó sẽ được cập nhật lên trang web thông qua hàm **control log list**. Hàm này sẽ có nhiệm vụ thực hiện thông số mới nhất và khi liên tục nhận dữ liệu từ cloud, hàm sẽ tiến hành kiểm tra xem thông số mới nhận được có thay đổi hay không, nếu thay đổi

thì sẽ đẩy lên hàng đầu của **log list** và in ra màn hình web, nếu không thay đổi thì sẽ không cập nhật để tránh tình trạng đưa lên các thông số chưa thay đổi. Chú ý rằng **API key** được lấy sẽ thêm tham số truy vấn **timezone** để chuyển múi giờ về **GMT+7** (chi tiết như ở đây).



Hình 34: Node lấy dữ liệu từ cloud



Hình 35: Thiết lập cho node https request

với giá trị **URL**: [https://api.thingspeak.com/channels/2239002/feeds.json?api\\_key=20EMGP6E0ZMPG8N1&results=1&timezone=Asia%2FJakarta](https://api.thingspeak.com/channels/2239002/feeds.json?api_key=20EMGP6E0ZMPG8N1&results=1&timezone=Asia%2FJakarta)

```

1  var logList = flow.get("logList")
2  var topic1 = "", payload1 = ""
3  var topic2 = "", payload2 = ""
4  var topic3 = "", payload3 = ""
5
6  var time = msg.payload.feeds[0].created_at
7  time = time.replace("T", " ")
8  time = time.replace("+07:00", "")
9
10 if (msg.payload.feeds[0].field1 != flow.get("prevMode"))
11 {
12     topic1 = "Mode"
13     payload1 = msg.payload.feeds[0].field1
14     flow.set("prevMode", payload1)
15 }
16
17 if (msg.payload.feeds[0].field2 != flow.get("prevTemperature"))
18 {
19     topic2 = "Temperature"
20     payload2 = msg.payload.feeds[0].field2
21     flow.set("prevTemperature", payload2)
22 }
23
24 if (msg.payload.feeds[0].field3 != flow.get("prevHumid"))
25 {
26     topic3 = "Humid"
27     payload3 = msg.payload.feeds[0].field3
28     flow.set("prevHumid", payload3)
29 }
30
31 if (topic1 != "")
32 {
33     msg = {}
34     msg.topic = topic1
35     msg.payload = payload1
36     msg.time = time
37     logList.unshift(msg)
38 }
39
40 if (topic2 != "")
41 {
42     msg = {}
43     msg.topic = topic2
44     msg.payload = payload2
45     msg.time = time
46     logList.unshift(msg)
47 }
48
49 if (topic3 != "")
50 {
51     msg = {}
52     msg.topic = topic3
53     msg.payload = payload3
54     msg.time = time
55     logList.unshift(msg)
56 }
57
58 if (logList.length > 10)
59 {
60     logList.shift();
61     logList.length = 10;
62 }
63
64 flow.set("logList", logList)
65 msg = {}
66 msg.payload = logList
67 return msg;
68

```

Listing 1: function of control log list

Ngoài ra để giúp in các thông tin được đồng nhất thì nhóm đã thiết kế hàm để kiểm soát thông số khởi tạo. Nếu trên cloud đang có thông số sẵn thì sẽ lấy thông số mới nhất của từng thông tin, và nếu chưa thì sẽ khởi tạo cho các thông số của thông tin giá trị cực nhỏ để khi bắt đầu chạy thiết bị thì sẽ lấy thông số hiện tại mà sản phẩm thu nhận được.

```
1  if (!msg.payload.feeds[0])
2  {
3      flow.set("prevMode", -999)
4      flow.set("prevTemperature", -999)
5      flow.set("prevHumid", -999)
6  }
7  else
8  {
9      flow.set("prevMode", msg.payload.feeds[0].field1)
10     flow.set("prevTemperature", msg.payload.feeds[0].field2)
11     flow.set("prevHumid", msg.payload.feeds[0].field3)
12 }
13
14 flow.set("logList", [])
15 return msg;
```

Hình 36: Hàm kiểm soát thông số khởi tạo

### 3 Phân công công việc

|         | Công việc   | Người thực hiện               |
|---------|---|-------------------------------|
| ESP32   | Cài đặt llop PIR  | Đức Việt                      |
|         | Cài đặt llop DHT  | Nhật Minh                     |
|         | Cài đặt llop IR   | Đức Việt                      |
|         | Cài đặt llop LCD  | Duy Quân                      |
|         | Cài đặt llop relay  | Đức Việt                      |
|         | Cài đặt hàm ReconnectMQTT   | Nhật Minh                     |
|         | Cài đặt hàm SyncToServer  | Nhật Minh                     |
|         | Cài đặt hàm SendRequestCloud  | Đức Việt                      |
|         | Cài đặt hàm callback  | Nhật Minh                     |
|         | Cài đặt hàm ReconnectWifi   | Duy Quân                      |
|         | Cài đặt hàm HandleWorkingMode   | Duy Quân                      |
|         | Cài đặt hàm HandleSafetyMode  | Đức Việt                      |
| WEB     | Dăng nhập, đăng xuất  | Nhật Minh                     |
|         | Gửi tin nhắn cảnh báo qua chat bot của telegram                       | Duy Quân                      |
|         | Tạo lệnh để bật tắt chế độ qua chat bot của telegram                  | Đức Việt                      |
|         | Thiết kế mô hình 3D của sản phẩm                                      | Nhật Minh                     |
|         | Chuyển dữ liệu lên cloud (nhiệt độ, độ ẩm, lịch sử chuyển đổi chế độ) | Đức Việt                      |
|         | About us  | Duy Quân                      |
|         | Cloud   | Đức Việt                      |
|         | Đồng bộ nút chuyển chế độ   | Đức Việt                      |
|         | Hiển thị nhiệt độ độ ẩm   | Nhật Minh                     |
| BÁO CÁO | Logs  | Nhật Minh                     |
|         | Mô tả chức năng thiết bị  | Nhật Minh, Duy Quân, Đức Việt |
|         | Mô tả chức năng web, giải thích flow node                             | Duy Quân                      |
|         | Vẽ sơ đồ truyền và nhận dữ liệu                                       | Đức Việt                      |
|         | Quay video demo   | Nhật Minh                     |
|         | Vẽ 3D sản phẩm  | Nhật Minh                     |

## 4 Tài liệu tham khảo

- Ý tưởng kết nối DHT11 và ESP32
- Ý tưởng thiết kế IR receiver
- Fitpaper
- HShop - shop linh kiện điện tử & Robot
- Tham khảo giá 'DHT11'
- Tham khảo giá 'Cảm biến siêu âm'
- Tham khảo giá 'mắt đọc hồng ngoại'
- Tham khảo giá 'LCD'
- Tham khảo giá 'Relay'
- Tham khảo giá 'điều khiển hồng ngoại'
- Tham khảo giá 'Board ESP32'
- Tham khảo giá 'Arduino uno r3' (kèm cáp USB 30cm)
- Tham khảo giá 'Dây cắm'
- Tham khảo giá 'nguồn 5V - 1A cho Arduino'
- Tham khảo giá 'hộp mica'