

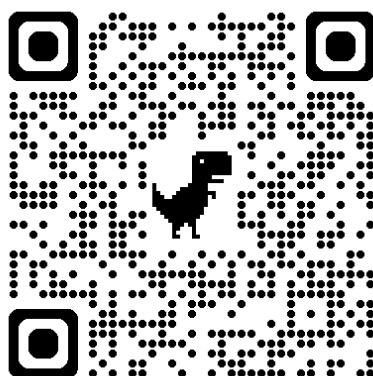
**TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP**

**KHOA ĐIỆN TỬ**

**Bộ môn: Công nghệ thông tin**



**BÀI TẬP LỚN**  
**MÔN HỌC LẬP TRÌNH PYTHON**



**Sinh viên: Nguyễn Duy Cao**

**Mssv: K205480106005**

**Lớp: K56KMT**

**Giáo viên hướng dẫn: Đỗ Duy Cốp**

**Thái Nguyên – 2024**

TRƯỜNG ĐHKTCN

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM

KHOA ĐIỆN TỬ

Độc lập - Tự do - Hạnh phúc

## PHIẾU GIAO ĐỀ TÀI BÀI TẬP LỚN MÔN LẬP TRÌNH PYTHON

Họ và tên: Nguyễn Duy Cao

Mssv: K205480106005

Lớp: K56KMT

Ngành: Kỹ thuật máy tính

Giáo viên hướng dẫn: Đỗ Duy Cốp

Ngày giao đề tài: 14/05/2024

Ngày hoàn thành: 26/05/2024

**1. Tên đề tài: Hệ thống cảm biến nhiệt độ và độ ẩm phòng**

**2. Nội dung:**

- Giới thiệu về ngôn ngữ Python
- Tổng quan đề tài
- Các bước tiến hành

**3. Báo cáo, chương trình:**

- Quyển báo cáo
- Demo bài làm trên máy tính cá nhân

**GIÁO VIÊN HƯỚNG DẪN**

(ký và ghi rõ họ tên)

[illegible]

## GIÁO VIÊN HƯỚNG DẪN

(Ký ghi rõ họ tên)

# MỤC LỤC

MỤC LỤC.....	4
LỜI MỞ ĐẦU .....	5
CHƯƠNG 1:GIỚI THIỆU VỀ NGÔN NGỮ PYTHON.....	6
CHƯƠNG 2 : TỔNG QUAN VỀ ĐỀ TÀI .....	8
2.1.Giới thiệu.....	8
2.2. Lý do lựa chọn đề tài.....	8
2.3.Mục tiêu nghiên cứu.....	8
CHƯƠNG 3 : CÁC BƯỚC TIẾN HÀNH .....	9
3.1. Kết nối mạch Arduino với cảm biến (DHT11) .....	9
3.2.Tạo API .....	10
3.3.Thiết kế NodeRed và CSDL .....	11
3.3.1.Thiết kế CSDL(dùng SQLSERVER).....	11
3.3.2.Thiết kế NodeRed .....	12
3.4.Tạo giao diện tương tác.....	15
3.5.Chạy chương trình.....	17
KẾT LUẬN .....	20

## LỜI MỞ ĐẦU

Trong cuộc sống hiện đại ngày nay, việc duy trì một môi trường sống và làm việc thoải mái là điều vô cùng quan trọng. Hệ thống cảm biến nhiệt độ và độ ẩm phòng ra đời nhằm đáp ứng nhu cầu này, mang lại sự tiện nghi và an toàn cho người sử dụng.

Hệ thống cảm biến nhiệt độ và độ ẩm phòng sử dụng các công nghệ tiên tiến để giám sát và điều chỉnh các điều kiện môi trường trong không gian của bạn. Các cảm biến chính xác cao được lắp đặt để liên tục đo lường nhiệt độ và độ ẩm, giúp cung cấp dữ liệu theo thời gian thực. Thông qua giao diện trực quan và dễ sử dụng, người dùng có thể theo dõi và quản lý các thông số môi trường một cách hiệu quả.

Một trong những điểm nổi bật của hệ thống này là khả năng cảnh báo tự động khi nhiệt độ hoặc độ ẩm vượt quá ngưỡng an toàn. Điều này không chỉ giúp duy trì môi trường sống lý tưởng mà còn bảo vệ sức khỏe của người dùng và đảm bảo sự hoạt động ổn định của các thiết bị điện tử trong phòng.

Hệ thống cảm biến nhiệt độ và độ ẩm phòng không chỉ là một giải pháp công nghệ thông minh mà còn là người bạn đồng hành đáng tin cậy trong việc chăm sóc môi trường sống của bạn. Với tính năng theo dõi liên tục, cảnh báo kịp thời và giao diện thân thiện, hệ thống này mang lại sự an tâm và thoải mái, góp phần nâng cao chất lượng cuộc sống hàng ngày.

Em xin chân thành cảm ơn sự quan tâm giúp đỡ của thầy **Đỗ Duy Cốp** đã tận tình chỉ bảo trong quá trình em thực hiện đề tài này.

# CHƯƠNG 1: GIỚI THIỆU VỀ NGÔN NGỮ PYTHON

Python là một ngôn ngữ lập trình được sử dụng rộng rãi trong các ứng dụng web, phát triển phần mềm, khoa học dữ liệu và máy học (ML). Các nhà phát triển sử dụng Python vì nó hiệu quả, dễ học và có thể chạy trên nhiều nền tảng khác nhau. Phần mềm Python được tải xuống miễn phí, tích hợp tốt với tất cả các loại hệ thống và tăng tốc độ phát triển.



Những lợi ích của Python bao gồm:

- Các nhà phát triển có thể dễ dàng đọc và hiểu một chương trình Python vì ngôn ngữ này có cú pháp cơ bản giống tiếng Anh.
- Python giúp cải thiện năng suất làm việc của các nhà phát triển vì so với những ngôn ngữ khác, họ có thể sử dụng ít dòng mã hơn để viết một chương trình Python.
- Python có một thư viện tiêu chuẩn lớn, chứa nhiều dòng mã có thể tái sử dụng cho hầu hết mọi tác vụ. Nhờ đó, các nhà phát triển sẽ không cần phải viết mã từ đầu.
- Các nhà phát triển có thể dễ dàng sử dụng Python với các ngôn ngữ lập trình phổ biến khác như Java, C và C++.

- Cộng đồng Python tích cực hoạt động bao gồm hàng triệu nhà phát triển nhiệt tình hỗ trợ trên toàn thế giới. Nếu gặp phải vấn đề, bạn sẽ có thể nhận được sự hỗ trợ nhanh chóng từ cộng đồng.
- Trên Internet có rất nhiều tài nguyên hữu ích nếu bạn muốn học Python. Ví dụ: bạn có thể dễ dàng tìm thấy video, chỉ dẫn, tài liệu và hướng dẫn dành cho nhà phát triển.
- Python có thể được sử dụng trên nhiều hệ điều hành máy tính khác nhau, chẳng hạn như Windows, macOS, Linux và Unix.

Ngôn ngữ Python được sử dụng nhiều trong lĩnh vực phát triển ứng dụng:

- Phát triển web phía máy chủ
- Tự động hóa bằng các tập lệnh Python
- Khoa học dữ liệu và máy học
- Phát triển phần mềm
- Tự động hóa kiểm thử phần mềm

## **CHƯƠNG 2 : TỔNG QUAN VỀ ĐỀ TÀI**

### **2.1.Giới thiệu**

Trong thời đại công nghệ 4.0, việc ứng dụng các giải pháp công nghệ vào cuộc sống hàng ngày ngày càng trở nên phổ biến và cần thiết. Một trong những ứng dụng quan trọng là hệ thống cảm biến nhiệt độ và độ ẩm, giúp giám sát và duy trì các điều kiện môi trường lý tưởng cho không gian sống và làm việc. Đề tài "Hệ thống cảm biến nhiệt độ và độ ẩm" nhằm nghiên cứu, thiết kế và triển khai một hệ thống giám sát thông minh, mang lại nhiều lợi ích thiết thực cho người sử dụng.

### **2.2. Lý do lựa chọn đề tài**

Nhiệt độ và độ ẩm là hai yếu tố quan trọng ảnh hưởng trực tiếp đến sức khỏe con người, sự thoải mái trong sinh hoạt và hiệu suất làm việc. Việc giám sát và điều chỉnh hai yếu tố này giúp tạo ra môi trường sống và làm việc lý tưởng, bảo vệ sức khỏe, và kéo dài tuổi thọ của các thiết bị điện tử. Đề tài này được lựa chọn nhằm tìm ra các giải pháp hiệu quả để theo dõi và kiểm soát nhiệt độ, độ ẩm trong các không gian khác nhau như nhà ở, văn phòng, nhà kho, phòng máy chủ, và nhiều ứng dụng khác.

### **2.3.Mục tiêu nghiên cứu**

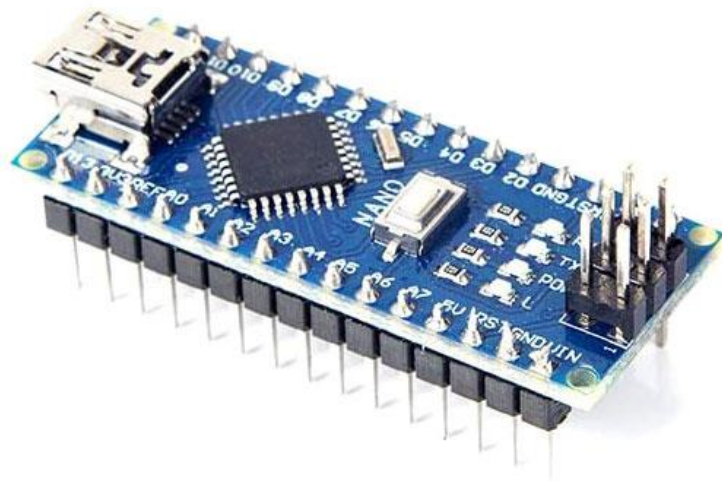
- Thiết kế và phát triển một hệ thống cảm biến nhiệt độ và độ ẩm có khả năng giám sát liên tục và chính xác.
- Cung cấp giao diện người dùng thân thiện để theo dõi dữ liệu thời gian thực và lịch sử.
- Tích hợp các tính năng cảnh báo tự động khi nhiệt độ hoặc độ ẩm vượt quá ngưỡng an toàn.
- Đánh giá hiệu quả của hệ thống trong việc duy trì môi trường ổn định và an toàn.



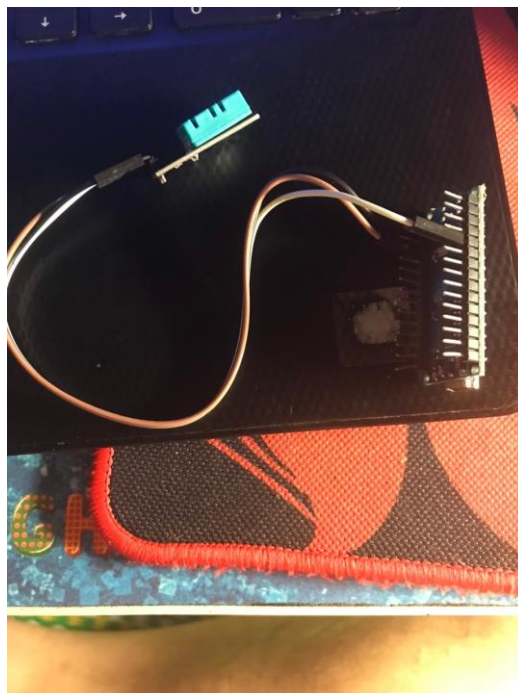
## CHƯƠNG 3 : CÁC BƯỚC TIỀN HÀNH

### 3.1. Kết nối mạch Arduino với cảm biến (DHT11)

Arduino Uno là một board mạch vi điều khiển được phát triển bởi Arduino.cc, một nền tảng điện tử mã nguồn mở chủ yếu dựa trên vi điều khiển AVR Atmega328P. Với Arduino chúng ta có thể xây dựng các ứng dụng điện tử tương tác với nhau thông qua phần mềm và phần cứng hỗ trợ.



Sau khi tìm hiểu về mạch thì ta kết nối với cảm biến nhiệt độ và độ ẩm DHT11 đây là 1 con cảm biến giá rẻ dễ sử dụng đồng thời có sự hỗ trợ của thư viện đi kèm



Sau khi kết nối mạch với cảm biến thành công ta cấu hình để nhận dữ liệu của cảm biến đo được và đưa nó về dạng JSON nhằm dễ dàng cho việc lấy dữ liệu sau này :

```
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  dht.begin();
}

void loop() {
  delay(2000); // Chờ 2 giây trước khi đọc dữ liệu mới

  // Đọc độ ẩm và nhiệt độ từ cảm biến
  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();

  // Kiểm tra nếu đọc dữ liệu thành công
  if (isnan(humidity) || isnan(temperature)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }

  // In dữ liệu ra cổng Serial
  Serial.print(F("Humidity: "));
  Serial.print(humidity);
  Serial.print(F("% Temperature: "));
  Serial.print(temperature);
  Serial.println(F("°C"));
```

Kết quả khi demo trên Arduino IDE :

```
Humidity: 94.00% Temperature: 30.80~C
{"temperature":30.80,"humidity":94.00,"timestamp":"2023"}
Humidity: 94.00% Temperature: 30.80°C
{"temperature":30.80,"humidity":94.00,"timestamp":"4081"}
```

### 3.2.Tạo API

Sau khi đo nhiệt độ và độ ẩm ta tạo Api nhằm lấy những dữ liệu này và cấu hình thêm biến thời gian cho nó :

```
def read_from_arduino():
    global latest_data
    ser = serial.Serial('COM3', 9600, timeout=1) # Adjust 'COM3' as necessary
    while True:
        if ser.in_waiting > 0:
            line = ser.readline().decode('utf-8').strip()
            try:
                data = json.loads(line)
                temperature = data['temperature']
                humidity = data['humidity']
                timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S')

                # Update timestamp in the JSON data
                data['timestamp'] = timestamp

                # Update the global variable with the latest data
                latest_data = data
            except json.JSONDecodeError:
                print("Received invalid JSON data")
```

Code ở trên ta cấu hình cho nó để lấy dữ liệu từ Arduino ở cổng serial mà nó trả về dưới dạng JSON ta đã cấu hình từ trước rồi thêm biến thời gian hiện tại đo được vào. Sau khi ta lấy được dữ liệu và cấu hình thêm biến thời gian ta tiến tới tạo Api bằng Python + FastApi nhằm gửi dữ liệu lên NodeRed để nó lưu vào CSDL:

```
@app.get("/api/data", response_model=SensorData)
def get_data():
    if not latest_data:
        raise HTTPException(status_code=404, detail="No data available")
    return latest_data
```

### 3.3.Thiết kế NodeRed và CSDL

#### 3.3.1.Thiết kế CSDL(dùng SQLSEVER)

Tạo 1 database và trong database đó tạo 1 bảng nhằm lưu trữ dữ liệu mà ta đã cấu hình từ trước:

	Column Name	Data Type	Allow Nulls
▶	temperature	nvarchar(50)	<input checked="" type="checkbox"/>
	humidity	nvarchar(50)	<input checked="" type="checkbox"/>
	timestamp	datetime	<input checked="" type="checkbox"/>
🔑	id	int	<input type="checkbox"/>
			<input type="checkbox"/>

Sau khi tạo bảng thành công ta tạo 1 Stored Procedures (thủ tục lưu trữ) nhằm làm chức năng thêm dữ liệu vào trong bảng để có thể sử dụng trong NodeRed một cách hiệu quả :

```
USE [Api_Cambien]
GO
/***** Object:  StoredProcedure [dbo].[InsertApi]    Scr
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

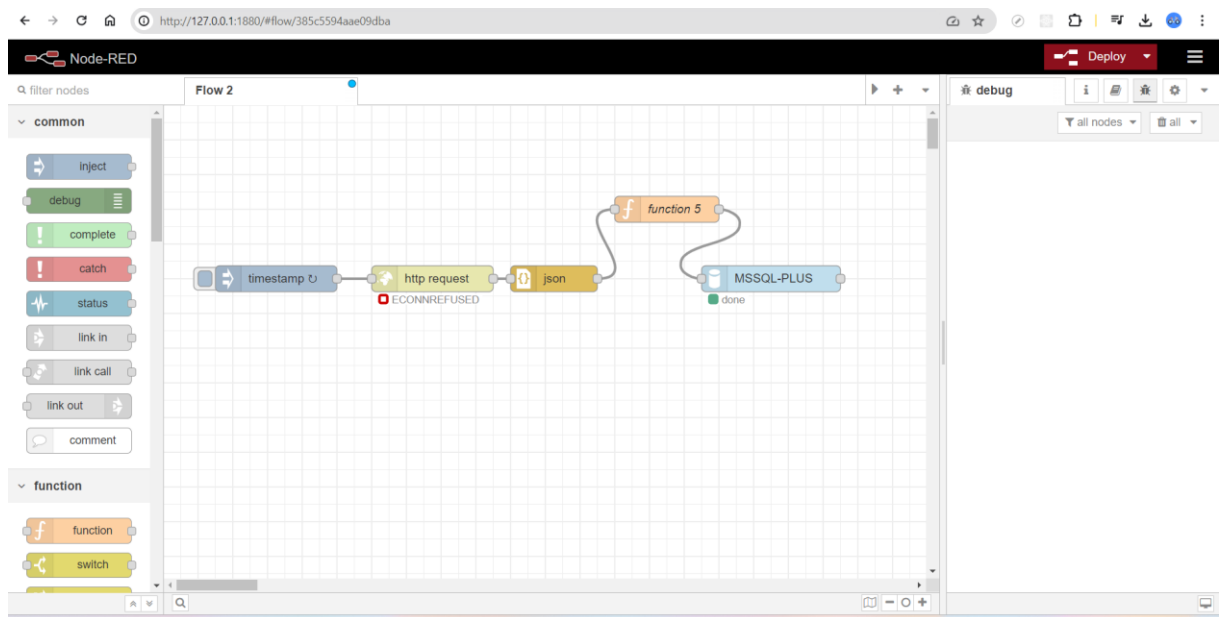
ALTER PROCEDURE [dbo].[InsertApi]
    @temperature NVARCHAR(50),
    @humidity NVARCHAR(50),
    @timestamp DATETIME
AS
BEGIN
    SET NOCOUNT ON;

    INSERT INTO Cambien(temperature, humidity, timestamp)
    VALUES (@temperature, @humidity, @timestamp);
END
```

### 3.3.2. Thiết kế NodeRed

Node-RED là một công cụ mã nguồn mở và trực quan được sử dụng để xây dựng các luồng làm việc (workflows) và ứng dụng Internet of Things (IoT)

Sau khi hiểu sơ qua về NodeRed ta bắt đầu vào quá trình thiết kế :

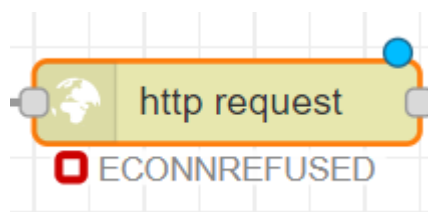


Như hình ở trên đây là giao diện chính của NodeRed đồng thời là nơi ta sẽ thiết kế và cấu hình cho nó :

Node Timestamp : đây là 1 node quản lý thời gian trong luồng dữ liệu , ta có thể cấu hình thời gian lấy dữ liệu cho nó



Node http request: Nó cho phép bạn tương tác với các API web, tải xuống dữ liệu từ các trang web, hoặc gửi dữ liệu đến các dịch vụ web. Đây là node chính lấy dữ liệu từ Api mà ta đã tạo trong Python



Edit http request node

Delete
Cancel
Done

⚙️ Properties

Method
GET

URL
http://127.0.0.1:8000/api/data

Payload
Ignore

☐ Enable secure (SSL/TLS) connection

☐ Use authentication

☐ Enable connection keep-alive

☐ Use proxy

☐ Only send non-2xx responses to Catch node

☐ Disable strict HTTP parsing

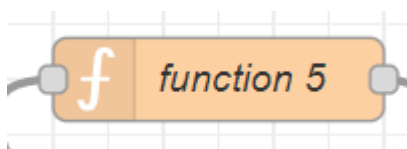
← Return
a parsed JSON object

Tip: If the JSON parse fails the fetched string is returned as-is.

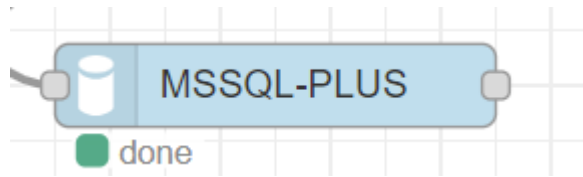
Node Json : Chuyển đổi giữa Json và string



Node function: được sử dụng để viết các đoạn mã JavaScript tùy chỉnh để xử lý các tin nhắn (messages).



Node MSSQL-PLUS : đây là node dùng để liên kết với SQL SEVER



Với node này ta cần cấu hình cho nó thì mới có thể sử dụng được đồng thời để sử dụng thủ tục lưu trữ mà ta đã tạo cũng cần cấu hình đúng mới làm được :

The image shows a 'Edit MSSQL node' configuration window. At the top, there are 'Delete', 'Cancel', and 'Done' buttons. Below is a 'Properties' section with a gear icon and three sub-sections: 'Connection' (set to 'Sql Sever'), 'Name' (set to 'Name'), and 'Query mode' (set to 'Execute Procedure'). Below these is a 'Query' section with a dropdown set to 'Editor' and a text area containing '1 InsertApi'. At the bottom is a 'Parameters' section with a dropdown set to 'Editor' and a list of three input parameters: 'Name tempera' (Type Float, Value msg. payload.temperature), 'Name humidit' (Type Float, Value msg. payload.humidity), and 'Name timesta' (Type nvarc, Value msg. payload.timestamp). Each parameter has a menu icon on the left and a delete 'x' button on the right.

### 3.4.Tạo giao diện tương tác

Ta sử dụng ngôn ngữ PHP để viết chương trình lấy dữ liệu từ SQL SEVER và vẽ 1 biểu đồ hiển thị lên trên web :

```

<?php
1 reference
function connect_to_database() {
    $serverName = "DUYCAO";
    $connectionOptions = array(
        "Database" => "Api_Cambien",
        "Uid" => "sa",
        "PWD" => "123456",
        "CharacterSet" => "UTF-8"
    );
    $conn = sqlsrv_connect($serverName, $connectionOptions);
    if ($conn === false) {
        die(print_r(sqlsrv_errors(), true));
    }
    return $conn;
}

1 reference
function execute_stored_procedure($proc_name, $params = array()) {
    $conn = connect_to_database();
    $query = "{CALL $proc_name(?)}";
    $stmt = sqlsrv_prepare($conn, $query, $params);
    if (!$stmt) {
        die(print_r(sqlsrv_errors(), true));
    }
    if (!sqlsrv_execute($stmt)) {
        die(print_r(sqlsrv_errors(), true));
    }
}

```

```

<?php
include 'database.php';

header('Content-Type: application/json');

if ($_SERVER['REQUEST_METHOD'] == 'GET') {
    $date = isset($_GET['date']) ? $_GET['date'] : get_current_date();
    $result = execute_stored_procedure('SelectDate', array($date));
    $response = array();
    foreach ($result as $row) {
        $response[] = array(
            'temperature' => $row['temperature'],
            'humidity' => $row['humidity'],
            'timestamp' => $row['timestamp']->format('Y-m-d H:i:s'),
            'id' => $row['id']
        );
    }
    echo json_encode($response);
} else {
    http_response_code(405);
    echo json_encode(array('error' => 'Method Not Allowed'));
}

```



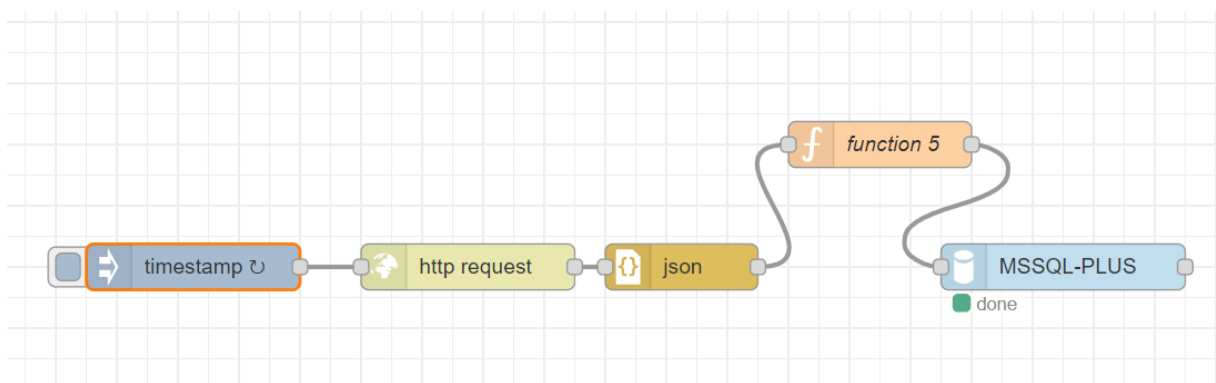


```
main.py x api.php index.php style.css
Api_cambien > main.py > Python > read_from_arduino
1 import serial
2 import json
3 from datetime import datetime, date
4 from fastapi import FastAPI, HTTPException, Query
5 from pydantic import BaseModel
6 from fastapi.middleware.cors import CORSMiddleware
7 import threading
8 import requests
9 import pyodbc
10 from typing import Optional
11
12 app = FastAPI()
13
14 app.add_middleware(
15     CORSMiddleware,
16     allow_origins=["*"],
17     allow_methods=["*"],
18     allow_headers=["*"],
19 )
20
21 if __name__ == '__main__':
22     uvicorn.run(app, host='127.0.0.1', port=8000)
```

PROBLEMS 2 DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR

File "C:\Program Files (x86)\Microsoft Visual Studio\Share  
next(self.gen)  
File "C:\Users\57\AppData\Roaming\Python\Python39\site-pack  
signal.raise\_signal(captured\_signal)  
KeyboardInterrupt  
PS C:\Users\57\Downloads\New folder\Api\_cambien> python -u "C:  
INFO: Started server process [23680]  
INFO: Waiting for application startup.  
INFO: Application startup complete.  
INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)  
Received invalid JSON data  
Received invalid JSON data

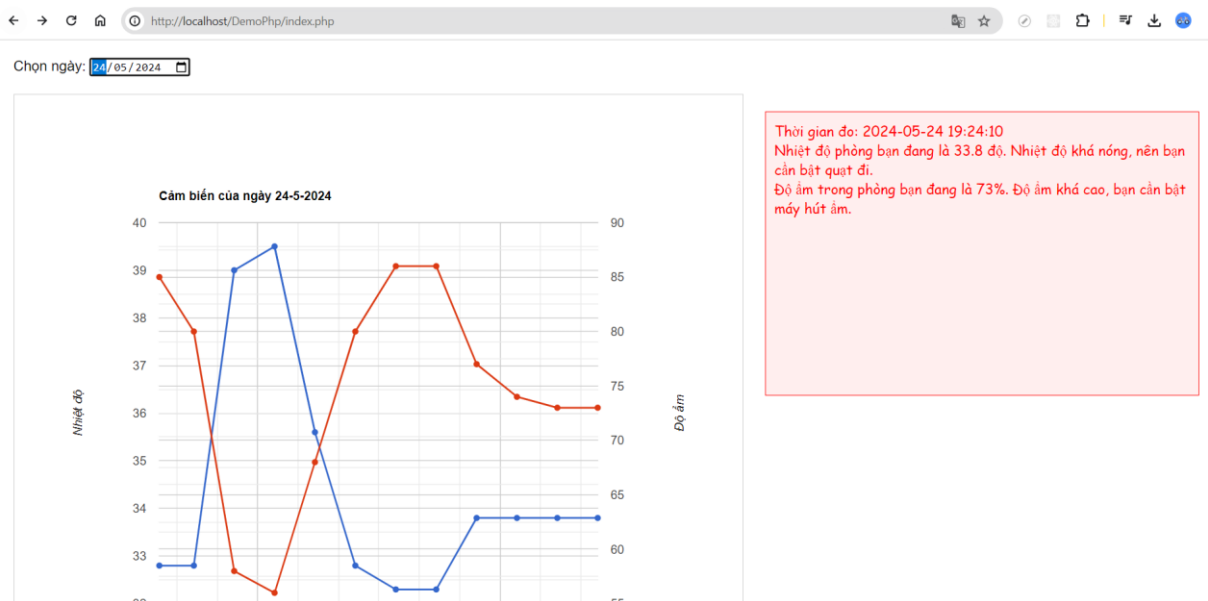
Sau khi chạy xong ta vào NodeRed cái này có thể tự cài đặt trên máy để lấy đường dẫn vào trang chính rồi nhấn flow của timestamp để bắt đầu :



Trong CSDL kiểm tra xem nó đã lưu dữ liệu chưa :

	temperature	humidity	timestamp	id
1	32.8	85	2024-05-24 17:35:37.000	455
2	32.8	80	2024-05-24 17:44:10.000	456
3	39	58	2024-05-24 17:54:09.000	457
4	39.5	56	2024-05-24 18:04:09.000	458
5	35.6	68	2024-05-24 18:14:09.000	459
6	32.8	80	2024-05-24 18:24:09.000	460
7	32.3	86	2024-05-24 18:34:11.000	461
8	32.3	86	2024-05-24 18:44:11.000	462
9	33.8	77	2024-05-24 18:54:11.000	463
10	33.8	74	2024-05-24 19:04:10.000	464
11	33.8	73	2024-05-24 19:14:10.000	465
12	33.8	73	2024-05-24 19:24:10.000	466

Sau khi kiểm tra nó đã lưu thành công ta chạy chương trình PHP ở đây tôi sử dụng XAMPP để chạy chương trình chính cho PHP và đây là trang web ta nhận được:



## KẾT LUẬN

Sau quá trình nghiên cứu từ nhiều nguồn khác nhau, em đã tích lũy được nhiều kiến thức mới về việc xây dựng API bằng Python và FastAPI, đồng thời học cách sử dụng Node-RED trong quá trình lấy và lưu dữ liệu vào cơ sở dữ liệu. Trong quá trình này, em đã hiểu rõ hơn về cách vận hành của một hệ thống cảm biến nhiệt độ và độ ẩm phòng, bao gồm cách thu thập, xử lý và hiển thị dữ liệu cảm biến theo thời gian thực.

Mặc dù đã đạt được những kết quả đáng kể, em nhận thấy rằng còn nhiều điều cần phải cải thiện và bổ sung. Một số tính năng mong muốn như việc tối ưu hóa hiệu suất hệ thống, cải thiện giao diện người dùng, và thêm các chức năng báo động khi nhiệt độ hoặc độ ẩm vượt ngưỡng nhất định vẫn chưa được triển khai hoàn chỉnh. Em cũng cần phải tìm hiểu thêm về các biện pháp bảo mật và quản lý quyền truy cập để đảm bảo hệ thống hoạt động an toàn và ổn định.

Trong tương lai, em dự định sẽ tiếp tục nghiên cứu và phát triển thêm các tính năng mới cho hệ thống, đồng thời tối ưu hóa các chức năng hiện tại để nâng cao hiệu quả và độ tin cậy của hệ thống cảm biến nhiệt độ và độ ẩm phòng.