

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: Vi Duy Quốc

Lớp: K56KMT.01

MSSV: k205480106034

Giáo viên hướng dẫn: Đỗ Duy Cốp

Thái Nguyên – 2024

BÀI TẬP LỚN

MÔN HỌC: LẬP TRÌNH PYTHON
BỘ MÔN CÔNG NGHỆ THÔNG TIN

Họ Tên: Vi Duy Quốc

MSSV:k205480106034

Lớp: k56KMT

Ngành: Kỹ thuật máy tính

Giáo viên hướng dẫn: Đỗ Duy Cốp

Ngày giao đề : 20-05-2024 Ngày hoàn thành : 26-5-2024

Tên đề tài : Quản lý nhiệt độ thông minh

Yêu cầu :

- Python FastAPI
- Node-red+SQL
- Tạo giao diện UI + Chart

GIÁO VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

Mục lục

I. Giới thiệu	4
1.1. Bối cảnh và Tầm Quan Trọng của Python	4
1.2. Mục Đích của Báo Cáo	4
1.3. Phạm Vi Nghiên Cứu	4
1.4. Phương Pháp Nghiên Cứu	4
II. Nội dung	5
2.1. Xây dựng dự án và các bước	5
2.2. Các bước thực hiện	5
Hình 1: Tạo API	6
Bước 2: Sử dụng Node-red để nhận dữ liệu từ API	7
Hình 2: API được tạo	7
Hình 3: Khởi tạo Node-red	8
Hình 4: Kéo thả các luồng làm việc Node	9
Hình 5: Http request gán URL của API đã tạo	9
Bước 3: Thêm dữ liệu vào SQL server	9
Hình 6: Cấu hình mssql để kết nối tới Sql server và Query thêm dữ liệu vào database	10
.....	Error! Bookmark not defined.
Hình 7: Dữ liệu đã được thêm	10
Bước 4: Lấy dữ liệu từ Sql server và đẩy lên web tạo UI	11
Hình 8: Tạo app.py bằng python để kết nối tới SQL	11
Hình 9: Tạo file index.html để hiển thị	11
Hình 10: Tạo file js để lấy dữ liệu từ SQL tạo Chart	12
Hình 11: Dữ liệu được hiển thị	12
III. Kết luận	13

I. Giới thiệu

1.1. Bối cảnh và Tầm Quan Trọng của Python

Trong kỷ nguyên số hóa hiện nay, lập trình đã trở thành một kỹ năng thiết yếu không chỉ đối với những người làm việc trong lĩnh vực công nghệ thông tin mà còn đối với nhiều ngành nghề khác. Trong số các ngôn ngữ lập trình, Python nổi lên như một ngôn ngữ phổ biến và mạnh mẽ nhờ vào cú pháp đơn giản, dễ học và tính linh hoạt cao. Python đã và đang được sử dụng rộng rãi trong nhiều lĩnh vực như khoa học dữ liệu, trí tuệ nhân tạo, phát triển web, tự động hóa, và nhiều ứng dụng khác.

1.2. Mục Đích của Báo Cáo

Báo cáo này được thực hiện với mục đích:

- Cung cấp một cái nhìn tổng quan về ngôn ngữ lập trình Python.
- Trình bày các kiến thức cơ bản và cấu trúc dữ liệu trong Python.
- Giới thiệu về cách định nghĩa và sử dụng hàm, module trong Python.
- Khám phá các khái niệm cơ bản của lập trình hướng đối tượng.
- Minh họa một số ứng dụng thực tế của Python trong các lĩnh vực khác nhau.

1.3. Phạm Vi Nghiên Cứu

Báo cáo tập trung vào những khía cạnh sau:

- Lịch sử phát triển và đặc điểm nổi bật của Python.
- Các kiểu dữ liệu cơ bản và cấu trúc dữ liệu trong Python.
- Cách sử dụng hàm và module trong Python.
- Các nguyên lý cơ bản của lập trình hướng đối tượng trong Python.
- Các ứng dụng thực tế của Python trong khoa học dữ liệu, trí tuệ nhân tạo, phát triển web, và tự động hóa.

1.4. Phương Pháp Nghiên Cứu

Để hoàn thành báo cáo này, em đã sử dụng các phương pháp nghiên cứu sau:

- Nghiên cứu tài liệu: Đọc và tham khảo từ các sách giáo trình, tài liệu hướng dẫn, và tài liệu trực tuyến liên quan đến Python.
- Thực hành: Áp dụng các kiến thức đã học vào các bài tập và dự án thực tế.
- Thảo luận: Trao đổi với giảng viên và các bạn cùng lớp để hiểu sâu hơn về các vấn đề phức tạp.

II. Nội dung

2. Ứng dụng thực tế của Python

2.1. Ý tưởng

- Dự Án: Hệ Thống Quản Lý Nhiệt Độ Thông Minh

Mô Tả: trong dự án này sẽ tạo ra một hệ thống theo dõi và quản lý nhiệt độ trong các phòng của một ngôi nhà (Các cảm biến nhiệt độ sẽ gửi dữ liệu đến Node-RED) sau đó dữ liệu này sẽ được lưu trữ vào SQL Server và có thể được truy xuất và hiển thị qua một API được xây dựng bằng Python FastAPI.

Các Thành Phần Chính Cảm Biến Nhiệt Độ (Giả Lập): Sử dụng một cảm biến giả lập để gửi dữ liệu nhiệt độ định kỳ đến Node-RED.

NodeRED: Nhận dữ liệu từ cảm biến. Xử lý và chuyển dữ liệu đến SQL Server=>Tạo các luồng xử lý và cảnh báo nếu nhiệt độ tăng hoặc giảm bất thường.

SQL Server: Lưu trữ dữ liệu nhiệt độ nhận được từ Node-RED (lưu trữ thông tin các phòng và các ngưỡng nhiệt độ an toàn).

Python FastAPI: Xây dựng API để truy xuất dữ liệu nhiệt độ từ SQL Server. Cung cấp Endpoint để quy định mức nhiệt độ cho phép của các phòng. Cung cấp Endpoint để lấy dữ liệu lịch sử nhiệt độ.

Giao Diện (Optional): giao diện web hiển thị dữ liệu nhiệt độ theo thời gian thực và cài đặt mức nhiệt độ.

- Các bước triển khai dự án quản lý nhiệt độ thông minh

Thiết Lập Node-RED Cài đặt Node-RED.

Tạo luồng để nhận dữ liệu từ cảm biến.

Sử dụng node mssql để gửi dữ liệu đến SQL Server.

Cấu hình SQL Sever tạo các bảng cần thiết : temperatureReadings, Thresholds,...

Thiết lập keeys nối từ Node-Red đến SQL Sever

Phát Triển FastAPI Cài đặt FastAPI và các thư viện (fastapi, pydantic, sqlalchemy).

Xây dựng các endpoint: GET /temperatures: Lấy dữ liệu nhiệt độ hiện tại của các phòng.

GET /temperatures/history: Lấy dữ liệu lịch sử nhiệt độ.

Thresholds: Cập nhật ngưỡng nhiệt độ cho phép.

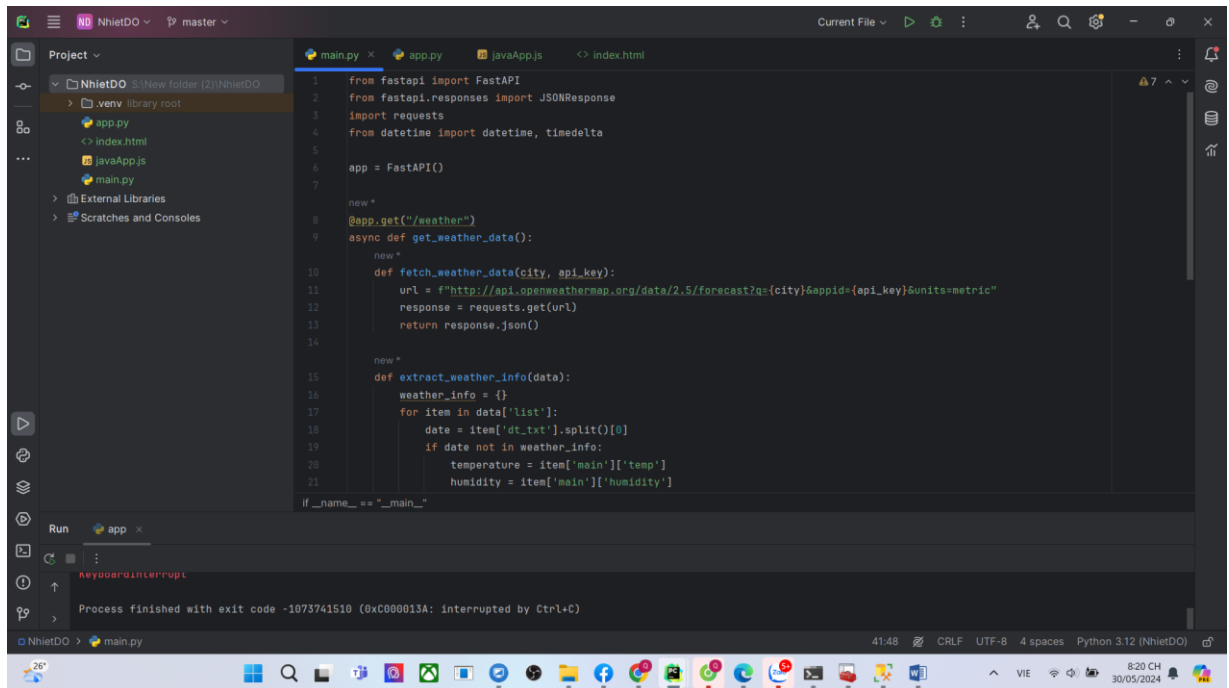
Kết nối FastAPI với SQL Server sử dụng thư viện **pyodbc** để kết nối với cơ sở dữ liệu và **SQLAlchemy** để quản lý ORM.

Xây Dựng Giao Diện Người Dùng Sử dụng HTML/CSS/JavaScript để xây dựng giao diện .

Hiển thị dữ liệu nhiệt độ theo thời gian thực (sử dụng thư viện như Chart.js). Tạo form để cập nhật ngưỡng nhiệt độ.

2.2. Các bước thực hiện

Bước 1: Tạo hàm giả lập cảm biến nhiệt độ bằng Python sử dụng fastAPI tạo API



```
1 from fastapi import FastAPI
2 from fastapi.responses import JSONResponse
3 import requests
4 from datetime import datetime, timedelta
5
6 app = FastAPI()
7
8 new *
9 @app.get("/weather")
10 async def get_weather_data():
11     new *
12     def fetch_weather_data(city, api_key):
13         url = f"http://api.openweathermap.org/data/2.5/forecast?q={city}&appid={api_key}&units=metric"
14         response = requests.get(url)
15         return response.json()
16
17     new *
18     def extract_weather_info(data):
19         weather_info = {}
20         for item in data['list']:
21             date = item['dt_txt'].split()[0]
22             if date not in weather_info:
23                 temperature = item['main']['temp']
24                 humidity = item['main']['humidity']
25
26 if __name__ == "__main__":
```

Hình 1: Tạo API

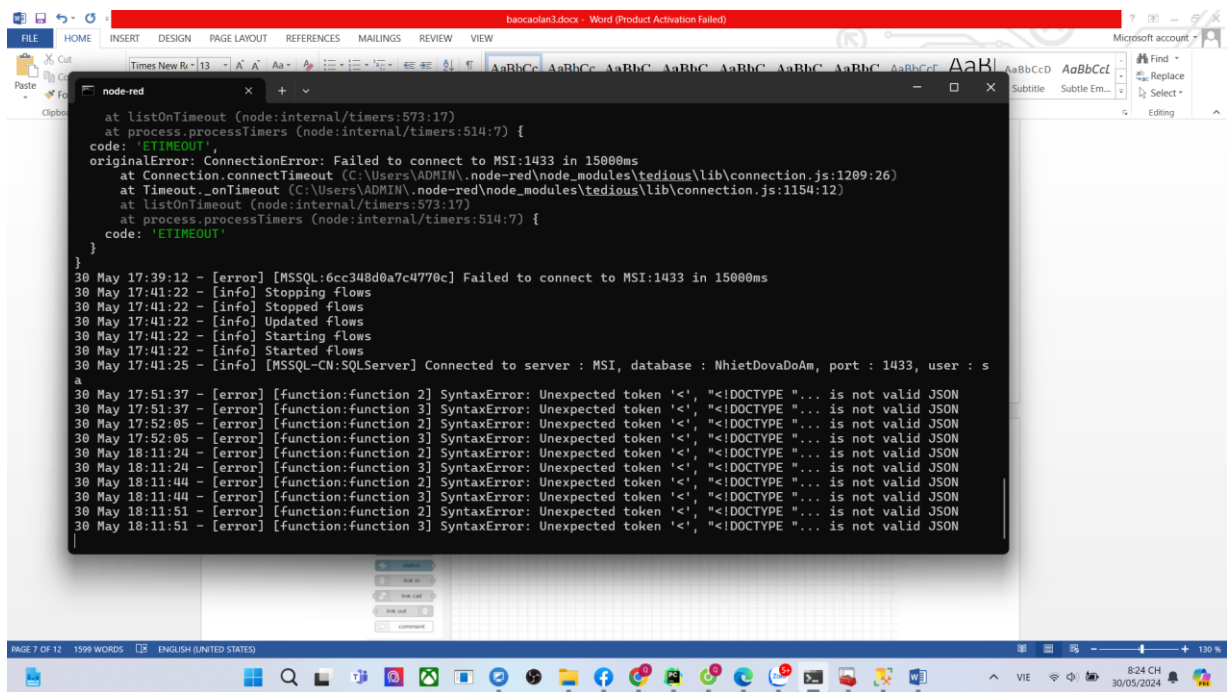
Bước 2: Sử dụng Node-red để nhận dữ liệu từ API



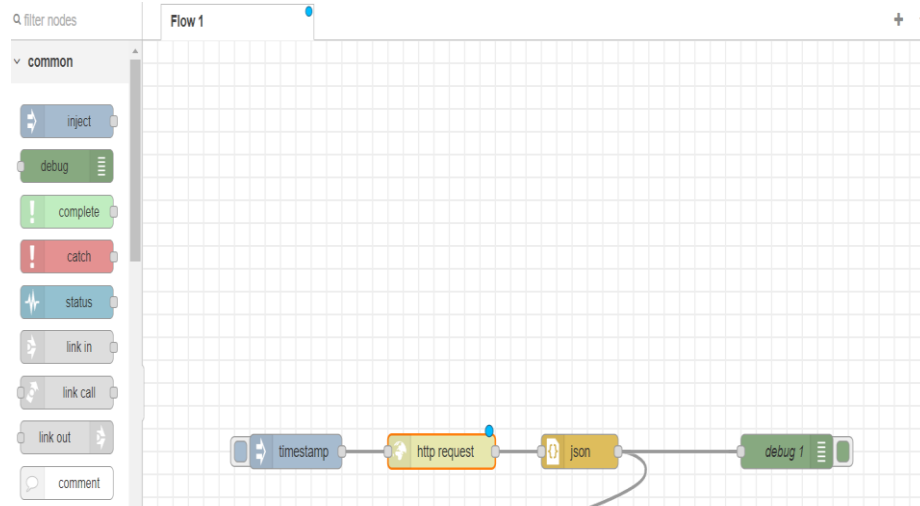
The screenshot shows a web browser window with the address bar displaying `127.0.0.1:1880/weather`. The main content area shows a JSON array of weather data. The data is structured as follows:

```
[
  {
    "Date": "2024-05-30",
    "Temperature": 24.88,
    "Humidity": 87
  },
  {
    "Date": "2024-05-31",
    "Temperature": 25.4,
    "Humidity": 88
  },
  {
    "Date": "2024-06-01",
    "Temperature": 26.18,
    "Humidity": 78
  },
  {
    "Date": "2024-06-02",
    "Temperature": 27.28,
    "Humidity": 83
  },
  {
    "Date": "2024-06-03",
    "Temperature": 26.71,
    "Humidity": 92
  },
  {
    "Date": "2024-06-04",
    "Temperature": 24.83,
    "Humidity": 95
  }
]
```

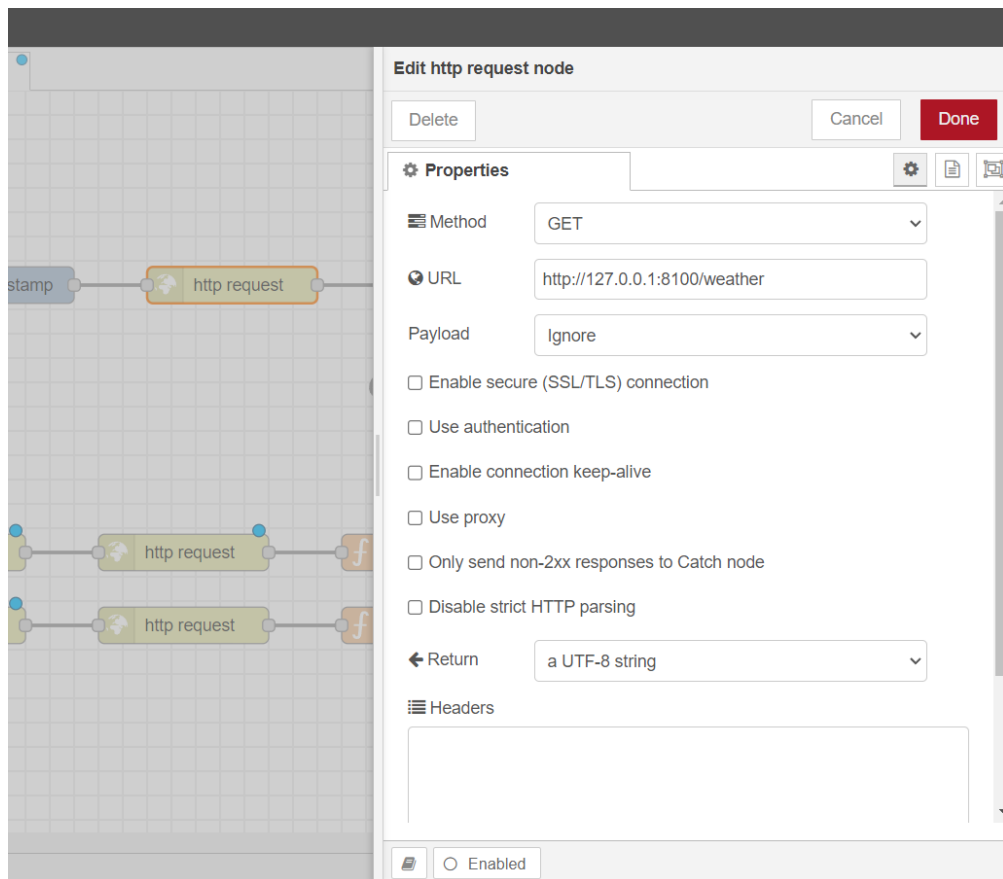
Hình 2: API được tạo



Hình 3: Khởi tạo Node-red

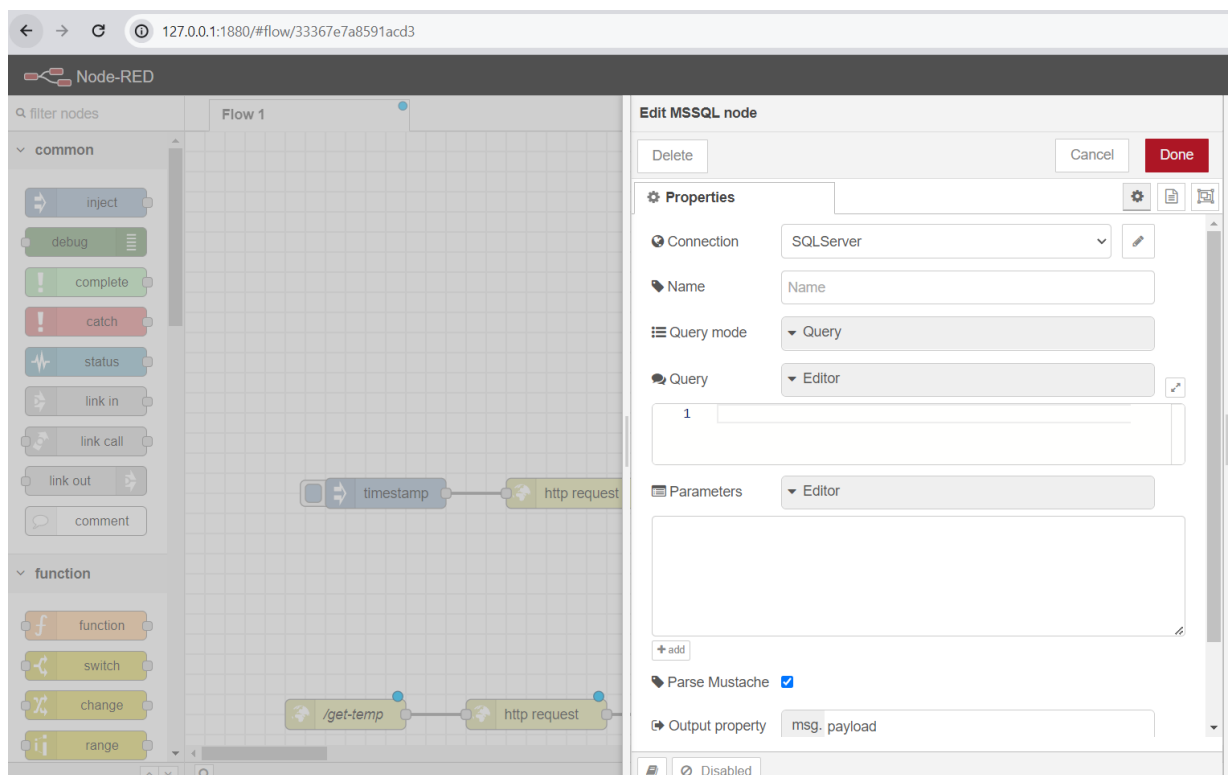


Hình 4: Kéo thả các luồng làm việc Node

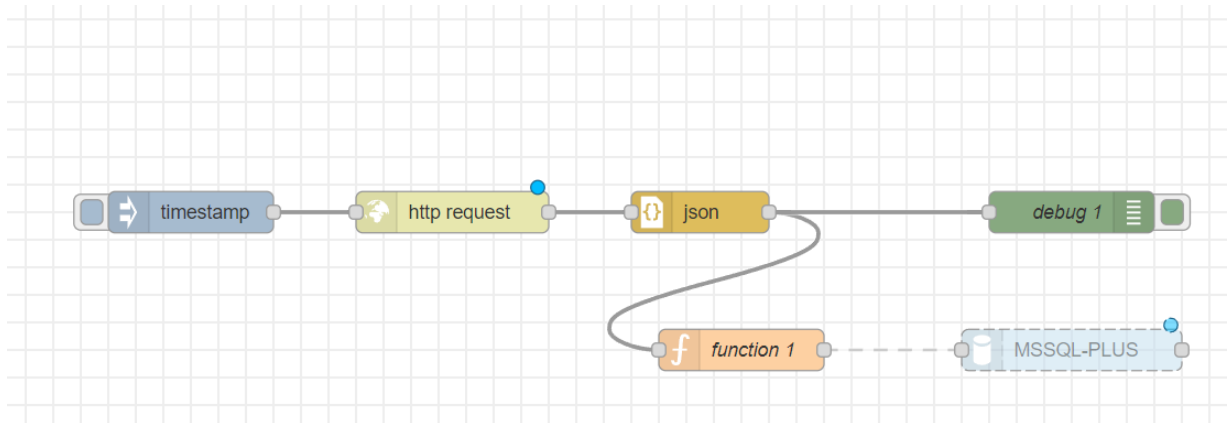


Hình 5: Http request gán URL của API đã tạo

Bước 3: Thêm dữ liệu vào SQL server



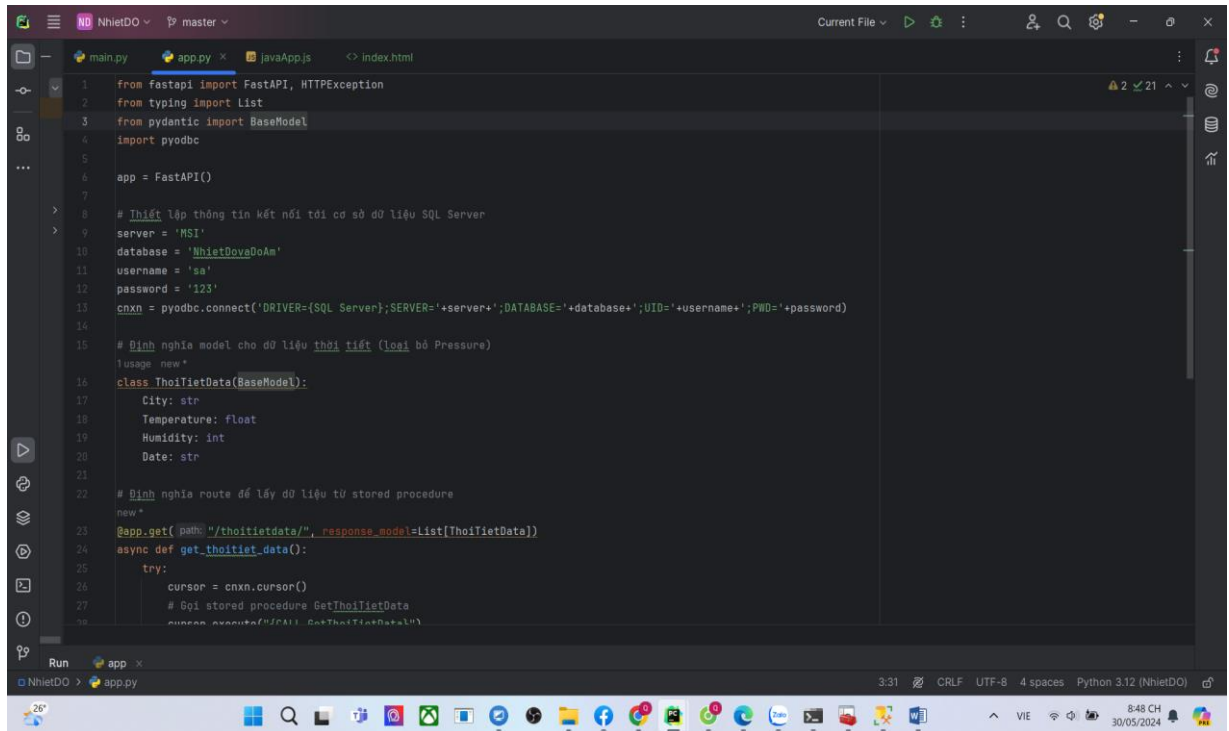
Hình 6: Cấu hình mssql để kết nối tới Sql server và Query



Hình 7: Dữ liệu đã được thêm

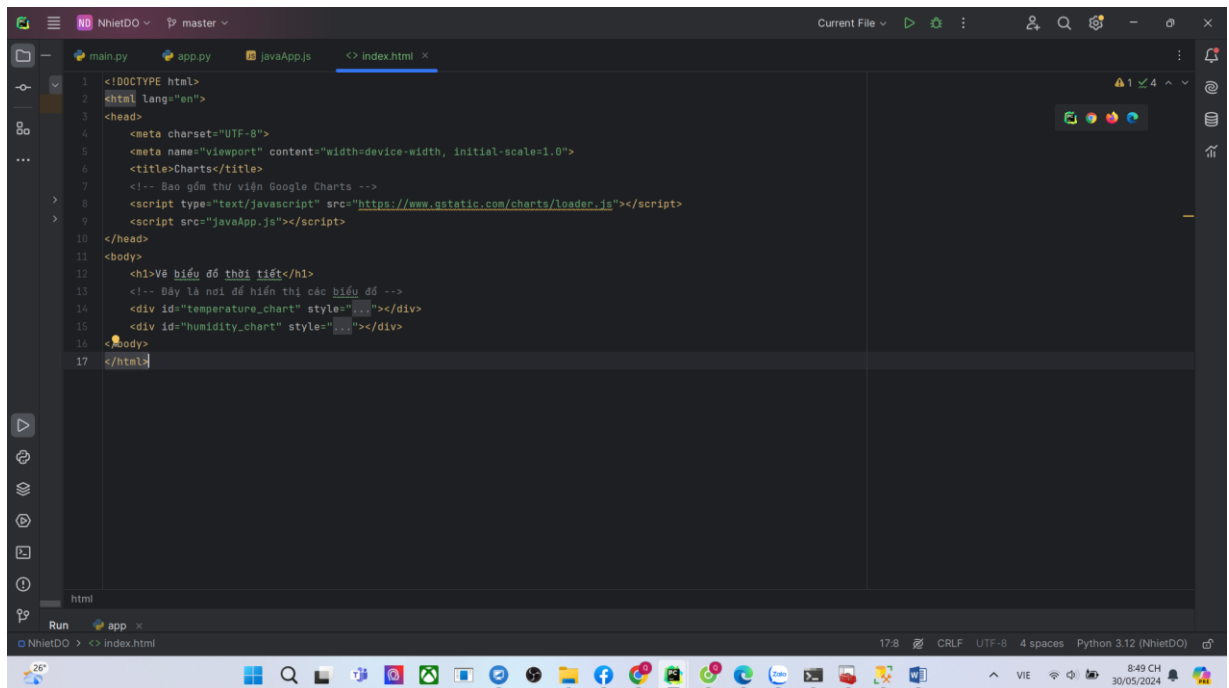
City	Date	Temperature	Humidity
undefined	2024-05-30	28.88	87
undefined	2024-05-31	25.4	88
undefined	2024-06-01	26.18	78
undefined	2024-06-02	27.28	83
undefined	2024-06-03	26.71	92
undefined	2024-06-04	24.83	95
NULL	NULL	NULL	NULL

Bước 4: Lấy dữ liệu từ Sql server và đẩy lên web tạo UI



```
1 from fastapi import FastAPI, HTTPException
2 from typing import List
3 from pydantic import BaseModel
4 import pyodbc
5
6 app = FastAPI()
7
8 # Thiết lập thông tin kết nối tới cơ sở dữ liệu SQL Server
9 server = 'MSI'
10 database = 'NhietDovaDoAm'
11 username = 'sa'
12 password = '123'
13 cnxn = pyodbc.connect('DRIVER={SQL Server};SERVER='+server+';DATABASE='+database+';UID='+username+';PWD='+password)
14
15 # Định nghĩa model cho dữ liệu thời tiết (loại bỏ Pressure)
16 usage new*
17 class ThoiTietData(BaseModel):
18     City: str
19     Temperature: float
20     Humidity: int
21     Date: str
22
23 # Định nghĩa route để lấy dữ liệu từ stored procedure
24 new*
25 @app.get("/thoitietdata/", response_model=List[ThoiTietData])
26 async def get_thoitiet_data():
27     try:
28         cursor = cnxn.cursor()
29         # Gọi stored procedure GetThoiTietData
30         cursor.execute("IF EXISTS GetThoiTietData+1")
```

Hình 8: Tạo app.py bằng python để kết nối tới SQL

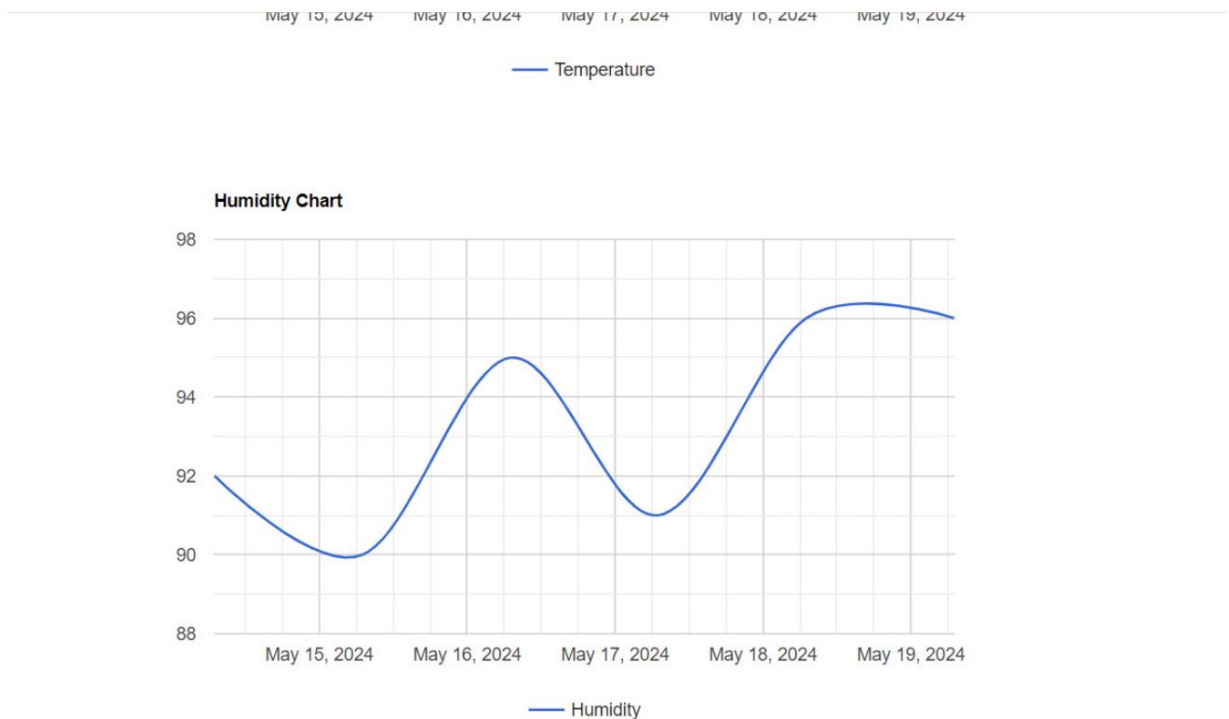


```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Charts</title>
7     <!-- Bao gồm thư viện Google Charts -->
8     <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
9     <script src="javaApp.js"></script>
10 </head>
11 <body>
12     <h1>Vẽ biểu đồ thời tiết</h1>
13     <!-- Đây là nơi để hiển thị các biểu đồ -->
14     <div id="temperature_chart" style="width: 50%; height: 100px;></div>
15     <div id="humidity_chart" style="width: 50%; height: 100px;></div>
16 </body>
17 </html>
```

Hình 9: Tạo file index.html để hiển thị

```
1 // Hàm để tải dữ liệu từ URL JSON và vẽ biểu đồ nhiệt độ và độ ẩm
2 google.charts.load('current', {'packages':['corechart']});
3 google.charts.setOnLoadCallback(drawCharts);
4
5 function drawCharts() {
6     drawTemperatureChart();
7     drawHumidityChart();
8 }
9
10 // usages new *
11 function loadData(url, callback) {
12     fetch(url).then(response => response.json()).then(data => callback(data)).catch(error => console.error('Error Loading data:', error));
13 }
14
15 // Hàm để vẽ biểu đồ nhiệt độ
16 function drawTemperatureChart() {
17     loadData('http://127.0.0.1:1880/get-temp', function(data) {
18         var chartData = [];
19         chartData.push(['Date', 'Temperature']);
20
21         data.forEach(function(entry) {
22             chartData.push([new Date(entry.Date), entry.Temperature]);
23         });
24
25         var dataTable = google.visualization.arrayToDataTable(chartData);
26
27         var options = {
28             // options
29         };
30         drawHumidityChart() // callback for loadData()
31     });
32 }
```

Hình 10: Tạo file js để lấy dữ liệu từ SQL tạo Chart



Hình 11: Dữ liệu được hiển thị

III. Kết luận

Xây dựng một trang web site “ Quản lý nhiệt độ thông minh” không chỉ là một dự án hấp dẫn và còn khám phá nhiệt độ và độ ẩm cập nhật lên xuống. Và hiểu biết các ngôn ngữ lập trình và đặt tay làm việc tự code lập trình trên các ngôn ngữ như là python, mssql server, node-red..., tiếp thu được những kinh nghiệm lập trình tạo lên và sửa lỗi để hoàn thiện dự án xây dựng web site này và rút ra kỹ năng và kinh nghiệm trong tương lai. Cuối cùng em cảm ơn thầy Đỗ Duy Cốp đã chỉ dạy em kinh nghiệm trong môn học lập trình này.