

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	5
1.1. Bối cảnh và Tầm Quan Trọng của Python	5
1.2. Mục Đích của Báo Cáo	5
1.3. Phạm Vi Nghiên Cứu	5
1.4. Phương Pháp Nghiên Cứu	5
II. Nội dung	6
2.1. Tổng quan về Python	6
2.2. Cấu trúc dữ liệu trong Python	6
2.3. Hàm và Module	6Error! Bookmark not defined.
2.4. Ứng dụng thực tế của Python	7
2.4.1. Ý tưởng	7
2.4.2. Các bước thực hiện	7
API	8
Hình 1: Tạo API	8
Bước 2: Sử dụng Node-red để nhận dữ liệu từ API	8
Hình 2: API được tạo	9
Hình 3: Khởi tạo Node-red	9
Hình 4: Kéo thả các luồng làm việc Node	10
Hình 5: Http request gán URL của API đã tạo	10
Bước 3: Thêm dữ liệu vào SQL server	10
Hình 6: Cấu hình mssql để kết nối tới Sql server và Query thêm dữ liệu vào	10
database	Error! Bookmark not defined.
Hình 7: Dữ liệu đã được thêm	11
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	12
Hình 9: Tạo file index.html để hiển thị	12

Hình 10: Tạo file js để lấy dữ liệu từ SQL tạo Chart	13
Hình 11: Dữ liệu được hiển thị.....	13
III. Kết luận	14
1. Python	14
2. Node-RED	14
3. Tích hợp giữa Python, Node-RED và SQL Server	14

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

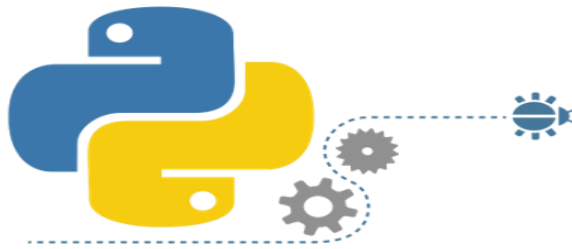
Để hoàn thành báo cáo này, em đã sử dụng các phương pháp nghiên cứu sau:

- 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

2.1. Tổng quan về Python

Python là một ngôn ngữ lập trình đa mục đích, dễ học và mạnh mẽ, được phát triển bởi Guido van Rossum và ra mắt lần đầu vào năm 1991. Với cú pháp đơn giản và gần gũi với ngôn ngữ tự nhiên, Python là một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới, được sử dụng rộng rãi trong các lĩnh vực khác nhau từ phát triển web, khoa học dữ liệu đến trí tuệ nhân tạo.



Python không chỉ dễ học mà còn linh hoạt và mở rộng, hỗ trợ nhiều phong cách lập trình và tích hợp tốt với các ngôn ngữ khác như C/C++, Java và .NET. Hệ sinh thái phong phú của Python cung cấp các thư viện và framework đa dạng, giúp lập trình viên dễ dàng phát triển các ứng dụng và dự án.

Điểm nổi bật của Python là cộng đồng lớn mạnh, với hàng triệu lập trình viên trên khắp thế giới, sẵn sàng chia sẻ kiến thức và kinh nghiệm. Nhờ vào điều này, Python không chỉ là một ngôn ngữ lập trình mà còn là một cộng đồng và một triển khai tri thức phong phú, đóng vai trò quan trọng trong việc giải quyết các thách thức hiện đại trong ngành công nghiệp và khoa học.

2.2. Cấu trúc dữ liệu trong Python

Python hỗ trợ các kiểu dữ liệu cơ bản như số nguyên (int), số thực (float), chuỗi (string), và các cấu trúc dữ liệu phức tạp hơn như list, tuple, set, và dictionary.

- List: là một danh sách có thể thay đổi, có thể chứa các phần tử khác nhau.
- Tuple: là một danh sách không thể thay đổi.
- Set: là một tập hợp các phần tử duy nhất.
- Dictionary: là một tập hợp các cặp key-value.

2.3. Hàm và Module

Hàm là một khối mã lệnh thực hiện một tác vụ cụ thể và có thể được gọi lại nhiều lần. Định nghĩa một hàm sử dụng từ khóa def:

```
def ham_mau():
```

```
    print("Đây là một hàm mẫu")
```

Module là một file chứa các định nghĩa và khai báo hàm, lớp, biến có thể được tái sử dụng trong các chương trình khác. Sử dụng từ khóa import để nhập module:

```
import math print(math.sqrt(16))
```

2.4. Ứng dụng thực tế của Python

2.4.1. Ý tưởng

Dự Án: Quản Lý Nhiệt Độ Thông Minh Mô Tả Dự án này sẽ tạo ra một hệ thống giám sát và quản lý nhiệt độ trong các phòng của một căn 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 tiếp dữ liệu đến SQL Server.

Lưu trữ dữ liệu nhiệt độ nhận được từ Node-RED. Lưu trữ thông tin các phòng và các ngưỡng nhiệt độ cho phép.

Python FastAPI: Xây dựng API để truy xuất dữ liệu nhiệt độ từ SQL Server.

Một giao diện web đơn giản để hiển thị dữ liệu nhiệt độ theo thời gian thực và cài đặt ngưỡng nhiệt độ. Các Bước Triển Khai

Thiết Lập Node-RED Cài đặt Node-RED. Sử dụng node mssql để gửi dữ liệu đến SQL Server.

SQL Server Tạo các bảng cần thiết

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

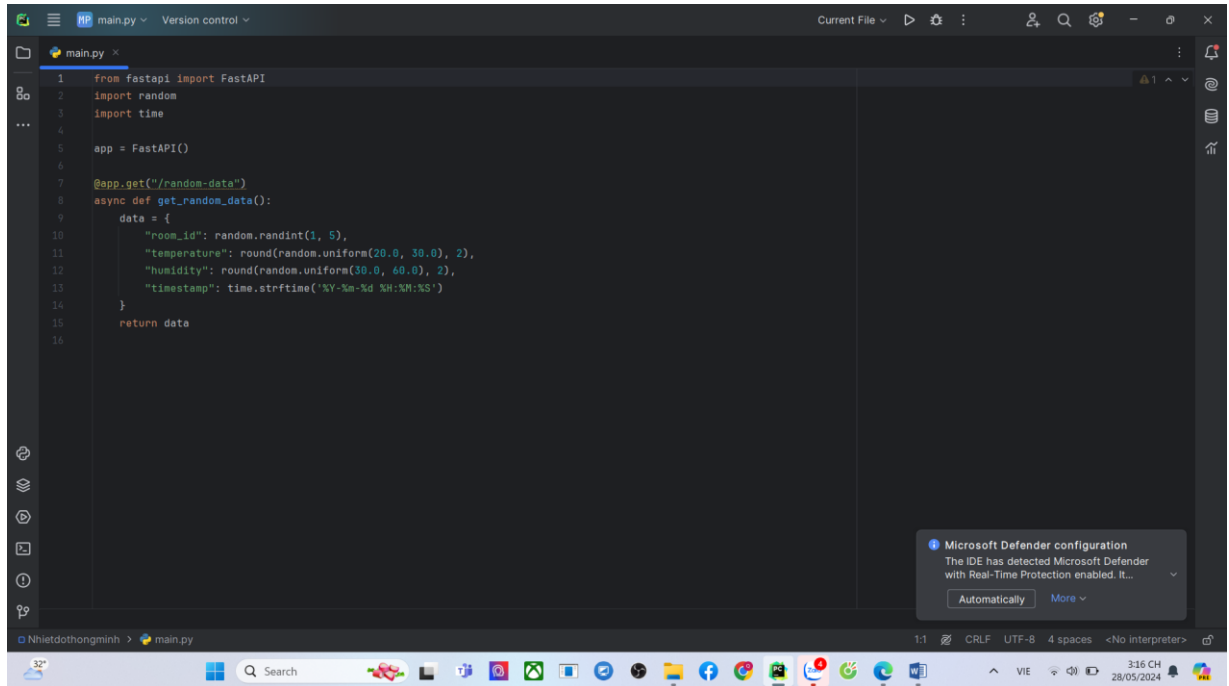
GET /temperatures/history: Lấy dữ liệu lịch sử nhiệt độ.

POST /thresholds: Cập nhật ngưỡng nhiệt độ cho phép kết nối FastAPI với SQL Server

2.4.2. Các bước thực hiện

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

tạo API



Hình 1: Tạo API

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

```
{"room_id":3,"nhiệt độ":22,27,"độ ẩm":49,5,"dấu thời gian":2024-05-26 16:33:28"}
```


Hình 2: API được tạo

```
node-red
26 May 16:33:52 - [info] Node-RED version: v3.1.9
26 May 16:33:52 - [info] Node.js version: v20.13.1
26 May 16:33:52 - [info] Windows_NT 10.0.18363 x64 LE
26 May 16:33:52 - [info] Loading palette nodes
26 May 16:33:54 - [info] Settings file : C:\Users\SIEU THI SO 4G\.node-red\settings.js
26 May 16:33:54 - [info] Context store : 'default' [module=memory]
26 May 16:33:54 - [info] User directory : \Users\SIEU THI SO 4G\.node-red
26 May 16:33:54 - [warn] Projects disabled : editorTheme.projects.enabled=false
26 May 16:33:54 - [info] Flows file : \Users\SIEU THI SO 4G\.node-red\flows.json
26 May 16:33:54 - [info] Server now running at http://127.0.0.1:1880/
26 May 16:33:54 - [warn]

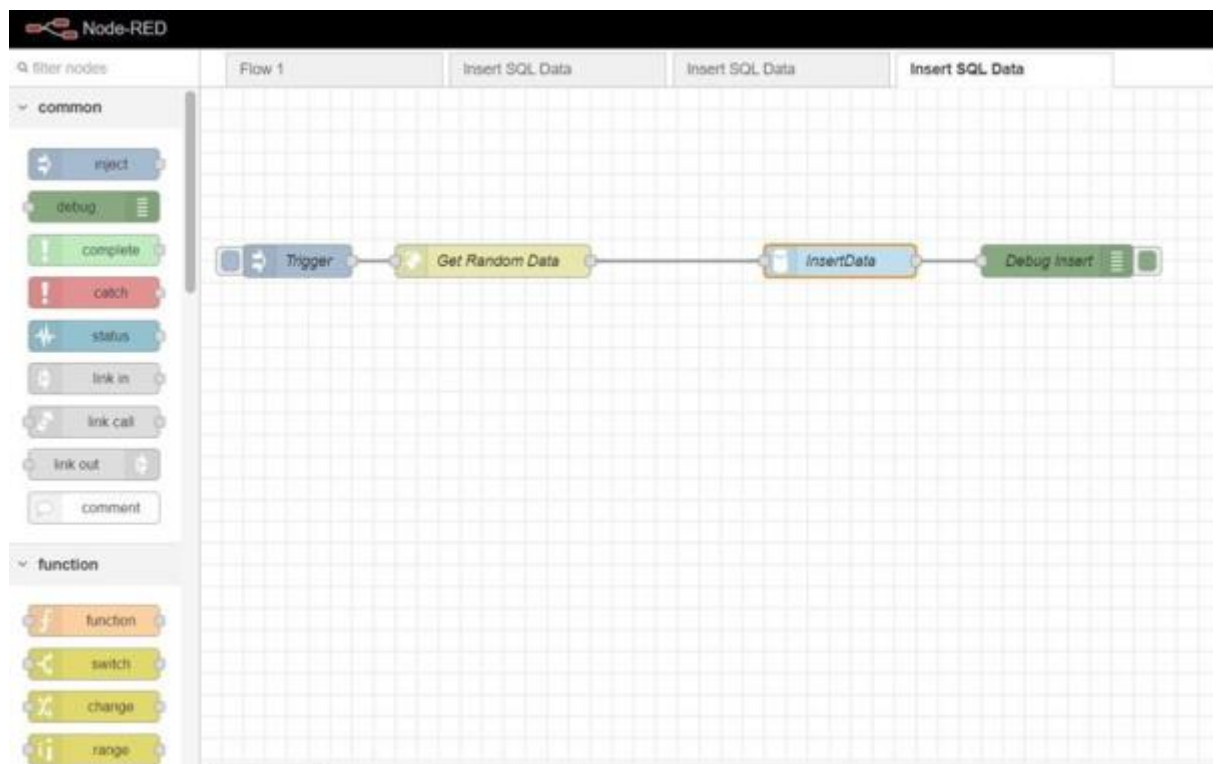
-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

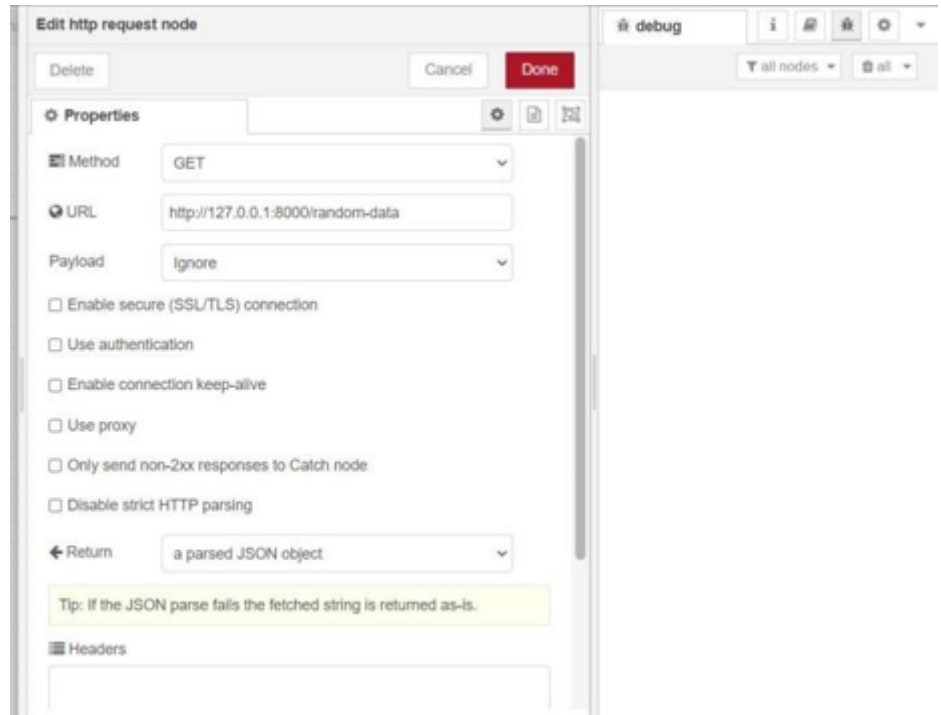
You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----

26 May 16:33:54 - [info] Starting flows
26 May 16:33:54 - [info] Started flows
```

Hình 3: Khởi tạo Node-red

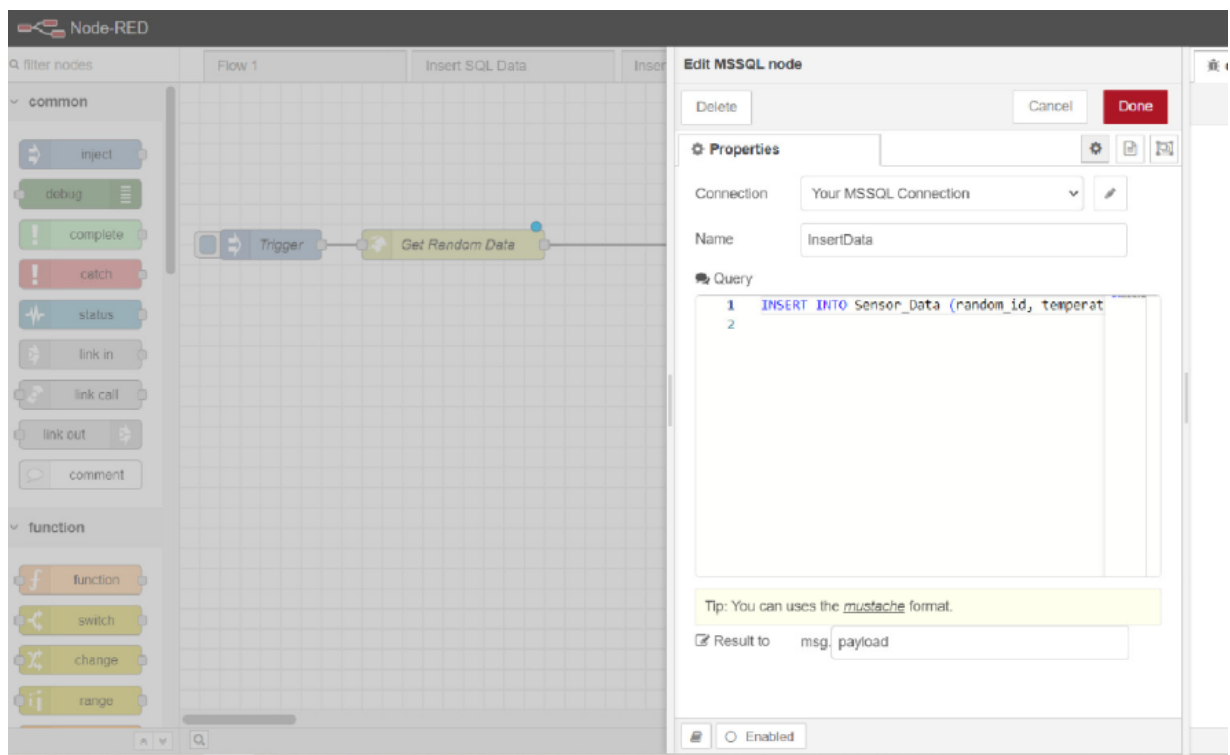


Hình 4: Kéo thả các luồng làm việc Node

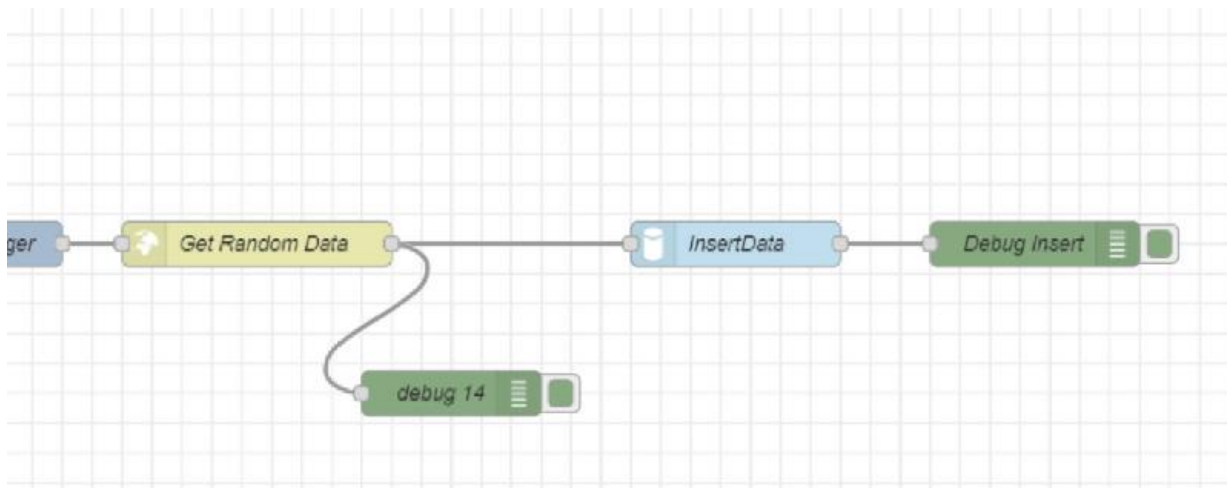


Hình 5: Http request gán URL của API đã tạo

Bước 3: Thêm dữ liệu vào SQL server



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

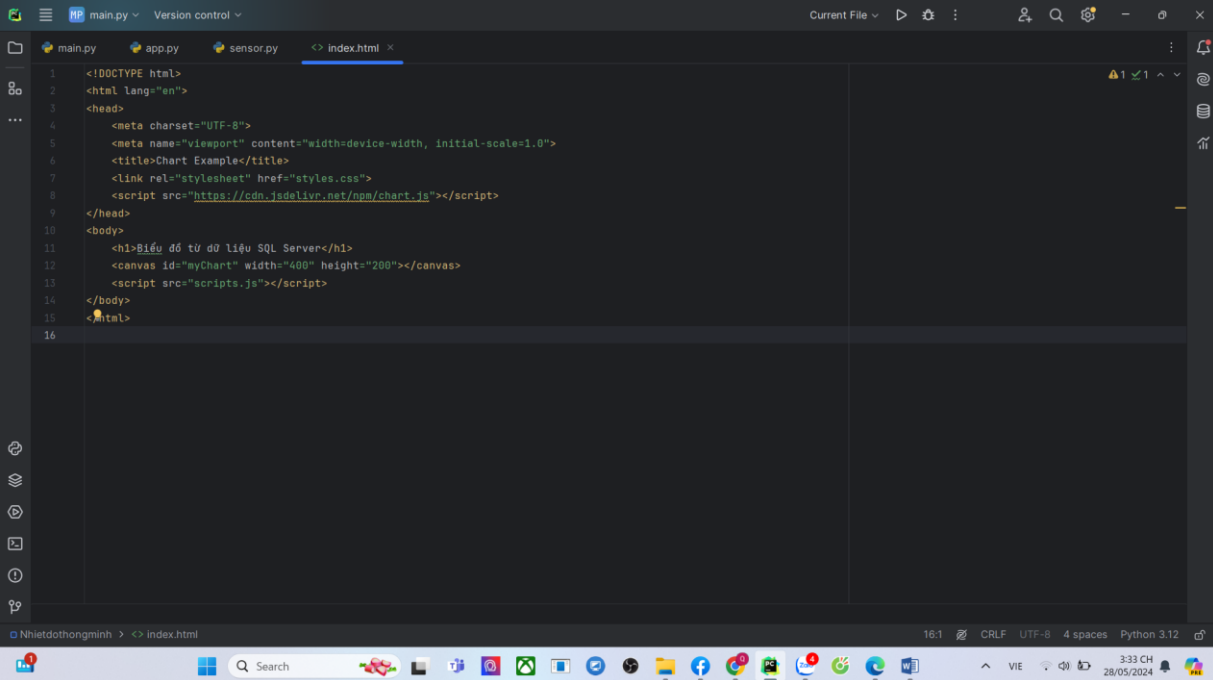


Hình 7: Dữ liệu đã được thêm

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

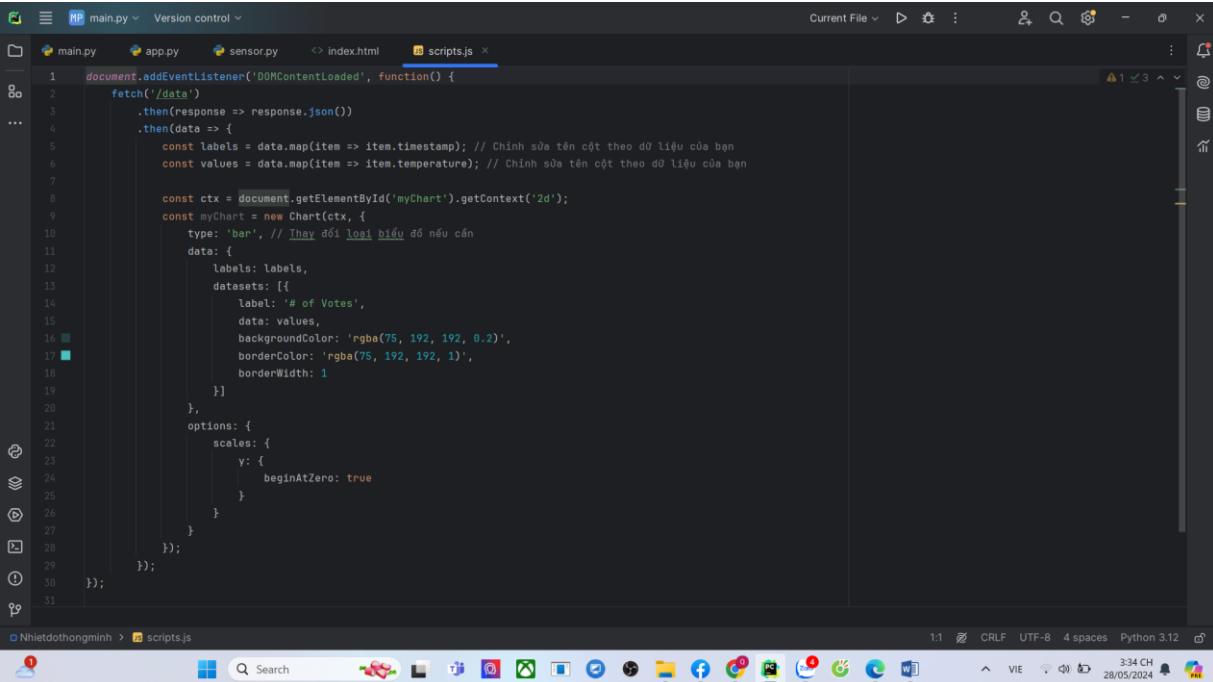
```
1 from flask import Flask, jsonify, render_template_string, send_from_directory
2 import pyodbc
3
4 app = Flask(__name__)
5
6 # Cấu hình kết nối đến SQL Server
7 server = 'MSIBRV-8T8090H'
8 database = 'SensorData'
9 username = 'sa'
10 password = '123'
11 connection_string = f'DRIVER={{SQL Server}};SERVER={server};DATABASE={database};UID={username};PWD={password}'
12
13 @app.route('/')
14 def index():
15     return render_template_string(open('index.html').read())
16
17 @app.route('/data')
18 def get_data():
19     conn = pyodbc.connect(connection_string)
20     cursor = conn.cursor()
21     cursor.execute('SELECT * FROM SensorData')
22     rows = cursor.fetchall()
23     data = [dict(zip([column[0] for column in cursor.description], row)) for row in rows]
24     conn.close()
25     return jsonify(data)
26
27 @app.route('/<path:filename>')
28 def static_files(filename):
29     return send_from_directory('.', filename)
30
31 if __name__ == '__main__':
```

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



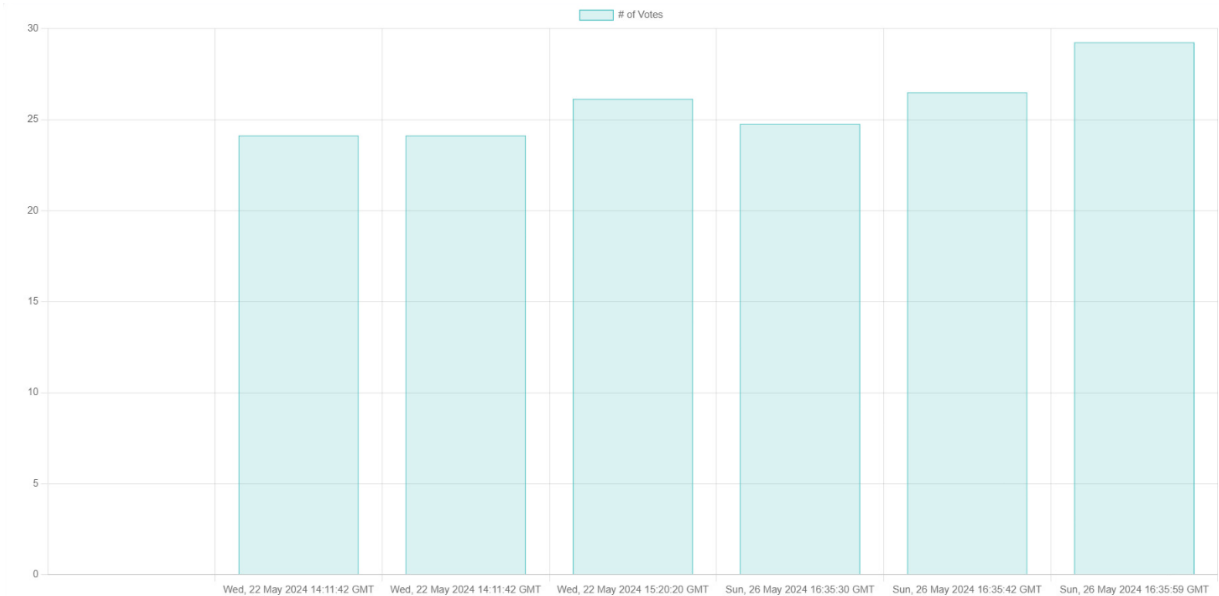
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Chart Example</title>
7   <link rel="stylesheet" href="styles.css">
8   <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
9 </head>
10 <body>
11   <h1>Biểu đồ từ dữ liệu SQL Server</h1>
12   <canvas id="myChart" width="400" height="200"></canvas>
13   <script src="scripts.js"></script>
14 </body>
15 </html>
16
```

Hình 9: Tạo file index.html để hiển thị



```
1 document.addEventListener('DOMContentLoaded', function() {
2   fetch('/data')
3     .then(response => response.json())
4     .then(data => {
5       const labels = data.map(item => item.timestamp); // Chính sửa tên cột theo dữ liệu của bạn
6       const values = data.map(item => item.temperature); // Chính sửa tên cột theo dữ liệu của bạn
7
8       const ctx = document.getElementById('myChart').getContext('2d');
9       const myChart = new Chart(ctx, {
10        type: 'bar', // Thay đổi loại biểu đồ nếu cần
11        data: {
12          labels: labels,
13          datasets: [{
14            label: '# of Votes',
15            data: values,
16            backgroundColor: 'rgba(75, 192, 192, 0.2)',
17            borderColor: 'rgba(75, 192, 192, 1)',
18            borderWidth: 1
19          }]
20        },
21        options: {
22          scales: {
23            y: {
24              beginAtZero: true
25            }
26          }
27        }
28      });
29    });
30 });
```

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



Hình 11: Dữ liệu được hiển thị

III. Kết luận

1. Python

Python là một ngôn ngữ lập trình mạnh mẽ, linh hoạt và dễ học, được sử dụng rộng rãi trong nhiều lĩnh vực như phát triển web, khoa học dữ liệu, trí tuệ nhân tạo và tự động hóa. Chúng ta đã hiểu về cú pháp đơn giản của Python, các cấu trúc dữ liệu cơ bản như list, tuple, dictionary, và cách sử dụng hàm và module. Python cung cấp một nền tảng mạnh mẽ để phát triển các ứng dụng phức tạp và giải quyết các vấn đề thực tiễn.

2. Node-RED

Node-RED là một công cụ lập trình trực quan (visual programming tool) được sử dụng rộng rãi trong Internet of Things (IoT), tự động hóa và xử lý dữ liệu. Chúng ta đã tìm hiểu cách sử dụng Node-RED để tạo ra các luồng làm việc phức tạp thông qua việc kéo và thả các nodes, và kết nối chúng lại với nhau để thực hiện các chức năng đa dạng. Node-RED cung cấp một cách tiếp cận dễ dàng và mạnh mẽ cho việc phát triển các ứng dụng IoT và tự động hóa.

3. Tích hợp giữa Python, Node-RED và SQL Server

Thông qua việc kết hợp Python, Node-RED và SQL Server, chúng ta có thể xây dựng các hệ thống phức tạp và mạnh mẽ. Python cung cấp các công cụ để xử lý và phân tích dữ liệu, Node-RED giúp kết nối các thiết bị và dịch vụ, và SQL Server là nơi lưu trữ và quản lý dữ liệu. Sự kết hợp giữa ba công nghệ này mở ra nhiều cơ hội mới trong việc phát triển các ứng dụng thông minh, hiệu quả và an toàn.

Tổng kết lại, việc tìm hiểu về Python, Node-RED và SQL Server đã giúp chúng ta có cái nhìn tổng quan về ba công nghệ quan trọng trong lĩnh vực công nghệ thông tin. Sự hiểu biết về cách các công nghệ này hoạt động và làm thế nào để kết hợp chúng với nhau sẽ là một lợi thế lớn trong việc phát triển các ứng dụng và giải quyết các vấn đề thực tiễn trong tương lai. Đó là những gì chúng ta đã tìm hiểu trong báo cáo này về Python, Node-RED và SQL Server. Hy vọng rằng thông tin này sẽ giúp bạn hiểu rõ hơn về các công nghệ này và tận dụng được tiềm năng của chúng trong công việc và nghiên cứu của mình.