

# FASTAPI-NODERED-MSSQL

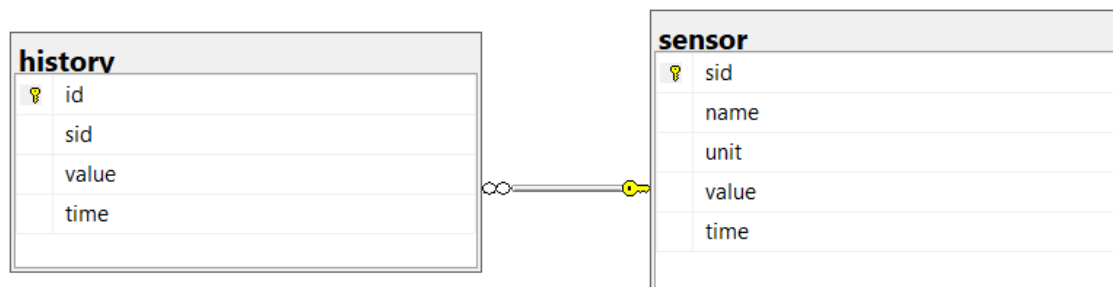
Họ Tên: Hà Trung Hiếu K205480106045

Đề:

1. Dùng FastApi của python, xây dựng API (tự đưa vào logic xử lý input => output)  
2. Cài đặt Node-Red trên windows (ko cần máy ảo), tạo chu trình tự động hoá gửi dữ liệu tới api, nhận về kết quả, lưu trữ vào database Sql server. 3. Tạo web đơn giản (html+js+css) với backend có thể là c# asp dot net, php, node-red, hoặc chính là python FastApi để lấy dữ liệu từ database Sql server, vẽ biểu đồ dữ liệu đã lưu. (Chart có thể dùng tùy ý thư viện thích hợp)

## 1. Chuẩn bị database

-Tạo 2 bảng và tạo mối liên kết



- Sử dụng dữ liệu từ bảng history để tạo ra một chuỗi JSON đại diện cho dữ liệu có thể được sử dụng trong ứng dụng hiển thị biểu đồ hoặc báo cáo. Chuỗi JSON này chứa thông tin về các mục trong bảng history, bao gồm các cột như id, sid, value, và time.

```

USE [chuchua]
GO
/***** Object:  StoredProcedure [dbo].[SP_Chart]    Script Date: 15/05/2024 1:47:26 SA *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[SP_Chart]
AS
BEGIN
    DECLARE @json nvarchar(max) = N'{"ok":1,"msg":"ok","data":[';

    SELECT @json += FORMATMESSAGE(N'{"id": "%d", "sid": "%d", "value": "%s", "time": "%s"}',
                                   [id], [sid], CONVERT(nvarchar(50), [value]), CONVERT(nvarchar(50), [time]))
    FROM history;

    IF RIGHT(@json, 1) = ','
    BEGIN
        SET @json = LEFT(@json, LEN(@json) - 1);
    END

    SET @json = @json + ']]';

    SELECT @json AS json;
END

```

2. Sử dụng FastAPI để tạo một API đơn giản trả về dữ liệu về nhiệt độ và độ ẩm.

```

1  from fastapi import FastAPI
2  import random
3
4  app = FastAPI()
5
6  # Hàm để sinh ngẫu nhiên giá trị nhiệt độ
7  def generate_random_temperature():
8
9      return round(random.uniform(25, 30), 2)
10
11 # Hàm để sinh ngẫu nhiên giá trị độ ẩm
12 def generate_random_humidity():
13
14     return round(random.uniform(80, 95), 2)
15
16 @app.get("/")
17 async def read_data():
18     # Gán giá trị nhiệt độ và độ ẩm bằng hàm sinh ngẫu nhiên
19     temperature = generate_random_temperature()
20     humidity = generate_random_humidity()
21     return {"temperature": temperature, "humidity": humidity}

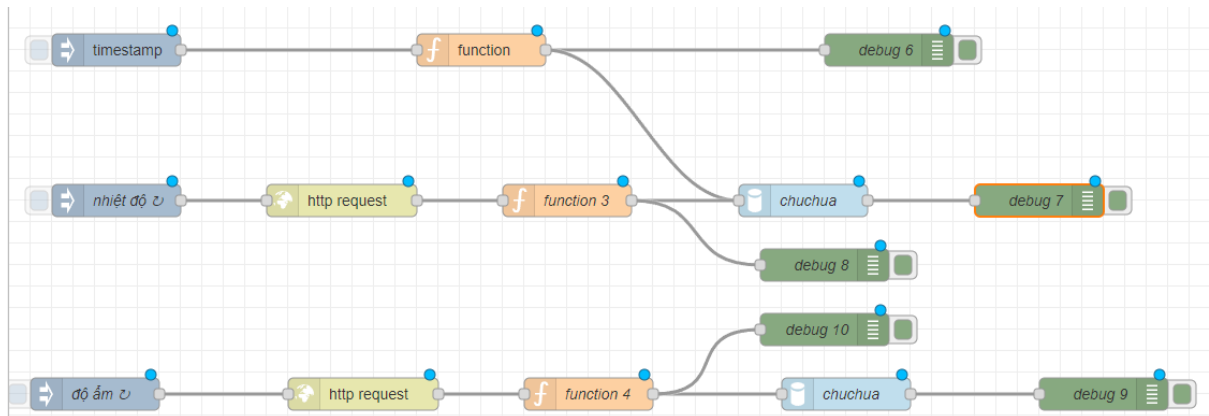
```

```

C:\Users\hatrg\OneDrive\Máy tính\hii>uvicorn main:app --reload
INFO: Will watch for changes in these directories: ['C:\\Users\\hatrg\\OneDrive\\Máy tính\\hii']
INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO: Started reloader process [13996] using WatchFiles
INFO: Started server process [15036]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: 127.0.0.1:50419 - "GET / HTTP/1.1" 200 OK
INFO: 127.0.0.1:50420 - "GET / HTTP/1.1" 200 OK

```

### 3. Node-red



1. **Inject Node:** Node này được sử dụng để tạo ra các sự kiện (events) bằng cách chuyển đổi thời gian (payloadType) thành dữ liệu ngày tháng (date).

**Edit inject node**

Delete Cancel Done

**Properties**

Name: nhiệt độ

msg.payload = timestamp

msg.topic = a\_z

2. **Function Node (Hàm 2):** Node này thực hiện chuyển đổi dữ liệu nhận được từ HTTP Request Node từ dạng JSON sang câu lệnh SQL **INSERT INTO history (sid, value) VALUES ('1', <nhiệt độ>)**. Câu lệnh SQL này sau đó được chuyển đến MSSQL Node để thực thi.

**Edit function node**

Delete Cancel Done

**Properties**

Name function 3

Setup On Start **On Message** On Stop

```
1 var json = JSON.parse(msg.payload);
2 msg.payload = `INSERT INTO history (sid, value) VALUES ('1',$
3 return msg;
```

4. Sử dụng api.aspx.cs và phần html, java script để đẩy dữ liệu lên web và vẽ biểu đồ