Họ Tên: Vi Duy Quốc

MSSV: K205480106034

Đề Bài: Nhiệt độ, độ ẩm random

Yêu cầu

1. Trình bày thuật toán xử lý của api, ý nghĩa
2. Mô tả các bước cài đặt+ snap màn hình minh hoạ.
3. Mô tả quá trình chạy demo, hiểu được luồng xử lý dữ liệu. Hình ảnh minh hoạ

4. Kết luận: đã tìm hiểu được những kỹ thuật gì? Đã cài đặt và cấu hình thành công phần mềm nào? Đã tạo đc api gì? Đã phối hợp các kỹ thuật lập trình gì để đạt được điều gì? Kết quả cuối cùng xấu đẹp ra sao?

Những mục chính đã làm

- Tạo file python sử dụng fash api tạo dữ liệu random cho nhiệt độ và độ ẩm

- Sử dụng node-red lấy dữ liệu từ địa chỉ local của fash api -

  sử dụng asp dot net để lấy dữ liệu và vẽ biểu đồ

Quá trình làm bài

1. Tạo file python sử dụng fash api tạo dữ liệu random cho nhiệt độ và độ ẩm

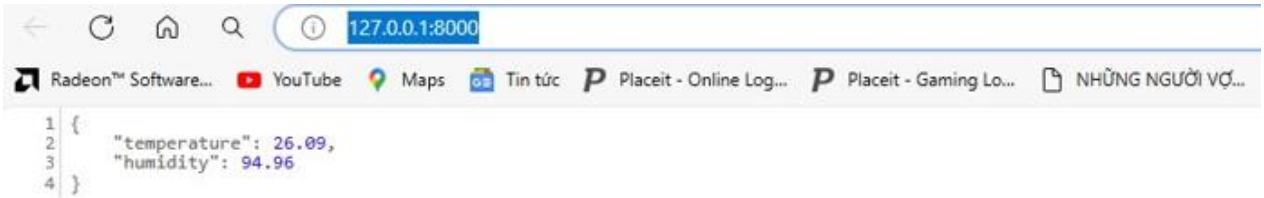
Em sẽ tạo một file python và sử dụng fash api trong đó có 2 hàm nhiệt dộ và độ ẩm sẽ được random để sinh ra giá trị ngẫu nhiên trong khoảng nhất định

```python
from fastapi import FastAPI
import random

app = FastAPI()

# Hàm để sinh ngẫu nhiên giá trị nhiệt độ
def generate_random_temperature():

    return round(random.uniform(25, 30), 2)

# Hàm để sinh ngẫu nhiên giá trị độ ẩm
def generate_random_humidity():

    return round(random.uniform(80, 95), 2)

@app.get("/")
async def read_data():
    # Gán giá trị nhiệt độ và độ ẩm bằng hàm sinh ngẫu nhiên
    temperature = generate_random_temperature()
    humidity = generate_random_humidity()
    return {"temperature": temperature, "humidity": humidity}
```

Khởi chạy fash api

```
PS D:\tai lieu hoc tap\laptrinhpython> uvicorn main:app --reload
INFO:     Will watch for changes in these directories: ['D:\\tai lieu hoc tap\\laptrinhpython']
INFO:     Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO:     Started reloader process [23348] using WatchFiles
INFO:     Started server process [11276]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
```

Sau khi khởi chạy nó sẽ trả về một chuỗi dạng json trên local của mình



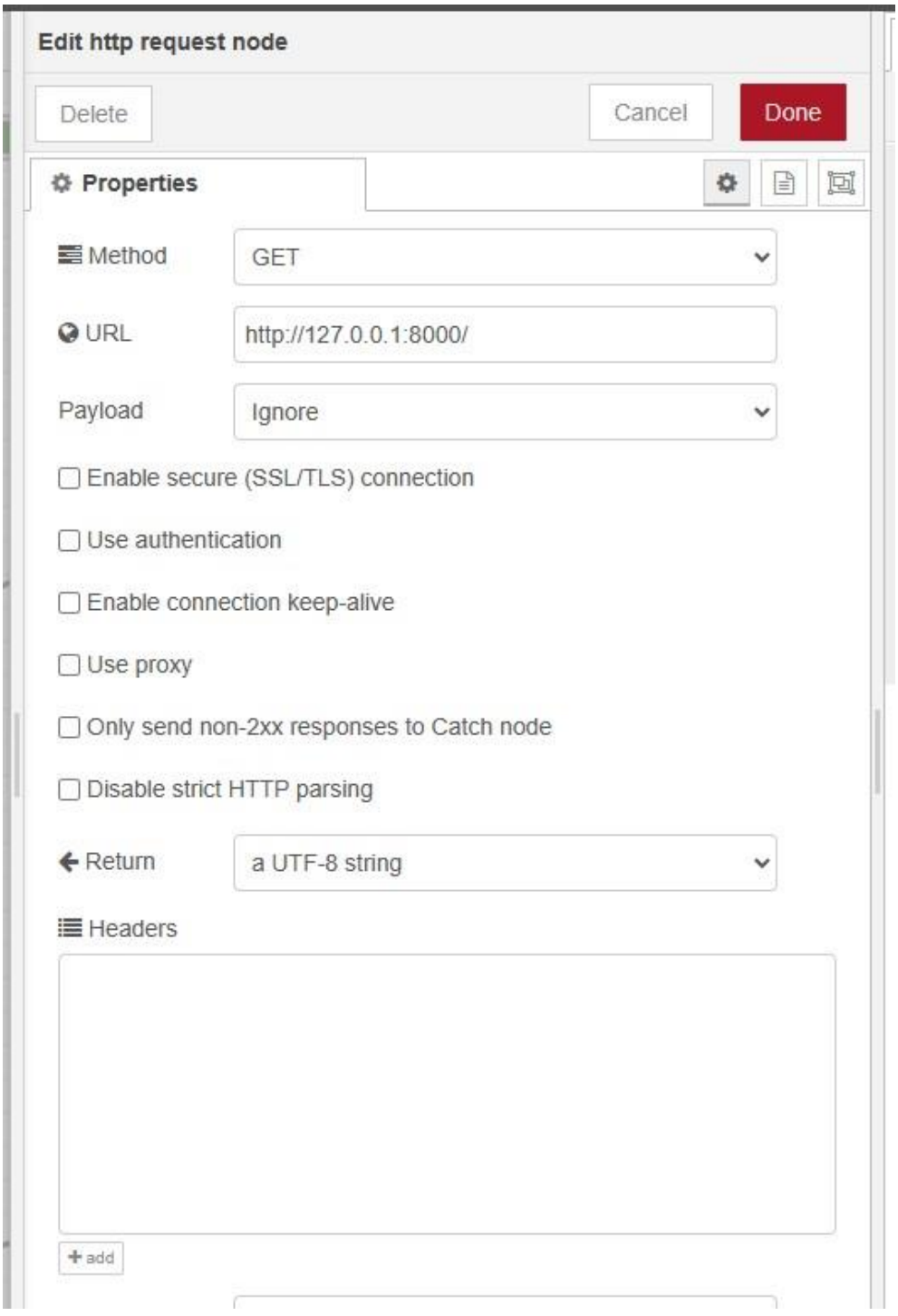
\

1.2. Sử dụng node-red lấy dữ liệu từ địa chỉ local của fash api

- tiếp theo em sẽ sử dụng node-red và dán đường link vào http response

- Ở đây em chia ra thành 2 nhánh nhiệt độ và độ ẩm và set thời gian chạy tự động

Sau mỗi khoảng thời gian nó sẽ tự lưu dữ liệu vào trong database cho mình.



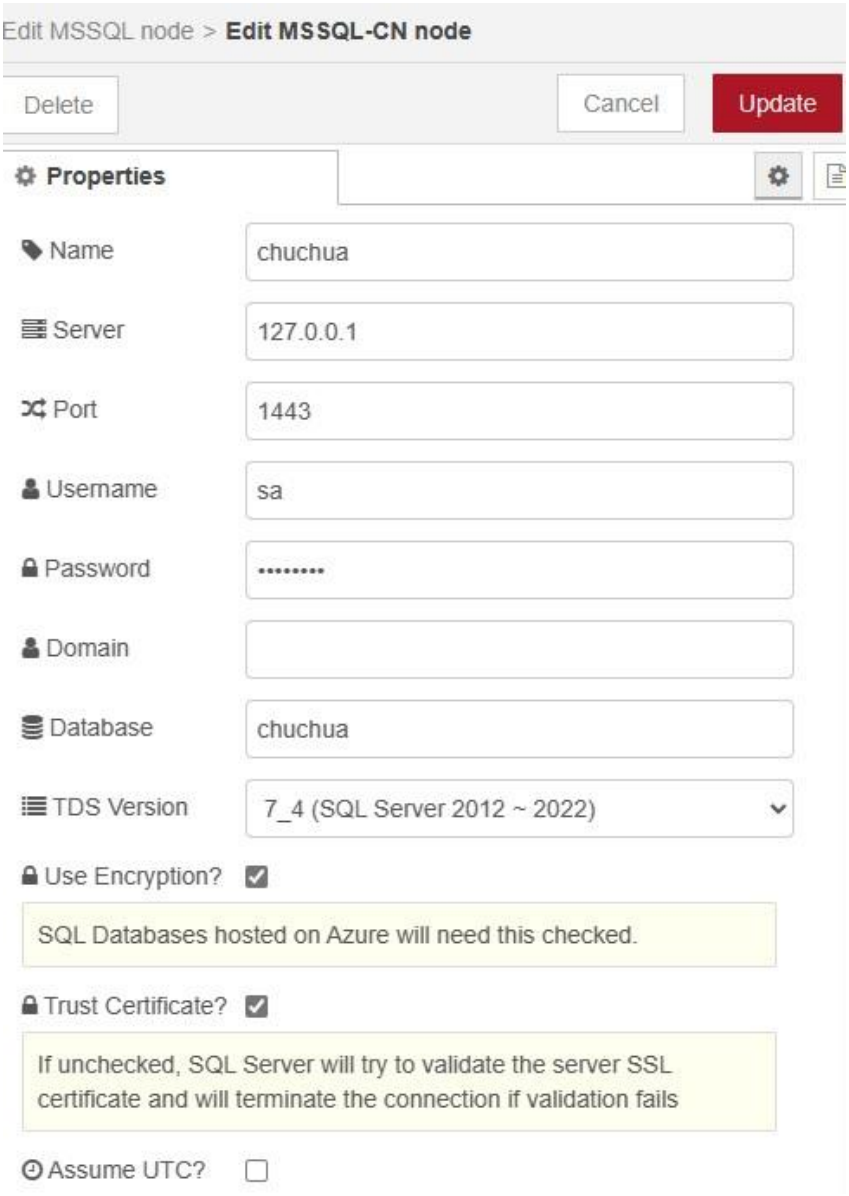
Sau đó viết một funcion để lưu dữ liệu vào database sql var

json = JSON.parse(msg.payload);

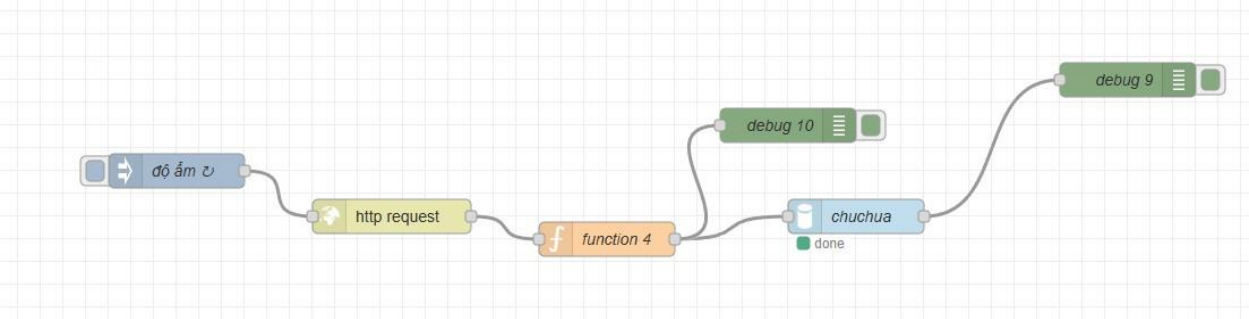
msg.payload = `INSERT INTO history (sid, value) VALUES

('1',${json.temperature})` return
msg;

- cài đặt node-red-contrib-mssql-plus: sau khi cài đặt cấu hình các thông tin cho node



- Cấu trúc của một đoạn code node-red sử dụng để lưu dữ liệu vào sql



- Sau đó em viết store procedure trả về dữ liệu dạng json để đẩy lên web

```
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
```

-store procedure: khởi tạo biến json và tạo chuỗi json để lặp qua từng dòng dữ liệu là lưu vào biến json sau đó kiểm tra phần tử cuối nếu là dấu phẩy thì xóa

1.3 sử dụng asp dot net để lấy dữ liệu và vẽ biểu đồ

cuối cùng em clone bài tập của thầy đã gửi trên nhóm và sau đó chỉnh lại chuỗi kết nối sql ở file api.aspx.cs và phần html, java script để đẩy dữ liệu lên web và vẽ biểu đồ như yêu cầu.

- Tạo chuỗi kết nối với databse, tạo đối tượng sql command để gọi store procedure và thực thi store procedure trả về một chuỗi json

```aspx
using System;
using System.Web;
using System.Web.UI;
using System.Data.SqlClient;
using System.Data;

namespace demo_api_57kmt
{
    public partial class api : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            string connectionString = "Data Source=127.0.0.1,1443;Initial Catalog=chuchua;User Id=sa;Password=123;";
            using (SqlConnection connection = new SqlConnection(connectionString))
            {
                using (SqlCommand command = new SqlCommand("SP_Chart", connection))
                {
                    command.CommandType = CommandType.StoredProcedure;

                    try
                    {
                        connection.Open();
                        object kq = command.ExecuteScalar();
                        string json = (string)kq;
                        this.Response.ContentType = "application/json";
                        this.Response.Write(json);
                    }
                    catch (Exception ex)
                    {
                        this.Response.ContentType = "application/json";
                        this.Response.Write("{\"ok\":0,\"msg\":\"" + ex.Message + "\"}");
                    }
                }
            }
        }
```

- file script có tác dụng tải dữ liệu từ API, kiểm tra và xử lý dữ liệu để tạo ra hai bộ dữ liệu cho nhiệt độ và độ ẩm, sau đó vẽ biểu đồ đường (line chart) để hiển thị dữ liệu này. Nếu có lỗi xảy ra trong quá trình lấy dữ liệu, nó sẽ được ghi vào console.

```javascript
document.addEventListener('DOMContentLoaded', function () {
    fetch('api.aspx')
        .then(response => {
            if (!response.ok) {
                throw new Error('Network response was not ok ' + response.statusText);
            }
            return response.json();
        })
        .then(data => {
            console.log(data); // Ghi nhật ký dữ liệu trả về để kiểm tra

            if (data.ok !== 1) {
                throw new Error('API returned an error: ' + data.msg);
            }

            let labels = [];
            let tempValues = [];
            let humidityValues = [];
            let currentTemp = null;
            let currentHumidity = null;

            data.data.forEach(item => {
                if (!labels.includes(item.time)) {
                    labels.push(item.time);
                }

                if (item.sid == 1) {
                    tempValues.push(item.value);
                    currentTemp = item.value; // Cập nhật nhiệt độ hiện tại
                    humidityValues.push(null);
                } else if (item.sid == 2) {
                    tempValues.push(null);
                    humidityValues.push(item.value);
                    currentHumidity = item.value; // Cập nhật độ ẩm hiện tại
                }
            });

            // Hiển thị nhiệt độ và độ ẩm hiện tại
            document.getElementById('currentTemp').innerText = currentTemp ? currentTemp + '°C' : 'N/A';
```

-Kết quả cuối cùng: