

Họ và tên : Nguyễn Thế Định
Mssv K205480106013

Đề bài : 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)

Yêu cầu:

1. Trình bày thuật toán xử lý của api, ý nghĩa của nó
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 đã làm :

- tạo file python sử dụng fashapi tạo dữ liệu random cho nhiệt độ
- 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 :

Tạo file python sử dụng fashapi tạo dữ liệu random cho nhiệt độ
Install các thư viện cần thiết

```
View Go Run ... bt_thaycop_nhietdoranom-master
main.py
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     return round(random.uniform(25, 30), 2)
9
10
11
12 @app.get("/")
13 async def read_data():
14     # Lấy giá trị nhiệt độ và độ ẩm bằng hàm sinh ngẫu nhiên
15     temperature = generate_random_temperature()
16     return {"temperature": temperature, "humidity": 87.56}
17
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS python3.9

```
units>api) (3.3.2)
Requirement already satisfied: certifi>2017.4.17 in c:\users\dingh\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from requests-
>api) (2024.2.2)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\dingh\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from requests-
>api) (2.2.1)
Requirement already satisfied: idna<4,>=2.5 in c:\users\dingh\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from requests-
>api) (3.6)
Building wheels for collected packages: fash, api
  Building wheel for fash (setup.py) ... done
  Created wheel for fash: filename=fash-0.0.4-py3-none-any.whl size=1135 sha256=3cbb0f0b65d71816a202a42bd8b24c436d6c385099e5a27eff52b5430ff6f2
  Stored in directory: c:\users\dingh\appdata\local\pip\cache\wheels\c2\63\27\5f6e52bd9c314de2a6210b9ad6c65a635835fefe72a123490
  Building wheel for api (setup.py) ... done
  Created wheel for api: filename=api-0.0.7-py3-none-any.whl size=2334 sha256=744700f8965846b8bd098a2845d177a03ef59b63008d4fb24c936fba02a88e
  Stored in directory: c:\users\dingh\appdata\local\pip\cache\wheels\fa\aa\62\7a58dc8d2d5b68a3b46506e97d896a7128456adb95489ba973
Successfully built fash api
Installing collected packages: nose, fash, api
WARNING: The scripts nosetests-3.4.exe and nosetests.exe are installed in 'C:\Users\Dinh\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9_qbz5n2kfra8p0\LocalCache\local-packages\Python39\Script
s' which is not on PATH.
```

Chạy fash api

```
INFO:      Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO:      Started reloader process [26880] using WatchFiles
INFO:      Started server process [2912]
INFO:      Waiting for application startup.
INFO:      Application startup complete.
INFO:      127.0.0.1:63150 - "GET / HTTP/1.1" 200 OK
INFO:      127.0.0.1:63293 - "GET / HTTP/1.1" 200 OK
```

sau khi chạy trả về chuỗi dạng json trên local của mình

```
← → ↺ 127.0.0.1:8000
Pretty-print
{"temperature":26.45,"humidity":87.56}
```

Sử dụng node - red kết nối node - red với local của mình :

Edit http request node

Delete

Cancel

Done

⚙️ Properties

Method

GET

▼

🌐 URL

http://127.0.0.1:8000/

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 UTF-8 string

▼

📋 Headers

+ add

🏷️ Name

Name

📄

Enabled

Sau đó viết 1 đoạn function để lưu dữ liệu vào database sql



Cài đặt node-red-contrib-mssql-plus ---> cài đặt cấu hình cho node

Edit MSSQL node > Edit MSSQL-CN node

Delete

Cancel

Update

⚙ Properties

⚙

📄

🔑 Name

thedin

🖨 Server

127.0.0.1

🔌 Port

1443

👤 Username

sa

🔒 Password

.....

👤 Domain

🗄 Database

anhdin

📋 TDS Version

7_4 (SQL Server 2012 ~ 2022) ▾

🔒 Use Encryption? ☒

SQL Databases hosted on Azure will need this checked.

🔒 Trust Certificate? ☒

If unchecked, SQL Server will try to validate the server SSL certificate and will terminate the connection if validation fails

⌚ Assume UTC? ☐

Pass time values in UTC or local time.

⌚ Connect Timeout

The number of milliseconds before the attempt to connect is considered failed.

⌚ Request Timeout

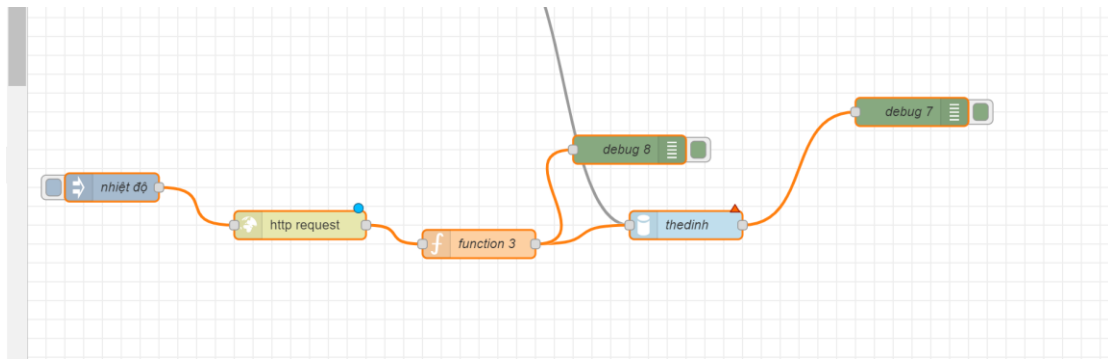
📄

☐ Enabled

👤 1

On all flows ▾

Cấu trúc của đoạn node-red để lưu dữ liệu về sql



Sau đó em viết SP trả dữ liệu dạng json rồi đẩy lên web

```

GO
/***** Object:  StoredProcedure [dbo].[SP_Chart]    Script Date: 5/15/2024 1:12:49 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[SP_Chart]
AS
BEGIN
    DECLARE @json nvarchar(max) = N'{"ok":1,"msg":"ok","data":[';

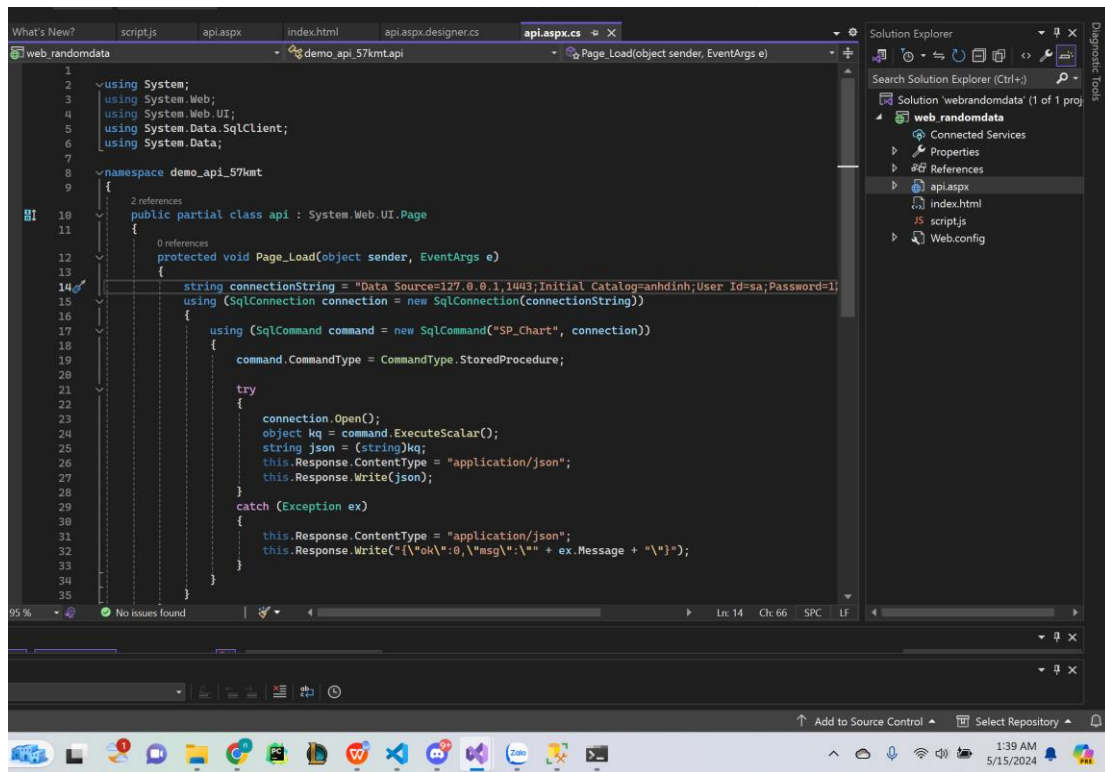
    SELECT @json += FORMATMESSAGE(N'{"id": "%d", "sid": "%d", "value": "%s", "time": "%s"}',
                                   [id], [sid], CONVERT(nvarchar(50), [value]), CONVERT(nvarchar(50),
    FROM history;

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

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

    SELECT @json AS json;
END
GO
USE [master]
GO
ALTER DATABASE [anh Dinh] SET READ_WRITE
GO
  
```

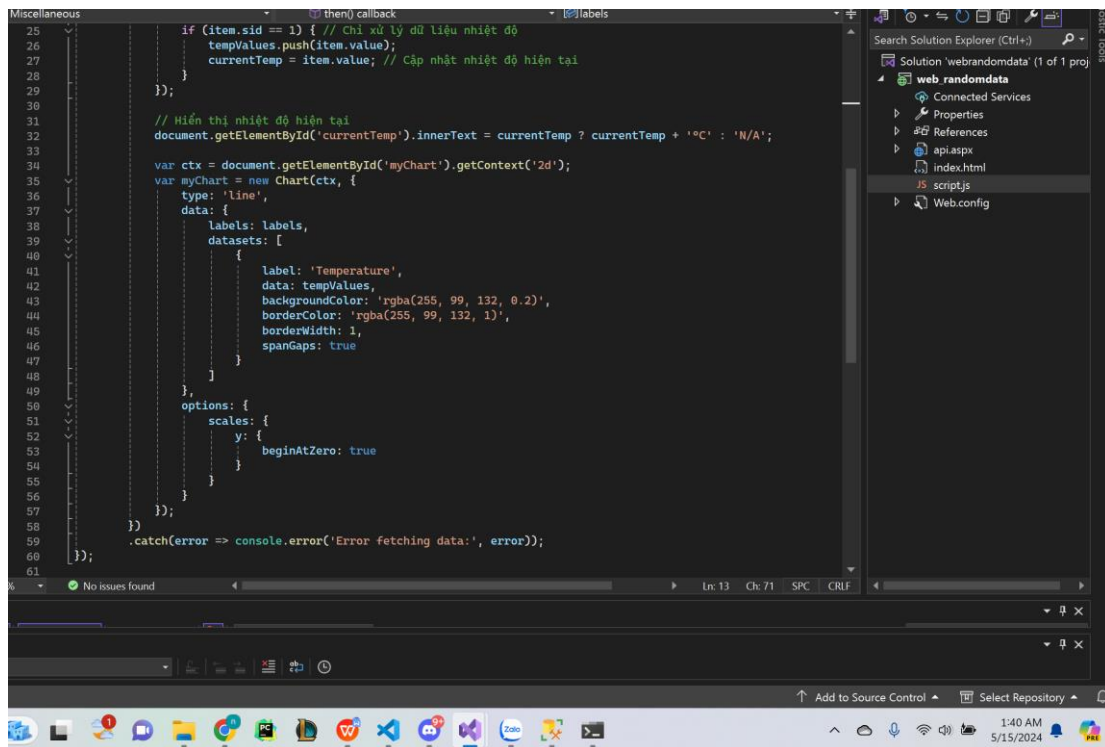
sau đó em clone bài của thầy gửi trên nhóm lớp về rồi sửa lại kết nối sql ,
aspx.cs ,html,java script để đẩy dữ liệu lên web và vẽ biểu đồ



The screenshot shows the Visual Studio IDE with the 'api.aspx.cs' file open. The code is a partial class for an ASP.NET Web API endpoint. It includes the following code:

```
1 using System;
2 using System.Web;
3 using System.Web.UI;
4 using System.Data.SqlClient;
5 using System.Data;
6
7 namespace demo_api_57kmt
8 {
9     2 references
10     public partial class api : System.Web.UI.Page
11     {
12         0 references
13         protected void Page_Load(object sender, EventArgs e)
14         {
15             string connectionString = "Data Source=127.0.0.1,1443;Initial Catalog=anhding;User Id=sa;Password=1;";
16             using (SqlConnection connection = new SqlConnection(connectionString))
17             {
18                 using (SqlCommand command = new SqlCommand("SP_Chart", connection))
19                 {
20                     command.CommandType = CommandType.StoredProcedure;
21                     try
22                     {
23                         connection.Open();
24                         object kq = command.ExecuteScalar();
25                         string json = (string)kq;
26                         this.Response.ContentType = "application/json";
27                         this.Response.Write(json);
28                     }
29                     catch (Exception ex)
30                     {
31                         this.Response.ContentType = "application/json";
32                         this.Response.Write("{\"ok\":0,\"msg\":\"" + ex.Message + "\"}");
33                     }
34                 }
35             }
36         }
37     }
38 }
```

The Solution Explorer on the right shows the project structure for 'web_randomdata', including 'api.aspx', 'index.html', 'script.js', and 'Web.config'.



The screenshot shows the Visual Studio IDE with the 'script.js' file open. The code is a JavaScript script that handles data fetching and chart rendering. It includes the following code:

```
25 if (item.sid == 1) { // Chỉ xử lý dữ liệu nhiệt độ
26     tempValues.push(item.value);
27     currentTemp = item.value; // Cập nhật nhiệt độ hiện tại
28 }
29
30 // Hiển thị nhiệt độ hiện tại
31 document.getElementById('currentTemp').innerText = currentTemp ? currentTemp + '°C' : 'N/A';
32
33 var ctx = document.getElementById('myChart').getContext('2d');
34 var myChart = new Chart(ctx, {
35     type: 'line',
36     data: {
37         labels: labels,
38         datasets: [
39             {
40                 label: 'Temperature',
41                 data: tempValues,
42                 backgroundColor: 'rgba(255, 99, 132, 0.2)',
43                 borderColor: 'rgba(255, 99, 132, 1)',
44                 borderWidth: 1,
45                 spanGaps: true
46             }
47         ]
48     },
49     options: {
50         scales: {
51             y: {
52                 beginAtZero: true
53             }
54         }
55     }
56 });
57
58 .catch(error => console.error('Error fetching data:', error));
59
60
61
```

The Solution Explorer on the right shows the project structure for 'web_randomdata', including 'api.aspx', 'index.html', 'script.js', and 'Web.config'.

Kết quả đạt được

