

**HỌ VÀ TÊN : ĐẶNG THỊ HÀ**  
**MSSV : K205480106015**  
**BÀI TẬP VỀ NHÀ PYTHON**  
**ĐỀ BÀI : HIỂN THỊ NHIỆT ĐỘ**

**Yêu cầu:**

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)

**Các mục cần làm**

- Cài đặt Node-Red
- Tạo CSDL trên SQL
- Kết nối SQL với Node – red
- Hiển thị lên web

## Bài làm

### - Đầu tiên em Download node.js

```
Command Prompt
Microsoft Windows [Version 10.0.19045.4291]
(c) Microsoft Corporation. All rights reserved.

C:\Users\PC>node
Welcome to Node.js v18.18.0.
Type ".help" for more information.
>
(To exit, press Ctrl+C again or Ctrl+D or type .exit)
>

C:\Users\PC>npm -v
9.8.1

C:\Users\PC>npm install -g node-red

added 303 packages in 44s

46 packages are looking for funding
  run `npm fund` for details

npm notice
npm notice New major version of npm available! 9.8.1 -> 10.7.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.7.0
npm notice Run `npm install -g npm@10.7.0` to update!
npm notice
C:\Users\PC>
```

### - Chạy chương trình node – red

```
node-red
C:\Users\PC>node C:\Users\PC\AppData\Roaming\npm\node_modules\node-red\red.js
14 May 13:12:11 - [info]

Welcome to Node-RED
=====

14 May 13:12:11 - [info] Node-RED version: v3.1.9
14 May 13:12:11 - [info] Node.js version: v18.18.0
14 May 13:12:11 - [info] Windows_NT 10.0.19045 x64 LE
14 May 13:12:13 - [info] Loading palette nodes
14 May 13:12:15 - [info] Settings file : C:\Users\PC\.node-red\settings.js
14 May 13:12:15 - [info] Context store : 'default' [module=memory]
14 May 13:12:15 - [info] User directory : C:\Users\PC\.node-red
14 May 13:12:15 - [warn] Projects disabled : editorTheme.projects.enabled=false
14 May 13:12:15 - [info] Flows file : C:\Users\PC\.node-red\flows.json
14 May 13:12:15 - [info] Creating new flow file
14 May 13:12:16 - [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.
-----

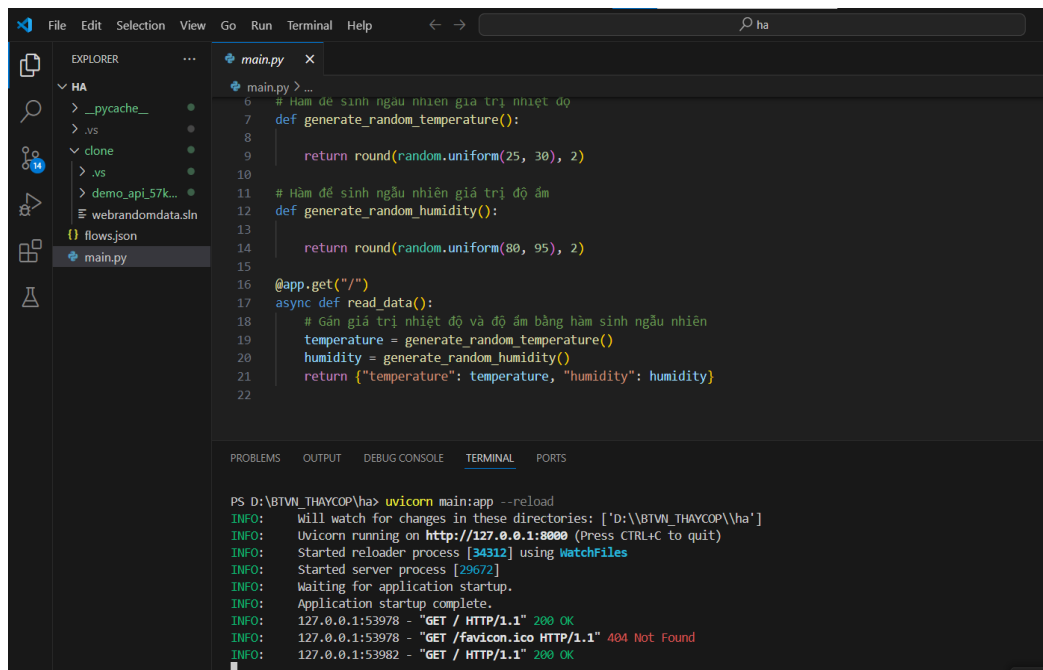
14 May 13:12:16 - [info] Server now running at http://127.0.0.1:1880/
14 May 13:12:16 - [warn] Encrypted credentials not found
14 May 13:12:16 - [info] Starting flows
14 May 13:12:16 - [info] Started flows
```

Bước 1: Em tạo FastAPI trong Python bằng VS code -> Lấy giá trị Temperature được random để sinh ra giá trị ngẫu nhiên trong khoảng từ 25-30.

```
main.py
main.py > read_data

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
12
13 @app.get("/")
14 async def read_data():
15     # Gán giá trị nhiệt độ bằng hàm sinh ngẫu nhiên
16     temperature = generate_random_temperature()
17
18     return {"temperature": temperature}
19
```

- Sau đó chạy lệnh `uvicorn main:app --reload`



```
File Edit Selection View Go Run Terminal Help
EXPLORER
  HA
  > __pycache__
  > .vs
  > clone
  > .vs
  > demo_api_57k...
  webbrandomdata.sln
  flows.json
  main.py
main.py
main.py > ...
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}
22
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\BTVN_THAYCOP\ha> uvicorn main:app --reload
INFO: Will watch for changes in these directories: ['D:\BTVN_THAYCOP\ha']
INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO: Started reloader process [34312] using WatchFiles
INFO: Started server process [29672]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: 127.0.0.1:53978 - "GET / HTTP/1.1" 200 OK
INFO: 127.0.0.1:53978 - "GET /favicon.ico HTTP/1.1" 404 Not Found
INFO: 127.0.0.1:53982 - "GET / HTTP/1.1" 200 OK
```

- Sau đó sẽ trả về chuỗi json

- Các giá trị random trả về random trong khoảng đặt trước đó

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

INFO:      127.0.0.1:54045 - "GET / HTTP/1.1" 200 OK
INFO:      127.0.0.1:54056 - "GET / HTTP/1.1" 200 OK
INFO:      127.0.0.1:54062 - "GET / HTTP/1.1" 200 OK
INFO:      127.0.0.1:54064 - "GET / HTTP/1.1" 200 OK
INFO:      127.0.0.1:54068 - "GET / HTTP/1.1" 200 OK
INFO:      127.0.0.1:54070 - "GET / HTTP/1.1" 200 OK
INFO:      127.0.0.1:54072 - "GET / HTTP/1.1" 200 OK
INFO:      127.0.0.1:54076 - "GET / HTTP/1.1" 200 OK
INFO:      127.0.0.1:54081 - "GET / HTTP/1.1" 200 OK
INFO:      127.0.0.1:54085 - "GET / HTTP/1.1" 200 OK
INFO:      127.0.0.1:54103 - "GET / HTTP/1.1" 200 OK
INFO:      127.0.0.1:54124 - "GET / HTTP/1.1" 200 OK
INFO:      127.0.0.1:54126 - "GET / HTTP/1.1" 200 OK
INFO:      127.0.0.1:54128 - "GET / HTTP/1.1" 200 OK
INFO:      127.0.0.1:54132 - "GET / HTTP/1.1" 200 OK
INFO:      127.0.0.1:54135 - "GET / HTTP/1.1" 200 OK
INFO:      127.0.0.1:54139 - "GET / HTTP/1.1" 200 OK
```

Bước 2 : Sử dụng node – red lấy dữ liệu từ địa chỉ local của FastAPI

- Chạy node – red

```
node-red
Microsoft Windows [Version 10.0.19045.4291]
(c) Microsoft Corporation. All rights reserved.

C:\Users\PC>node-red
14 May 23:59:30 - [info]

Welcome to Node-RED
=====

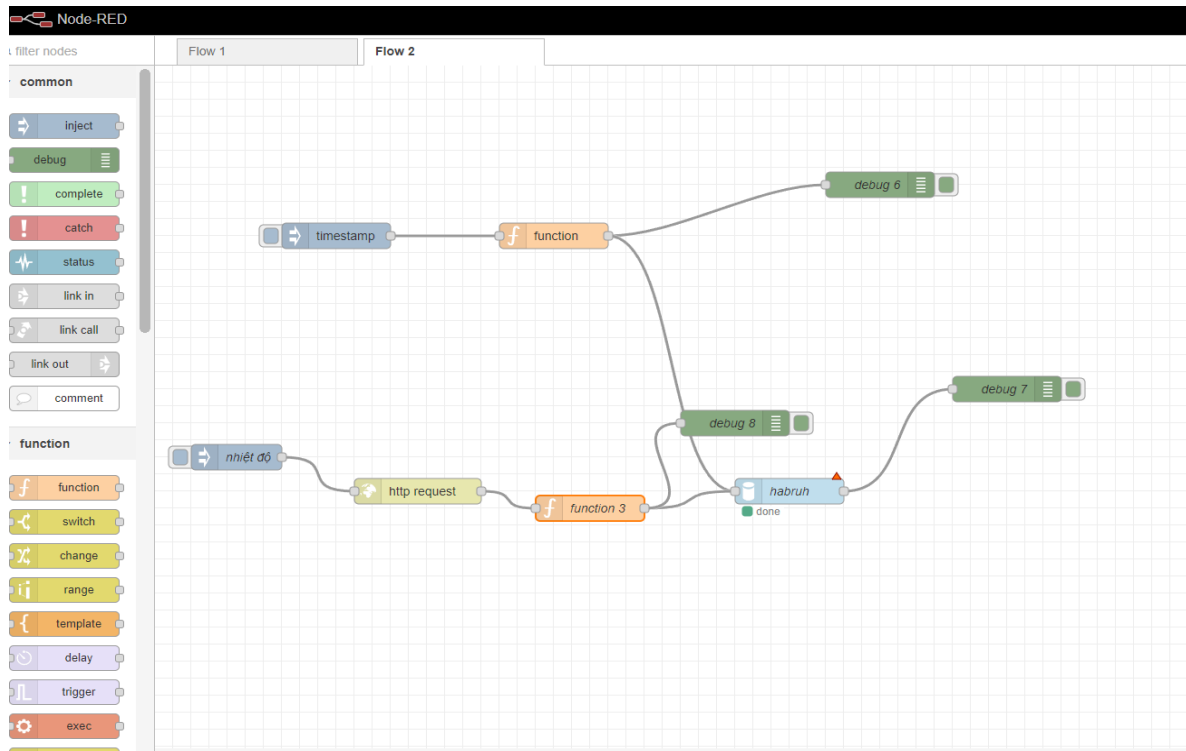
14 May 23:59:30 - [info] Node-RED version: v3.1.9
14 May 23:59:30 - [info] Node.js version: v18.18.0
14 May 23:59:30 - [info] Windows_NT 10.0.19045 x64 LE
14 May 23:59:34 - [info] Loading palette nodes
14 May 23:59:41 - [info] Settings file : C:\Users\PC\.node-red\settings.js
14 May 23:59:41 - [info] Context store : 'default' [module=memory]
14 May 23:59:41 - [info] User directory : \Users\PC\.node-red
14 May 23:59:41 - [warn] Projects disabled : editorTheme.projects.enabled=false
14 May 23:59:41 - [info] Flows file : \Users\PC\.node-red\flows.json
14 May 23:59:41 - [info] Server now running at http://127.0.0.1:1880/
14 May 23:59:41 - [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
```

## - Tạo các function



## - Kết nối SQL với Node Red

The screenshot shows the **Edit MSSQL node** configuration window. The window has a title bar with **Delete**, **Cancel**, and **Update** buttons. The **Properties** tab is selected, showing the following fields:

- Name**: dangha
- Server**: 127.0.0.1
- Port**: 1433
- Username**: sa
- Password**: .....
- Domain**:
- Database**: dangha
- TDS Version**: 7\_4 (SQL Server 2012 ~ 2022)
- Use Encryption?**: ☒
- Trust Certificate?**: ☒
- Assume UTC?**: ☐
- Connect Timeout**:

Below the fields, there are three informational messages:

- SQL Databases hosted on Azure will need this checked.
- If unchecked, SQL Server will try to validate the server SSL certificate and will terminate the connection if validation fails.
- Pass time values in UTC or local time.
- The number of milliseconds before the attempt to connect is considered failed.

At the bottom, there are buttons for **Save**, **Enabled**, and **1**, and a dropdown menu set to **On all flows**.

Bước

3:

CSDL

trên

SQL

The screenshot displays two SQL queries in SQL Server Enterprise Manager. The left query, 'SQLQuery2.sql - 12...33.dangha (sa (53))', is a SELECT TOP (1000) statement from the [dangha].[dbo].[history] table, selecting columns [id], [sid], [value], and [time]. The right query, 'SQLQuery3.sql - 12...33.dangha (sa (66))', is a SELECT TOP (1000) statement from the [dangha].[dbo].[sensor] table, selecting columns [name], [unit], [value], and [time]. Below the queries, the results of the first query are shown in a table grid. The table has four columns: id, sid, value, and time. The first row shows id=1, sid=1, value=3, and time=2024-04-03 ...

id	sid	value	time
1	1	3	2024-04-03 ...
7	1	28	2024-05-14 ...
8	1	27	2024-05-14 ...
9	2	85	2024-05-14 ...
10	2	93	2024-05-14 ...
11	2	88	2024-05-14 ...
12	2	91	2024-05-14 ...
13	2	87	2024-05-14 ...
14	1	28	2024-05-14 ...
15	2	91	2024-05-14 ...
16	1	30	2024-05-14 ...
17	2	84	2024-05-14 ...
18	1	29	2024-05-14 ...
19	2	95	2024-05-14 ...
20	2	95	2024-05-14 ...
21	1	25	2024-05-14 ...
22	2	83	2024-05-14 ...
23	1	28	2024-05-14 ...
24	2	86	2024-05-14 ...
25	1	29	2024-05-14 ...
26	2	90	2024-05-14 ...
27	1	27	2024-05-14 ...
28	2	93	2024-05-14 ...
29	1	26	2024-05-14 ...
30	2	92	2024-05-14 ...
31	2	87	2024-05-14 ...
32	1	25	2024-05-14 ...
33	2	92	2024-05-14 ...
34	1	29	2024-05-14 ...
35	2	86	2024-05-14 ...

- Tạo store procedures trả về dữ liệu dạng json đẩy lên web

```
SQLQuery4.sql - 12...33.dangha (sa (66)) - DESKTOP-T80TUON\S...gha - dbo.history
USE [dangha]
GO
/***** Object: StoredProcedure [dbo].[SP_Chart]    Script Date: 5/15/2024 12:50:49 AM *****/
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 += FORMATTMESSAGE(N'{"id": "%d", "sid": "%d", "value": "%s", "time": "%s"}',
                                   [id], [sid], CONVERT(nvarchar(50), [value]), CONVERT(nvarchar(50), [time]))
    FROM history;

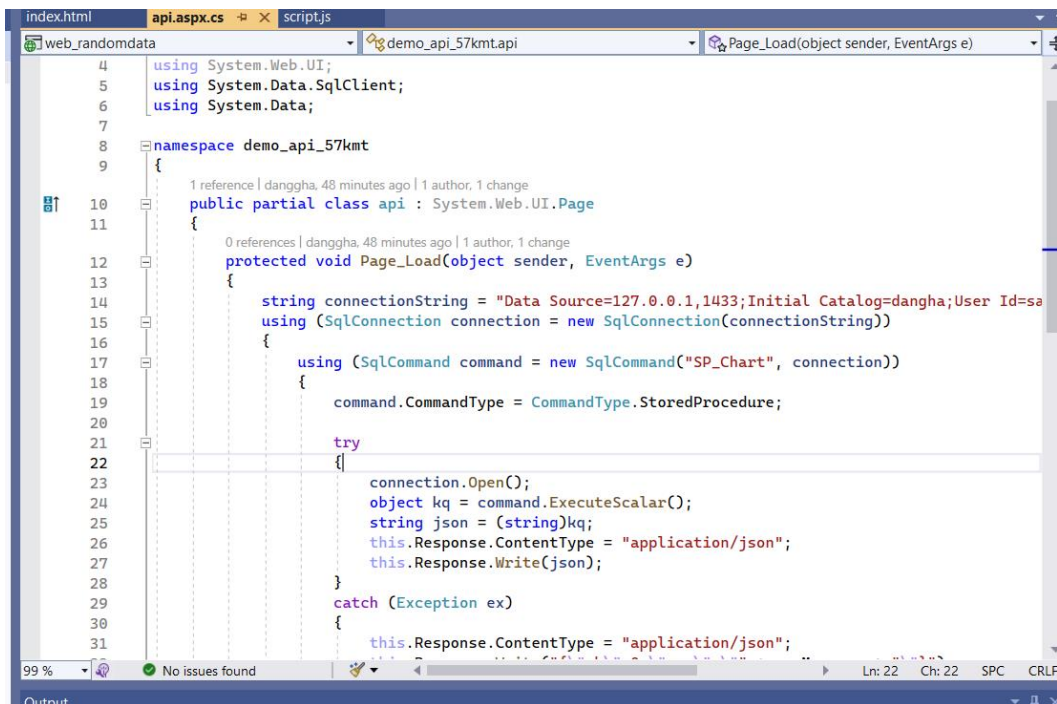
    IF RIGHT(@json, 1) = ','
    BEGIN
        SET @json = LEFT(@json, LEN(@json) - 1);
    END

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

    SELECT @json AS json;
END
```

#### Bước 4: Sử dụng asp dot-net để lấy dữ liệu và vẽ biểu đồ

- Tạo 1 chuỗi kết nối đến Database trên SQL

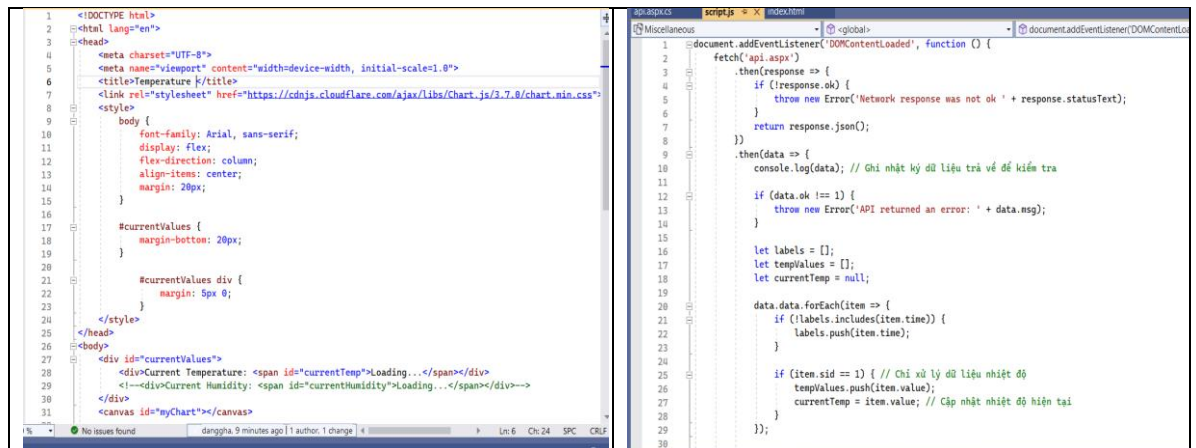


```
using System.Web.UI;
using System.Data.SqlClient;
using System.Data;

namespace demo_api_57kmt
{
    public partial class api : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            string connectionString = "Data Source=127.0.0.1,1433;Initial Catalog=danggha;User Id=sa";
            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";
                    }
                }
            }
        }
    }
}
```

- file index.html và cript.js để vẽ biểu đồ



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Temperature h</title>
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/3.7.0/chart.min.css">
  <style>
    body {
      font-family: Arial, sans-serif;
      display: flex;
      flex-direction: column;
      align-items: center;
      margin: 20px;
    }
    #currentValues {
      margin-bottom: 20px;
    }
    #currentValues div {
      margin: 5px 0;
    }
  </style>
</head>
<body>
  <div id="currentValues">
    <div>Current Temperature: <span id="currentTemp">Loading...</span></div>
    <div>Current Humidity: <span id="currentHumidity">Loading...</span></div>
  </div>
  <canvas id="myChart"></canvas>
</body>
</html>
```

```
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 currentTemp = null;

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

        if (item.sid === 1) { // Chỉ xử lý dữ liệu nhiệt độ
          tempValues.push(item.value);
          currentTemp = item.value; // Cập nhật nhiệt độ hiện tại
        }
      });
    });
});
```

- Kết quả

