

Họ tên: Dương Thị Bích Nguyệt

MSSV: K205480106021

## **ĐỀ BÀI : HIỂN THỊ NHIỆT ĐỘ**

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. 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)

- Mục đích chính đã làm:

- Dùng FastApi của python
- Cài đặt Node-Red trên windows
- Tạo cơ sở dữ liệu trên SQL sever
- Kết nối SQL với Node Red
- Hiển thị lên web

- Chi tiết các bước đã làm

- Dùng FastApi của python:  
Đầu tiên sẽ phải cài FastApi vào python bằng câu lệnh :

pip install fastapi

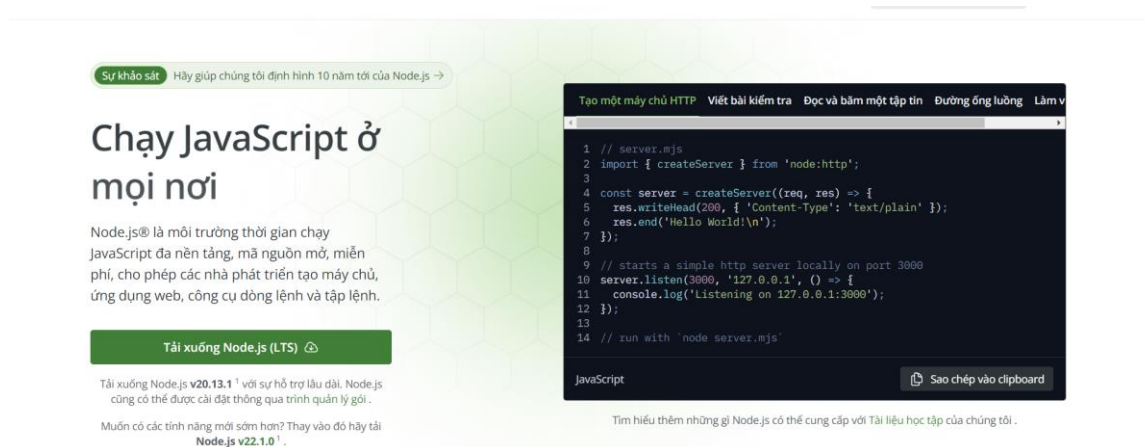
pip install uvicorn

Chạy ứng dụng FastAPI của bạn bằng cách sử dụng uvicorn

Kiểm tra bằng cách điều hướng đến <http://127.0.0.1:8000/docs>

- Cài đặt Node-Red

Đầu tiên phải tải Node JS về windows

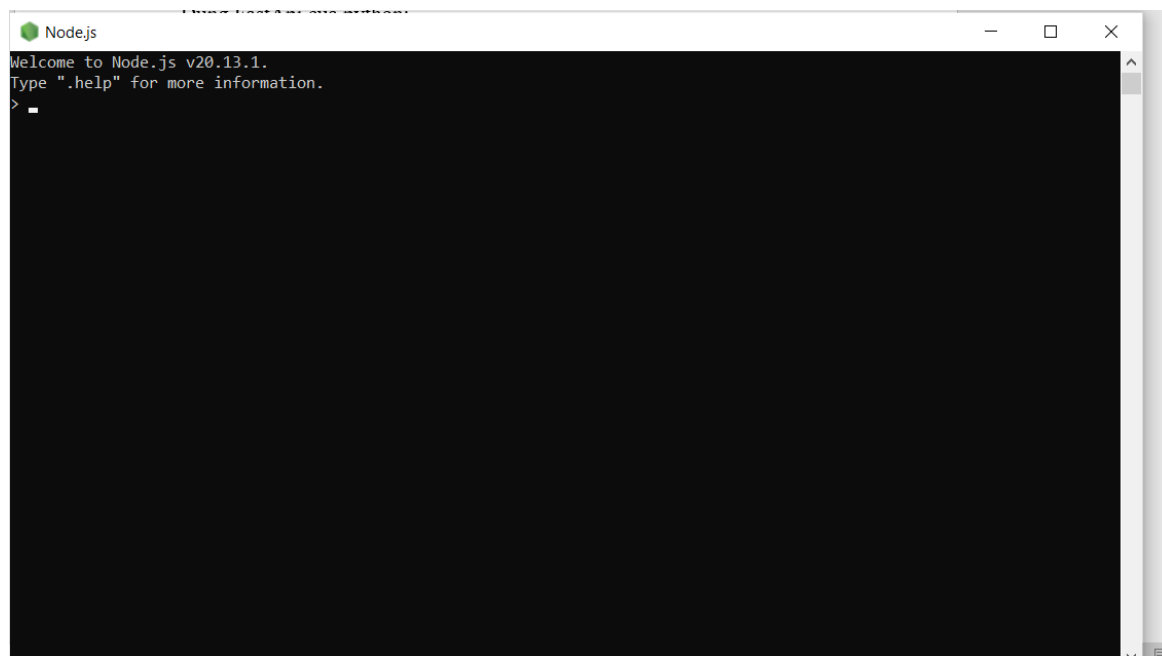


The screenshot shows the Node.js website with a green header and a large heading "Chạy JavaScript ở mọi nơi". Below the heading, there is a paragraph about Node.js being a multi-platform, open-source, and free JavaScript runtime. A green button labeled "Tải xuống Node.js (LTS)" is prominent. To the right, there is a code editor showing a simple HTTP server script. The script is as follows:

```
1 // server.js
2 import { createServer } from 'node:http';
3
4 const server = createServer((req, res) => {
5   res.writeHead(200, { 'Content-Type': 'text/plain' });
6   res.end('Hello World!\n');
7 });
8
9 // starts a simple http server locally on port 3000
10 server.listen(3000, '127.0.0.1', () => {
11   console.log('Listening on 127.0.0.1:3000');
12 });
13
14 // run with 'node server.js'
```

Below the code editor, there is a link to the Node.js documentation and a note about the LTS version.

Tiếp theo đó mở Node JS để tập tục cài đặt



Sau đó mở cmd của máy để cài Node Red

```
node-red
Microsoft Windows [Version 10.0.19042.1165]
(c) Microsoft Corporation. All rights reserved.

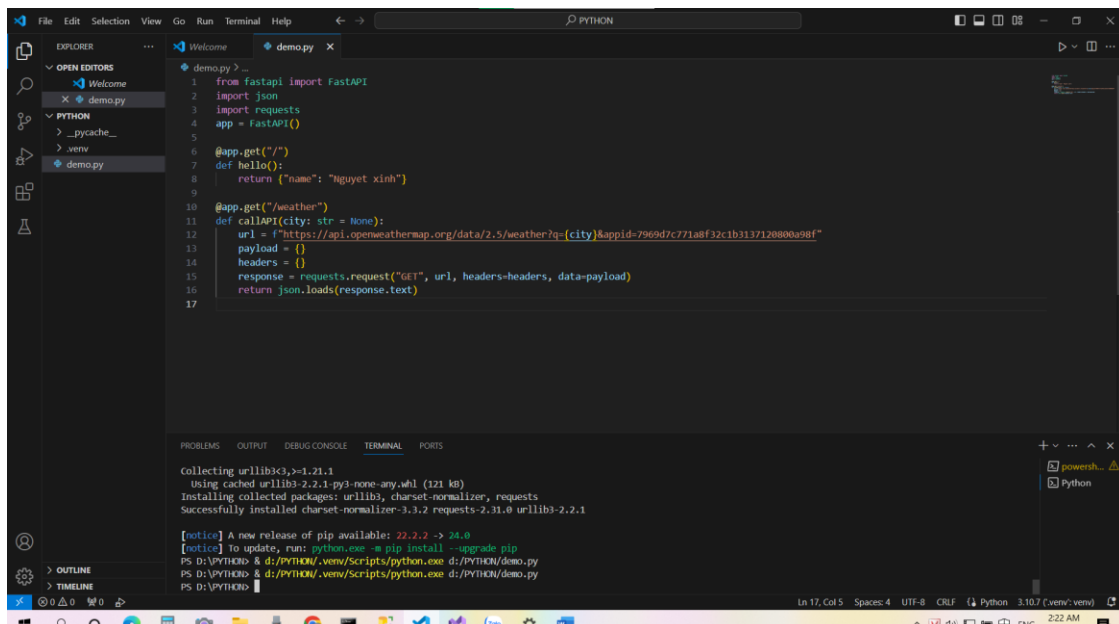
C:\Users\A D M I N>node-red
14 May 10:22:07 - [info]

Welcome to Node-RED
=====
14 May 10:22:07 - [info] Node-RED version: v3.1.9
14 May 10:22:07 - [info] Node.js version: v20.13.1
14 May 10:22:07 - [info] Windows_NT 10.0.19042 x64 LE
14 May 10:22:07 - [info] Loading palette nodes
14 May 10:22:08 - [warn] Missing node modules:
14 May 10:22:08 - [warn]   - node-red-contrib-mssql-pool (0.0.6): MSSQL-CN, MSSQL
14 May 10:22:08 - [info] Removing modules from config
14 May 10:22:08 - [info] Settings file : C:\Users\A D M I N\.node-red\settings.js
14 May 10:22:08 - [info] Context store : 'default' [module=memory]
14 May 10:22:08 - [info] User directory : \Users\A D M I N\.node-red
14 May 10:22:08 - [warn] Projects disabled : editorTheme.projects.enabled=false
14 May 10:22:08 - [info] Flows file : \Users\A D M I N\.node-red\flows.json
14 May 10:22:08 - [info] Server now running at http://127.0.0.1:1880/
14 May 10:22:08 - [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.
```

## Bước 1: Tạo FastAPI trong Python bằng VS code



The screenshot shows the Visual Studio Code editor with a Python file named `demo.py` open. The file contains the following code:

```
1 from fastapi import FastAPI
2 import json
3 import requests
4 app = FastAPI()
5
6 @app.get("/")
7 def hello():
8     return {"name": "Nguyet kinh"}
9
10 @app.get("/weather")
11 def callAPI(city: str = None):
12     url = f"https://api.openweathermap.org/data/2.5/weather?q={city}&appid=7969d7c771a8f32c1b3137128800a98f"
13     payload = {}
14     headers = {}
15     response = requests.request("GET", url, headers=headers, data=payload)
16     return json.loads(response.text)
17
```

The terminal window at the bottom shows the following output:

```
Collecting urllib3<3.2.1, >=1.21.1
Using cached urllib3-2.2.1-py3-none-any.whl (121 kB)
Installing collected packages: urllib3, charset-normalizer, requests
Successfully installed charset-normalizer-3.3.2 requests-2.31.0 urllib3-2.2.1

[notice] A new release of pip available: 22.2.2 -> 24.0
[notice] To update, run: python.exe -m pip install --upgrade pip
PS D:\PYTHON> & d:\PYTHON\venv\Scripts\python.exe d:\PYTHON\demo.py
PS D:\PYTHON>
```

- Sau đó chạy lệnh `uvicorn main:app—reload` để trả về chuỗi json

```
17
[notice] To update, run: python.exe -m pip install --upper, requests
rade pip
-2.31.0 urllib3-2.2.1
PS D:\PYTHON> & d:/PYTHON/.venv/Scripts/python.exe d:/PY
THON/demo.py
PS D:\PYTHON> & d:/PYTHON/.venv/Scripts/python.exe d:/PYrade pip
THON/demo.py
PS D:\PYTHON> uvicorn main:app --reload
INFO: Will watch for changes in these directories: ['D:\PYTHON']
INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO: Started reloader process [17272] using WatchFiles
```

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

- Chạy node – red

```
node-red
Microsoft Windows [Version 10.0.19042.1165]
(c) Microsoft Corporation. All rights reserved.

C:\Users\A D M I N>node-red
14 May 23:38:37 - [info]

Welcome to Node-RED
=====

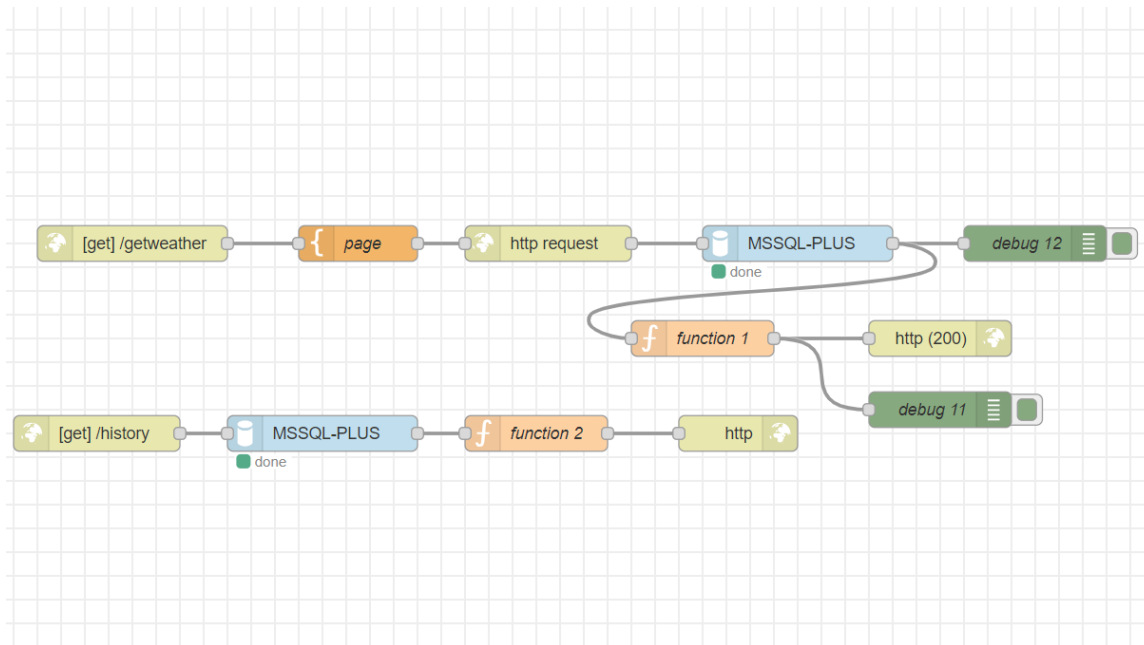
14 May 23:38:37 - [info] Node-RED version: v3.1.9
14 May 23:38:37 - [info] Node.js version: v20.13.1
14 May 23:38:37 - [info] Windows_NT 10.0.19042 x64 LE
14 May 23:38:37 - [info] Loading palette nodes
14 May 23:38:39 - [info] Settings file : C:\Users\A D M I N\.node-red\settings.js
14 May 23:38:39 - [info] Context store : 'default' [module=memory]
14 May 23:38:39 - [info] User directory : \Users\A D M I N\.node-red
14 May 23:38:39 - [warn] Projects disabled : editorTheme.projects.enabled=false
14 May 23:38:39 - [info] Flows file : \Users\A D M I N\.node-red\flows.json
14 May 23:38:39 - [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
using your chosen key the next time you deploy a change.
```

Tạo các function



- Kết nối SQL với Node Red

Edit MSSQL node > **Edit MSSQL-CN node**

Delete Cancel Update

**Properties**

Name Connection Name

Server 127.0.0.1

Port 1443

Username sa

Password .....

Domain

Database demo

TDS Version 7\_4 (SQL Server 2012 ~ 2022) ▼

Use Encryption? ☒

SQL Databases hosted on Azure will need this checked.

Trust Certificate? ☒

Microsoft SQL Server will not validate the server certificate.

Enabled 1 On all flows

### Bước 3: Cấu trúc dữ liệu

```
SELECT TOP (1000) [ID]
      , [IDThanhPho]
      , [NhietDo]
      , [Time]
FROM [demo].[dbo].[history]
```

```
SELECT TOP (1000) [ID]
      , [ThanhPho]
      , [NhietDo]
FROM [demo].[dbo].[NDo]
```

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



```
USE [demo]
GO
/***** Object:  StoredProcedure [dbo].[SP_Chart]    Script Date: 5/15/2024 1:26:40 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 += FORMATMESSAGE(N'{"id": "%d", "sid": "%d", "value": "%s", "time": "%s"}',
                                   [id], [IDThanhPho], CONVERT(nvarchar(50), [NhietDo]), CONVERT(nvarchar(5), [time], 108))
    FROM history order by [time]

    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

```

1  using System;
2  using System.Web;
3  using System.Web.UI;
4  using System.Data.SqlClient;
5  using System.Data;
6
7
8  namespace demo_api_57kmt
9  {
10     public partial class api : System.Web.UI.Page
11     {
12         protected void Page_Load(object sender, EventArgs e)
13         {
14             string connectionString = "Data Source=127.0.0.1,1443;Initial Catalog=chuchua;User Id=sa;Password=123;";
15             using (SqlConnection connection = new SqlConnection(connectionString))
16             {
17                 using (SqlCommand command = new SqlCommand("SP_Chart", connection))
18                 {
19                     command.CommandType = CommandType.StoredProcedure;
20
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 }

```

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

```

1  $(document).ready(function () {
2
3      $.get("http://127.0.0.1:1880/history", function (data) {
4          var a = [{"Time", "Temp"}];
5          for (var i of data.data) {
6              var time = i.time;
7              var value = parseInt(i.value);
8              a.push([time, value]);
9          }
10         google.charts.load('current', { 'packages': ['corechart'] });
11         google.charts.setOnLoadCallback(drawChart);
12
13         function drawChart() {
14             var data = google.visualization.arrayToDataTable(a);
15
16             var options = {
17                 title: 'Lịch sử',
18                 hAxis: { title: 'Hours', titleTextStyle: { color: '#333' } },
19                 vAxis: { minValue: 0 }
20             };
21
22             var chart = new google.visualization.AreaChart(document.getElementById('chart_div'));
23             chart.draw(data, options);
24         }
25     });
26 }

```



```
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>Temperature and Humidity Chart</title>
7 <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.7.1/jquery.min.js" integrity="sha512-v2CJ7UaYy4JwqLDIr2
8 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/3.7.0/chart.min.css">
9 <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
10 <style>
11 body {
12     font-family: Arial, sans-serif;
13     display: flex;
14     flex-direction: column;
15     align-items: center;
16     margin: 20px;
17 }
18
19 #currentValues {
20     margin-bottom: 20px;
21 }
22
23 #currentValues div {
24     margin: 5px 0;
25 }
26 </style>
27 </head>
28 <body>
29 <div id="chart_div" style="width: 100%; height: 500px;"></div>
30 <script src="script.js"></script>
31 </body>
32 </html>
33
```

- Kết quả

-

