Họ tên: Vũ Ngọc Trang

Lớp K56KMT.01

Mssv: K205480106035

Báo cáo bài tập

Bài tập python ~ điểm thi học kỳ.

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 tuỳ ý thư viện thích hợp)

Yêu cầu:

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

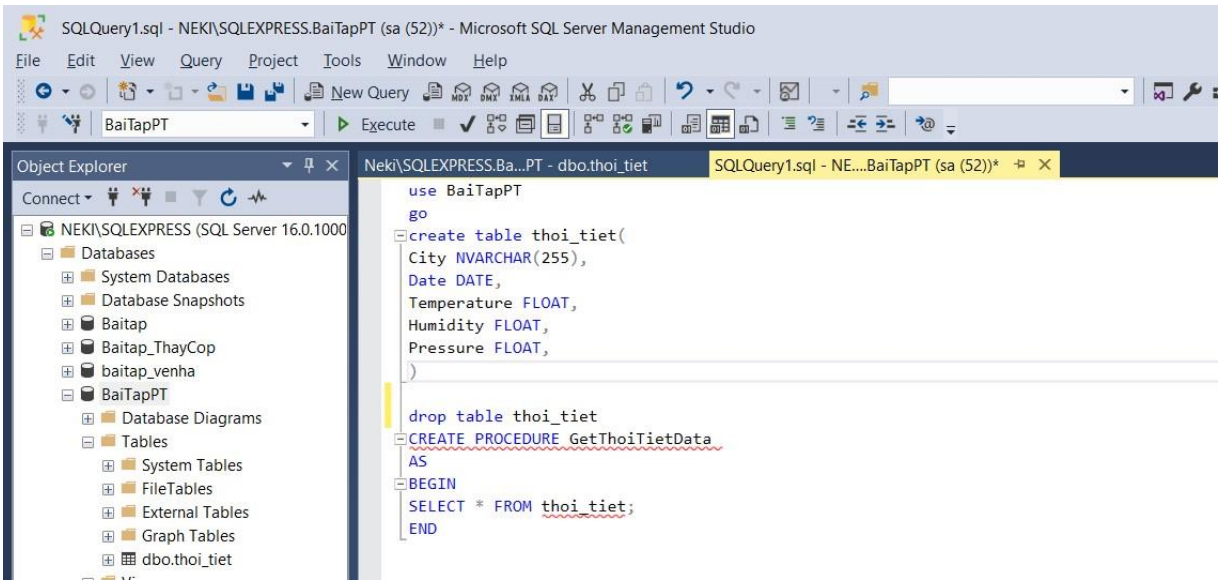
2. Mô tả các bước cài đặt+ snap màn hình minh hoạ.

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?....

Bước đầu tiên: Tải các môi trường để lập trình

- **Cài sql (để lưu dữ liệu database)**
  Ở đây thì em tạo 2 bảng



1 bảng và lưu các trường thông tin và 1 bảng sp_db

- Tải visual studio code (để viết code)

  ở đây em không dùng visual code, em dùng **Pycharm**(thao tác dễ dàng)

  link tải: [Download PyCharm: The Python IDE for data science and web development by JetBrains]

  cần tải các thư viện như:

  + FastAPI và 1 số thư viện liên quan mà mình code sẽ có phần thông tin gợi ý tải…

- Tải node-red (là công cụ kéo-thả để kết nối và tự động hóa IoT, API và dịch vụ web)

  Cách tải node-red. Ta cần cài node.js

  Link tải: [Node.js — Download Node.js® (nodejs.org)]

Sau khi tải xong ta copy ứng dụng vào note path của hệ thống

➔ Sau đó mở cmd or terminal cài node-red thì dung lệnh sau:

install -g --unsafe-perm node-red

- cài xong thì ta dụng lệnh : node-red để tạo địa chỉ : http://127.0.0.1:1880
- copy đường dẫn đó vào duyệt trình web.

➔ Ta đã cài và vào được môi trường sử dụng node-red

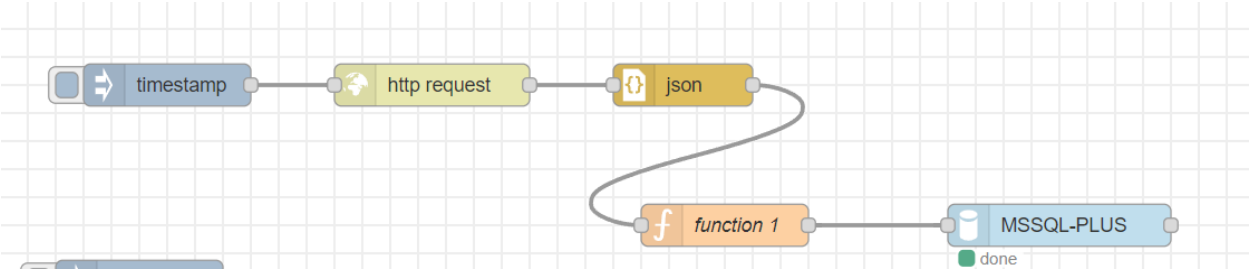
Bắt đầu vào làm việc của bài:

Ta lên OpenWeather (để tạo 1 tài khoản lấy key)

➔ File code py fastAPI để lấy dữ liệu.

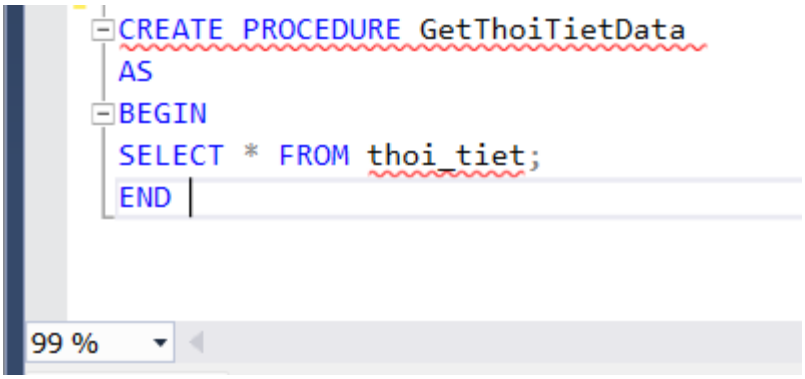
```python
from fastapi import FastAPI
from fastapi.responses import JSONResponse

app = FastAPI()

new *
@app.get("/weather") #endopoint
async def get_weather_data():
    # Đoạn mã lấy dữ liệu thời tiết
    import requests
    from datetime import datetime, timedelta

    new *
    def get_weather_data(city, api_key, date):
        url = f"http://api.openweathermap.org/data/2.5/forecast?q={city}&appid={api_key}&units=metric"
        response = requests.get(url)
        data = response.json()
        return data

    new *
    def extract_weather_info(data):
        weather_info = {}
        for item in data['list']:
```

Để lấy thông tin của openweather -> sử dụng fastAPI để gửi lên: http://127.0.0.1:8810/weather -> sau đó ta sử dụng node-red để lấy dữ liệu này lưu về database.



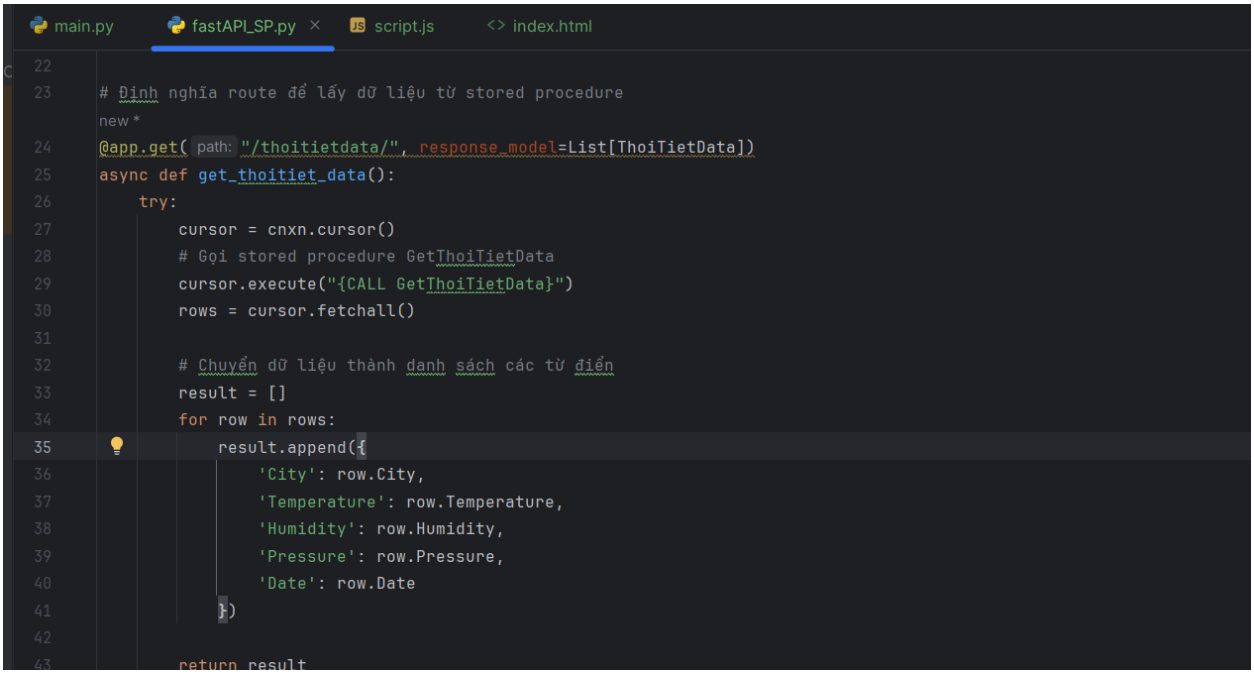
- Chú ý: cần install MSSQL-PLUS (để kết nối SQL và Node-red)

- Ở đây ta tạo gồm có: inject, http request, json, function, MSSQL-Plus
- Em hiểu và làm. Mình tạo 1 file py fastAPI có đường dẫn là http://127.0.0.1:8000/weather rồi lưu vào http request rồi gắn với json để chuyển sang thành file json từ json đến function để chuyển thành từ json sang thành mảng các đối tượng rồi lưu vào SQL.



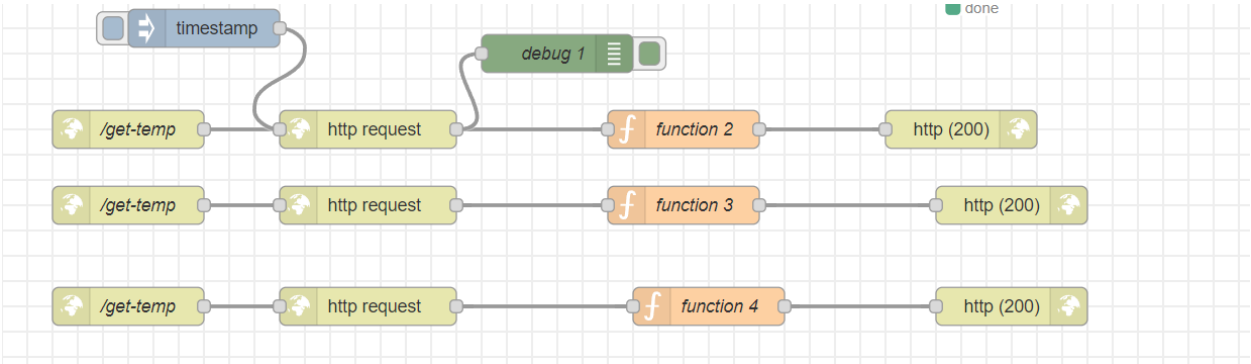
➔ Sau đó từ db lấy dữ liệu ra endpoint bằng py api bằng cách gọi sp_getthoitietdata.

- Ta cần phải tạo thêm 1 file.py(trong file này gồm có fastAPI và pyodbc)



Tạo xong ta có đường dẫn: 127.0.0.1:8810/thoitietdata/ nó sẽ đẩy hết thông tin của bảng đó ra.

➔ Sau khi gọi api thì sử dụng http request lấy thông tin và đưa vào endpoint.



ở đây có 3 hàng tượng trưng cho 3 trường. “Temperature, Humidity, Pressure”

đầy tiên ta gọi và đặt endpoint get cho từng trường sau chuyển file json sang hướng đối tượng đó sử dụng địa chỉ: 127.0.0.1:8810/thoitietdata/ để hiện thị cho thông tin của mỗi trường đã được lọc.

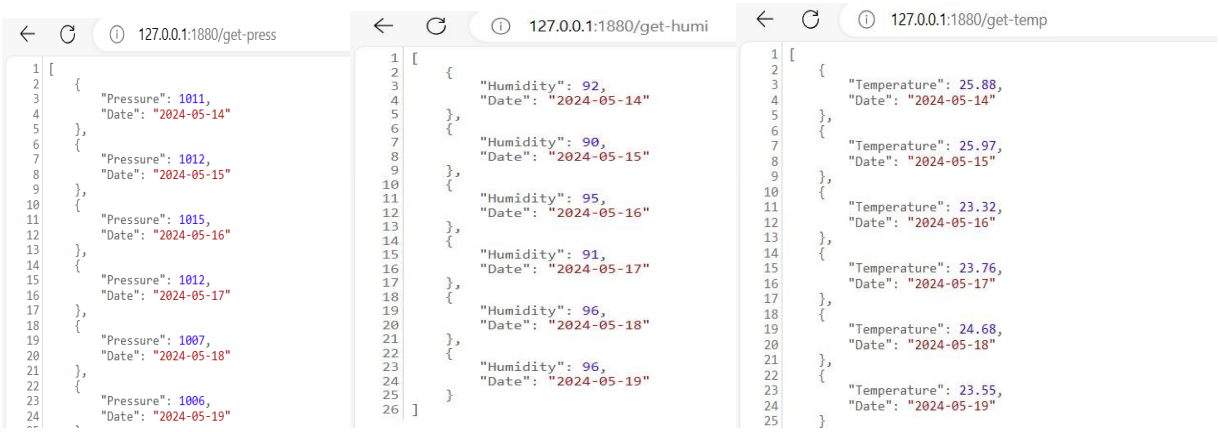
➔ Ta tạo được địa chỉ của 3 tường này:

http://127.0.0.1:1880/get-temp

http://127.0.0.1:1880/get-humi

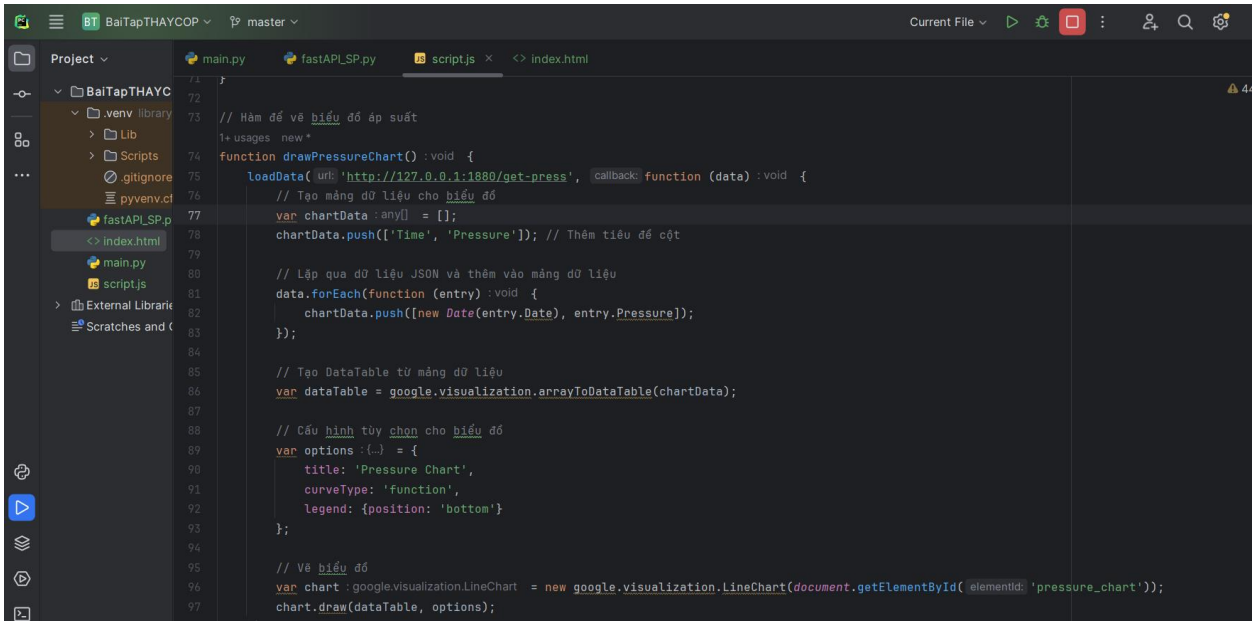
http://127.0.0.1:1880/get-press

- Từ đó ta có 3 file js để vẽ biểu đồ mà ta muốn.



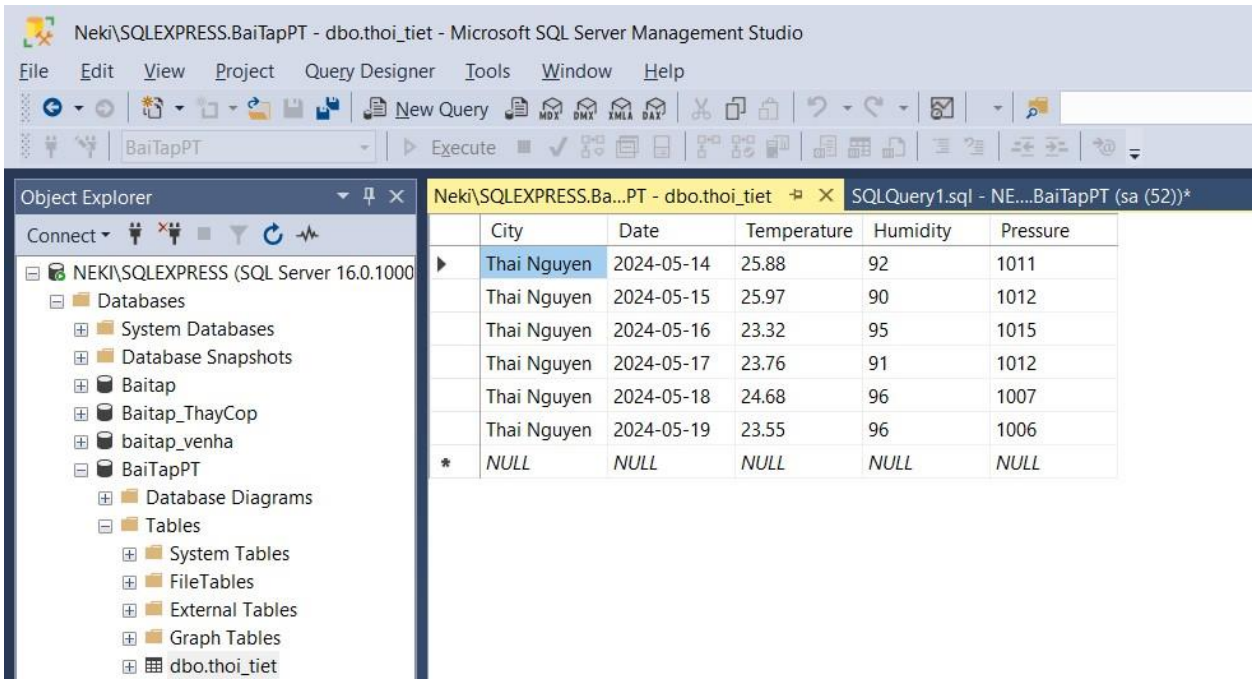
➔ Ta đã được 3 file js và cuối cùng thì vẽ biểu đồ.

Sau khi muốn tại file.js thì chạy node-red và fastapi để lấy dữ liệu vẽ biểu đồ.

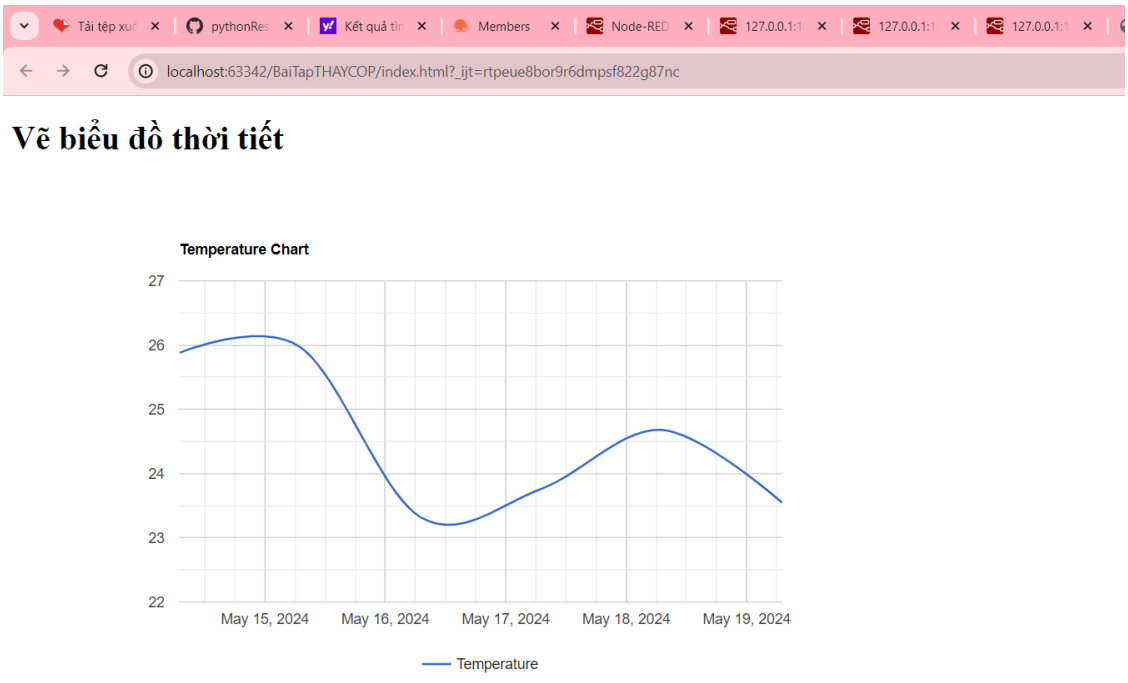


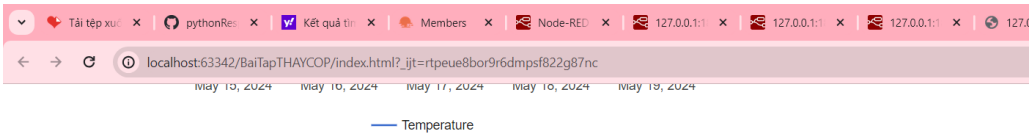
➔ Ta có kết quả:

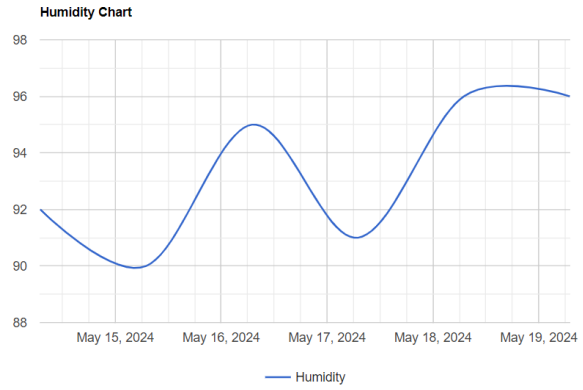
1. SQL:

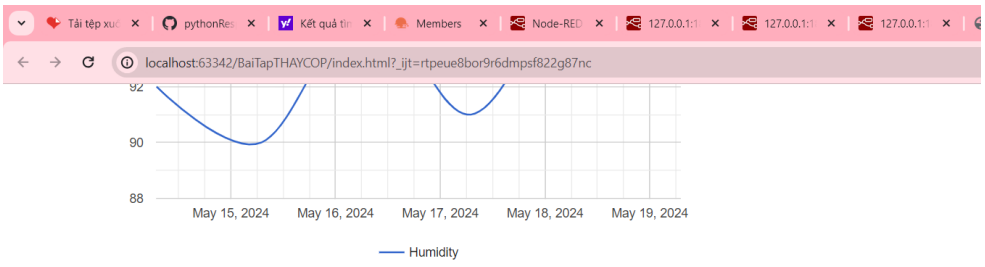


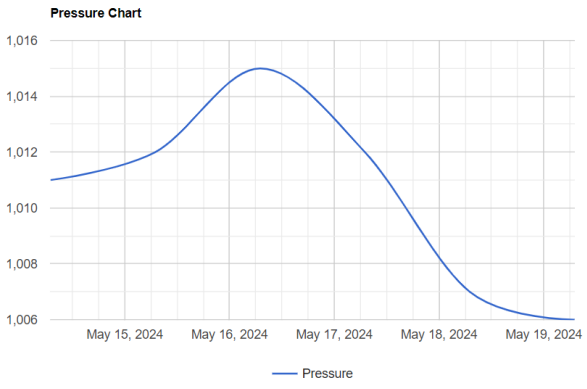
2. Biểu đồ:




localhost:63342/BaiTapTHAYCOP/index.html?_ijt=rtpeue8bor9r6dmpsf822g87nc
Temperature


Humidity Chart
98
96
94
92
90
88
May 15, 2024
May 16, 2024
May 17, 2024
May 18, 2024
May 19, 2024
Humidity


localhost:63342/BaiTapTHAYCOP/index.html?_ijt=rtpeue8bor9r6dmpsf822g87nc
90
88
May 15, 2024
May 16, 2024
May 17, 2024
May 18, 2024
May 19, 2024
Humidity


Pressure Chart
1,016
1,014
1,012
1,010
1,008
1,006
May 15, 2024
May 16, 2024
May 17, 2024
May 18, 2024
May 19, 2024
Pressure

3.Code

- code gọi dữ liệu openweather về để lưu vào sql

```
from fastapi import FastAPI
from fastapi.responses import JSONResponse

app = FastAPI()

@app.get("/weather") #endopoint
async def get_weather_data():
    # Đoạn mã lấy dữ liệu thời tiết
    import requests
    from datetime import datetime, timedelta

    def get_weather_data(city, api_key, date):
        url =
f"http://api.openweathermap.org/data/2.5/forecast?q={city}&appid={api_key}&un
its=metric"
        response = requests.get(url)
        data = response.json()
        return data

    def extract_weather_info(data):
        weather_info = {}
        for item in data['list']:
            # Lấy chỉ ngày từ chuỗi datetime
            date = item['dt_txt'].split()[0]
            if date not in weather_info:
                temperature = item['main']['temp']
                humidity = item['main']['humidity']
                pressure = item['main']['pressure']
                rain = item.get('rain', {}).get('3h', 0)  # Lượng mưa trong 3
giờ (nếu có)
                wind_speed = item['wind']['speed']
                weather_info[date] = {'Date': date, 'Temperature':
temperature, 'Humidity': humidity,
                                      'Pressure': pressure, 'City': city}
        return list(weather_info.values())

    def get_past_date(days):
        today = datetime.now()
        past_date = today - timedelta(days=days)
        return past_date.strftime("%Y-%m-%d")

    city = "Thai Nguyen"
    api_key = "093bc10f5336f52a5c3c761e7b731280"
    past_date = get_past_date(7)  # Lấy dữ liệu 1 tuần trước

    weather_data = get_weather_data(city, api_key, past_date)
    weather_info = extract_weather_info(weather_data)

    return JSONResponse(content=weather_info)
```

- Code fastAPI.py

```
- from fastapi import FastAPI, HTTPException
  from typing import List
  from pydantic import BaseModel
  import pyodbc

  app = FastAPI()

  # Thiết lập thông tin kết nối tới cơ sở dữ liệu SQL Server
  server = 'Neki'
  database = 'BaiTapPT'
  username = 'sa'
  password = '2402'
  cnxn = pyodbc.connect('DRIVER={SQL
  Server};SERVER='+server+';DATABASE='+database+';UID='+username+';PWD='+
  password)

  # Định nghĩa model cho dữ liệu thời tiết
  class ThoiTietData(BaseModel):
      City: str
      Temperature: float
      Humidity: int
      Pressure: int
      Date: str

  # Định nghĩa route để lấy dữ liệu từ stored procedure
  @app.get("/thoitietdata/", response_model=List[ThoiTietData])
  async def get_thoitiet_data():
      try:
          cursor = cnxn.cursor()
          # Gọi stored procedure GetThoiTietData
          cursor.execute("{CALL GetThoiTietData}")
          rows = cursor.fetchall()

          # Chuyển dữ liệu thành danh sách các từ điển
          result = []
          for row in rows:
              result.append({
                  'City': row.City,
                  'Temperature': row.Temperature,
                  'Humidity': row.Humidity,
                  'Pressure': row.Pressure,
                  'Date': row.Date
              })

          return result
      except Exception as e:
          raise HTTPException(status_code=500, detail=str(e))

  if __name__ == "__main__":
      import uvicorn
      uvicorn.run(app, host="127.0.0.1", port=8810)
```

- Code script.js

```
// Hàm để tải dữ liệu từ URL JSON
google.charts.load('current', {'packages':['corechart']});
google.charts.setOnLoadCallback(drawCharts);
function drawCharts() {
    drawTemperatureChart();
    drawHumidityChart();
    drawPressureChart();
}
function loadData(url, callback) {
    fetch(url)
        .then(response => response.json())
        .then(data => callback(data))
        .catch(error => console.error('Error loading data:', error));
}

// Hàm để vẽ biểu đồ nhiệt độ
function drawTemperatureChart() {
    loadData('http://127.0.0.1:1880/get-temp', function(data) {
        // Tạo mảng dữ liệu cho biểu đồ
        var chartData = [];
        chartData.push(['Date', 'Temperature']); // Thêm tiêu đề cột

        // Lặp qua dữ liệu JSON và thêm vào mảng dữ liệu
        data.forEach(function(entry) {
            chartData.push([new Date(entry.Date), entry.Temperature]);
        });

        // Tạo DataTable từ mảng dữ liệu
        var dataTable =
google.visualization.arrayToDataTable(chartData);

        // Cấu hình tùy chọn cho biểu đồ
        var options = {
            title: 'Temperature Chart',
            curveType: 'function',
            legend: { position: 'bottom' }
        };

        // Vẽ biểu đồ
        var chart = new
google.visualization.LineChart(document.getElementById('temperature_cha
rt'));
        chart.draw(dataTable, options);
    });
}

// Hàm để vẽ biểu đồ độ ẩm
// Hàm để vẽ biểu đồ độ ẩm
function drawHumidityChart() {
    loadData('http://127.0.0.1:1880/get-humi', function(data) {
        // Tạo mảng dữ liệu cho biểu đồ
        var chartData = [];
        chartData.push(['Time', 'Humidity']); // Thêm tiêu đề cột
```

```
            // Lặp qua dữ liệu JSON và thêm vào mảng dữ liệu
            data.forEach(function(entry) {
                chartData.push([new Date(entry.Date), entry.Humidity]);
            });

            // Tạo DataTable từ mảng dữ liệu
            var dataTable =
    google.visualization.arrayToDataTable(chartData);

            // Cấu hình tùy chọn cho biểu đồ
            var options = {
                title: 'Humidity Chart',
                curveType: 'function',
                legend: { position: 'bottom' }
            };

            // Vẽ biểu đồ
            var chart = new
    google.visualization.LineChart(document.getElementById('humidity_chart'
    ));
            chart.draw(dataTable, options);
        });
    }

    // Hàm để vẽ biểu đồ áp suất
    function drawPressureChart() {
        loadData('http://127.0.0.1:1880/get-press', function (data) {
            // Tạo mảng dữ liệu cho biểu đồ
            var chartData = [];
            chartData.push(['Time', 'Pressure']); // Thêm tiêu đề cột

            // Lặp qua dữ liệu JSON và thêm vào mảng dữ liệu
            data.forEach(function (entry) {
                chartData.push([new Date(entry.Date), entry.Pressure]);
            });

            // Tạo DataTable từ mảng dữ liệu
            var dataTable =
    google.visualization.arrayToDataTable(chartData);

            // Cấu hình tùy chọn cho biểu đồ
            var options = {
                title: 'Pressure Chart',
                curveType: 'function',
                legend: {position: 'bottom'}
            };

            // Vẽ biểu đồ
            var chart = new
    google.visualization.LineChart(document.getElementById('pressure_chart'
    ));
            chart.draw(dataTable, options);
        });
    }
```

- Code html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Charts</title>
    <!-- Bao gồm thư viện Google Charts -->
    <script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></script>
    <script src="script.js"></script>
</head>
<body>
    <h1>Vẽ biểu đồ thời tiết</h1>
    <!-- Đây là nơi để hiển thị các biểu đồ -->
    <div id="temperature_chart" style="width: 900px; height: 500px;"></div>
    <div id="humidity_chart" style="width: 900px; height: 500px;"></div>
    <div id="pressure_chart" style="width: 900px; height: 500px;"></div>
</body>
</html>
```

Kết luận

Sau bài này em cảm nhận được từ lập trình trên nhiều nền tảng rất là thú vị không thể thiếu đó là chatgpt cho ta am hiểu về code hơn. Phần mảng này và mảng nọ công dụng rất có ích cho tương lai sau này em sẽ cố gắng phát triển thêm…