

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP KHOA ĐIỆN TỬ
Bộ môn: Công nghệ thông tin



BÀI TẬP LỚN
LẬP TRÌNH PYTHON



Sinh viên: Nguyễn Phi Vũ

Mssv: K205480106032

Lớp: K56KMT

Giáo viên hướng dẫn: Đỗ Duy Cốp

PHIẾU GIAO ĐỀ TÀI

Họ và tên: Nguyễn Phi Vũ Mssv:

K205480106032

Lớp: K56KMT

Ngành: Kỹ thuật máy tính

Giáo viên hướng dẫn: Đỗ Duy Cốp

Ngày giao đề tài: 16/05/2024

Ngày hoàn thành: 26/05/2024

1. **Tên đề tài: Xây dựng web theo dõi giá đất.**

2. **Nội dung yêu cầu làm:**

1. Cơ sở dữ liệu: mssql, hoặc mysql (ko dùng sqlite). có các bảng gì? ghi chú đầy đủ PK, FK. Có các SP_ nào.
2. Module đọc dữ liệu: sử dụng Python + FastAPI, lấy dữ liệu từ cảm biến, hoặc từ web, hoặc từ 1 nguồn nào đó, ko sinh random. Mô tả nguồn dữ liệu và thuật toán để lấy được nó. kết hợp python với FastAPI để tạo thành 1 API cho ứng dụng khác khai thác. đóng gói nó thành service với NSSM.
3. Nodered: Chu trình tự động gọi API python để lấy về dữ liệu, có thể phải xử lý thêm, gọi node tương tác với csdl để đẩy được dữ liệu vào database (nên gọi SP và truyền tham số, ko gọi trực tiếp lệnh INSERT).
4. web: thực hiện việc lấy dữ liệu từ db ra để làm gì đó, vd như vẽ biểu đồ, hoặc làm các thứ khác liên quan đến dữ liệu đã lấy được.

GIÁO VIÊN HƯỚNG DẪN
(ký và ghi rõ họ tên)

[illegible]

GIÁO VIÊN HƯỚNG DẪN

(Ký ghi rõ họ tên)

MỤC LỤC

LỜI CAM ĐOAN	5
CHƯƠNG I: GIỚI THIỆU VỀ NGÔN NGỮ PYTHON	6
CHƯƠNG II: TỔNG QUAN VỀ ĐỀ TÀI	7
2.1. Giới thiệu đề tài.	7
2.2. Lý do chọn đề tài.	7
2.3. Mục tiêu nghiên cứu.	8
CHƯƠNG III: CÁC NGÔN NGỮ ĐỂ LẬP TRÌNH	9
3.1. Cài đặt ngôn ngữ lập trình python gồm có các ngôn ngữ để lập trình python như Pycharm, Visual Studio, Visual code...	9
3.2. Cài đặt node-red:	9
3.3. Cài đặt SQL Server:	11
CHƯƠNG IV: LẬP TRÌNH VÀ KIỂM THỬ	12
4.1. Lập trình	12
4.2. Link sản phẩm:	18
CHƯƠNG V: KẾT LUẬN	19

LỜI CAM ĐOAN

Tôi xin cam đoan đề tài “**Xây dựng web theo dõi giá đất**” là đề tài nghiên cứu cá nhân của tôi trong thời gian qua. Mọi số liệu sử dụng phân tích trong đề tài và kết quả nghiên cứu là do tôi tự tìm hiểu, phân tích một cách khách quan, trung thực, có nguồn gốc rõ ràng và chưa được công bố dưới bất kỳ hình thức nào.

Tên sinh viên

Nguyễn Phi Vũ

CHƯƠNG I: GIỚI THIỆU VỀ NGÔN NGỮ PYTHON

Python là một ngôn ngữ lập trình bậc cao, được tạo ra bởi Guido van Rossum và phát hành lần đầu tiên vào năm 1991. Với cú pháp rõ ràng và dễ hiểu, Python đã nhanh chóng trở thành một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới. Python được thiết kế để dễ đọc và dễ viết, giúp cho các lập trình viên, kể cả những người mới bắt đầu, có thể nhanh chóng làm quen và sử dụng hiệu quả.

Python là một ngôn ngữ đa năng, hỗ trợ nhiều phong cách lập trình như lập trình hướng đối tượng, lập trình chức năng, và lập trình mệnh lệnh. Nó có một thư viện chuẩn rất phong phú, cung cấp các module và gói phần mềm để giải quyết nhiều vấn đề khác nhau, từ xử lý chuỗi và tính toán số học đến lập trình web và phát triển ứng dụng trí tuệ nhân tạo.

Một trong những điểm mạnh của Python là tính linh hoạt và khả năng mở rộng. Python có thể được sử dụng trong nhiều lĩnh vực khác nhau, từ phát triển web với các framework như Django và Flask, đến phân tích dữ liệu với các thư viện như Pandas và NumPy, và cả trong học máy với TensorFlow và scikit-learn.

Python cũng được cộng đồng lập trình viên ủng hộ mạnh mẽ, với một lượng lớn tài liệu, khóa học, và diễn đàn hỗ trợ. Điều này giúp cho việc học và phát triển với Python trở nên dễ dàng hơn bao giờ hết.

Với những ưu điểm vượt trội, Python đã và đang được sử dụng rộng rãi trong nhiều ngành công nghiệp, từ công nghệ thông tin, tài chính, y tế đến giáo dục, và trở thành một công cụ không thể thiếu đối với các lập trình viên trên toàn thế giới.

CHƯƠNG II: TỔNG QUAN VỀ ĐỀ TÀI

2.1. Giới thiệu đề tài.

Dự án xây dựng website theo dõi giá đất nhằm cung cấp cho người dùng một công cụ tiện lợi và hiệu quả để theo dõi biến động của giá đất theo thời gian thực. Trong bối cảnh thị trường bất động sản liên tục biến động, giá đất luôn là một chỉ số quan trọng được nhiều người quan tâm, từ các nhà đầu tư đến người tiêu dùng cá nhân.

Website sẽ tổng hợp dữ liệu từ các nguồn tin cậy như các cơ quan chính phủ, các công ty địa ốc uy tín, và các tổ chức nghiên cứu thị trường bất động sản, sau đó hiển thị giá đất dưới hình thức biểu đồ biến động. Mục tiêu của dự án là tạo ra một nền tảng dễ sử dụng, cung cấp thông tin chính xác và kịp thời, giúp người dùng đưa ra quyết định đầu tư bất động sản một cách thông minh.

Thông qua việc cập nhật thường xuyên và phân tích chi tiết về biến động giá đất, website sẽ giúp người dùng có cái nhìn toàn diện về thị trường bất động sản và đưa ra các quyết định dựa trên thông tin chính xác và đáng tin cậy.

2.2. Lý do chọn đề tài.

Lý do chọn đề tài xây dựng website theo dõi giá đất cũng bắt nguồn từ những yếu tố tương tự như đề tài theo dõi giá vàng. Dưới đây là một số lý do chính:

1. Nhu cầu Thực tế Cao: Giống như giá vàng, giá đất là một yếu tố quan trọng trong nền kinh tế và có ảnh hưởng đến nhiều khía cạnh trong đời sống cá nhân và doanh nghiệp. Việc cung cấp thông tin chính xác và kịp thời về giá đất là điều cần thiết cho những quyết định đầu tư và kinh doanh.
2. Công nghệ Tiện ích: Sự tiên bộ của công nghệ web cho phép tạo ra một nền tảng theo dõi giá đất linh hoạt và dễ sử dụng. Website này có thể cập nhật thông tin về giá đất từ các nguồn đáng tin cậy, hiển thị dưới dạng biểu đồ và phân tích chi tiết để người dùng dễ dàng theo dõi và đánh giá.
3. Phục vụ Cộng đồng: Website theo dõi giá đất không chỉ hỗ trợ cá nhân và doanh nghiệp có nhu cầu đầu tư mà còn giúp tăng cường sự minh bạch và hiệu quả trong các giao dịch bất động sản. Nó cũng có thể hỗ trợ các tổ chức chính phủ và địa phương trong việc quản lý và phát triển đô thị.
4. Học hỏi và Phát triển Kỹ năng: Đề tài này cũng mang lại cơ hội để thử nghiệm và phát triển các kỹ năng lập trình web, xử lý dữ liệu, và phân tích thông tin, đồng thời giúp người thực hiện đối mặt với các thách thức thực tế trong lĩnh vực bất động sản.

2.3.Mục tiêu nghiên cứu.

Mục tiêu nghiên cứu của dự án xây dựng website theo dõi giá đất là phát triển một nền tảng trực tuyến cung cấp thông tin về giá đất một cách chính xác, kịp thời và dễ hiểu cho người dùng. Các mục tiêu cụ thể bao gồm:

1. Cung cấp Thông tin Theo Thời gian Thực: Xây dựng hệ thống cập nhật giá đất liên tục từ các nguồn tin cậy, đảm bảo rằng người dùng luôn được cung cấp thông tin mới nhất và chính xác nhất để hỗ trợ trong quá trình ra quyết định đầu tư và kinh doanh bất động sản.
2. Hiển thị Dữ liệu Đa dạng và Trực quan: Thiết kế giao diện người dùng thân thiện, cho phép hiển thị giá đất dưới nhiều hình thức như bảng giá, biểu đồ biến động, và các phân tích xu hướng. Điều này giúp người dùng dễ dàng theo dõi và hiểu rõ tình hình thị trường địa ốc.
3. Hỗ trợ Quyết định Đầu tư: Cung cấp các công cụ phân tích và dự đoán giá đất dựa trên dữ liệu lịch sử và các thuật toán phân tích nâng cao, giúp người dùng đưa ra các quyết định đầu tư thông minh và hiệu quả.
4. Khả năng Mở rộng và Tích hợp: Xây dựng nền tảng với khả năng mở rộng để có thể dễ dàng tích hợp thêm các tính năng mới hoặc kết nối với các dịch vụ tài chính và thị trường bất động sản khác trong tương lai, từ đó tăng tính linh hoạt và hiệu quả cho người dùng.

CHƯƠNG III: CÁC NGÔN NGỮ ĐỂ LẬP TRÌNH

3.1. Cài đặt ngôn ngữ lập trình python gồm có các ngôn ngữ để lập trình python như Pycharm, Visual Studio, Visual code...

Ở đây em dùng Pycharm để sử dụng dự án của đề tài này:



Để sử dụng ngôn ngữ này, ta cần cài ứng dụng này như sau:

Bước 1: Lên web search Pycharm download về và cài đặt

Bước 2: Cài đặt thêm Python phiên bản mới nhất và path ứng dụng này vào trong hệ thống có đường dẫn như này (“D:\python\python.exe”) ở đây em cài phiên bản 3.10.0.



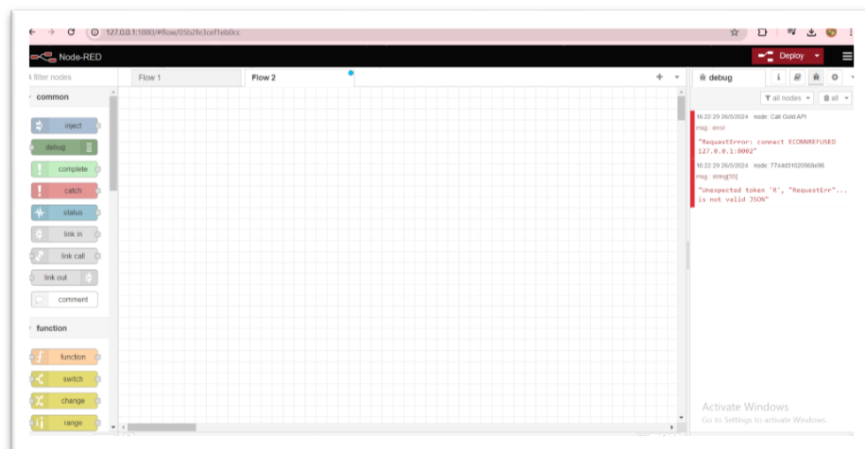
Bước 3: Vào Pycharm tạo New Project để sử dụng lập trình trong dự án.

3.2. Cài đặt node-red:

Bước 1: lên web search tìm node.js download về và path ứng dụng này vào trong hệ thống ở máy tính có đường dẫn sau: (“D:\node-red\node.exe”) sau khi path xong thì chúng ta kiểm tra cài đặt dùng lệnh: “node -v” nếu nó xuất hiện ra phiên bản thì thành công.



Bước 2: Mở Terminal hoặc Command Prompt. ấn dòng lệnh code: node-red, nó sẽ hiện ra đường dẫn <http://127.0.0.1:1880/> copy đường dẫn này lên web google hoặc chrome để sử dụng môi trường lập trình của node-red. Nếu thành công nó sẽ hiện giao diện web như này.



3.3.Cài đặt SQL Server:

Bước 1: Cần cài đặt ngôn ngữ này lên web search download SQL Server, cần thiết thì cài phiên bản mới nhất cũng được, tránh trường hợp lỗi.



Bước 2: Sau khi cài xong thì ta cần Tạo tài khoản user name pass word để kết nối dữ liệu với ngôn ngữ khác.

CHƯƠNG IV: LẬP TRÌNH VÀ KIỂM THỬ

4.1.Lập trình

Sql

```
use Batdongsan

CREATE TABLE Property (
    ID INT PRIMARY KEY IDENTITY(1,1),
    TenDuAn INT,
    DiaChi FLOAT,
    GiaTri FLOAT,
    NgayCapNhat DATETIME DEFAULT CURRENT_TIMESTAMP
);

drop procedure InsertPropertyData
CREATE PROCEDURE InsertPropertyData
    @TenDuAn INT,
    @DiaChi FLOAT,
    @GiaTri FLOAT
AS
BEGIN
    INSERT INTO Property (
        TenDuAn, DiaChi, GiaTri
    ) VALUES (
        @TenDuAn, @DiaChi, @GiaTri
    );
END;
```

Main.py

```
from fastapi import FastAPI, HTTPException
import pandas as pd
import asyncio
import nest_asyncio
import uvicorn

app = FastAPI()

CSV_FILE_PATH =
"C:/Users/Admin/Downloads/City_zhvi_uc_sfrcondo_tier_0.33_0.67_sm_sa_month.csv"
UPDATE_INTERVAL = 60 # Update interval (seconds)

land_price = None

async def fetch land price():
    global land_price
    try:
        df = pd.read_csv(CSV_FILE_PATH)
        # Assuming you want to return the entire dataframe as JSON
        land_price = df.to_dict(orient='records')
    except Exception as e:
        print(f"Error reading CSV file: {str(e)}")
        raise HTTPException(status_code=500, detail=f"Error reading CSV file:
```

```

{str(e)}")

async def update_land_price_periodically():
    while True:
        try:
            await fetch_land_price()
        except Exception as e:
            print(f"Error while updating land price: {e}")
            await asyncio.sleep(UPDATE_INTERVAL)

@app.get("/land_price")
async def get_land_price():
    if land_price is None:
        await fetch_land_price()
    return land_price

if __name__ == "__main__":
    nest_asyncio.apply()
    loop = asyncio.new_event_loop()
    asyncio.set_event_loop(loop)
    loop.create_task(update_land_price_periodically())
    loop.run_until_complete(fetch_land_price())
    uvicorn.run(app, host="127.0.0.1", port=8242)

```

App.py

```

from fastapi import FastAPI, HTTPException
import pandas as pd
import asyncio
import pyodbc
from datetime import datetime

app = FastAPI()

# Cấu hình kết nối với cơ sở dữ liệu SQL Server
DATABASE_URL = "DRIVER={ODBC Driver 17 for SQL Server};SERVER=CONGANH\\SQLEXPRESS;DATABASE=Batdongsan;UID=sa;PWD=abc1234;"
connection = pyodbc.connect(DATABASE_URL)

# Thiết lập các thông số cho kết nối SQL Server
cursor = connection.cursor()

CSV_FILE_PATH = r"C:\Users\Admin\Downloads\City_zhvi_uc_sfrcondo_tier_0.33_0.67_sm_sa_month.csv" # Thay đổi đường dẫn đến file CSV
UPDATE_INTERVAL = 60 # Thời gian cập nhật (giây)

csv_data = None

async def fetch_csv_data():
    global csv_data
    try:
        csv_data = pd.read_csv(CSV_FILE_PATH)
    except Exception as e:
        raise HTTPException(status_code=500, detail=f"Failed to fetch data from CSV file: {e}")

async def update_data_periodically():

```

```

while True:
    try:
        await fetch_csv_data()
    except Exception as e:
        print(f"Error while updating data: {e}")
        await asyncio.sleep(UPDATE_INTERVAL)

@app.get("/csv_data")
async def get_csv_data():
    if csv_data is None:
        await fetch_csv_data()
    return csv_data.to_dict(orient="records")

@app.on_event("startup")
async def startup_event():
    asyncio.create_task(update_data_periodically())

@app.post("/properties/")
async def create_property(TenDuAn: str, DiaChi: str, GiaTri: float):
    try:
        cursor.execute("INSERT INTO Property (TenDuAn, DiaChi, GiaTri, NgayCapNhat)
VALUES (?, ?, ?, ?)", (TenDuAn, DiaChi, GiaTri, datetime.now()))
        connection.commit()
        return {"message": "Property created successfully"}
    except Exception as e:
        raise HTTPException(status_code=500, detail=f"Error while inserting property:
{e}")

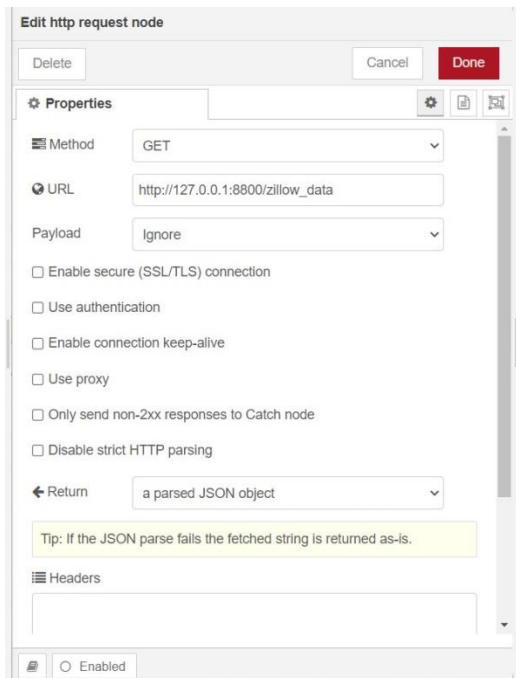
@app.get("/properties/")
async def read_properties():
    try:
        cursor.execute("SELECT * FROM Property")
        properties = cursor.fetchall()
        return properties
    except Exception as e:
        raise HTTPException(status_code=500, detail=f"Error while fetching properties:
{e}")

if __name__ == "__main__":
    import uvicorn
    uvicorn.run(app, host="127.0.0.1", port=8800)

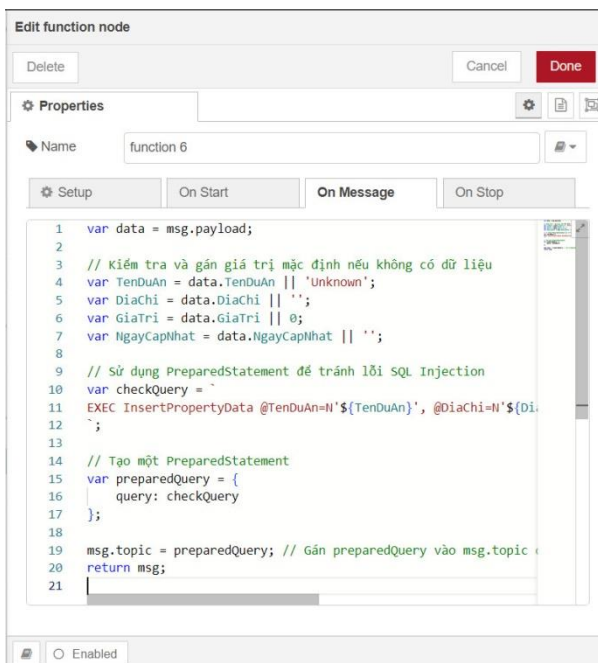
```

Ở Node-RED dùng node https request lấy địa chỉ có chứa endpoint.

Thiết lập một flow trong Node-RED để gửi HTTP request tới endpoint địa phương (localhost)



Tiếp đó em viết một function để xử lí dữ liệu và trả về json



Thực thi procedure để thêm dữ liệu vào SQL

Dialog: Edit MSSQL node

Buttons: Delete, Cancel, Done

Properties:

- Connection: CONGANH.Batdongsan
- Name: Name
- Query mode: Execute Procedure
- Query: Editor (1 InsertPropertyData)
- Parameters: Editor
 - Input: Name DiaChi, Type FLOA, Value msg.payload.DiaChi
 - Input: Name GiaTri, Type FLOA, Value msg.payload.GiaTri
- Parse Mustache: ☐
- Output property: msg.payload

Cài đặt NODE-RED- MSSQL-PLUS: sau khi cài đặt cấu hình các thông tin cho node

Dialog: Edit MSSQL node > Edit MSSQL-CN node

Buttons: Delete, Cancel, Update

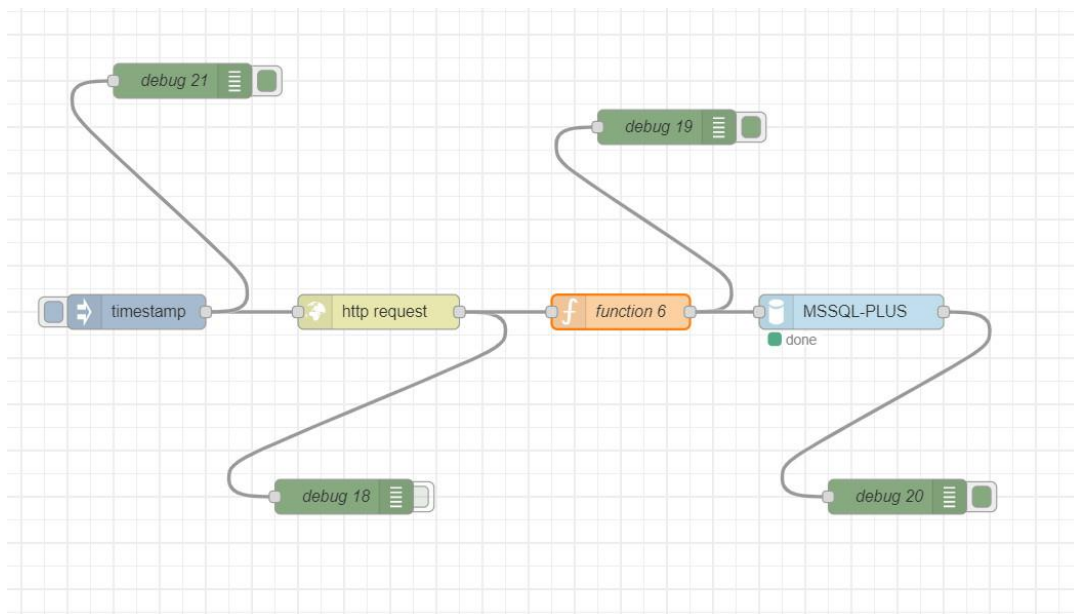
Properties:

- Name: Connection Name
- Server: CONGANH
- Port: 1433
- Username: sa
- Password:
- Domain:
- Database: Batdongsan
- TDS Version: 7_4 (SQL Server 2012 ~ 2022)
- Use Encryption?: ☐
- Trust Certificate?: ☐

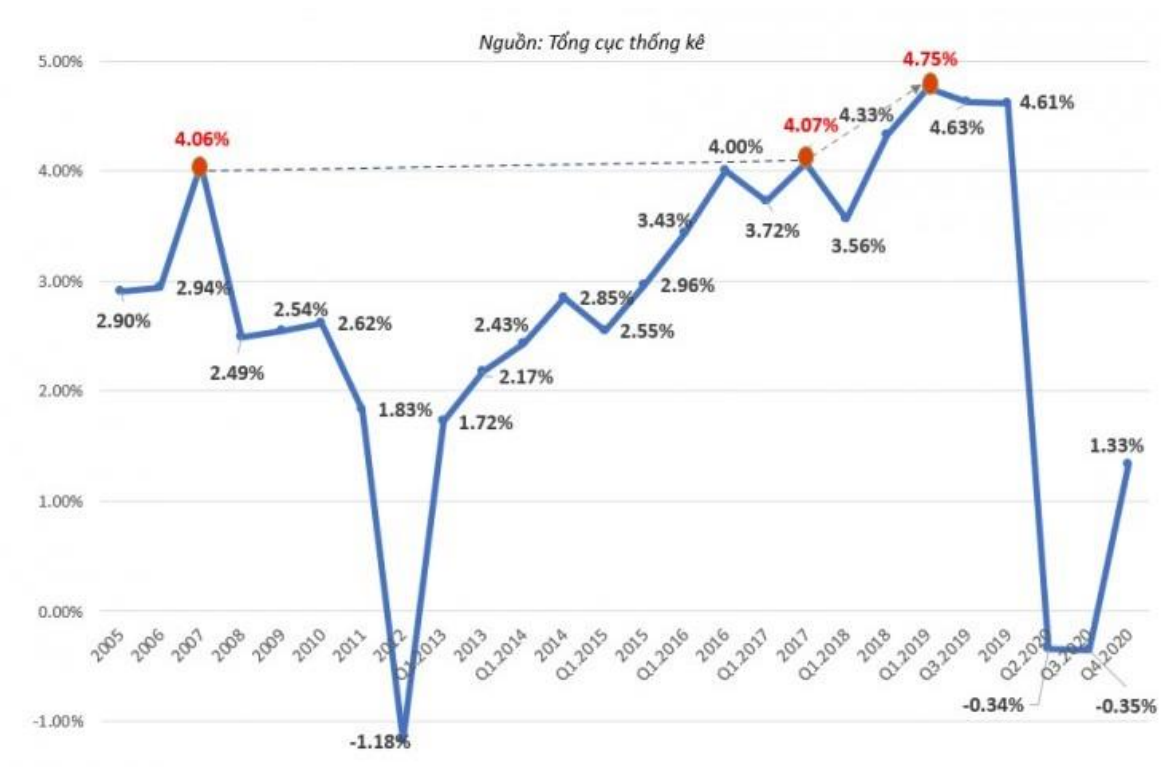
SQL Databases hosted on Azure will need this checked.

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

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



Kết quả trên web



4.2.Link sản phẩm:



CHƯƠNG V: KẾT LUẬN

Bài tập lớn này thực sự là một thành công đáng khen ngợi! Việc kết hợp FastAPI, Node-RED và cơ sở dữ liệu SQL đã tạo ra một hệ thống mạnh mẽ và linh hoạt, cho phép thu thập, xử lý và hiển thị dữ liệu về không khí một cách tự động và hiệu quả.

FastAPI đã giúp bạn dễ dàng tạo ra các API RESTful, tạo điều kiện cho việc giao tiếp giữa các phần của hệ thống trở nên đơn giản và hiệu quả. Node-RED đã đóng vai trò quan trọng trong việc thiết lập các luồng công việc và tích hợp các thành phần khác nhau của hệ thống. Sử dụng cơ sở dữ liệu SQL đã giúp bạn lưu trữ dữ liệu một cách có tổ chức và dễ dàng truy xuất.

Giao diện người dùng đã được thiết kế một cách trực quan và thân thiện, giúp người dùng dễ dàng nắm bắt thông tin quan trọng về tình hình chất lượng không khí. Hệ thống này cũng có tiềm năng để mở rộng và phát triển trong tương lai, với khả năng tích hợp thêm các tính năng mới và xử lý nhiều loại dữ liệu khác nhau.

Tóm lại, qua bài tập lớn này, bạn đã có cơ hội hiểu sâu hơn về ngôn ngữ lập trình Python và các công cụ như Node-RED và SQL. Cảm ơn bạn đã chia sẻ và chúc mừng bạn đã hoàn thành một công việc tuyệt vời!