Quá trình hoàn thiện bài tập.

Họ và tên: Nguyễn Đức Mạnh

Mssv: K205480106038

Lớp : 56KMT.01

Môn: Lập trình Python

Đề bài Theo dõi giá xe hơi Vinfast

1. Quá trình cài đặt và môi trường
   * Firstly, sử dụng anaconda để chạy pycham để sử dụng không gian riêng quản lý bởi anaconda
   * Secondly, trong terminal của workspace cài đặt nodeJS và fastapi <https://fastapi.tiangolo.com/#installation><https://nodejs.org/en/download/current>

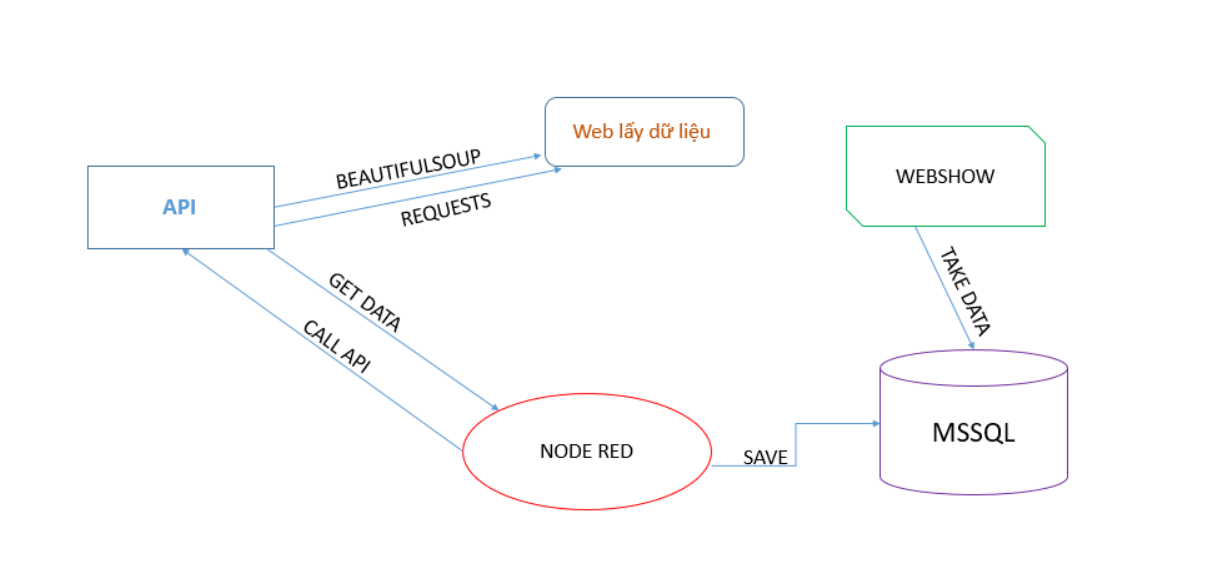
* Cài đặt node-red: yêu cầu cài nodejs trước để kích hoạt npm sau đó:

npm install -g --unsafe-perm node-red

* Cài SQl Server 2022 :

1. Truy cập vào đường link: https://www.microsoft.com/en-us/sql-server/sql-server-downloads.
2. Tìm đến Express edition of **SQL** **Server** **2022** và chọn **Download** now, chọn Open file.
3. Chọn Basic -> Accept -> Chọn Browser lưu trữ -> Install.
4. Sau khi **tải** về thành công, ấn Install SSMS.
5. Sau khi cửa sổ **Download** SSMS, ấn vào Link **Download** SSMS.
6. Sau khi **Download** thành công, ấn Install.
7. Sau khi cài đặt mở Microsoft **SQL** **Server** Management Studio 19, chọn Connect.
8. Làm bài

- Sơ đồ hoạt động



Quá trình :

- Tạo api fast api python lấy data trên web data

- Dùng node red get data api,lưu data về mssql

- Dùng python để gọi dữ liệu từ mssql ra đưa lên web

* + Input: Thông tin từ https://vinfastauto.com/vn\_vi/bang-gia-xe-vinfast-moi-nhat
  + Output: Biểu đồ thông tin
  + Quá trình tìm hiểu và nghiên cứu có thời gian khá ngắn nên không hiểu chi tiết về từng code và công dụng nên phải hỏi chatGPT khá nhiều.

Tạo cơ sở dữ liệu

* Tạo ra 1 bảng có tên là Cars, trong bảng gồm có CarID, Brand, Model, Year, Price

USE [CarDatabase]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Cars] Script Date: 05/17/2024 6:27:51 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Cars](

[CarID] [int] IDENTITY(1,1) NOT NULL,

[Brand] [nvarchar](50) NULL,

[Model] [nvarchar](50) NULL,

[Year] [datetime] NULL,

[Price] [float] NULL,

PRIMARY KEY CLUSTERED

(

[CarID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

Tạo Stored Procedures để kết nối với node red

* Nó cung cấp một cách để thực thi các tập hợp câu lệnh SQL đã được biên dịch trước, giúp tăng hiệu suất và dễ quản lý các thao tác phức tạp với cơ sở dữ liệu

use CarDatabase

go

CREATE PROCEDURE AddCar

@Brand NVARCHAR(50),

@Model NVARCHAR(50),

@Year DateTime,

@Price FLOAT

AS

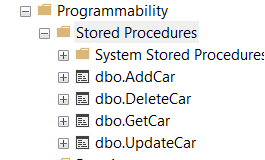
BEGIN

INSERT INTO Cars (Brand, Model, Year, Price)

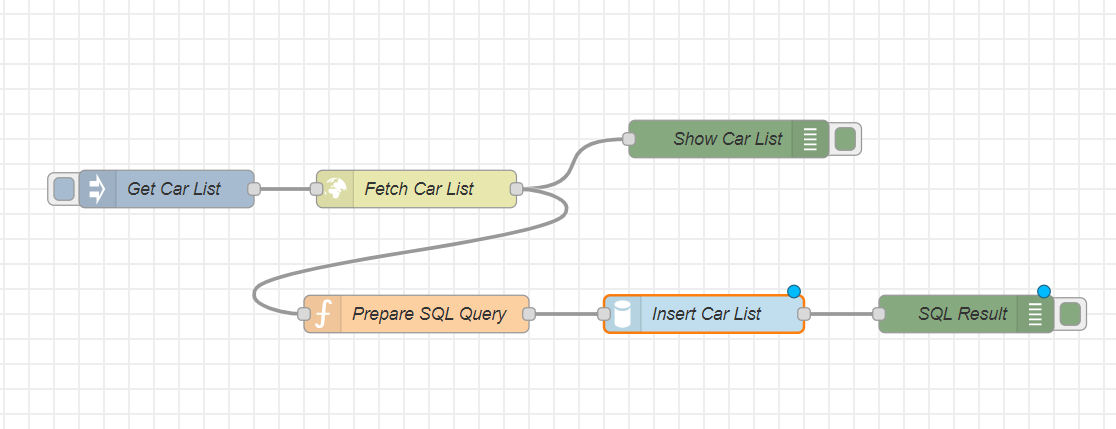
VALUES (@Brand, @Model, @Year, @Price);

END;

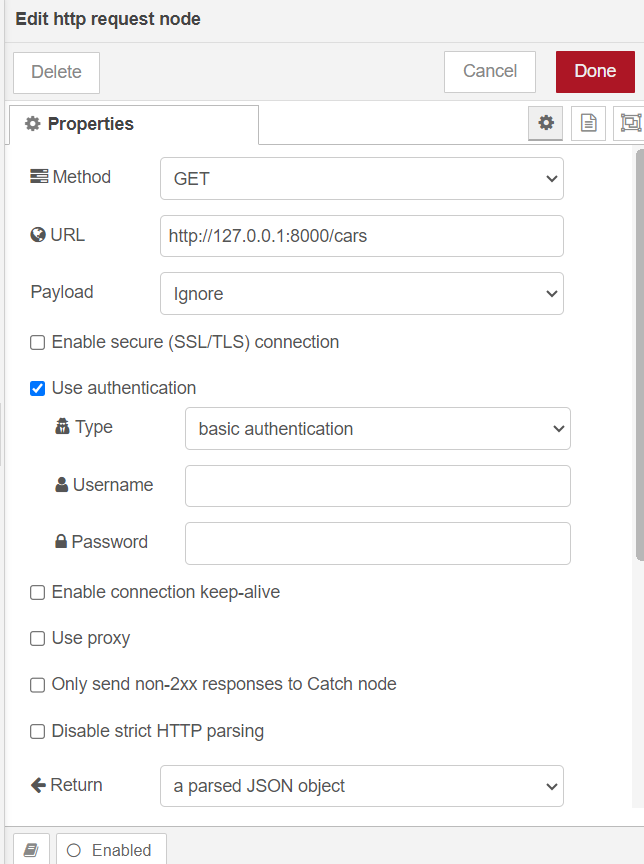
Kết quả hiển thị



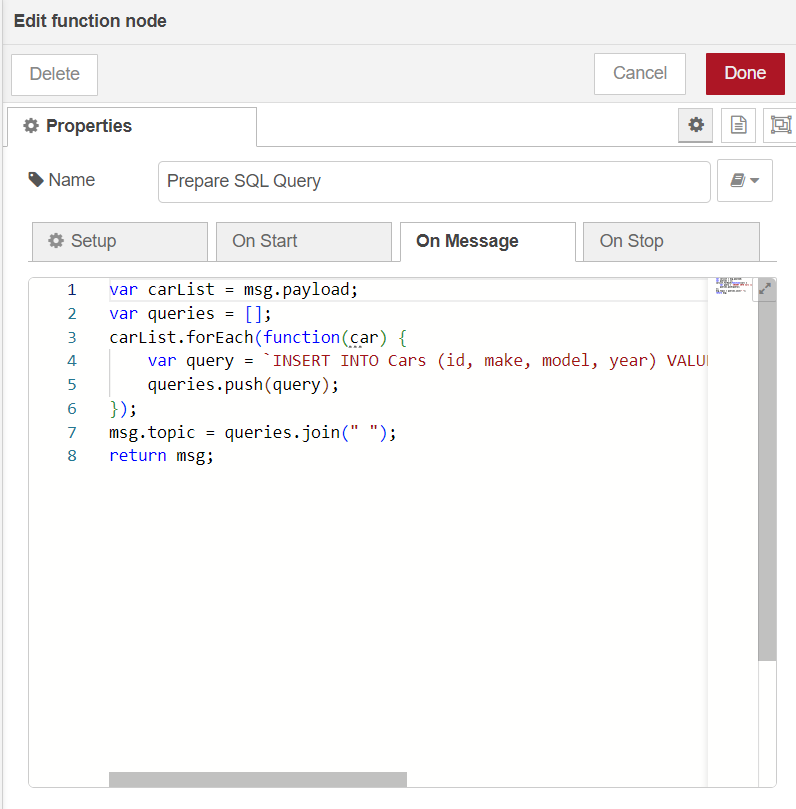
Tạo Node-red



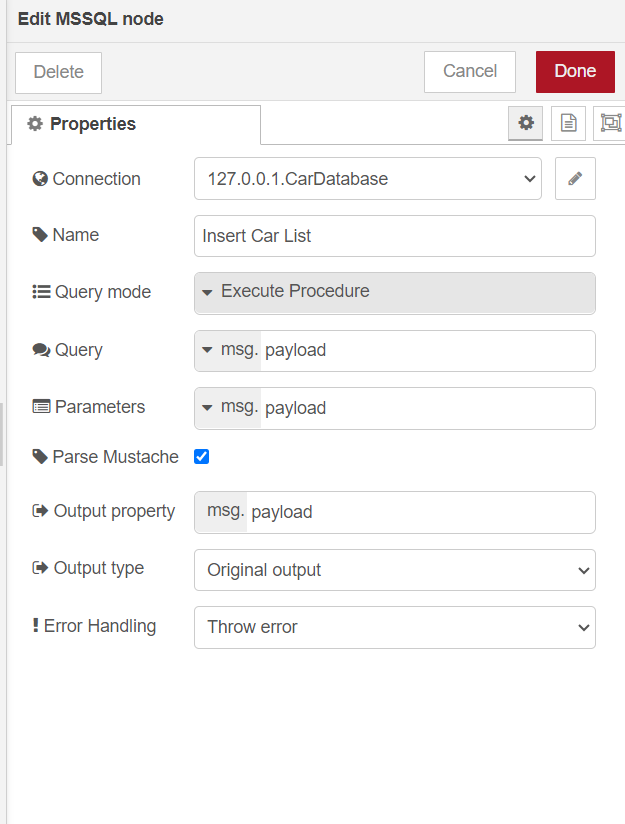
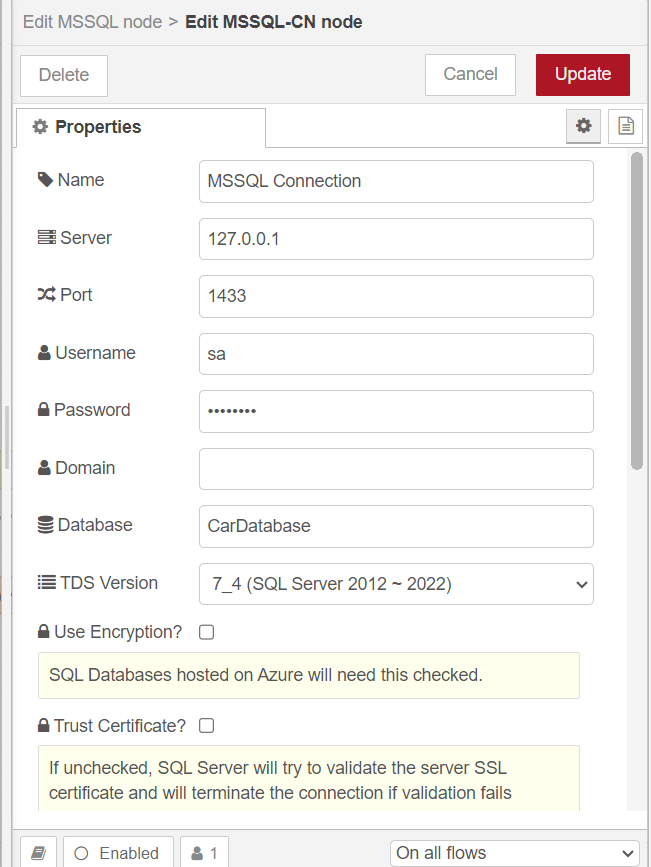
Tiếp theo em sẽ sử dụng node-red và dán đường link vào http response



* Tạo các câu lệnh SQL từ dữ liệu đầu vào: Node này nhận một danh sách các đối tượng xe từ msg.payload, sau đó tạo các câu lệnh SQL để chèn dữ liệu vào bảng "Cars".
* Chuẩn bị cho việc chèn dữ liệu vào cơ sở dữ liệu: Các câu lệnh SQL được chuẩn bị và gán vào msg.topic, sẵn sàng để được thực thi bởi một node khác trong luồng Node-RED, chẳng hạn như một node cơ sở dữ liệu.



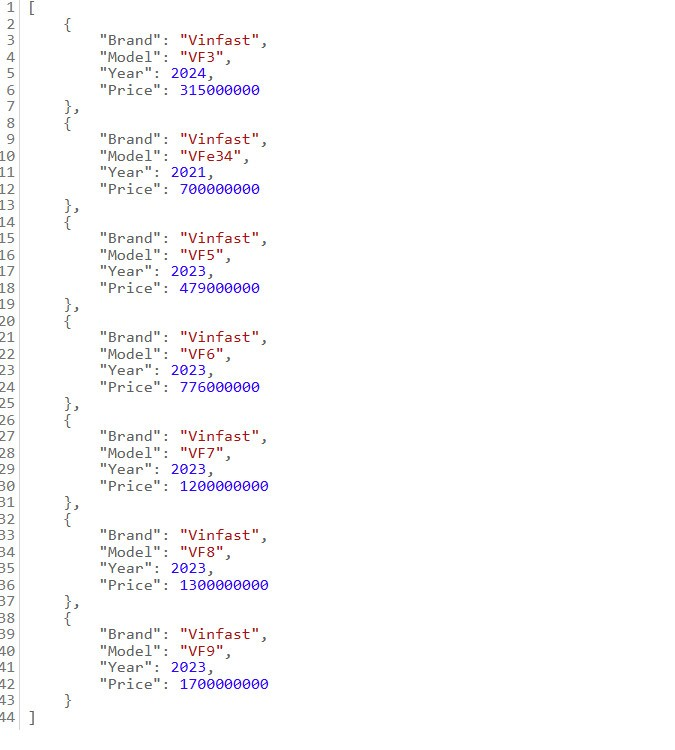
* Kết nối sql server với MSSQL node trên Node RED



Lấy dữ liệu api

@app.get("/Cars", response\_class=HTMLResponse)  
def get\_data() -> List[Dict[str, str]]:  
 url = 'https://vinfastauto.com/vn\_vi/bang-gia-xe-vinfast-moi-nhat'  
 try:  
 response = requests.get(url)  
 response.raise\_for\_status() # Kiểm tra trạng thái của yêu cầu HTTP  
 except requests.RequestException as e:  
 raise HTTPException(status\_code=500, detail=f"Lỗi khi truy cập URL: {e}")  
  
 soup = BeautifulSoup(response.content, 'html.parser')  
  
 # Trích xuất dữ liệu từ trang web  
 data = []  
 table = soup.find('table')  
 if not table:  
 raise HTTPException(status\_code=404, detail="Không tìm thấy bảng dữ liệu")  
  
 rows = table.find\_all('tr')[1:]  
 for row in rows:  
 cols = row.find\_all('td')  
 if len(cols) > 2:  
 car\_info = {  
 'Brand': cols[0].get\_text(strip=True),  
 'Model': cols[1].get\_text(strip=True),  
 'Year': cols[2].get\_text(strip=True),  
 'Price': cols[2].get\_text(strip=True)  
 }  
 data.append(car\_info)  
 print(data)  
 return data  
 return HTMLResponse(content=html\_content)

* Mảng Json lấy đc từ web data



Xuất ra kết quả



Kết quả sau khi chạy

.

