

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP

KHOA ĐIỆN TỬ

Bộ môn: Công nghệ thông tin



BÀI TẬP LỚN

MÔN HỌC

LẬP TRÌNH PYTHON

Họ và tên: Đỗ Tuấn Anh

Mã số sinh viên: K205480106002

Lớp K56KMT

Giáo viên hướng dẫn: Đỗ Duy Cốp

Thái Nguyên – 2024

TRƯỜNG ĐHKTCN
KHOA ĐIỆN TỬ

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

PHIẾU GIAO ĐỀ TÀI

Họ và tên: Đỗ Tuấn Anh

Mssv: K205480106002

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 Yên Nhật.

2. Nội dung thực hiện:

- Sử dụng API của các nguồn dữ liệu từ forex (có update realtime).
- Tạo một cơ sở dữ liệu trong SQL Server để lưu trữ dữ liệu về
- Sử dụng Node-RED để xây dựng các luồng dữ liệu tự động, kết nối và xử lý dữ liệu từ API đến cơ sở dữ liệu.
- Sử dụng FastAPI để tạo các endpoint API để truy xuất dữ liệu từ cơ sở dữ liệu.
- Sử dụng các công nghệ front-end (HTML, CSS, JavaScript, React.js) để xây dựng giao diện người dùng.
- Tạo biểu đồ thể hiện sự thay đổi của đồng Yên Nhật.

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

[illegible]

(Ký ghi rõ họ tên)

MỤC LỤC

MỤC LỤC	4
LỜI CAM ĐOAN	5
LỜI NÓI ĐẦU	6
CHƯƠNG I: GIỚI THIỆU VÀ HƯỚNG THỰC HIỆN ĐỀ TÀI.....	7
1. Giới thiệu về công cụ thực hiện đề tài.....	7
1.1 Visual Studio.....	7
1.2. Python.....	8
1.3. Node-red	8
1.4. SQL Server	9
2. Hướng thực hiện đề tài.	10
2.1. Giải pháp đề ra.....	10
2.2. Hướng giải quyết.	10
CHƯƠNG II: LẬP TRÌNH VÀ KIỂM THỬ	12
CHƯƠNG III: KẾT LUẬN	21

LỜI CAM ĐOAN

Em xin cam đoan đề tài “**Xây dựng web theo dõi giá Yên Nhật**” là đề tài nghiên cứu cá nhân của em trong thời gian qua. Mọi số liệu sử dụng phân tích trong đề tài và kết quả nghiên cứu là do em 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

Đỗ Tuấn Anh

LỜI NÓI ĐẦU

Trong thời đại kỷ nguyên số ngày nay, công nghệ thông tin không những là một phương tiện hữu ích mà còn giữ vai trò ngày càng quan trọng trong cuộc sống, quyết định trình độ phát triển từng khu vực, từng quốc gia, thông qua việc ngày càng làm tăng năng suất một cách tối đa. Việc áp dụng công nghệ tân tiến trong mọi mặt đời sống là một điều tất yếu.

Công nghệ thông tin đã thay đổi cách chúng ta tiếp cận và xử lý thông tin. Việc theo dõi các thông tin qua mạng Internet trở thành một phương pháp tiện lợi và hiệu quả cho mọi người, vì nó cho phép truy cập thông tin ở bất kỳ đâu và bất kỳ khi nào. Điều này đặc biệt quan trọng trong thời đại mà tiền tệ tất cả các nước đều cần được cập nhật. Các loại tiền tệ của các quốc gia rất đa dạng và biến động liên tục, khiến việc theo dõi giá trị của chúng trở thành nhu cầu cấp thiết đối với nhà đầu tư và người quan tâm.

Chính từ nhu cầu thực tiễn đó, em đã quyết định lựa chọn đề tài “Theo dõi giá Yên Nhật” với mong muốn giúp mọi người có thể theo dõi giá của đồng Yên Nhật một cách thuận tiện và nhanh chóng. Đề tài này không chỉ tập trung vào việc cập nhật giá Yên Nhật mà còn cung cấp các công cụ và phương pháp để người dùng có thể phân tích và dự đoán xu hướng giá, từ đó đưa ra quyết định đầu tư thông minh và hiệu quả.

Do kiến thức còn hạn hẹp nên báo cáo không tránh khỏi sai sót, rất mong sự đóng góp ý kiến của thầy cô và các bạn để báo cáo được hoàn chỉnh hơn.

CHƯƠNG I: GIỚI THIỆU VÀ HƯỚNG THỰC HIỆN ĐỀ TÀI

1. Giới thiệu về công cụ thực hiện đề tài

1.1 Visual Studio



Visual Studio là môi trường phát triển tích hợp (IDE) toàn diện do Microsoft phát triển, hỗ trợ nhiều ngôn ngữ lập trình như C#, C++, Visual Basic .NET, Python, và JavaScript. Ra đời từ năm 1997, Visual Studio nổi bật với khả năng hỗ trợ đầy đủ các giai đoạn phát triển phần mềm từ viết mã, gỡ lỗi, kiểm thử đến triển khai. Công cụ IntelliSense tự động hoàn thành mã và cung cấp thông tin cú pháp theo thời gian thực, giúp tăng hiệu suất và giảm thiểu lỗi cú pháp. Công cụ gỡ lỗi mạnh mẽ cho phép kiểm tra và sửa lỗi trên nhiều nền tảng khác nhau.

Visual Studio tích hợp nhiều dịch vụ như Git, Azure DevOps và các công cụ CI/CD, giúp dễ dàng quản lý mã nguồn và tự động hóa quá trình xây dựng và triển khai. Visual Studio Marketplace cung cấp hàng ngàn tiện ích mở rộng, cho phép tùy biến và mở rộng tính năng theo nhu cầu.

Với giao diện thân thiện và tính năng mạnh mẽ, Visual Studio là công cụ lý tưởng cho cả lập trình viên chuyên nghiệp và người mới bắt đầu, đóng vai trò quan trọng trong việc nâng cao hiệu quả phát triển phần mềm.

1.2. Python



Python là một ngôn ngữ lập trình cao cấp, được Guido van Rossum phát triển và ra mắt lần đầu vào năm 1991. Python nổi tiếng với cú pháp rõ ràng và dễ hiểu, giúp giảm thiểu thời gian viết và đọc mã, phù hợp cho cả người mới học lập trình và các lập trình viên chuyên nghiệp. Python hỗ trợ nhiều mô hình lập trình, bao gồm lập trình hướng đối tượng, lập trình chức năng và lập trình thủ tục, mang lại sự linh hoạt trong phát triển phần mềm.

Một trong những điểm mạnh của Python là thư viện phong phú và cộng đồng người dùng đông đảo. Các thư viện như NumPy, pandas, và TensorFlow hỗ trợ mạnh mẽ trong các lĩnh vực như khoa học dữ liệu, học máy và phát triển trí tuệ nhân tạo. Ngoài ra, Python còn được sử dụng rộng rãi trong phát triển web với các framework như Django và Flask.

Python cũng nổi bật với khả năng tích hợp và mở rộng, dễ dàng tương tác với các ngôn ngữ khác và tích hợp với nhiều hệ thống khác nhau. Nhờ tính đơn giản, mạnh mẽ và đa năng, Python đã trở thành một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới, được sử dụng rộng rãi trong giáo dục, nghiên cứu và công nghiệp.

1.3. Node-red



Node-RED là một công cụ lập trình dựa trên dòng chảy (flow-based) được phát triển bởi IBM vào năm 2013, chuyên dùng để kết nối các thiết bị phần cứng, API và dịch vụ trực tuyến. Với giao diện đồ họa trực quan, Node-RED cho phép người dùng tạo ra các ứng dụng Internet of Things (IoT) và tích hợp các hệ thống mà không cần viết nhiều mã.

Node-RED hoạt động dựa trên việc sử dụng các "node" (nút), mỗi node thực hiện một chức năng cụ thể và có thể được kéo và thả vào giao diện để tạo ra các luồng dữ liệu. Các node này có thể được kết nối với nhau để xây dựng quy trình xử lý dữ liệu phức tạp từ việc thu thập dữ liệu, xử lý và gửi dữ liệu đến các dịch vụ khác nhau.

Một trong những điểm mạnh của Node-RED là kho thư viện phong phú với hàng ngàn node có sẵn, hỗ trợ nhiều giao thức và dịch vụ khác nhau như MQTT, HTTP, WebSockets và nhiều hơn nữa. Node-RED cũng hỗ trợ tích hợp với nhiều dịch vụ đám mây và các nền tảng như IBM Cloud, AWS, và Microsoft Azure.

Với tính linh hoạt và dễ sử dụng, Node-RED là công cụ lý tưởng cho các dự án IoT, tự động hóa nhà cửa, và tích hợp hệ thống, giúp người dùng nhanh chóng xây dựng và triển khai các giải pháp thông minh.

1.4. SQL Server



SQL Server là hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mạnh mẽ do Microsoft phát triển, ra mắt lần đầu vào năm 1989. Nó hỗ trợ ngôn ngữ SQL (Structured Query Language) để quản lý và thao tác dữ liệu, cho phép dễ dàng tạo, đọc, cập nhật và xóa thông tin.

SQL Server nổi bật với tính năng tích hợp sâu với các sản phẩm khác của Microsoft như Windows Server, Azure và Power BI, giúp triển khai và quản lý cơ sở dữ liệu trong môi trường doanh nghiệp hiệu quả. Nó cung cấp các công cụ và tính năng bảo

mật mạnh mẽ, như mã hóa dữ liệu và quản lý quyền truy cập, đảm bảo an toàn cho dữ liệu nhạy cảm.

SQL Server có nhiều phiên bản khác nhau, từ phiên bản miễn phí như SQL Server Express, phù hợp cho các ứng dụng nhỏ, đến phiên bản cao cấp như SQL Server Enterprise với các tính năng tiên tiến và khả năng mở rộng lớn. Với hiệu suất cao, tính ổn định và khả năng tích hợp rộng rãi, SQL Server là lựa chọn hàng đầu cho nhiều doanh nghiệp trong việc xây dựng và quản lý hệ thống cơ sở dữ liệu.

2. Hướng thực hiện đề tài.

2.1. Giải pháp đề ra.

Để giải quyết vấn đề này, cần thiết phải xây dựng một hệ thống thu thập, xử lý và hiển thị dữ liệu ra đồng Yên Nhật một cách hiệu quả và chính xác. Hệ thống này không chỉ giúp cung cấp thông tin mới nhất mà còn giúp cho chúng ta thấy được sự thay đổi của đồng Yên Nhật. Mục tiêu của đề tài này là phát triển một giải pháp toàn diện bao gồm các bước sau:

- Thu thập dữ liệu tỷ giá đồng Yên Nhật (real time): Dữ liệu sẽ được lấy từ các nguồn uy tín và được cập nhật liên tục để đảm bảo tính chính xác và kịp thời.
- Xử lý dữ liệu: Sử dụng FastAPI để xây dựng các API dịch vụ và NodeRED để tổ chức các luồng xử lý dữ liệu, giúp dữ liệu được chuẩn hóa.
- Lưu trữ dữ liệu: Dữ liệu sau khi được xử lý sẽ được lưu trữ vào cơ sở dữ liệu SQL, đảm bảo khả năng truy xuất và quản lý dữ liệu một cách hiệu quả.
- Xây dựng trang web: hiển thị ra tỷ giá mới nhất của đồng Yên Nhật và biểu đồ thể hiện sự thay đổi của đồng đó.

2.2. Hướng giải quyết.

Để giải quyết bài toán và hiển thị thông tin một cách hiệu quả, chúng ta sẽ thực hiện các bước sau:

Bước 1: Dùng request để lấy dữ liệu thô từ API và gửi nó lên endpoint của FastAPI.

Sử dụng Python để gửi các yêu cầu POST đến endpoint của FastAPI nhằm thu thập dữ liệu thô. Thư viện `requests` trong Python sẽ hỗ trợ chúng ta trong việc này.

Bước 2: Ở Node-RED dùng node https request lấy địa chỉ có chứa endpoint (thường là cổng 127.0.0.1:8000/xxx).

Thiết lập một flow trong Node-RED để gửi HTTP request tới endpoint địa phương (localhost). Node-RED sẽ đóng vai trò trung gian, giúp ta gửi yêu cầu HTTP và nhận phản hồi từ endpoint.

Bước 3: Lưu vào SQL.

Khi nhận được dữ liệu từ endpoint, chúng ta sẽ lưu trữ dữ liệu này vào cơ sở dữ liệu SQL. Sử dụng ORM (Object Relational Mapping) như SQLAlchemy trong Python để tương tác với cơ sở dữ liệu một cách dễ dàng và hiệu quả.

Bước 4: Xây dựng giao diện người dùng lấy dữ liệu từ SQL để vẽ biểu đồ.

Sử dụng html,css,js để lấy dữ liệu từ sqlserver thông qua asp.net (api.aspx) Với hướng giải quyết này, chúng ta sẽ có một hệ thống toàn diện cho phép thu thập, xử lý và hiển thị hiệu quả, đồng thời cung cấp thông tin đáng tin cậy và dễ hiểu cho người dùng.

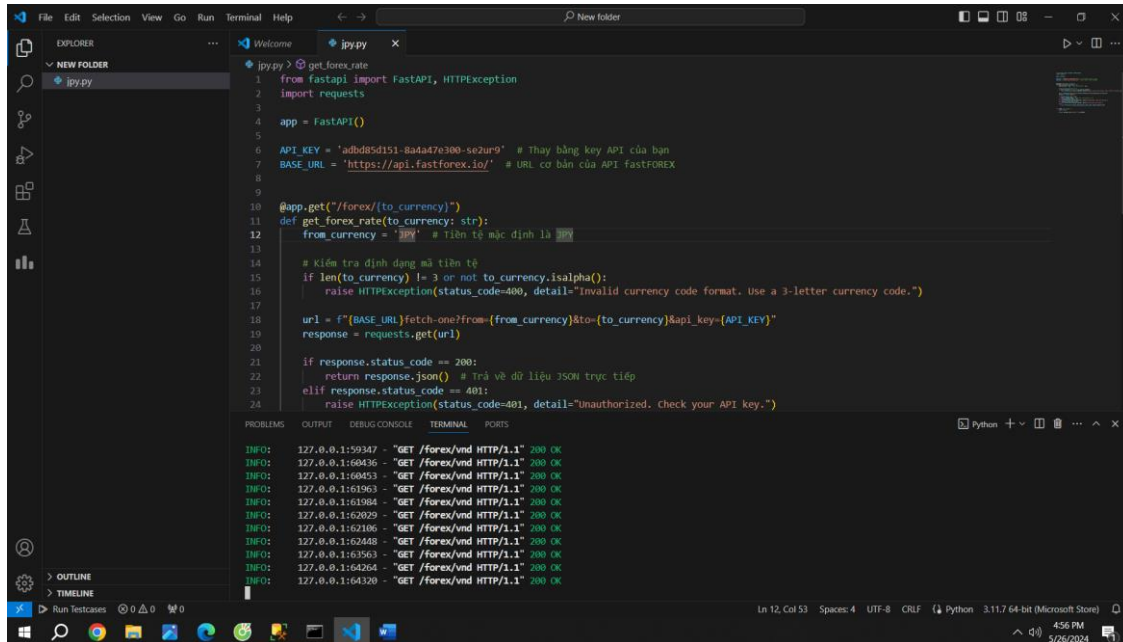
Giới hạn đề tài

Bài làm của em cơ bản đã hoàn thành được những yêu cầu đặt ra ở đầu. Nhưng do kiến thức còn hạn hẹp nên bài làm của em còn nhiều thiếu sót. Trong tương lai em sẽ cố gắng khắc phục những hạn chế để giúp hệ thống trở nên hoàn thiện hơn, đáp ứng tốt hơn nhu cầu của người dùng.

CHƯƠNG II: LẬP TRÌNH VÀ KIỂM THỬ

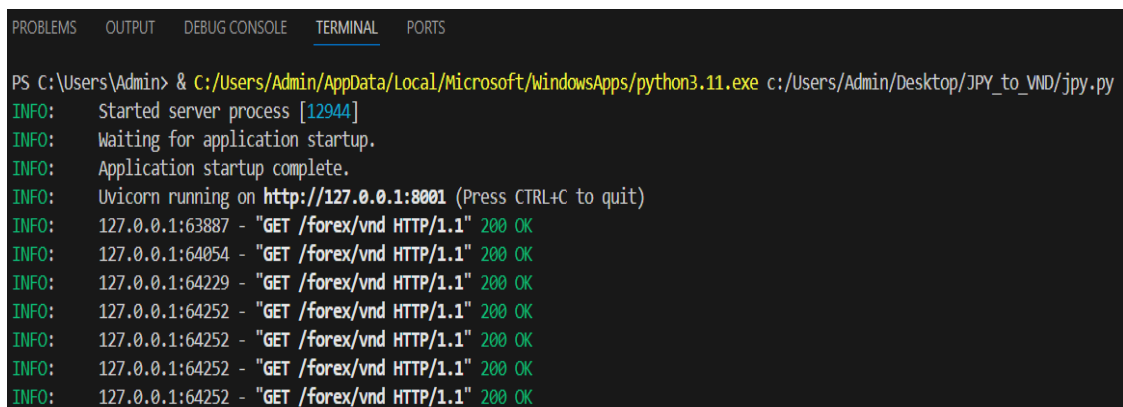
Các bước thực hiện:

Tạo file python sử dụng fastapi lấy dữ liệu realtime từ web forex Install các thư viện cần thiết:



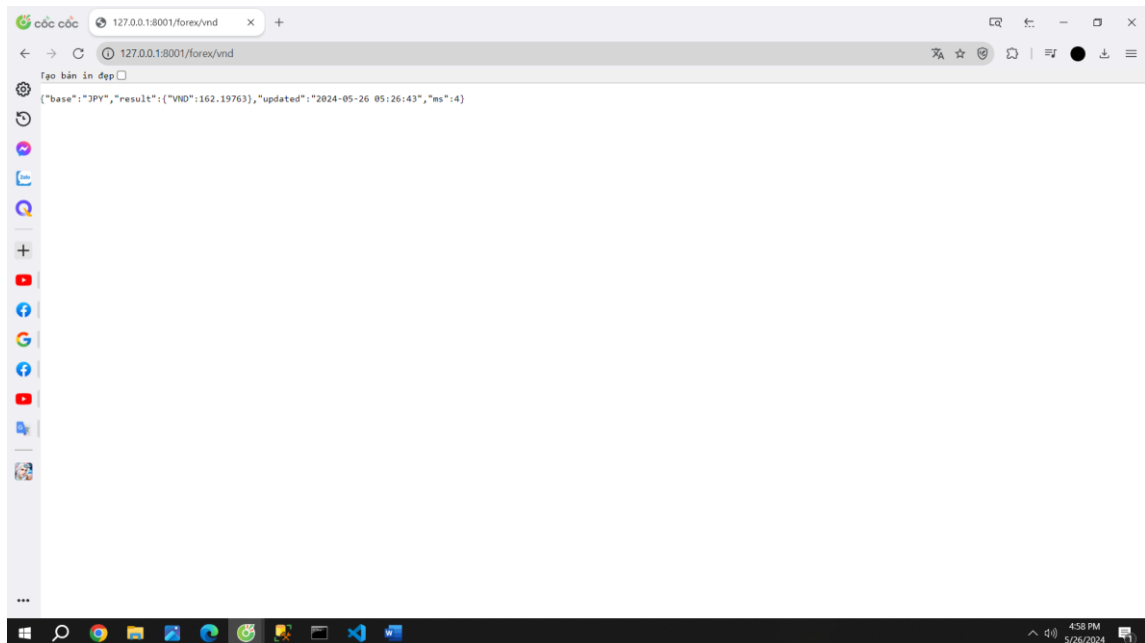
```
1 from fastapi import FastAPI, HTTPException
2 import requests
3
4 app = FastAPI()
5
6 API_KEY = 'adbd85d151-8a4a47e300-se2ur9' # Thay bằng key API của bạn
7 BASE_URL = 'https://api.fastforex.io/' # URL cơ bản của API fastFOREX
8
9
10 @app.get("/forex/{to_currency}")
11 def get_forex_rate(to_currency: str):
12     from_currency = 'JPY' # tiền tệ mặc định là JPY
13
14     # kiểm tra định dạng mã tiền tệ
15     if len(to_currency) != 3 or not to_currency.isalpha():
16         raise HTTPException(status_code=400, detail="Invalid currency code format. Use a 3-letter currency code.")
17
18     url = f"{BASE_URL}fetch-one?from={from_currency}&to={to_currency}&api_key={API_KEY}"
19     response = requests.get(url)
20
21     if response.status_code == 200:
22         return response.json() # Trả về dữ liệu JSON trực tiếp
23     elif response.status_code == 401:
24         raise HTTPException(status_code=401, detail="Unauthorized. Check your API key.")
```

Mở TERMINAL và chạy dòng lệnh

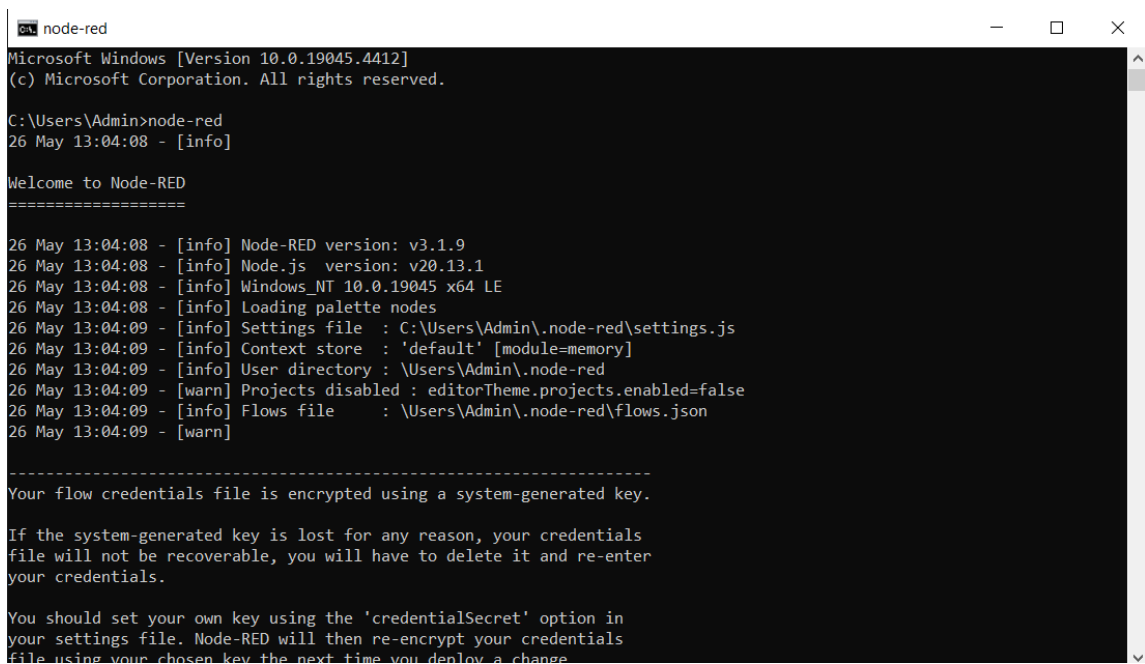


```
PS C:\Users\Admin> & C:/Users/Admin/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/Admin/Desktop/JPY_to_VND/jpy.py
INFO: Started server process [12944]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: Uvicorn running on http://127.0.0.1:8001 (Press CTRL+C to quit)
INFO: 127.0.0.1:63887 - "GET /forex/vnd HTTP/1.1" 200 OK
INFO: 127.0.0.1:64054 - "GET /forex/vnd HTTP/1.1" 200 OK
INFO: 127.0.0.1:64229 - "GET /forex/vnd HTTP/1.1" 200 OK
INFO: 127.0.0.1:64252 - "GET /forex/vnd HTTP/1.1" 200 OK
INFO: 127.0.0.1:64252 - "GET /forex/vnd HTTP/1.1" 200 OK
INFO: 127.0.0.1:64252 - "GET /forex/vnd HTTP/1.1" 200 OK
INFO: 127.0.0.1:64252 - "GET /forex/vnd HTTP/1.1" 200 OK
```

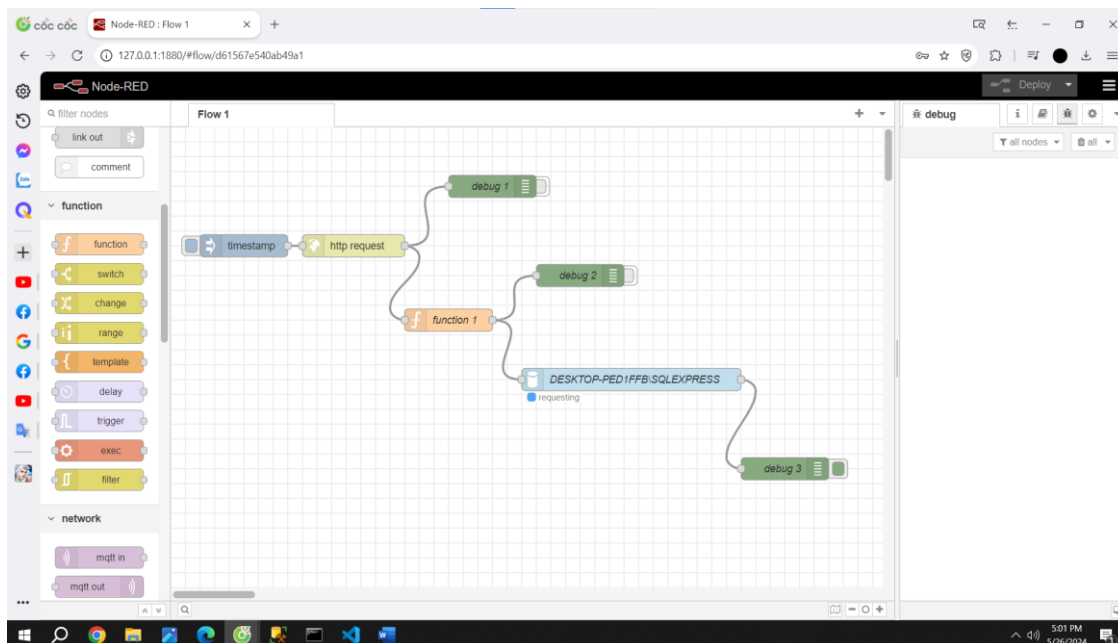
Sau khi chạy ta được api dạng chuỗi json:



Bước tiếp theo chúng ta mở CMD chạy node-red:



Đây là cấu trúc node-red của bài:



Tiếp theo cài 1 phút lấy dữ liệu 1 lần:

Edit inject node

Delete Cancel Done

Properties

Name Name

msg.payload = timestamp

msg.topic = a_z

+add inject now

☐ Inject once after 0.1 seconds, then

Repeat interval

every 2 minutes

☐ Enabled

Điền địa chỉ api mình chạy fast Api.

Edit http request node

Delete

Cancel

Done

⚙ Properties

⚙

📄

🖨

☰ Method

GET

▼

🌐 URL

http://127.0.0.1:8001/forex/vnd

Payload

Ignore

▼

☐ Enable secure (SSL/TLS) connection

☐ Use authentication

☐ Enable connection keep-alive

☐ Use proxy

☐ Only send non-2xx responses to Catch node

☐ Disable strict HTTP parsing

⬅ Return

a UTF-8 string

▼

☰ Headers

Cài đặt node-red-contrib-mssql-plus---> cài đặt cấu hình cho node.

Edit MSSQL node

Delete

Cancel

Done

⚙️ Properties

⚙️

📄

🖨️

🌐 Connection

forex

▼

✎

🔖 Name

Name

☰ Query mode

▼ Execute Procedure

💬 Query

▼ Editor

🔗

1

InsertDulieuJson

📋 Parameters

▼ Editor

☰

▼ Input

Name

json

Type

NVarChar

▼ msg.

payload

✕

+ add

🔖 Parse Mustache

☒

🔗 Output property

msg. payload

🔗 Output type

Original output

▼

⚠️ Error Handling

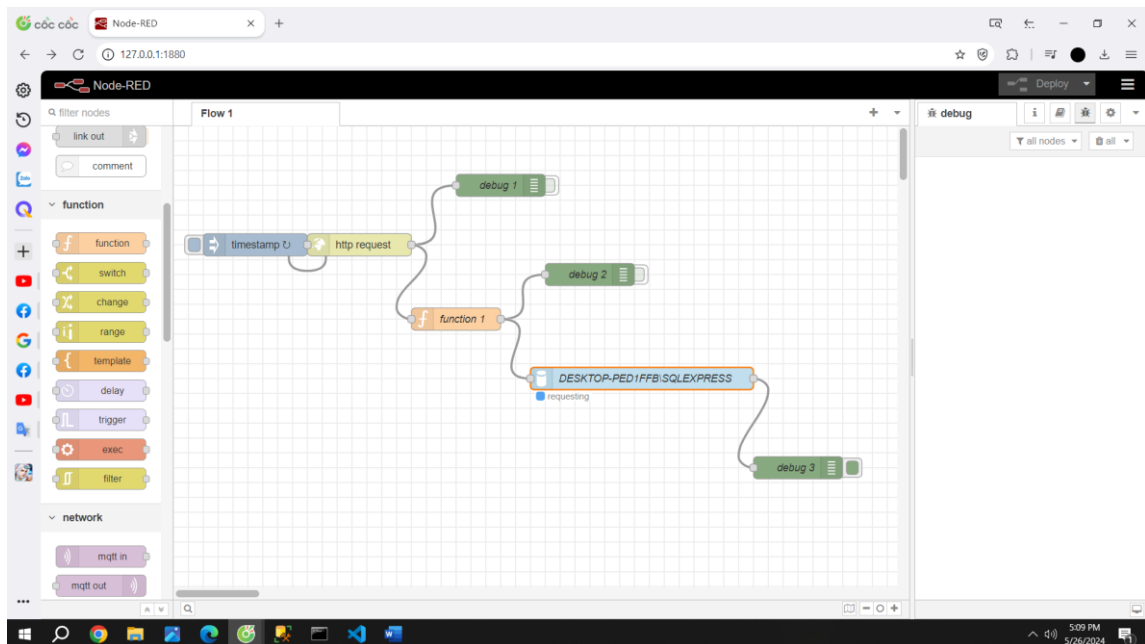
Throw error

▼

📄

☐ Enabled

Cấu trúc của đoạn node-red để lưu dữ liệu về sql.



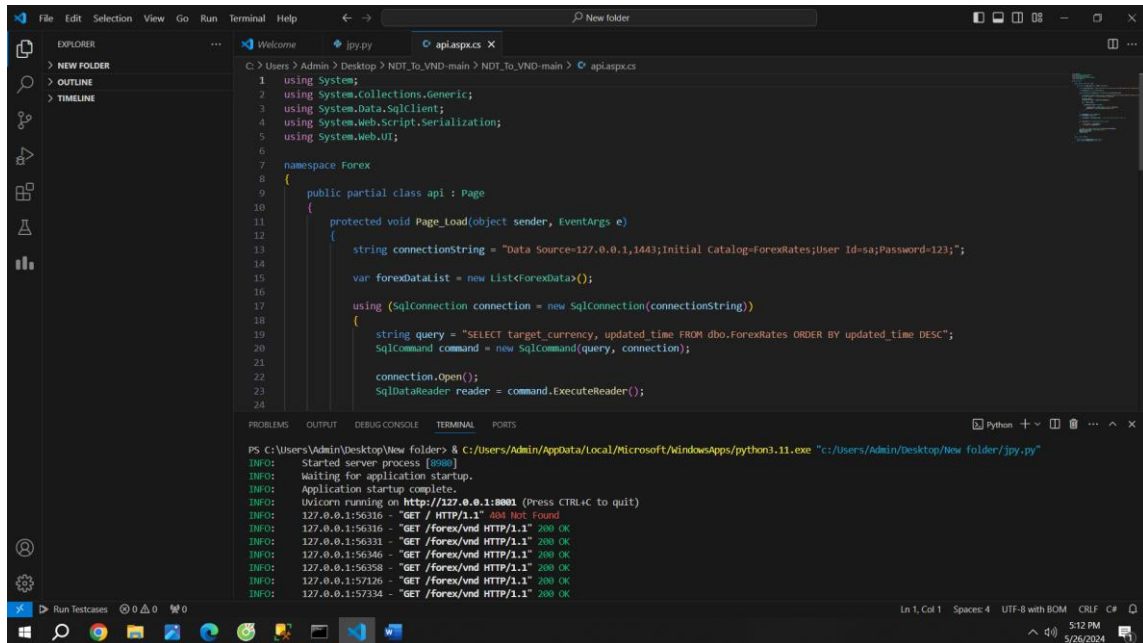
Sau đó viết SP trả dữ liệu dạng json rồi đẩy lên web.

```
ALTER PROCEDURE [dbo].[InsertDulieuJson]
    @json NVARCHAR(MAX)= null
AS
BEGIN
    DECLARE @base_currency NVARCHAR(50);
    DECLARE @target_currency DECIMAL(18, 0);
    DECLARE @exchange_rate FLOAT;
    DECLARE @updated_time DATETIME;

    -- Phân tích JSON và gán giá trị cho các biến
    SELECT
        @base_currency = JSON_VALUE(@json, '$.base'),
        @target_currency = CAST(JSON_VALUE(@json, '$.result.VND') AS DECIMAL(18, 0)),
        @exchange_rate = CAST(JSON_VALUE(@json, '$.result.VND') AS FLOAT),
        @updated_time = CONVERT(DATETIME, JSON_VALUE(@json, '$.updated'));

    -- Chèn dữ liệu vào bảng ExchangeRates
    INSERT INTO ForexRates(base_currency, target_currency, exchange_rate, updated_time)
    VALUES (@base_currency, @target_currency, @exchange_rate, @updated_time);
END;
```

Tiếp theo em kết nối SQL với VS studio lấy dữ liệu lần cập nhật cuối cùng và dữ liệu tiền và datetime đưa lên web bằng APS.NET.



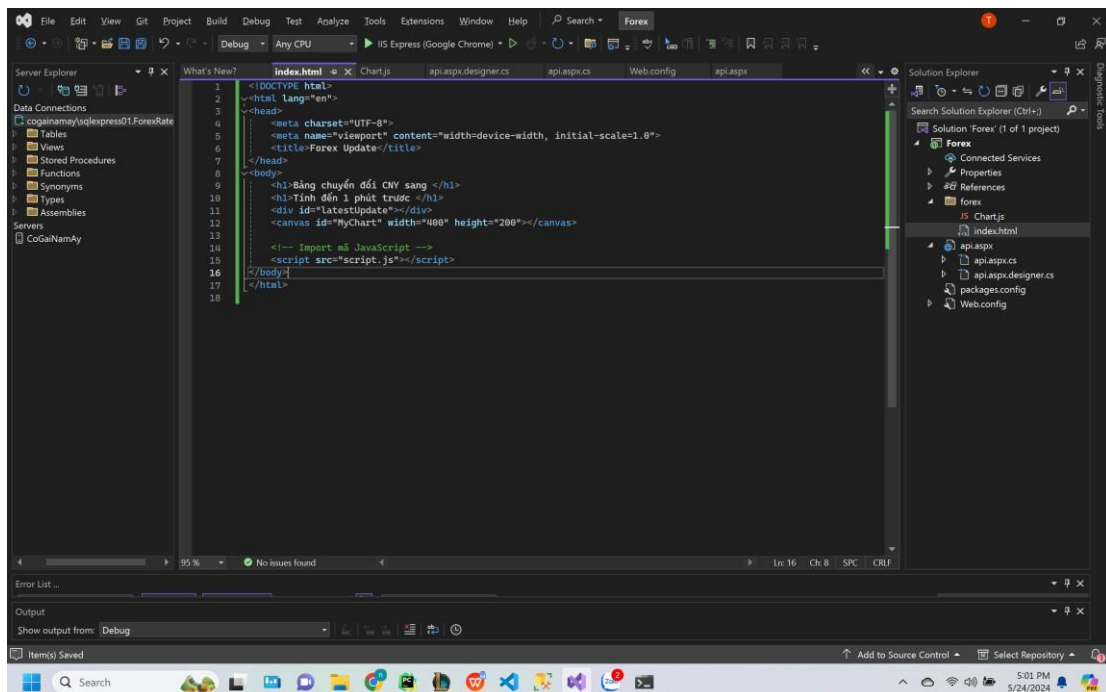
The screenshot shows the Visual Studio IDE with a C# file named `api.aspx.cs` open. The code is a partial class `api : Page` that implements `Page_Load`. It connects to a SQL database, executes a query to get the latest exchange rate, and returns it as JSON. The terminal at the bottom shows the application running on `http://127.0.0.1:8001` and handling several GET requests for the exchange rate.

```
1 using System;
2 using System.Collections.Generic;
3 using System.Data.SqlClient;
4 using System.Web.Script.Serialization;
5 using System.Web.UI;
6
7 namespace Forex
8 {
9     public partial class api : Page
10     {
11         protected void Page_Load(object sender, EventArgs e)
12         {
13             string connectionString = "Data Source=127.0.0.1,1443;Initial Catalog=ForexRates;User Id=sa;Password=123;";
14
15             var forexDataList = new List<ForexData>();
16
17             using (SqlConnection connection = new SqlConnection(connectionString))
18             {
19                 string query = "SELECT target_currency, updated_time FROM dbo.ForexRates ORDER BY updated_time DESC";
20                 SqlCommand command = new SqlCommand(query, connection);
21
22                 connection.Open();
23                 SqlDataReader reader = command.ExecuteReader();
24
25                 while (reader.Read())
26                 {
27                     var forexData = new ForexData
28                     {
29                         TargetCurrency = reader.GetString(0),
30                         UpdatedTime = reader.GetDateTime(1)
31                     };
32                     forexDataList.Add(forexData);
33                 }
34             }
35
36             var json = new JavaScriptSerializer().Serialize(forexDataList);
37             Response.ContentType = "application/json";
38             Response.Write(json);
39         }
40     }
41 }
```

Terminal Output:

```
PS C:\Users\Admin\Desktop\New folder> & C:\Users\Admin\AppData\Local\Microsoft\WindowsApps\python3.11.exe "c:\Users\Admin\Desktop\New folder\jpy.py"
INFO: started server process [8001]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: Uvicorn running on http://127.0.0.1:8001 (Press CTRL+C to quit)
INFO: 127.0.0.1:56316 - "GET / HTTP/1.1" 404 Not Found
INFO: 127.0.0.1:56316 - "GET /Forex/Vnd HTTP/1.1" 200 OK
INFO: 127.0.0.1:56331 - "GET /Forex/Vnd HTTP/1.1" 200 OK
INFO: 127.0.0.1:56346 - "GET /Forex/Vnd HTTP/1.1" 200 OK
INFO: 127.0.0.1:56358 - "GET /Forex/Vnd HTTP/1.1" 200 OK
INFO: 127.0.0.1:57126 - "GET /Forex/Vnd HTTP/1.1" 200 OK
INFO: 127.0.0.1:57334 - "GET /Forex/Vnd HTTP/1.1" 200 OK
```

Tiếp theo viết mã code js và html.

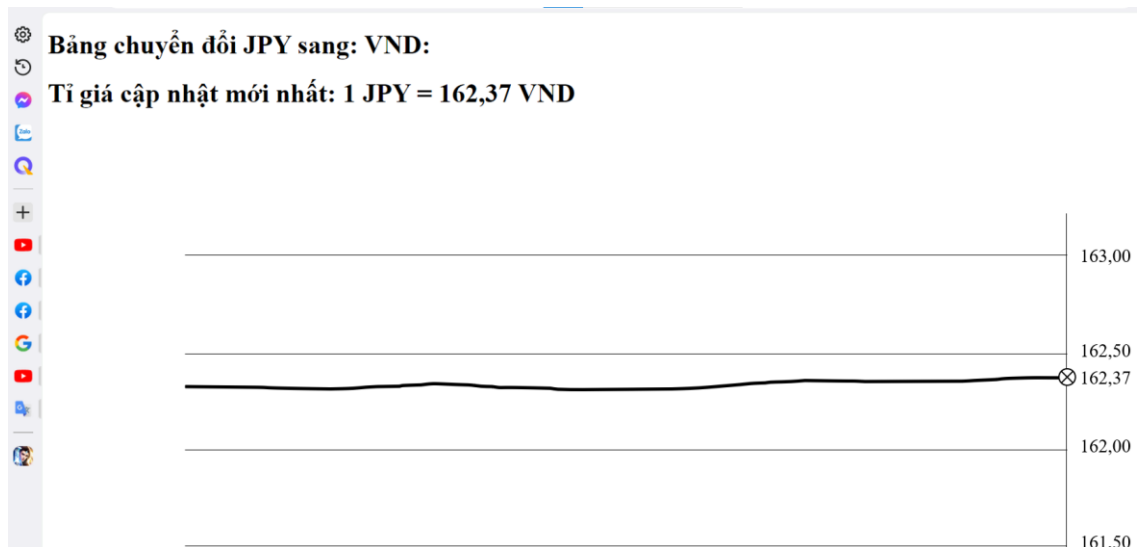


The screenshot shows the Visual Studio IDE with an `index.html` file open. The code is an HTML document that includes a meta charset, a viewport, and a title. It also includes a script tag for `script.js`. The page content includes a heading, a paragraph, and a canvas element for a chart.

```
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>Forex Update</title>
7 </head>
8 <body>
9   <h1>Bảng chuyển đổi VND sang </h1>
10  <h1>Tính đến 1 phút trước </h1>
11  <div id="latestUpdate"></div>
12  <canvas id="MyChart" width="400" height="200"></canvas>
13
14  <!-- Import mã JavaScript -->
15  <script src="script.js"></script>
16 </body>
17 </html>
```

```
File Edit Selection View Go Run Terminal Help
C:\Users\Admin\Desktop\NDT_To_VND-main > NDT_To_VND-main > JS Chart.js > ...
1 document.addEventListener("DOMContentLoaded", function () {
2 // Gửi yêu cầu GET đến API để nhận dữ liệu JSON
3 fetch('api.aspx')
4 .then(response => {
5 if (!response.ok) {
6 throw new Error('Không thể tải dữ liệu');
7 }
8 return response.json();
9 })
10 .then(data => {
11 // Hiển thị thông tin lần cập nhật mới nhất
12 const updateTime = new Date(data.latestUpdate.updatedTime);
13 const formattedTime = updateTime.toLocaleString('vi-VN');
14 document.getElementById('latestUpdate').innerHTML = `
15 <p>Lần cập nhật mới nhất: ${data.latestUpdate.TargetCurrency} vào lúc ${formattedTime}</p>
16 `;
17 // Chuẩn bị dữ liệu cho biểu đồ
18 const forexData = data.forexData.map(item => ({
19 x: new Date(item.updatedTime),
20 y: parseInt(item.TargetCurrency)
21 }));
22 // Vẽ biểu đồ sử dụng Chart.js
23 const ctx = document.getElementById('MyChart').getContext('2d');
24 const forexChart = new Chart(ctx, {
25 type: 'line',
26 data: {
27 datasets: [{
28 label: 'Giá tiền',
29 data: forexData,
30 borderColor: 'blue',
31 fill: false
32 }]
33 },
34 options: {
35 scales: {
36
```

Kết quả đạt được



Mã Qr code link github:



CHƯƠNG III: KẾT LUẬN

Trong bài tập lớn này, em đã có cơ hội triển khai một dự án và hiểu sâu hơn về nhiều công nghệ khác nhau. Đặc biệt, em đã làm quen với ngôn ngữ Python và framework FastAPI, giúp dễ dàng tạo và lấy về các API theo yêu cầu. Em cũng đã học cách cài đặt Node.js, chạy Node-RED, và cấu hình Node-RED để nhận API từ FastAPI rồi chuyển dữ liệu vào SQL Server. Quá trình này cũng bao gồm việc tạo các bảng và stored procedures (SP) để truyền và lưu trữ dữ liệu.

Hơn nữa, em đã tìm hiểu về ASP.NET cùng với các ngôn ngữ và công nghệ web như JavaScript và HTML để có thể hiển thị dữ liệu từ SQL Server lên trang web theo mong muốn. Tất cả những kiến thức và kỹ năng này không chỉ giúp em hoàn thành bài tập lớn mà còn mở rộng hiểu biết của em về lập trình và phát triển web.

Em xin gửi lời cảm ơn sâu sắc đến thầy Đỗ Duy Cốp, người đã hướng dẫn và giúp đỡ em trong suốt quá trình thực hiện dự án này. Nhờ sự chỉ dẫn tận tình của thầy, em đã hoàn thành bài tập lớn một cách thành công và hiệu quả.