

**TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP  
KHOA ĐIỆN TỬ**

**BỘ MÔN: CÔNG NGHỆ THÔNG TIN**



# **BÀI TẬP LỚN MÔN HỌC LẬP TRÌNH PYTHON**

**SINH VIÊN THỰC HIỆN:** Nguyễn Thế Định

**LỚP:** K56KMT.01

**GIÁO VIÊN HƯỚNG DẪN:** ĐỖ DUY CẬP

**Thái Nguyên 2024**

**PHIẾU GIAO ĐỀ TÀI BÀI TẬP LỚN MÔN HỌC**  
**LẬP TRÌNH PYTHON**

**I. Thông tin sinh viên**

Sinh viên thực hiện: Nguyễn Thế Định

Mã SV: K205480106027

**II. Tên đề tài**

**XÂY DỰNG WEBSITE THEO DÕI TỶ GIÁ NHÂN DÂN TỆ**

**III. Mục tiêu**

- Lấy dữ liệu đồng nhân dân tệ
- Xử lý dữ liệu :sử dụng FastAPI và Node-RED, sau đó lưu vào cơ sở dữ liệu.
- Xây dựng trang web

**IV. Nội dung thực hiện**

1. Sử dụng API của các nguồn dữ liệu từ forex (có update realtime)
2. Tạo một cơ sở dữ liệu trong SQL Server để lưu trữ dữ liệu về
3. 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.
4. Sử dụng FastAPI để tạo các endpoint API để truy xuất dữ liệu từ cơ sở dữ liệu.
5. Sử dụng các công nghệ front-end (HTML, CSS, JavaScript, React.js) để xây dựng giao diện người dùng.
6. Tạo biểu đồ thể hiện sự thay đổi của đồng nhân dân tệ

**V. Ngày giao nhiệm vụ: 15/5/20204**

**VI. Ngày hoàn thành: 25/5/2024**

**VII. Giáo viên hướng dẫn: Đỗ Duy Cốp**

**NHẬN XÉT CỦA GIÁO VIÊN**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Thái Nguyên, ngày.....tháng.....năm 2024

**CHỮ KÝ CỦA GIÁO VIÊN**

*(ký, ghi rõ họ tên)*



## I. TỔNG QUAN CHUNG

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 nhân dân tẻ 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 nhân dân tẻ. 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 nhân dân tẻ (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 Node-RED 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 nhân dân tẻ và biểu đồ thể hiện sự thay đổi của đồng đó

### 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:

#### a. 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.

#### b. 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.

#### c. 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.

#### d. 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 dữ liệu số ca nhiễm COVID-19 một cách 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.

## II. CƠ SỞ LÝ THUYẾT

### 2.1. SQL Server Management Studio

SQL Server Management Studio (SSMS) là một ứng dụng phần mềm của Microsoft, được thiết kế để quản lý và tương tác với các cơ sở dữ liệu SQL Server. Được phát triển từ năm 2005, SSMS là một công cụ quản lý cơ bản và quan trọng cho các quản trị viên cơ sở dữ liệu, nhà phát triển và các chuyên gia dữ liệu.

SQL Server được phát triển lần đầu tiên vào năm 1989 bởi Microsoft, hợp tác với Sybase và Ashton-Tate. Từ đó, nó đã trải qua nhiều phiên bản cải tiến với những tính năng và khả năng mới, trở thành một trong những hệ quản trị cơ sở dữ liệu phổ biến nhất trên thế giới.



SSMS cung cấp một giao diện người dùng đồ họa (GUI) thân thiện và dễ sử dụng cho việc quản lý cơ sở dữ liệu SQL Server. Giao diện này cho phép người dùng thực hiện các tác vụ quản lý dữ liệu một cách dễ dàng và hiệu quả. SSMS cung cấp một loạt các công cụ quản lý tích hợp, cho phép người dùng thực hiện các tác vụ như tạo, sửa đổi và xóa cơ sở dữ liệu, bảng, chỉ mục và thủ tục lưu trữ. Nó cũng cho phép quản trị viên sao lưu và phục hồi dữ liệu, kiểm tra và theo dõi hiệu suất, và quản lý bảo mật. SSMS cho phép người dùng thực hiện các truy vấn SQL và xem dữ liệu từ các bảng trong cơ sở dữ liệu. Nó cung cấp một trình soạn thảo truy vấn mạnh mẽ với tính năng gợi ý cú pháp và điều hướng thông minh giúp tăng hiệu suất lập trình.

SQL Server cung cấp cho người dùng các công cụ và tính năng để quản lý, lưu trữ, xử lý các truy vấn dữ liệu, kiểm soát truy cập, xử lý giao dịch và hỗ trợ tích hợp dữ liệu từ nhiều nguồn khác nhau.

Ngoài ra, SQL Server cũng cung cấp các công cụ để tạo báo cáo, phân tích và quản lý cơ sở dữ liệu trực quan thông qua giao diện người dùng hoặc các script lệnh SQL. SQL Server được xây dựng dựa trên SQL, một ngôn ngữ lập trình tiêu chuẩn để tương tác với cơ sở dữ liệu quan hệ. SQL Server được liên kết với Transact-SQL hoặc T-SQL, triển khai SQL của Microsoft có bổ sung một tập hợp các cấu trúc lập trình độc quyền.

### 2.2. Ngôn ngữ lập trình Python

Python là một ngôn ngữ lập trình đa mục đích, dễ học và mạnh mẽ, được phát triển bởi Guido van Rossum và ra mắt lần đầu vào năm 1991. Với cú pháp đơn giản và gần gũi với ngôn ngữ tự nhiên, Python là một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới, được sử dụng rộng rãi trong các lĩnh vực khác nhau từ phát triển web, khoa học dữ liệu đến trí tuệ nhân tạo.



Python không chỉ dễ học mà còn linh hoạt và mở rộng, hỗ trợ nhiều phong cách lập trình và tích hợp tốt với các ngôn ngữ khác như C/C++, Java và .NET. Hệ sinh thái phong phú của Python cung cấp các thư viện và framework đa dạng, giúp lập trình viên dễ dàng phát triển các ứng dụng và dự án.

Điểm nổi bật của Python là cộng đồng lớn mạnh, với hàng triệu lập trình viên trên khắp thế giới, sẵn sàng chia sẻ kiến thức và kinh nghiệm. Nhờ vào điều này, Python không chỉ là một ngôn ngữ lập trình mà còn là một cộng đồng và một triển khai tri thức phong phú, đóng vai trò quan trọng trong việc giải quyết các thách thức hiện đại trong ngành công nghiệp và khoa học.

### **2.3. Visual Studio 2022**

Microsoft Visual Studio là môi trường phát triển tích hợp (IDE) được thiết kế dành cho giới lập trình viên và các nhà phát triển ứng dụng. Đây là công cụ hỗ trợ phát triển phần mềm mạnh mẽ của Microsoft, cho phép người dùng viết, dịch mã và gỡ lỗi các ứng dụng dựa trên nhiều ngôn ngữ lập trình khác nhau như C++, C#, Visual Basic, Python, JavaScript... Visual Studio bao gồm một trình biên tập mã nguồn, các công cụ gỡ lỗi và xây dựng ứng dụng đa nền tảng. Nó giúp tăng năng suất và hiệu quả công việc cho các lập trình viên.

Microsoft Visual Studio nổi bật với khả năng hỗ trợ một loạt các ngôn ngữ lập trình, bao gồm JavaScript, TypeScript, Python, C#, Java, Go, Ruby... Điều này biến nó trở thành công cụ lý tưởng cho các nhà phát triển làm việc trên nhiều dự án với đa ngôn ngữ lập trình. Ngoài ra, khả năng hỗ trợ đa ngôn ngữ của Microsoft Visual Studio còn giúp giới lập trình viên dễ dàng chuyển đổi giữa các ngôn ngữ và dự án mà không cần phải thay đổi môi trường làm việc, từ đó tiết kiệm thời gian, đồng thời tối ưu hiệu quả công việc.

Visual Studio 2022 là một phiên bản mới nhất của môi trường phát triển tích hợp (IDE) Visual Studio, được phát triển bởi Microsoft. Được công bố vào tháng 11 năm 2021, Visual Studio 2022 mang đến nhiều cải tiến và tính năng mới so với các phiên bản trước đó.





Visual Studio 2022 đi kèm với hỗ trợ đầy đủ cho .NET 6, bao gồm C# 10 và F# 6. .NET 6 là một phiên bản mới của nền tảng phát triển phần mềm .NET, với nhiều cải tiến về hiệu suất, độ ổn định và tính năng mới. Visual Studio 2022 tích hợp với GitHub Codespaces, cho phép bạn phát triển ứng dụng trực tiếp từ trình duyệt web mà không cần cài đặt môi trường phát triển trên máy cục bộ.

Visual Studio 2022 được tối ưu hóa về hiệu suất, bao gồm tăng tốc khởi động và thời gian phản hồi của các tính năng và công cụ.

Visual Studio 2022 là một bước tiến quan trọng trong việc cung cấp một môi trường phát triển hiệu quả và mạnh mẽ cho các nhà phát triển phần mềm, với sự hỗ trợ đa nền tảng và tích hợp sâu sắc với các công nghệ và dịch vụ mới.

## **2.4. Node\_red**

Node-RED là một công cụ mã nguồn mở được phát triển bởi IBM và cung cấp một giao diện trực quan để kết nối các thiết bị, dịch vụ và ứng dụng một cách linh hoạt và dễ dàng. Nó được xây dựng dựa trên Node.js và sử dụng một giao diện trực quan dựa trên trình duyệt để tạo, quản lý và triển khai các luồng làm việc (flow) dựa trên sự kết hợp của các "nút" và "luồng".

Các nút trong Node-RED đại diện cho các chức năng hoặc dịch vụ cụ thể, và chúng có thể được kéo và thả vào khung làm việc để tạo ra các luồng làm việc. Mỗi nút thường thực hiện một chức năng nhất định, từ xử lý dữ liệu đến gửi và nhận thông điệp qua các giao thức mạng khác nhau.



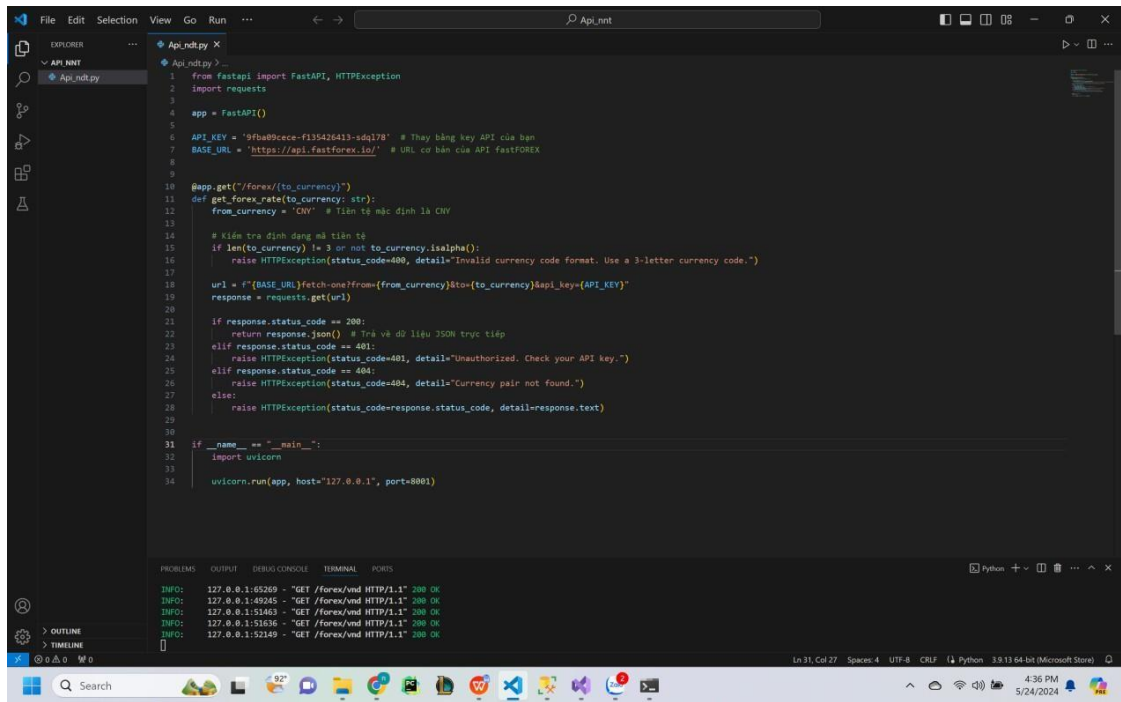
Node-RED được sử dụng rộng rãi trong Internet of Things (IoT) và trong các ứng dụng tự động hóa, nơi nó có thể giúp kết nối và tự động hóa các thiết bị và dịch vụ từ nhiều nhà sản xuất khác nhau. Nó cũng thích hợp cho việc xử lý dữ liệu thời gian thực và tích hợp các dịch vụ web khác nhau.

Node-RED cung cấp một cộng đồng lớn và sôi động, với nhiều nút và gói mở rộng được phát triển và chia sẻ miễn phí. Điều này giúp người dùng mở rộng và tùy chỉnh Node-RED theo nhu cầu và yêu cầu cụ thể của họ.

### **III. NỘI DUNG THỰC HIỆN**

Sau đây là các bước thực hiện đề tài của em:

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



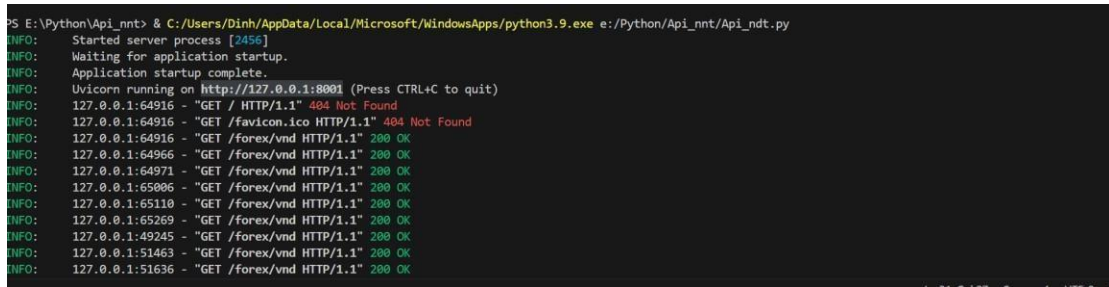
The screenshot shows a VS Code editor window with a file named `Api.ndt.py` open. The code is a FastAPI application that uses the `requests` library to fetch data from the `fastforex.io` API. The application is configured with an API key and a base URL. It defines a `get_forex_rate` endpoint that takes a `to_currency` parameter and returns the exchange rate for a specific currency pair (defaulting to CNY). The application is run using `uvicorn` on `127.0.0.1:8001`.

```
1 from fastapi import FastAPI, HTTPException
2 import requests
3
4 app = FastAPI()
5
6 API_KEY = '9fba09cece-f135426413-sdq178' # 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 = 'CNY' # Tiền tệ mặc định là CNY
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.")
25     elif response.status_code == 404:
26         raise HTTPException(status_code=404, detail="Currency pair not found.")
27     else:
28         raise HTTPException(status_code=response.status_code, detail=response.text)
29
30
31 if __name__ == "__main__":
32     import uvicorn
33
34     uvicorn.run(app, host="127.0.0.1", port=8001)
```

The terminal output shows the application running successfully on `127.0.0.1:8001`. It lists several incoming requests and their responses:

```
INFO: 127.0.0.1:8000 - "GET /forex/vnd HTTP/1.1" 200 OK
INFO: 127.0.0.1:49245 - "GET /forex/vnd HTTP/1.1" 200 OK
INFO: 127.0.0.1:51463 - "GET /forex/vnd HTTP/1.1" 200 OK
INFO: 127.0.0.1:51636 - "GET /forex/vnd HTTP/1.1" 200 OK
INFO: 127.0.0.1:52349 - "GET /forex/vnd HTTP/1.1" 200 OK
```

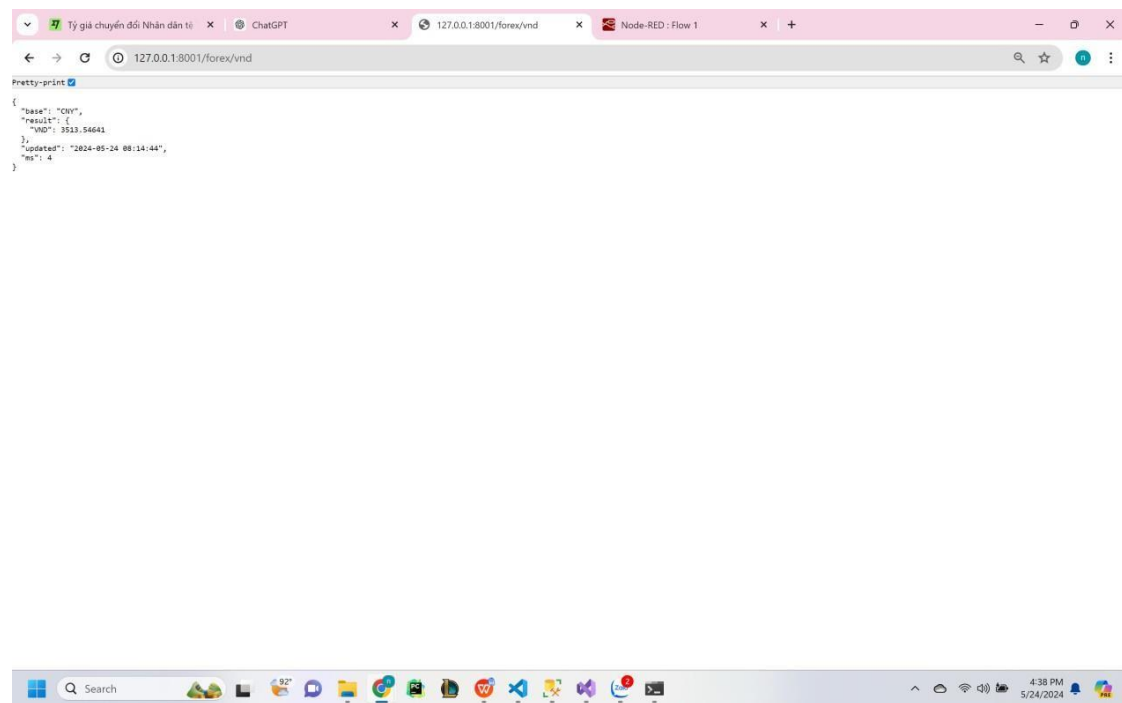
## Chạy fash api



The screenshot shows a Windows command prompt window with the following commands and output:

```
PS E:\Python\api_nnt> & C:/Users/Dinh/AppData/Local/Microsoft/WindowsApps/python3.9.exe e:/Python/Api_nnt/Api.ndt.py
INFO: Started server process [2456]
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: 127.0.0.1:64916 - "GET / HTTP/1.1" 404 Not Found
INFO: 127.0.0.1:64916 - "GET /favicon.ico HTTP/1.1" 404 Not Found
INFO: 127.0.0.1:64916 - "GET /forex/vnd HTTP/1.1" 200 OK
INFO: 127.0.0.1:64966 - "GET /forex/vnd HTTP/1.1" 200 OK
INFO: 127.0.0.1:64971 - "GET /forex/vnd HTTP/1.1" 200 OK
INFO: 127.0.0.1:65006 - "GET /forex/vnd HTTP/1.1" 200 OK
INFO: 127.0.0.1:65110 - "GET /forex/vnd HTTP/1.1" 200 OK
INFO: 127.0.0.1:65269 - "GET /forex/vnd HTTP/1.1" 200 OK
INFO: 127.0.0.1:49245 - "GET /forex/vnd HTTP/1.1" 200 OK
INFO: 127.0.0.1:51463 - "GET /forex/vnd HTTP/1.1" 200 OK
INFO: 127.0.0.1:51636 - "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

```
node-red
Microsoft Windows [Version 10.0.22621.3593]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Dinh>node-red
24 May 16:21:20 - [info]

Welcome to Node-RED
=====

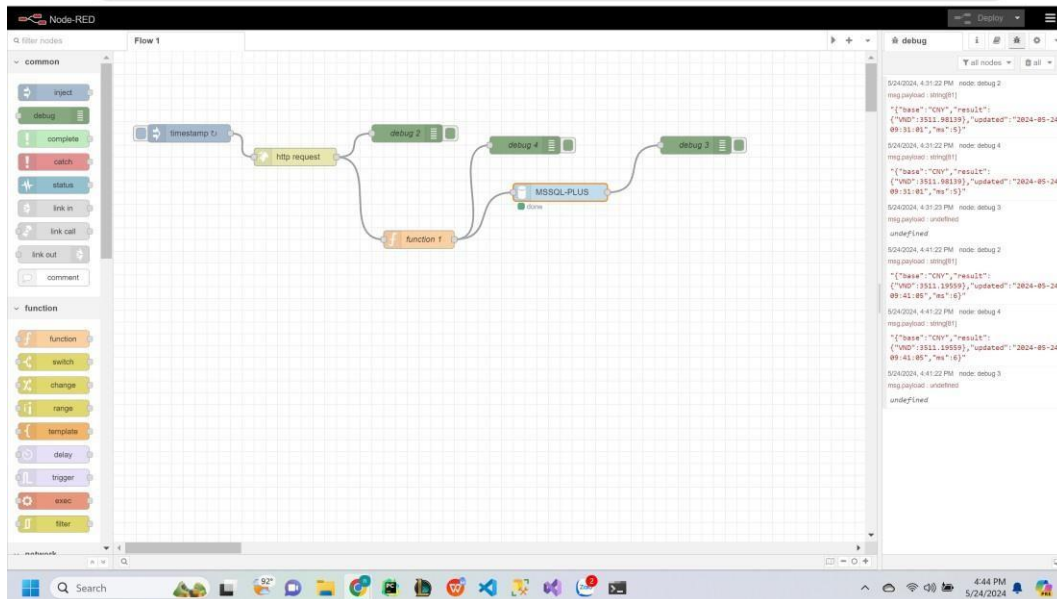
24 May 16:21:20 - [info] Node-RED version: v3.1.9
24 May 16:21:20 - [info] Node.js version: v20.13.1
24 May 16:21:20 - [info] Windows_NT 10.0.22621 x64 LE
24 May 16:21:21 - [info] Loading palette nodes
24 May 16:21:21 - [info] Settings file : C:\Users\Dinh\.node-red\settings.js
24 May 16:21:21 - [info] Context store : 'default' [module=memory]
24 May 16:21:21 - [info] User directory : \Users\Dinh\.node-red
24 May 16:21:21 - [warn] Projects disabled : editorTheme.projects.enabled=false
24 May 16:21:21 - [info] Flows file : \Users\Dinh\.node-red\flows.json
24 May 16:21:21 - [info] Server now running at http://127.0.0.1:1880/
24 May 16:21:21 - [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
```

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

msg. payload = timestamp

msg. topic = a\_z

+ add inject now

☐ Inject once after 0.1 seconds, then

Repeat interval

every 1 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

+ add

👤 Name

Name

📄

○ Enabled

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

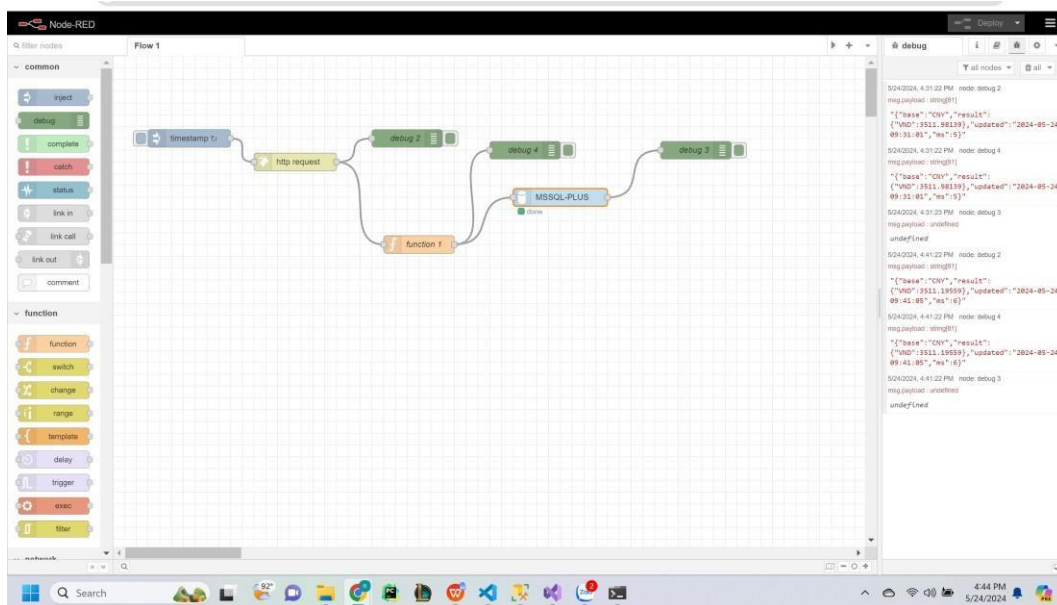
▼

📄

☐ Enabled

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





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

```

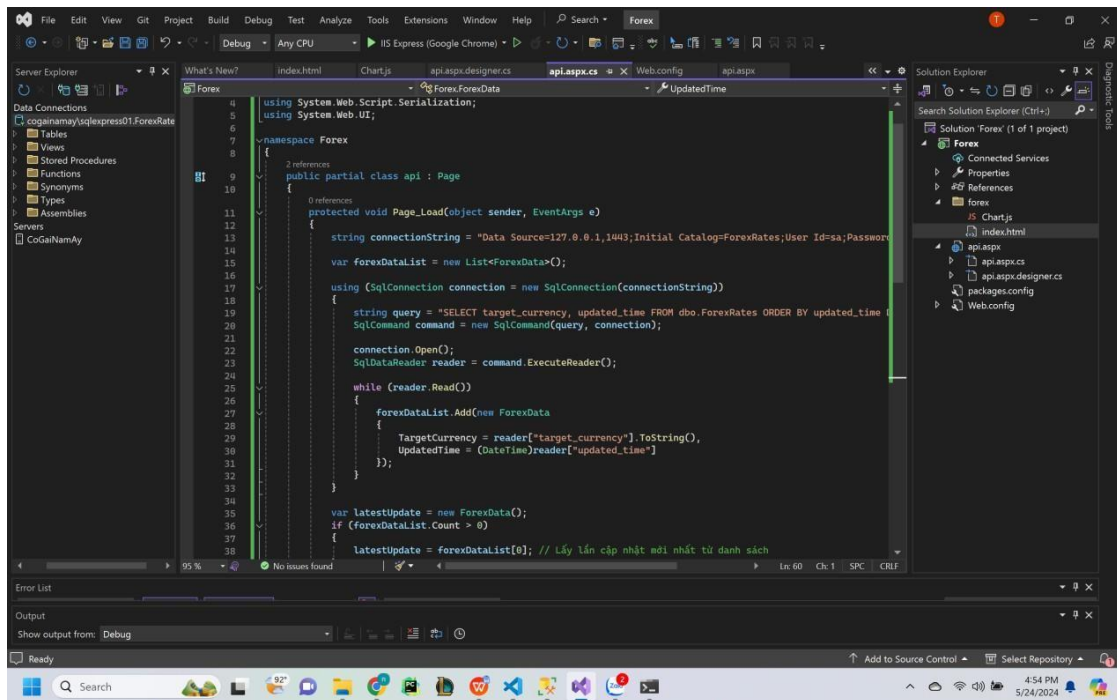
--
ALTER PROCEDURE [dbo].[InsertDulieuJson]
    @json NVARCHAR(MAX) = null
AS
BEGIN
    DECLARE @base_currency NVARCHAR(50);
    DECLARE @target_currency DECIMAL(18, 0);
    DECLARE @exchange_rate FLOAT;
    DECLARE @updated_time DATETIME;

    -- 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'));

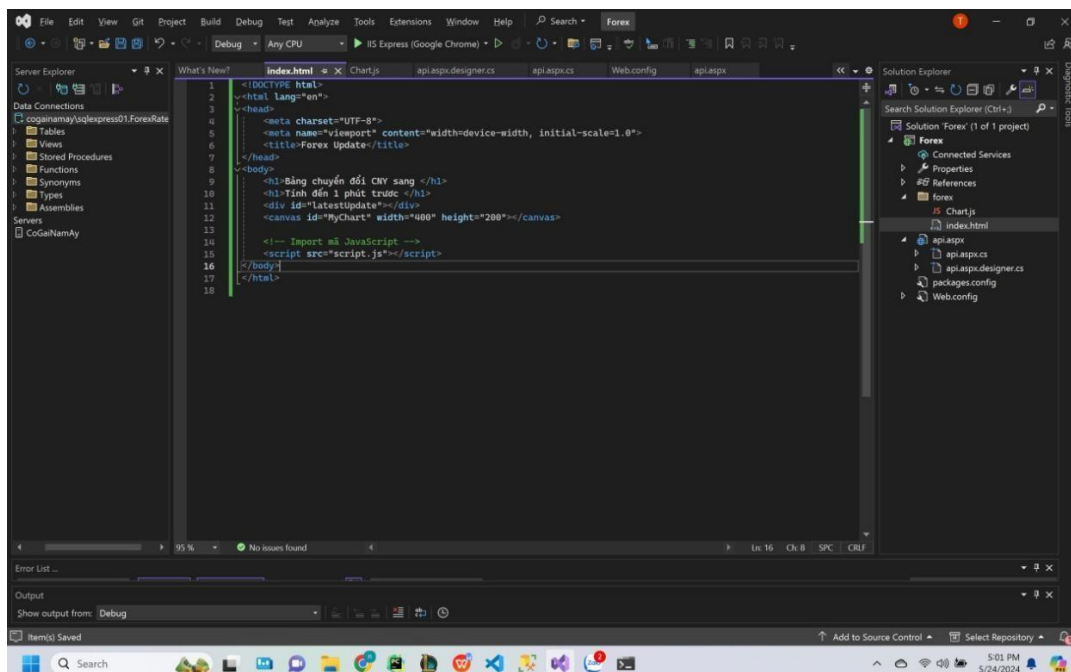
    -- Chèn dữ liệu vào bảng ExchangeRates
    INSERT INTO ForexRates(base_currency, target_currency, exchange_rate, updated_time)
    VALUES (@base_currency, @target_currency, @exchange_rate, @updated_time);
END;

```

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

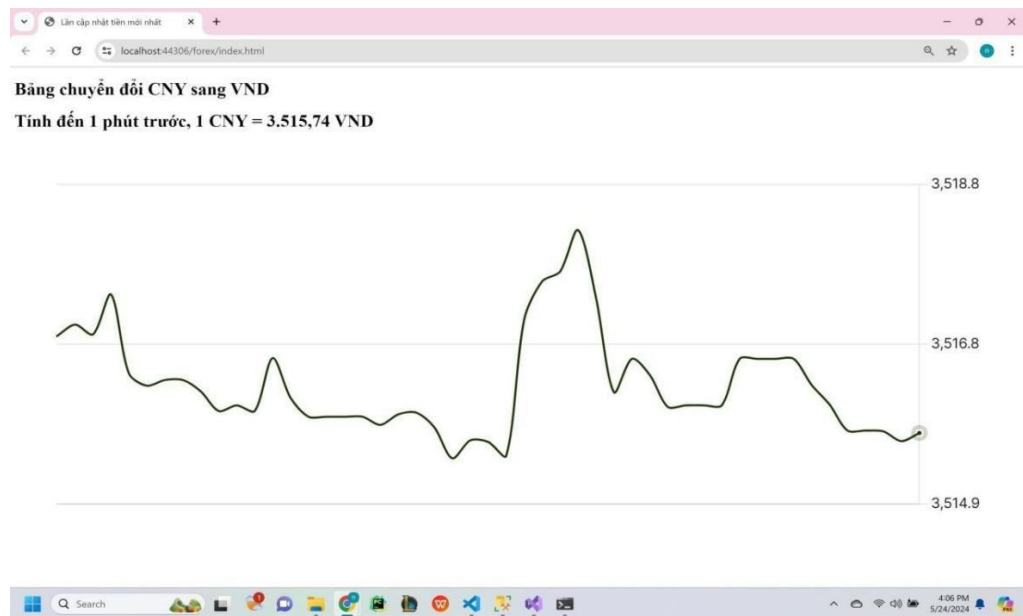


Tiếp theo viết mã code js và html



```
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
18      // Chuẩn bị dữ liệu cho biểu đồ
19      const forexData = data.forexData.map(item => ({
20        x: new Date(item.updatedTime),
21        y: parseInt(item.targetCurrency)
22      }));
23
24      // Vẽ biểu đồ sử dụng Chart.js
25      const ctx = document.getElementById('MyChart').getContext('2d');
26      const forexChart = new Chart(ctx, {
27        type: 'line',
28        data: {
29          datasets: [{
30            label: 'Giải tiền',
31            data: forexData,
32            borderColor: 'blue',
33            fill: false
34          }]
35        },
36        options: {
```

Kết quả đạt được



## Kết Luận

Trong bài tập lớn này em đã biết được cách triển khai 1 project hiểu thêm về ngôn ngữ python đặc biệt là fastapi cho phép chúng ta dễ dàng tạo ra , hoặc lấy về các api mình muốn.

Hiểu thêm về nodejs cách cài đặt nodejs , chạy node-red , cấu hình cho node-red để nhận api từ fastapi rồi lại chuyển về trong sql sever , biết cách tạo các bảng , Sp để chuyển và chứa dữ liệu

Biết thêm về ASP.Net , js,html để đưa dữ liệu từ sql lên web theo mong muốn

Em cảm ơn thầy Đỗ Duy Cốp đã giúp em hiểu và hoàn thành bài tập lớn này.