

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: Dương Thị Bích Nguyệt

Lớp: 56KMT.01

Giáo viên hướng dẫn: Đỗ Duy Cốp

Thái Nguyên – 2024

## BÀI TẬP LỚN

**MÔN HỌC: LẬP TRÌNH PYTHON**  
**BỘ MÔN : CÔNG NGHỆ THÔNG TIN**

*Sinh viên: Dương Thị Bích Nguyệt.....*

*Lớp: 56KMT.01*

*Ngành: Kỹ thuật máy tính*

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

*Ngày giao đề: 15/05/2024..... Ngày hoàn thành: 26/05/2024 .....*

**Tên đề tài : Xây dựng web theo dõi tỉ giá hối đoái toàn cầu**

**Yêu cầu :**

- Tạo cơ sở dữ liệu SQL để lưu trữ dữ liệu về tỉ giá hối đoái toàn cầu, bao gồm thông tin như loại tiền, thay đổi tỉ giá, thời gian.
- Sử dụng FastAPI để tạo các endpoint API để truy xuất dữ liệu tiền tệ từ cơ sở dữ liệu.
- Sử dụng Node-RED để kết nối và lấy dữ liệu từ các nguồn khác nhau như API của các trang web tỷ giá hối đoái toàn cầu hoặc các dịch vụ cung cấp dữ liệu tỷ giá hối đoái.
- Hiện thị biểu đồ để biểu diễn biến động của tỷ giá

**GIÁO VIÊN HƯỚNG DẪN**

*(Ký và ghi rõ họ tên)*

## NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

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

**GIÁO VIÊN HƯỚNG DẪN**

*(Ký ghi rõ họ tên)*

# MỤC LỤC

LỜI NÓI ĐẦU .....	5
DANH MỤC HÌNH ẢNH .....	6
CHƯƠNG 1. TỔNG QUAN.....	7
1.1. Đặt vấn đề .....	7
1.2. Mục tiêu.....	7
1.3. Hướng giải quyết.....	7
1.4. Giới hạn đề tài .....	8
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT .....	9
2.1. SQL Server Management Studio .....	9
2.2. Ngôn ngữ lập trình PyThon.....	10
2.3. Visual Studio 2022.....	11
2.4. Node_red .....	12
CHƯƠNG 3. NỘI DUNG THỰC HIỆN .....	14
3.1. Tạo cơ sở dữ liệu và bảng .....	14
3.2. Thiết lập PyThon + FastAPI .....	14
3.3. Thiết lập Node-Red .....	15
3.4. Viết store Procedure .....	19
3.5. Xây dựng giao diện người dùng.....	20
3.6. Kết quả trên web .....	20
CHƯƠNG 5. KẾT LUẬN.....	21
LỜI CẢM ƠN .....	22
TÀI LIỆU THAM KHẢO.....	23

## LỜI NÓI ĐẦU

Trong thế giới kinh doanh ngày nay, việc theo dõi và dự đoán biến động của tỷ giá hối đoái là một phần không thể thiếu của chiến lược tài chính của mọi tổ chức và cá nhân. Tỷ giá hối đoái không chỉ ảnh hưởng đến giá trị tài sản và chi phí, mà còn có thể tác động mạnh mẽ đến cả nền kinh tế quốc gia.

Trong bối cảnh này, việc có một công cụ theo dõi tỷ giá hối đoái toàn cầu trở nên cực kỳ quan trọng. Điều này không chỉ giúp các nhà đầu tư và doanh nghiệp định hình chiến lược giao dịch, mà còn cho phép người dùng thông thạo quản lý rủi ro và tận dụng cơ hội đầu tư trong môi trường biến động.

Web theo dõi tỷ giá hối đoái toàn cầu với mục tiêu cung cấp một nền tảng web đáng tin cậy và tiện ích, cho phép người dùng theo dõi tỷ giá hối đoái từ các thị trường quốc tế hàng đầu, cũng như cung cấp các công cụ phân tích và thông tin chi tiết để hỗ trợ quyết định đầu tư.

## **DANH MỤC HÌNH ẢNH**

- Hình 1. Ngôn ngữ lập trình Python
- Hình 2. Visual Studio 2022
- Hình 3. Node-Red
- Hình 4. Bảng ExchangeRatesDB
- Hình 5. File main.py
- Hình 6. Khởi chạy FastAPI
- Hình 7. Chuỗi dạng JSON
- Hình 8. Node JS
- Hình 9. Chạy Node-Red trên Command Prompt
- Hình 10. Thiết lập HTTP request node
- Hình 11. Thiết lập function node
- Hình 12. MSSQL node
- Hình 13. Edit MSSQL node
- Hình 14. Liên kết các node
- Hình 15. Dữ liệu
- Hình 16. store Procedure
- Hình 17. Index.html
- Hình 18. Biểu đồ

# CHƯƠNG 1. TỔNG QUAN

## 1.1. Đặt vấn đề

Trong nền kinh tế toàn cầu hiện nay, tỷ giá hối đoái là một trong những yếu tố kinh tế quan trọng nhất, ảnh hưởng mạnh mẽ đến thương mại quốc tế, đầu tư và chính sách tiền tệ của các quốc gia. Sự biến động của tỷ giá hối đoái không chỉ ảnh hưởng đến giá cả hàng hóa xuất nhập khẩu mà còn tác động đến lợi nhuận của các doanh nghiệp đa quốc gia, giá trị tài sản, và tình hình tài chính của các cá nhân.

Với sự phát triển nhanh chóng của công nghệ thông tin và truyền thông, việc tiếp cận thông tin tỷ giá hối đoái trở nên dễ dàng hơn bao giờ hết. Tuy nhiên, vấn đề đặt ra là làm sao có thể thu thập, xử lý và hiển thị thông tin tỷ giá một cách nhanh chóng, chính xác và tiện lợi cho người dùng. Đặc biệt, đối với các nhà đầu tư, doanh nghiệp và cá nhân có nhu cầu theo dõi tỷ giá thường xuyên, một nền tảng trực tuyến hiệu quả và đáng tin cậy là vô cùng cần thiết.

## 1.2. Mục tiêu

Mục tiêu chính của đề tài này là xây dựng một trang web theo dõi tỷ giá hối đoái toàn cầu với các tính năng nổi bật sau:

- Cung cấp thông tin tỷ giá hối đoái theo thời gian thực: Đảm bảo người dùng có thể theo dõi sự biến động của tỷ giá từ các thị trường tài chính.
- Hiển thị dữ liệu lịch sử: Cung cấp các biểu đồ và công cụ phân tích giúp người dùng hiểu rõ hơn về xu hướng tỷ giá trong quá khứ và dự báo tương lai..

## 1.3. Hướng giải quyết

Để hiện thực hóa các mục tiêu trên, đề tài sẽ được triển khai theo các bước chính sau:

- Thu thập và xử lý dữ liệu: Sử dụng API từ ExchangeRate-API. Dữ liệu sẽ được cập nhật liên tục và xử lý để đảm bảo tính chính xác và kịp thời.
- Phát triển hệ thống hiển thị và phân tích: Sử dụng HTML, CSS, JavaScript để phát triển giao diện người dùng. Các công cụ phân tích

dữ liệu sẽ được tích hợp để cung cấp biểu đồ, báo cáo và dự báo xu hướng.

- Thiết kế giao diện người dùng: Tập trung vào việc thiết kế giao diện thân thiện, dễ sử dụng, đáp ứng tốt trên cả máy tính và thiết bị di động.

#### **1.4. Giới hạn đề tài**

Mặc dù bài làm đã cơ bản hoàn thành được nhưng yêu cầu đặt ra, nhưng bên cạnh đó vẫn còn những thiếu sót nhất định. Vì vậy hệ thống sẽ cần khắc phục những hạn chế để trở nên hoàn thiện và đáp ứng được như cầu người dùng cũng như hướng phát triển đề tài



## CHƯƠNG 2. 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 gì?

Python là ngôn ngữ lập trình máy tính bậc cao thường được sử dụng để xây dựng trang web và phần mềm, tự động hóa các tác vụ và tiến hành phân tích dữ liệu. Python là ngôn ngữ có mục đích chung, nghĩa là nó có thể được sử dụng để tạo nhiều chương trình khác nhau và không chuyên biệt cho bất kỳ vấn đề cụ thể nào.



Hình 1. Ngôn ngữ lập trình Python

- Ứng dụng của Python

Python thường được sử dụng để phát triển trang web và phần mềm, tự động hóa tác vụ, phân tích dữ liệu và trực quan hóa dữ liệu. Vì tương đối dễ học, Python đã được nhiều người không phải là lập trình viên như kế toán và nhà khoa học áp dụng cho nhiều công việc hàng ngày, chẳng hạn như tổ chức tài chính.

- Đặc tính của Python

- Ngôn ngữ thông dịch: Python được xử lý trong thời gian chạy bởi Trình thông dịch Python. Ngôn ngữ hướng đối tượng: Nó hỗ trợ các tính năng và kỹ thuật lập trình hướng đối tượng.
- Ngôn ngữ lập trình tương tác: Người dùng có thể tương tác trực tiếp với trình thông dịch python để viết chương trình.
- Ngôn ngữ dễ học: Python rất dễ học, đặc biệt là cho người mới bắt đầu.
- Cú pháp đơn giản: Việc hình thành cú pháp Python rất đơn giản và dễ hiểu, điều này cũng làm cho nó trở nên phổ biến.
- Dễ đọc: Mã nguồn Python được xác định rõ ràng và có thể nhìn thấy bằng mắt.
- Di động: Mã Python có thể chạy trên nhiều nền tảng phần cứng có cùng giao diện.
- Có thể mở rộng: Người dùng có thể thêm các mô-đun cấp thấp vào trình thông dịch Python.
- Có thể cải tiến: Python cung cấp một cấu trúc cải tiến để hỗ trợ các chương trình lớn sau đó là shell-script.

### 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.



Hình 2. Visual Studio 2022

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.

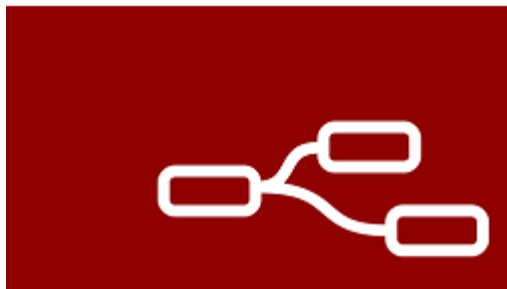
Visual Studio 2022 là một bước tiến quan trọng trong việc cung cấp một môi trường phát triển hiệu quả và mạnh mẽ cho các nhà phát triển phần mềm, với sự hỗ trợ đa nền tảng và tích hợp sâu sắc với các công nghệ và dịch vụ mới.

## 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



Hình 3. Node-Red

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.

## CHƯƠNG 3. NỘI DUNG THỰC HIỆN

### 3.1. Tạo cơ sở dữ liệu và bảng

Đầu tiên em tạo một cơ sở dữ liệu và bảng để lưu trữ dữ liệu tỉ giá hối đoái bằng SQL Server Management Studio

	Column Name	Data Type	Allow Nulls
🔑	ID	int	<input type="checkbox"/>
	CurrencyPair	nvarchar(10)	<input checked="" type="checkbox"/>
	Rate	decimal(18, 6)	<input checked="" type="checkbox"/>
	Timestamp	datetime	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Hình 4. Bảng ExchangeRatesDB

### 3.2. Thiết lập PyThon + FastAPI

- Cài đặt FastAPI và các thư viện cần thiết bằng lệnh “*pip install fastapi uvicorn sqlalchemy pyodbc*”
- Tạo một file Python có tên là main.py và sử dụng FastAPI để lấy dữ liệu tỉ giá hối đoái từ một API công cộng và trả về dưới dạng JSON. Cấu hình kết nối với MSSQL.

```
main.py X
main.py > ...
1 from fastapi import FastAPI
2 from fastapi.responses import FileResponse
3 import pyodbc
4
5 app = FastAPI()
6
7 # Cấu hình kết nối MSSQL
8 server = '127.0.0.1,1443'
9 database = 'ExchangeRatesDB'
10 username = 'sa'
11 password = '123'
12 connection_string = f'DRIVER={{ODBC Driver 17 for SQL Server}};SERVER={server};DATABASE={database};UID={username};PWD={password}'
13
14 @app.get("/")
15 async def read_root():
16     return {"message": "Welcome to the Exchange Rates API"}
17
18 @app.get("/rates")
19 def read_rates():
20     with pyodbc.connect(connection_string) as conn:
21         cursor = conn.cursor()
22         cursor.execute("SELECT * FROM ExchangeRates ORDER BY Timestamp DESC")
23         rows = cursor.fetchall()
24         return [{"CurrencyPair": row.CurrencyPair, "Rate": row.Rate, "Timestamp": row.Timestamp} for row in rows]
25
26 @app.get("/favicon.ico", include_in_schema=False)
27 async def favicon():
28     return FileResponse("path/to/your/favicon.ico")
29
30 if __name__ == "__main__":
31     import uvicorn
32     uvicorn.run(app, host="localhost", port=8000)
33
```

Hình 5. File main.py

- Khởi chạy FastAPI bằng lệnh “*uvicorn main:app --reload*”

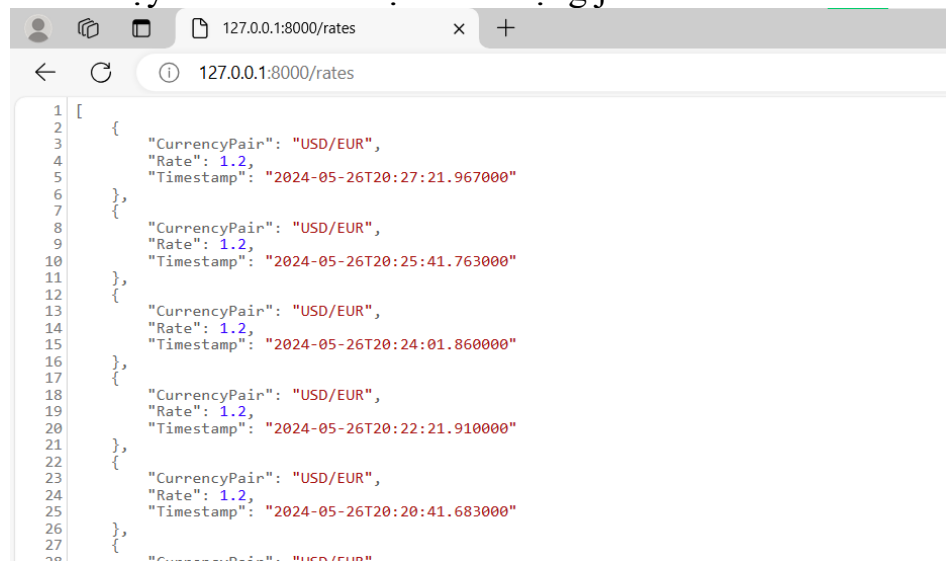
```

PS D:\PYTHON> uvicorn main:app --reload
>>
INFO:      Will watch for changes in these directories: ['D:\\PYTHON']
INFO:      Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO:      Started reloader process [9700] using WatchFiles
INFO:      Started server process [13764]
INFO:      Waiting for application startup.
INFO:      Application startup complete.

```

Hình 6. Khởi chạy FastAPI

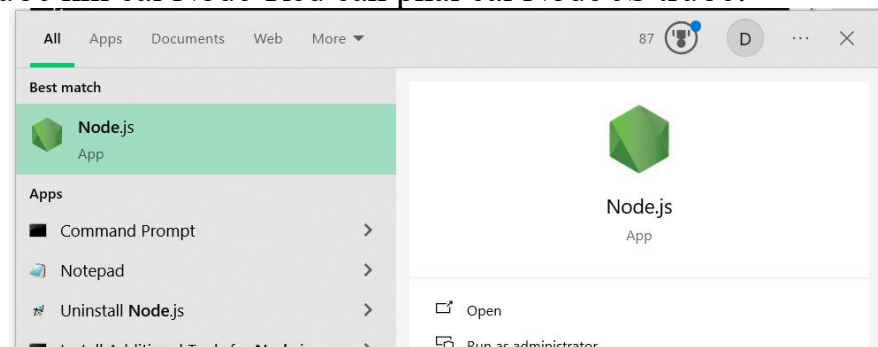
Sau khi khởi chạy nó sẽ trả về một chuỗi dạng json trên local của mình



Hình 7. Chuỗi dạng JSON

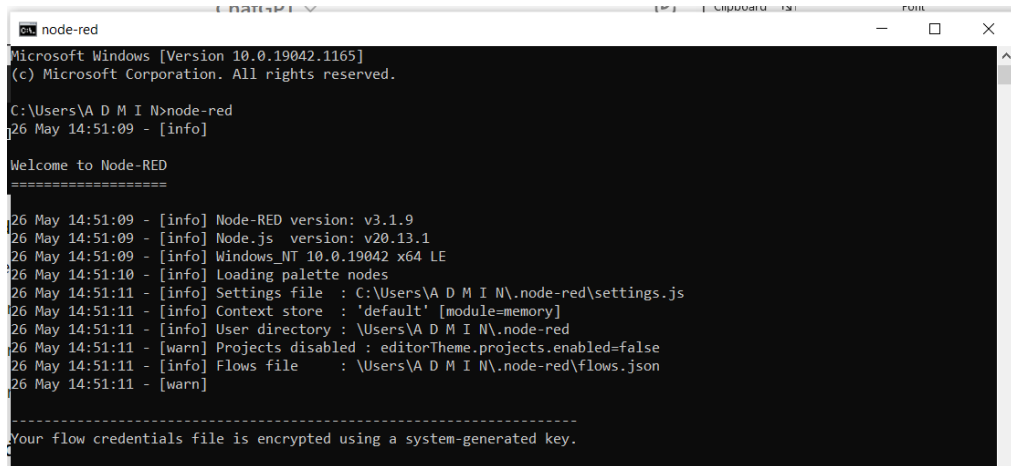
### 3.3. Thiết lập Node-Red

- Trước khi cài Node-Red cần phải cài Node JS trước:



Hình 8. Node JS

- Sau khi đã cài xong Node JS, em mở Command Prompt trên máy để bắt đầu cài đặt Node-Red, sử dụng lệnh “*npm install -g node-red node-red*” để cài đặt. Cài đặt xong nhập “*node-red*” để chạy Node-Red.



```
node-red
Microsoft Windows [Version 10.0.19042.1165]
(c) Microsoft Corporation. All rights reserved.

C:\Users\A D M I N>node-red
26 May 14:51:09 - [info]

Welcome to Node-RED
=====

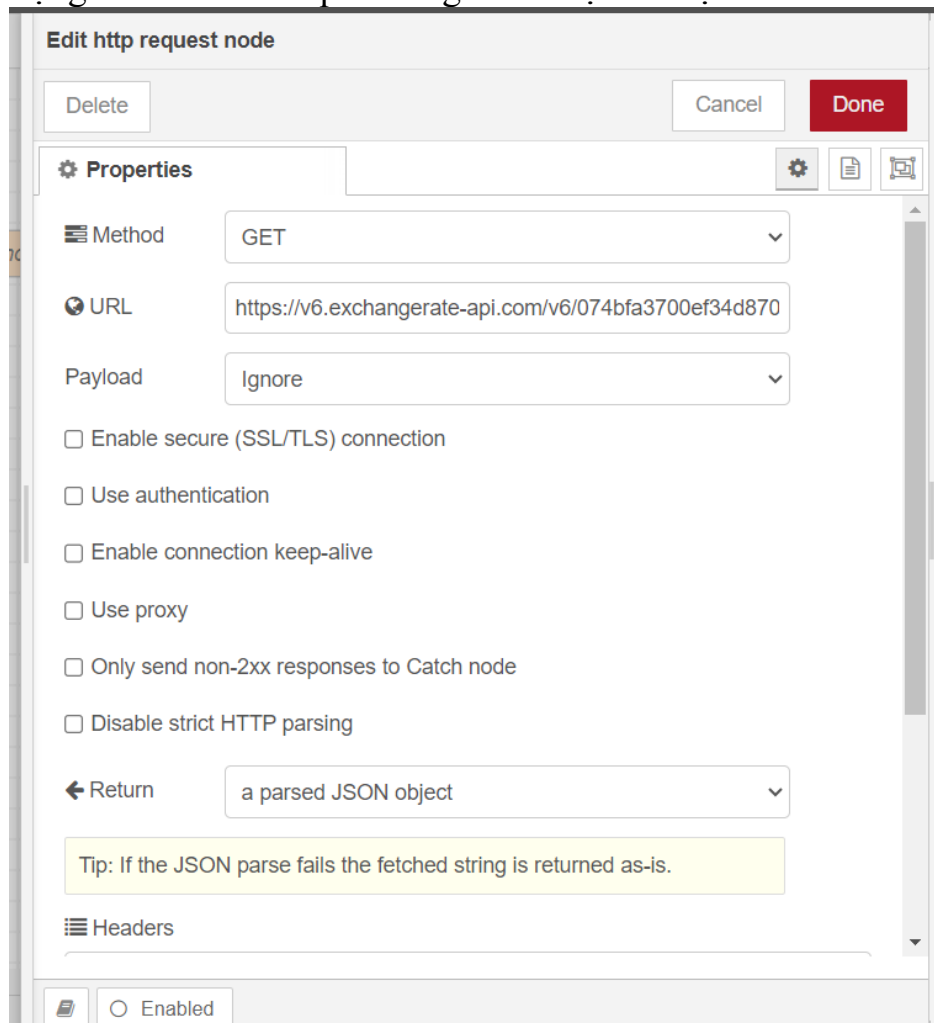
26 May 14:51:09 - [info] Node-RED version: v3.1.9
26 May 14:51:09 - [info] Node.js version: v20.13.1
26 May 14:51:09 - [info] Windows_NT 10.0.19042 x64 LE
26 May 14:51:10 - [info] Loading palette nodes
26 May 14:51:11 - [info] Settings file : C:\Users\A D M I N\.node-red\settings.js
26 May 14:51:11 - [info] Context store : 'default' [module-memory]
26 May 14:51:11 - [info] User directory : \Users\A D M I N\.node-red
26 May 14:51:11 - [warn] Projects disabled : editorTheme.projects.enabled=false
26 May 14:51:11 - [info] Flows file : \Users\A D M I N\.node-red\flows.json
26 May 14:51:11 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.
```

Hình 9. Chạy Node-Red trên Command Prompt

- Mở Node-Red tại <http://localhost:1880/#> và bắt đầu tạo flow trong Node-Red

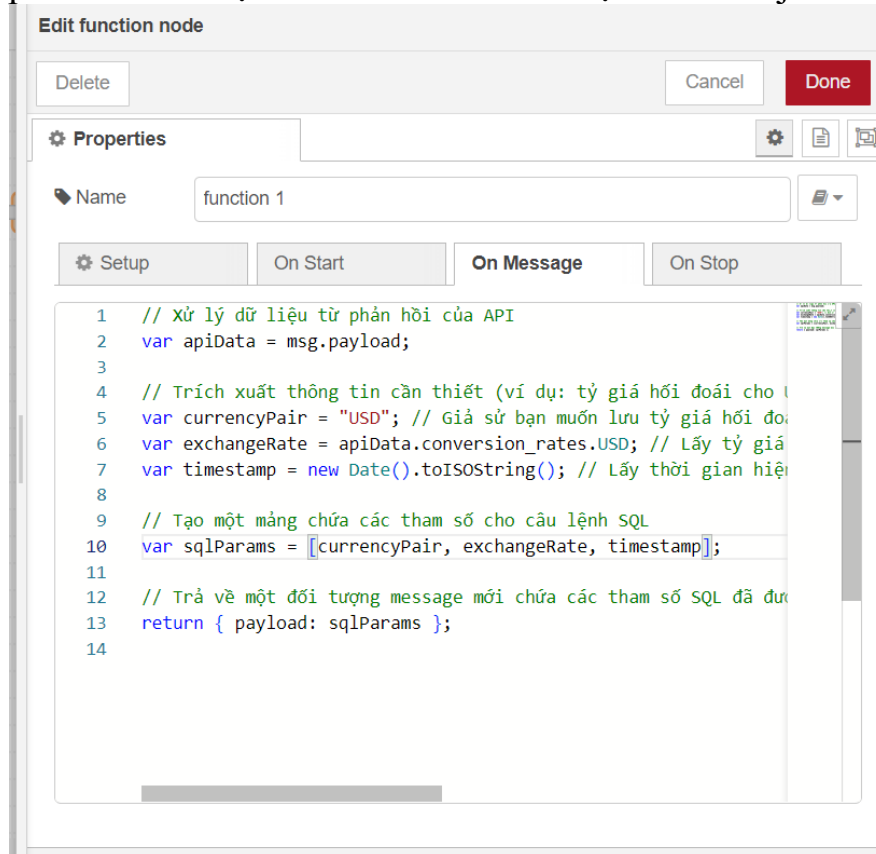
Sử dụng node HTTP request để gửi và nhận dữ liệu từ FastAPI



Hình 10. Thiết lập HTTP request node

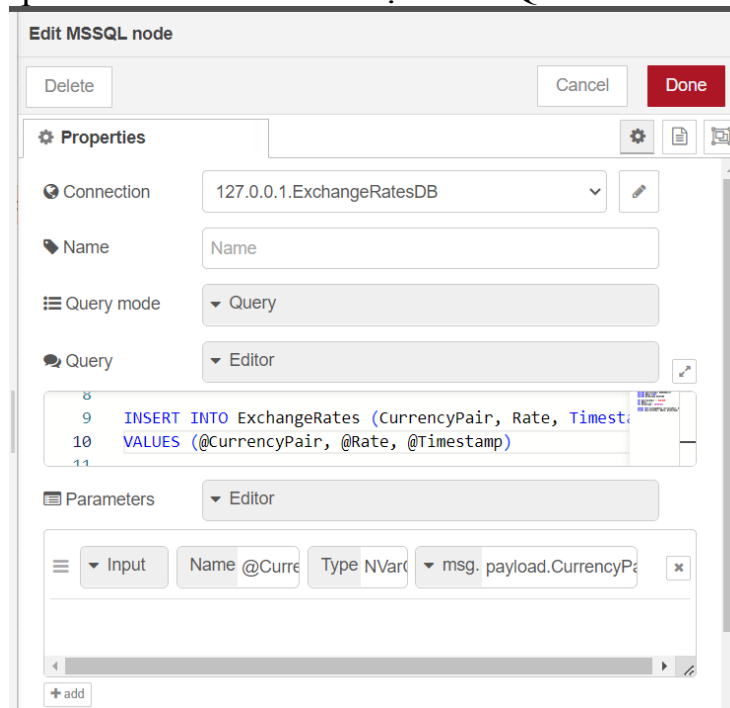


Tiếp đó em viết một function để xử lý dữ liệu và trả về json



Hình 11. Thiết lập function node

Thực thi procedure để thêm dữ liệu vào SQL



Hình 12. MSSQL node

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

Edit MSSQL node > Edit MSSQL-CN node

Delete Cancel Update

Properties

Name Connection Name

Server 127.0.0.1

Port 1443

Username sa

Password .....

Domain

Database ExchangeRatesDB

TDS Version 7\_4 (SQL Server 2012 ~ 2022)

Use Encryption? ☒

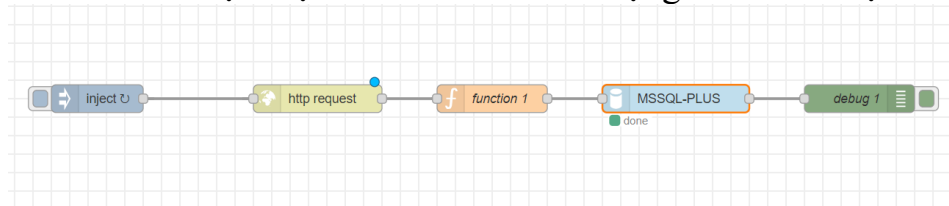
SQL Databases hosted on Azure will need this checked.

Trust Certificate? ☒

done

Hình 13. Edit MSSQL node

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



Hình 14. Liên kết các node

### 3.4. Viết store Procedure

- Dữ liệu được lưu vào dbo. ExchangeRates

NGUYET_DUONG\SQ...o.ExchangeRates				
	ID	CurrencyPair	Rate	Timestamp
▶	1	3	3.000000	2024-05-25 ...
	2	3	4.000000	2024-05-25 ...
	3	3	560000.000...	2024-05-25 ...
	4	3	4.000000	2024-05-25 ...
	5	3	45.000000	2024-05-25 ...
	6	3	7.000000	2024-05-25 ...
	7	3	4.000000	2024-05-25 ...
	8	3	4.000000	2024-05-25 ...
	9	3	4.000000	2024-05-25 ...
	10	3	4.000000	2024-05-25 ...
	11	3	4.000000	2024-05-25 ...
	12	3	4.000000	2024-05-25 ...
	13	3	4.000000	2024-05-25 ...
	14	3	4.000000	2024-05-25 ...
	15	3	4.000000	2024-05-25 ...
	16	3	4.000000	2024-05-25 ...
	17	3	4.000000	2024-05-25 ...
	18	3	4.000000	2024-05-25 ...
	19	3	4.000000	2024-05-25 ...
	20	3	4.000000	2024-05-25 ...

Hình 15. Dữ liệu

- Viết store Procedure

```
USE [ExchangeRatesDB]
GO
/***** Object: StoredProcedure [dbo].[SP_Chart]    Script Date: 5/27/2024 12:31:35 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
ALTER PROCEDURE [dbo].[SP_Chart]
AS
BEGIN
    DECLARE @json nvarchar(max) = N'{"ok":1,"msg":"ok","data":[';

    SELECT @json += FORMATMESSAGE(N'{"id": "%d", "sid": "%d", "value": "%s", "time": "%s"}',
                                   [ID], [CurrencyPair], CONVERT(nvarchar(10), [Rate]), CONVERT(nvarchar(5), [Timestamp], 108 ))
    FROM ExchangeRates order by [Timestamp]

    IF RIGHT(@json, 1) = ','
    BEGIN
        SET @json = LEFT(@json, LEN(@json) - 1);
    END

    SET @json = @json + ']}';

    SELECT @json AS json;
END
```

Hình 16. store Procedure

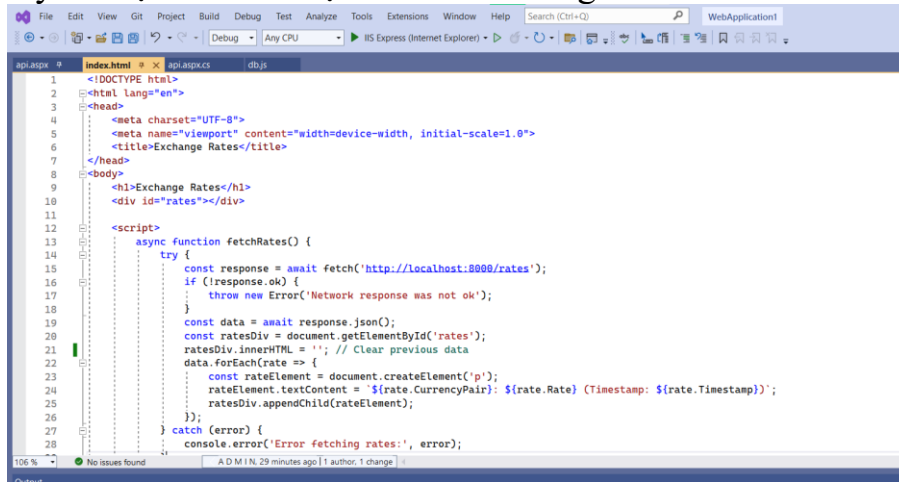
### 3.5. Xây dựng giao diện người dùng

- Em sử dụng html,css,js để lấy dữ liệu từ sqlserver thông qua asp.net (api.aspx)
- Kết nối với SQL

```
string connectionString = "Data Source=127.0.0.1,1443;Initial Catalog=chuchua;User Id=sa;Password=123;";  
using (SqlConnection connection = new SqlConnection(connectionString))  
{
```

Hình 14. Api.aspx.cs

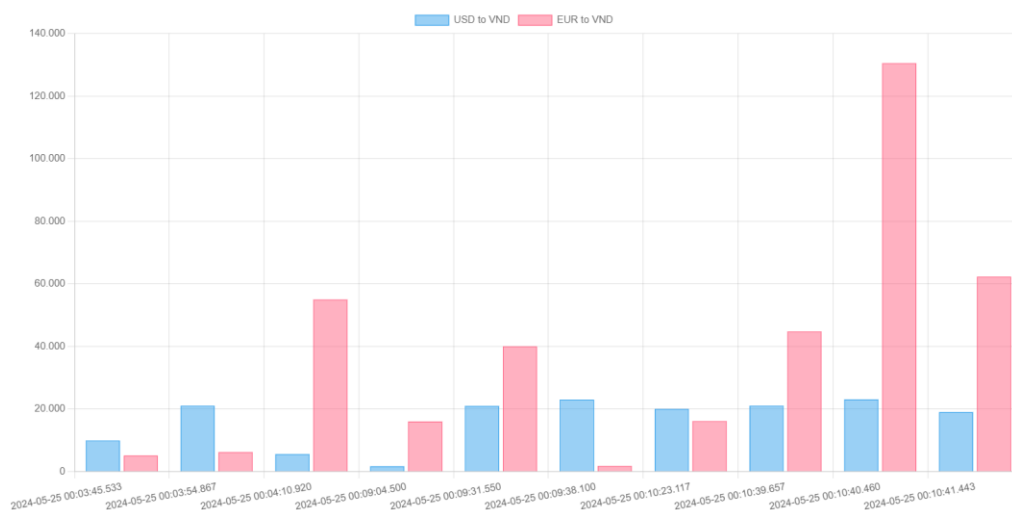
- Xử lý dữ liệu và hiển thị biểu đồ lên trang web



Hình 17. Index.html

### 3.6. Kết quả trên web

Theo dõi tỷ giá hối đoái



Hình 18. Biểu đồ

## CHƯƠNG 5. KẾT LUẬN

Trong bài tập lớn này, em đã xây dựng một hệ thống thu thập, xử lý và hiển thị dữ liệu về tỷ giá hối đoái toàn cầu từ một nguồn dữ liệu công cộng. Sự kết hợp giữa FastAPI, Node-RED và cơ sở dữ liệu SQL đã tạo ra một giải pháp mạnh mẽ và linh hoạt, qua đó giúp em hiểu rõ hơn về quá trình xây dựng các ứng dụng.

Giao diện người dùng của hệ thống cho phép hiển thị dữ liệu một cách trực quan và thân thiện, giúp người dùng dễ dàng nắm bắt các thông tin quan trọng về tỷ giá hối đoái. Hệ thống này cũng có tiềm năng 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.

Thông qua bài tập lớn này, em đã có cơ hội hiểu sâu hơn về ngôn ngữ lập trình Python cũng như một số công cụ khác như Node-RED, SQL,... Tuy nhiên, em cũng nhận thấy rằng kiến thức và kinh nghiệm của mình vẫn còn hạn chế, khiến cho đề tài chưa được hoàn thiện tốt và còn nhiều sai sót. Em sẽ nỗ lực học hỏi và tìm hiểu thêm để hoàn thiện đề tài này trong tương lai.

## **LỜI CẢM ƠN**

Do trình độ của em còn kém và chưa có nhiều kinh nghiệm trong quá trình làm bài và còn nhiều thiếu sót, em xin gửi lời cảm ơn thầy Đỗ Duy Cốp đã giúp đỡ và tạo điều kiện để em hoàn thành đề tài này.

Em xin chân thành cảm ơn!

## TÀI LIỆU THAM KHẢO

<https://chat.openai.com/>

<https://www.exchangerate-api.com/>

<https://developers.google.com/chart?hl=vi>