

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP
KHOA ĐIỆN TỬ
BỘ MÔN: CÔNG NGHỆ THÔNG TIN



BÀI TẬP LỚN
LẬP TRÌNH PYTHON

Giáo viên hướng dẫn: Đỗ Duy Cốp
Sinh viên thực hiện: Phương Minh Duy
Ngành học: Kỹ thuật máy tính
Lớp: K56KMT.01

Thái Nguyên 2024

NHIỆM VỤ BÀI TẬP LỚN MÔN

Sinh viên: Phương Minh Duy

MSSV: K205480106011

Lớp: K56KMT.01

Khoá: K56

Ngành học: Kỹ thuật máy tính

Giáo viên hướng dẫn: Đỗ Duy Cốp

1. Tên đề tài:

WEBSITE THEO DÕI GIÁ ĐỒNG GBP (BẢNG ANH)

2. Nội dung thực hiện

- Cơ sở dữ liệu: Lưu thông tin gồm thời gian và số liệu
- Sử dụng FastApi và python để tạo ra 1 API từ các trang web về giá đồng GBP
- Xây dựng một chu trình trong Node-RED để tự động gọi API Python để lấy dữ liệu. Sau đó, xử lý dữ liệu và ghi dữ liệu vào database
- Xây dựng trang Web để hiển thị thông tin : Xây dựng một ứng dụng web để hiển thị dữ liệu từ cơ sở dữ liệu. Hiển thị giá đồng yên theo ngày hoặc giờ

3. Các sản phẩm, kết quả :

- Sản phẩm phần mềm theo yêu cầu của bài tập lớn.
- Thuyết minh đồ án theo mẫu chung của khoa Điện tử.

4. Ngày giao nhiệm vụ: 15/05/2024

5. Ngày hoàn thành nhiệm vụ: 26/05/2024

GIÁO VIÊN HƯỚNG DẪN

Đỗ Duy Cốp

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Thái Nguyên , ngày...tháng...năm...

Giáo viên hướng dẫn

(Ký và ghi rõ họ tên)

MỤC LỤC

CHƯƠNG I. GIỚI THIỆU VỀ PYTHON	6
1.1. Thực trạng hệ thống.....	6
1.2. Đặc Điểm Chính của Python	6
1.3. Ứng dụng của Python.....	6
1.4. Các công cụ và môi trường phát triển Python.....	7
CHƯƠNG II. PHÂN TÍCH VÀ THIẾT KẾ	8
2.1. Mục Tiêu của Dự Án.....	8
2.2. Các Thành Phần Của Dự Án.....	8
2.3 Các bước thực hiện.....	9
CHƯƠNG III. KẾT LUẬN	16

LỜI NÓI ĐẦU

Dưới bối cảnh toàn cầu hóa và sự phát triển nhanh chóng của thị trường tài chính, tỷ giá hối đoái đóng một vai trò quan trọng trong các hoạt động kinh tế và thương mại quốc tế. Hiểu và theo dõi sự biến động của tỷ giá hối đoái là cần thiết không chỉ đối với các tổ chức tài chính mà còn đối với các doanh nghiệp và cá nhân tham gia vào thị trường quốc tế.

Bài toán này tập trung vào việc xây dựng một hệ thống tự động cập nhật và hiển thị tỷ giá hối đoái giữa đồng GBP (bảng Anh) và VND theo thời gian thực. Hệ thống này không chỉ cung cấp thông tin kịp thời và chính xác về tỷ giá hối đoái mà còn giúp người dùng dễ dàng theo dõi sự biến động của tỷ giá qua các biểu đồ trực quan.

Em xin cảm ơn Thầy Đỗ Duy Cốp đã hướng dẫn em hoàn thành đề tài này.

CHƯƠNG I. GIỚI THIỆU VỀ PYTHON

1.1. Thực trạng hệ thống

Python là một ngôn ngữ lập trình được tạo ra bởi Guido van Rossum và phát hành lần đầu vào năm 1991. Ban đầu, Python được phát triển như một dự án sở thích nhằm tạo ra một ngôn ngữ lập trình dễ đọc và dễ viết hơn so với các ngôn ngữ hiện có. Tên "Python" được lấy cảm hứng từ nhóm hài kịch Monty Python, thể hiện sự hài hước và dễ tiếp cận của ngôn ngữ này.

Trong suốt thập kỷ đầu tiên, Python đã thu hút được sự quan tâm từ cộng đồng lập trình viên nhờ vào triết lý thiết kế nhấn mạnh vào sự rõ ràng và đơn giản. Phiên bản 2.0 ra mắt năm 2000, mang đến nhiều cải tiến quan trọng như khả năng xử lý Unicode và bộ sưu tập các công cụ chuẩn (standard library) phong phú hơn. Phiên bản 3.0, ra mắt vào năm 2008, là một bước tiến lớn với nhiều thay đổi không tương thích ngược, nhằm loại bỏ các khuyết điểm của ngôn ngữ và cải thiện hiệu năng.

1.2. Đặc Điểm Chính của Python

Dễ Học và Sử Dụng: Python được thiết kế với cú pháp rõ ràng và dễ đọc, làm cho việc học lập trình trở nên dễ dàng hơn cho người mới bắt đầu. Các khái niệm cơ bản như biến, hàm, và vòng lặp được thể hiện một cách trực quan.

Đa Nền Tảng: Python có thể chạy trên nhiều hệ điều hành khác nhau như Windows, macOS, Linux, và các hệ điều hành khác, giúp lập trình viên dễ dàng phát triển và triển khai ứng dụng trên nhiều môi trường.

Thư Viện Phong Phú: Python có một hệ thống thư viện phong phú và đa dạng, hỗ trợ nhiều lĩnh vực như web development (Django, Flask), data analysis (Pandas, NumPy), machine learning (Scikit-learn, TensorFlow), và nhiều lĩnh vực khác.

Mã Nguồn Mở: Python là ngôn ngữ mã nguồn mở, cho phép người dùng tự do sử dụng, thay đổi và phân phối. Điều này đã thúc đẩy một cộng đồng lớn mạnh và sự phát triển liên tục của ngôn ngữ này.

1.3. Ứng dụng của Python

Web Development: Python được sử dụng rộng rãi trong phát triển web với các framework nổi tiếng như Django và Flask. Django cung cấp một bộ công cụ mạnh mẽ và hoàn chỉnh để xây dựng các ứng dụng web phức tạp, trong khi Flask là một framework nhẹ, linh hoạt cho các dự án nhỏ hơn.

Data Science và Machine Learning: Python là ngôn ngữ chính trong lĩnh vực khoa học dữ liệu và học máy. Các thư viện như Pandas, NumPy, và Matplotlib cung

cấp các công cụ mạnh mẽ để phân tích và trực quan hóa dữ liệu, trong khi Scikit-learn và TensorFlow hỗ trợ các thuật toán học máy phức tạp.

Tự Động Hóa: Python được sử dụng để viết các script tự động hóa các tác vụ lặp đi lặp lại, từ việc quản lý hệ thống đến xử lý dữ liệu.

Phát Triển Ứng Dụng: Python cũng được sử dụng trong phát triển ứng dụng desktop (với PyQt, Tkinter) và trò chơi (với Pygame).

1.4. Các công cụ và môi trường phát triển Python

IDLE: Một môi trường phát triển tích hợp đơn giản đi kèm với Python, phù hợp cho người mới bắt đầu.

PyCharm: Một IDE mạnh mẽ dành cho Python, hỗ trợ nhiều tính năng như gỡ lỗi, quản lý phiên bản, và tự động hoàn thành mã.

Jupyter Notebook: Một công cụ mạnh mẽ cho phép viết và chạy mã Python trong một môi trường tương tác, rất phổ biến trong lĩnh vực khoa học dữ liệu. Nó cho phép người dùng tạo và chia sẻ tài liệu chứa mã nguồn, hình ảnh, và các chú thích.

CHƯƠNG II. PHÂN TÍCH VÀ THIẾT KẾ

2.1. Mục Tiêu của Dự Án

Mục tiêu

1. **Thu thập dữ liệu tỷ giá:** Sử dụng API từ dịch vụ FastForex.
2. **Lưu trữ dữ liệu:** Lưu trữ thông tin vào cơ sở dữ liệu SQL Server.
3. **Hiển thị dữ liệu:** Phát triển giao diện hiển thị dữ liệu tỷ giá bằng biểu đồ.

Công nghệ sử dụng

- **FastAPI:** Xây dựng API.
- **SQL Server:** Lưu trữ dữ liệu.
- **Chart.js:** Tạo biểu đồ.
- **HTML/CSS và JavaScript:** Xây dựng giao diện người dùng.

Nội dung thực hiện

1. **Thiết lập cơ sở dữ liệu:** Tạo bảng JPY_VND.
2. **Phát triển API:** Sử dụng FastAPI để lấy và cập nhật dữ liệu tỷ giá.
3. **Tự động hóa cập nhật dữ liệu:** Sử dụng Node-RED để tự động cập nhật mỗi phút.
4. **Hiển thị dữ liệu:** Xây dựng giao diện người dùng với biểu đồ tỷ giá USD/VND.

2.2. Các Thành Phần Của Dự Án

Cơ Sở Dữ Liệu:

- Bảng: `forex_rates` : Lưu thông tin về giá USD, các thông tin base (USD), to_currency (VND), rate (25441.20195), và updated_at (thời gian cập nhật) vào bảng JPY_VND.

Module đọc dữ liệu: Sử dụng Python và FastAPI để tạo một API để lấy dữ liệu từ trang web chứa tỷ giá USD như FastForex.....

Mô tả nguồn dữ liệu:

- Sử dụng Web Scraping hoặc lấy dữ liệu qua API của các trang web chuyên về Tỷ giá đồng GBP (bảng Anh).
- Dữ liệu bao gồm thông tin về Tỷ giá GBP và thời gian cập nhật. Node-RED:

- Xây dựng một chu trình trong Node-RED để tự động gọi Stored Procedure SP_SaveGoldPrice để lưu giá vàng vào cơ sở dữ liệu.
- Node-RED sẽ gọi API Python (hoặc một dịch vụ khác) để lấy dữ liệu và thời gian cập nhật. Sau đó, dữ liệu này sẽ được truyền sang vào bảng JPY_VND để lưu trữ. Web:
- Xây dựng một ứng dụng web để hiển thị dữ liệu từ cơ sở dữ liệu.
- Sử dụng các công nghệ như HTML, CSS, JavaScript để tạo giao diện web.

2.3 Các bước thực hiện.

Bước 1: Cài đặt môi trường

- Cài đặt Python và các thư viện cần thiết: FastAPI, requests, pyodbc, uvicorn.
- Cài đặt SQL Server và tạo cơ sở dữ liệu.

Bước 2: Tạo cơ sở dữ liệu và bảng

- Tạo bảng JPY_VND trong SQL Server để lưu trữ tỷ giá:

```
sql
Sao chép mã
SELECT TOP (1000) [ID]
      , [base]
      , [to_currency]
      , [rate]
      , [updated_at]
FROM [Gia_JPY].[dbo].[JPY_VND]);

import asyncio
from fastapi import FastAPI, HTTPException
import requests
from datetime import datetime, timedelta, timezone

app = FastAPI()

API_KEY = '1af198eb76-8362cbab0b-sdx021' # Thay bằng key API của bạn
BASE_URL = 'https://api.fastforex.io/' # URL cơ bản của API fastFOREX

# Định nghĩa múi giờ UTC và múi giờ Việt Nam
UTC_timezone = timezone.utc
VN_timezone = timezone(timedelta(hours=7)) # UTC+7

async def fetch_forex_data():
    while True:
        from_currency = 'GBP' # Tiền tệ mặc định là JPY

        url = f"{BASE_URL}fetch-one?from={from_currency}&to=GBP&api_key={API_KEY}" # Chỉnh sửa đây
        response = requests.get(url)

        if response.status_code == 200:
            forex_data = response.json()
```

```

        # Chuyển đổi thời gian từ múi giờ UTC sang múi giờ Việt
        Nam
        updated_time_utc =
datetime.strptime(forex_data['updated'], "%Y-%m-%d %H:%M:%S")
        updated_time_vn =
updated_time_utc.replace(tzinfo=UTC_timezone).astimezone(VN_timezone)
        forex_data['updated'] = updated_time_vn.strftime("%Y-%m-%d
%H:%M:%S %Z%z")
        print(forex_data) # In ra dữ liệu JSON cập nhật
        elif response.status_code == 401:
            raise HTTPException(status_code=401, detail="Unauthorized.
Check your API key.")
        elif response.status_code == 404:
            raise HTTPException(status_code=404, detail="Currency pair
not found.")
        else:
            raise HTTPException(status_code=response.status_code,
detail=response.text)

        await asyncio.sleep(30) # Chờ 30 giây trước khi gửi yêu cầu
kế tiếp

@app.on_event("startup")
async def startup_event():
    asyncio.create_task(fetch_forex_data()) # Bắt đầu lặp vô hạn để
cập nhật dữ liệu

@app.get("/forex/{to_currency}")
def get_forex_rate(to_currency: str):
    from_currency = 'GBP' # Tiền tệ mặc định là JPY

    # Kiểm tra định dạng mã tiền tệ
    if len(to_currency) != 3 or not to_currency.isalpha():
        raise HTTPException(status_code=400, detail="Invalid currency
code format. Use a 3-letter currency code.")

    url = f"{BASE_URL}fetch-
one?from={from_currency}&to={to_currency}&api_key={API_KEY}"
    response = requests.get(url)

    if response.status_code == 200:
        forex_data = response.json()
        # Chuyển đổi thời gian từ múi giờ UTC sang múi giờ Việt Nam
        updated_time_utc = datetime.strptime(forex_data['updated'],
"%Y-%m-%d %H:%M:%S")
        updated_time_vn =
updated_time_utc.replace(tzinfo=UTC_timezone).astimezone(VN_timezone)
        forex_data['updated'] = updated_time_vn.strftime("%Y-%m-%d
%H:%M:%S %Z%z")
        return forex_data # Trả về dữ liệu JSON trực tiếp
    elif response.status_code == 401:
        raise HTTPException(status_code=401, detail="Unauthorized.
Check your API key.")
    elif response.status_code == 404:
        raise HTTPException(status_code=404, detail="Currency pair not
found.")
    else:

```

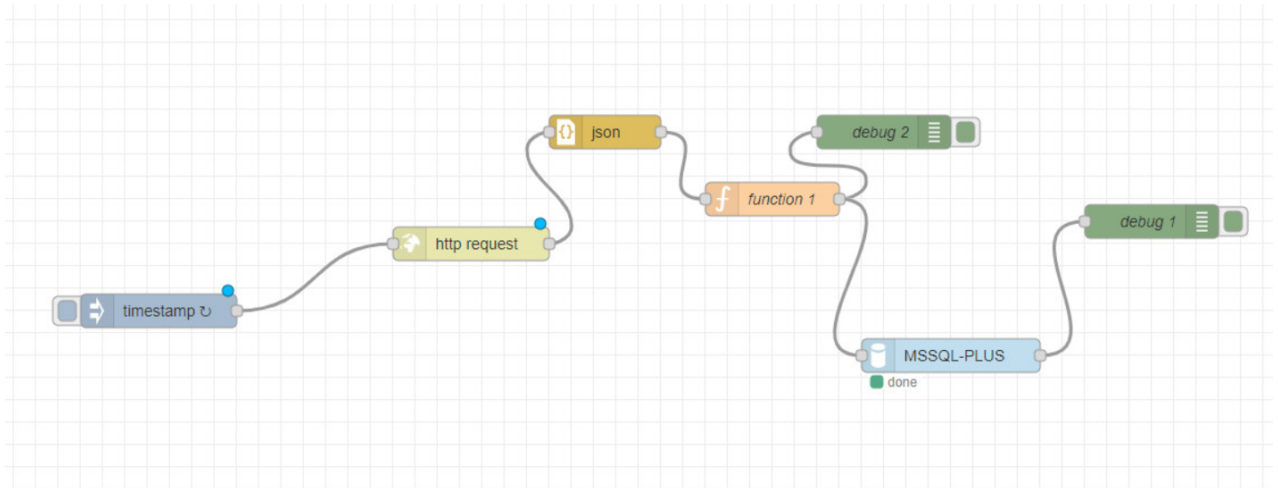
```
        raise HTTPException(status_code=response.status_code,
                             detail=response.text)

    if __name__ == "__main__":
        import uvicorn

        uvicorn.run(app, host="127.0.0.1", port=8001)
```

Bước 3: Xây dựng server FastAPI

Bước 4: Cấu hình node-red để đọc dữ liệu từ request và gửi vào database



- Thực hiện cấu hình nốt request để lấy dữ liệu:

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

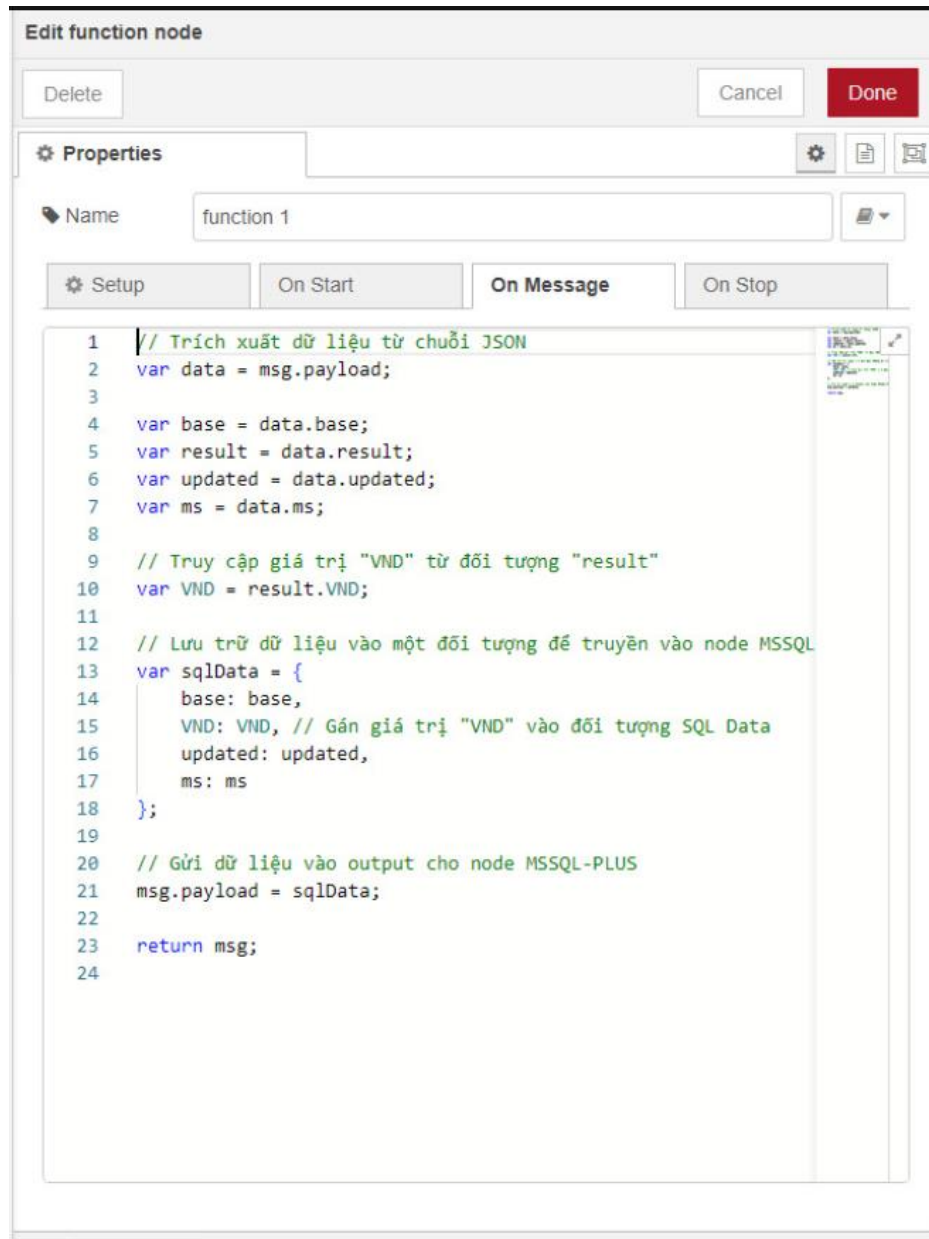
⬅ Return

a UTF-8 string

▼

☰ Headers

- Viết hàm xử lý dữ liệu ở nút function để gửi vào SQL:



- Thực hiện các cấu hình ở nút SQL để nhận dữ liệu từ request:

Edit MSSQL node

Delete Cancel Done

Properties

Connection: DESKTOP-7UI35O8.Gia_JPY

Name: Name

Query mode: Execute Procedure

Query: Editor

1 GBP_VND

Parameters

Editor

Input	Name	Type	Value
▼ Input	base	VarCh	msg.payload.base
▼ Input	rate	Decin	msg.payload.ms
▼ Input	to_curre	Float	msg.payload.VND

+ add

Parse Mustache ☒

Output property: msg.payload

Output type: Original output

Error Handling: Throw error

- Dữ liệu được gọi thành công:

```

tạo bản in đẹp
[{"ID": "42", "base": "GBP", "to_currency": "32381.900000000001", "rate": "2.00000", "updated_at": "2024-05-24 20:33:39.567"},
{"ID": "60", "base": "GBP", "to_currency": "32381.900000000001", "rate": "5.00000", "updated_at": "2024-05-24 20:51:39.170"},
{"ID": "67", "base": "GBP", "to_currency": "32433.200000000001", "rate": "2.00000", "updated_at": "2024-05-26 08:54:01.283"},
{"ID": "44", "base": "GBP", "to_currency": "32381.900000000001", "rate": "2.00000", "updated_at": "2024-05-24 20:55:39.027"},
{"ID": "14", "base": "GBP", "to_currency": "32378.299999999999", "rate": "2.00000", "updated_at": "2024-05-23 14:26:58.540"},
{"ID": "54", "base": "GBP", "to_currency": "32381.900000000001", "rate": "5.00000", "updated_at": "2024-05-24 20:45:39.247"},
{"ID": "23", "base": "GBP", "to_currency": "32381.900000000001", "rate": "2.00000", "updated_at": "2024-05-24 20:23:38.787"},
{"ID": "32", "base": "GBP", "to_currency": "32381.900000000001", "rate": "2.00000", "updated_at": "2024-05-24 20:23:45.080"},
{"ID": "75", "base": "GBP", "to_currency": "32433.200000000001", "rate": "2.00000", "updated_at": "2024-05-26 08:59:32.480"},
{"ID": "27", "base": "GBP", "to_currency": "32381.900000000001", "rate": "3.00000", "updated_at": "2024-05-24 20:23:43.950"},
{"ID": "73", "base": "GBP", "to_currency": "32433.200000000001", "rate": "2.00000", "updated_at": "2024-05-26 08:57:32.463"},
{"ID": "47", "base": "GBP", "to_currency": "32381.900000000001", "rate": "7.00000", "updated_at": "2024-05-24 20:38:39.087"},
{"ID": "46", "base": "GBP", "to_currency": "32381.900000000001", "rate": "2.00000", "updated_at": "2024-05-24 20:37:39.063"},
{"ID": "77", "base": "GBP", "to_currency": "32433.200000000001", "rate": "3.00000", "updated_at": "2024-05-26 09:01:32.500"},
{"ID": "41", "base": "GBP", "to_currency": "32381.900000000001", "rate": "1.00000", "updated_at": "2024-05-24 20:32:39.040"}]

```

Bước 5: Xây dựng giao diện bằng php và gọi dữ liệu từ Database để hiển thị ra giao diện.

Trong thư mục php gồm có:

- File api.php để thực hiện kết nối với SQL
- File index.js, index.html và index.css để thực hiện tạo giao diện hiển thị

Kết quả cuối cùng của bài toán:

Bài tập				
STT	Base	To Currency	Rate	Updated At
1	GBP	32433.200000000001	5.00000	2024-05-26 11:27:06.677
2	GBP	32433.200000000001	5.00000	2024-05-26 12:20:38.027
3	GBP	32433.200000000001	4.00000	2024-05-26 15:47:10.890
4	GBP	32433.200000000001	5.00000	2024-05-26 15:25:10.683
5	GBP	32388.900000000001	5.00000	2024-05-24 20:54:39.313
6	GBP	32433.200000000001	5.00000	2024-05-26 09:29:14.510
7	GBP	32433.200000000001	4.00000	2024-05-26 15:59:10.980
8	GBP	32433.200000000001	5.00000	2024-05-26 11:36:06.990
9	GBP	32381.900000000001	1.00000	2024-05-24 20:26:38.970
10	GBP	32381.900000000001	2.00000	2024-05-24 20:23:38.970

CHƯƠNG III. KẾT LUẬN

Trong dự án này, chúng tôi đã xây dựng một trang web theo dõi tỷ giá hối đoái giữa đồng GBP (bảng Anh) và VND bằng cách sử dụng FastAPI cho backend và Chart.js cho frontend. Chúng tôi đã thành công trong việc lấy dữ liệu từ API của dịch vụ fastFOREX, xử lý và lưu trữ dữ liệu vào cơ sở dữ liệu SQL Server, và hiển thị dữ liệu trực quan trên giao diện người dùng.

Trang web không chỉ cung cấp khả năng theo dõi tỷ giá một cách liên tục và chính xác mà còn minh họa cách tích hợp và triển khai các công nghệ web hiện đại. Kết quả đạt được cho thấy tỷ giá hối đoái được cập nhật và hiển thị một cách hiệu quả, đáp ứng nhu cầu theo dõi và phân tích của người dùng.

Dự án này là một ví dụ điển hình cho việc sử dụng các công nghệ tiên tiến để giải quyết các vấn đề thực tế trong việc theo dõi và phân tích dữ liệu tài chính. Trong tương lai, chúng tôi có thể mở rộng thêm các tính năng như cảnh báo thay đổi tỷ giá, hỗ trợ nhiều cặp tiền tệ khác, và tối ưu hóa hiệu suất hệ thống để phục vụ người dùng tốt hơn.

Cuối cùng, chúng tôi xin chân thành cảm ơn sự hỗ trợ và đồng hành của các đồng nghiệp, bạn bè, và những người đã giúp đỡ chúng tôi trong suốt quá trình thực hiện dự án. Chúng tôi hy vọng rằng, với sự nỗ lực này, trang web theo dõi tỷ giá hối đoái sẽ trở thành một công cụ hữu ích, đóng góp vào sự phát triển chung của xã hội trong lĩnh vực giao tiếp đa ngôn ngữ.