

**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: Hoảng Trần Phẫu**  
**Ngành học: Kỹ thuật máy tính**  
**Lớp: K56KMT.01**

**Thái Nguyên 2024**



## NHIỆM VỤ BÀI TẬP LỚN MÔN

*Sinh viên:* Hoàng Trần Phâu

*MSSV:* K205480106022

*Lớp:* K56KMT.01

*Khoá:* K56

*Ngành học:* Kỹ thuật máy tính

*Giáo viên hướng dẫn:* Đỗ Duy Cốp

1. Tên đề tài: Hệ thống hiển thị tỷ và theo dõi giá đô la mỹ (USD)

2. Nội dung thực hiện:

Cơ sở dữ liệu:

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 ĐÔ la Mỹ.
- Dữ liệu bao gồm thông tin về Tỷ giá USD 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 forex\_rates 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.

**GIÁO VIÊN HƯỚNG DẪN**

**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 .....	7
1.1.  Thực trạng hệ thống .....	7
1.2.  Đặc Điểm Chính của Python .....	7
1.3.  Ứng dụng của Python .....	7
1.4.  Các công cụ và môi trường phát triển Python .....	8
CHƯƠNG II. PHÂN TÍCH VÀ THIẾT KẾ.....	9
2.1. Mục Tiêu của Dự Án .....	9
2.2. Các Thành Phần Của Dự Án.....	9
2.3 Các bước thực hiện. ....	10
CHƯƠNG III. KẾT LUẬN.....	16

## LỜI NÓI ĐẦU

Trong 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 USD 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 forex\_rates.
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 forex\_rates.

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 ĐÔ la Mỹ.

- Dữ liệu bao gồm thông tin về Tỷ giá USD 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 `forex_rates` 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 `forex_rates` trong SQL Server để lưu trữ tỷ giá:

```
sql
Sao chép mã
CREATE TABLE forex_rates (
    id INT PRIMARY KEY IDENTITY(1,1),
    base VARCHAR(10) NOT NULL,
    rate DECIMAL(18, 5) NOT NULL,
    updated_at DATETIME DEFAULT GETDATE()
);
```

### *Bước 3: Xây dựng server FastAPI*

```
from fastapi import FastAPI, HTTPException
import requests

app = FastAPI()

API_KEY = 'dd086108ea-41eb72e8f1-se2v28' # Thay bằng key API của bạn
BASE_URL = 'https://api.fastforex.io/' # URL cơ bản của API fastFOREX

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

    # 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)
```

```

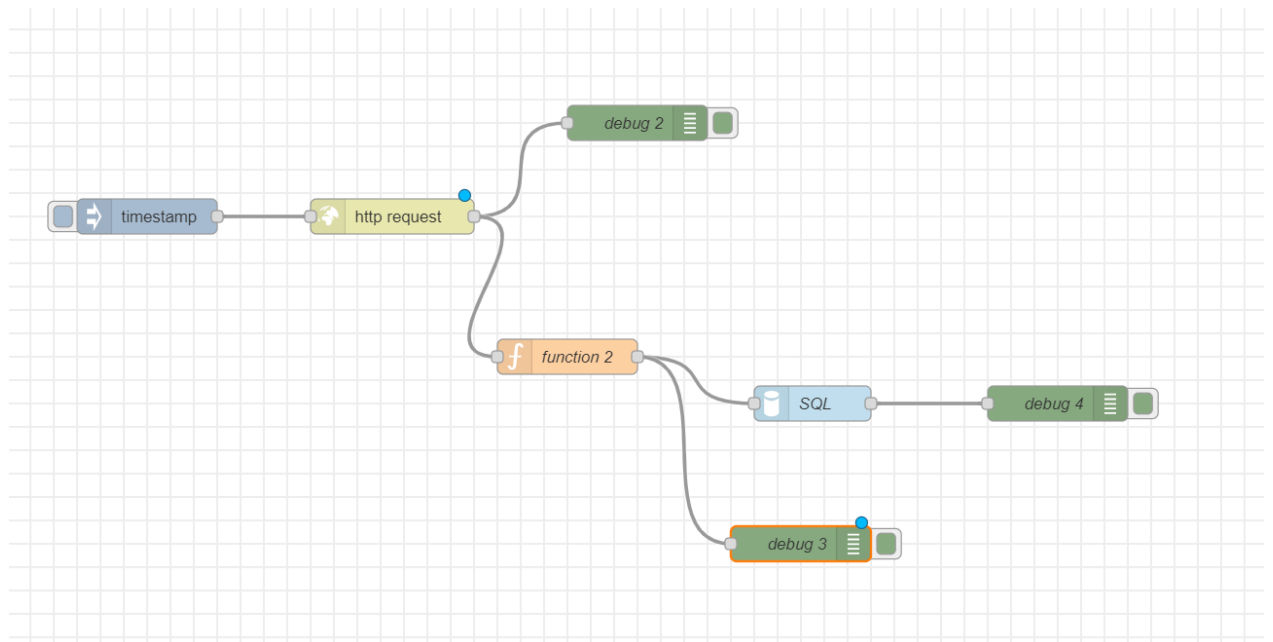
if response.status_code == 200:
    data = response.json()

    return data
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=8000)

```

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



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

**Edit http request node**

Delete

Cancel

Done

**Properties**

Method

GET

▼

URL

http://127.0.0.1:8000/forex/USD

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

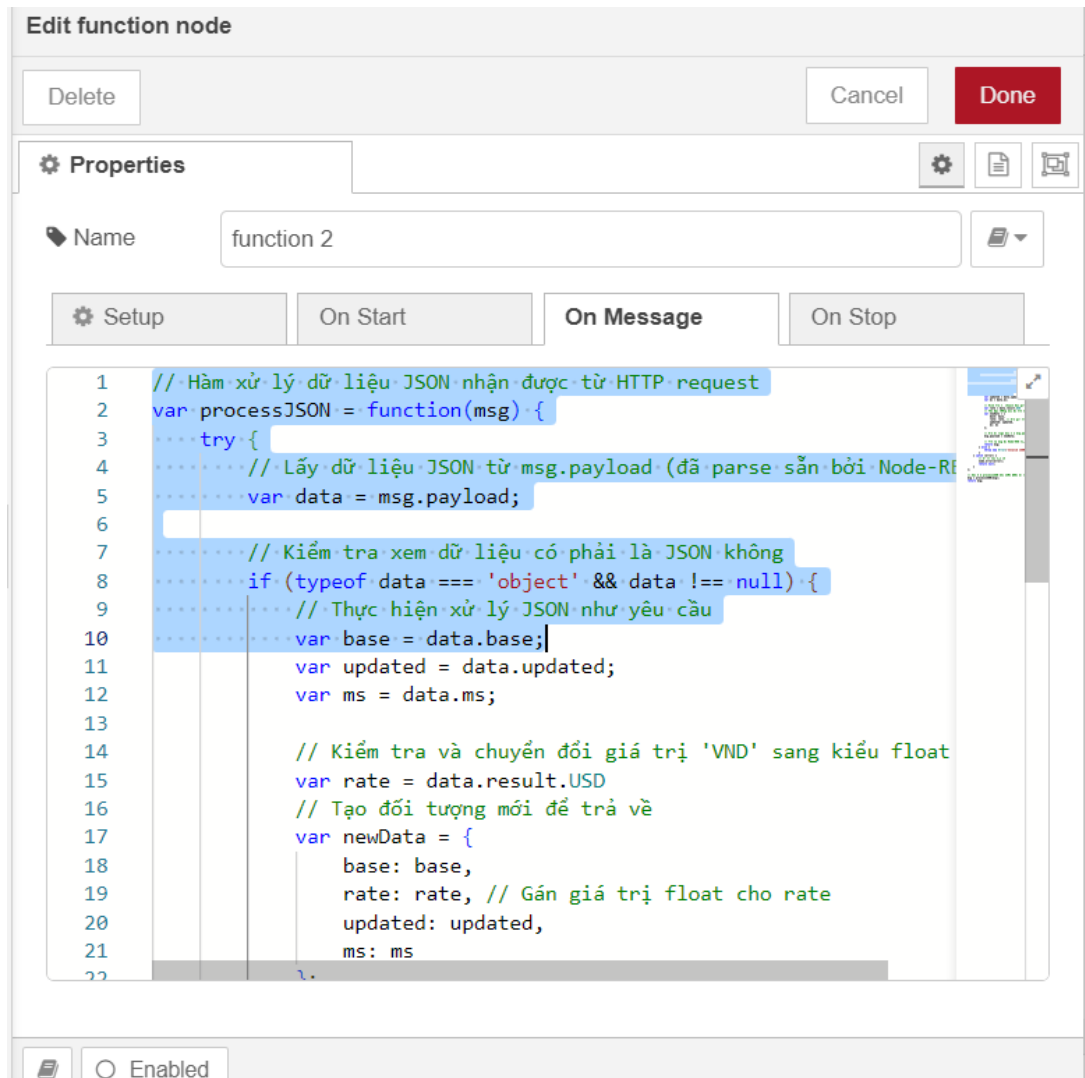
a parsed JSON object

▼

Tip: If the JSON parse fails the fetched string is returned as-is.

Headers

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





*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:



1	USD	25480.979930000001	2024-05-23
2	USD	25480.979930000001	2024-05-23
3	USD	25480.979930000001	2024-05-23
4	USD	25480.979930000001	2024-05-23
5	USD	25480.979930000001	2024-05-23
6	USD	25480.979930000001	2024-05-23
7	USD	25480.979930000001	2024-05-23
8	USD	25480.979930000001	2024-05-23
9	USD	25480.979930000001	2024-05-23
10	USD	25480.979930000001	2024-05-23
11	USD	25480.979930000001	2024-05-23

### CHƯƠNG III. KẾT LUẬN

Trong dự án này, chúng tôi đã xây dựng một ứng dụng theo dõi tỷ giá hối đoái USD/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.

Ứng dụng 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 thầy cô, bạn bè, và đồng nghiệp trong suốt quá trình thực hiện dự án. Chúng tôi hy vọng rằng, với sự nỗ lực này, ứng dụng dịch tự động 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.