

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP
KHOA ĐIỆN TỬ

Bộ môn: Công nghệ thông tin



BÀI TẬP LỚN LẬP TRÌNH PYTHON

Tên đề tài: GIÁM SÁT GIÁ BITCOIN

Sinh viên thực hiện: Phạm Tuấn Dương

MSSV: K205480106042

Lớp: 56KMT.01

Thái Nguyên – 2024

Ngày hoàn thành, ngày 26 tháng 05 năm 2024

BÀI TẬP LỚN LẬP TRÌNH PYTHON

BỘ MÔN CÔNG NGHỆ THÔNG TIN

Sinh viên: Phạm Tuấn Dương

MSSV: K205480106042

Ngành: Kỹ thuật máy tính

Lớp học phần: 56KMT.01

Tên đề tài: **GIÁM SÁT GIÁ BITCOIN**

I. Mục tiêu

Lấy dữ liệu bitcoin.

Xử lý dữ liệu: sử dụng FastAPI và Node-RED, sau đó lưu vào cơ sở dữ liệu.

Xây dựng trang web.

II. Cơ sở dữ liệu

Tạo 2 bảng là bảng coin và bảng price để lưu:

Coin gồm: cid, name.

Price gồm: id, cid, priceCoin, time

Store procedure gồm: lưu giá bitcoin mới vào cơ sở dữ liệu, lấy ra giá và thời gian của giá bitcoin.

III. Module đọc dữ liệu

Sử dụng Python và FastAPI để tạo một API để lấy dữ liệu từ dịch vụ API liên quan đến tiền điện tử.

Lấy dữ liệu qua API của các trang về tiền điện tử.

Dữ liệu bao gồm thông tin và giá đồng bitcoin.

Node-RED: 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 cơ sở dữ liệu (SQL Server).

IV. Web

Sử dụng ASP.NET để vào DB lấy dữ liệu rồi trả về.

Sử dụng HTML, CSS, JS để tạo giao diện web.

Hiển thị giá bitcoin hiện tại và vẽ biểu đồ giá bitcoin theo ngày.

GIÁO VIÊN HƯỚNG DẪN

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Thái Nguyên, ngày....tháng.....năm 20....
GIÁO VIÊN HƯỚNG DẪN
(Ký ghi rõ họ tên)

MỤC LỤC

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP	1
MỤC LỤC	1
LỜI NÓI ĐẦU	2
CHƯƠNG I: GIỚI THIỆU.....	3
1.1. Những phần chính	3
CHƯƠNG II: LẤY DỮ LIỆU BITCOIN	4
2.1. Trang web lấy dữ liệu	4
2.2. Sử dụng python và fash API lấy dữ liệu	6
2.2. Tạo cơ sở dữ liệu.....	7
2.3. Sử dụng node-red	9
CHƯƠNG III: THIẾT KẾ GIAO DIỆN.....	13
3.1. Tạo File aspx để lấy dữ liệu từ cơ sở dữ liệu.....	13
3.2. file html	14
3.3. file css.....	15
3.4. file Javascript	15
CHƯƠNG IV: KIỂM THỬ	20
4.1. Giao diện web	20
4.2. Nhập số lượng bitcoin sẽ quy ra giá tiền cụ thể.....	20
4.3. Xuất excel.....	21
KẾT LUẬN	22

LỜI NÓI ĐẦU

Trong thời đại kỹ 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 điện tử đang phát triển mạnh mẽ và được nhiều quốc gia công nhận. Các loại tiền điện tử như Bitcoin không chỉ có mệnh giá cao mà còn 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 “Giám sát giá Bitcoin” với mong muốn giúp mọi người có thể theo dõi giá của đồng Bitcoin một cách thuận tiện và nhanh chóng. Đề tài này không chỉ tập trung vào việc cập nhật giá Bitcoin 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

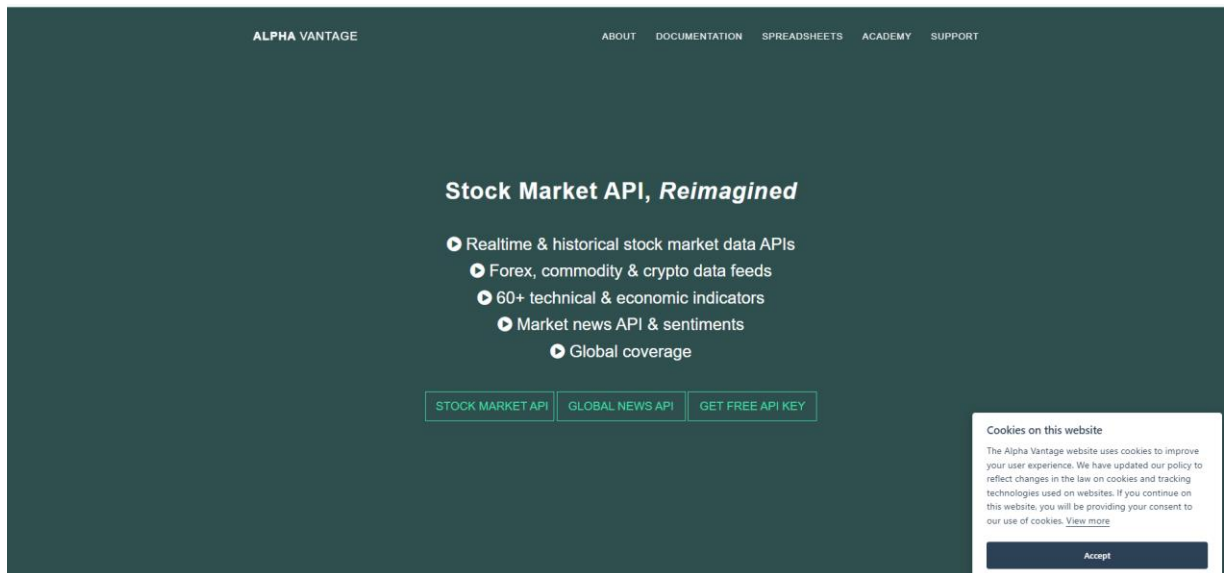
1.1. Những phần chính

- Sử dụng Python và fash API lấy dữ liệu giá bitcoin từ một API có tên alphavantage.
- Sử dụng node-red lấy dữ liệu từ fash API đó lưu vào cơ sở dữ liệu.
- Sử dụng ASP.NET để lấy dữ liệu từ SQL Server và hiển thị lên trang web.
- Thiết kế giao diện cho trang web.

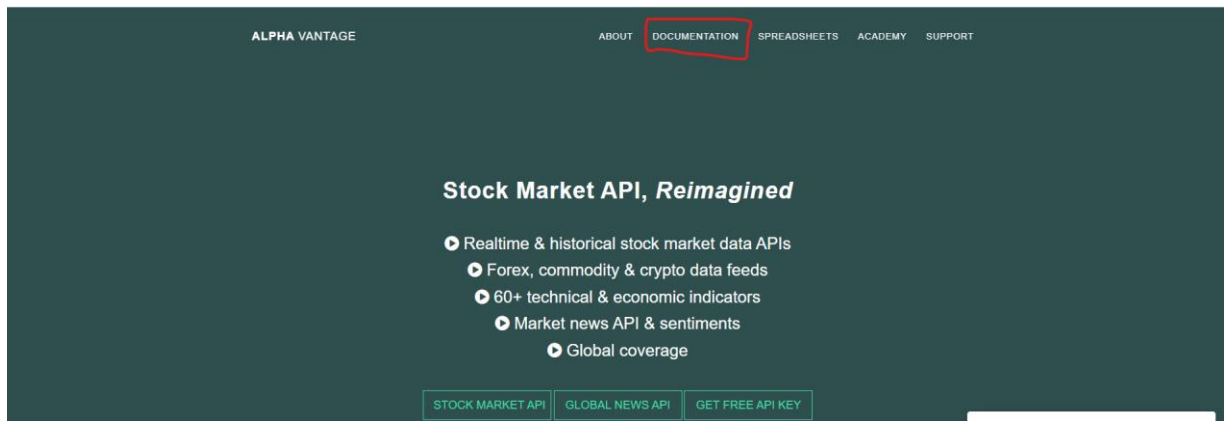
CHƯƠNG II: LẤY DỮ LIỆU BITCOIN

2.1. Trang web lấy dữ liệu

-Trang web có tên ALPHA VANTAGE



- B1: Trộn vào DOCUMENTATION



B2: Trộn vào Exchange Rates ở phần Cryptocurrencies

Fundamental Data

- Company Overview
- Income Statement **Trending**
- Balance Sheet **Trending**
- Cash Flow **Trending**
- Earnings
- Listing & Delisting Status
- Earnings Calendar
- IPO Calendar

Forex (FX)

- Exchange Rates **Trending**
- Intraday **Premium**
- Daily
- Weekly
- Monthly

Cryptocurrencies

- Exchange Rates **Trending**
- Intraday **Premium**
- Daily
- Weekly
- Monthly

Commodities

APIs under this section provide a wide range of data feed for digital and crypto currencies such as Bitcoin.

CURRENCY_EXCHANGE_RATE **Trending**

This API returns the realtime exchange rate for any pair of digital currency (e.g., Bitcoin) or physical currency (e.g., USD).

API Parameters

Required: function
The function of your choice. In this case, `function=CURRENCY_EXCHANGE_RATE`

Required: from_currency
The currency you would like to get the exchange rate for. It can either be a physical currency or digital/crypto currency. For example: `from_currency=USD` or `from_currency=BTC`.

Required: to_currency
The destination currency for the exchange rate. It can either be a physical currency or digital/crypto currency. For example: `to_currency=USD` or `to_currency=BTC`.

Required: apikey
Your API key. Claim your free API key [here](#).

Examples (click for JSON output)

Bitcoin to Euro:
https://www.alobavantage.co/query/function=CURRENCY_EXCHANGE_RATE&from_currency=BTC&to_currency=EUR&apikey=deu

US Dollar to Japanese Yen:
https://www.alobavantage.co/query/function=CURRENCY_EXCHANGE_RATE&from_currency=USD&to_currency=JPY&apikey=deu

Language-specific guides

- Python
- NodeJS
- PHP
- C#/NET
- Other

B3: Ấn vào link của Bitcoin Euro

Fundamental Data

- Company Overview
- Income Statement **Trending**
- Balance Sheet **Trending**
- Cash Flow **Trending**
- Earnings
- Listing & Delisting Status
- Earnings Calendar
- IPO Calendar

Forex (FX)

- Exchange Rates **Trending**
- Intraday **Premium**
- Daily
- Weekly
- Monthly

Cryptocurrencies

- Exchange Rates **Trending**
- Intraday **Premium**
- Daily
- Weekly
- Monthly

Commodities

APIs under this section provide a wide range of data feed for digital and crypto currencies such as Bitcoin.

CURRENCY_EXCHANGE_RATE **Trending**

This API returns the realtime exchange rate for any pair of digital currency (e.g., Bitcoin) or physical currency (e.g., USD).

API Parameters

Required: function
The function of your choice. In this case, `function=CURRENCY_EXCHANGE_RATE`

Required: from_currency
The currency you would like to get the exchange rate for. It can either be a physical currency or digital/crypto currency. For example: `from_currency=USD` or `from_currency=BTC`.

Required: to_currency
The destination currency for the exchange rate. It can either be a physical currency or digital/crypto currency. For example: `to_currency=USD` or `to_currency=BTC`.

Required: apikey
Your API key. Claim your free API key [here](#).

Examples (click for JSON output)

Bitcoin to Euro:
https://www.alobavantage.co/query/function=CURRENCY_EXCHANGE_RATE&from_currency=BTC&to_currency=EUR&apikey=deu

US Dollar to Japanese Yen:
https://www.alobavantage.co/query/function=CURRENCY_EXCHANGE_RATE&from_currency=USD&to_currency=JPY&apikey=deu

Language-specific guides

- Python
- NodeJS
- PHP
- C#/NET
- Other

- Cuối cùng sẽ hiển thị ra chuỗi json chứa thông tin về giá của bitcoin

```
{
  "Realtime Currency Exchange Rate": {
    "1. From_Currency Code": "BTC",
    "2. From_Currency Name": "Bitcoin",
    "3. To_Currency Code": "EUR",
    "4. To_Currency Name": "Euro",
    "5. Exchange Rate": "63616.49000000",
    "6. Last Refreshed": "2024-05-25 09:19:10",
    "7. Time Zone": "UTC",
    "8. Bid Price": "63613.84300000",
    "9. Ask Price": "63617.24700000"
  }
}
```


2.2. Sử dụng python và fash API lấy dữ liệu

- Sử dụng 2 thư viện là fash API và requests
- Dùng lệnh pip install <tên thư viện> để cài đặt các thư viện.

```
from fastapi import FastAPI
import requests

app = FastAPI()

@app.get("/")
async def read_data():
    api_key = "1FOA33CQKAGPWIVJ"
    url = f"https://www.alphavantage.co/query?function=CURRENCY_EXCHANGE_RATE&from_currency=BTC&to_currency=EUR&apikey={api_key}"
    response = requests.get(url)
```

- Khởi tạo một ứng dụng fash api.
- Tạo một api key (key được cung cấp từ trang web) và một url dẫn đến endpoint chứa dữ liệu.
- cuối cùng gửi yêu cầu get đến trang.

```
if response.status_code == 200:
    # chuyển đổi json thành 1 đối tượng python
    exchange_data = response.json().get("Realtime Currency Exchange Rate")
    if exchange_data:
        # Truy xuất các phần tử cụ thể từ chuỗi JSON từ api alphavantage
        from_currency_code = exchange_data.get("1. From_Currency Code")
        from_currency_name = exchange_data.get("2. From_Currency Name")
        to_currency_code = exchange_data.get("3. To_Currency Code")
        to_currency_name = exchange_data.get("4. To_Currency Name")
        exchange_rate = exchange_data.get("5. Exchange Rate")
        last_refreshed = exchange_data.get("6. Last Refreshed")
        time_zone = exchange_data.get("7. Time Zone")
        bid_price = exchange_data.get("8. Bid Price")
        ask_price = exchange_data.get("9. Ask Price")
```

- Kiểm tra xem yêu cầu http có thành công hay không (mã 200 nghĩa là thành công)
- Truy xuất vào các phần tử json và lưu lại thành các biến trên (làm như vậy để em có thể truyền paramater ở node sql trên node-red. Do phần tử json từ đầu có dấu cách không lấy ra được).

```

# Trả về các phần tử cụ thể dưới dạng JSON( e chỉnh lại tên của phần tử để lưu vào db)
return {
    "from_currency_code": from_currency_code,
    "from_currency_name": from_currency_name,
    "to_currency_code": to_currency_code,
    "to_currency_name": to_currency_name,
    "exchange_rate": exchange_rate,
    "last_refreshed": last_refreshed,
    "time_zone": time_zone,
    "bid_price": bid_price,
    "ask_price": ask_price
}

else:
    return {"error": "Không tìm thấy dữ liệu"}

else:
    return {"error": "Không thể nạp dữ liệu"}#yêu cầu http không thành công trả về tb này

```

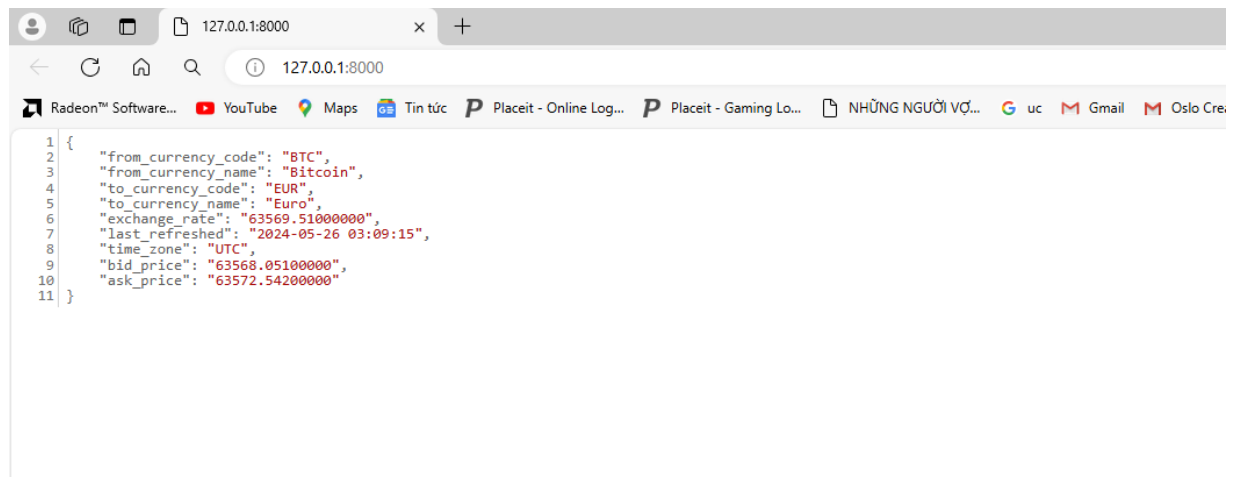
- Trả về các phần tử dưới dạng json.
- Sử dụng lệnh `uvicorn main:app --reload` để chạy file

```

PS D:\tai lieu hoc tap\laptrinhpython\bit> uvicorn main:app --reload
INFO: Will watch for changes in these directories: ['D:\\tai lieu hoc tap\\laptrinhpython\\bit']
INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO: Started reloader process [17648] using WatchFiles
INFO: Started server process [13940]
INFO: Waiting for application startup.
INFO: Application startup complete.

```


Kết quả: Sau khi chạy lên máy chủ web có địa chỉ `http://127.0.0.1:8000`




2.2. Tạo cơ sở dữ liệu

Tạo 2 bảng là Coin và Price (dù lấy mỗi đồng bitcoin nhưng em tạo thêm bảng coin để có thể sau này tìm được một trang web cung cấp api lấy được thêm những đồng coin khác).

- Bảng Coin: Dùng để lưu id và tên của đồng điện tử

	Column Name	Data Type	Allow Nulls
	cid	int	<input type="checkbox"/>
	name	nvarchar(50)	<input type="checkbox"/>
			<input type="checkbox"/>

- Bảng Price gồm: dùng để lưu giá theo từng thời gian cụ thể

	Column Name	Data Type	Allow Nulls
	id	int	<input type="checkbox"/>
	cid	int	<input type="checkbox"/>
	priceCoin	decimal(18, 0)	<input checked="" type="checkbox"/>
	time	datetime	<input checked="" type="checkbox"/>

-Store Procedure gồm: Lưu dữ liệu từ trên node-red vào cơ sở dữ liệu, tạo chuỗi json từ bảng Price.

+ lưu dữ liệu: ban đầu em chỉ khai báo 3 biến vì id ở bảng này để tự tăng, khi trả về các giá trị đã chèn vào bảng thì gán hàm tự tăng vào id.

```

]ALTER PROCEDURE [dbo].[SP_insertPrice]
    @cid int,
    @priceCoin decimal(18, 0),
    @time datetime
AS
]BEGIN
    SET NOCOUNT ON;

    -- Chèn dữ liệu vào bảng 'history' với cột 'sid' và 'time' được cung cấp
]    INSERT INTO Price(cid, priceCoin, time)
    VALUES (@cid, @priceCoin,@time);

    -- Trả về các giá trị đã chèn vào bảng 'history'
    SELECT SCOPE_IDENTITY() AS id, @cid AS sid, @priceCoin AS value,@time AS time;
]END

```

+ Tạo chuỗi json từ bảng Price (để ASP.net gọi ở phần sau): lấy ra 2 giá trị là giá của bitcoin và thời gian của mức giá đó và chuyển thành json

```
ALTER PROCEDURE [dbo].[Sp_Bit]
AS
BEGIN
    SET NOCOUNT ON;


    -- Tạo JSON từ bảng history
    DECLARE @json nvarchar(max);

    SELECT @json = (
        SELECT
            1 AS ok,
            'ok' AS msg,
            (SELECT
                [id],
                [cid],
                CONVERT(nvarchar(50), [priceCoin]) AS value,
                CONVERT(nvarchar(50), [time]) AS time
                FROM Price
                FOR JSON PATH) AS data
            FOR JSON PATH, WITHOUT_ARRAY_WRAPPER
        );

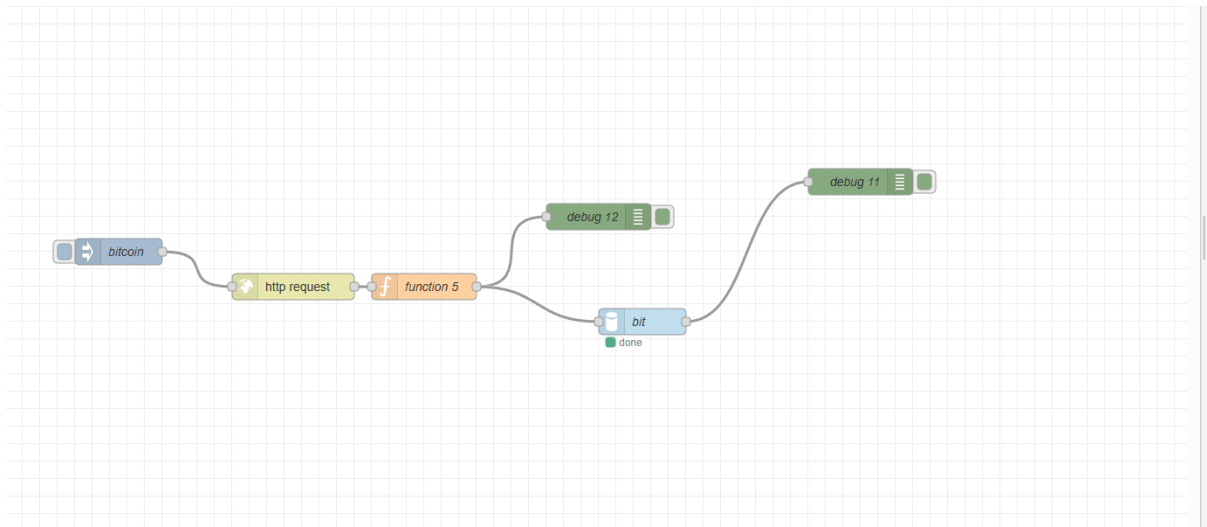
    SELECT @json AS json;
END
```

2.3. Sử dụng node-red

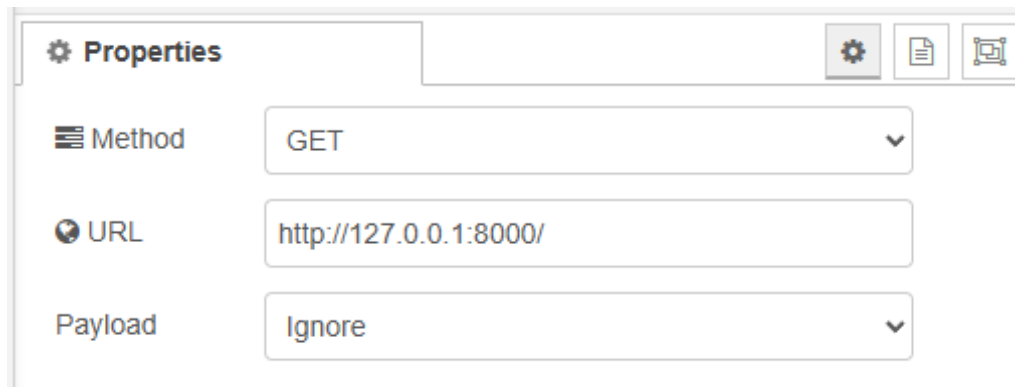
- Em có sử dụng nssm để tạo service cho node-red chạy khi bật máy nhưng khi bật service đó bị pause, em có tham khảo một số cách trên mạng cũng như thử xóa đi cài lại thì kết quả vẫn như vậy nên em chưa giải quyết được vấn đề này.

Name	PID	Description	Status	Group
 node-redsv	5268	node-redsv	Paused	

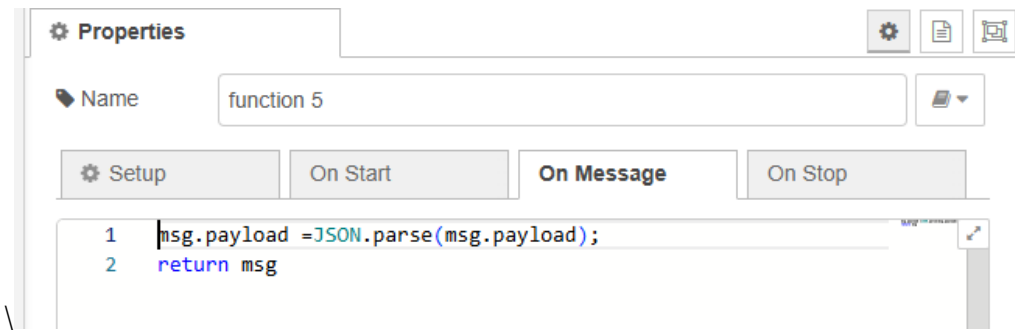
- Cấu trúc của phần node red để lấy dữ liệu và lưu về cơ sở dữ liệu.



- http request: gán link sau khi chạy file python bằng lệnh uvircon vào để node-red thực hiện get.



- Function 5: viết code” msg.payload =JSON.parse(msg.payload);return msg; “ để chắc chắn dữ liệu trả về là json.



Edit MSSQL node > **Edit MSSQL-CN node**

Delete Cancel Update

Properties

Name bit

Server 127.0.0.1

Port 1443

Username sa

Password

Domain

Database bit

TDS Version 7_4 (SQL Server 2012 ~ 2022)

Use Encryption? ☒

SQL Databases hosted on Azure will need this checked.

Trust Certificate? ☒

If unchecked, SQL Server will try to validate the server SSL certificate and will terminate the connection if validation fails

- Node mssql có tên bit:



B1: cấu hình để kết nối với cơ sở dữ liệu gồm: Địa chỉ local, cổng (tùy vào người đặt ở phần cài đặt sql server configuration), tài khoản và mật khẩu sql, tên database.

+ Tích thêm vào phần Trust Certificate để có thể kết nối đến cơ sở dữ liệu.

+ Trước em có cài thử node-red-contrib-mssql thì cấu hình xong không thể kết nối được.

 node-red-contrib-mssql

Node for node-red to connect to a Microsoft MS SQL Database

 0.0.7  7 years, 8 months ago

conflict

+ Sau đó cài node-red-contrib-mssql-plus giống của thầy thì đã ổn và hoạt động bình thường.

B2: gọi store procedure và truyền tham số

The screenshot shows a database client interface with the following sections:

- Properties:**
 - Connection: bit
 - Name: bit
 - Query mode: Execute Procedure
 - Query: Editor
- Query Editor:** Contains the text `1 SP_insertPrice`.
- Parameters:**
 - Editor: A table with three input parameters:

Input	Name	Type	Value
Input	cid	int	1
Input	priceCo	Decin	msg.payload.exchange_
Input	time	Date1	msg.payload.last_refrest
 - + add
 - Parse Mustache: ☒

- Kết quả sau khi chạy node

```
5/25/2024, 3:31:46 PM node: debug 12
msg.payload : Object
  {
    from_currency_code: "BTC",
    from_currency_name: "Bitcoin",
    to_currency_code: "EUR",
    to_currency_name: "Euro",
    exchange_rate: "63628.72000000" ... }

5/25/2024, 3:31:46 PM node: debug 11
msg.payload : array[1]
  array[1]
    0: object
      id: 9
      sid: 1
      value: 63629
      time: "2024-05-25T01:31:13.000Z"
```

CHƯƠNG III: THIẾT KẾ GIAO DIỆN

3.1. Tạo File aspx để lấy dữ liệu từ cơ sở dữ liệu

```
using System;
using System.Collections.Generic;

using System.Data.SqlClient;
using System.Data;

using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace giamsatbitcoin
{
    1 reference
    public partial class api : System.Web.UI.Page
    {
        0 references
        protected void Page_Load(object sender, EventArgs e)
        {
            //thiết lập chuỗi kết nối với csdl
            string connectionString = "Data Source=127.0.0.1,1443;Initial Catalog=bit;User Id=sa;Password=123;";
            using (SqlConnection connection = new SqlConnection(connectionString))
            {
                using (SqlCommand command = new SqlCommand("Sp_Bit", connection))// gọi store procedure
                {
                    command.CommandType = CommandType.StoredProcedure;

                    try
                    {
                        connection.Open();// mở chuỗi kết nối đến csdl
                        object kq = command.ExecuteScalar();//thực hiện sp và trả về dữ liệu( sp này trả về chuỗi json)
                        string json = (string)kq;
                        this.Response.ContentType = "application/json";
                        this.Response.Write(json);
                        Console.WriteLine(json);
                    }
                    catch (Exception ex)
                    {
                        this.Response.ContentType = "application/json";// đặt nd phản hồi thành kiểu json
                        this.Response.Write("{\"ok\":0,\"msg\":\"" + ex.Message + "\"}");
                    }
                }
            }
        }
    }
}
```

- thiết lập chuỗi kết nối tới cơ sở dữ liệu

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


- Gọi store procedure và trả về dữ liệu dưới dạng json

```
using (SqlCommand command = new SqlCommand("Sp_Bit", connection))// gọi store procedure
{
    command.CommandType = CommandType.StoredProcedure;

    try
    {
        connection.Open();// mở chuỗi kết nối đến csdl
        object kq = command.ExecuteScalar();//thực hiện sp và trả về dữ liệu( sp này trả về chuỗi json)
        string json = (string)kq;
        this.Response.ContentType = "application/json";
        this.Response.Write(json);
        Console.WriteLine(json);
    }
    catch (Exception ex)
    {
        this.Response.ContentType = "application/json";// đặt nd phản hồi thành kiểu json
        this.Response.Write("{\"ok\":0,\"msg\":\"" + ex.Message + "\"}");
    }
}
```

3.2. file html

- Link file js, css và một số thư viện khác vào file html.

```
<script src="Scripts/jquery-3.7.1.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/exceljs/dist/exceljs.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
```

- Tạo các thẻ để hiển thị những thông tin cần thiết.

```
<body>
    <div class="container">
        <h1>Bitcoin Mần Gô</h1>
        <div class="current-price">
            <p>Giá BitCoin hiện tại: <span id="currentPrice"></span> Euro</p>
        </div>
        <div class="conversion">
            <button id="exportExcelBtn">Xuất Excel</button>

            <label for="bitcoinAmount">Nhập số lượng Bitcoin:</label>
            <input type="number" id="bitcoinAmount" step="0.0001" min="1">
            <p>Giá trị tương đương bằng VND: <span id="vndValue"></span></p>
            <p>Giá trị tương đương bằng EUR: <span id="eurValue"></span></p>
        </div>
        <canvas id="myChart" width="800" height="400"></canvas>
        <div class="history">
            <h2>Lịch sử giá Bitcoin</h2>
            <ul id="historyList"></ul>
        </div>
    </div>
    <script src="script.js"></script>
</body>
```

3.3. file css

- CSS Các thuộc tính của file html để trang web đẹp và sinh động hơn

```
style.css  X  script.js  index.html  api.aspx.cs
1  body {
2      font-family: 'Arial', sans-serif;
3      background-color: #f4f4f9;
4      margin: 0;
5      padding: 0;
6      display: flex;
7      justify-content: center;
8      align-items: center;
9      height: 100vh;
10     color: #333;
11 }
12
13 .container {
14     width: 90%;
15     max-width: 90%;
16     background: #fff;
17     padding: 20px;
18     box-shadow: 0 2px 8px rgba(0, 0, 0, 0.1);
19     border-radius: 8px;
20 }
21
22 h1 {
23     text-align: center;
24     margin-bottom: 20px;
25     padding-top: 150px
26 }
27
28 .current-price, .conversion {
29     text-align: center;
30     font-size: 1.2em;
31     margin-bottom: 20px;
32 }
33
34
35 #myChart {
36     width: 100%;
37     height: 400px;
38     max-height: 400px; /* Ensuring the chart doesn't grow indefinitely */
39 }
40 #currentPrice{
```

3.4. file Javascript

- Dữ dụng thư viện jquery để viết.

```
fetch('api.aspx')
    .then(response => {
        if (!response.ok) {
            throw new Error('Network response was not ok ' + response.statusText);
        }
        return response.json();
    })
    .then(data => {
        console.log(data); // Ghi nhật ký dữ liệu trả về để kiểm tra

        if (data.ok !== 1) {
            throw new Error('API trả về lỗi: ' + data.msg);
        }

        let labels = [];
        let bitcoinValues = [];

        12 references
```

- dùng fetch('api.aspx') để gửi yêu cầu HTTP đến API và nhận phản hồi từ máy chủ.
- Sử dụng phương thức .then() để xử lý phản hồi từ máy chủ và kiểm tra có lỗi hay không, nếu không lỗi chuyển đổi dữ liệu nhận được thành định dạng JSON.

- Tiếp tục sử dụng .then() để xử lý dữ liệu JSON đã được chuyển đổi. Kiểm tra nếu dữ liệu trả về từ API không hợp lệ thì sẽ báo lỗi .
- lưu dữ liệu nhận(ở đây là thời gian tương ứng với mỗi mẫu dữ liệu bitcoin) và giá của bitcoin

```
12 references
$.each(data.data, function (index, item) {
    labels.push(item.time);
    bitcoinValues.push(parseFloat(item.value));
});
```

- Lặp qua các phần tử của mảng dữ liệu trả về có tên data sau đó lưu dữ liệu thời gian vào mảng labels và giá vào mảng bitcoinValues.

```
// Lấy giá Bitcoin hiện tại
let currentPrice = bitcoinValues[bitcoinValues.length - 1];
$('#currentPrice').text(currentPrice.toFixed(2) + ' EUR');
```

- lấy giá bitcoin hiện tại

```
let historyList = $('#historyList');
for (let i = 0; i < bitcoinValues.length; i++) {
    // Thêm một mục vào danh sách lịch sử cho mỗi phần tử
    historyList.append('<li>${labels[i]}: ${bitcoinValues[i].toFixed(2)} EUR</li>');
}
```

- hiển thị lịch sử của bitcoin gồm giá và thời gian cụ thể

```
// Chuyển đổi giá
const eurToVndRate = 25000;

// Hàm tính toán giá trị tương đương khi nhập số lượng Bitcoin
0 references
$('#bitcoinAmount').on('input', function () {
    let bitcoinAmount = parseFloat($('#bitcoinAmount').val());
    if (isNaN(bitcoinAmount) || bitcoinAmount < 1) {
        $('#vndValue').text('');
        $('#eurValue').text('');
        return;
    }

    let equivalentVND = bitcoinAmount * currentPrice * eurToVndRate;
    let equivalentEUR = bitcoinAmount * currentPrice;

    $('#vndValue').text(equivalentVND.toLocaleString('vi-VN') + ' VND');
    $('#eurValue').text(equivalentEUR.toFixed(2) + ' EUR');
});
```

- Tạo biến eurToVndRate để lưu giá trị vnd phục vụ chuyển đổi (giá chỉ là minh họa)

- Hàm bắt sự kiện khi nhập nếu nhập nhỏ hơn một sẽ không hiển thị gì nếu không thì sẽ tính toán giá euro và vnd tùy vào số lượng bitcoin nhập vào.

```
var ctx = $('#myChart');
var myChart = new Chart(ctx, {
  type: 'line',
  data: {
    labels: labels.slice(-7),
    datasets: [{
      label: 'Bitcoin Price',
      data: bitcoinValues.slice(-7),
      backgroundColor: 'rgba(75, 192, 192, 0.2)',
      borderColor: 'rgba(75, 192, 192, 1)',
      borderWidth: 2, // Độ dày của đường
      pointBackgroundColor: 'rgba(75, 192, 192, 1)',
      pointBorderColor: 'rgba(75, 192, 192, 1)',
      pointHoverBackgroundColor: 'rgba(75, 192, 192, 1)',
      pointHoverBorderColor: 'rgba(75, 192, 192, 1)',
      pointRadius: 3,
      pointHoverRadius: 5,
      pointHitRadius: 10,
      spanGaps: true
    }]
  },
  options: {
    maintainAspectRatio: true,
    responsive: true,
    scales: {
      y: {
        beginAtZero: false,
        grid: {
          color: 'rgba(0, 0, 0, 0.1)'
        }
      },
      x: {
        grid: {
          color: 'rgba(0, 0, 0, 0.1)'
        }
      }
    },
    plugins: {
      tooltip: {
        backgroundColor: 'rgba(0, 0, 0, 0.8)',
        borderColor: 'rgba(0, 0, 0, 0.8)',
        borderWidth: 1,
        titleColor: 'fff',
        bodyColor: 'fff',
        footerColor: 'fff'
      }
    },
    elements: {
      line: {
        tension: 0.4 // Độ cong của đường biểu đồ
      }
    }
  }
});
.catch(error => console.error('Lỗi khi lấy dữ liệu:', error));
```

- sử dụng google chart để vẽ biểu đồ giá 7 ngày gần nhất. Có thể chỉnh các thông số tùy theo ý của mình như màu, độ cong đường biểu đồ và độ dày của đường...

- Hàm xuất file excel

```
function createExcel(data) {
  // Kiểm tra dữ liệu
  if (!data || !Array.isArray(data) || data.length === 0) {
    console.error('Dữ liệu không hợp lệ');
    return;
  }

  // Tạo workbook và worksheet
  var workbook = new ExcelJS.Workbook();
  var worksheet = workbook.addWorksheet('Bitcoin Prices');

  // Tạo một hàng trống
  worksheet.addRow([]);

  // Tạo một hàng tiêu đề lớn
  var titleRow = worksheet.addRow(['Dữ liệu giá Bitcoin']);
  titleRow.font = { size: 16, bold: true };
  worksheet.mergeCells('A2:B2'); // Hợp nhất các ô tiêu đề
  titleRow.getCell(1).alignment = { horizontal: 'center' }; // Căn giữa tiêu đề

  // Thêm một hàng trống sau tiêu đề
  worksheet.addRow([]);

  // Thêm tiêu đề cho cột và định dạng
  var headerRow = worksheet.addRow(['Thời Gian', 'Giá (EUR)']);
  headerRow.font = { bold: true };
  headerRow.alignment = { horizontal: 'center' }; // Căn giữa tiêu đề cột
  headerRow.eachCell((cell) => {
    cell.fill = {
      type: 'pattern',
      pattern: 'solid',
      fgColor: { argb: 'FFFF00' } // Màu nền vàng cho tiêu đề cột
    };
  });
```

- kiểm tra dữ liệu xem có hợp lệ hay không
- tạo bảng
- thêm tiêu đề vào file
- thêm tiêu đề cho các cột, hợp nhất các ô và căn giữa...

```
    cell.border = {
      top: { style: 'thin' },
      left: { style: 'thin' },
      bottom: { style: 'thin' },
      right: { style: 'thin' }
    };
  });

  // Thêm dữ liệu từ API vào worksheet và định dạng
  data.forEach(item => {
    var row = worksheet.addRow([item.time, item.value]);
    row.alignment = { horizontal: 'center' }; // Căn giữa dữ liệu
    row.eachCell((cell) => {
      cell.border = {
        top: { style: 'thin' },
        left: { style: 'thin' },
        bottom: { style: 'thin' },
        right: { style: 'thin' }
      };
    });
  });

  // Điều chỉnh độ rộng cột
  worksheet.columns = [
    { width: 20 },
    { width: 20 }
  ];
};
```

- thêm dữ liệu từ api vào các cột (dữ liệu có giá và thời gian)

```

workbook.xlsx.writeBuffer().then(buffer => {
  var blob = new Blob([buffer], { type: 'application/vnd.openxmlformats-officedocument.spreadsheetml.sheet' });
  var url = URL.createObjectURL(blob);

  // Tạo liên kết tải xuống và thêm vào DOM
  var a = document.createElement('a');
  a.href = url;
  a.download = 'bitcoin_prices.xlsx';
  a.click();
  URL.revokeObjectURL(url);
});

```

```

// Tạo tệp Excel
workbook.xlsx.writeBuffer().then(buffer => {
  var blob = new Blob([buffer], { type: 'application/vnd.openxmlformats-officedocument.spreadsheetml.sheet' });
  var url = URL.createObjectURL(blob);

  // Tạo liên kết tải xuống và thêm vào DOM
  var a = document.createElement('a');
  a.href = url;
  a.download = 'bitcoin_prices.xlsx';
  a.click();
  URL.revokeObjectURL(url);
});
}

```

Tạo một tệp Excel từ workbook đã tạo trước đó bằng cách sử dụng phương thức `writeBuffer()` của đối tượng `workbook`.

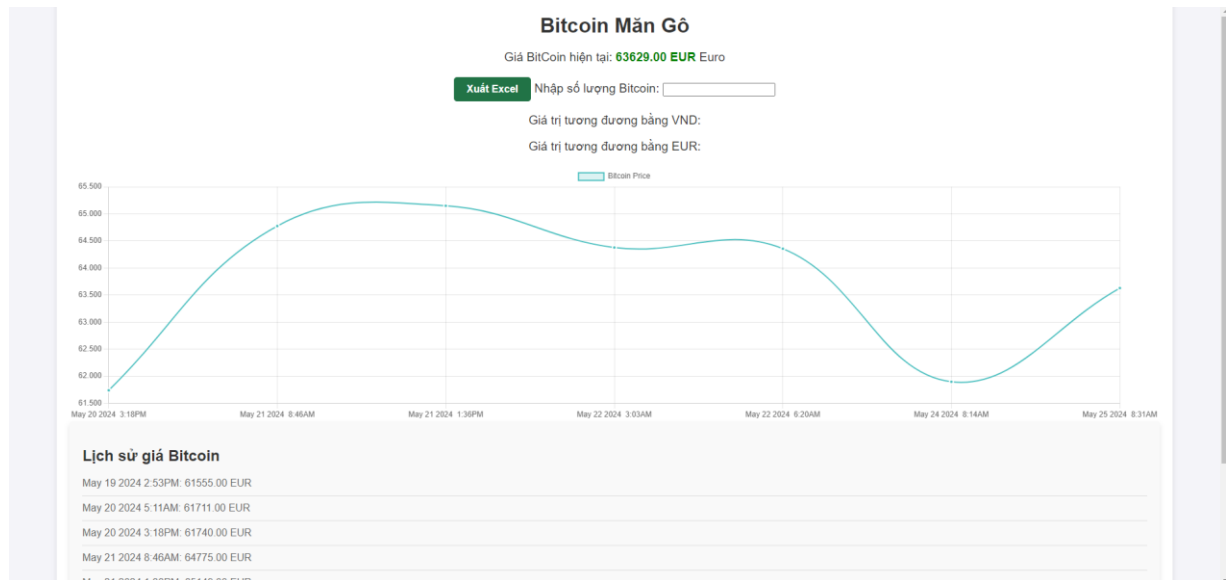
-Sau đó, trong hàm `then` của promise, nó tạo một blob từ buffer bằng cách sử dụng đối tượng `Blob` của JavaScript. Blob là một đối tượng lưu trữ dữ liệu như tệp dưới dạng một loại MIME cụ thể.

- Sau khi tạo blob, nó tạo một URL đối tượng từ blob bằng cách sử dụng `URL.createObjectURL()`. URL này là một địa chỉ đặc biệt dẫn đến dữ liệu blob.

- Tiếp theo, nó tạo một phần tử `<a>` (liên kết) mới trong DOM và thiết lập thuộc tính `href` của nó để trỏ đến URL của blob. Nó cũng thiết lập thuộc tính `download` để đặt tên của tệp khi được tải xuống.

CHƯƠNG IV: KIỂM THỬ

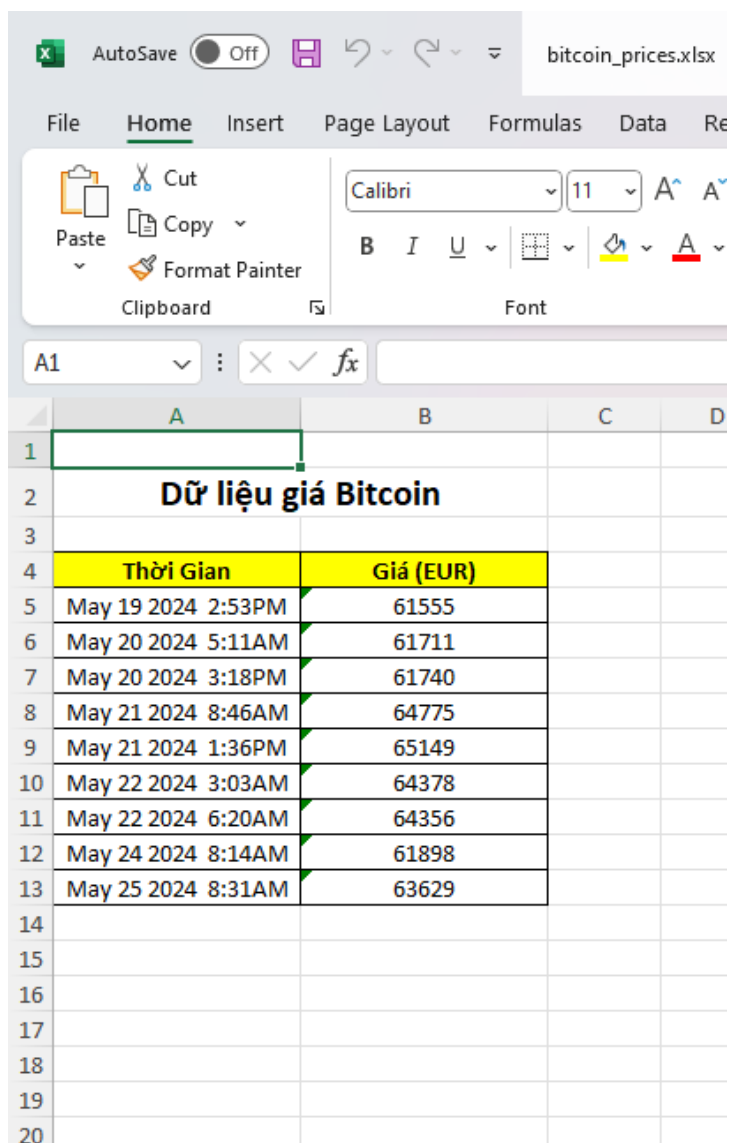
4.1. Giao diện web



4.2. Nhập số lượng bitcoin sẽ quy ra giá tiền cụ thể



4.3. Xuất excel



The screenshot shows the Microsoft Excel interface with the file 'bitcoin_prices.xlsx' open. The 'Home' tab is selected in the ribbon. The 'Clipboard' group shows 'Cut', 'Copy', 'Paste', and 'Format Painter' options. The 'Font' group shows the font 'Calibri', size '11', and various formatting options like bold, italic, underline, and text color. The active cell is A1. The spreadsheet contains a table of Bitcoin prices with the following data:

Thời Gian	Giá (EUR)
May 19 2024 2:53PM	61555
May 20 2024 5:11AM	61711
May 20 2024 3:18PM	61740
May 21 2024 8:46AM	64775
May 21 2024 1:36PM	65149
May 22 2024 3:03AM	64378
May 22 2024 6:20AM	64356
May 24 2024 8:14AM	61898
May 25 2024 8:31AM	63629

KẾT LUẬN

- Qua bài tập lớn em đã học được cách lấy dữ liệu từ một trang web cung cấp api.
- sử dụng node red để lưu được dữ liệu từ api vào cơ sở dữ liệu.
- Lấy dữ liệu từ cơ sở dữ liệu và đưa lên web.
- Một số yêu cầu cao hơn như đưa lên IIS thì em chưa làm được.

Github:

