

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: Ma Bách Duy

Mssv: K205480106010

Lớp: K56KMT

Giáo viên hướng dẫn: Đỗ Duy Cốp

Thái Nguyên – 2024

TRƯỜNG ĐHKTCN

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM

KHOA ĐIỆN TỬ

Độc lập - Tự do - Hạnh phúc

PHIẾU GIAO ĐỀ TÀI BÀI TẬP LỚN MÔN LẬP TRÌNH PYTHON

Họ và tên: Ma Bách Duy

Mssv: K205480106010

Lớp: K56KMT

Ngành: Kỹ thuật máy tính

Giáo viên hướng dẫn: Đỗ Duy Cốp

Ngày giao đề tài: 14/05/2024

Ngày hoàn thành: 26/05/2024

1. Tên đề tài: Xây dựng website thông tin về vũ trụ và trái đất

2. Nội dung:

- Giới thiệu ngôn ngữ lập trình
- Lý do chọn đề tài
- Các bước thực hiện

3. Báo cáo, chương trình:

- Quyển báo cáo
- Demo bài làm trên máy tính cá nhân

GIÁO VIÊN HƯỚNG DẪN

(ký và ghi rõ họ tên)

[illegible]

(Ký ghi rõ họ tên)

MỤC LỤC

MỤC LỤC	4
LỜI MỞ ĐẦU	5
CHƯƠNG 1:GIỚI THIỆU VỀ NGÔN NGỮ PYTHON.....	6
CHƯƠNG 2 : TỔNG QUAN VỀ ĐỀ TÀI	7
2.1.Giới thiệu	7
2.2. Lý do lựa chọn đề tài	7
2.3.Mục tiêu nghiên cứu	7
CHƯƠNG 3 : CÁC BƯỚC TIẾN HÀNH	8
3.1. Tìm kiếm API dữ liệu	8
3.1.1. APOD (Astronomy Picture of the Day)	8
3.1.2. EONET (Earth Observatory Natural Event Tracker).....	9
3.1.3. Mars Rover Photos	10
3.2. Tạo các endpoint bằng FastAPI	11
3.3. Thiết kế Node-red và cơ sở dữ liệu	14
3.3.1. Thiết kế CSDL (dùng SQLSEVER).....	14
3.3.2.Thiết kế NodeRed	17
3.4. Thiết kế giao diện	23
3.5.Chạy chương trình	25
KẾT LUẬN	30

LỜI MỞ ĐẦU

Trong cuộc sống hiện đại ngày nay, việc khám phá và hiểu biết về vũ trụ và Trái Đất là điều vô cùng quan trọng. Website thông tin về vũ trụ và Trái Đất ra đời nhằm đáp ứng nhu cầu này, mang lại kiến thức phong phú và cập nhật cho người sử dụng.

Website của em sử dụng các công nghệ tiên tiến và tích hợp dữ liệu từ API của NASA để cung cấp thông tin chính xác về các hiện tượng và khám phá mới nhất trong lĩnh vực thiên văn học và không gian. Các dữ liệu chi tiết và hình ảnh chất lượng cao từ NASA được liên tục cập nhật, giúp người dùng dễ dàng tiếp cận với những thông tin mới nhất về các sự kiện tự nhiên (EONET), ảnh thiên văn(APOD) và các ảnh chụp từ sao hỏa.

Một trong những điểm nổi bật của website này là khả năng cung cấp thông tin theo thời gian thực về các sự kiện vũ trụ và các nghiên cứu khoa học đang diễn ra. Điều này không chỉ giúp người dùng có cái nhìn sâu sắc hơn về vũ trụ bao la mà còn nâng cao nhận thức về tầm quan trọng của việc bảo vệ hành tinh xanh của chúng ta.

Website thông tin về vũ trụ và Trái Đất không chỉ là một nền tảng kiến thức hữu ích mà còn là người bạn đồng hành đáng tin cậy trong việc khám phá và hiểu rõ hơn về môi trường sống của chúng ta. Với giao diện thân thiện và dữ liệu phong phú, website này mang lại sự hài lòng và tiện ích cho người dùng, góp phần nâng cao chất lượng cuộc sống và tri thức hàng ngày.

Em xin chân thành cảm ơn sự quan tâm giúp đỡ của thầy **Đỗ Duy Cốp** đã tận tình chỉ bảo trong quá trình em thực hiện đề tài này.

CHƯƠNG 1: GIỚI THIỆU VỀ NGÔN NGỮ PYTHON

Python là một ngôn ngữ lập trình được sử dụng rộng rãi trong các ứng dụng web, phát triển phần mềm, khoa học dữ liệu và máy học (ML). Các nhà phát triển sử dụng Python vì nó hiệu quả, dễ học và có thể chạy trên nhiều nền tảng khác nhau. Phần mềm Python được tải xuống miễn phí, tích hợp tốt với tất cả các loại hệ thống và tăng tốc độ phát triển.



Những lợi ích của Python bao gồm:

- Các nhà phát triển có thể dễ dàng đọc và hiểu một chương trình Python vì ngôn ngữ này có cú pháp cơ bản giống tiếng Anh.
- Python giúp cải thiện năng suất làm việc của các nhà phát triển vì so với những ngôn ngữ khác, họ có thể sử dụng ít dòng mã hơn để viết một chương trình Python.
- Python có một thư viện tiêu chuẩn lớn, chứa nhiều dòng mã có thể tái sử dụng cho hầu hết mọi tác vụ. Nhờ đó, các nhà phát triển sẽ không cần phải viết mã từ đầu.
- Các nhà phát triển có thể dễ dàng sử dụng Python với các ngôn ngữ lập trình phổ biến khác như Java, C và C++.
- Cộng đồng Python tích cực hoạt động bao gồm hàng triệu nhà phát triển nhiệt tình hỗ trợ trên toàn thế giới. Nếu gặp phải vấn đề, bạn sẽ có thể nhận được sự hỗ trợ nhanh chóng từ cộng đồng.
- Trên Internet có rất nhiều tài nguyên hữu ích nếu bạn muốn học Python. Ví dụ: bạn có thể dễ dàng tìm thấy video, chỉ dẫn, tài liệu và hướng dẫn dành cho nhà phát triển.
- Python có thể được sử dụng trên nhiều hệ điều hành máy tính khác nhau, chẳng hạn như Windows, macOS, Linux và Unix.

CHƯƠNG 2 : TỔNG QUAN VỀ ĐỀ TÀI

2.1.Giới thiệu

Trong thời đại công nghệ 4.0, việc ứng dụng các giải pháp công nghệ vào cuộc sống hàng ngày ngày càng trở nên phổ biến và cần thiết. Để mọi người tìm hiểu thêm kiến thức và khám phá về vũ trụ và trái đất em đã chọn và thực hiện đề tài "Xây dựng website thông tin về vũ trụ và trái đất" mang lại nhiều thông tin, kiến thức lợi ích thiết thực cho người sử dụng.

2.2. Lý do lựa chọn đề tài

Với sự hứng thú tìm hiểu kiến thức về vũ trụ và ngắm các ảnh thiên văn đẹp đẽ và mong muốn chia sẻ tới mọi người, đó là lý do em lựa chọn đề tài: “Xây dựng website thông tin về vũ trụ và trái đất” gồm các mục sau: APOD (Astronomy Picture of the Day) cung cấp những hình ảnh đẹp và độc đáo về vũ trụ, EONET (Earth Observatory Natural Event Tracker) giúp theo dõi sự kiện tự nhiên trên Trái Đất, và Mars Rover Photos mang lại cái nhìn đầy hứng thú về cuộc khám phá trên sao Hỏa. Sự kết hợp của ba nguồn thông tin này sẽ tạo ra một trang web đa chiều và phong phú, thúc đẩy sự hiểu biết và tò mò về vũ trụ và hành tinh của mọi người.

2.3.Mục tiêu nghiên cứu

- Dữ liệu được cập nhật liên tục để mang đến thông tin nhanh nhất
- Cung cấp giao diện người dùng thân thiện để theo dõi
- Tải ảnh hoặc video để có thể xem trực tiếp cho người dùng

CHƯƠNG 3 : CÁC BƯỚC TIỀN HÀNH

3.1. Tìm kiếm API dữ liệu

API của NASA (<https://api.nasa.gov/>) là một nguồn tài nguyên quý báu cho cộng đồng khoa học và công chúng, cung cấp truy cập vào một loạt các dữ liệu và dịch vụ từ Cơ quan Hàng không và Vũ trụ Hoa Kỳ (NASA). API này cho phép các nhà phát triển, nhà nghiên cứu và người dùng cuối sử dụng các dữ liệu, hình ảnh, video và thông tin thiên văn một cách dễ dàng thông qua các yêu cầu HTTP.

Thông qua API của NASA, người dùng có thể truy cập vào các dự án và nhiệm vụ vũ trụ đang diễn ra, như thông tin về sao Hỏa, Hubble Space Telescope, International Space Station (ISS), Curiosity Rover trên sao Hỏa, và nhiều nhiệm vụ và dự án nghiên cứu khác của NASA.

Sau khi nghiên cứu và đọc thư viện tài liệu về các loại API có thể sử dụng để thiết kế website. Em quyết định chọn API về APOD (Bức ảnh thiên văn trong ngày), EONET (Thông tin về các sự kiện tự nhiên), Mars Rover Photos (Cung cấp ảnh chụp bề mặt sao Hỏa).

3.1.1. APOD (*Astronomy Picture of the Day*)

Một trong những trang web phổ biến nhất tại NASA là APOD. Thực tế, trang web này là một trong những trang web phổ biến nhất trong tất cả các cơ quan liên bang. Nó có sức hút phổ biến như một video của Justin Bieber. Điểm cuối này cấu trúc hình ảnh APOD và siêu dữ liệu liên quan để nó có thể được sử dụng lại cho các ứng dụng khác. Ngoài ra, nếu tham số `concept_tags` được thiết lập thành True, thì các từ khóa được tạo ra từ giải thích hình ảnh được trả về. Những từ khóa này có thể được sử dụng như các hashtag được tự động tạo cho twitter hoặc instagram, nhưng nó chủ yếu giúp cho việc phát hiện ảnh phù hợp. Tham số truy vấn:

Tham số	Kiểu	Mặc định	Mô tả
Date	YYYY-MM-DD	today	Ngày truy xuất hình ảnh APOD
start_date	YYYY-MM-DD	none	Phạm vi ngày bắt đầu, khi yêu cầu ngày cụ thể trong phạm vi này thì không thể sử dụng với ngày ngoài phạm vi
end_date	YYYY-MM-DD	today	Ngày kết thúc của phạm vi ngày
count	int	none	Nếu count được sử dụng thì số ảnh ngẫu nhiên được trả về. Không thể dùng chung với start_date và end_date

thumbs	bool	False	Trả về ảnh thu nhỏ của video, nếu APOD không phải video thì tham số này được bỏ qua
api_key	string	YOUR_KEY	Khóa sử dụng API

3.1.2. EONET (Earth Observatory Natural Event Tracker)

Sử dụng các ứng dụng khách, như Worldview của NASA EOSDIS, người dùng có thể duyệt toàn bộ trái đất hàng ngày và tìm kiếm các sự kiện tự nhiên khi chúng xảy ra. Các cơn bão thường xuyên xuất hiện ở vùng nhiệt đới, cơn bão cát trên sa mạc, cháy rừng vào mùa hè. Những sự kiện này diễn ra liên tục và hình ảnh NRT của NASA có thể đại diện cho tất cả chúng bằng cách sử dụng một loạt các thông số dữ liệu khác nhau. Tuy nhiên, trải nghiệm của người dùng được tùy chỉnh và do đó bị hạn chế bởi ứng dụng khách. Nhưng liệu có một API cung cấp một danh sách được lựa chọn các sự kiện tự nhiên và cung cấp một cách để liên kết những sự kiện đó với các lớp hình ảnh NRT liên quan đến sự kiện? Đó chính là EONET.

Bộ theo dõi Sự kiện Tự nhiên Trái Đất (EONET) là một dịch vụ web nguyên mẫu với mục tiêu:

- Cung cấp một nguồn dữ liệu siêu dữ liệu về các sự kiện tự nhiên được cập nhật liên tục;
- Cung cấp một dịch vụ liên kết những sự kiện tự nhiên đó với các nguồn hình ảnh được bật sẵn thông qua dịch vụ web có chủ đề liên quan (ví dụ: qua WMS, WMTS, v.v.). Vui lòng xem tài liệu API của chúng tôi để tìm hiểu thêm về cách sử dụng các dịch vụ web EONET.

Phát triển của EONET bắt đầu từ năm 2015 và đã được hỗ trợ bởi Trạm Quan sát Trái Đất của NASA và Dự án Hệ thống Thông tin và Dữ liệu Khoa học Trái Đất (ESDIS) của NASA.

Tham khảo thư viện [Tại đây](#)

3.1.3. Mars Rover Photos

API này được thiết kế để thu thập dữ liệu hình ảnh được thu thập bởi các robot Curiosity, Opportunity và Spirit của NASA trên sao Hỏa và làm cho nó dễ dàng hơn để có sẵn cho các nhà phát triển khác, giáo viên và các nhà khoa học công dân. API này được duy trì bởi Chris Cerami.

Mỗi robot có bộ ảnh riêng được lưu trữ trong cơ sở dữ liệu, có thể được truy vấn một cách riêng biệt. Có một số truy vấn khả dụng có thể được thực hiện đối với API. Hình ảnh được tổ chức theo sol (quay của sao Hỏa hoặc ngày) mà chúng được chụp, đếm từ ngày hạ cánh của robot. Ví dụ, một bức ảnh được chụp vào ngày quay thứ 1000 của Curiosity khám phá sao Hỏa sẽ có thuộc tính sol là 1000. Nếu bạn thích tìm kiếm theo ngày Trái Đất mà một bức ảnh được chụp, bạn cũng có thể làm điều đó.

Ngoài việc truy vấn theo ngày, kết quả cũng có thể được lọc bởi máy ảnh mà nó được chụp và các phản hồi sẽ được giới hạn thành 25 ảnh mỗi lần gọi. Các truy vấn có thể trả về hơn 25 ảnh sẽ được chia thành nhiều trang, có thể được truy cập bằng cách thêm tham số 'page' vào truy vấn.

Tham số truy vấn:

Tham số	Kiểu	Mặc định	Mô tả
sol	int	none	sol (phạm vi từ 0 đến giá trị tối đa được tìm thấy trong điểm cuối)
camera	string	all	FHAZ-RHAZ MAST-CHEMCAM MAHLI-MARDI-NAVCAM-PANCAM-MINUTES
page	int	1	25 mục trên mỗi trang được trả về
api_key	string	DEMO_KEY	khóa API

3.2. Thực hiện lấy dữ liệu từ API

3.2.1. Lấy dữ liệu và gửi chúng lên các endpoint

Bước này sẽ cần xử lý lấy dữ liệu bằng thư viện request

Sau khi có được các dữ liệu thô từ API của api.nasa.gov. Thực hiện gửi các dữ liệu đó lên các endpoint:

```
api_key = "NbAeciW6tcGxCk3lF7fRh13vh95ykBTua0oLJDdx"
```

APOD:

```
@app.get("/apod") #endpoint
async def get_data_apod():
    # Gửi yêu cầu GET đến API
    now = datetime.now()
    formatted_date = now.strftime("%Y-%m-%d")
    url = f"http://127.0.0.1:8000/v1/apod/?concept_tags=True&date="
    response = requests.get(url) //Sử dụng tv request để lấy dữ liệu
    if response.status_code == 200:
        data = response.json()
    else:
        print(f"Error: {response.status_code}")

    return JsonResponse(data)
```

EONET:

```
@app.get("/EONET")
async def get_data_eonet():
    url = "https://eonet.gsfc.nasa.gov/api/v3/events/geojson"
    response = requests.get(url)
    if response.status_code == 200:
        data = response.json()
    else:
        print(f"Error: {response.status_code}")

    return JsonResponse(data)
```

Mars Rover Photo:

```
@app.get("/Mars-Rover-Photos")
async def get_data_mars_rover_photos():
    url = f"https://api.nasa.gov/mars-photos/api/v1/rovers/curiosity/photos?sol=1000&api_key={api_key}"
    response = requests.get(url)
    if response.status_code == 200:
        data = response.json()
    else:
        print(f"Error: {response.status_code}")

    return JsonResponse(data)
```

Do sử dụng project cung cấp API của Nasa nên không cần tới khóa API. Tuy nhiên vẫn cần tới đường dẫn do thư viện đó cung cấp. Tham khảo [tại đây](#).

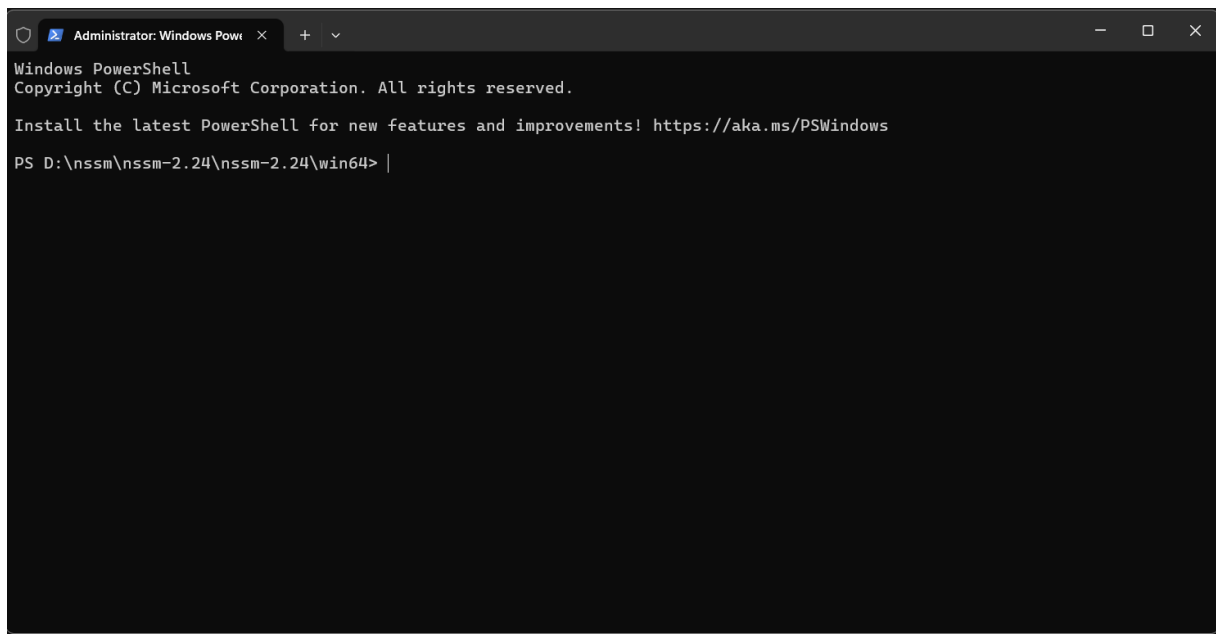
3.2.2. Cài đặt tự động chạy với nssm

Để dữ liệu luôn luôn tự động cập nhật và đưa lên các endpoint tạo ra từ FastAPI thì file main.py phải chạy liên tục. Do đó em sử dụng nssm để tạo một service run main.py mỗi khi máy tính mở lên.

Các bước thực hiện:

B1: Tải nssm tại <https://nssm.cc/download>

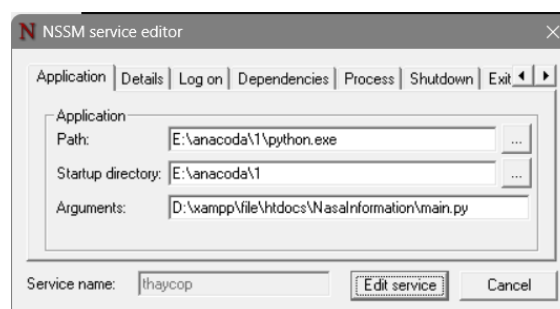
B2: Sử dụng PowerShell chạy với quyền admin trở tới thư mục nssm vừa tải



B3: Cài đặt các thông số trong giao diện

Sử dụng lệnh `./nssm install thaycop` để tạo 1 service

`./nssm edit thaycop` để mở giao diện



Trong đây sẽ cần đặt các thông số như sau:

Path: Đường dẫn tới trình thông dịch python (python.exe)

Startup directory: Đường dẫn đến thư mục chứa tệp (python.exe)

Arguments: Đường dẫn đến tệp main.py. (Tệp lấy dữ liệu và gửi lên endpoint của FastAPI)

B4: Khởi động service

./nssm start thaycop

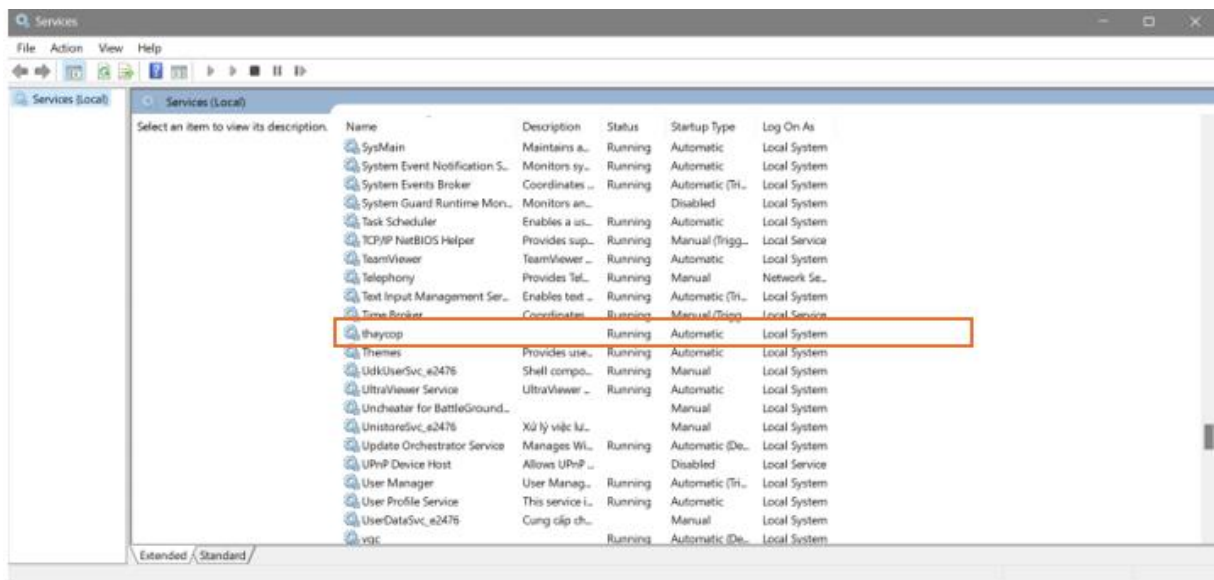
STOP: The operation completed successfully.
START: The operation completed successfully.

B5: Thực hiện check lại trong service.msc

Ấn tổ hợp phím Window + R để mở run

Nhập vào services.msc

Tìm tới service vừa tạo

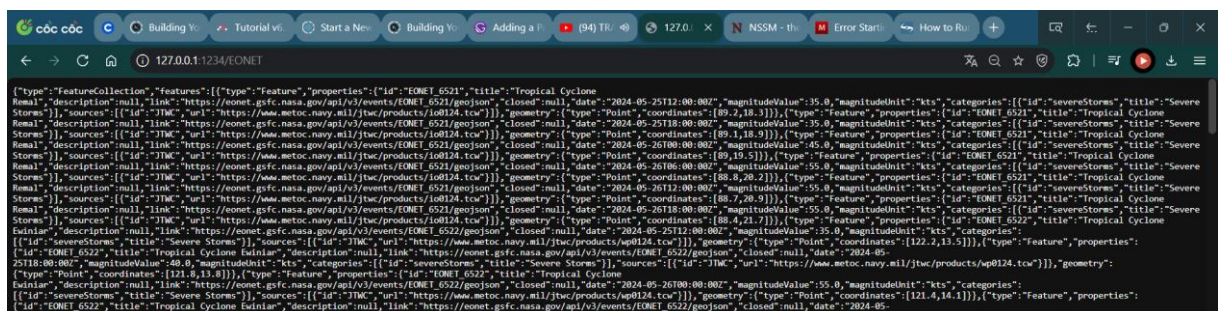


Nếu chưa thấy running thì khởi động thủ công và nếu lỗi thì có thể tham khảo:

<https://forum.manager.io/t/error-starting-manager-as-a-service-with-nssm/19703/13>

hoặc <https://chatgpt.com/share/9343c3eb-647b-4d7b-a4f3-c0e98b6586ff>

Kết quả:



3.3. Thiết kế Node-red và cơ sở dữ liệu

3.3.1. Thiết kế CSDL (dùng SQLSERVER)

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) được phát triển bởi Microsoft. Nó cung cấp một môi trường mạnh mẽ để quản lý và lưu trữ dữ liệu trong các ứng dụng và hệ thống doanh nghiệp. SQL Server hỗ trợ nhiều tính năng như lưu trữ dữ liệu, truy vấn dữ liệu, quản lý bảo mật, và tự động sao lưu dữ liệu. Nó cũng cung cấp khả năng mở rộng và tích hợp tốt với các ứng dụng và công nghệ khác của Microsoft.



Đối với 3 API lấy được từ nasa sẽ cần 3 bảng khác nhau để lưu trữ dữ liệu, do đó các bảng sẽ được thiết kế như sau:

Bảng APOD:

```
CREATE TABLE apod (  
    id INT IDENTITY(1,1) PRIMARY KEY, --Giá trị t? t?ng  
    copyright NVARCHAR(255),  
    date DATE,  
    explanation TEXT,  
    hdurl NVARCHAR(255),  
    media_type NVARCHAR(50),  
    service_version NVARCHAR(50),  
    title NVARCHAR(255),  
    url NVARCHAR(255),  
    --translate_explanation ntext  
);
```

Bảng EONET_Events:

```
CREATE TABLE EONET_Events (  
    uid int identity(1,1) primary key,  
    id NVARCHAR(50),  
    title NVARCHAR(255),  
    descriptions NVARCHAR(MAX),  
    link NVARCHAR(255),  
    closed NVARCHAR(50),  
    date_eonet DATE,  
    magnitudeValue FLOAT,  
    magnitudeUnit NVARCHAR(50),  
    urls NVARCHAR(255)  
);
```

Bảng MarsRoverPhotos:

```
CREATE TABLE MarsRoverPhotos (  
    id INT PRIMARY KEY,  
    sol INT,  
    full_name NVARCHAR(255),  
    img_src NVARCHAR(MAX),  
    total_photos INT,  
    earth_date DATE,  
    launch_date DATE,  
    landing_date DATE  
);
```

Tương ứng với 3 bảng cần có các Store Procedure liên quan:

SP GetLatestApodData:

```
CREATE PROCEDURE GetLatestApodData  
AS  
BEGIN  
    SELECT TOP 1 *  
    FROM apod  
    ORDER BY date DESC;  
END
```

SP GetYesterdayApodData:

```
CREATE PROCEDURE [dbo].[GetYesterdayApodData]  
AS  
BEGIN  
    SELECT TOP 1 *  
    FROM apod  
    WHERE date < CONVERT(date, GETDATE())  
    ORDER BY date DESC;  
END
```

SP GetNextApodData:

```
CREATE PROCEDURE [dbo].[GetNextApodData]
    @currentDate DATE
AS
BEGIN
    SELECT TOP 1 *
    FROM apod
    WHERE date > @currentDate
    ORDER BY date;
```

END

SP SelectAllMarsRoverPhotos:

```
CREATE PROCEDURE SelectAllMarsRoverPhotos
AS
BEGIN
    SET NOCOUNT ON;
    SELECT * FROM MarsRoverPhotos;
```

END;

GO

SP SelectEonetEvents:

```
create procedure SelectEonetEvents
    @id NVARCHAR(50),
    @title NVARCHAR(255),
    @description NVARCHAR(MAX),
    @link NVARCHAR(255),
    @closed NVARCHAR(50),
    @eventDate DATETIME,
    @magnitudeValue FLOAT,
    @magnitudeUnit NVARCHAR(50),
    @longitude FLOAT,
    @latitude FLOAT
```

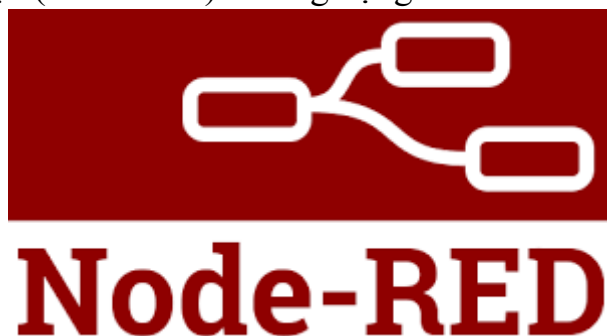
AS

```
BEGIN
    SELECT * FROM EONET_Events
    ORDER BY date_eonet DESC;
```

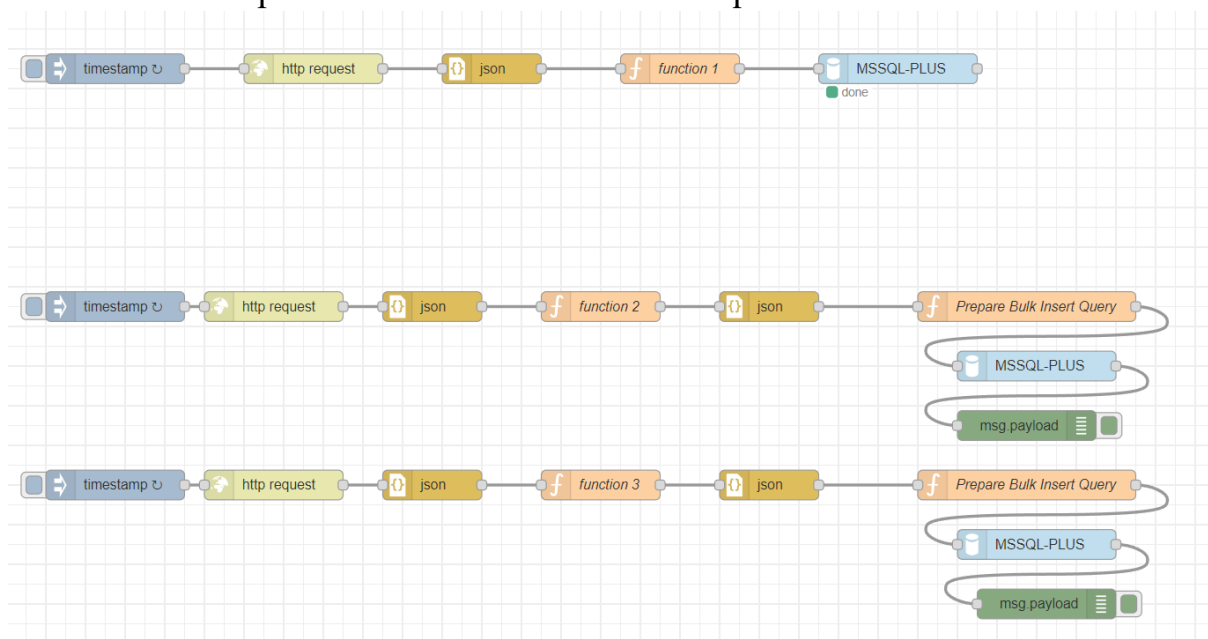
END

3.3.2. Thiết kế NodeRed

Node-RED là một công cụ mã nguồn mở và trực quan được sử dụng để xây dựng các luồng làm việc (workflows) và ứng dụng Internet of Things (IoT)



Sau khi hiểu sơ qua về NodeRed ta bắt đầu vào quá trình thiết kế :



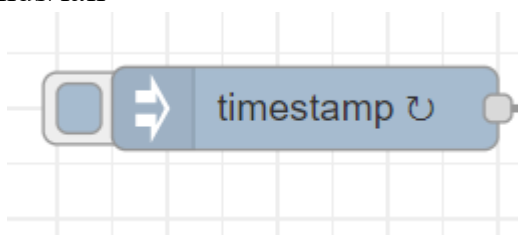
Như hình ở trên đây là giao diện chính của NodeRed đồng thời là nơi ta sẽ thiết kế và cấu hình cho các node:

Node Inject: Node này sẽ Chèn một tin nhắn vào luồng một cách thủ công hoặc theo các khoảng thời gian đều đặn. Dữ liệu tin nhắn có thể là nhiều loại khác nhau, bao gồm chuỗi, đối tượng JavaScript hoặc thời gian hiện tại. Trong bài này em sẽ dùng chế độ Repeat: interval, cụ thể như sau:

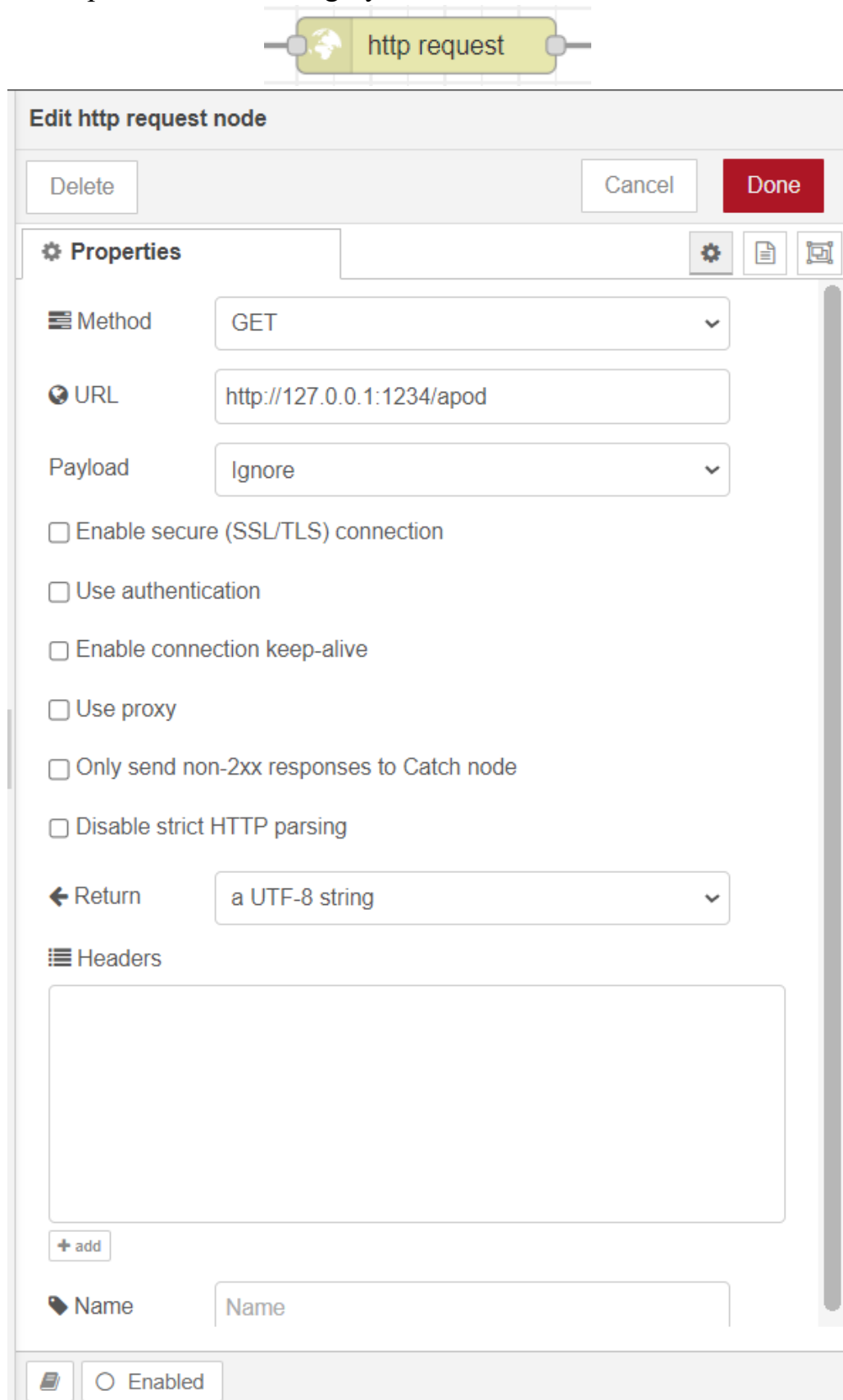
APOD: 24 hours/lần

EONET: 5 seconds/lần

MarsRoverPhotos: 5 seconds/lần



Node http request: cho phép bạn tương tác với các API web, tải xuống dữ liệu từ các trang web, hoặc gửi dữ liệu đến các dịch vụ web. Đây là node chính lấy dữ liệu từ Api mà ta đã tạo trong Python



The image shows a workflow editor interface. At the top, there is a yellow node labeled 'http request' with a globe icon. Below it is a dialog box titled 'Edit http request node'. The dialog has three buttons at the top: 'Delete', 'Cancel', and 'Done'. The 'Properties' tab is selected, showing various configuration options:

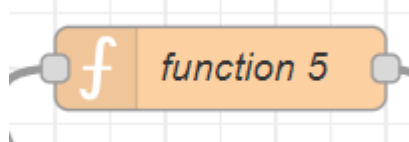
- Method:** A dropdown menu set to 'GET'.
- URL:** A text input field containing 'http://127.0.0.1:1234/apod'.
- Payload:** A dropdown menu set to 'Ignore'.
- ☐ Enable secure (SSL/TLS) connection
- ☐ Use authentication
- ☐ Enable connection keep-alive
- ☐ Use proxy
- ☐ Only send non-2xx responses to Catch node
- ☐ Disable strict HTTP parsing
- Return:** A dropdown menu set to 'a UTF-8 string'.
- Headers:** A large empty text area for adding headers, with a '+ add' button below it.
- Name:** A text input field containing 'Name'.

At the bottom of the dialog, there is a status bar with a document icon and a radio button labeled 'Enabled'.

Node Json : Chuyển đổi giữa Json và string



Node function: được sử dụng để viết các đoạn mã JavaScript tùy chỉnh để xử lý các tin nhắn (messages).



Trong node function này tùy loại API của nasa sẽ cần xử lý khác nhau
APOD:

```
1 // Lấy dữ liệu JSON từ đầu vào
2 var data = msg.payload;
3 // Trích xuất dữ liệu từ đối tượng JSON
4 var copyright = data.copyright;
5 var date = data.date;
6 var explanation = data.explanation;
7 var hdurl = data.hdurl;
8 var media_type = data.media_type;
9 var service_version = data.service_version;
10 var title = data.title;
11 var url = data.url;
12 var translate_explanation = data.translate_explanation;
13 // Lưu trữ dữ liệu vào một đối tượng để truyền vào node mssql
14 var sqlData = {
15     copyright: copyright,
16     date: date,
17     explanation: explanation,
18     hdurl: hdurl,
19     media_type: media_type,
20     service_version: service_version,
21     title: title,
22     url: url,
23     translate_explanation: translate_explanation
24 };
25 // Gửi dữ liệu vào output cho node mssql
26 msg.sqlData = sqlData;
27
28 return msg;
```

EONET:

Đối với EONET với dữ liệu từ các sự kiện trước và mới. API của nasa chỉ cung cấp 200 sự kiện gần đây nhất. Do đó không thể insert tất cả sự kiện theo cách gọi SP thông thường, vì vậy cần phải dùng tới Bulk Insert.

Node function:

```
1 // Lấy dữ liệu từ msg.payload
2 var events = msg.payload.features;
3 var newData = [];
4
5 // Chuyển đổi dữ liệu thành định dạng phù hợp cho endpoint
6 events.forEach(function (event) {
7     var eventDate = new Date(event.properties.date);
8     if (eventDate.getFullYear() === 2024) {
9         var eventData = {
10             id: event.properties.id,
11             title: event.properties.title,
12             date: event.properties.date,
13             descriptions: event.properties.descriptions,
14             link: event.properties.link,
15             closed: event.properties.closed,
16             magnitudeValue: event.properties.magnitudeValue,
17             magnitudeUnit: event.properties.magnitudeUnit,
18             url: (event.properties.sources && event.properties
19         };
20         newData.push(eventData);
21     }
22 });
23
24 // Gửi dữ liệu đã được xử lý qua output
25 msg.payload = newData;
26 return msg;
27
```

Node function Bulk Insert:

```
1 var data = msg.payload; // Giả sử msg.payload là mảng các đối tượng JSON
2 var bulkInsertValues = [];
3
4 data.forEach(function(event) {
5     var values = `('${event.id}',
6     '${event.title}', '${event.date}',
7     '${event.link}',
8     ${event.closed !== null ? `'${event.closed}'` : 'NULL'},
9     ${event.magnitudeValue !== null ? event.magnitudeValue : 'NULL'},
10    ${event.magnitudeUnit !== null ? `'${event.magnitudeUnit}'` : 'NULL'},
11    '${event.url}')`;
12    bulkInsertValues.push(values);
13 });
14
15 var bulkInsertQuery = `INSERT INTO EONET_Events
16 (id, title, date_eonet, link, closed, magnitudeValue, magnitudeUnit, urls)
17 VALUES ${bulkInsertValues.join(", ")}`;
18
19 msg.payload = bulkInsertQuery;
20 return msg;
```

Mars Rover Photos:

Tương tự EONET thì Mars Rover Photos cũng có các dữ liệu cũ và mới và rất nhiều object trong chuỗi json, vì vậy cũng dùng Bulk Insert để lưu vào cơ sở dữ liệu.

```
1  if (msg.payload && msg.payload.photos) {
2      var photos = msg.payload.photos;
3      var filteredData = [];
4      // Lặp qua mảng photos
5      photos.forEach(function (photo) {
6          // Kiểm tra ngày của ảnh
7          var eventDate = new Date(photo.earth_date);
8          if (eventDate.getFullYear() === 2015) {
9              // Nếu ảnh được chụp vào năm 2015, thêm vào mảng filteredData
10             var eventData = {
11                 id: photo.id,
12                 sol: photo.sol,
13                 full_name: photo.camera.full_name,
14                 img_src: photo.img_src,
15                 total_photos: photo.rover.total_photos,
16                 earth_date: photo.earth_date,
17                 launch_date: photo.rover.launch_date,
18                 landing_date: photo.rover.landing_date
19             };
20             filteredData.push(eventData);
21         }
22     });
23     msg.payload = filteredData;
24 } else {
25     console.error("Không có dữ liệu photos trong msg.payload");
26 }
27 return msg;
28
```

Node Bulk Insert:

```
1  var data = msg.payload; // Giả sử msg.payload là mảng các đối tượng JSON
2  var bulkInsertValues = [];
3
4  data.forEach(function (event) {
5      var values = `('${event.id}',
6                  '${event.sol}', '${event.full_name}',
7                  '${event.img_src}', ${event.total_photos},
8                  '${event.earth_date}', '${event.launch_date}',
9                  '${event.landing_date}')`;
10     bulkInsertValues.push(values);
11 });
12
13 var bulkInsertQuery = `INSERT INTO MarsRoverPhotos
14 (id, sol, full_name, img_src, total_photos, earth_date, launch_date, landing_date)
15 VALUES ${bulkInsertValues.join(", ")}`;
16
17 msg.payload = bulkInsertQuery;
18 return msg;
19
```

Node MSSQL-PLUS : đây là node dùng để liên kết với SQL SEVER.
Trong node này sẽ cần tùy chỉnh các thông tin để kết nối với cơ sở dữ liệu và thực hiện các truy vấn hoặc gọi tới Store Procedure

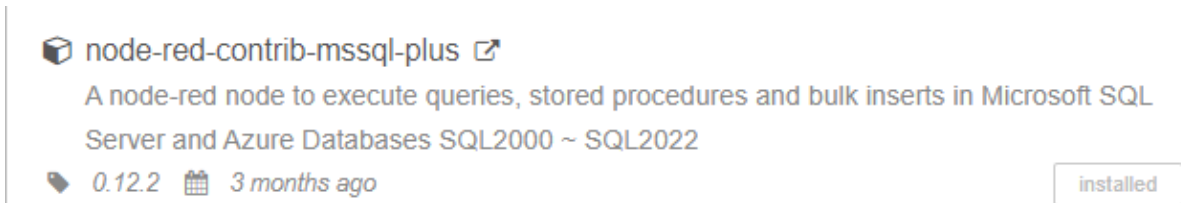
Đối với node này cần cài đặt thêm trong node-red palette

Các bước thực hiện:

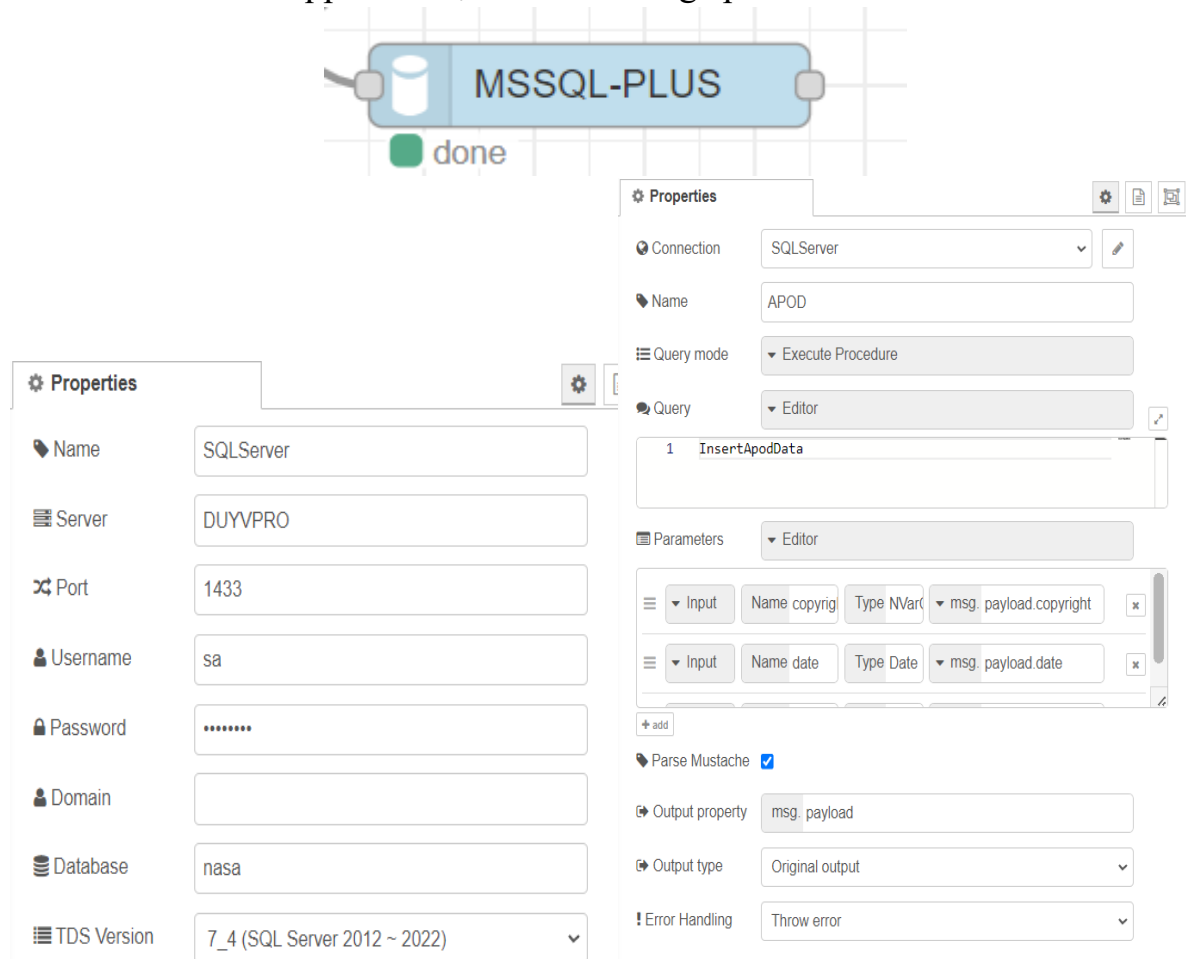
B1: ấn tổ hợp Alt + Shift + p

B2: Chuyển sang tab install

B3: Tìm như hình sau:



Cấu hình node mssqlplus ví dụ cho chức năng apod:



3.4. Thiết kế giao diện

Trong đề tài này em sẽ sử dụng ngôn ngữ PHP để gọi tới các SP đã được tạo ở trên và sử dụng PHP, HTML 5, CSS 3, Javascript để tạo giao diện có thể hiển thị ảnh và nhúng đường dẫn video trực tiếp.

Code JS giao tiếp với API PHP (Ví dụ chức năng apod):

```
function apod_load() {
    var next_apod = document.getElementById('next_apod');
    next_apod.style.display = "none";
    $.ajax({
        type: "GET",
        url: "api/apiAPOD_getlastest.php",
        dataType: "json",
        success: function (response) {
            if (response && response[0]) {
                var author = response[0].copyright;
                var date = response[0].date.date.split(' ')[0];
                var title = response[0].title;
                var explanation = response[0].explanation;
                var imageUrl = response[0].hdurl;
                var videoUrl = response[0].url;

                $("#author").text(author);
                $("#date_post").text(date);
                $("#title_post").text(title);
                $("#explanation").text(explanation);
                var iframe = document.querySelector('#video_apod iframe');
                iframe.setAttribute('src', videoUrl);
                if (imageUrl !== null) {
                    var element = document.getElementById('video_apod');
                    element.style.display = "none";
                    $("#img_apod img").attr("src", imageUrl);
                } else {
                    var element = document.getElementById('video_apod');
                    element.style.display = "block";
                    $("#img_apod img").css("display", "none");
                }
            } else {
                console.error("Dữ liệu trả về từ API không hợp lệ hoặc không tồn tại.");
            }
        },
        error: function (xhr, status, error) {
            console.error("Lỗi khi gửi yêu cầu AJAX: " + error);
            console.log("Dữ liệu JSON trả về từ máy chủ: ", xhr.responseText);
        }
    });
}
```

Ngày hôm qua **Bức ảnh thiên văn trong ngày** Ngày kế tiếp



Chirag Upreti

2024-05-18

North Celestial Aurora

Graceful star trail arcs reflect planet Earth's daily rotation in this colorful night skyscape. To create the timelapse composite, on May 12 consecutive exposures were recorded with a camera fixed to a tripod on the shores of the Ashokan Reservoir, in the Catskills region of New York, USA. North star Polaris is near the center of the star trail arcs. The broad trail of a waxing crescent Moon is on the left, casting a strong reflection across the reservoir waters. With intense solar activity driving recent geomagnetic storms, the colorful aurora borealis or northern lights, rare to the region, shine under Polaris and the north celestial pole. AuroraSaurus: Report your aurora observations

Ngày hôm qua **Bức ảnh thiên văn trong ngày** Ngày kế tiếp



2024-05-19

Jupiter Diving

Take this simulated plunge and dive into the upper atmosphere of Jupiter, the Solar System's ruling gas giant. The awesome animation is based on image data from JunoCam, and the microwave radiometer on board the Jupiter-orbiting Juno spacecraft. Your view will start about 3,000 kilometers above the southern Jovian cloud tops, and you can track your progress on the display at the left. As altitude decreases, temperature increases while you dive deeper at the location of Jupiter's famous Great Red Spot. In fact, Juno data indicates the Great Red Spot, the Solar System's largest storm system, penetrates some 300 kilometers into the giant planet's atmosphere. For comparison, the deepest point for planet Earth's oceans is just under 11 kilometers down. Don't worry though, you'll fly back out again. Dive into the Universe: Random APOD Generator

3.5. Back-end

3.5.1. Nền tảng PHP

Trong bài tập lớn này em sử dụng php làm API để giao tiếp giữa Database và phần giao diện.

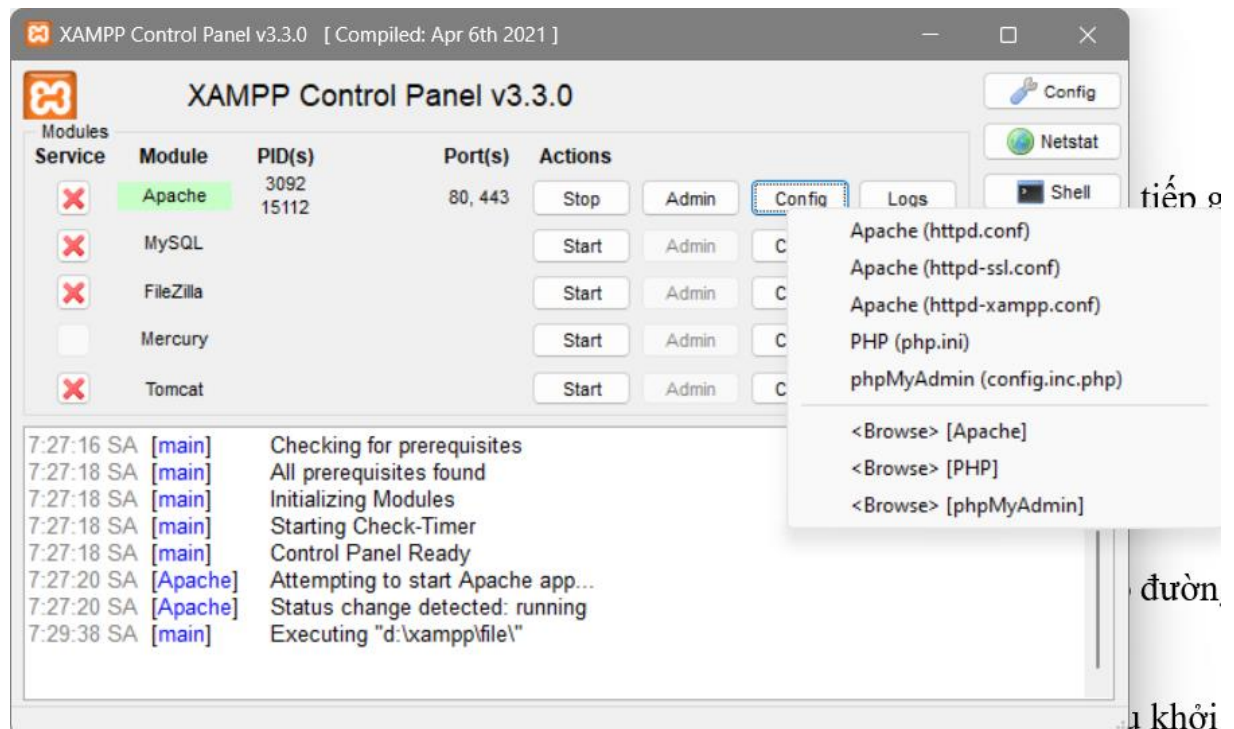
Để có thể sử dụng PHP cần cài đặt Xampp và cấu hình để kết nối với SQL Server.

Các bước cấu hình:

B1: Tải Driver PHP cho SQL Server [tại đây](#)

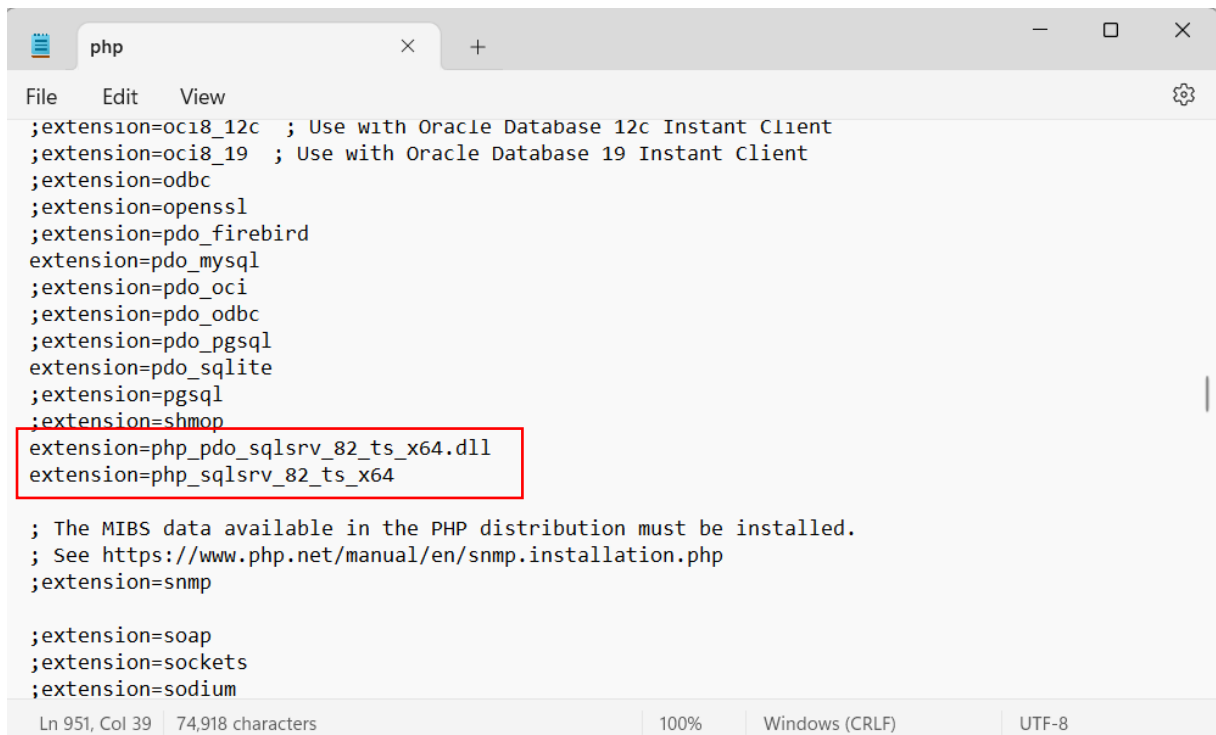
B2: Chuyển tệp tin vừa tải php_pdo_sqlsrv_82_ts_x64.dll vào đường dẫn sau của Xampp: D:\xampp\file\php\ext

B3: Thực hiện thêm config trong Xampp bằng cách chọn vào PHP



B4: Thay đổi nội dung trong txt như sau:

Thêm 2 dòng khoanh đỏ trong hình dưới



```
File Edit View
;extension=oci8_12c ; Use with Oracle Database 12c Instant Client
;extension=oci8_19 ; Use with Oracle Database 19 Instant Client
;extension=odbc
;extension=openssl
;extension=pdo_firebird
extension=pdo_mysql
;extension=pdo_oci
;extension=pdo_odbc
;extension=pdo_pgsql
extension=pdo_sqlite
;extension=pgsql
;extension=shmop
extension=php_pdo_sqlsrv_82_ts_x64.dll
extension=php_sqlsrv_82_ts_x64

; The MIBS data available in the PHP distribution must be installed.
; See https://www.php.net/manual/en/snmp.installation.php
;extension=snmp

;extension=soap
;extension=sockets
;extension=sodium
Ln 951, Col 39 74,918 characters 100% Windows (CRLF) UTF-8
```

B5: Lưu lại và khởi động Module Apache

Sau khi cài đặt xong sẽ có thể kết nối với SQL Server như sau: (Ví dụ chức năng apod)

```
<?php
$serverName = "DUYVPRO";
$connectionOptions = array(
    "Database" => "nasa",
    "Uid" => "sa",
    "PWD" => "makaeenm1"
);
$conn = sqlsrv_connect($serverName, $connectionOptions);
// Check connect
if (!$conn) {
    echo "Kết nối thất bại: " . sqlsrv_errors();
    exit;
}
$sql = "EXEC GetLatestApodData";
$stmt = sqlsrv_query($conn, $sql);
if ($stmt === false) {
    echo "Lỗi khi thực thi stored procedure: " . sqlsrv_errors();
    exit;
}
// Khởi tạo mảng để lưu trữ dữ liệu
$data = array();
// Lặp qua kết quả trả về từ stored procedure và thêm vào mảng
while ($row = sqlsrv_fetch_array($stmt, SQLSRV_FETCH_ASSOC)) {
    $data[] = $row;
}
```

```

}
// Đóng kết nối
sqlsrv_free_stmt($stmt);
sqlsrv_close($conn);
// Trả về dữ liệu dưới dạng JSON
header('Content-Type: application/json');
echo json_encode($data);
?>

```

Sau khi đã hoàn thành thiết lập, code API trên nền tảng PHP và tạo giao diện, bắt đầu khởi động chương trình.

Chạy node-red bằng terminal:

```

PS D:\xampp\file\htdocs\NasaInformation> node-red
25 May 21:11:42 - [info]

Welcome to Node-RED
=====

25 May 21:11:42 - [info] Node-RED version: v3.1.9
25 May 21:11:42 - [info] Node.js version: v22.1.0
25 May 21:11:42 - [info] Windows NT 10.0.22631 x64 LE
(node:6332) [DEP0059] DeprecationWarning: The `util.log` API is deprecated. Please use console.log() with a custom formatter or a third-party logger instead
.
(Use `node --trace-deprecation ...` to show where the warning was created)
25 May 21:11:42 - [info] Loading palette nodes
(node:6332) [DEP0040] DeprecationWarning: The `punycode` module is deprecated. Please use a userland alternative instead.
25 May 21:11:44 - [info] Settings file : C:\Users\Admin\.node-red\settings.js
25 May 21:11:44 - [info] Context store : 'default' [module=memory]
25 May 21:11:44 - [info] User directory : C:\Users\Admin\.node-red


```

Cuối cùng là khởi động XAMPP để có thể chạy mainpage.php trên trình duyệt.

<http://localhost/NasaInformation/mainpage.php#>

3.6. Kết quả


← → ↺ 📄 🌐 localhost/NasaInformation/mainpage.php 🔍 ⭐ 🛡️ 📱 📺 ⬇️ ☰

 APOD EONET Ảnh chụp bề mặt sao hỏa

Ngày hôm qua

Bức ảnh thiên văn trong ngày

Ngày kế tiếp



Amiel Contuliano

2024-05-27

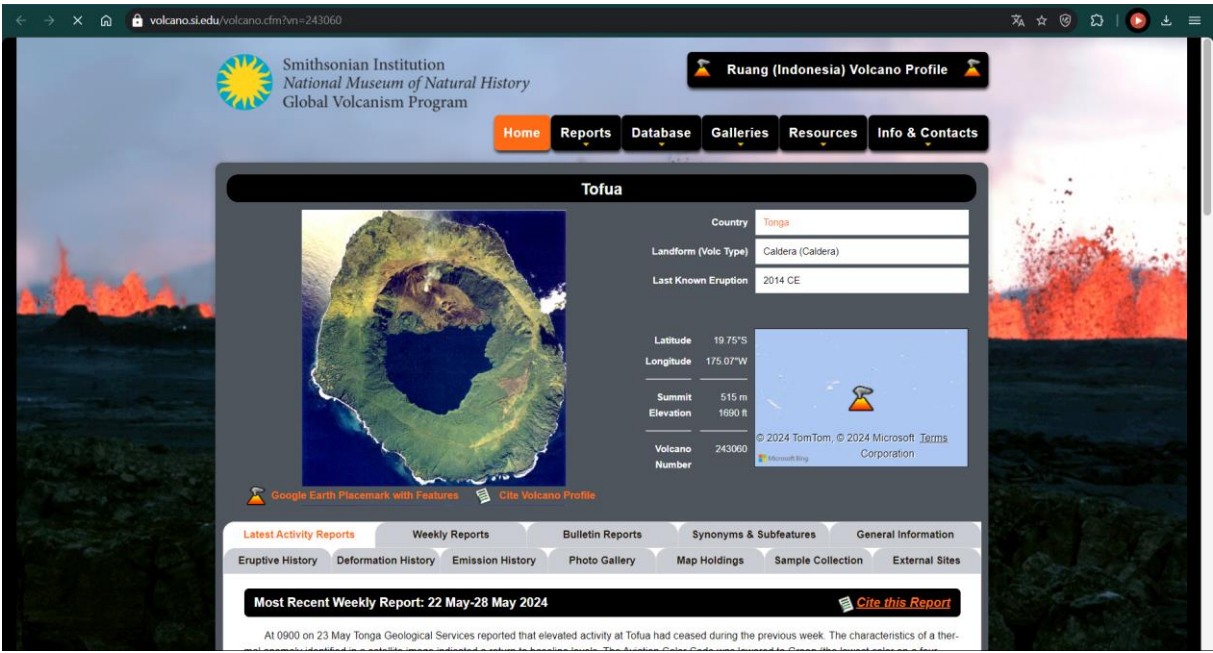
Chamaeleon I Molecular Cloud

Dark markings and bright nebulae in this telescopic southern sky view are telltale signs of young stars and active star formation. They lie a mere 650 light-years away, at the boundary of the local bubble and the Chamaeleon molecular cloud complex. Regions with young stars identified as dusty reflection nebulae from the 1946 Cederblad catalog include the C-shaped Ced 110 just above and right of center, and bluish Ced 111 below it. Also a standout in the frame, the orange tinted V-shape of the Chamaeleon Infrared Nebula (Cha IRN) was carved by material streaming from a newly formed low-mass star. The well-composed image spans 1.5 degrees. That's about 17 light-years at the estimated distance of the nearby Chamaeleon I molecular cloud.

Chức năng apod

EONET Events								
ID	Tiêu đề	Mô tả	Link dữ liệu	Closed	Ngày diễn ra	Giá trị độ lớn	Đơn vị độ lớn	Link tới sự kiện
EONET_6518	Tropical Cyclone Ialy	null	More info	null	2024-05-22	45	kts	More info
EONET_6518	Tropical Cyclone Ialy	null	More info	null	2024-05-22	35	kts	More info
EONET_6518	Tropical Cyclone Ialy	null	More info	null	2024-05-22	30	kts	More info
EONET_6518	Tropical Cyclone Ialy	null	More info	null	2024-05-22	45	kts	More info
EONET_6518	Tropical Cyclone Ialy	null	More info	null	2024-05-22	35	kts	More info
EONET_6518	Tropical Cyclone Ialy	null	More info	null	2024-05-22	30	kts	More info
EONET_6518	Tropical Cyclone Ialy	null	More info	null	2024-05-21	55	kts	More info
EONET_6518	Tropical Cyclone Ialy	null	More info	null	2024-05-21	60	kts	More info
EONET_6518	Tropical Cyclone Ialy	null	More info	null	2024-05-21	60	kts	More info
EONET_6518	Tropical Cyclone Ialy	null	More info	null	2024-05-21	60	kts	More info
EONET_6518	Tropical Cyclone Ialy	null	More info	null	2024-05-21	55	kts	More info
EONET_6518	Tropical Cyclone Ialy	null	More info	null	2024-05-21	60	kts	More info
EONET_6518	Tropical Cyclone Ialy	null	More info	null	2024-05-21	60	kts	More info
EONET_6518	Tropical Cyclone Ialy	null	More info	null	2024-05-20	55	kts	More info
EONET_6518	Tropical Cyclone Ialy	null	More info	null	2024-05-20	50	kts	More info
EONET_6518	Tropical Cyclone Ialy	null	More info	null	2024-05-20	50	kts	More info
localhost/NasaInformation/mainpage.php#	Ialy	null	More info	null	2024-05-20	55	kts	More info

Chức năng EONET



Link tới sự kiện trong EONET

Thông tin về ảnh chụp sao hỏa							
ID	Ngày ở trên sao hỏa	Tên đầy đủ	Ảnh chụp	Số ảnh đã chụp	Ngày theo Trái Đất	Ngày phóng	Ngày hạ cánh
102693	1000	Front Hazard Avoidance Camera		695670	2015-05-30	2011-11-26	2012-08-06
102694	1000	Front Hazard Avoidance Camera		695670	2015-05-30	2011-11-26	2012-08-06
102850	1000	Rear Hazard Avoidance Camera		695670	2015-05-30	2011-11-26	2012-08-06
102851	1000	Rear Hazard Avoidance Camera		695670	2015-05-30	2011-11-26	2012-08-06
103383	1000	Navigation Camera		695670	2015-05-30	2011-11-26	2012-08-06
103384	1000	Navigation Camera		695670	2015-05-30	2011-11-26	2012-08-06

Chức năng Mar Rover Photos

KẾT LUẬN

Kết luận, xây dựng một trang web thông tin về vũ trụ và trái đất không chỉ là một dự án thú vị mà còn là một cơ hội để khám phá vô số điều kỳ diệu của vũ trụ. Sự kết hợp của các nguồn thông tin như APOD, EONET và Mars Rover Photos không chỉ mở ra cánh cửa cho việc tìm hiểu sâu rộng về vũ trụ và hành tinh của chúng ta mà còn thúc đẩy sự hiểu biết và tò mò trong cộng đồng. Việc sử dụng API của NASA cung cấp cơ hội để khám phá và tận dụng nguồn tài nguyên vô tận từ không gian, đồng thời tạo ra những trải nghiệm trực quan và thú vị cho người dùng trên khắp thế giới. Đồng thời giúp cho bản thân em có thêm kiến thức về Python, thư viện FastAPI, Node-red và các kỹ năng tìm kiếm và sửa lỗi trong quá trình hoàn thành đề tài.