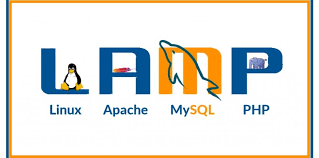
**Kiến Trúc LAMP: Nền Tảng Mạnh Mẽ Cho Phát Triển Ứng Dụng Web**

Kiến trúc LAMP là một mô hình phần mềm có uy tín, kết hợp bốn công nghệ hàng đầu để tạo ra môi trường phát triển web động và mạnh mẽ. Các thành phần chính của nó - Linux, Apache, MySQL và PHP - cùng nhau đóng vai trò quan trọng trong việc xây dựng, triển khai và duy trì các ứng dụng web hiện đại.



**Linux: Hệ Điều Hành Nền Tảng Mở:**

Linux, là hệ điều hành mã nguồn mở, chịu trách nhiệm về cấp độ đầu tiên trong ngăn xếp LAMP. Linh hoạt và ổn định, Linux cung cấp một môi trường chạy mạnh mẽ cho các ứng dụng web. Sự linh hoạt của nó cho phép người quản trị tinh chỉnh hệ thống để đáp ứng đòi hỏi cụ thể của ứng dụng.



**Apache: Máy Chủ Web Đa Nhiệm:**

Apache, là máy chủ web nguồn mở hàng đầu, đóng vai trò ở cấp độ thứ hai trong ngăn xếp. Với khả năng xử lý đồng thời nhiều yêu cầu, Apache là nơi các yêu cầu web đầu tiên được gửi đến. Nó quản lý việc tìm kiếm, lưu trữ, và truyền thông tin qua giao thức HTTP, cung cấp một cơ sở cho tất cả các ứng dụng web trong mô hình LAMP.

A screenshot of a computer

Description automatically generated

**MySQL: Hệ Quản Trị Cơ Sở Dữ Liệu Hiệu Quả:**

MySQL, là hệ thống quản lý cơ sở dữ liệu quan hệ, đảm bảo việc lưu trữ và truy xuất dữ liệu một cách hiệu quả. Là lớp thứ ba của ngăn xếp, MySQL chịu trách nhiệm về việc tạo và quản lý cơ sở dữ liệu. Dữ liệu này thường bao gồm thông tin về người dùng, sản phẩm, đơn hàng, và nhiều loại dữ liệu khác quan trọng cho ứng dụng web.



**PHP: Ngôn Ngữ Lập Trình Động:**

PHP, viết tắt của "PHP: Hypertext Preprocessor," là ngôn ngữ lập trình chủ yếu của ngăn xếp LAMP. Nó đóng vai trò là lớp thứ tư và cuối cùng, cho phép nhúng mã lệnh vào HTML để tạo ra các trang web động và tương tác. PHP kết nối các thành phần khác của LAMP và cung cấp khả năng xử lý động, giúp ứng dụng thích ứng và phản hồi nhanh chóng đối với yêu cầu người dùng.



**phpMyAdmin: Giao Diện Quản Trị MySQL Thân Thiện và Mạnh Mẽ**

**1. Giới Thiệu phpMyAdmin:**

phpMyAdmin là một ứng dụng web mã nguồn mở được thiết kế để quản lý và tương tác với hệ thống quản lý cơ sở dữ liệu MySQL thông qua giao diện người dùng web. Được viết bằng ngôn ngữ PHP, phpMyAdmin giúp người quản trị cơ sở dữ liệu và nhà phát triển tương tác với MySQL một cách dễ dàng và hiệu quả.



**2. Chức Năng Chính của phpMyAdmin**:

Quản lý Cơ Sở Dữ Liệu: phpMyAdmin cung cấp một giao diện thân thiện để tạo, sửa đổi và xóa cơ sở dữ liệu MySQL. Người dùng có thể thực hiện các thao tác quản lý cơ sở dữ liệu mà không cần phải sử dụng dòng lệnh SQL.

Bảng và Dữ Liệu: Nó cho phép quản lý bảng và dữ liệu bên trong chúng. Người dùng có thể thêm, sửa đổi, xóa dữ liệu một cách thuận tiện thông qua giao diện đồ họa.

Thực Hiện Các Truy Vấn SQL: phpMyAdmin cho phép người dùng thực hiện các truy vấn SQL trực tiếp từ giao diện người dùng. Điều này làm cho việc tương tác với cơ sở dữ liệu trở nên linh hoạt và dễ dàng.

Quản Lý Người Dùng và Quyền Truy Cập: Bạn có thể tạo và quản lý người dùng MySQL cũng như quyền truy cập của họ thông qua giao diện đồ họa của phpMyAdmin. Điều này giúp kiểm soát và bảo mật cơ sở dữ liệu một cách hiệu quả.

Xem và Tối Ưu Hóa Cơ Sở Dữ Liệu: phpMyAdmin cung cấp các công cụ để xem thông tin chi tiết về cơ sở dữ liệu, bao gồm kích thước, số liệu thống kê, và cung cấp cơ hội để tối ưu hóa cơ sở dữ liệu để hiệu suất tốt nhất.

**3. Ưu Điểm Của phpMyAdmin:**

Giao Diện Người Dùng Thân Thiện: Giao diện người dùng của phpMyAdmin được thiết kế đơn giản và thân thiện, giúp người dùng mới dễ dàng thích nghi và sử dụng.

Đa Ngôn Ngữ và Đa Ký Tự: Hỗ trợ nhiều ngôn ngữ và bảng mã ký tự, làm cho phpMyAdmin trở thành một công cụ linh hoạt và toàn diện cho cộng đồng quốc tế.

Mã Nguồn Mở: Được phát triển theo hình thức mã nguồn mở, phpMyAdmin cho phép cộng đồng người dùng đóng góp vào phát triển và cải thiện ứng dụng.

Hỗ Trợ Đa Nền Tảng: Có thể cài đặt phpMyAdmin trên nhiều nền tảng và hệ điều hành khác nhau, tăng tính linh hoạt và tính khả dụng.

**Sử dụng phpMyAdmin để quản lý cơ sở dữ liệu MySQL**

Khởi động WSL trên Windown.

Kiểm tra trạng thái của MySQL đã bật hay chưa.

Sử dụng lệnh **sudo service mysql status**

A computer screen with white text

Description automatically generated

Kiểm tra trạng thái của máy chủ Web Apache.

Sử dụng lệnh **sudo service apache2 status**

A black screen with white text

Description automatically generated

Sau khi máy cơ sở dữ liệu và máy chủ Web được bật thì ta có thể truy cập phpMyAdmin thông qua trình duyệt.

Truy cập phpMyAdmin qua Trình duyệt:

Mở trình duyệt web và nhập địa chỉ URL của phpMyAdmin. Thông thường, nó sẽ là http://localhost/phpmyadmin hoặc http://127.0.0.1/phpmyadmin. Nếu phpMyAdmin đã được cài đặt và máy chủ web đang chạy, bạn sẽ thấy giao diện đăng nhập.

A screenshot of a login box

Description automatically generated

Đăng nhập vào thì ta sẽ có giao diện như sau.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Ta tạo một database có tên là ESP\_Data bằng cách nhấn vào new và điền tên database vào ô **database name** sau đó nhấn **create**

Nhấn vào new trong database vừa tạo để tạo một bảng chứa các dữ liệu cần được tải lên từ thingspeak.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

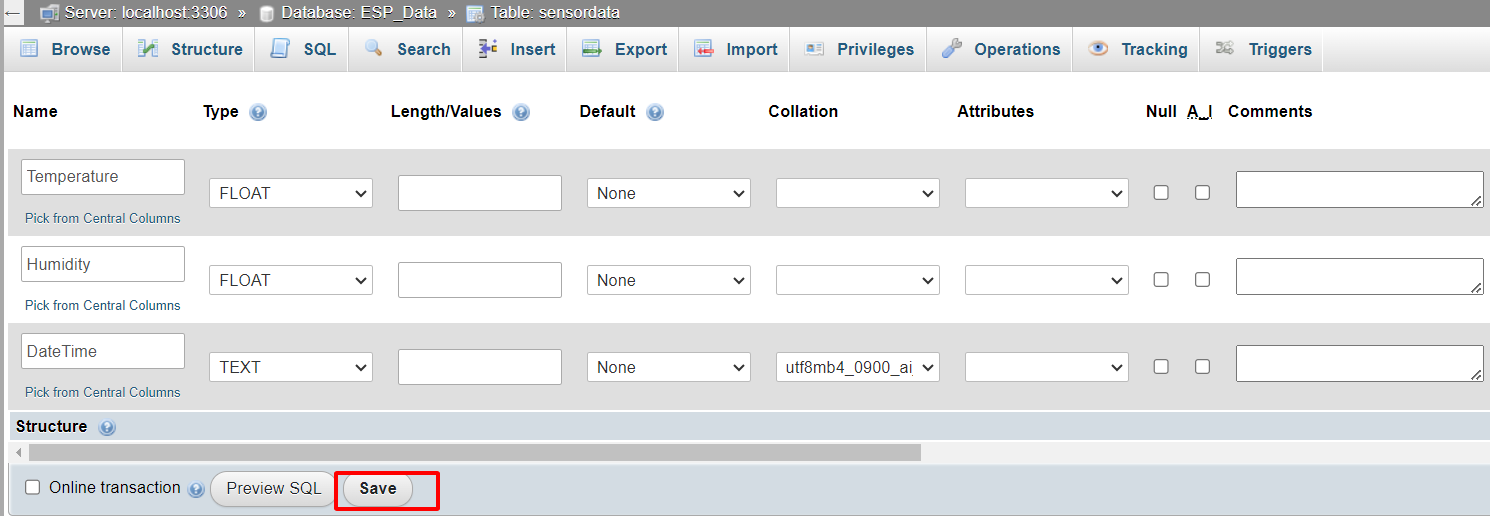
Ta sẽ tạo được bảng với các thuộc tính mà đề bài yêu cầu.

A screenshot of a computer

Description automatically generated

Vì giá trị của trường Humidity và Temperature có thể là số thập phân nên kiểu dữ liệu là type.

Trường Date Time có kiểu dữ liệu là text.



Nhấn save để tạo và lưu bảng dữ liệu.

A screenshot of a computer

Description automatically generatedbảng được tạo sẽ chứa các trường dữ liệu theo định dạng đã tạo trước đó.

Lập trình python trên vs code để lấy dữ liệu từ thingspeak để đưa lên MySQL.

Thêm các thư viện cần thiết vào chương trình.

import requests

import mysql.connector

from datetime import datetime

import time

import pytz

**Thư viện requests**

Chức năng: Thư viện requests được sử dụng để gửi HTTP requests. Trong mã có thể được sử dụng để tương tác với các API hoặc trang web từ xa để lấy dữ liệu.

**mysql.connector**

Chức năng: Thư viện mysql.connector cung cấp các công cụ để kết nối và tương tác với cơ sở dữ liệu MySQL từ Python. Trong mã có thể được sử dụng để thực hiện các truy vấn SQL và tương tác với cơ sở dữ liệu.

**datetime**

Chức năng: Thư viện datetime được sử dụng để làm việc với dữ liệu ngày giờ. Trong mã nó được sử dụng để tạo đối tượng ngày giờ và thực hiện các thao tác liên quan đến thời gian.  
**time**

Chức năng: Thư viện time cung cấp các hàm liên quan đến thời gian. Trong mã , nó được sử dụng để thực hiện các thao tác liên quan đến đợi một khoảng thời gian cụ thể.

**pytz**

Chức năng: Thư viện pytz giúp làm việc với múi giờ và thời gian có độ chính xác cao. Trong mã sử dụng để xác định múi giờ.

Tạo một biến có tên là channel\_url để lưu địa chỉ của kênh thingspeak cần lấy dữ liệu.

channel\_url = "https://api.thingspeak.com/channels/2258169/feeds.json"

tạo các biến để lưu trữ các thông tin liên quan đến việc kết nối với MySQL.

# Thông tin kết nối MySQL

host = "localhost"  # Địa chỉ máy chủ MySQL

user = "root"  # Tên người dùng MySQL

password = "thong2002"  # Mật khẩu MySQL

database = "ESP\_Data"  # Tên cơ sở dữ liệu MySQL

thiết lập khoảng thời gian giữa các lần cập nhật dữ liệu là 5 phút

# Thời gian giữa các lần lấy dữ liệu (ví dụ: 5 phút)

update\_interval = 300

 # Múi giờ của Thingspeak (UTC)

    thingspeak\_tz = pytz.timezone('UTC')

    # Múi giờ GMT+7

gmt7\_tz = pytz.timezone('Asia/Bangkok')

Hai đối tượng múi giờ này được sử dụng để chuyển đổi thời gian giữa các múi giờ khác nhau. thingspeak\_tz được sử dụng để xác định múi giờ của dữ liệu từ ThingSpeak, trong khi gmt7\_tz được sử dụng để chuyển đổi thời gian từ UTC sang múi giờ GMT+7.

# Lấy dữ liệu từ Thingspeak

            response = requests.get(channel\_url)

**requests.get** là một phương thức của thư viện requests được sử dụng để thực hiện yêu cầu HTTP GET.

**channel\_url** là địa chỉ URL của kênh ThingSpeak mà bạn muốn lấy dữ liệu.

Sau khi thực hiện yêu cầu trạng thái HTTP nếu đã thành công thì trạng thái sẽ là (response.status\_code == 200 thường là mã trạng thái OK).

if response.status\_code == 200:

                data = response.json()

                feeds = data.get('feeds', [])

**Kiểm Tra Mã Trạng Thái HTTP:**

**if response.status\_code == 200:**: Kiểm tra xem mã trạng thái HTTP của phản hồi có phải là 200 không. Mã trạng thái 200 thường đại diện cho một yêu cầu thành công.

**Chuyển Đổi Dữ Liệu JSON:**

**data = response.json():** Nếu mã trạng thái là 200, đoạn mã này chuyển đổi dữ liệu JSON từ phản hồi thành một đối tượng Python. Thông thường, dữ liệu JSON từ ThingSpeak chứa thông tin về các feeds.

**Truy Cập Feeds từ Dữ Liệu:**

**feeds = data.get('feeds', []):** Dùng phương thức get để truy cập giá trị của key 'feeds' từ đối tượng dữ liệu Python. Nếu không có key 'feeds', một danh sách trống [] sẽ được gán cho biến feeds.

  if not feeds:

                    print("Không có dữ liệu trong trường 'feeds'")

Kiểm tra feeds có phải là danh sách trống hay không nếu có thì sẽ in ra thông báo “ không có dữ liệu trong trường feeds”.

Nếu có dữ liệu thì sẽ bắt đầu kết nối với MySQL bằng các thông tin kết nối đã lưu vào các biến trước đó.

# Kết nối đến cơ sở dữ liệu MySQL

                  conn = mysql.connector.connect(host=host, user=user, password=password, database=database)

                    cursor = conn.cursor()

Sau khi kết nối bắt đầu duyệt qua các phần tử trên feeds để lấy các dữ liệuc cần thiết.

feed in feeds:

                        Temperature = feed.get('field1')

                        Humidity = feed.get('field2')

                        created\_at = feed.get('created\_at')  # Lấy thời gian từ Thingspeak

**Temperature = feed.get('field1'):** Lấy giá trị của trường 'field1' từ feed hiện tại và gán cho biến Temperature. Trường 'field1' chứa thông tin về nhiệt độ.

**Humidity = feed.get('field2'):** Tương tự, lấy giá trị của trường 'field2' từ feed hiện tại và gán cho biến Humidity. Trường 'field2' chứa thông tin về độ ẩm.

**created\_at = feed.get('created\_at'):** Lấy giá trị của trường 'created\_at' từ feed hiện tại và gán cho biến created\_at. Trường này chứa thông tin về thời điểm feed được tạo ra trên ThingSpeak.

  if Temperature is not None and Humidity is not None and created\_at is not None:

       # Chuyển đổi thời gian từ UTC sang GMT+7

   datetime\_created\_utc = datetime.strptime(created\_at, "%Y-%m-%dT%H:%M:%SZ")

   datetime\_created\_utc = thingspeak\_tz.localize(datetime\_created\_utc)

   datetime\_created\_gmt7 = datetime\_created\_utc.astimezone(gmt7\_tz)

kiểm tra xem các dữ liệu có khác None hay không.nếu không thì

**datetime\_created\_utc = datetime.strptime(created\_at, "%Y-%m-%dT%H:%M:%SZ"):** chuyển đổi thời gian tạo ra từ định dạng chuỗi sang đối tượng datetime trong múi giờ UTC. Hàm strptime được sử dụng để phân tích chuỗi thời gian created\_at với định dạng "%Y-%m-%dT%H:%M:%SZ".

**datetime\_created\_utc = thingspeak\_tz.localize(datetime\_created\_utc):** sử dụng đối tượng múi giờ thingspeak\_tz để chuyển đối datetime\_created\_utc từ múi giờ không có điều chỉnh (UTC) sang múi giờ UTC.

**datetime\_created\_gmt7 = datetime\_created\_utc.astimezone(gmt7\_tz):** chuyển đổi datetime\_created\_utc từ múi giờ UTC sang múi giờ GMT+7 (gmt7\_tz), để có thời gian theo múi giờ địa phương.

 # Thực hiện truy vấn để chèn dữ liệu vào cơ sở dữ liệu

                        insert\_query = "INSERT INTO sensordata (Temperature, Humidity, DateTime) VALUES (%s, %s, %s)"

                        cursor.execute(insert\_query, (Temperature, Humidity, formatted\_datetime))

                        conn.commit()

**insert\_query = "INSERT INTO sensordata (Temperature, Humidity, DateTime) VALUES (%s, %s, %s)":** Đây là câu truy vấn SQL INSERT được định nghĩa để chèn dữ liệu vào bảng sensordata. Câu truy vấn này có ba trường là Temperature, Humidity, và DateTime, và các giá trị sẽ được đặt vào bằng cách sử dụng các tham số %s.

**cursor.execute(insert\_query, (Temperature, Humidity, formatted\_datetime)):** thực hiện truy vấn SQL với các giá trị cụ thể của Temperature, Humidity, và formatted\_datetime. Hàm execute của đối tượng cursor được sử dụng để thực hiện câu truy vấn.

**conn.commit():**thực hiện lệnh commit trên đối tượng kết nối conn. Lệnh commit là cần thiết để xác nhận rằng các thay đổi (trong trường hợp này là việc chèn dữ liệu) được lưu trữ vào cơ sở dữ liệu MySQL.

Bằng cách này, sau mỗi vòng lặp trong quá trình lấy dữ liệu từ ThingSpeak, dữ liệu nhiệt độ, độ ẩm và thời gian sẽ được chèn vào cơ sở dữ liệu MySQL thông qua câu truy vấn INSERT.

# Đóng kết nối với cơ sở dữ liệu MySQL

                cursor.close()

                conn.close()

sau khi truy vấn xong thì đóng kết nối với MySQL.

Nếu trong quá trình thực hiện có lỗi thì sẽ được thông báo các lỗi tương ứng.

  print("Lỗi trong việc lấy dữ liệu từ Thingspeak")

    except requests.exceptions.RequestException as e:

        print("Lỗi kết nối Thingspeak: ", e)

    except mysql.connector.Error as err:

        print("Lỗi kết nối MySQL: ", err)

cập nhật dữ liệu sau mỗi 5 phút.

 # Ngủ một khoảng thời gian trước khi lấy dữ liệu tiếp theo

    time.sleep(update\_interval)

Sau khi chạy chương trình thì thu được kết quả là. Dữ liệu nhiệt độ và độ ẩm được đọc.

A screenshot of a computer screen

Description automatically generated

Kiểm tra xem dữ liệu đã được lưu lên MySQL hay chưa.

A screenshot of a table

Description automatically generated

Như vậy thông tin về về nhiệt độ độ ẩm và thời gian cập nhật của nhiệt độ đã được lưu trữ vào các trường của MySQL.

Tóm lại việc lấy dữ liệu từ Thingspeak và lưu trữ trên MySQL đã thành công. Việc thu thập dữ liệu và lưu trữ dữ liệu này có ý nghĩa hết sức quan trọng trong việc giám sát sự biến động của môi trường từ đó có các phương án phù hợp để đưa ra các giải pháp tối ưu cho các hoạt động của các dự án khác có liên quan.