D:\Internship week 02\phishing_simulator.py

```python
import smtplib
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
from flask import Flask, request, render_template
import mysql.connector
import datetime

# MySQL Connection
DB_CONFIG = {
    "host": "localhost",      #Enter the mysql host,user,password
    "user": "root",
    "password": "Pv9819733054@",
    "database": "phishing_simulator"
}

def get_db_connection():
    return mysql.connector.connect(**DB_CONFIG)

# Flask App Setup
app = Flask(__name__)

# Email Sender Configuration
SMTP_SERVER = "smtp.gmail.com"  # Corrected SMTP server
SMTP_PORT = 587
EMAIL_ADDRESS = "prathmeshvarma2003@gmail.com"   #Email address to send email
EMAIL_PASSWORD = "gwda iugz yuzf avnj"

# Function to Send Phishing Emails
def send_phishing_email(target_email, phishing_link):
    try:
        subject = "Important Update - Action Required"
        body = f"Dear User,\n\nPlease click the link below to verify your account:\n{phishing_link}\n\nRegards,\nSupport Team"

        message = MIMEMultipart()
        message["From"] = EMAIL_ADDRESS
        message["To"] = target_email
        message["Subject"] = subject
        message.attach(MIMEText(body, "plain"))

        with smtplib.SMTP(SMTP_SERVER, SMTP_PORT) as server:
            server.starttls()
            server.login(EMAIL_ADDRESS, EMAIL_PASSWORD)
            server.sendmail(EMAIL_ADDRESS, target_email, message.as_string())

        log_email_sent(target_email)
        print(f"Email sent to {target_email}")
    except Exception as e:
        print(f"Failed to send email to {target_email}: {e}")

# Log Email Sent to Database
def log_email_sent(email):
```

```
    conn = get_db_connection()
    cursor = conn.cursor()
    query = "INSERT INTO email_logs (email, timestamp) VALUES (%s, %s)"
    cursor.execute(query, (email, datetime.datetime.now()))
    conn.commit()
    cursor.close()
    conn.close()

# Route: Default Home Page
@app.route("/")
def home():
    return """
    <h1>Welcome to the Phishing Simulator</h1>
    <p>Please visit the phishing page <a href='/phishing'>here</a>.</p>
    """

# Route: Phishing Page
@app.route("/phishing", methods=["GET", "POST"])
def phishing_page():
    if request.method == "POST":
        username = request.form.get("username")
        password = request.form.get("password")
        log_user_interaction(username, password)
        return render_template("feedback.html")

    return render_template("phishing.html")

# Log User Interaction to Database
def log_user_interaction(username, password):
    conn = None
    cursor = None
    try:
        print(f"Logging interaction - Username: {username}, Password: {password}")
        conn = get_db_connection()
        cursor = conn.cursor()
        query = "INSERT INTO user_interactions (username, password, timestamp) VALUES (%s,
%s, %s)"
        cursor.execute(query, (username, password, datetime.datetime.now()))
        conn.commit()
    except Exception as e:
        print(f"Error logging interaction: {e}")
    finally:
        if cursor:
            cursor.close()
        if conn:
            conn.close()

# Route: Feedback Page
@app.route("/feedback")
def feedback_page():
    return """
    <h1>Security Awareness Feedback</h1>
    <p>Thank you for participating in this exercise. Here are some tips to identify phishing
emails:</p>
```

```
    <ul>
        <li>Check the sender's email address carefully.</li>
        <li>Hover over links to see the actual URL before clicking.</li>
        <li>Look for spelling and grammatical errors in the email.</li>
        <li>Be cautious with emails creating a sense of urgency.</li>
    </ul>
    <p>Stay safe online!</p>
    """

# Database Initialization (Run Once)
def initialize_database():
    conn = get_db_connection()
    cursor = conn.cursor()

    # Create Tables
    cursor.execute("""
    CREATE TABLE IF NOT EXISTS email_logs (
        id INT AUTO_INCREMENT PRIMARY KEY,
        email VARCHAR(255) NOT NULL,
        timestamp DATETIME NOT NULL
    )
    """)

    cursor.execute("""
    CREATE TABLE IF NOT EXISTS user_interactions (
        id INT AUTO_INCREMENT PRIMARY KEY,
        username VARCHAR(255),
        password VARCHAR(255),
        timestamp DATETIME NOT NULL
    )
    """)

    conn.commit()
    cursor.close()
    conn.close()

if __name__ == "__main__":
    # Uncomment the line below to initialize the database
    # initialize_database()

    # Specify the target email and phishing link
    target_email = "prathmeshvarma50@gmail.com"  # Replace with the target's email address
    phishing_link = "http://127.0.0.1:5000/phishing"  # Replace with your phishing page URL

    # Send phishing email
    send_phishing_email(target_email, phishing_link)

    # Start Flask Server
    app.run(debug=True)

# Templates (phishing.html)
# Save this as phishing.html in a templates/ folder
"""
<!DOCTYPE html>
```

```
<html>
<head>
    <title>Login</title>
</head>
<body>
    <h1>Login</h1>
    <form method="POST">
        <label for="username">Username:</label>
        <input type="text" id="username" name="username" required><br>
        <label for="password">Password:</label>
        <input type="password" id="password" name="password" required><br>
        <button type="submit">Login</button>
    </form>
</body>
</html>
"""

```