import random

import smtplib

#(1) Function to generate a 6-digit OTP randomly

def OTP\_generation():

    OTP = random.randint(100000, 999999)  # Generate a random 6-digit OTP

    return OTP

#(2) Function to simulate sending the OTP to the user's email address

def send\_otp\_email(receiver\_email, OTP):

    # Setting up the SMTP server

    server = smtplib.SMTP('smtp.gmail.com', 587)

    server.starttls()

    # Login to the server (Use an app password if you have 2FA enabled)

    password = ""  # Senders credentials

    server.login("", password)

    # Creating the email content

    body = f"Your OTP is {OTP}."

    subject = "OTP Verification"

    message = f"Subject: {subject}\n\n{body}"

    # Send the OTP to the user's email address

    server.sendmail("Senders\_email", receiver\_email, message)

    print(f"OTP has been sent to {receiver\_email}")

    server.quit()

#(3) Function to prompt the user to enter the OTP received in their email

def prompt\_otp\_entry():

    try:

        received\_OTP = int(input("Enter the OTP you received: "))

        return received\_OTP

    except ValueError:

        print("Invalid input! Please enter a numeric value.")

        return prompt\_otp\_entry()  # Retry if invalid input

#(4) Function to verify if the entered OTP matches the generated OTP

def verify\_otp(received\_OTP, generated\_OTP):

    if received\_OTP == generated\_OTP:

        return True

    else:

        return False

# Function to verify if the email is valid

def email\_verification(receiver\_email):

    email1 = ["gmail", "hotmail", "yahoo", "outlook"]

    email2 = [".com", ".in", ".org", ".edu", ".co.in"]

    count = 0

    for x in email1:

        if x in receiver\_email:

            count += 1

    for y in email2:

        if y in receiver\_email:

            count += 1

    if "@" not in receiver\_email or count != 2:

        print("Invalid email ID")

        new\_receiver\_email = input("Enter correct email ID: ")

        return email\_verification(new\_receiver\_email)

    return receiver\_email

# Main function to manage OTP generation, sending, and verification

def otp\_verification\_system():

    receiver\_email = input("Enter your email address: ")

    # Validate the email address

    valid\_receiver\_email = email\_verification(receiver\_email)

    # Generate OTP

    generated\_OTP = OTP\_generation()

    # Send OTP to the user's email

    send\_otp\_email(valid\_receiver\_email, generated\_OTP)

    # Allow the user to enter OTP up to 3 times if incorrect

    attempts = 3

    while attempts > 0:

        received\_OTP = prompt\_otp\_entry()  # Get OTP entered by the user

        # Verify OTP

        if verify\_otp(received\_OTP, generated\_OTP):

            print("OTP verified successfully!")

            break

        else:

            attempts -= 1

            if attempts > 0:

                print(f"Incorrect OTP. You have {attempts} attempts left.")

            else:

                print("Incorrect OTP. You've used all attempts.")

                retry = input("Would you like to receive a new OTP? (yes/no): ").lower()

                if retry == "yes":

                    otp\_verification\_system()  # Restart the OTP verification process

                else:

                    print("OTP verification failed. Exiting system.")

                    break

# Run the OTP verification system

otp\_verification\_system()

