



Python - Capstone Project

OTP
Verification
System

SURYAKANT RANJAN

Problem statement:

You are tasked with developing an OTP (One-Time Password) verification system in Python. The system should generate a 6-digit OTP and send it to the user's email address for verification. Upon receiving the OTP, the user should enter it into the system for validation. If the entered OTP matches the generated OTP, access should be granted; otherwise, access should be denied.

Project requirements:

- Implement a function to generate a 6-digit OTP randomly.
- Develop a function to simulate sending the OTP to the user's email address.
- Create a function to prompt the user to enter the OTP received in their email.
- Implement a function to verify if the entered OTP matches the generated OTP.
- Ensure proper error handling and user-friendly prompts throughout the system.
- Allow the user to retry OTP entry in case of incorrect input.

Project deliverables :

- Python script containing the implementation of the OTP verification system.
- Documentation explaining the functionality of each function, how to run the program, and any dependencies required.
- Test cases to ensure the system functions correctly under various scenarios, including correct and incorrect OTP entries.
- Optionally, you can create a simple GUI interface for the OTP verification system to enhance user experience.

Project Evaluation:

- Correctness and functionality of the OTP generation, sending, and verification process.
- Code quality, including adherence to Python best practices, readability, and documentation.
- Error handling and user interaction aspects of the system.
- Robustness and reliability of the system under different scenarios.
- Optional: Creativity and usability of the GUI interface (if implemented).

```
import random
```

Provides functions to generate random numbers.

```
import smtplib
```

Used to send mail to any Internet machine with an SMTP listener domain..

```
OTP = random.randint(100000,999999)
```

Generating a random 6-digit OTP using random.randint.

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

setting up server - This creates an SMTP server instance with the host address smtp.gmail.com and port number 587, which is commonly used for secure email transmission.

```
server.starttls()
```

Initiates a secure connection to the SMTP server using Transport Layer Security (TLS)

```
name = input("Enter your name here: ")
global receiver_email
receiver_email = input("Enter your email id: ")
```

Name use to user input their name and stores it in the variable name.

reciver_email to use user input their email address and stores it in the variable receiver_email.

Enter your name here: SURYAKANT

Enter your email id: ranjansuryakant981@gmail.com

```
def email_verification(receiver_email):
    email_check_1 = ["gmail","yahoo","outlook","hotmail"]
    email_check_2 = [".co.in",".org",".com",".in",".edu"]
    count = 0
    for domain in email_check_1:
        if domain in receiver_email:
            count+=1
    for site in email_check_2:
        if site 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: ")
        email_verification(new_receiver_email)
        return new_receiver_email
    return receiver_email
```

this prompts the user to re-enter the email address if it doesn't match the criteria.

```
valid_receiver_email = email_verification(receiver_email)
password = "wryp krww gwmz jyrw"
server.login("votp0317@gmail.com",password)

body = "dear "+name+","+"\\n"+ "\\n"+ " your otp for project verification is "+str(OTP)+"."
subject = "OTP verification using python by suryakant"
message = f"subject:{subject}\\n\\n{body}"
server.sendmail("votp0317@gmail.com",valid_receiver_email,message)
```

email_verification() function to validate the entered email address and stores the validated email address in the variable valid_receiver_email.

logs in to the Gmail SMTP server using the provided email address and password.

generate password using google account

Sends the email message to the validated receiver email address.

```
def sending_otp(receiver_email):
    new_otp = random.randint(100000,999999)

    body = "Dear"+name+","+"\n"+ "\n"+ " your otp is "+str(new_otp)+"."
    message = f"subject:{subject}\n\n{body}"
    server.sendmail("votp0317@gmail.com",receiver_email,message)
    print("OTP has been sent to: "+receiver_email)
    received_otp = int(input("Enter your OTP: "))

    if received_otp == new_otp:
        print("your is verified")
    else:
        print("oops! invalid otp")
        print("sending new otp")
        sending_otp(receiver_email)
```

The sending_otp(receiver_email) function sends a new OTP to the provided email address, verifies the received OTP, and ask the user to enter the OTP. If the OTP is correct, it prints a success message; otherwise, it redirect to resend OTP.

checks if the OTP entered by the user matches the initially generated OTP. If not, it provides options to resend OTP to the same email or enter a new email address for OTP verification.

```
print("otp has sent to "+valid_receiver_email)
received_otp = int(input("Enter your otp here: "))

if received_otp == OTP:
    print("your otp verified")
else:
    print("Oops!! Invalid OTP")
    answer = input("Enter 'YES' to resend OTP on same email and 'NO' to enter a new email id:")
    YES = ['YES', 'yes', 'Yes']
    NO = ['NO', 'no', 'No']
    if answer in YES:
        sending_otp(valid_receiver_email)
    elif answer in NO:
        new_receiver_email = input("Enter new email id: ")
        email_verification(new_receiver_email)
        sending_otp(new_receiver_email)
    else:
        print("Invalid input")

server.quit()
```

otp has sent to ranjansuryakant981@gmail.com
Enter your otp here: 739417
your otp verified

Thank you!

