

# OTP Verification System Using Python

## Overview:

The OTP (One Time Password) Verification System is a Python application designed to provide a secure method for verifying user identities through the use of one-time passwords sent via email. This project utilizes the Tkinter library to create a graphical user interface (GUI) for user interaction.

### **\*\*The main functionalities of the OTP Verification System include\*\***

1. **OTP Generation:** The system generates a random six-digit OTP whenever a user requests verification.
2. **Email Sending:** The generated OTP is sent to the user's email address using the SMTP (Simple Mail Transfer Protocol) server. The system utilizes a Gmail SMTP server for email delivery.
3. **OTP Validation:** Upon receiving the OTP, the user enters it into the application. The system then verifies whether the entered OTP matches the generated OTP. If the OTP is valid, the user is granted access; otherwise, an error message is displayed.
4. **Limiting OTP Attempts:** The system limits the number of OTP verification attempts to enhance security. After a certain number of unsuccessful attempts, the system restricts further verification attempts for a specified period.
5. **User-Friendly Interface:** The application features a user-friendly GUI that allows users to easily enter their email address, receive OTPs, and verify their identity.

## **Functionality:**

1. **generateOTP():** This function generates a random six-digit OTP using the random module in Python (One Time Password) using numbers from 0 to 9.

2. **sendOTP():** This function sends the generated OTP to the email address entered by the user. It uses SMTP (Simple Mail Transfer Protocol) to send the email via a Gmail SMTP server.
3. **checkOTP():** This function checks whether the entered OTP matches the generated OTP. If it matches, it displays a success message; otherwise, it increments the attempt count and displays an error message. If the maximum number of attempts is reached, it disables the "Send OTP" button.

### How to Run the OTP Verification system:

1. Ensure that you have Python and required libraries installed on your system.
2. Below are the libraries in python required to run the system
3. Install the required dependencies: smtplib for email sending and tkinter for the GUI.
4. Copy the provided code into a Python file (e.g., otp\_verification.py).
5. Update the sender and password variables with your Gmail email address and password.
6. Run the Python script (python otp\_verification.py).
7. A GUI window will appear, prompting you to enter the recipient's email address.
8. Click the "Send OTP" button to send the OTP to the entered email address.
9. Enter the received OTP in the OTP field and click the "Verify" button to validate it.

### Dependencies to run the system:

The OTP verification system has the following dependencies:

random: Standard library module for generating random numbers.

smtplib: Standard library module for sending emails using SMTP.

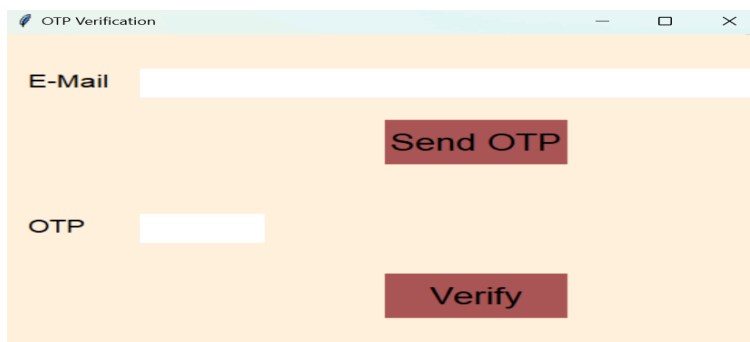
tkinter: Standard library module for creating GUI applications.

```
A. import random
B. import smtplib
C. from tkinter import *
D. from tkinter import messagebox
E. import threading
F. import time
```

Ensure that these modules are installed and accessible in your Python environment.

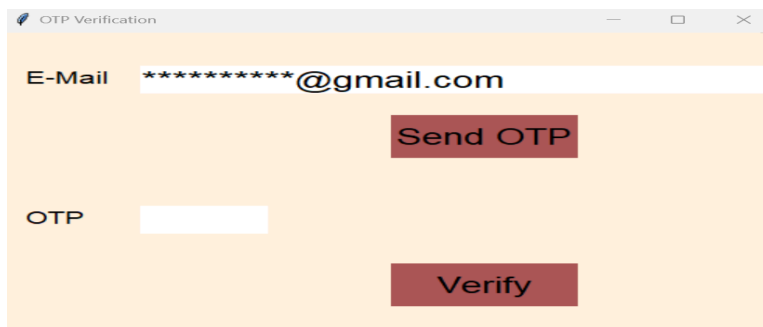
**Here are some test cases to ensure the OTP Verification System functions correctly under various scenarios:**

1. Once the program run the window like shown in the below screenshot will appear.



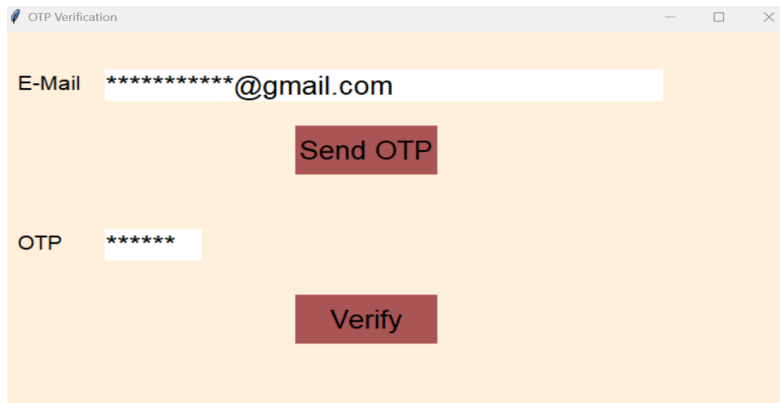
The screenshot shows a window titled "OTP Verification". It has a light orange background. At the top, there is a label "E-Mail" followed by an empty text input field. Below this, there is a dark red button labeled "Send OTP". Further down, there is a label "OTP" followed by an empty text input field. At the bottom, there is a dark red button labeled "Verify".

2. User will be asked to enter the valid email id and click on send OTP



The screenshot shows the same "OTP Verification" window. The "E-Mail" input field is now filled with the text "\*\*\*\*\*@gmail.com". The "Send OTP" button is still present and visible. The "OTP" input field and the "Verify" button remain unchanged from the previous state.

3. Once the OTP received on the entered mail-id enter the 6 digit otp in OTP field and click ok verify.



OTP Verification

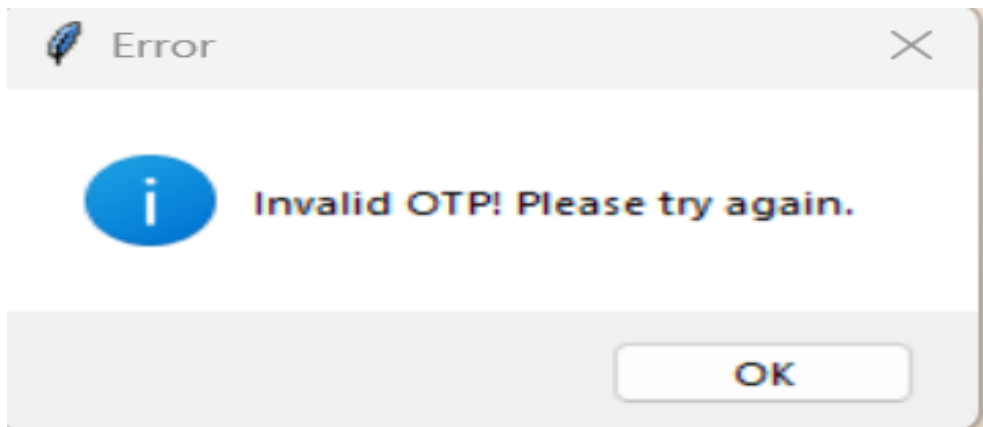
E-Mail \*\*\*\*\*@gmail.com

Send OTP

OTP \*\*\*\*\*

Verify

4. If the entered OTP is invalid then the pop like the below screenshot will appear

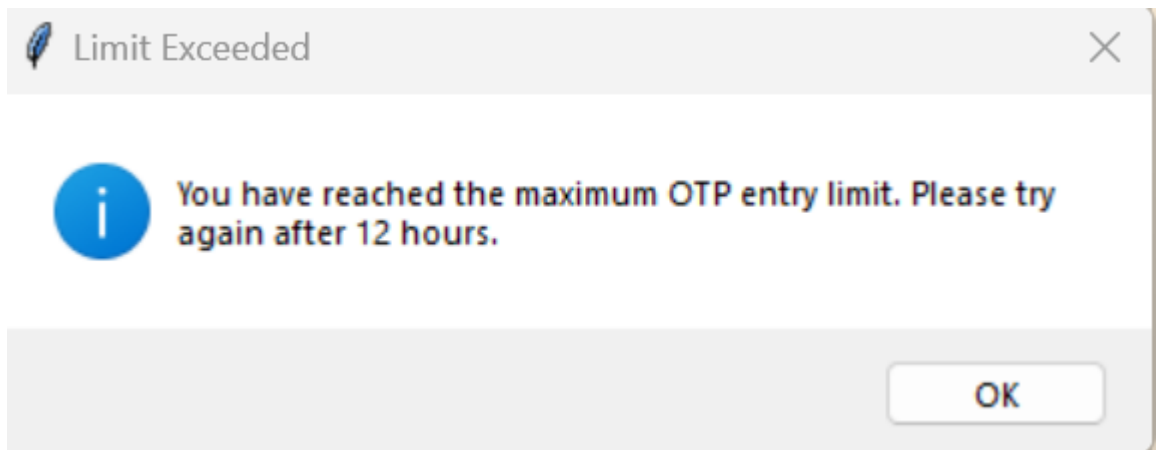


5. Once the entered OTP is valid the below pop will appear:

6.



7. If the user reach the max OTP enter limit that is 3 the below warning window will appear:



User need to try verifying the mail after 12 hours.