**Python\_Capstone\_Project\_Documentation**

### **Title: OTP Verification System**

### **Overview**

This Python application provides a simple system for verifying One-Time Passwords (OTPs). The system sends an OTP to a user's email and allows them to enter it in the GUI to gain access. If the OTP entered is correct, access is granted; if not, the system allows a limited number of retries.

### **Dependencies**

### **To run the OTP verification system, the following Python libraries are required:**

### **smtplib: For sending emails via SMTP.**

### **random: For generating a random OTP.**

### **tkinter: For creating the graphical user interface.**

### **email.mime.text & email.mime.multipart: For constructing and sending email messages.**

### **Functionality of Each Component**

1. **generate\_otp()**:
   * Generates a random 6-digit OTP to be sent to the user’s email address.
2. **send\_email(recipient\_email, otp)**:
   * Sends the generated OTP to the provided email address using SMTP.
   * Configures Gmail's SMTP server for sending the email.
3. **send\_otp\_gui()**:
   * Retrieves the email entered by the user and triggers the sending of the OTP.
   * Resets the retry count to 3 each time a new OTP is sent.
4. **verify\_otp\_gui()**:
   * Verifies the entered OTP against the generated OTP.
   * Allows up to 3 retry attempts. If the OTP is incorrect, the user is informed of the remaining attempts.
   * If the OTP is correct, the application grants access and closes the window.
5. **validate\_otp\_input(new\_value)**:
   * Ensures that only digits are entered in the OTP input field and restricts the input to a maximum of 6 digits.
6. **GUI Configuration**:
   * The graphical user interface is built using tkinter, which provides input fields for email and OTP, buttons for sending and verifying OTPs, and feedback messages for success, warnings, and errors.

### **How to Run the Program**

1. **Prepare the Python environment**:
   * Ensure Python is installed on your machine.
   * Install the necessary libraries using pip if they aren't already installed.
2. **Set up the email account**:
   * In the code, replace the sender email and the corresponding app password with your own Gmail account details.
   * Ensure that the sender’s email account allows "Less Secure Apps" or generate an App Password for Gmail to authenticate SMTP connections (for Google accounts with 2FA enabled).
3. **Run the script**:
   * Execute the Python script, and the OTP verification system will appear as a GUI.
   * Enter the email, click "Send OTP", and an OTP will be sent to the provided email address.
   * Once you receive the OTP, enter it into the application and click "Verify OTP".
4. **Test the application** by entering both correct and incorrect OTPs.

**Python Script: OTP Verification System**

import smtplib

import random

import tkinter as tk

from tkinter import messagebox

from email.mime.text import MIMEText

from email.mime.multipart import MIMEMultipart

**# Function to generate a 6-digit OTP**

def generate\_otp():

return str(random.randint(100000, 999999))

**# Function to send the OTP to the user's email**

def send\_email(recipient\_email, otp):

try:

**# Email configurations**

sender\_email = "monakumari08@gmail.com"

sender\_password = "niueguwoshoruejc"

**# Setting up the email message**

subject = "Your OTP Verification Code"

body = f"Your One-Time Password (OTP) is: {otp}\n\nThis OTP is valid for 5 minutes."

message = MIMEMultipart()

message['From'] = sender\_email

message['To'] = recipient\_email

message['Subject'] = subject

message.attach(MIMEText(body, 'plain'))

**# Sending email using SMTP server**

with smtplib.SMTP('smtp.gmail.com', 587) as server:

server.starttls() # Enable encryption

server.login(sender\_email, sender\_password)

server.send\_message(message)

print("OTP sent successfully to email.")

except smtplib.SMTPAuthenticationError:

messagebox.showerror("Error", "Authentication failed. Check email or app password.")

exit(1)

except Exception as e:

messagebox.showerror("Error", f"Failed to send email: {e}")

exit(1)

**# GUI: Function to handle sending OTP**

def send\_otp\_gui():

email = email\_entry.get()

if email:

otp = generate\_otp()

send\_email(email, otp)

global generated\_otp

generated\_otp = otp

global retries

retries = 3 # Reset retries when a new OTP is sent

messagebox.showinfo("Success", "OTP sent to your email!")

else:

messagebox.showwarning("Warning", "Please enter an email address.")

**# GUI: Function to handle verifying OTP with retries**

def verify\_otp\_gui():

global retries # Track retries

entered\_otp = otp\_entry.get().strip()

if entered\_otp == generated\_otp:

messagebox.showinfo("Success", "OTP verified successfully! Access granted.")

root.destroy() # Close the application upon successful verification

else:

retries -= 1

if retries > 0:

messagebox.showwarning("Warning", f"Incorrect OTP. You have {retries} attempt(s) remaining.")

else:

messagebox.showerror("Error", "Verification failed. Access denied.")

root.destroy() # Close the application after exhausting retries

**# Validation function to restrict OTP input to 6 digits**

def validate\_otp\_input(new\_value):

**# Allow only digits and restrict to a maximum of 6 characters**

return new\_value.isdigit() and len(new\_value) <= 6

**# Setting up GUI**

root = tk.Tk()

root.title("OTP Verification System")

root.geometry("400x300") # Set window size

root.configure(bg="#7BC9FF") # Set background color

**# Custom Styles**

label\_font = ("Arial", 11, "bold")

entry\_font = ("Arial", 11)

button\_font = ("Arial", 11, "bold")

button\_bg = "#4834d4" # Dark blue for buttons

button\_fg = "#ffffff" # White text for buttons

label\_fg = "#130f40" # Dark color for text labels

**# Header Label**

header\_label = tk.Label(root, text="OTP Verification System", font=("Arial", 16, "bold"), bg="#D9EAFD", fg="#333")

header\_label.pack(pady=10)

**# Email Label and Entry**

email\_label = tk.Label(root, text="Enter Email:", font=label\_font, bg="#D9EAFD", fg="#333")

email\_label.pack(pady=(10, 5))

email\_entry = tk.Entry(root, font=entry\_font, width=30, relief="solid", bd=1) # Added border with `relief` and `bd`

email\_entry.pack(pady=(0, 10))

**# Send OTP Button**

send\_otp\_button = tk.Button(root, text="Send OTP", font=button\_font, bg=button\_bg, fg=button\_fg, command=send\_otp\_gui)

send\_otp\_button.pack(pady=(0, 20))

**# OTP Label and Entry**

otp\_label = tk.Label(root, text="Enter OTP:", font=label\_font, bg="#D9EAFD", fg="#333")

otp\_label.pack(pady=(10, 5))

**# Set up validation for OTP entry**

otp\_validation = root.register(validate\_otp\_input)

otp\_entry = tk.Entry(root, font=entry\_font, width=15, validate="key", validatecommand=(otp\_validation, "%P"), relief="solid", bd=1)

otp\_entry.pack(pady=(0, 10))

**# Verify OTP Button**

verify\_otp\_button = tk.Button(root, text="Verify OTP", font=button\_font, bg=button\_bg, fg=button\_fg, command=verify\_otp\_gui)

verify\_otp\_button.pack(pady=(0, 20))

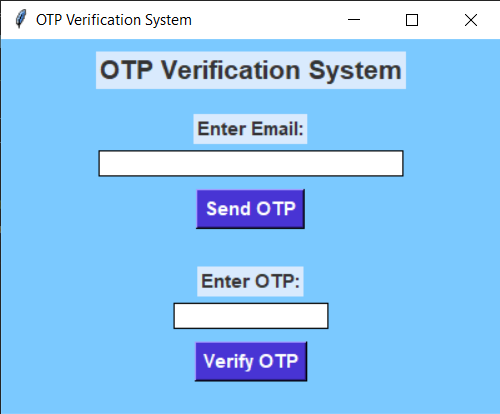
**# Run the GUI**

root.mainloop()

**GUI Interface**

To create a GUI for this project, I used **tkinter**. Below is a brief outline of how the GUI version could work:

1. **Design:**
   * Textbox to enter the user's email.
   * Button to send OTP.
   * Textbox to enter the received OTP.
   * Button to verify OTP.
   * Label to display results (success or failure).



## ****Test Cases****

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Test Case Description** | **Steps** | **Expected Result** |
| TC01 | Sending OTP | Enter a valid email address and click "Send OTP". | An OTP is sent to the specified email address. The user is informed with a "Success" message: "OTP sent to your email!". The retry count is reset to 3. |
| TC02 | Correct OTP Verification | Enter the correct OTP received in the email and click "Verify OTP". | The system successfully verifies the OTP. A success message "OTP verified successfully! Access granted." appears. The application window closes. |
| TC03 | Incorrect OTP with Retry | Enter an incorrect OTP and click "Verify OTP". | The system shows a warning message: "Incorrect OTP. You have 2 attempt(s) remaining." (assuming 1 attempt already). Retry count decreases. |
| TC04 | Incorrect OTP with Retry (continued) | Enter 2 more incorrect OTPs. | After 3 incorrect attempts, the system shows an error message: "Verification failed. Access denied." The application window closes. |
| TC05 | Input Validation for OTP Field | Enter non-digit characters in the OTP input field. | The OTP input field rejects non-digit characters and only allows up to 6 digits. |
| TC06 | Edge Case - Empty Email | Click "Send OTP" without entering an email address. | A warning message: "Please enter an email address" is displayed. |
| TC07 | Edge Case - Empty OTP | Click "Verify OTP" without entering any value in the OTP field. | The system will not verify the OTP and may display an error or remain inactive until a valid OTP is entered. |
| TC08 | Retry Limit Exceeded | Enter 4 consecutive incorrect OTPs. | After the 3rd incorrect attempt, the system should close with the message: "Verification failed. Access denied." |