**OTP Verification System Documentation**

**Overview:**

This program is an OTP (One-Time Password) verification system built using Python, Streamlit, and Gmail's SMTP service. The system sends a randomly generated 6-digit OTP to a user's email, allowing them to verify their identity by entering the OTP. The user has a limited number of attempts to enter the correct OTP.

**Dependencies:**

The program requires the following Python libraries:

1. Streamlit: For creating the web app interface.
   * Install via pip install streamlit
2. smtplib: For sending emails via Gmail's SMTP service (built-in Python library).
3. EmailMessage: For composing and sending emails (built-in Python library).
4. random: For generating random OTPs (built-in Python library).
5. os: For accessing environment variables (built-in Python library).
6. dotenv: For loading environment variables from a .env file.
   * Install via pip install python-dotenv

**How to Run the Program:**

1. Install the Dependencies: Install the required libraries using pip:

pip install streamlit python-dotenv

1. Create a .env File: In the same directory as your script, create a .env file that will contain your Gmail credentials.

EMAIL\_ADDRESS=your\_email@gmail.com

EMAIL\_PASSWORD=your\_app\_password

Here, app password is used to generate the password

1. Run the Streamlit Application: Open a terminal in your project directory and run the following command to start the Streamlit app:

streamlit run otp\_verification.py

This will launch the app and provide a link to access it in your browser.

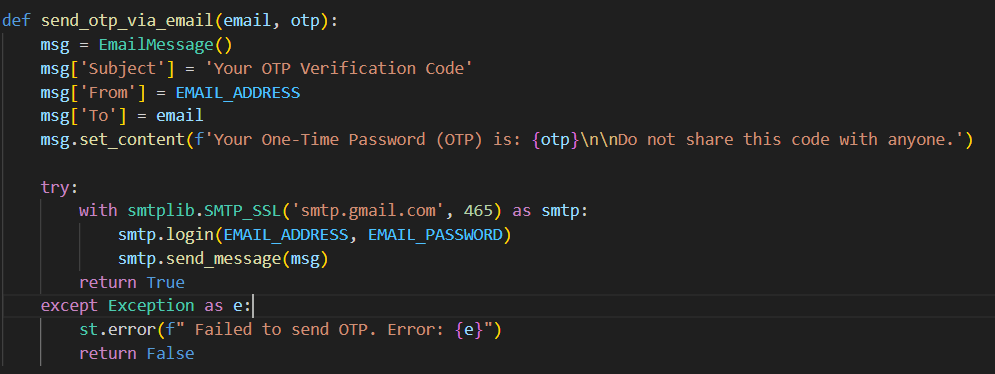
**Explanation of Functions:**

1. **generate\_otp():**
   * Description: This function generates a random 6-digit OTP.
   * Return Value: A 6-digit string representing the OTP.

**A black background with white text

AI-generated content may be incorrect.**

1. **send\_otp\_via\_email(email, otp):**
   * Description: This function sends the OTP to the specified email address using Gmail’s SMTP service.
   * Parameters:
     + email: The recipient’s email address (string).
     + otp: The OTP to be sent (string).
   * Return Value: Returns True if the email was successfully sent, otherwise False.

****

The values in the msg are used to compose the email which contains the otp.

**smtplib.SMTP\_SSL('smtp.gmail.com', 465):**

* smtplib.SMTP\_SSL: This is a class in Python's smtplib module that is used for sending email over a secure SSL (Secure Socket Layer) connection.
* 'smtp.gmail.com': This is the Gmail SMTP server address. SMTP stands for Simple Mail Transfer Protocol, and it's the protocol used for sending emails. smtp.gmail.com is the server address used to send emails through Gmail.
* 465: This is the port number used for SSL (secure) email communication. Port 465 is commonly used for secure email sending (SSL encryption) with Gmail. This ensures that the communication between the client (your code) and the Gmail server is encrypted and secure.

1. **Streamlit UI:**
   * st.title(" OTP Verification System"): Displays the title of the application.
   * Email Input: Users are prompted to enter their email address.

A black background with white text

AI-generated content may be incorrect.

email\_input = st.text\_input("Enter your email address:")

* + Send OTP Button: When clicked, the app generates an OTP, sends it to the email, and sets the session state for OTP and attempt tracking.

**A computer screen with text

AI-generated content may be incorrect.**

* + OTP Verification Input: If the OTP has been sent, users are prompted to enter the OTP they received.

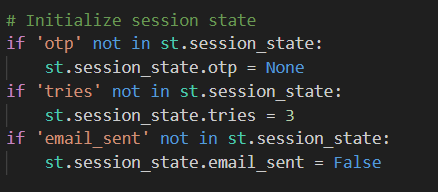
user\_input = st.text\_input("Enter the OTP received:")

* + Verify OTP Button: When clicked, the app compares the entered OTP with the generated OTP and provides feedback to the user.

**A screen shot of a computer program

AI-generated content may be incorrect.**

1. **Session State:**
   * st.session\_state is used to store the OTP, number of attempts remaining, and whether the email has been sent.
   * This ensures that the state persists across interactions within the session.

****

**Usage Instructions:**

1. **Start the App: Once the app is running, the user will be prompted to enter their email address.**

**A screen shot of a computer

AI-generated content may be incorrect.**

1. **Request OTP:** After clicking the "Send OTP" button, an OTP will be generated and sent to the entered email address.
2. **Verify OTP:** After receiving the OTP, the user will be prompted to enter the OTP into the app.
   * If the OTP is correct, the app will display a success message: "Access Granted. OTP Verified Successfull
   * If the OTP is incorrect, the user will be given 3 attempts to enter the correct OTP. If all attempts fail, access will be denied.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer error

AI-generated content may be incorrect.**A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.**