**OTP Generation and Verification with GUI using Python**

**Overview**

This project provides a Python-based solution for generating, sending, and verifying a 6-digit OTP (One-Time Password). The application includes a GUI (Graphical User Interface) built using the tkinter library. Users can generate an OTP, have it sent to their email, and then verify the OTP through the application interface.

**Features**

* **OTP Generation**: A 6-digit OTP is randomly generated.
* **Email Sending**: The OTP is sent to a specified email address.
* **OTP Verification**: The user can verify the OTP by entering it into the application.
* **GUI Interface**: A user-friendly interface to interact with the application.
* **Exception Handling**: Manages invalid email addresses, authentication errors, and OTP validation errors.

**Requirements**

* Python 3.x
* tkinter (included with Python)
* Gmail account for sending OTPs

**Files**

* **main.py**: The main script containing the OTP generation, sending, verification logic, and the GUI.

**Class & Function Descriptions**

**OTPGenerator Class**

This class handles the OTP generation, email sending, and OTP verification.

* **Attributes**:
  + otp: Stores the generated OTP.
  + expiry\_time: Stores the expiry time for the OTP.
* **Methods**:
  + generate\_otp(): Generates a 6-digit OTP and sets an expiry time of 5 minutes.
  + send\_otp(to\_email): Sends the generated OTP to the specified email address.
    - **Parameters**:
      * to\_email: The recipient's email address.
    - **Exceptions**:
      * SMTPAuthenticationError: Raised if the email credentials are incorrect.
      * SMTPRecipientsRefused: Raised if the recipient's email address is invalid.
      * Generic exception handling for other unexpected errors.
  + verify\_otp(input\_otp): Verifies the input OTP against the generated OTP.
    - **Parameters**:
      * input\_otp: The OTP entered by the user.
    - **Returns**:
      * Success or error message based on the verification outcome.
    - **Exceptions**:
      * ValueError: Raised if the input OTP is not a number.
      * Generic exception handling for other unexpected errors.

**OTPApp Class**

This class manages the GUI using tkinter.

* **Attributes**:
  + root: The main window of the application.
  + otp\_generator: An instance of the OTPGenerator class.
* **Methods**:
  + send\_otp(): Gets the email address from the GUI input and calls the send\_otp method of the OTPGenerator class.
  + verify\_otp(): Gets the OTP from the GUI input and calls the verify\_otp method of the OTPGenerator class.

**Usage Instructions**

1. **Setting Up Email Credentials**:
   * Replace your\_email@gmail.com and your\_app\_password in the send\_otp method with your Gmail address and App Password (if 2-Step Verification is enabled).
2. **Running the Application**:
   * Execute the main.py script.
   * A GUI window will appear where you can:
     + Enter an email address and click "Send OTP" to receive the OTP.
     + Enter the received OTP and click "Verify OTP" to verify it.
3. **Handling Common Errors**:
   * If you encounter SMTPAuthenticationError, ensure your email and password (or App Password) are correct.
   * If the email address is invalid, an error message will be displayed.
   * If the OTP is incorrect or expired, appropriate feedback will be provided.

**Security Considerations**

* **Use App Passwords**: If you have 2-Step Verification enabled on your Gmail account, use an App Password instead of your regular password for sending OTPs.
* **Environment Variables**: Store your email credentials in environment variables or a secure configuration file instead of hardcoding them in the script.

### ****Example usage in GUI****

### root = tk.Tk()

### app = OTPApp(root)

### root.mainloop()

### ****Conclusion****

This project demonstrates a simple yet functional OTP-based authentication system with a user-friendly GUI. It is designed to be easily extendable and secure, with robust error handling to manage common issues during email communication and OTP validation.





