**Documentation for OTP Verification System project**

**#OTP Verification System Documentation**

Project Overview:

The OTP Verification System is a Python-based application designed to generate and validate OTPs (One-Time Passwords) for user authentication. The system generates a 6-digit OTP and sends it to the user's email address. The user then enters the received OTP for verification. If the entered OTP matches the generated OTP, access is granted; otherwise, access is denied.

Project Structure:

The project consists of two main components: the backend logic and the graphical user interface (GUI). The backend logic handles OTP generation, email sending, and OTP verification, while the GUI provides a user-friendly interface for interacting with the system.

Components and Its Functionality:

**#OTPverify\_Backend.py :**

* **OTPVerifyFeatureBackend Class :** This class contains the backend logic for OTP generation, email sending, and OTP verification. It includes methods for generating OTPs, validating email addresses, sending OTPs via email, prompting users for OTP input, and verifying entered OTPs.

1. **generate\_otp(self) :**
   * Generates a 6-digit OTP using random integers.
   * Returns the generated OTP if it consists of exactly 6 digits; otherwise, returns None.
2. **verify\_email(self, email) :**
   * Verifies the format of the provided email address using a regular expression.
   * Returns True if the email format is valid; otherwise, returns False.
3. **send\_otp(self, email) :**
   * Generates an OTP using the `generate\_otp` method.
   * Constructs an email message containing the OTP and sends it to the provided email address using SMTP.
   * Prints a success message if the OTP is sent successfully and returns a tuple containing a boolean indicating success and the generated OTP.
4. **prompt\_for\_otp(self) :**
   * Prompts the user to enter the OTP sent to their email address.
   * Returns the entered OTP as a string.
5. **verify\_otp(self, generated\_otp, entered\_otp) :**
   * Compares the entered OTP with the generated OTP.
   * Returns True if the entered OTP matches the generated OTP; otherwise, returns False.
6. Main Function

**Def main (self) : (if \_\_name\_\_ == '\_\_main\_\_'):**

* + Contains commented-out code for testing the functionality of the OTPVerifyFeatureBackend class.
  + Allows manual testing of OTP generation, sending, and verification by interacting with the console.
  + Validates the OTP verification system's functionality and error handling.

**Note:** The commented-out main function is intended for manual testing and validation purposes. Uncommenting and executing this section will allow testing of OTP generation, sending, and verification directly through the console.

* **OTPVerification Class :** This class acts as an interface between the GUI and the backend features. It provides methods for sending OTPs and verifying entered OTPs.

1. **\_\_init\_\_(self) :**
   * Initializes the OTPVerification class.
   * Creates an instance of the OTPVerifyFeatureBackend class to access OTP generation, sending, and verification features.
2. **send\_otp(self, email) :**
   * Calls the `send\_otp` method of the OTPVerifyFeatureBackend class to send the OTP to the provided email address.
   * Returns a tuple containing a boolean indicating whether the OTP sending was successful and the generated OTP.
3. **verify\_otp(self, generated\_otp, entered\_otp) :**
   * Calls the `verify\_otp` method of the OTPVerifyFeatureBackend class to verify the entered OTP against the generated OTP.
   * Returns a boolean indicating whether the entered OTP matches the generated OTP.

**#OTPverify\_GUI.py :**

* **OTPVerifyGUI Class :** This class implements the graphical user interface using PyQt5. It includes widgets for inputting email addresses and OTPs, buttons for sending and verifying OTPs, and labels for displaying status messages.

1. **\_\_init\_\_(self) :**
   * Initializes the OTPVerifyGUI class, creating the graphical user interface (GUI) for OTP verification.
   * Sets the window title, background color, and geometry (size and position).
   * Creates labels, input fields, status labels, and buttons for email, OTP, status messages, and actions (send OTP, verify OTP).
   * Configures the style (font size, weight, color, background color, border) of labels, buttons, and input fields.
   * Sets placeholder text for email and OTP input fields and disables the OTP input initially.
   * Connects the `clicked` signals of the send OTP and verify OTP buttons to their respective methods.
2. **send\_otp\_button(self) :**
   * Retrieves the email entered by the user from the email input field.
   * Validates the email format using a regular expression.
   * If the email format is valid, calls the `send\_otp` method of the `OTPVerification` class to send the OTP.
   * If OTP sending is successful, enables the OTP input field, updates the status label, and stores the generated OTP.
   * If OTP sending fails, disables the OTP input field and updates the status label with an error message.
3. **verify\_otp\_button(self) :**
   * Retrieves the entered OTP from the OTP input field.
   * Retrieves the generated OTP from the instance variable `generated\_otp`.
   * If a generated OTP exists and the entered OTP is not empty, calls the `verify\_otp` method of the `OTPVerification` class to verify the OTP.
   * Updates the status label with either a success message for correct OTP or an error message for incorrect OTP.
   * If no OTP is generated, updates the status label with an error message prompting the user to send OTP first.

**Main Function (if \_\_name\_\_ == '\_\_main\_\_'):**

* + Initializes a PyQt5 application.
  + Creates an instance of the OTPVerifyGUI class, representing the OTP verification GUI window.
  + Shows the OTP verification GUI window.
  + Executes the PyQt5 application event loop.

**Note:** The `OTPVerifyGUI` class encapsulates the graphical user interface for OTP verification, while the functionality for OTP generation, sending, and verification is implemented in the `OTPVerification` class imported from `OTPverify\_Backend`.

Dependencies & Packages :

* + Python 3.x
  + PyQt5 (for GUI implementation)
  + SMTPlib (for email feature )
  + Re
  + Random
  + Sys
  + Tkinter (2nd option for GUI)

Usage:

1. Run the OTPVerifyGUI.py script to launch the OTP Verification System.
2. Enter your email address in the provided input field and click the "Send OTP" button.
3. Check your email inbox for the OTP sent by the system.
4. Enter the received OTP in the OTP input field and click the "Verify OTP" button.
5. If the entered OTP is valid, access will be granted; otherwise, access will be denied.

How to Run the Program:

1. Clone the repository to your local machine.
2. Navigate to the project directory.
3. Run the Python script OTPverify\_Backend.py
4. After successfull execution of OTPverify\_Backend.py execute the OTPverify\_GUI.py
5. Follow the prompts to generate, send, and verify the OTP.

Test Cases:

1. **Valid Email Address :** Verify that the system accepts valid email addresses for OTP delivery.
2. **Invalid Email Address :** Verify that the system rejects invalid email addresses and prompts the user to enter a valid one.
3. **Successful OTP Generation :** Verify that the system successfully generates a 6-digit OTP and sends it to the user's email address.
4. **Invalid OTP Entry :** Verify that the system prompts the user to enter a valid OTP if the entered OTP is incorrect.
5. **OTP Verification :** Verify that the system correctly verifies the entered OTP and grants access if it matches the generated OTP.