# OTP Verification System

## Overview

This script implements an OTP (One-Time Password) verification system that:

1. Generates a random 6-digit OTP.
2. Sends the OTP to a verified email address.
3. Prompts the user to input the OTP.
4. Verifies the input OTP by hashing it and comparing it to the original.

## Functionality of Each Function

### 1. `otp_generator()`

- **Purpose:** Generates a random 6-digit OTP.
- **Process:**
    - Uses the `random` library to generate 6 random integers between 0 and 9.
    - Concatenates them as a string.
- **Returns:** A string containing the 6-digit OTP.

### 2. `verify_email()`

- **Purpose:** Verifies if the entered email address matches a valid pattern.
- **Process:**
    - Uses regular expressions (regex) to validate the format of the entered email.
    - Returns the email if valid; otherwise, prints an error message and returns None.
- **Returns:** A valid email address or None if invalid.

### 3. `send_otp(np)`

- **Purpose:** Sends the generated OTP via email to the user.
- **Parameters:**
    - np: The OTP to be sent.

- **Process:**
    - Uses the `smtplib` library to send emails through a Gmail account.
    - Calls `verify_email()` to validate the recipient's email.
    - If the email is valid, sends the OTP in an email body.
- **Returns:** The OTP sent to the user.

### 4. `enter_otp()`

- **Purpose:** Prompts the user to input the OTP received via email.
- **Returns:** The OTP entered by the user as a string.

### 5. `hash_otp(otp)`

- **Purpose:** Hashes the OTP using the `bcrypt` library for secure storage.
- **Parameters:**
    - `otp`: The OTP string to hash.
- **Process:**
    - Hashes the OTP using `bcrypt.hashpw()` with a generated salt.
- **Returns:** A hashed OTP (as bytes).

### 6. `verify_otp(enter_otp, hash_otp)`

- **Purpose:** Verifies the entered OTP against the hashed OTP.
- **Parameters:**
    - `enter_otp`: Function to get user input OTP.
    - `hash_otp`: The hashed OTP for comparison.
- **Process:**
    - Compares the hashed OTP with the user-input OTP using `bcrypt.checkpw()`.
- **Returns:** True if the OTP matches; otherwise, `False`.

## How to Run the Program

1. **Dependencies**

Install the following libraries before running the script:

   a. `bcrypt` for hashing: `pip install bcrypt`

   b. `smtplib` and `re` (both are part of Python's standard library).

2. **Configuration**
   a. Update the sender email and app-specific password in the `send_otp()` function:

```
sender = 'youremail@gmail.com'
password = 'your-app-specific-password'
```

   b. Enable "Allow less secure apps" in your Gmail account settings (or use app-specific passwords).

3. **Running the Script**

Execute the script in a Python environment (Python 3.x):

```
python otp_verification_system.py
```

4. **Steps:**
   a. Enter a valid email address when prompted.
   b. Check your email for the OTP.
   c. Enter the OTP in the terminal.
   d. The program verifies the OTP with a maximum of **3 attempts**.

## Test Cases

### Test Case 1: Successful OTP Verification

- **Input:**
  - A valid email.
  - Correct OTP received via email.
- **Expected Output:** `OTP Verified`

### Test Case 2: Incorrect OTP Entry

- **Input:**

    - A valid email.
    - Incorrect OTP entered within 3 attempts.
- **Expected Output:**
```
OTP Incorrect
Retry
OTP Incorrect
Retry
OTP Incorrect
Retry
```

### Test Case 3: Invalid Email Entry

- **Input:**
    - An invalid email address.
- **Expected Output:** `Invalid email address`

### Test Case 4: Hash Verification

- **Purpose:** Ensures the OTP hashing works and matches only the correct OTP.
- **Steps:**
    - Hash a specific OTP.
    - Input the correct OTP to verify.
    - Input an incorrect OTP to ensure it fails.
- **Expected Behavior:**
    - Correct OTP → `True`.
    - Incorrect OTP → `False`.

## Notes

1. Ensure you have internet access to send emails.
2. The program uses Gmail's SMTP server for email sending. Adjust the email settings if using a different provider.
3. The program handles invalid email inputs and incorrect OTP entries gracefully.