OTP VERIFICATION SYSTEM

BY:- HIMANSHI AGARWAL

Introduction

•**Overview of the Project:**

•Security and privacy are crucial in today's digital world. Authentication processes, like OTPs (One-Time Passwords), ensure that only authorized individuals gain access. This project focuses on generating a 6-digit OTP and sending it to the user's email, providing a secure and reliable verification method for scenarios like online banking and account security

•**What is OTP (One-Time Password):-**

"An OTP is a unique password that is valid for only one login session or transaction, providing an additional layer of security.“

•**Importance of OTP in Secure Authentication:**

"OTP systems are widely used in securing online transactions and login processes, reducing the risk of unauthorized access.“

Problem Statement

- **Problem: Need for Secure User Verification:**

"With increasing online security threats, there’s a growing need for systems that can securely verify user identity.“

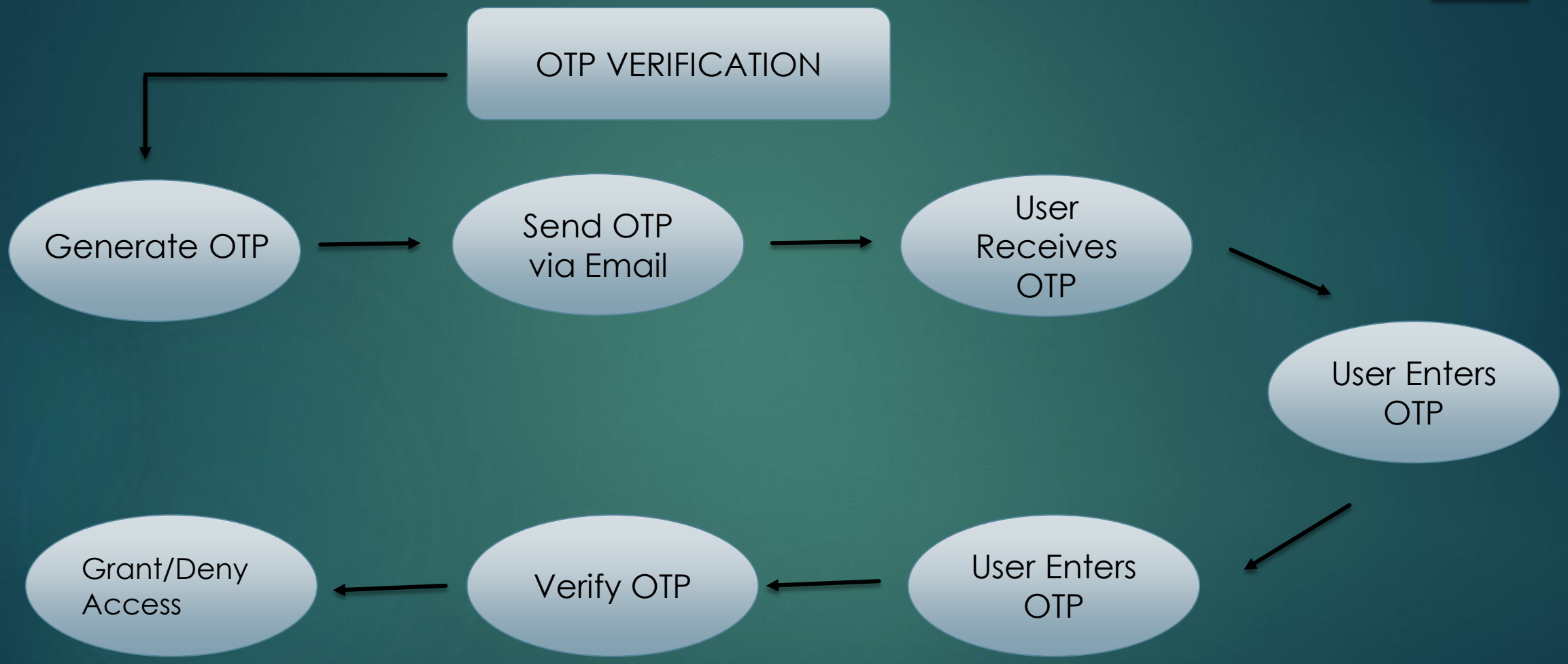
- **Goal: Build a Secure OTP Verification System:**

"The objective of this project is to develop a system that generates, sends, and verifies OTPs, ensuring only authenticated users gain access."

Project Requirements

- Core Requirements: Generate a 6-digit OTP.
- Send OTP via email.
- User OTP input and verification.
- Error handling and user prompts.
- Optional: GUI Interface.

System Architecture



Code Implementation

```python
import random
import smtplib
import time
import os
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
import tkinter as tk
from tkinter import messagebox
```

OTP Generation & Email Sending:

"The system generates a 6-digit OTP using Python's **random** module. It then prepares and sends this OTP to the user's email using the **smtplib** module. The email sending process involves logging into an SMTP server, preparing the email content, and securely sending it to the user."

```python
# Initialize the attempt counter
attempts = 3

# Function to generate OTP and send email
def send_otp():
    global otp, otp_generated_time, attempts
    otp = str(random.randint(100000, 999999))
    otp_generated_time = time.time()

    # Reset attempts
    attempts = 3

    # Retrieve sender's email and password from environment variables
    sender_email = os.getenv("SENDER_EMAIL")
    sender_password = os.getenv("SENDER_PASSWORD")

    if not sender_email or not sender_password:
        messagebox.showerror("Error", "Sender's email or password not set in environment variables.")
        return

    to_email = email_entry.get()
    if not to_email:
        messagebox.showerror("Error", "Please enter your email address.")
        return
```

```python
# Prepare the email
subject = "Your OTP Verification Code"
body = f"{otp} is your One-Time Password for verification."

msg = MIMEMultipart()
msg['From'] = sender_email
msg['To'] = to_email
msg['Subject'] = subject
msg.attach(MIMEText(body, 'plain'))

# Send the email
try:
    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.starttls()
    server.login(sender_email, sender_password)
    server.sendmail(sender_email, to_email, msg.as_string())
    server.quit()
    messagebox.showinfo("Success", "OTP has been sent to your email address.")
except Exception as e:
    messagebox.showerror("Error", f"Failed to send OTP: {e}")
```

•**OTP Verification:Explanation:**

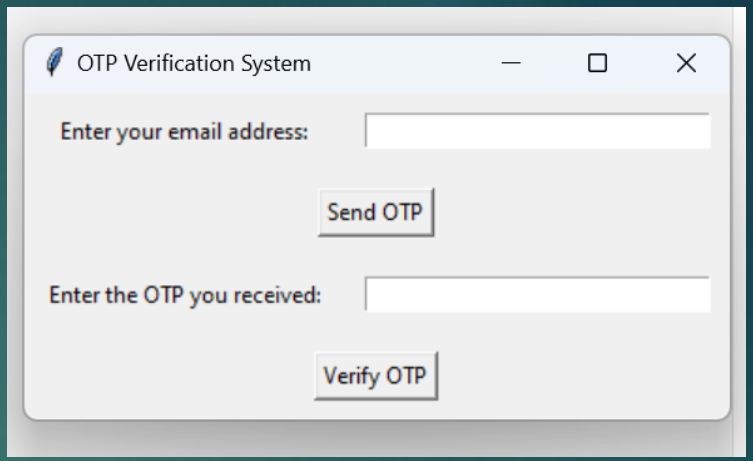
"Once the user receives the OTP, they enter it into the system. The system then compares this user-entered OTP with the originally generated OTP. If they match, access is granted. Otherwise, the user is prompted to retry. Additionally, the system ensures that the OTP expires after 60 seconds, enhancing security."

•**Handling Incorrect Attempts:**

•"If the user enters the wrong OTP, they are given up to 3 attempts to retry. After the third failed attempt, the system denies access, providing a secure environment."

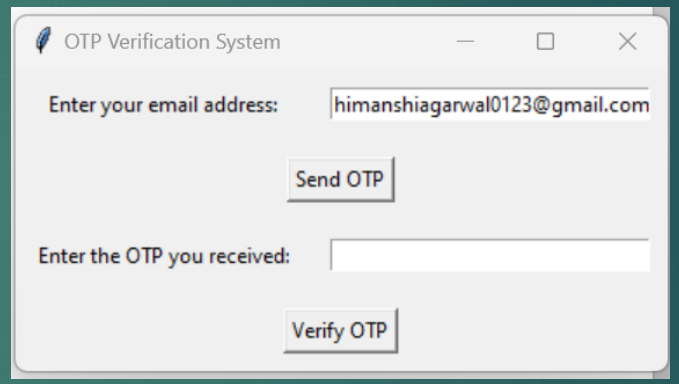
```python
# Function to verify the OTP
def verify_otp():
    global attempts
    user_otp = otp_entry.get()

    # Check if the OTP has expired
    current_time = time.time()
    if current_time - otp_generated_time > 60:  # 60 seconds expiry time
        messagebox.showwarning("Expired", "The OTP has expired. Please request a new one.")
        root.destroy()
        return

    if user_otp == otp:
        messagebox.showinfo("Success", "Your account has been successfully verified.")
        root.destroy()  # Close the GUI window
    else:
        attempts -= 1
        if attempts > 0:
            messagebox.showwarning("Incorrect OTP", f"Incorrect OTP. You have {attempts} attempt(s) left.")
        else:
            messagebox.showerror("Failed", "You have exhausted all attempts. Verification failed.")
            root.destroy()  # Close the GUI window after 3 failed attempts
```

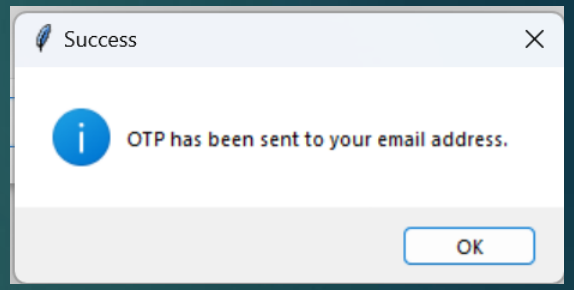
GUI Interface

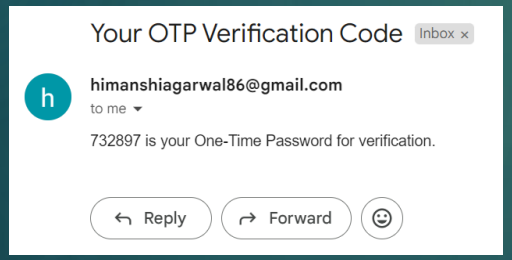


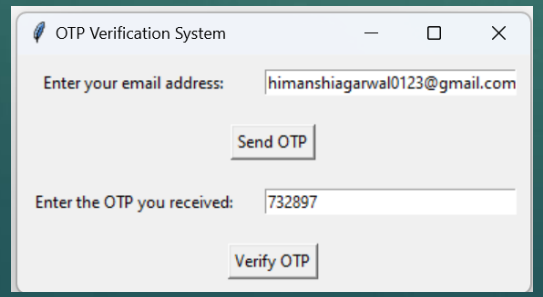
•User Experience:

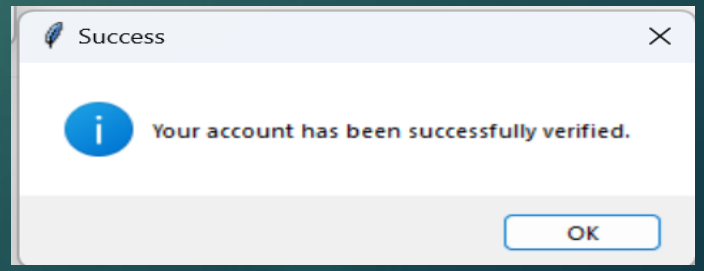
"To make the system more accessible, I've developed a simple graphical user interface (GUI) using **Tkinter.** This GUI allows users to enter their email, request an OTP, and then verify it. User prompts and feedback are provided through dialog boxes, ensuring a smooth user experience."











Testing and Validation

•Test Cases:

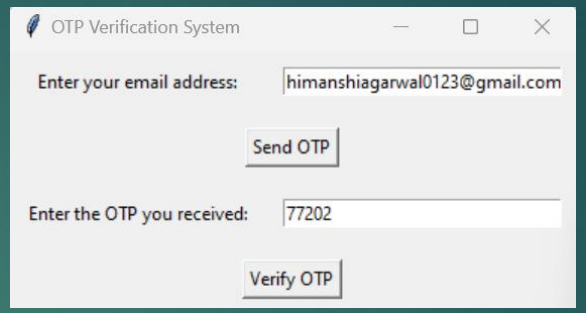
"To ensure the system's reliability, I developed several test cases. These tests cover scenarios such as entering the correct OTP, entering incorrect OTPs, and testing OTP expiration. The system successfully handles all these cases, ensuring robustness."

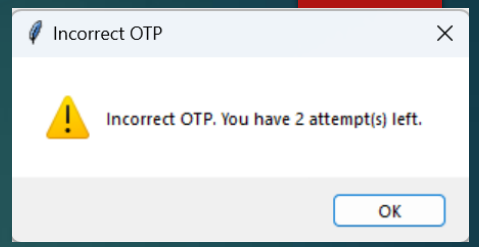
•Handling Correct/Incorrect OTP Entries:

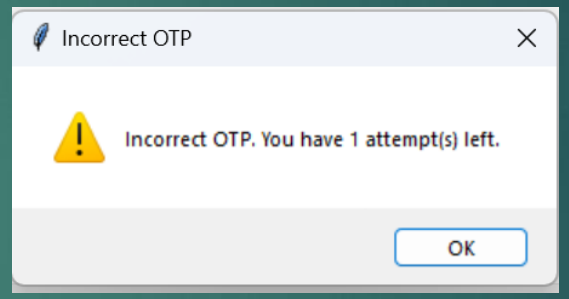
"The system correctly grants or denies access based on the user's input, providing clear feedback at each step."

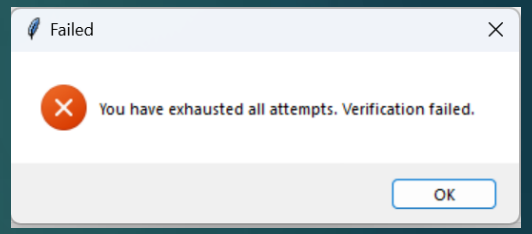
•Testing OTP Expiration and Multiple Attempts:

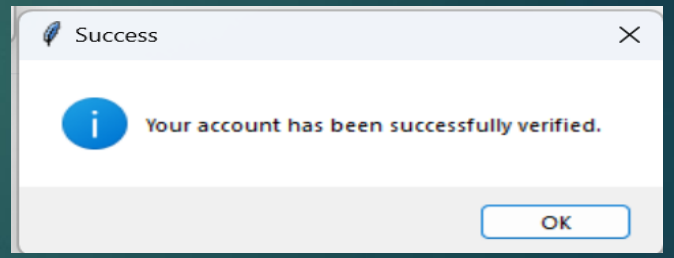
"The expiration of the OTP after 60 seconds and limiting the number of attempts are crucial security features that were thoroughly tested."

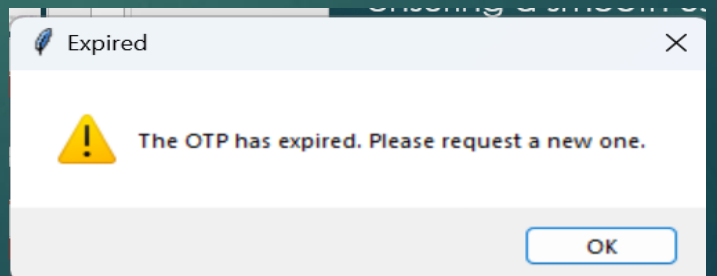












Conclusion

•Summary:

"In conclusion, this OTP Verification System effectively secures user authentication by generating, sending, and verifying one-time passwords. The system's robust error handling and user-friendly GUI enhance the overall user experience."

•Future Improvements:

"Looking ahead, possible improvements could include adding support for sending OTPs via SMS and enhancing the GUI with more features."