Project Overview

This is a simple OTP (One-Time Password) verification system using Python with a Tkinter GUI interface. The program sends an OTP to the user's email, which is then verified by the user. It has attributes like:

- Verification of emails
- The creation and expiration of OTP (after 3 minutes)
- A countdown timer that indicates how long the OTP still has validity
- Limited OTP verification attempts (up to three)
- The SMTP protocol for emailing the OTP to the user.

Dependencies

smtplib: For sending OTP emails via SMTP.

re: Regular expressions used for validating the email format.

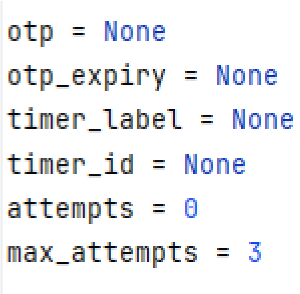
random: Used for generating the random 6-digit OTP.

tkinter: GUI toolkit for creating the graphical interface.

datetime: Used for setting OTP expiration.

messagebox: Displays messages like warnings, errors, and information.

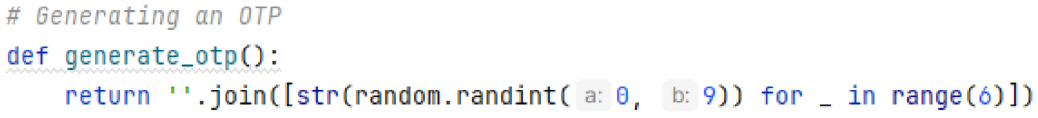
Global Variables



```
otp = None
otp_expiry = None
timer_label = None
timer_id = None
attempts = 0
max_attempts = 3
```

otp: Holds the OTP that was produced.
otp_expiry: This variable holds the OTP's expiration time (three minutes from the moment it is produced).
timer_label: This is where the countdown timer is shown.
timer_id: To record the timer's information for later stopping.
attempts: The user's total number of OTP verification attempts.
max_attempts: The maximum quantity of OTP verification tries that are permitted.

Generating OTP

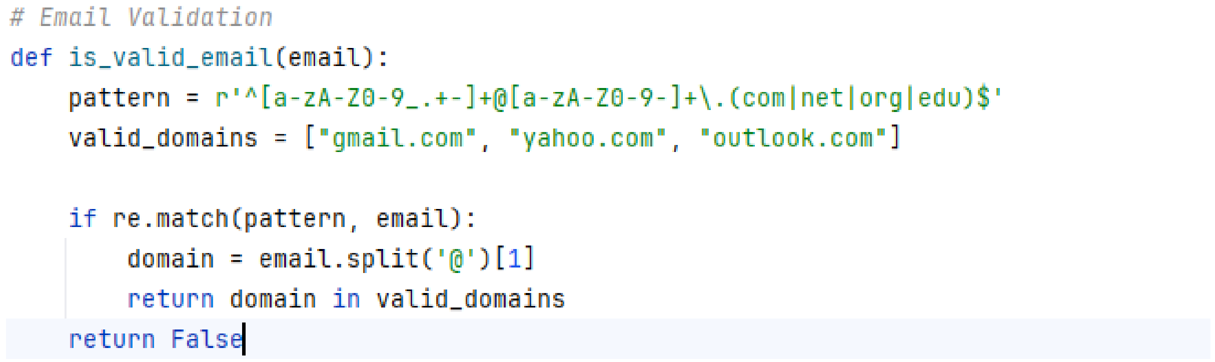


```python
# Generating an OTP
def generate_otp():
    return ''.join([str(random.randint( a: 0,  b: 9)) for _ in range(6)])
```

Purpose: Generates a 6-digit random OTP using the random.randint() function.

Usage: To generate a one-time password, which is sent to the user for verification.

Email Validation

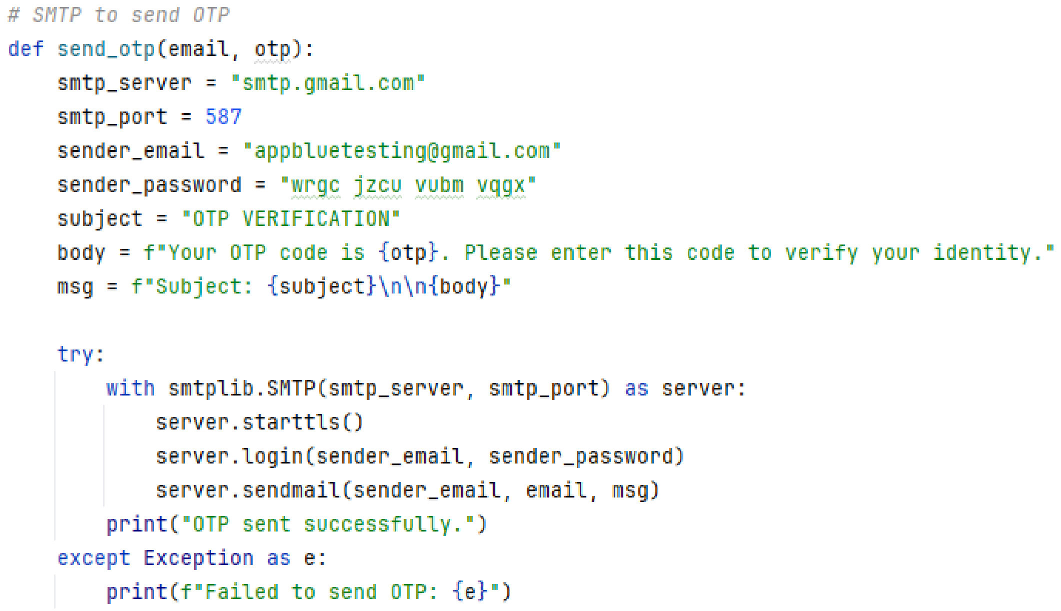


```python
# Email Validation
def is_valid_email(email):
    pattern = r'^[a-zA-Z0-9_.+-]+@[a-zA-Z0-9-]+\.(com|net|org|edu)$'
    valid_domains = ["gmail.com", "yahoo.com", "outlook.com"]

    if re.match(pattern, email):
        domain = email.split('@')[1]
        return domain in valid_domains
    return False
```

Purpose: Using a regular expression, the email format is validated

Usage: Before transmitting the OTP, it verifies that the email address supplied is valid and in the right format.

SMTP Setup and sending OTP through email



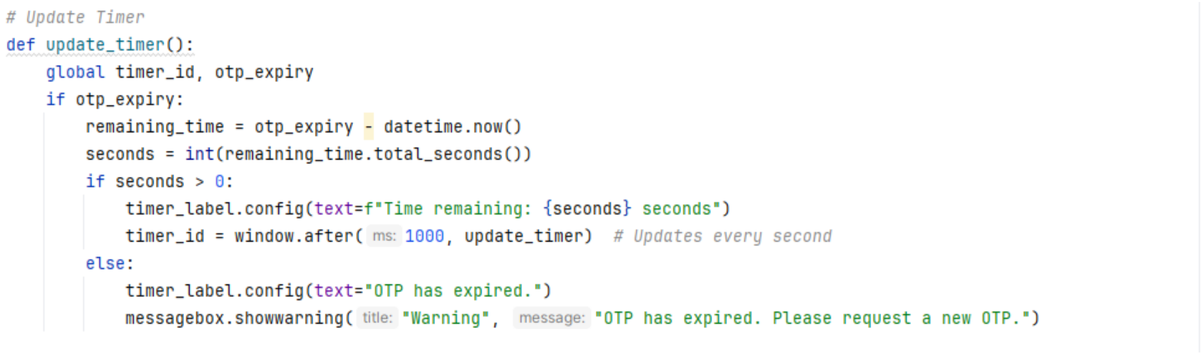
```python
# SMTP to send OTP
def send_otp(email, otp):
    smtp_server = "smtp.gmail.com"
    smtp_port = 587
    sender_email = "appbluetesting@gmail.com"
    sender_password = "wrgc jzcu vubm vqgx"
    subject = "OTP VERIFICATION"
    body = f"Your OTP code is {otp}. Please enter this code to verify your identity."
    msg = f"Subject: {subject}\n\n{body}"

    try:
        with smtplib.SMTP(smtp_server, smtp_port) as server:
            server.starttls()
            server.login(sender_email, sender_password)
            server.sendmail(sender_email, email, msg)
        print("OTP sent successfully.")
    except Exception as e:
        print(f"Failed to send OTP: {e}")
```

Purpose: Uses the SMTP protocol to deliver the generated OTP to the designated email address.

How it works: To send the OTP email, it establishes a connection with the Gmail SMTP server (smtp.gmail.com), launches TLS for encryption, and then logs in using the supplied credentials.

Usage: securely delivers the OTP to the user's email address for validation.

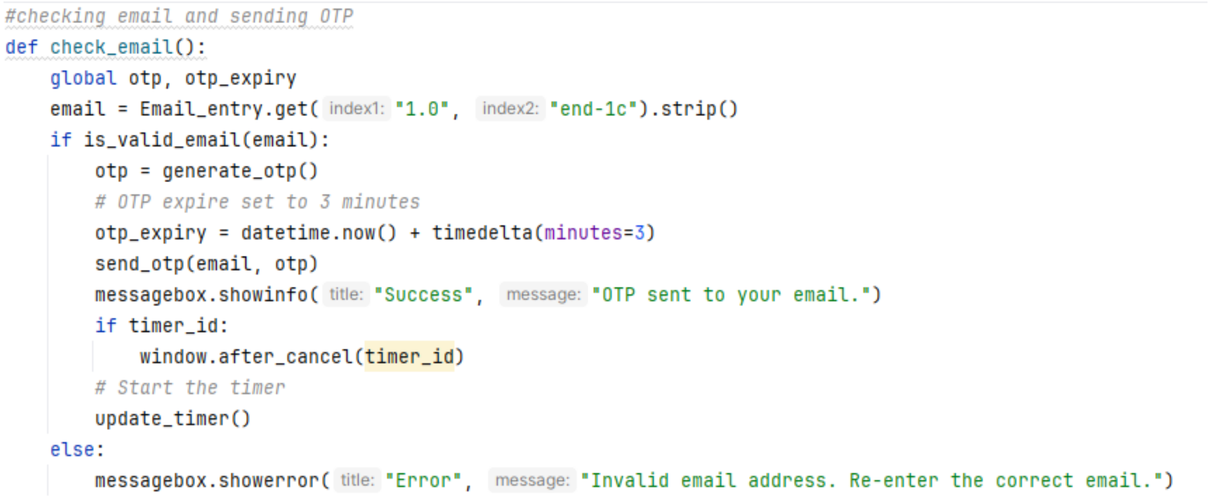
OTP Expiration Timer



```
# Update Timer
def update_timer():
    global timer_id, otp_expiry
    if otp_expiry:
        remaining_time = otp_expiry - datetime.now()
        seconds = int(remaining_time.total_seconds())
        if seconds > 0:
            timer_label.config(text=f"Time remaining: {seconds} seconds")
            timer_id = window.after( ms: 1000, update_timer)  # Updates every second
        else:
            timer_label.config(text="OTP has expired.")
            messagebox.showwarning( title: "Warning",  message: "OTP has expired. Please request a new OTP.")
```

Purpose: Updates the GUI's countdown timer to reflect the remaining time before the OTP expires.
How it Works: Using the window.after() function, it determines the difference between the current and otp_expiry times and updates the label once every second.
Usage: gives the user a visual indicator about how long the OTP will take to expire.

Checking email and Sending OTP (Everything together)

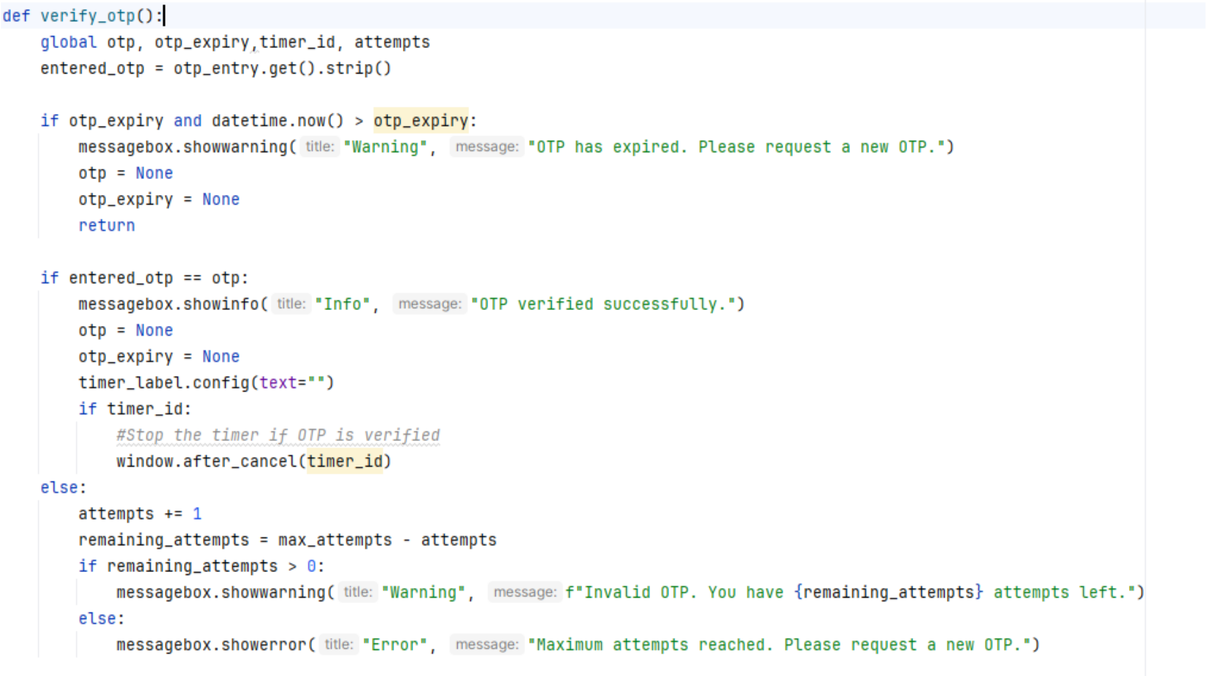


```
#checking email and sending OTP
def check_email():
    global otp, otp_expiry
    email = Email_entry.get( index1: "1.0",  index2: "end-1c").strip()
    if is_valid_email(email):
        otp = generate_otp()
        # OTP expire set to 3 minutes
        otp_expiry = datetime.now() + timedelta(minutes=3)
        send_otp(email, otp)
        messagebox.showinfo( title: "Success",  message: "OTP sent to your email.")
        if timer_id:
            window.after_cancel(timer_id)
        # Start the timer
        update_timer()
    else:
        messagebox.showerror( title: "Error",  message: "Invalid email address. Re-enter the correct email.")
```

Purpose: Sends the OTP and verifies the validity of the email entered.
How it works: Make use of is_valid_email() to verify the given email address.
Creates an OTP and sets the expiration time (three minutes from generation) if it is valid. initiates the expiration timer and sends the OTP via send_otp().

Usage: When a user enters their email, the process of generating and delivering an OTP is initiated.

OTP verification



```
def verify_otp():
    global otp, otp_expiry,timer_id, attempts
    entered_otp = otp_entry.get().strip()

    if otp_expiry and datetime.now() > otp_expiry:
        messagebox.showwarning( title: "Warning",  message: "OTP has expired. Please request a new OTP.")
        otp = None
        otp_expiry = None
        return

    if entered_otp == otp:
        messagebox.showinfo( title: "Info",  message: "OTP verified successfully.")
        otp = None
        otp_expiry = None
        timer_label.config(text="")
        if timer_id:
            #Stop the timer if OTP is verified
            window.after_cancel(timer_id)
    else:
        attempts += 1
        remaining_attempts = max_attempts - attempts
        if remaining_attempts > 0:
            messagebox.showwarning( title: "Warning",  message: f"Invalid OTP. You have {remaining_attempts} attempts left.")
        else:
            messagebox.showerror( title: "Error",  message: "Maximum attempts reached. Please request a new OTP.")
```

Purpose: Verifies the entered OTP against the OTP that was generated.

How it works: Compares the current time with otp_expiry to see if the OTP has expired.

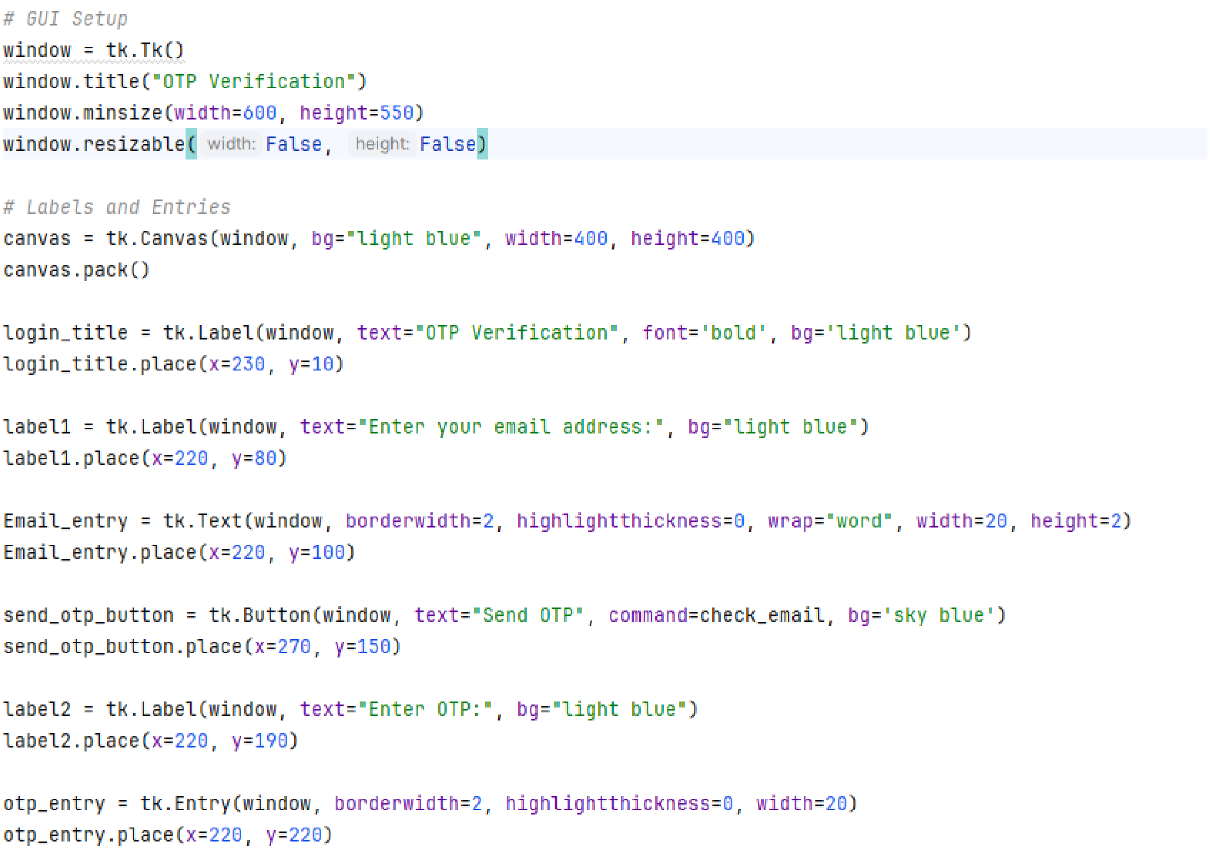
contrasts the produced OTP with the one that was entered.

It resets the timer and displays a success message if it is verified.

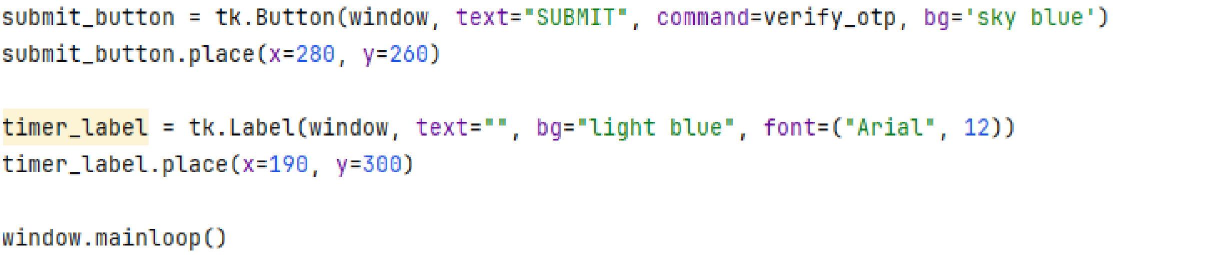
Increases the number of attempts and notifies the user of the number of attempts left if it is not validated.

Usage: To manage the verification of OTP and monitor unsuccessful attempts.

GUI setup



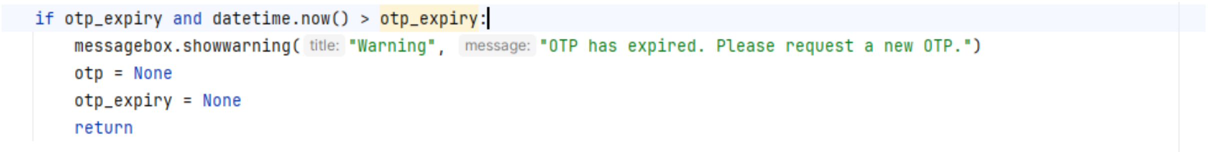
```
# GUI Setup
window = tk.Tk()
window.title("OTP Verification")
window.minsize(width=600, height=550)
window.resizable( width: False,  height: False)

# Labels and Entries
canvas = tk.Canvas(window, bg="light blue", width=400, height=400)
canvas.pack()

login_title = tk.Label(window, text="OTP Verification", font='bold', bg='light blue')
login_title.place(x=230, y=10)

label1 = tk.Label(window, text="Enter your email address:", bg="light blue")
label1.place(x=220, y=80)

Email_entry = tk.Text(window, borderwidth=2, highlightthickness=0, wrap="word", width=20, height=2)
Email_entry.place(x=220, y=100)

send_otp_button = tk.Button(window, text="Send OTP", command=check_email, bg='sky blue')
send_otp_button.place(x=270, y=150)

label2 = tk.Label(window, text="Enter OTP:", bg="light blue")
label2.place(x=220, y=190)

otp_entry = tk.Entry(window, borderwidth=2, highlightthickness=0, width=20)
otp_entry.place(x=220, y=220)
```



```
submit_button = tk.Button(window, text="SUBMIT", command=verify_otp, bg='sky blue')
submit_button.place(x=280, y=260)

timer_label = tk.Label(window, text="", bg="light blue", font=("Arial", 12))
timer_label.place(x=190, y=300)

window.mainloop()
```

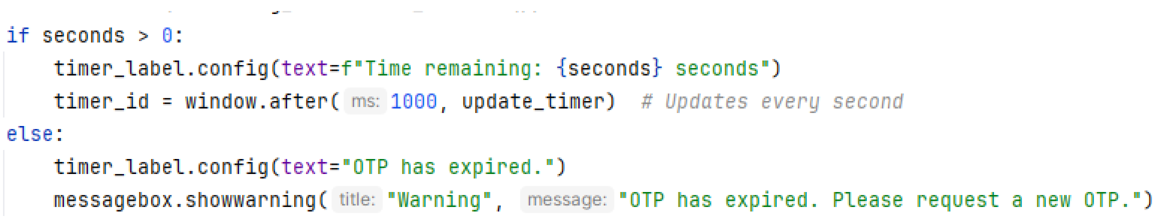
Purpose: Interface to interact test the OTP verification

Key Code Explanations



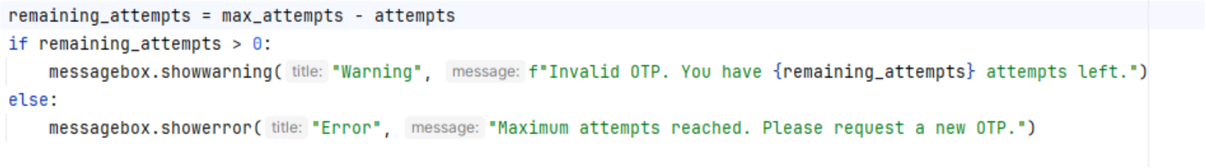
```
if otp_expiry and datetime.now() > otp_expiry:
    messagebox.showwarning( title: "Warning",  message: "OTP has expired. Please request a new OTP.")
    otp = None
    otp_expiry = None
    return
```

- The three-minute validity of the OTP is guaranteed by this block. The OTP is invalidated and the user is asked to request a new one if the current time exceeds the expiry time.



```
if seconds > 0:
    timer_label.config(text=f"Time remaining: {seconds} seconds")
    timer_id = window.after( ms: 1000, update_timer)  # Updates every second
else:
    timer_label.config(text="OTP has expired.")
    messagebox.showwarning( title: "Warning",  message: "OTP has expired. Please request a new OTP.")
```

- The window.after() function is used to update the timer label every second (1000 ms). It recursively calls itself to continue updating the countdown until the OTP expires.

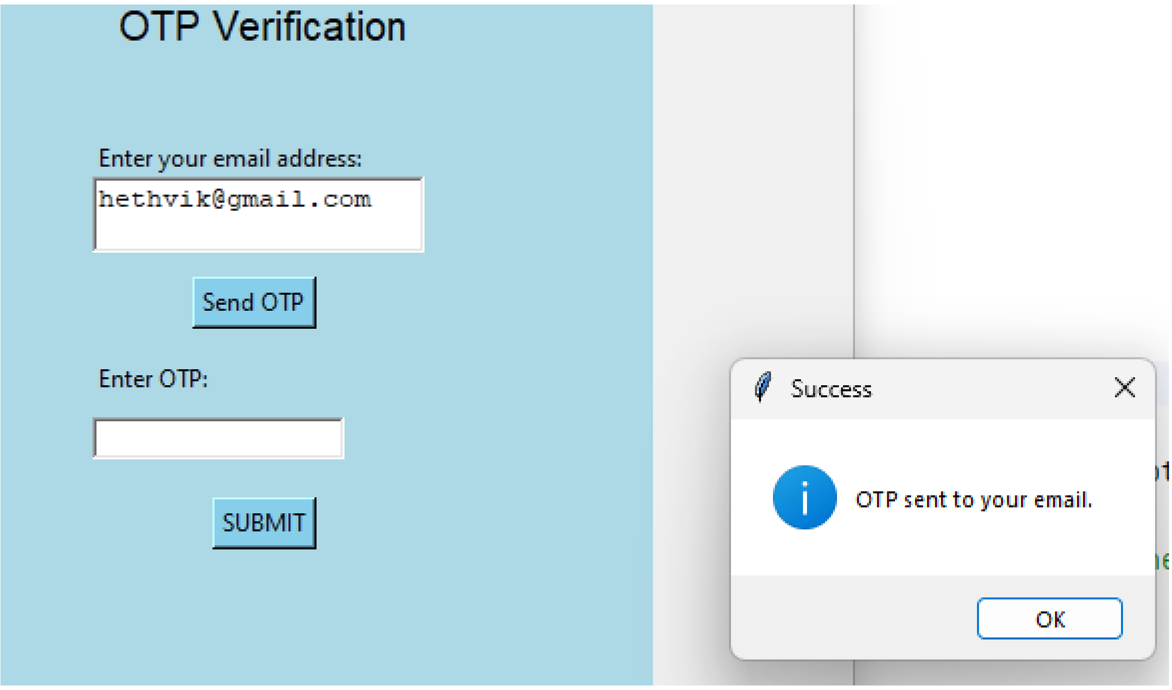


```
remaining_attempts = max_attempts - attempts
if remaining_attempts > 0:
    messagebox.showwarning( title: "Warning",  message: f"Invalid OTP. You have {remaining_attempts} attempts left.")
else:
    messagebox.showerror( title: "Error",  message: "Maximum attempts reached. Please request a new OTP.")
```

- This code limits the number of attempts for OTP verification to three. If the user exceeds the limit, they must request a new OTP.

Test Cases

Valid Email & Correct OTP:



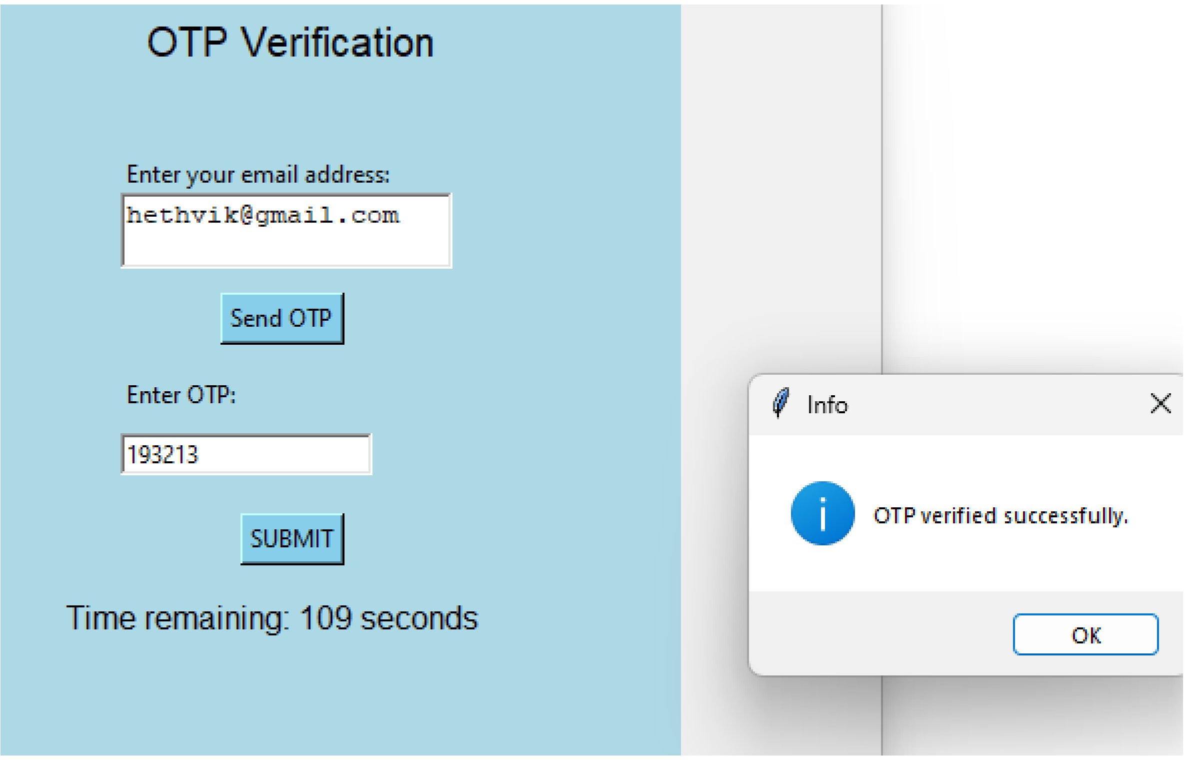
OTP Verification

Enter your email address: hethvik@gmail.com

Enter OTP:

Success

OTP sent to your email.



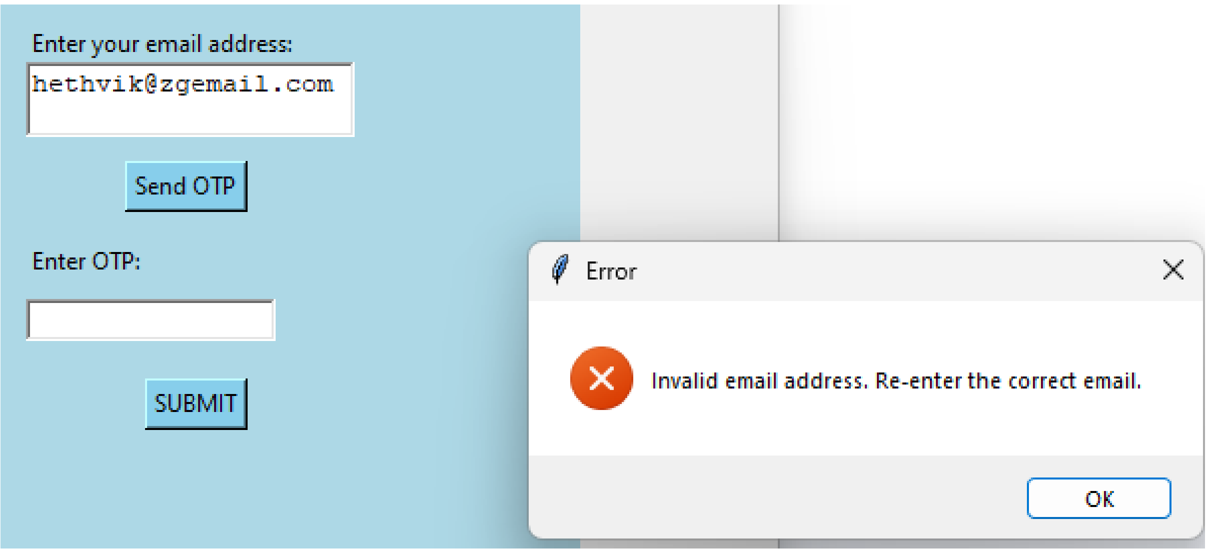
Enter your email address: hethvik@gmail.com

Enter OTP: 193213

Time remaining: 109 seconds

OTP verified successfully.

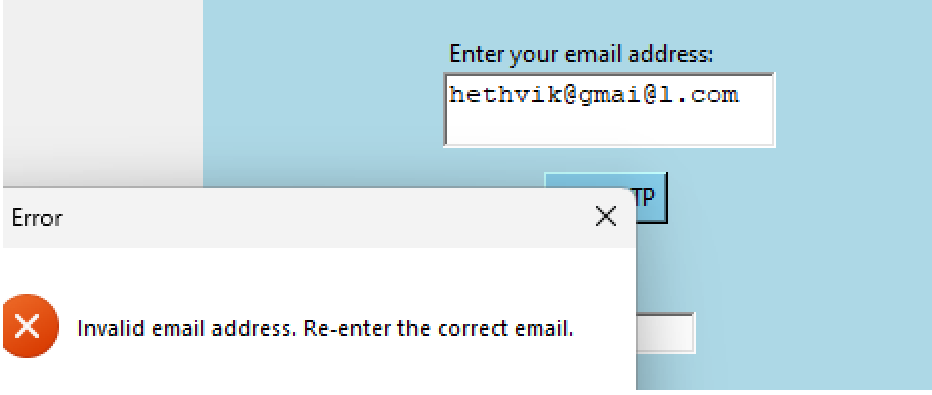
Invalid Email:



Enter your email address: hethvik@zgemail.com

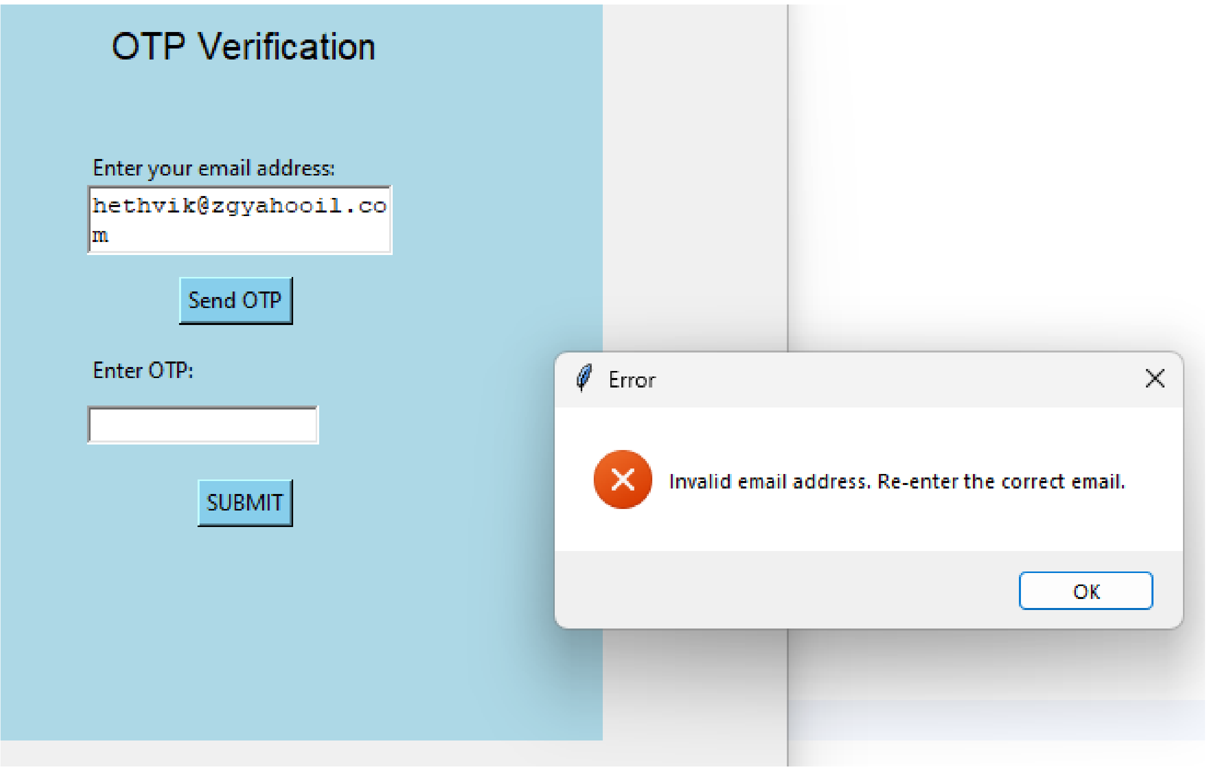
Enter OTP:

Invalid email address. Re-enter the correct email.



Enter your email address: hethvik@gmai@l.com

Invalid email address. Re-enter the correct email.



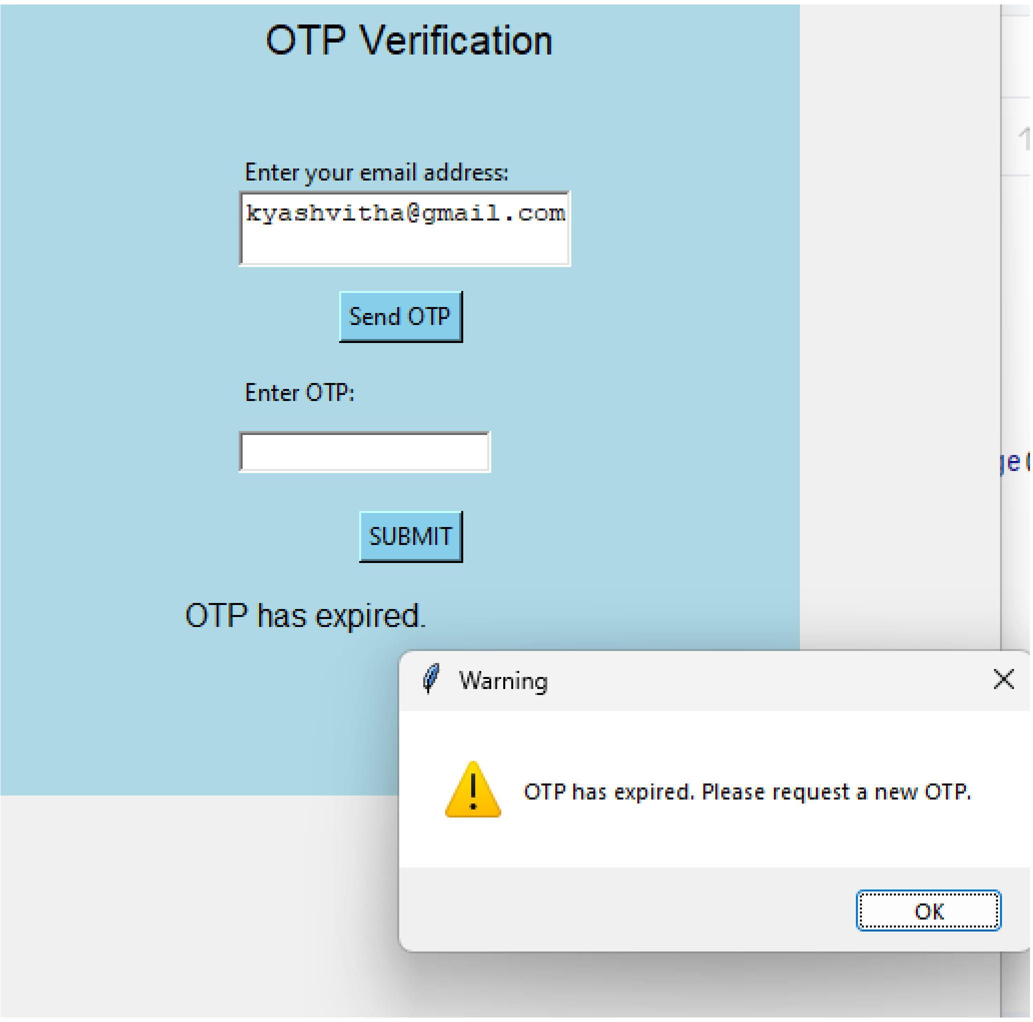
OTP Verification

Enter your email address:

hethvik@zgyahooil.com

Invalid email address. Re-enter the correct email.

OTP Expired:

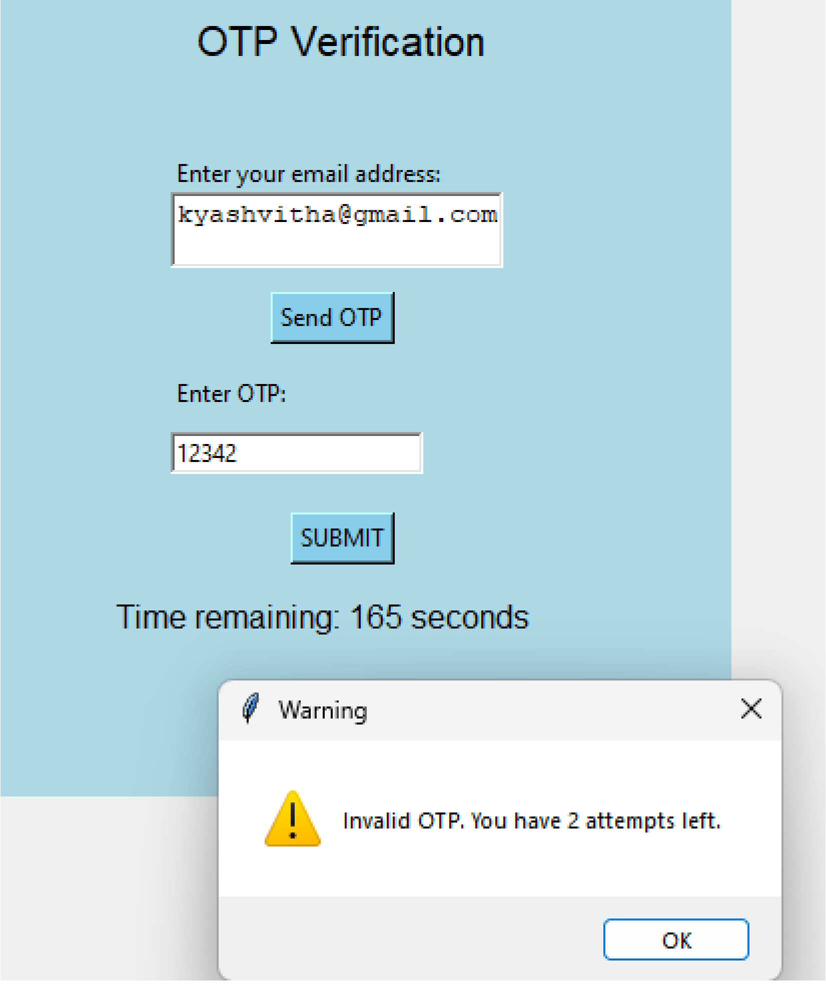


kyashvitha@gmail.com

OTP has expired.

OTP has expired. Please request a new OTP.

Incorrect OTP:



OTP Verification

Enter your email address:

kyashvitha@gmail.com

Enter OTP:

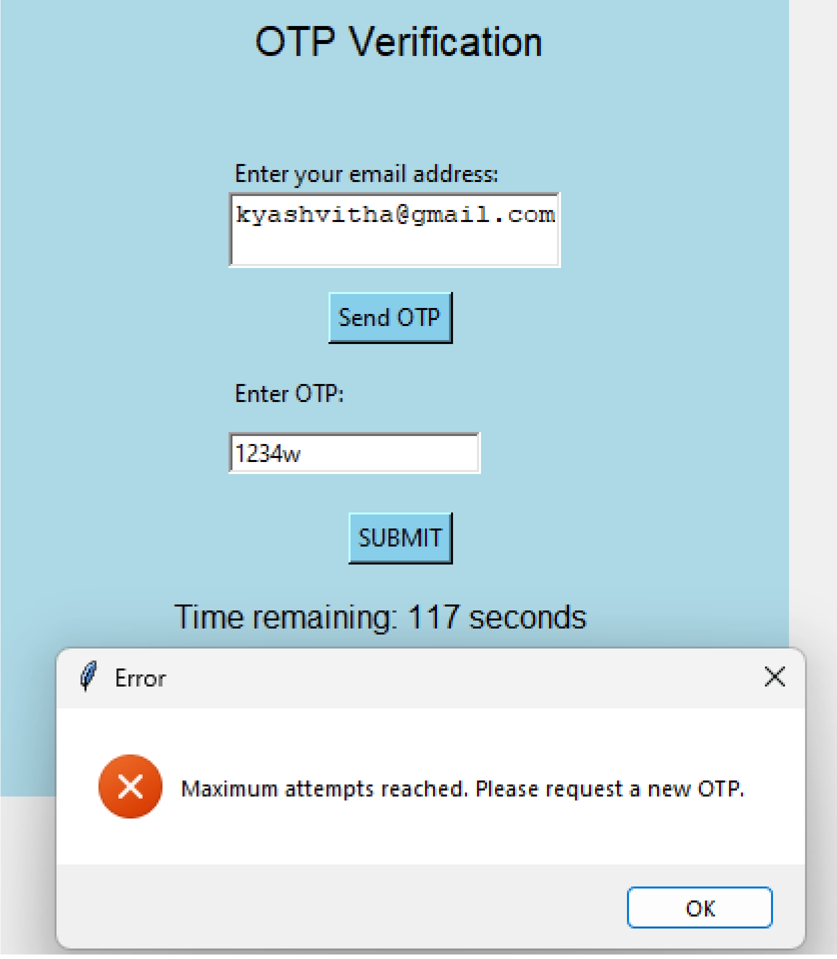
12342

Time remaining: 165 seconds

Warning

Invalid OTP. You have 2 attempts left.

Max Attempts Exceeded:



Enter your email address: kyashvitha@gmail.com

Enter OTP: 1234w

Time remaining: 117 seconds

Maximum attempts reached. Please request a new OTP.

Conclusion

The OTP system is explained in this documentation, along with the main code segments and functions of each. The test cases make sure the system works properly in a range of situations.