PROJECT

OTP VERIFICATION GRAPHICAL USER INTERFACE (GUI) USING TKINTER

AIM:

To create an OTP verification GUI by using tkinter library from python.

INTRODUCTION:

OTP Verification is the process of verifying a user by sending a unique password so that the user can be verified before completing a registration or payment process. Most of the time, we get an OTP when we make an online payment, or when we forget our password, or when creating an account on any online platform. Thus, the sole purpose of an OTP is to verify the identity of a user by sending a unique password.

STEPS TO CREATE OTP VERIFICATION GUI USING PYTHON:

We can easily create an application for the task of OTP verification using Python by following the steps mentioned below:

1. First, create a 4-digit random number.
2. Then store the number in a variable.

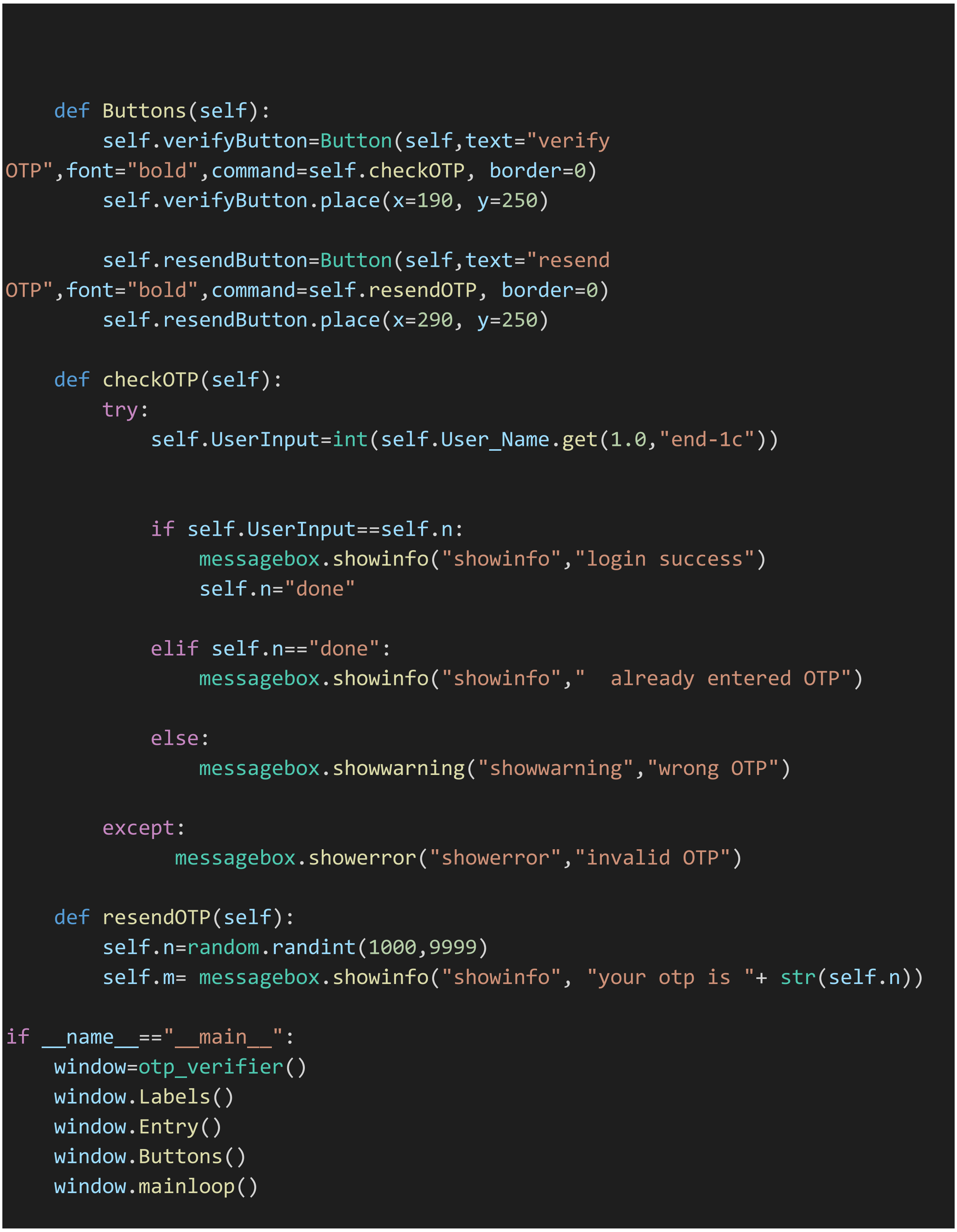
3. Then we need to write a program to create GUI.
4. We need to use OTP as a message.

LIBRARIES USED:

1. Import Tkinter
2. Import Random
3. From tkinter import messagebox

OTP verification GUI code:

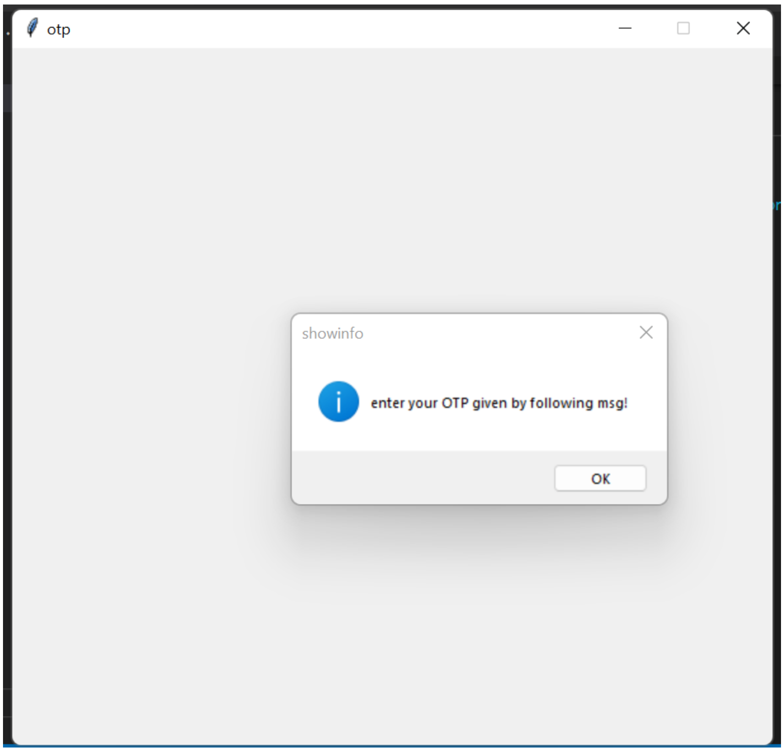
```
    from tkinter import *
from tkinter import messagebox
import random



class otp_verifier(Tk):
    def __init__(self):
        super().__init__()
        self.geometry("600x550")
        self.resizable(False,False)
        self.title('otp')
        self.n=random.randint(1000,9999)
        self.m=messagebox.showinfo("showinfo", "enter your OTP given by
following msg!")


    def Labels(self):
        self.c =Canvas(self,bg="white",width=400,height=200)
        self.c.place(x=100,y=90)

        self.Login_Title=Label(self, text="OTP Verification", font= "bold,
20", bg="white")
        self.Login_Title.place(x=190, y=93)

    def Entry(self):
        self.User_Name= Text(self, borderwidth=2, wrap="word",width=29,
height=2)
        self.m=messagebox.showinfo("showinfo", "your otp is "+ str(self.n))
        self.User_Name.place(x=190,y=160)
```



```
    def Buttons(self):
        self.verifyButton=Button(self,text="verify
OTP",font="bold",command=self.checkOTP, border=0)
        self.verifyButton.place(x=190, y=250)

        self.resendButton=Button(self,text="resend
OTP",font="bold",command=self.resendOTP, border=0)
        self.resendButton.place(x=290, y=250)

    def checkOTP(self):
        try:
            self.UserInput=int(self.User_Name.get(1.0,"end-1c"))


            if self.UserInput==self.n:
                messagebox.showinfo("showinfo","login success")
                self.n="done"

            elif self.n=="done":
                messagebox.showinfo("showinfo","  already entered OTP")

            else:
                messagebox.showwarning("showwarning","wrong OTP")

        except:
              messagebox.showerror("showerror","invalid OTP")

    def resendOTP(self):
        self.n=random.randint(1000,9999)
        self.m= messagebox.showinfo("showinfo", "your otp is "+ str(self.n))

if __name__=="__main__":
    window=otp_verifier()
    window.Labels()
    window.Entry()
    window.Buttons()
    window.mainloop()
```

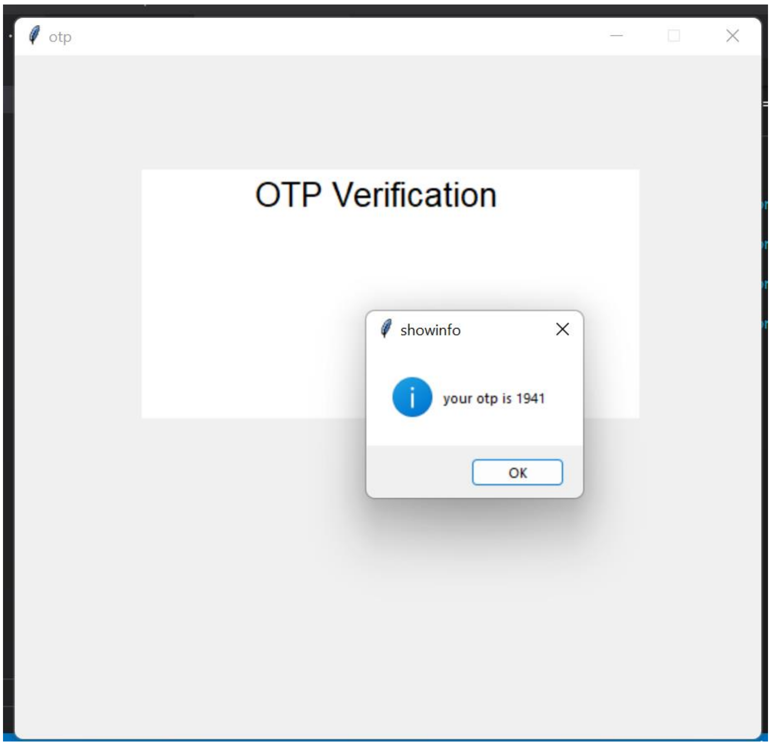
OUTPUT:

1.



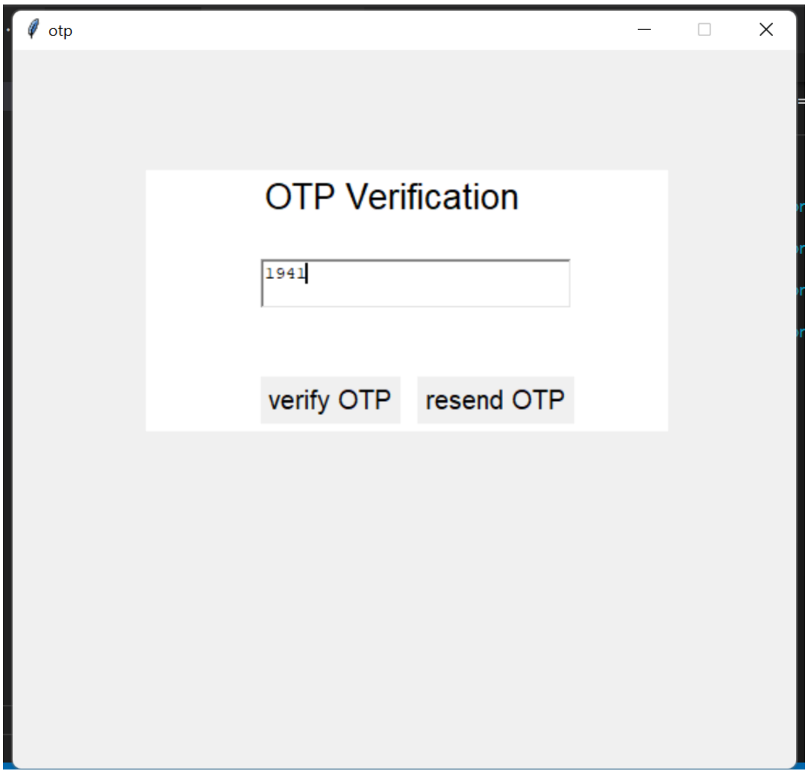
enter your OTP given by following msg!

2.



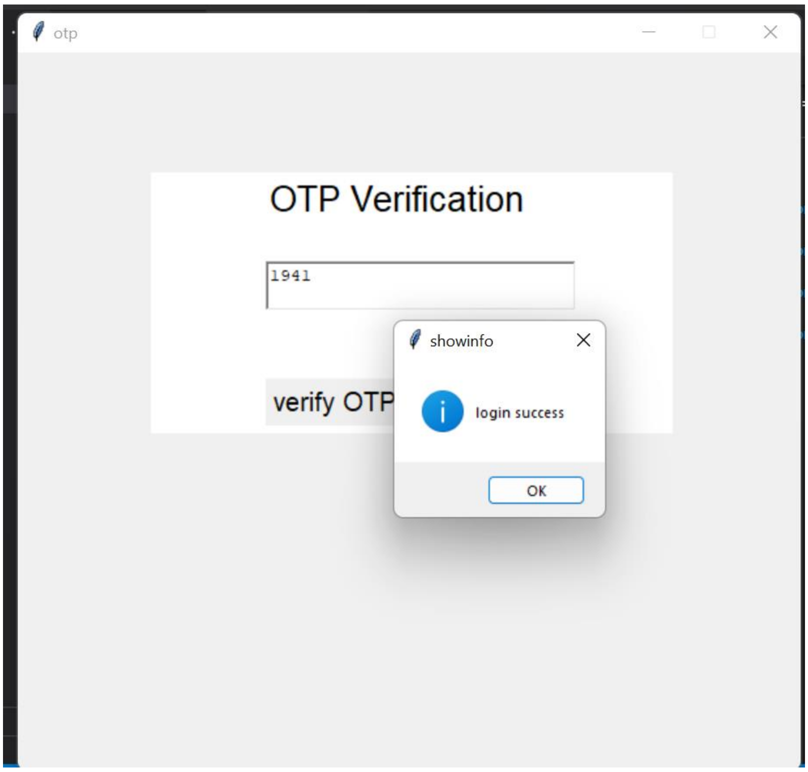
your otp is 1941

3.



1941

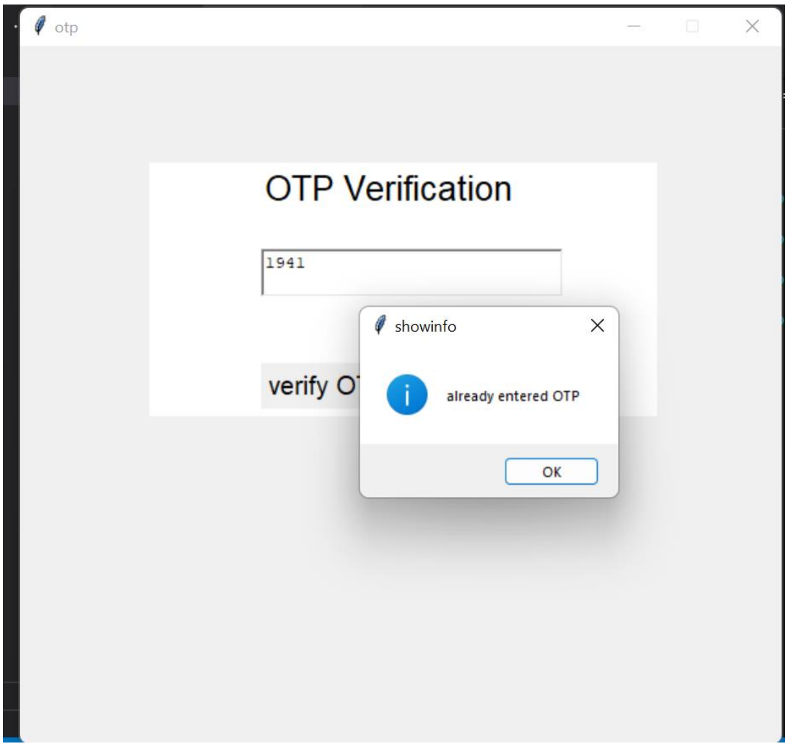
4.



1941

login success

5.

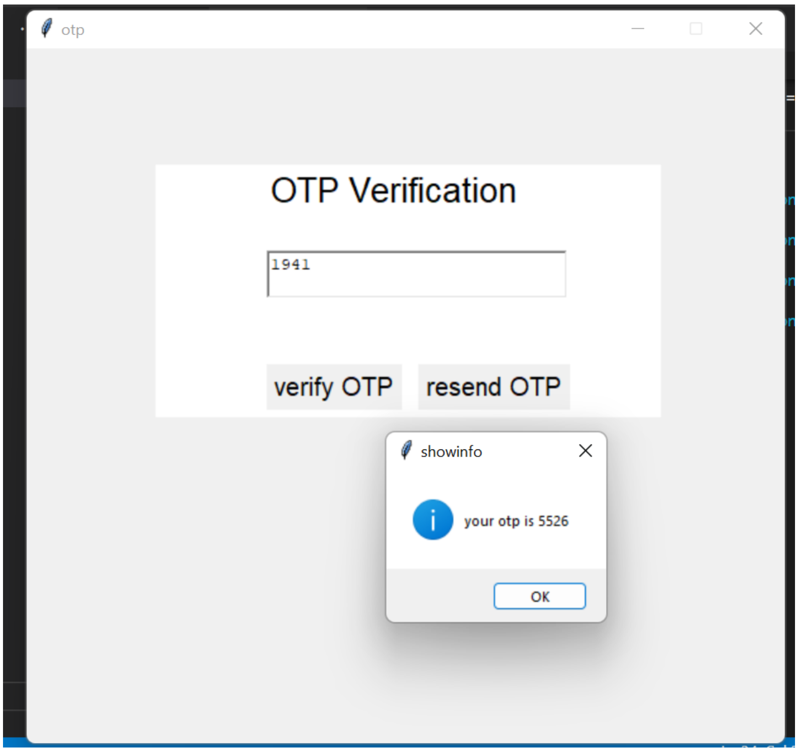


1941

showinfo

already entered OTP

6.

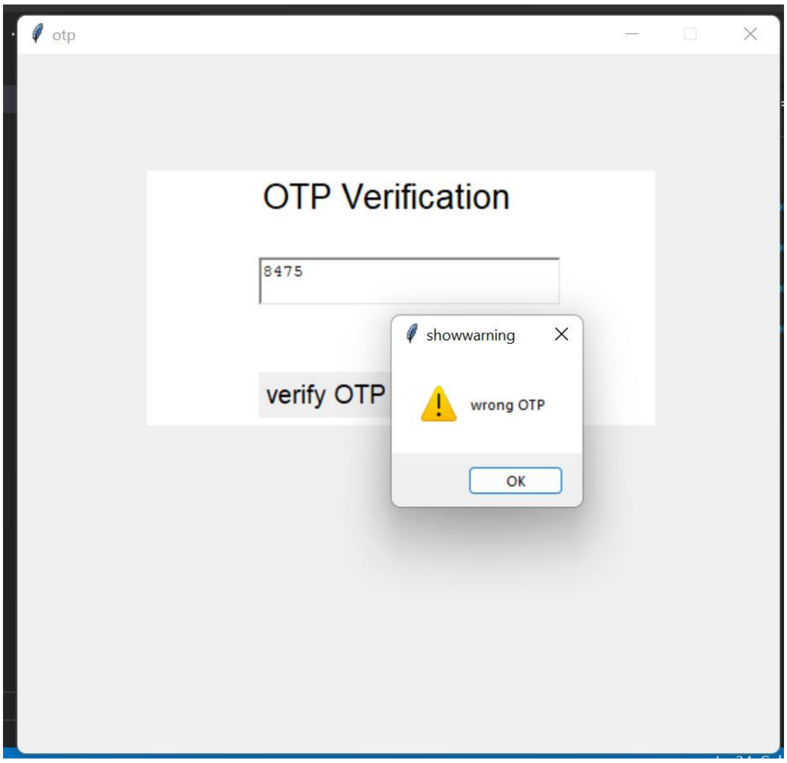


1941

showinfo

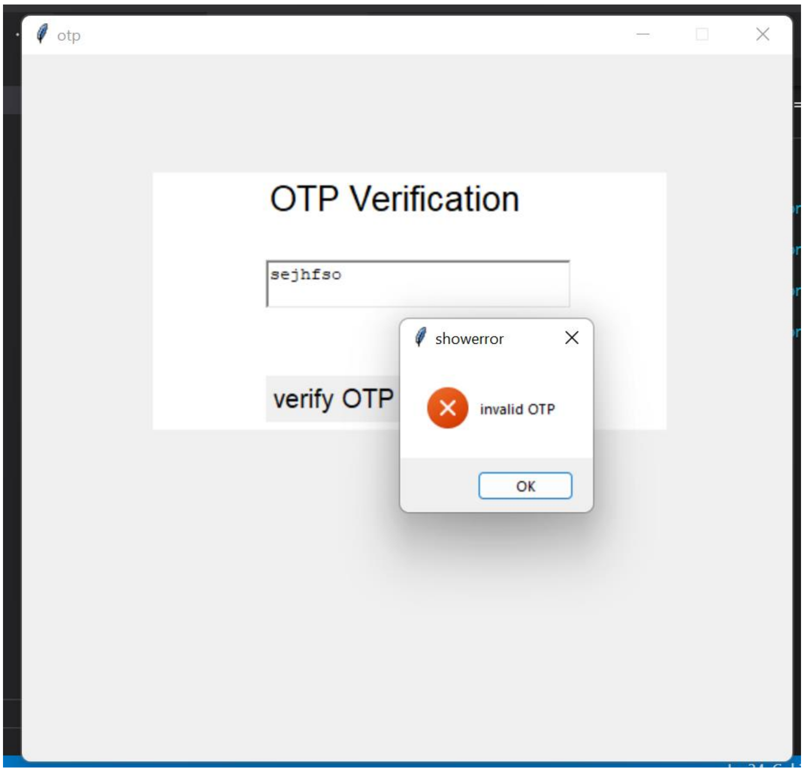
your otp is 5526

7.



wrong OTP

8.



invalid OTP

CONCLUSION:

Hope this will clear the basic idea on how the OTP verification process works and how it can be implemented easily in python.

THANK YOU!!!

~ BY

SWETHA.V