Simple Maths CAPTCHA Using Python

Final Report

by

Name	Registration No.	Roll No.
Saurabh	11709236	*26*

School of Computer Science and Engineering

Lovely Professional University, Jalandhar

(2018-19)

Submitted To:-
Ms. Chavi Kapoor
(Python Programming Language)
Lovely Professional University

ACKNOWLEDGEMENT

It gives me immense pleasure to present the Project on “Simple Maths-Based Captcha” by using Python. I feel privileged to have got this all along the completion of our “project”. I thank our teacher, Ms. Chavi Kapoor, for giving me an opportunity to do this Project and providing me all the support and guidance due to which this project is Completed on time.

I would like to extend sincere regards to our parents and my friends for their encouragement and for their timely support and guidance.

Saurabh Upadhyay

11714489

TABLES OF CONTENTS

INTRODUCTION

A CAPTCHA is a program that protects websites against bots by generating and grading tests that humans can pass but current computer programs cannot. For example, humans can read distorted text, but current programs can't.

The term CAPTCHA (for Completely Automated Public Turing Test to Tell Computers and Humans Apart) was coined in 2000 by Luis von Ahn, Manuel Blum, Nicholas Hopper and John Langford of Carnegie Mellon University.

Applications of CAPTCHAs

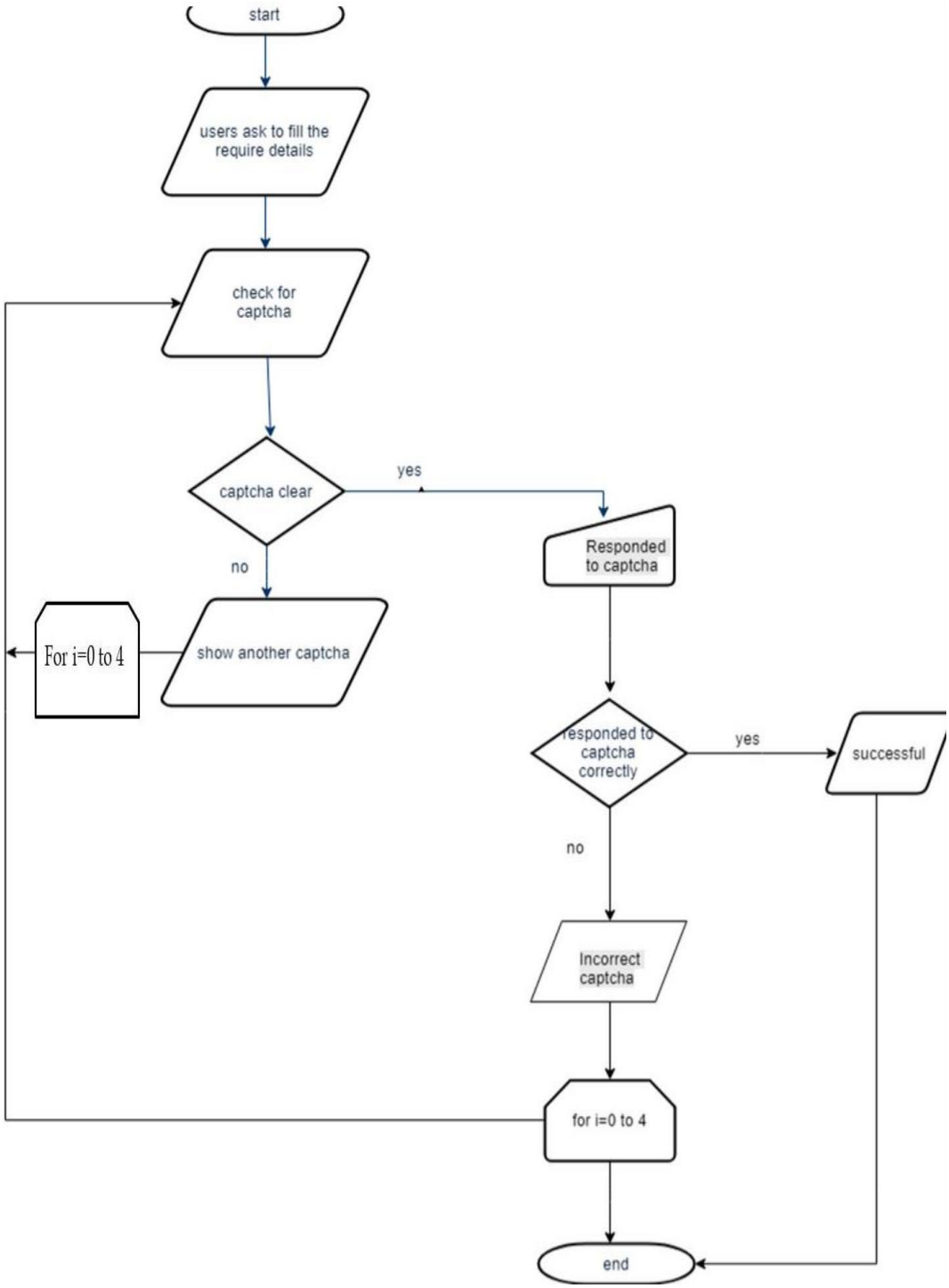
CAPTCHAs have several applications for practical security, including (but not limited to):

- **Preventing Comment Spam in Blogs:** Most bloggers are familiar with programs that submit bogus comments, usually for the purpose of raising search engine ranks of some website (e.g., "buy penny stocks here"). This is called comment spam. By using a CAPTCHA, only humans can enter comments on a blog. There is no need to make users sign up before they enter a comment, and no legitimate comments are ever lost!
- **Protecting Website Registration:** Several companies offer free email services. Up until a few years ago, most of these services suffered from a specific type of attack: "bots" that would sign up for thousands of email accounts every minute. The solution to this problem was to use CAPTCHAs to ensure that only humans obtain free accounts. In general, free services should be protected with a CAPTCHA in order to prevent abuse by automated scripts.
- **Protecting Email Addresses from Scrapers.** Spammers crawl the Web in search of email addresses posted in clear text. CAPTCHAs provide an effective mechanism to hide your email address from Web scrapers. The idea is to require users to solve a CAPTCHA before showing your email address.
- **Online Polls:** In November 1999, **slashdot.org** released an online poll asking which was the best graduate school in computer science (a dangerous question to ask over the web!). As is the case with most online polls, IP addresses of voters were recorded in order to prevent single users from voting more than once. However, students at Carnegie Mellon found a way to stuff the ballots using programs that voted for CMU thousands of times. CMU's score started growing rapidly. The next day, students at MIT wrote their own program and the poll became a contest between

voting "bots." MIT finished with 21,156 votes, Carnegie Mellon with 21,032 and every other school with less than 1,000. Can the result of any online poll be trusted? Not unless the poll ensures that only humans can vote.

- **Preventing Dictionary Attacks.** CAPTCHAs can also be used to prevent dictionary attacks in password systems. The idea is simple: prevent a computer from being able to iterate through the entire space of passwords by requiring it to solve a CAPTCHA after a certain number of unsuccessful logins. This is better than the classic approach of locking an account after a sequence of unsuccessful logins, since doing so allows an attacker to lock accounts at will.
- **Search Engine Bots.** It is sometimes desirable to keep webpages unindexed to prevent others from finding them easily. There is an html tag to prevent search engine bots from reading web pages. The tag, however, doesn't guarantee that bots won't read a web page; it only serves to say "no bots, please." Search engine bots, since they usually belong to large companies, respect web pages that don't want to allow them in. However, in order to truly guarantee that bots won't enter a web site, CAPTCHAs are needed.
- **Worms and Spam.** CAPTCHAs also offer a plausible solution against email worms and spam: "I will only accept an email if I know there is a human behind the other computer." A few companies are already marketing this idea.

PROJECT DESCRIPTION



MODULES INVOLVED IN THE PROJECT ALONG WITH WORK DIVISION

The math-based captcha project using python has various modules which is divided among the group as follows:

- **Coding**
- **GUI**
- **Coding and Report**

CODE FOR THE PROGRAM

```
from tkinter import*
import random
from tkinter import ttk


def quit():
    global root


    root.quit()


def end():
    global top
    top.end()
def nain_des():
    B3.pack()
def main_des():
    B8.pack()
def gain_des():
    B4.pack()
def main():
    #main_des()
    #B8.destroy()
    #B8.pack(side=LEFT)
    B3.destroy()
```

```
    B4.destroy()

    canvas.create_image(0,0,anchor=NW,image=img3)
    mtext=ment.get()
    global c
    if mtext=='4':
        B8.destroy()
        top=Tk()
        mlabel2=Label(top,text='succesfull',bd=15).pack()
        Button(root, text="Quit", command=root.destroy).pack()
        Button(top, text="end", command=top.destroy).pack()
        top.mainloop()
    elif (c==5):
        root.destroy()
        print("You are out of tries")
    else:
        B8.destroy()
        #ran()
        #callback()
        #B3.destroy()
        #B4.destroy()
        c=c+1
        mlabel2=Label(root,text='oops!retry').pack()
    #B8.destroy()
def nain():
    #nain_des()
```

```
    #B3.destroy()
    #B3.pack(side=LEFT)
    #B.pack(side=LEFT)
    B8.destroy()
    B4.destroy()
    canvas.create_image(0,0,anchor=NW,image=img2)
    mtext=ment.get()
    global c
    if mtext=='15':
        top=Tk()
        mlabel2=Label(top,text='succesfull',bd=15).pack()
        B3.destroy()
        Button(root, text="Quit", command=root.destroy).pack()
        Button(top, text="end", command=top.destroy).pack()
        top.mainloop()
    elif (c==5):
        root.destroy()
        print("You are out of tries")
    else:


        #callback()
        #B8.destroy()
        B3.destroy()
        #ran()
        c=c+1
```

```
        mlabel2=Label(root,text='oops!retry').pack()
    #refresh()
  #B3.destroy()
def gain():
  #gain_des()
  #B4.destroy()
  #B4.pack(side=LEFT)
  #B.pack(side=LEFT)
  B8.destroy()
  B3.destroy()
  canvas.create_image(0,0,anchor=NW,image=img4)
  mtext=ment.get()
  global c
  if mtext=='14':
    top=Tk()
        mlabel2=Label(top,text='succesfull',bd=15).pack()
    B4.destroy()
    Button(root, text="Quit", command=root.destroy).pack()
    Button(top, text="end", command=top.destroy).pack()
    top.mainloop()
  elif (c==5):
    root.destroy()
    print("You are out of tries")
  else:
```

```
        B4.destroy()
        #callback()
        #B8.destroy()
        #B3.destroy()
        c=c+1
        mlabel2=Label(root,text='oops!retry').pack()
       # B4.destroy()
        #ran()


def tain():
    mtext=ment.get()
    canvas.create_image(0,0,anchor=NW,image=img5)
    global c
    if mtext=='7':
        top=Tk()
        mlabel2=Label(top,text='succesfull',bd=15).pack()
        Button(root, text="Quit", command=root.destroy).pack()
        Button(top, text="end", command=top.destroy).pack()
        top.mainloop()
    elif (c==5):
        root.destroy()
        print("You are out of tries")
    else:
        B5.destroy()
        c=c+1
```

```
        mlabel2=Label(root,text='oops!retry').pack()
def hain():
    mtext=ment.get()
    canvas.create_image(0,0,anchor=NW,image=img6)
    global c
    if mtext=='5':
        top=Tk()
        mlabel2=Label(top,text='succesfull',bd=15).pack()
        Button(root, text="Quit", command=root.destroy).pack()
        Button(top, text="end", command=top.destroy).pack()
        top.mainloop()
    elif (c==5):
        root.destroy()
        print("You are out of tries")
    else:
        B6.destroy()
        c=c+1
        mlabel2=Label(root,text='oops!retry').pack()
def fain():
    mtext=ment.get()
    canvas.create_image(0,0,anchor=NW,image=img7)
    global c
    if mtext=='132':
        top=Tk()
        mlabel2=Label(top,text='succesfull',bd=15).pack()
```

```
        Button(top, text="end", command=top.destroy).pack()
        Button(root, text="Quit", command=root.destroy).pack()
        top.mainloop()
    elif (c==5):
        root.destroy()
        print("You are out of tries")
    else:
        B7.destroy()
        c=c+1
        mlabel2=Label(root,text='oops!retry').pack()
def cain():
    mtext=ment.get()
    canvas.create_image(0,0,anchor=NW,image=img8)
    global c
    if mtext=='13':
        top=Tk()
        mlabel2=Label(top,text='succesfull',bd=15).pack()
        Button(top, text="end", command=top.destroy).pack()
        Button(root, text="Quit", command=root.destroy).pack()
        top.mainloop()
    elif (c==4):
        root.destroy()
        print("You are out of tries")
    else:
        B11.destroy()
```

```
        c=c+1
        mlabel2=Label(root,text='oops!retry').pack()
def dain():
    mtext=ment.get()
    canvas.create_image(0,0,anchor=NW,image=img9)
    global c
    if mtext=='79':
        top=Tk()
        mlabel2=Label(top,text='succesfull',bd=15).pack()
        Button(top, text="end", command=top.destroy).pack()
        Button(root, text="Quit", command=root.destroy).pack()
        top.mainloop()
    elif (c==5):
        root.destroy()
        print("You are out of tries")
    else:
        B9.destroy()
        c=c+1
        mlabel2=Label(root,text='oops!retry').pack()
def eain():
    mtext=ment.get()
    canvas.create_image(0,0,anchor=NW,image=img10)
    global c
    if mtext=='72':
        top=Tk()
```

```
        mlabel2=Label(top,text='succesfull',bd=15).pack()
    Button(top, text="end", command=top.destroy).pack()
    Button(root, text="Quit", command=root.destroy).pack()
    top.mainloop()
  elif (c==5):
    root.destroy()
    print("You are out of tries")
  else:
    B10.destroy()
    c=c+1
        mlabel2=Label(root,text='oops!retry').pack()
def ran():
  B.destroy()
  rand=(random.randint(1,9))
  print(rand)

  if rand==1:
    nain()

    B3=Button(root, text ="SUBMIT", command =nain,fg="red" ,bg="WHITE")
    B3.place(x=280,y=380)



        #B8.pack(side=LEFT)
        #B8.destroy()
```

```
        #B4.destroy()
    elif rand==2:
        B8=Button(root, text ="SUBMIT", command =main,fg="red" ,bg="WHITE")
        B8.place(x=280,y=380)

        main()
        #B3.destroy()
        #B4.destroy()
    elif rand==3:
        B5=Button(root, text ="SUBMIT", command =tain,fg="red" ,bg="WHITE")
        B5.place(x=280,y=380)

        tain()
        #B3.destroy()
        #B4.destroy()
    elif rand==4:
        B6=Button(root, text ="SUBMIT", command =hain,fg="red" ,bg="WHITE")
        B6.place(x=280,y=380)

        hain()
        #B3.destroy()
        #B4.destroy()
    elif rand==5:
        B7=Button(root, text ="SUBMIT", command =fain,fg="red" ,bg="WHITE")
        B7.place(x=280,y=380)
```

```
    fain()
        #B3.destroy()
        #B4.destroy()
  elif rand==6:
    B11=Button(root, text ="SUBMIT", command =cain,fg="red" ,bg="WHITE")
    B11.place(x=280,y=380)

    cain()
        #B3.destroy()
        #B4.destroy()
  elif rand==7:
    B9=Button(root, text ="SUBMIT", command =dain,fg="red" ,bg="WHITE")
    B9.place(x=280,y=380)


    dain()
        #B3.destroy()
        #B4.destroy()
  elif rand==8:
    B10=Button(root, text ="SUBMIT", command =eain,fg="red" ,bg="WHITE")
    B10.place(x=280,y=380)

    eain()
        #B3.destroy()
```

```
        #B4.destroy()


    else:
        #B8.destroy()
        #B3.destroy()
      B4=Button(root, text ="SUBMIT", command =gain,fg="red" ,bg="WHITE")
      B4.place(x=280,y=380)


      gain()




def callback():
    mtext=ment.get()
    if mtext=='89':
        top=Tk()
        B.destroy()
        mlabel2=Label(top,text='succesfull',bd=15).pack()
        Button(top, text="end", command=top.destroy).pack()
        Button(root, text="Quit", command=root.destroy).pack()
        top.mainloop()
    else:
        B.destroy()
        ran()
```

```
def refresh():
    #B2.destroy()
    #B8.destroy()
    #B4.destroy()
    #B3.destroy()
    ran()


root=Tk()
ment=StringVar()

root.title('maths captcha')
c=0


L1 = Label(root, text="type your answer here -> ")
L1.pack( side = LEFT)
E1 = Entry(root,textvariable=ment,bd =5)
E1.insert(0,'')
E1.pack(side = LEFT)
B =Button(root, text ="SUBMIT", command =callback,fg="red" ,bg="WHITE")
B.pack(side=LEFT)
B2=Button(root, text ="REFRESH", command =refresh,fg="red" ,bg="WHITE")
B2.pack(side=LEFT)
B8=Button(root, text ="SUBMIT", command =main,fg="red" ,bg="WHITE")
```

B3=Button(root, text ="SUBMIT", command =nain,fg="red" ,bg="WHITE")

B4=Button(root, text ="SUBMIT", command =gain,fg="red" ,bg="WHITE")

B5=Button(root, text ="SUBMIT", command =tain,fg="red" ,bg="WHITE")

B6=Button(root, text ="SUBMIT", command =hain,fg="red" ,bg="WHITE")

B7=Button(root, text ="SUBMIT", command =fain,fg="red" ,bg="WHITE")

B9=Button(root, text ="SUBMIT", command =dain,fg="red" ,bg="WHITE")

B10=Button(root, text ="SUBMIT", command =eain,fg="red" ,bg="WHITE")

B11=Button(root, text ="SUBMIT", command =cain,fg="red" ,bg="WHITE")

canvas=Canvas(root,width=500,height=500)

canvas.pack()

img1=PhotoImage(file='file:///C:/Users/Saurabh .LAPTOP-5IILUD98/Desktop/Python Project/img1.jpg')

img2=PhotoImage(file='file:///C:/Users/Saurabh .LAPTOP-5IILUD98/Desktop/Python Project/img2.jpg')

img3=PhotoImage(file='file:///C:/Users/Saurabh .LAPTOP-5IILUD98/Desktop/Python Project/img3.jpg')

img4=PhotoImage(file='file:///C:/Users/Saurabh .LAPTOP-5IILUD98/Desktop/Python Project/img4.jpg')

img5=PhotoImage(file='file:///C:/Users/Saurabh .LAPTOP-5IILUD98/Desktop/Python Project/img5.jpg')

img6=PhotoImage(file='file:///C:/Users/Saurabh .LAPTOP-5IILUD98/Desktop/Python Project/img6.jpg')

img7=PhotoImage(file='file:///C:/Users/Saurabh .LAPTOP-5IILUD98/Desktop/Python Project/img7.jpg')

img8=PhotoImage(file='file:///C:/Users/Saurabh .LAPTOP-5IILUD98/Desktop/Python Project/img8.jpg')

img9=PhotoImage(file='file:///C:/Users/Saurabh .LAPTOP-5IILUD98/Desktop/Python Project/img9.jpg')

img10=PhotoImage(file='file:///C:/Users/Saurabh .LAPTOP-5IILUD98/Desktop/Python Project/img10.jpg')

canvas.create_image(0,0,anchor=NW,image=img1)

root.mainloop()

SCREENSHOTS

1) On clicking on RUN , we will get the output ,as given below(fig-1), depicting a maths-based captcha and ask the user to answer this captcha to access the next window.

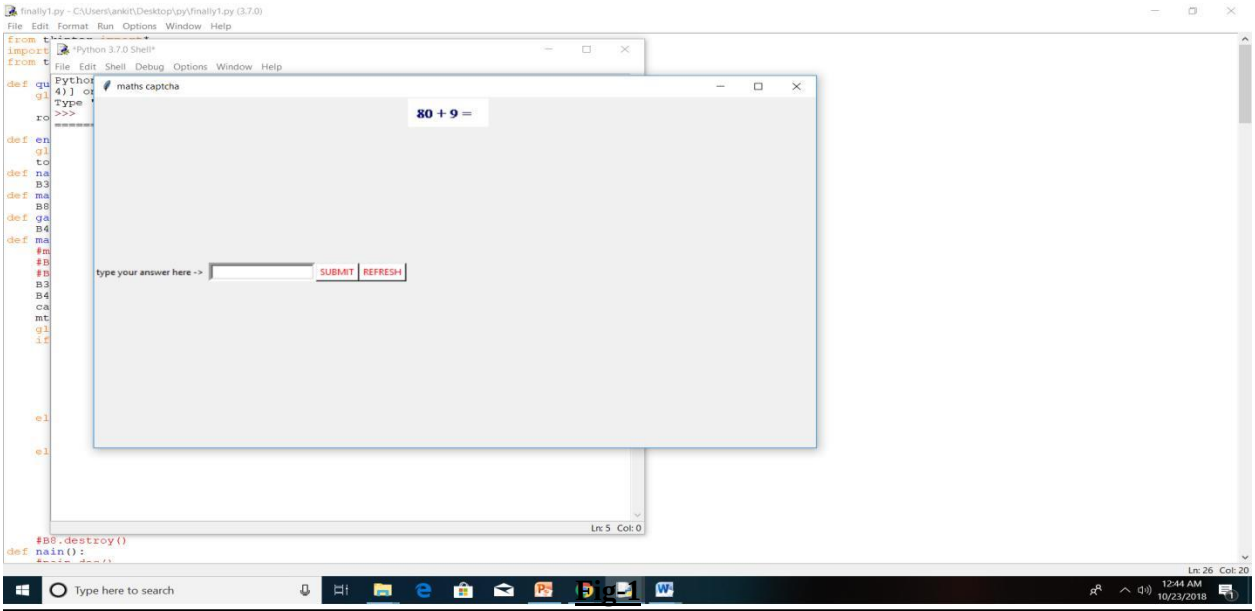
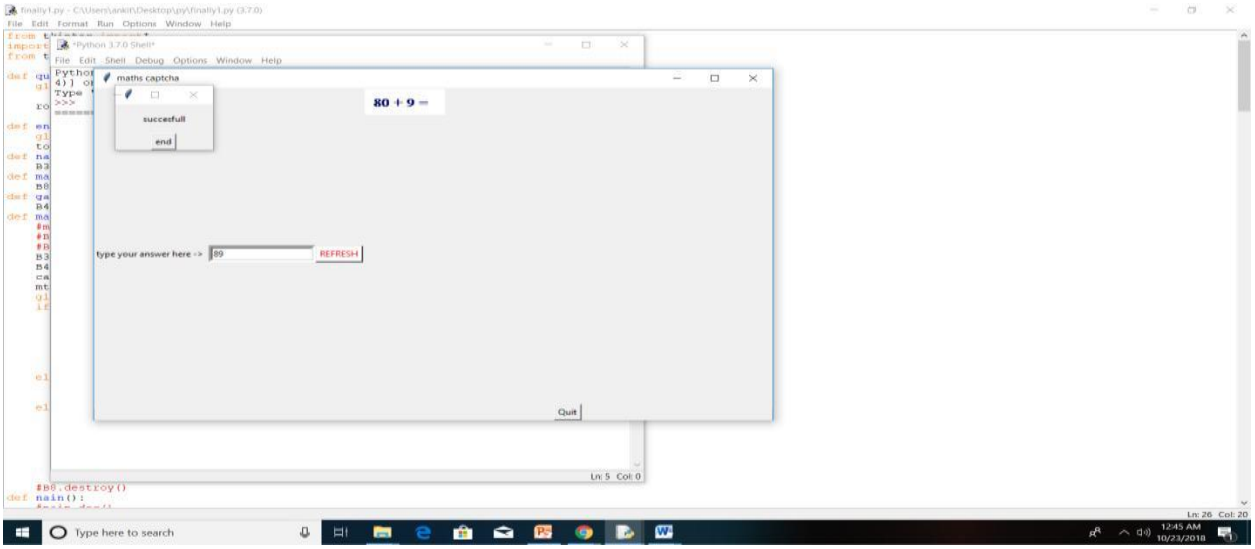


Fig-1

If the user answered the captcha correctly, then the next window will pop up and then user can easily access that window.



Fig(2)

As shown in the fig-2, the new window having the accessibility of button like “quit” and “end”, which is same function we have use in program just for showing it is working properly, can be able to access now for the user after submitting correct answer for the given maths-based captcha.

2.) In the case if Captcha seems not clear to the user, then the user can click on the refresh button and the user will get to see the new captcha (fig-3, fig-4 supporting the statement).

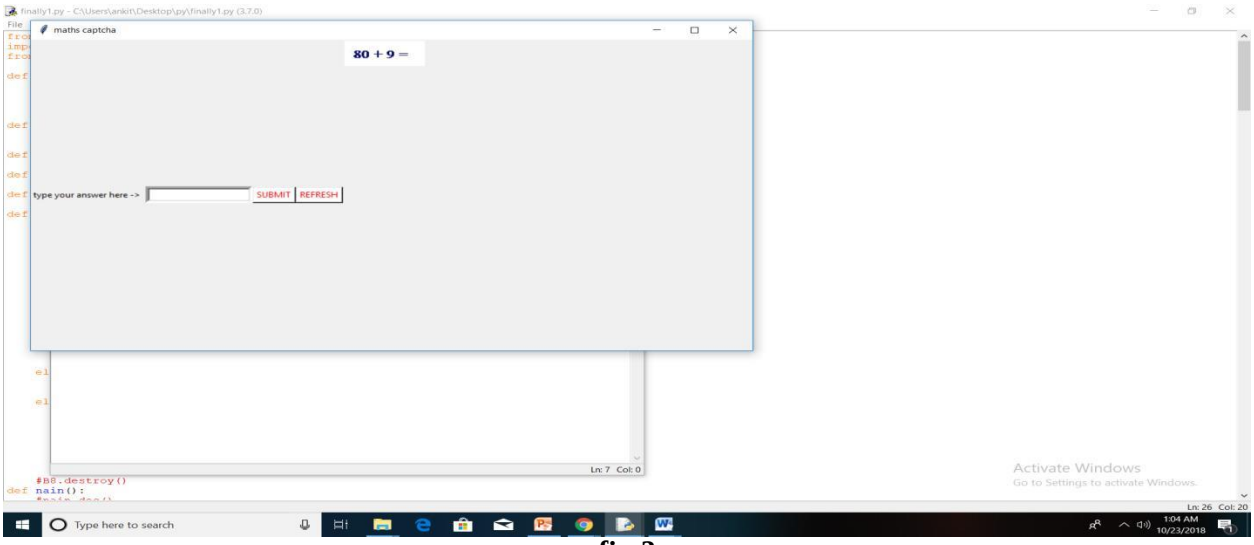


fig-3

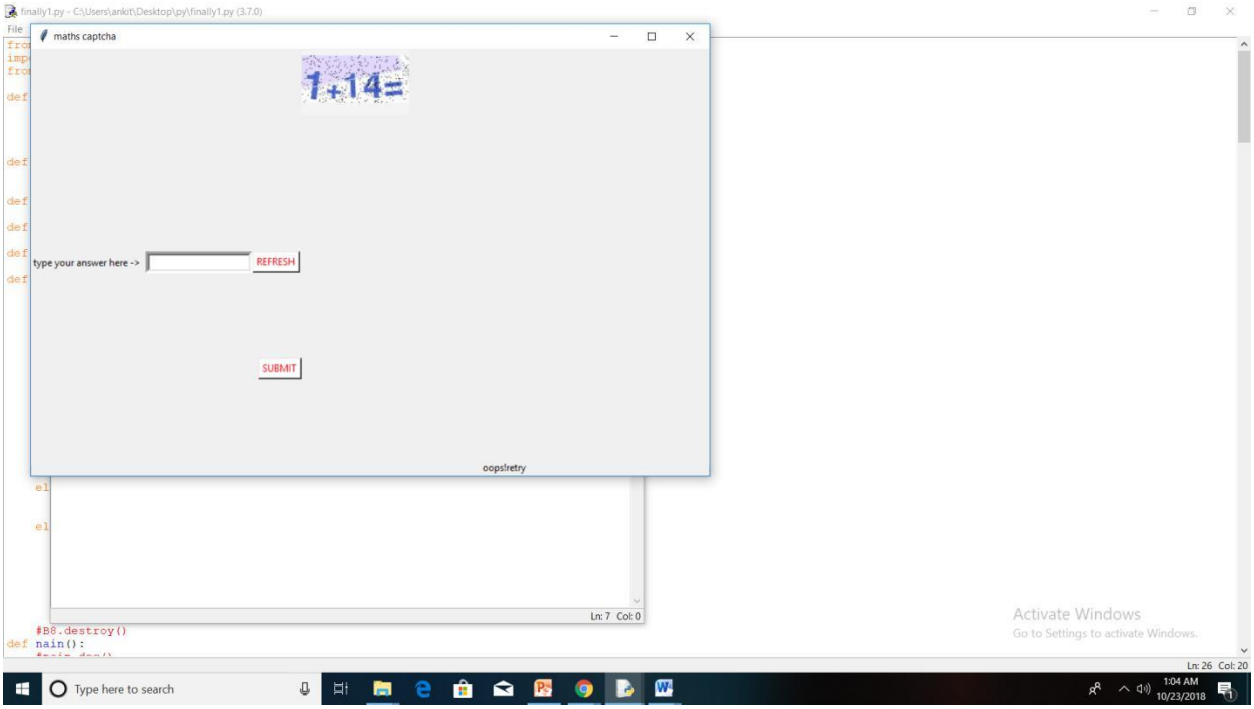


fig-4

3.) If the user cannot able to see the Captcha clearly after refreshing it once, then the user can also further refresh the captcha but limitation for refreshing the captcha is only five time. If the user exceeds the refreshing limit of captcha then the program will close and shows the message “You are out of tries”(fig-5,fig-6 in support of statement).

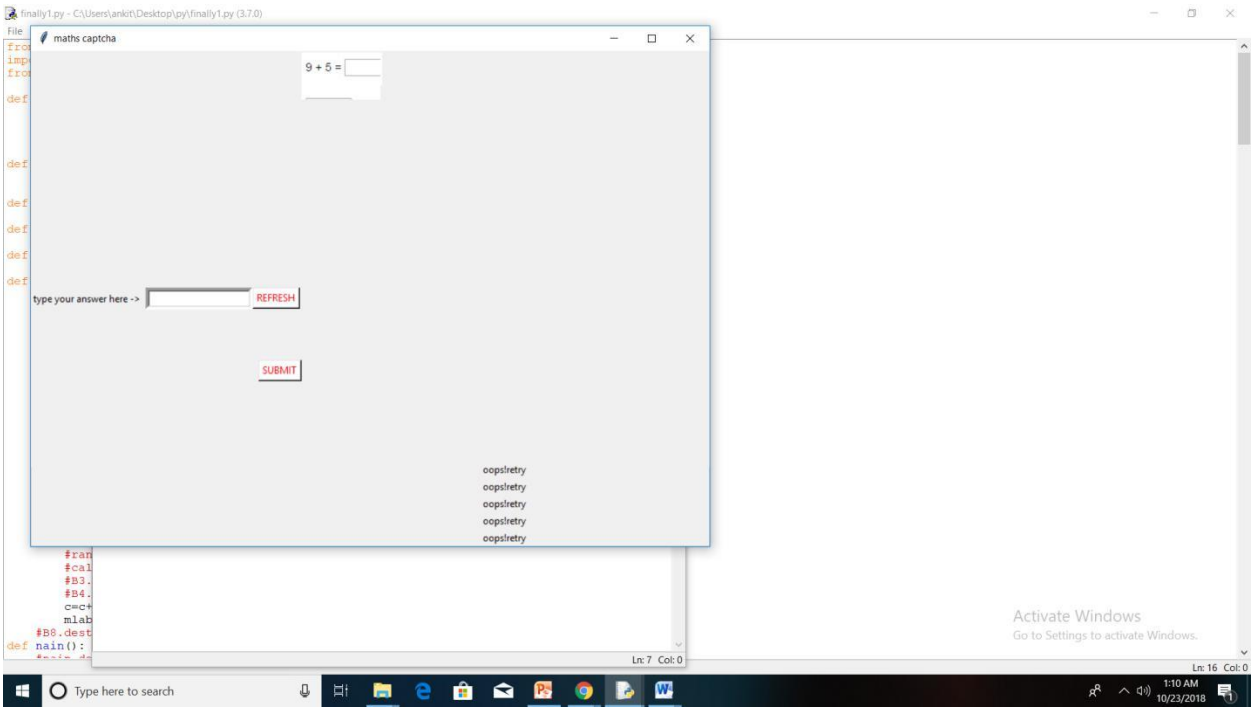
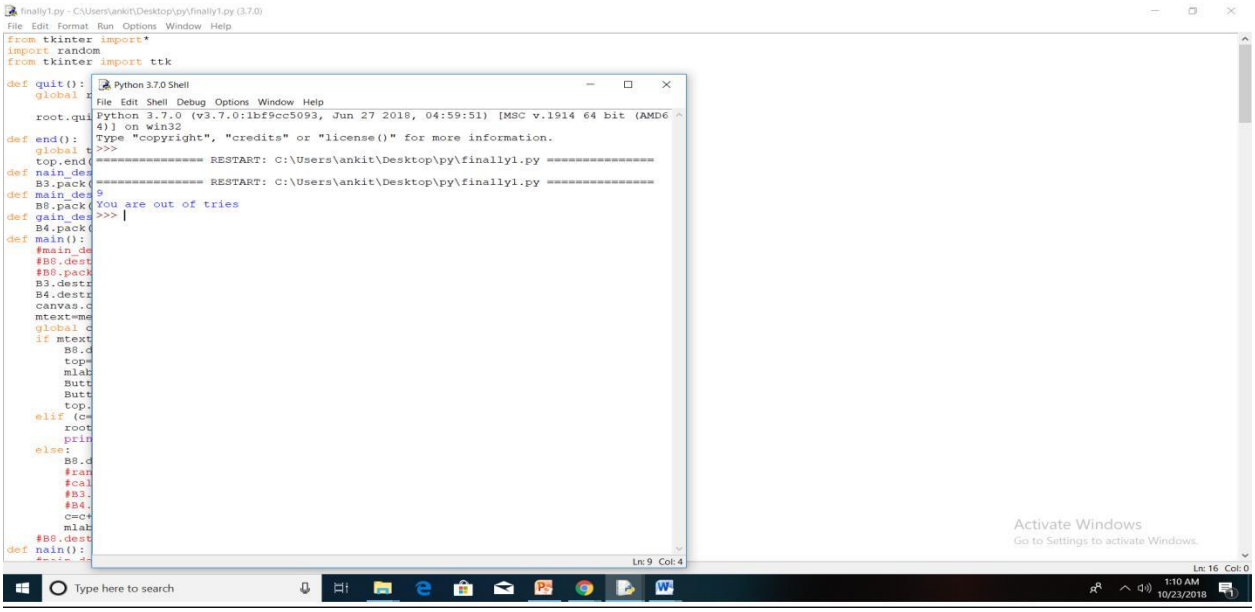


Fig-5:- Five times Refreshing the Captcha



Fig(6)

After Five times Entered Incorrect Captcha, next attempt will let automatically end the program showing message “You are out of tries”, in this case user needs to run the program again.