Title: Text Based Captcha Using Python

Submitted By: Priyanshi Soni(11902073)
Ajay Jangir (11906202)

Submitted to: Neha Mam
Section: RK19TS

Brief summary about code: The project is to generate a random text based captcha using GUI. The program is generating random captcha every time when you start it. Now you have to verify it by entering the captcha. If you write wrong, an alert will pop up in the tkinter screen and saying 'Entered Captcha is wrong' and then a new captcha will automatically generated and now you have to write new captcha. Now If your entered captcha is right , an alert will pop up and say 'Entered captcha is right' then also a new captcha is generated, now you have to write new captcha & the process is carry on. This is how it works.

Now Let's Understand the code !

```
from tkinter import *   #importing tkinter to use GUI
import random    #importing random to generate random numbers
import string    #importing string
import tkinter.messagebox as tmsg
#importing tkinter messagebox as tmsg to show alert

root=Tk() #initialising tkinter with its entry widget root to create a root window

root.geometry("700x600") #Setting dimensions of root window

root.title("Text Based Captcha")
#Putting title 'Text Based captcha' which is our Project name
```

Now we have to generate random numbers & alphabets to make a captcha. What we will do, we will generate random no's and random alphabets's apart and after we will concat them both to make our desired captcha.

These random no's and random alphabets will also having random size & for this we have created below two variables z & p

```
z=random.randint(3,5)  #generating random numbers from 3 to 5
p=random.randint(3,5)  #generating random numbers from 3 to 5

concat=''    #creation of empty string, which we will use later

list1=[]      #creation of empty list1 to store the generated random numbers
list2=[]      #creation of empty list2 to store the generated random alphabets

for i in range(0,z):
    list1.append(str(random.randint(0,9)))
#Now it will take the random size generated by z and will generate the random numbers from 0 to 9 and after typecasting it to string we will storing it in list1

for i in range(0,p):
    list2.append(random.choice(string.ascii_letters
)) #Now it will take the random size generated by p and will generate the random alphabets from (a to z , A to Z) and after generating we will storing it in list2

for i in range(0,random.randint(2,4)):
    concat=concat+list1[random.randint(0,len(list1)
-1)]+list2[random.randint(0,len(list2)-1)]
#Now we are again generating random no in for loop to get everytime random size for our captcha. Inside for loop we are taking random no and alphabet from our stored lists and and storing it in our empty string concat, which we have initialised earlier.
```

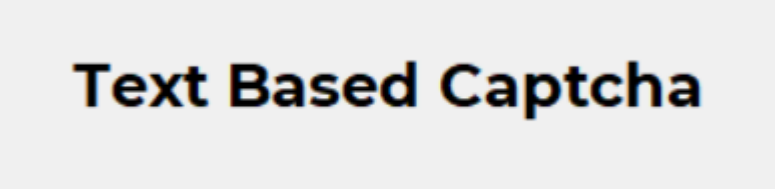
Now we are having our generated captcha in variable 'concat'. We are good to go further .

```
Label(root,text="Text Based Captcha",font="Montserrat 18 bold").pack(pady=70,side=TOP)
```

#Created a label in root window, which will show our project title heading. We dosen't create variable for the label and pack it in one line, because we are not going to do anything with this label further.

We have used font family called 'Montserrat' and set a size 16 and weight bold for the text. Then after I pack my label and align this label to the TOP.

Output:




```
photo=PhotoImage(file="captcha4.png",height=200,width=450)
```

#Created a widget PhotoImage in photo variable and set a default picture namely 'captcha4.png', having height 200 and width 450.

```
labelPhoto=Label(image=photo,text=f"{concat}",font="Aswell 28 overstrike",fg="#575757",compound="center")
labelPhoto.pack()
```

#Created a label to show the image, and we have to show the generated captcha On the image, So we have used a property called compound='center' which shows my text on the image and will center it. Now the text shown is in variable called 'concat' but we have to write generated captcha in double quotes, so we use something called fstring.

We have used font-family called Aswell to give it a captcha style and overstrike to strike my text.

Output:

Now we have successfully generated our captcha and shown it to user as well. Now we require a entry widget so that user can enter the captcha and then we can verify it whether it is correct or not.

```
uservalue=StringVar()
```
#This is the creation of string variable which we are creating for our entry widget which is specifying that entry widget can will accept the values having datatype string.

```
entry= Entry(root,textvariable=uservalue,relief=SUNKEN,borderwidth=4,font="Montserrat 12",justify=CENTER)
entry.pack(ipady=10,pady=60)
```
#Entry widget created inside root window, having datatype string(textvariable=uservalue). We have provided borderwidth and style *SUNKEN* to it. That justify=*CENTER* will write our text from center of entry box, and ipady we have used to give it a height we want.

Output:

The below described function is call when button Is clicked. We have created button below to it and defining it here.

```
def fun1():
    concat2=''  #Creation of empty string
    list3=[]    #Creation of empty list
    list4=[]    #Creation of empty list
```

#This labelPhoto['text'] will give us the the text of that variable, and we have store our generated captcha there so we will get the value of our captcha by this...Now we are checking whether generated captcha and the value entered by user in entry box is same or not...if same then captcha will be verified.

```
    if(labelPhoto['text']==uservalue.get()):
        list5=['captcha.png','captcha3.png','captcha2.png','captcha4.png']#Here we are storing name of random images which we are using in it.

        tmsg.showinfo("Captcha Verification","Entered Captcha is Right") #Alert will pop up here

        for i in range(0,z):
            list3.append(str(random.randint(0,9)))
```

#We are again generating random no's like we did above and storing it in a list

```
        for i in range(0,p):
```

```
            list4.append(random.choice(string.ascii_letters))
```
#We are again generating random alphabet's like we did above and storing it in a list

```
        for i in range(0,random.randint(2,4)):

            concat2=concat2+list3[random.randint(0,len(list1)-1)]+list4[random.randint(0,len(list2)-1)]
```
#Now We are taking randomly data from both the lists and concating it to our empty string which will gives a new captcha which stored in 'concat2'. This captcha is also having random size as we defined range of for loop randomly.

```
        photo['file']=f"{list5[random.randint(0,2)]}"
```
#As we have generated our new captcha now finally we are changing the image randomly. It is taking the image names from list5 which we have created in the starting of if block. We are changing photo with the property Photo['file'] . We are assigning random image from the stored images in list5. We have used fstring because we are using variable and we have to write string value.

```
        labelPhoto['text']=f"{concat2}"
```
#Assigning the generated captcha to text of label variable 'labelPhoto'

```
        uservalue.set('')
```
#We are setting now entry widget value to the empty so that we can write new captcha.

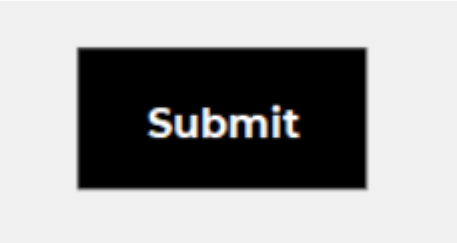
#We are doing the same process which we have done in if part. As we are generating random captcha in both the cases whether user have written captcha right or not.

```
    else:
        tmsg.showinfo("Captcha Verification","Entered Captcha is wrong")
```
#Alert will pop up here

```
        list5=['captcha.png','captcha3.png','captch
a2.png','captcha4.png']


        for i in range(0,z):
            list3.append(str(random.randint(0,9)))

        for i in range(0,p):
            list4.append(random.choice(string.ascii
_letters))

        for i in range(0,random.randint(2,4)):

            concat2=concat2+list3[random.randint(0,
len(list1)-1)]+list4[random.randint(0,len(list2)-
1)]

        photo['file']=f"{list5[random.randint(0,2)]
}"
        labelPhoto['text']=f"{concat2}"
        uservalue.set('')


#Now we finally creating our submit button which is calling the above function.
Button(root,text="Submit",fg="#fff",bg="#000", font
="Montserrat 12 bold",command=fun1,padx=20,pady=10)
.pack()
```

#Button created inside root window having background colour and text color value in hex code. That command=fun1 telling us to which function we have to call.

Output:



`root.mainloop()` #With the help of mainloop, we are running our application.

This is the End of the Project report. Now we are finally sharing the github link to access the whole code , images and the font-family we used. I request you to install the font-family we are sharing so that you can get the same output.

Github link: https://github.com/ajayjangir0044/Python-Project-text-_based_captcha