

# HACKATHON (dec)

**TEAM :1**

**PROJECT : *password generator* .**

## Table of Contents

Sr. No.	Topic	Page No.
1.	Abstract	
2.	Problem Definition	
3.	Introduction	
4.	Description of the modules used	
5.	Implementation details with screen-shots (stepwise)	
6.	Conclusion and Future Scope	
7.	References	
8.	Acknowledgement	

### Abstract

Text password is a very common user authentication technique. Users face a major problem, namely that of many site-unique and strong (i.e. non-guessable) passwords. One way of addressing this is by using a password generator which generates (and regenerates) site-specific strong passwords on demand, with minimal user input. Password Generator enables the generation of passwords that meet important real world requirements, including forced password changes, use of pre-specified characters, displaying the number of characters used in password and checking passwords strength.

### Problem Definition

In this project our motto is to generate a random password based on user's need. User will input his/her need of no of digits, special character, small alphabets, capital alphabets and based on these numbers a random password will be generated. After generating the random password a button of 'Copy to Clipboard' will provide the functionality of copying the generated password

to our clipboard.

It also enables the users to check the strength of their passwords that they have had used on their multipurpose sites. It enables user to check the character they have used to create the password.

## Introduction

Passwords remain a very widely used method for user authentication, despite widely shared concerns about the level of security they provide. There are many potential replacement technologies, including combinations of biometrics and trusted personal devices, but it seems likely that it will be some time before passwords are relegated to history. Given their current and likely future wide use, finding ways of improving the use and management of passwords remains a vitally important issue. We focus here on an important practical matter, namely how to make password more secure and more convenient. Passwords can be stored either locally or on a trusted server; most browsers provide a local-storage password manager. However, the shortcomings of password managers have also been widely documented.

Password Generator enables the user to generate the password of their choice like the number of words, small case alphabets, digits, etc. A Clipboard enables user to copy the password that is generated using the Password Generator. It also enables the user to check the strength of the passwords. It also displays the no. of characters, no of alphabets, and no of symbols used in the passwords.

## Description of the modules used

### Description of Module\_1 (Password\_Generator):

In this module we have imported **tkinter** package and its modules, **pyperclip** module, **random** module. Then initialize tkinter using **Tk()** method.

A variable of string type has been declared named '**passstr**' to store the generated password.

Similarly 4 variables of integer type has been declared named —> '**passlen\_smallalpha**', '**passlen\_bigalpha**', '**passlen\_digits**', '**passlen\_specialcharac**' to store the length of small alphabet characters, capital alphabet characters, digits, special characters which the user will input on his/her choice. The above four variables are set to zero initially using **IntVar()**

A user-defined-function to generate a password will be used named 'generate()'. We have declared four list —> **pass1**, **pass2**, **pass3**, **pass4** which will be having small alphabet characters, capital alphabet characters, digits, special characters as elements in their respective list. A password string is declared, which is initially empty, 'mylist' list will consist of pass1, pass2, pass3, pass4 as its elements.

Then an infinite While-loop is runned which will append the characters to the '**password**' string one by one & will terminate on a specific condition, condition of all big, small, digits, special variables being equal to zero(0).

In while loop the '**mylist**' list is shuffled and then the shuffled list is copied to '**list1**' list.

so the 'list1' is a list having four lists in it(nested lists). Every first element of the nested-lists will be checked to identify the type of elements of that list.

A For-loop is iterating for four times(from 0 to 3) with the second-indexing being fixed to zero (list1[q][0]) as there are four-nested lists. As we check the first element of the first nested-list, the corresponding list is taken (pass1 or pass2 or pass3 or either pass4) and character is taken randomly using **random.choice()** function, from that list. And that character is appended to

'password' string. And the corresponding variable is decremented by 1.

Similarly we proceed forward for the remaining three 3 nested lists. Again the While-loop runs, again the 'mylist' list is shuffled and copied to 'list1' list the same procedure continues. And we proceed forward for the remaining three 3 nested lists.

Finally this While-loop is terminated when condition1 is true.

**(condition1=(small==0 and big==0 and digits==0 and special==0)).**And the password is set to **passstr** using **passstr.set()** function.

### **Description of Module\_2(Strength\_Checker) -**

This module is used to tell the user the strength percent of his/her input password.

As the user inputs his password in the text-field and click on '**Check**' button. A percentage will be displayed indicating the strength of the password. And with the percentage a color box will also be displayed, red color indicates-weak password, yellow color indicates-medium password, green color indicates-strong password.

### **Description of Module\_3 (Check\_character):**

1)Import Libraries

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

2)Initialize window

```
root=Tk()  
root.geometry('400x400')
```

—>>Tk() : It is use to initialize tkinter and create window.

—>>geometry() : It is use to set the width and height of the window.

3)Inside check\_ch function()

```
    for i in range (len(string)):  
        if(string[i].isalpha()):  
            alphabets=alphabets+1  
        elif(string[i].isdigit()):  
            digits=digits+1  
        else:  
            special=special+1
```

—>>.isalpha() : It is use to check character is alphabet or not.

—>>.isdigit() : Its is use to check character is digit or not.

—>>**Variable i** start from 1st character to last character and pass each character to if condition.

In if condition if character is alphabet than alphabet will increment by 1 else if it is digit than digits variable get increment by 1 and if both condition will false than special variable get increment by 1.This process will get repeat until i variable reach at last character in a string.

```

l1=Label(root, text={" alphabets ",alphabets})
l1.pack()
l2=Label(root, text={" digits ",digits})
l2.pack()
l3=Label(root, text={" special symbol ",special})
l3.pack()

```

–>>**Label()** : It is use to display the the text that users can not modify. –>>**root** : It is a name given by programmer to our window.

```

label=Label(root, text=" Enter your password ")
label.pack()
entry=Entry(root, textvariable="")
entry.pack()
button=Button(root, text="Check Character", command=
check_ch) button.pack()
root.mainloop()

```

–>>**text** : Which we display on the label.  
–>>**Entry()** : It will create input filled for user.  
–>>**Button()** : Use to display the button on our window.  
–>>**command()** : It will work when button is click.  
–>>**textvariable** : It is use to retrieve the text to the entry.

### **Implementation details with screen-shots (stepwise)**

**Code:-**

```

1  from tkinter import *
2  import pyperclip
3  import random
4  from password_strength import *
5  import math
6  from tkinter import messagebox
7
8  root=Tk()
9  root.geometry('350x350')
10 root.title('Python MiniProject')
11 root.iconbitmap('bitmap.ico')
12
13 passstr = StringVar()
14 passlen_smallalpha = IntVar()
15 passlen_bigalpha = IntVar()
16 passlen_digits = IntVar()
17 passlen_specialcharac = IntVar()
18 passstr2 = StringVar()
19 passstr1 = StringVar()
20
21 passlen_smallalpha.set(0)
22 passlen_bigalpha.set(0)
23 passlen_digits.set(0)
24 passlen_specialcharac.set(0)
25
26
27 def generate():
28
29     list1=[]
30     small=passlen_smallalpha.get()
31     big=passlen_bigalpha.get()
32     digits = passlen_digits.get()
33     special = passlen_specialcharac.get()
34
35     pass1 = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j',
36             'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't',
37             'u', 'v', 'w', 'x', 'y', 'z']

```

```

37         'u', 'v', 'w', 'x', 'y', 'z']
38
39     pass2 = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N',
40             'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']
41
42     pass3 = ['1', '2', '3', '4', '5', '6', '7', '8', '9', '0']
43     pass4 = ['!', '@', '#', '$', '%', '&', '*']
44
45     password = ""
46     mylist=[pass1,pass2,pass3,pass4]
47     k=1
48
49     while(k==1):
50         random.shuffle(mylist)
51         list1=mylist
52         for q in range(4):
53             if(list1[q][0]=='A' and big != 0):
54                 password = password + random.choice(pass2)
55                 big-=1
56             if(list1[q][0]=='a' and small!=0):
57                 password = password + random.choice(pass1)
58                 small-=1
59             if(list1[q][0]=='1' and digits != 0):
60                 password = password + random.choice(pass3)
61                 digits-=1
62             if(list1[q][0]=='!' and special != 0):
63                 password = password + random.choice(pass4)
64                 special-=1
65             if(small==0 and big==0 and digits==0 and special==0):
66                 break
67
68
69     passstr.set(password)
70
71     def copy_to_clipboard():
72         random_password = passstr.get()
73         pyperclip.copy(random_password)

```

```

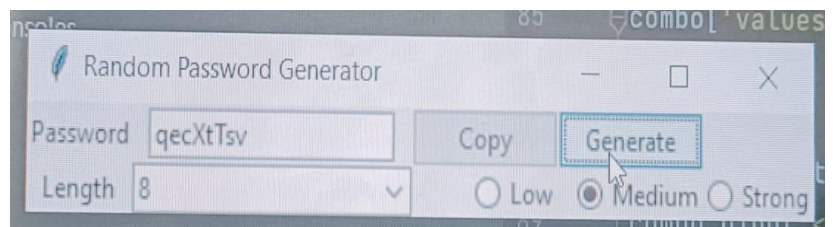
74
75 def module_1():
76     f1=Frame()
77     f1.place(x=0,y=0,width=350,height=350)
78     Label(f1, text="Password Generator ", font="calibre 20 bold").pack()
79
80     Label(f1, text="Enter no. of Small Alphabets").pack(pady=3)
81     Entry(f1, textvariable=passlen_smallalpha).pack(pady=3)
82
83     Label(f1, text="Enter no. of Big Alphabets").pack(pady=3)
84     Entry(f1, textvariable=passlen_bigalpha).pack(pady=3)
85
86     Label(f1, text="Enter no. of Digits").pack(pady=3)
87     Entry(f1, textvariable=passlen_digits).pack(pady=3)
88
89     Label(f1, text="Enter no. of Special Characters").pack(pady=3)
90     Entry(f1, textvariable=passlen_specialcharac).pack(pady=3)
91
92     Button(f1, text="Generate Password", command=generate).pack(pady=7)
93     Entry(f1, textvariable=passstr).pack(pady=3)
94     Button(f1, text="Copy to clipboard", command=copy_to_clipboard).pack()
95     Button(f1, text='->',command=home).place(x=0,y=0)
96
97 def strength_checker():
98     f1=Frame()
99     f1.place(x=0,y=0,width=350,height=350)
100     if passstr2.get() == "":
101         messagebox.showinfo("Error", "Password Can't be Empty")
102     else:
103         result = PasswordStats(passstr2.get())
104         final = result.strength()
105         label1 = Label(f1,text="")
106         w = Canvas(f1,height=100, width=600)
107         w.pack()
108         label1.place(x=170, y=100)
109         label1["text"] = str(math.ceil(final * 100)) + " %"
110         if final >= 0.66:

```

```

111         w.create_rectangle(70, 50, 280, 100, fill="#27cf54", outline="white")
112     elif final > 0.10 and final < 0.40:
113         w.create_rectangle(70, 50, 280, 100, fill="#f0f007", outline="white")
114     elif final <= 0.10:
115         w.create_rectangle(70, 50, 280, 100, fill="#de3c3c", outline="white")
116     b=Button(f1,text="->",command=module_2)
117     b.place(x=0,y=0)
118
119 def module_2():
120     f1=Frame()
121     f1.place(x=0,y=0,width=350,height=350)
122     head = Label(f1, text="Password Strength Calculator", font=("helvetica", 15, "bold"))
123     head.pack(ipadx=12, ipady=12)
124     label = Label(f1, text="Enter Your Password", font=("helvetica", 10, "bold"))
125     label.pack(ipadx=5, ipady=5)
126     entry =Entry(f1,textvariable=passstr2)
127     entry.pack(ipadx=30, ipady=5)
128     button = Button(f1, text="check", command=strength_checker)
129     button.pack(ipadx=5, ipady=5)
130     b=Button(f1,text="->",command=home)
131     b.place(x=0,y=0)
132
133 def check_ch():
134     f1=Frame()
135     f1.place(x=0,y=0,width=350,height=350)
136     string = passstr1.get()
137     alphabets=digits=special=0
138
139     for i in range (len(string)):
140         if(string[i].isalpha()):
141             alphabets=alphabets+1
142         elif(string[i].isdigit()):
143             digits=digits+1
144         else:
145             special=special+1
146     b=Button(f1,text="->",command=module_3)
147     b.place(x=0,y=0)

```

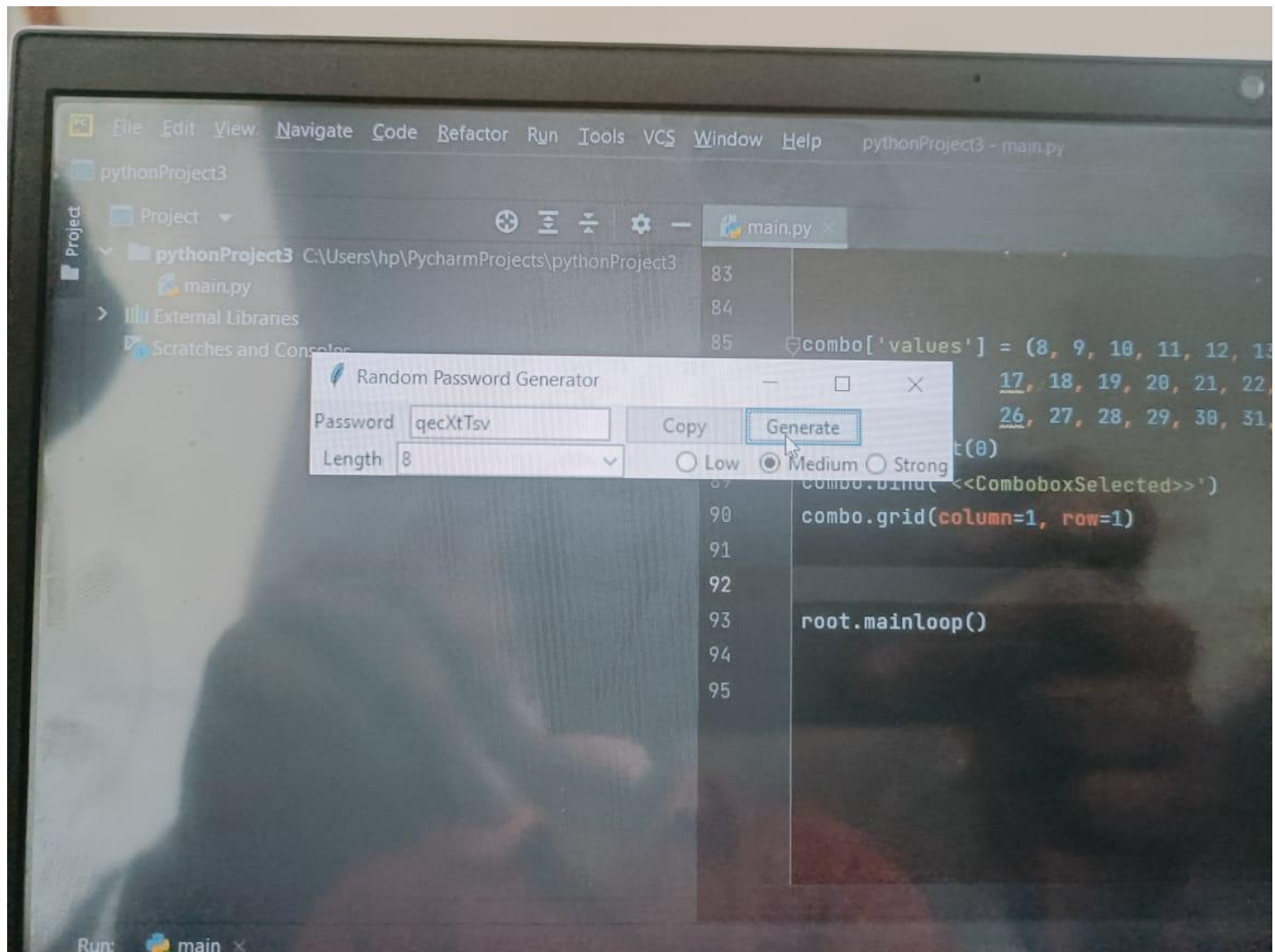


# Output:-









## Conclusion and Future Scope

With these steps, we have successfully created a random password generator project using python. We used popular tkinter library to rendering graphics in our display window and we also learned about pyperclip and random library.

We learned how to create buttons, input text field and labels. In this way, we successfully created our password generator python project.

## References

- [1] <https://docs.python.org/3/library/tkinter.html>
- [2] [https://www.tutorialspoint.com/python3/python\\_gui\\_programming.htm](https://www.tutorialspoint.com/python3/python_gui_programming.htm)
- [3] <https://www.geeksforgeeks.org/python-tkinter-tutorial/>
- [4] <https://www.geeksforgeeks.org/python-strings/>
- [5] <https://stackoverflow.com/questions/51777956/link-gui-to-main-class>

## Acknowledgement

We would like to express my special thanks of gratitude to our Prof. sanjiven sir)who gave us the golden opportunity to do this wonderful project on the topic Password Genrator, which also helped us in doing a lot of Research and we came to know about so many new things we are really thankful to you. Secondly we would also like to thank our friends who helped us a lot in finalizing this project within the limited time frame.