```
'''PROJECT TITLE: PASSWORD GENERATOR
   CODE LANGUAGE: PYTHON 3.7.0a3(32-bit)
   FILE NAME: passwordGenerator.py'''

'''SOURCE CODE:'''

# importing python modules
import random
from tkinter import *
from tkinter.ttk import *
import time
import datetime

# Function to generate multiple password
def qwe(len,ch):

        # Function to cop multiple generated passwords
    def copy():
        import pyperclip
        c=""
        s=(lis.get(0,END))
        for i in s:
            c+=i+"\n"
        pyperclip.copy(c.strip())


    # Function to reset the window
    def Reset():
        lis.delete(0,END)
        E1.delete(0,END)
        entry5.delete(0,END)
        entry5.insert(0,"Enter the count")

        # Function to display message
    def msg(s):
        entry5.insert(END, s)

    # Function to check the input given by the user
    def reqpw_except(cc):
        reqPWs=cc
        try:
            aa=int(reqPWs)
            if(aa==0):
                si="Enter count greater than 0"
                return msg(si)
            elif aa<0:
                si="Error:Negative value!"
                return msg(si)
            else:return aa
        except ValueError:
            si="Error:Invalid Input count"
            return msg(si)

    # Function to destroy multiple password generator window
    def qExit():
        root1.destroy()

    # Function to generate passwords
    def multiple(len,ch):
        entry5.delete(0,END)
        count=E1.get()
        count4=reqpw_except(count)
        password=""
        for i in range(0, count4):
            for j in range(len):
                password+=random.choice(ch)
            password+="\n"
        entry5.insert(0,"Copy your passwords")
        lis.delete(0,END)
        for index,i in enumerate(password.split("\n")):
            lis.insert(index,i)

```

```
73        #GUI for multiple password generator
74        root1=Tk()
75        root1.title("Multiple Password generator")
76
77        #frames
78        fm1 = Frame(root1, width = 20, relief = SUNKEN)
79        fm1.pack(side = TOP,padx=10,pady=10)
80
81        fm = Frame(root1, width = 20, relief = SUNKEN)
82        fm.pack(side = TOP,padx=10,pady=10)
83
84        fm2 = Frame(root1, width = 20, relief = SUNKEN)
85        fm2.pack(side = TOP,padx=10,pady=10)
86
87        fm3 = Frame(root1, width = 20, relief = SUNKEN)
88        fm3.pack(side = TOP,padx=10,pady=10)
89
90        fm4 = Frame(root1, width = 20, relief = SUNKEN)
91        fm4.pack(side = TOP,padx=10,pady=10)
92
93
94        password_count = Label(fm1, text=" Password Count: ",font = ('timesnewroman',
          12,'bold'))
95        password_count.pack(side=LEFT,padx=2,pady=2)
96        E1 = Entry(fm1,font = ('timesnewroman', 15))
97        E1.pack(side = LEFT)
98
99
100       lblInfo =Label(fm, font=('timesnewroman',12,'bold'),text = " Message:
          ")
101       lblInfo.pack(side=LEFT,padx=2,pady=2)
102       entry5 =Entry(fm,width=35,font = ('timesnewroman', 15, 'bold'))
103       entry5.pack(side=LEFT)
104       entry5.insert(0,"Enter password count")
105
106
107       scrollbar=Scrollbar(fm2)
108       scrollbar.pack(side=RIGHT, fill=Y)
109       lis=Listbox(fm2,width=50,height=10,font=('timesnewroman,14,bold'))
110       lis.pack(side=LEFT,padx=5,pady=5,expand=1)
111       lis.config(yscrollcommand=scrollbar.set)
112       scrollbar.config(command=lis.yview)
113
114       #buttons
115       generate_button = Button(fm3,text="Generate",command=lambda:multiple(len,ch))
116       generate_button.pack(side=LEFT)
117
118       copy_button1 = Button(fm3, text="Copy",command=copy)
119       copy_button1.pack(side=LEFT)
120
121       reset1= Button(fm3, text="Reset",command=Reset)
122       reset1.pack(side=LEFT)
123
124       exit1= Button(fm3, text="Exit",command=qExit)
125       exit1.pack(side=LEFT)
126
127       #TIME
128       localtime = time.asctime(time.localtime(time.time()))
129       lblInfo = Label(fm4, font=('timesnewroman',12,'bold'),text =
          localtime)
130       lblInfo.pack(side=BOTTOM)
131
132
133       root.mainloop()
134
135
136   #single password generator
137
138   # Function to check length
139   def length_except(a):
140       length=a
141       try:
```

```
        bb=int(length)
        if(bb==0 or bb==1 or bb==2 or bb==3):
            si="Select length(4-32)"
            return msg(si)
        elif bb>32:
            si="Enter the Length less than 32"
            return msg(si)
        elif bb<0:
            si="Error:Negative Value Length"
            return msg(si)
        else:
            return bb
    except ValueError:
        si="Error:Invalid Input Length"
        return msg(si)

# Function to display msg
def msg(s):
    entry1.insert(END, s+"\n                                        ")

# Function to check the single/multiple choice
def check(val,ch,length):
        if val==1:
                 return single(length,ch)
        elif val==2:
                 if length>=4 and length<=32:
                         qwe(length,ch)
        else:
                s="select single/multiple"
                return msg(s)

# Function to generate Single password
def single(length,ch):
    password=""
    for i in range(0, length):
            password +=random.choice(ch)
    return password


    if value==1:
        return single(length,ch)
    else:
        s="select single/multiple"
        return msg(s)

# Function for calculation of password
def calculation():
    entry.delete(0, END)
    entry1.delete(0,END)
    # Get the length of password
    length=var1.get()
    value=var.get()
    length=length_except(length)
    password=""
    num='0123456789'
    SLet='abcdefghijklmnopqrstuvwxyz'
    CLet='ABCDEFGHIJKLMNOPQRSTUVWXYZ'
    punc="!\#$%&()*+-/:<=>?@[\\]^_{|}~"
    if v1.get()== 1 and v2.get()== 0 and v3.get()== 0 and v4.get()== 0:
        ch=num
        return check(value,ch,length)
    elif v1.get()== 0 and v2.get()== 0 and v3.get()== 0 and v4.get()== 4:
        ch=punc
        return check(value,ch,length)
    elif v1.get()== 0 and v2.get()== 0 and v3.get()== 3 and v4.get()== 0:
        ch=SLet
        return check(value,ch,length)
    elif v1.get()== 0 and v2.get()== 2 and v3.get()== 0 and v4.get()== 0:
        ch=CLet
        return check(value,ch,length)
    elif v1.get()== 1 and v2.get()== 2 and v3.get()== 3 and v4.get()== 4:
        ch=num+punc+SLet+CLet
```

```
            return check(value,ch,length)
        elif v1.get()== 0 and v2.get()== 2 and v3.get()== 3 and v4.get()== 0:
            ch=CLet+SLet
            return check(value,ch,length)
        elif v1.get()== 1 and v2.get()== 0 and v3.get()== 3 and v4.get()== 0:
            ch=num+SLet
            return check(value,ch,length)
        elif v1.get()== 1 and v2.get()== 2 and v3.get()== 0 and v4.get()== 0:
            ch=num+CLet
            return check(value,ch,length)
        elif v1.get()== 0 and v2.get()== 0 and v3.get()== 3 and v4.get()== 4:
            ch=punc+SLet
            return check(value,ch,length)
        elif v1.get()== 0 and v2.get()== 2 and v3.get()== 0 and v4.get()== 4:
            ch=punc+CLet
            return check(value,ch,length)
        elif v1.get()== 1 and v2.get()== 0 and v3.get()== 0 and v4.get()== 4:
            ch=punc+num
            return check(value,ch,length)
        elif v1.get()== 1 and v2.get()== 0 and v3.get()== 3 and v4.get()== 4:
            ch=punc+num+SLet
            return check(value,ch,length)
        elif v1.get()== 1 and v2.get()== 2 and v3.get()== 0 and v4.get()== 4:
            ch=punc+CLet+num
            return check(value,ch,length)
        elif v1.get()== 1 and v2.get()== 2 and v3.get()== 3 and v4.get()== 0:
            ch=num+CLet+SLet
            return check(value,ch,length)
        elif v1.get()== 0 and v2.get()== 2 and v3.get()== 3 and v4.get()== 4:
            ch=punc+SLet+CLet
            return check(value,ch,length)
        else:
            return msg("Tick options")


    # Function for generation of password
    def generate():
        password = calculation()
        entry.insert(END, password)
        msg("copy your password")

    # Function for copying password to clipboard
    def copy1():
        import pyperclip
        random_password = entry.get()
        pyperclip.copy(random_password)

    # exit function
    def qExit():
        root.destroy()
        exit(0)

    # Function to reset the window
    def Reset():
        v1.set(0)
        v2.set(0)
        v3.set(0)
        v4.set(0)
        var.set(0)
        var1.set("4")
        entry.delete(0,END)
        entry1.delete(0,END)


    # Main Function

    # create GUI window
    root = Tk()

    # defining size of window
    root.geometry("590x370")

```

```
286
287   #frame1
288   top = Frame(root, width = 20, relief = SUNKEN)
289   top.pack(side = TOP,padx=10,pady=10)
290
291   #frame2
292   tops = Frame(root, width = 20, relief = SUNKEN)
293   tops.pack(side = TOP,padx=10,pady=10)
294
295   #frame3
296   radio = Frame(root, width = 20, relief = SUNKEN)
297   radio.pack(side = TOP,padx=10,pady=10)
298
299   #frame4
300   f2 = Frame(root, width = 20, height = 20,relief = SUNKEN)
301   f2.pack(side = TOP,padx=10,pady=10)
302
303   #frame5
304   fw = Frame(root, width = 20, height = 20,relief = SUNKEN)
305   fw.pack(side = TOP,padx=10,pady=10)
306
307   #frame6
308   last = Frame(root, width = 20, height = 20,relief = SUNKEN)
309   last.pack(side = TOP,padx=10,pady=10)
310
311   #frame7
312   f5 = Frame(root, width = 20, height = 20,relief = SUNKEN)
313   f5.pack(side = TOP,padx=10,pady=10)
314
315   #frame8
316   f4= Frame(root, width = 20, height = 20,relief = SUNKEN)
317   f4.pack(side = TOP,padx=10,pady=10)
318
319   var=IntVar()
320   v1=IntVar()
321   v2=IntVar()
322   v3=IntVar()
323   v4=IntVar()
324   var1=StringVar()
325
326
327   # Title of your GUI window
328   root.title("Password Generator")
329
330   Random_password = Label(last, text="Generated Password: ",font = ('timesnewroman',
      12,'bold'))
331   Random_password.pack(side=LEFT,padx=5)
332   entry =Entry(last,width=30,font = ('timesnewroman', 15, 'bold'))
333   entry.pack(side=LEFT,padx=2,fill=Y)
334
335
336   c_label = Label(top, text="Length: ",font = ('timesnewroman', 12, 'bold'))
337   c_label.pack(side=LEFT,padx=5)
338   combo = Combobox(top, textvariable=var1,font = ('timesnewroman', 15))
339   combo['values'] = (4,5,6,7,8, 9, 10, 11, 12, 13, 14, 15, 16,
340                      17, 18, 19, 20, 21, 22, 23, 24, 25,
341                      26, 27, 28, 29, 30, 31, 32)
342   combo.current(0)
343   combo.bind('<<ComboboxSelected>>')
344   combo.pack(side=LEFT,padx=2)
345
346   #checkButtons
347
348   a=Checkbutton(tops, text="DIGITS",variable=v1,onvalue=1,offvalue=0)
349   a.pack(side=LEFT,fill=BOTH)
350
351   b=Checkbutton(tops, text="UPPER", variable=v2,onvalue=2,offvalue=0)
352   b.pack(side=LEFT,fill=BOTH)
353
354   c=Checkbutton(tops, text="LOWER", variable=v3,onvalue=3,offvalue=0)
355   c.pack(side=LEFT,fill=BOTH)
356
```

```
d=Checkbutton(tops, text="SYMBOLS", variable=v4,onvalue=4,offvalue=0)
d.pack(side=LEFT,fill=BOTH)


#radioButtons

radio_low = Radiobutton(radio,text="SINGLE PASSWORD", variable=var, value=1)
radio_low.pack(side=LEFT,fill=BOTH)

radio_strong = Radiobutton(radio, text="MULTIPLE PASSWORDS", variable=var, value=2)
radio_strong.pack(side=LEFT,fill=BOTH)


#buttons

generate_button = Button(f2,text="Generate",command=generate)
generate_button.pack(side=LEFT,fill=BOTH)

copy_button = Button(f5, text="Copy",command=copy1)
copy_button.pack(side=LEFT)

copy_button1 = Button(f5, text="Exit",command=qExit)
copy_button1.pack(side=RIGHT)

btnReset = Button(f5,text = "Reset",command = Reset).pack(side=RIGHT)

lblInfo =Label(fw,font=('timesnewroman',12,'bold'),text = " Message:
")
lblInfo.pack(side=LEFT,padx=2,pady=2)
entry1 =Entry(fw,width=36,font = ('timesnewroman', 15, 'bold'))
entry1.pack(side=LEFT)

#TIME
localtime = time.asctime(time.localtime(time.time()))
lblInfo = Label(f4, font=('timesnewroman',12,'bold'),text =
localtime)
lblInfo.pack(side=BOTTOM)


# start the GUI
root.mainloop()

```