```
import string

import random

from tkinter import *

from tkinter import messagebox

import re

import sqlite3

with sqlite3.connect("users.db") as db:

  cursor = db.cursor()

cursor.execute("CREATE TABLE IF NOT EXISTS users(Username TEXT NOT NULL, GeneratedPassword TEXT NOT NULL);")

cursor.execute("SELECT * FROM users")

db.commit()
```

```
db.close()

class GUI():

  def __init__(self,master):

      self.master = master

      self.username = StringVar()

      self.passwordlen = IntVar()

      self.generatedpassword = StringVar()

      self.n_username = StringVar()

      self.n_generatedpassword = StringVar()

      self.n_passwordlen = IntVar()

      root.title('Password Generator')
```

```
root.geometry('660x500')

root.config(bg='#FF8000')

root.resizable(False, False)

self.label = Label(text=":PASSWORD GENERATOR:", anchor=N, fg='darkblue', bg='#FF8000', font='arial 20 bold underline')

self.label.grid(row=0, column=1)

self.blank_label1 = Label(text="")

self.blank_label1.grid(row=1, column=0, columnspan=2)

self.blank_label2 = Label(text="")

self.blank_label2.grid(row=2, column=0, columnspan=2)

self.blank_label2 = Label(text="")

self.blank_label2.grid(row=3, column=0, columnspan=2)
```

```
self.user = Label(text="Enter User Name: ", font='times 15 bold', bg='#FF8000', fg='darkblue')

self.user.grid(row=4, column=0)

self.textfield = Entry(textvariable=self.n_username, font='times 15', bd=6, relief='ridge')

self.textfield.grid(row=4, column=1)

self.textfield.focus_set()

self.blank_label3 = Label(text="")

self.blank_label3.grid(row=5, column=0)

self.length = Label(text="Enter Password Length: ", font='times 15 bold', bg='#FF8000', fg='darkblue')

self.length.grid(row=6, column=0)

self.length_textfield = Entry(textvariable=self.n_passwordlen, font='times 15', bd=6, relief='ridge')

self.length_textfield.grid(row=6, column=1)
```

```
    self.blank_label4 = Label(text="")

    self.blank_label4.grid(row=7, column=0)

    self.generated_password = Label(text="Generated Password: ", font='times 15 bold', bg='#FF8000', fg='darkblue')

    self.generated_password.grid(row=8, column=0)

    self.generated_password_textfield = Entry(textvariable=self.n_generatedpassword, font='times 15', bd=6, relief='ridge', fg='#DC143C')

    self.generated_password_textfield.grid(row=8, column=1)

    self.blank_label5 = Label(text="")

    self.blank_label5.grid(row=9, column=0)

    self.blank_label6 = Label(text="")

    self.blank_label6.grid(row=10, column=0)

    self.generate = Button(text="GENERATE PASSWORD", bd=3, relief='solid', padx=1, pady=1, font='Verdana", 15 bold', fg='#68228B',
bg='#BCEE68', command=self.generate_pass)
```

```
        self.generate.grid(row=11, column=1)

        self.blank_label5 = Label(text="")

        self.blank_label5.grid(row=12, column=0)

        self.accept = Button(text="ACCEPT", bd=3, relief='solid', padx=1, pady=1, font='Helvetica 15 bold italic', fg='#458B00', bg='#FFFAF0',
command=self.accept_fields)

        self.accept.grid(row=13, column=1)

        self.blank_label1 = Label(text="")

        self.blank_label1.grid(row=14, column=1)

        self.reset = Button(text="RESET", bd=3, relief='solid', padx=1, pady=1, font='Helvetica 15 bold italic', fg='#458B00', bg='#FFFAF0',
command=self.reset_fields)

        self.reset.grid(row=15, column=1)

    def generate_pass(self):
```

```
upper = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

lower = "abcdefghijklmnopqrstuvwxyz"

chars = "@#%&()\"?!"

numbers = "1234567890"

upper = list(upper)

lower = list(lower)

chars = list(chars)

numbers = list(numbers)

name = self.textfield.get()

leng = self.length_textfield.get()

with sqlite3.connect("users.db") as db:
```

```
    cursor = db.cursor()
if name=="":
    messagebox.showerror("Error","Name cannot be empty")
    return
if name.isalpha()==False:
    messagebox.showerror("Error","Name must be a string")
    self.textfield.delete(0, 25)
    return
length = int(leng)
if length<6:
    messagebox.showerror("Error", "Password must be atleast 6 characters long")
```

```
    self.textfield.configure(text="")
    return
else:
    self.blank_label1.configure(text="")
self.generated_password_textfield.delete(0, length)
u = random.randint(1, length-3)
l = random.randint(1, length-2-u)
c = random.randint(1, length-1-u-l)
n = length-u-l-c
password = random.sample(upper, u)+random.sample(lower, l)+random.sample(chars, c)+random.sample(numbers, n)
random.shuffle(password)
```

```
    gen_passwd = "".join(password)

    n_generatedpassword = self.generated_password_textfield.insert(0, gen_passwd)

  def accept_fields(self):

    with sqlite3.connect("users.db") as db:

        cursor = db.cursor()

        find_user = ("SELECT * FROM users WHERE Username = ?")

        cursor.execute(find_user, [(self.n_username.get())])

        if cursor.fetchall():

         messagebox.showerror("This username already exists!", "Please use an another username")

        else:

         insert = str("INSERT INTO users(Username, GeneratedPassword) VALUES(\'%s\', \'%s\');"%(self.n_username.get(),
self.n_generatedpassword.get()))
```

```
            cursor.execute(insert)

            db.commit()

            messagebox.showinfo("Success!", "Password generated successfully")

    def reset_fields(self):

        self.textfield.delete(0, 25)

        self.length_textfield.delete(0, 25)

        self.generated_password_textfield.delete(0, 25)

if __name__=='__main__':

    root = Tk()

    pass_gen = GUI(root)

    root.mainloop()
```