Project*: *Random Password Generator with GUI

Aim : The aim of this project is to develop a random password generator that can generate strong and unique passwords for different users and applications.

Description : The random password generator is a Python mini-project that generates strong and unique passwords for different users and applications, providing a user-friendly GUI that allows users to generate passwords by specifying the length. Overall, the random password generator is a useful and practical application that provides a simple and effective solution for generating passwords. With its user-friendly interface, robust security features, and flexible customization options, the random password generator is a valuable tool for anyone looking to improve their online security and privacy.

Scope : The scope of the project is to develop a Python application that generates strong and unique passwords for different users and applications. The application will be designed with a simple and intuitive user interface that allows users to easily generate passwords by specifying the length and complexity requirements.

Concepts Used :

1) Objected-oriented Programming (Oops)
2) File Operations
3) DBM Database
4) Logic behind the application (Generating random password)
5) Tkinter (For GUI)

Program :

```
"""
The below code is importing four modules in Python: `random`, `dbm`, `os`, and `tkinter`. These modules provide various functionalities such as generating random numbers, interacting with databases, accessing operating system functionalities, and creating graphical user interfaces. However, the code itself does not perform any specific task or operation.
"""
import random
import dbm
import os
import tkinter



"""
The below code is changing the current working directory to below mentioned directory.
"""
os.chdir(r"C:\Users\PAVAN KALYAN\Desktop\Random Password Generator")



"""The below code is creating a new or opening an existing database file named "accountsDB" using the dbm module. The 'c' argument specifies that the database should be created if it does not exist. The variable "access_to_save" is initialized to False.
"""
accountsDB = dbm.open("accountsDB", 'c')
access_to_save = False



# 8 : (1 up + 2 lc + 2 num + 3 spc)
# 9 : (2 up + 2 lc + 2 num + 3 spc)
# 10: (2 up + 3 lc + 2 num + 3 spc)
# 11: (2 up + 3 lc + 3 num + 3 spc)
# 12: (2 up + 4 lc + 3 num + 3 spc)
# 13: (2 up + 4 lc + 3 num + 4 spc)
# 14: (3 up + 4 lc + 3 num + 4 spc)
# 15: (3 up + 4 lc + 4 num + 4 spc)
# 16: (4 up + 4 lc + 4 num + 4 spc)
```

```
"""
The "allChars" class generates a random password of varying length and complexity based on the
length of the password, with a mix of uppercase letters, lowercase letters, numbers, and special
characters.
"""

class allChars:
    def _init_(self):
        """
        The function initializes four lists containing special characters, numbers, lowercase
        letters, and uppercase letters.
        """
        self.special_chars = ['~', '!', '@', '#', '$', '%', ^', '&', '*','=', '+', '_', <', '>', '?', '[', ']', '{', '}']
        self.numbers = ['1', '2', '3', '4', '5', '6', '7', '8', '9', '0']
        self.lower_case = ['a', 'b', 'c', 'd', 'e', 'f','g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r','s', 't', 'u',
                           'v', 'w', 'x', 'y', 'z']

        self.upper_case = ['A', 'B', 'C', 'D', 'E', 'F',
                           'G', 'H', 'I', 'J', 'K', 'L',
                           'M', 'N', 'O', 'P', 'Q', 'R',
                           'S', 'T', 'U', 'V', 'W', 'X',
                           'Y', 'Z']

    def generate(self):
        """
        This function generates a random password of varying length and complexity based on
        the length of the password.
            :return: a randomly generated password of length between 8 and 16 characters, with a mix of
            uppercase letters, lowercase letters, numbers, and special characters. The specific mix of characters is
            determined by the length of the password, with longer passwords having more of each type of
            character. The password is shuffled before being returned.
        """
        self.password_length = random.choice(range(8,17))
        if self.password_length==8:
            up  = 1
            lc  = 2
            num = 2
            spc = 3
        elif self.password_length==9:
            up  = 2
            lc  = 2
            num = 2
            spc = 3
        elif self.password_length==10:
            up  = 2
            lc  = 3
            num = 2
```

```
    spc = 3
elif self.password_length==11:
    up  = 2
    lc  = 3
    num = 3
    spc = 3
elif self.password_length==12:
    up  = 2
    lc  = 4
    num = 3
    spc = 3
elif self.password_length==13:
    up  = 2
    lc  = 4
    num = 3
    spc = 4
elif self.password_length==14:
    up  = 3
    lc  = 4
    num = 3
    spc = 4
elif self.password_length==15:
    up  = 3
    lc  = 4
    num = 4
    spc = 4
else:
    up  = 4
    lc  = 4
    num = 4
    spc = 4

str_u=[]
for i in range(up):
    str_u.append(random.choice(self.upper_case))

str_l=[]
for i in range(lc):
    str_l.append(random.choice(self.lower_case))

str_n=[]
for i in range(num):
    str_n.append(random.choice(self.numbers))
```

```
        str_s=[]
        for i in range(spc):
            str_s.append(random.choice(self.special_chars))


        pswrd = str_u+str_l+str_n+str_s
        """
        The above code is shuffling the elements of the list `pswrd` randomly using the
        `shuffle()` function from the `random` module in Python.
        """
        random.shuffle(pswrd)

        return ("".join(pswrd))


class interface(allChars):
    """
    This is a Python class that creates a graphical user interface for a login and signup system.
    """
    def LOGIN(self):
        login_window = tkinter.Tk()
        screen_width  = login_window.winfo_screenwidth()
        screen_height = login_window.winfo_screenheight()
        login_window.state('zoomed')
        login_window.title("Login")

        bg_login = tkinter.PhotoImage(file="login.png")
        bg_Label = tkinter.Label(login_window,
                                 width=screen_width,
                                 height=screen_height,
                                 image=bg_login)

        bg_Label.pack()

        def goto_signup():
            """
            This function closes the login window and opens the signup window.
            """
            login_window.destroy()
            self.SIGNUP()

        def verify():
            """
            This function verifies the login credentials entered by the user and displays
            appropriate messages based on the validity of the credentials.
            """
```

```
    get_username = username_entry.get()
    get_password = password_entry.get()
    if len(get_username)==0:
        info2["text"] = "Enter Username"
    else:
        if get_username.encode() in accountsDB:
            if get_password == accountsDB[get_username].decode():
                info2['text'] = "Login successful"
            else:
                if len(get_password)==0:
                    info2["text"] = "Enter Password"
                else:
                    info2['text'] = "Invalid Password!!"
        else:
            info2["text"] = "Invalid Username!!"

def forgot():
    """
    The function checks if a given username exists in a database.
    """
    search_username = username_entry.get()
    if search_username.encode() in accountsDB:
        def show():
            info2["text"] = f"Your password is {accountsDB[search_username].decode()}"
        def fetch():
            info2["text"] = "Fetching your password..."
            info2.after(8000,show)
        def verified():
            info2["text"] = "Verified"
            info2.after(1000,fetch)
        def verify_email():
            info2["text"] = "Verification code is sent to your E-mail"
            info2.after(3500,verified)
        verify_email()
    else:
        if len(search_username)==0:
            info2["text"] = "Enter Username"
        else:
            info2["text"] = "Invalid Username!!"

def saved():
    """
    This is a GUI function for a login page with options to remember password, forgot
    password,login , and sign up.
    """
```

```
        password_entry.delete(0, 'end')
        find_username = username_entry.get()
        if len(find_username) == 0:
            info2["text"] = "Enter Username/Password"
        else:
            fptr = open(r'remember_me.txt', 'r')
            username_list = fptr.readlines()
            fptr.close()
            if find_username + '\n' in username_list:
                pswrd = accountsDB[find_username]
                password_entry.insert(0,pswrd.decode())
            else:
                info2['text'] = "Password not saved for this username"

# The above code is creating a GUI login window using the tkinter module in Python. It
# includes labels for username and password, entry fields for the user to input their login
# information, buttons for logging in, remembering the user's login information, and signing
# up for a new account. It also includes frames for organizing the layout of the window.

    frame = tkinter.Frame(login_window,
                          width=400, height=500,
                          bg="black")

    frame.place(x=1000,y=100)

    loginL = tkinter.Label(frame,
                           text="LOGIN",
                           fg="white", bg="black",
                           font=("Times New Roman", 50, 'bold'),
                           pady=65)

    loginL.pack(anchor="center")

    frame1 = tkinter.Frame(frame,
                           width=400, height=400,
                           bg="black")

    frame1.pack(padx=10, pady=20)

    frame2 = tkinter.Frame(frame,
                           width=400,
                           height=200,
                           bg="black")
```

```
frame2.pack(padx=10, pady=20)

info2 = tkinter.Label(frame,
                      fg="yellow", bg="black",
                      width=32,
                      font=("Times New Roman", 14, 'italic'))

info2.pack(anchor="center")

username = tkinter.Label(frame1,
                         text="Username ",
                         height=1,
                         font=("Times of Roman", 14, 'bold'))

username.grid(row=0, column=0, pady=10, padx=5)

username_entry = tkinter.Entry(frame1,
                               width=30,
                               bd=3,
                               font=(15))

username_entry.grid(row=0, column=1, pady=10, padx=15, sticky='e')

password = tkinter.Label(frame1,
                         text="Password ",
                         height=1,
                         font=("Times of Roman", 14, 'bold'))

password.grid(row=1, column=0, pady=10, padx=5)

password_entry = tkinter.Entry(frame1,
                               width=30,
                               font=(15),
                               bd=3)

password_entry.grid(row=1, column=1, pady=10, padx=15, sticky='e')

savedButton = tkinter.Button(frame2,
                             text="Remembered me?",
                             padx=0, pady=0,
                             bd=0, borderwidth=2,
                             font=("Times New Roman", 13, 'italic'),
                             command=saved)
```

```
    savedButton.place(x=55, y=19)

    ForgetButton = tkinter.Button(frame2,
                                  text="Forgot password?",
                                  padx=0, pady=0,
                                  bd=0, borderwidth=2,
                                  font=("Times New Roman", 13, 'italic'),
                                  command=forgot)

    ForgetButton.place(x=215, y=19)

    loginButton1 = tkinter.Button(frame2,
                                  text="Login",
                                  width=30,
                                  padx=0, pady=0,
                                  bd=0,
                                  borderwidth=2,
                                  font=("Times New Roman", 15, 'italic'),
                                  command=verify)

    loginButton1.place(x=38, y=67)

    signupButton = tkinter.Button(frame2,
                                  text="Don't have account? Sign Up",
                                  width=30,
                                  padx=0, pady=0,
                                  bd=0, borderwidth=2,
                                  font=("Times New Roman", 15, 'italic'),
                                  command=goto_signup)

    signupButton.place(x=38, y=118)

    login_window.mainloop()


def SIGNUP(self):
    """
    This function creates a sign-up window with a background image and a button to go to
    the login window.
    """
    signUP_window = tkinter.Tk()
    screen_width = signUP_window.winfo_screenwidth()
    screen_height = signUP_window.winfo_screenheight()
    signUP_window.state('zoomed')
    signUP_window.title("Create Account")
```

```
signup = tkinter.PhotoImage(file="signup.png")
bg_signup = tkinter.Label(signUP_window,
                          width=screen_width,
                          height=screen_height,
                          image=signup)
bg_signup.pack()

"""
The function checks if the entered username and password meet certain criteria and
saves the username to a file if it is not already saved.
"""
def gotoLogin():
    signUP_window.destroy()
    self.LOGIN()

def remember():
    """
    This function checks if a username and password meet certain criteria and saves the
    username to a file if it is not already present.
    """
    find_username = username_entry.get()
    pswrd = password_entry.get()
    if len(find_username)==0 or len(pswrd)==0:
        info["text"] = "Enter username/password"
    elif len(pswrd)<8 or len(pswrd)>16:
        info["text"] = "Password must be of 8-16 characters"
    else:
        fptr = open(r'remember_me.txt','r')
        username_list = fptr.readlines()
        fptr.close()
        if find_username+'\n' in username_list:
            info['text'] = "Password already saved"
        else:
            fptr = open(r'remember_me.txt', 'a')
            fptr.write(find_username+'\n')
            fptr.close()
            info['text'] = "Password saved successfully"

def createAccount():
    """
    This function creates a new account by taking user inputs for name, email, username,
    and password, and checks for validity before adding it to the accounts database.
    """
    global access_to_save
    info["text"] = ""
```

```
    get_name = name_entry.get()
    get_email = email_entry.get()
    get_username = username_entry.get()
    get_password = password_entry.get()
    if len(get_name)==0:
        info["text"] = "Enter your Name"
    else:
        if len(get_email)==0:
            info["text"] = "Enter your E-mail"
        else:
            if len(get_username)==0:
                info["text"] = "Enter username"
            else:
                if get_username.encode() in accountsDB:
                    info["text"] = "Username already exists...\nChoose a different username"
                else:
                    if 8<=len(get_password)<=16:
                        accountsDB[get_username] = get_password
                        if access_to_save:
                            remember()
                            access_to_save=False
                        info["text"] = "Account created successfully... Please Login"
                        accountButton["state"] = "disabled"
                        suggestButton['state'] = 'disabled'
                    else:
                        info["text"] = "Password must be of 8-16 characters"

def suggestpass():
    """
    This function generates a random password and inserts it into a password entry field
    while also setting a global variable to allow for saving the password.
    """
    global access_to_save
    password_entry.delete(0,'end')
    generated_password = self.generate()
    password_entry.insert(0,generated_password)
    suggestButton["text"] = "Suggest another password"
    access_to_save = True


"""
The below code is creating a GUI interface for a sign-up page using the tkinter module
 in Python. It includes labels, entry fields, buttons, and frames for organizing the layout.
 The code also includes functions for creating an account, remembering the user,
 suggesting a password, and navigating to the login page. Finally, the code closes the
  database connection.
"""
```

```
frame = tkinter.Frame(signUP_window,
                      width=500, height=400,
                      bg="#2e0742")

frame.place(x=500,y=75)

signupL = tkinter.Label(frame,
                        text="SIGN UP",
                        fg="white",
                        bg="#2e0742",
                        font=("Times New Roman", 50, 'bold'),
                        pady=65)

signupL.pack(anchor="center")



frame1 = tkinter.Frame(frame,
                       width=500,
                       bg="#2e0742")

frame1.pack(padx=10, pady=0)

frame2 = tkinter.Frame(frame,
                       width=400, height=150,
                       bg="#2e0742")

frame2.pack(padx=10, pady=5)

name = tkinter.Label(frame1,
                     text="Name ",
                     height=1,width=8,
                     font=("Times of Roman", 14, 'bold'))

name.grid(row=0, column=0, pady=10, padx=5)

name_entry = tkinter.Entry(frame1,
                           width=30,
                           bd=3,
                           font=(15))

name_entry.grid(row=0, column=1, pady=10, padx=15)
```

```
email = tkinter.Label(frame1,
                      text="Email ",
                      height=1, width=8,
                      font=("Times of Roman", 14, 'bold'))

email.grid(row=1, column=0, pady=10, padx=5)

email_entry = tkinter.Entry(frame1,
                            width=30,
                            bd=3,
                            font=(15))

email_entry.grid(row=1, column=1, pady=10, padx=15)

username = tkinter.Label(frame1,
                         text="Username ",
                         height=1,
                         font=("Times of Roman", 14, 'bold'))

username.grid(row=2, column=0, pady=10, padx=5)

username_entry = tkinter.Entry(frame1,
                               width=30, bd=3,
                               font=(15))

username_entry.grid(row=2, column=1, pady=10, padx=15)

password = tkinter.Label(frame1,
                         text="Password ",
                         height=1,
                         font=("Times of Roman", 14, 'bold'))

password.grid(row=3, column=0, pady=10, padx=5)

password_entry = tkinter.Entry(frame1,
                               width=30,
                               font=(15),
                               bd=3)

password_entry.grid(row=3, column=1, pady=10, padx=15)

rememberButton = tkinter.Button(frame1,
                               text="Remember Me",
                               padx=0, pady=0,
```

```
                        bd=0, borderwidth=2,
                        font=("Times New Roman", 11, 'italic'),
                         command=remember)

rememberButton.grid(row=4, column=1, sticky='w', padx=15)

suggestButton = tkinter.Button(frame1,
                        text="Suggest a password",
                        padx=0, pady=0,
                        bd=0, borderwidth=2,
                        font=("Times New Roman", 11, 'italic'),
                        command=suggestpass)

suggestButton.grid(row=4,column=1,sticky='e',padx=15)

info = tkinter.Label(frame,
                bg="#2e0742",fg="yellow",
                width=50,height=2,
                font=("Times New roman", 15, "italic"))

info.pack(pady=15)

accountButton = tkinter.Button(frame2,
                        text="Create Account",
                        width=30,
                        padx=0, pady=0,
                        bd=0, borderwidth=2,
                        font=("Times New Roman", 15, 'italic'),
                        command=createAccount)

accountButton.place(x=30, y=50)

loginButton = tkinter.Button(frame2,
                        text="Already have account? Login",
                        width=30,
                        padx=0, pady=0,
                        bd=0, borderwidth=2,
                        font=("Times New Roman", 15, 'italic'),
                        command=gotoLogin)

loginButton.place(x=30, y=100)

signUP_window.mainloop()
```

```
# The code is calling the `SIGNUP()` method of an object named `start` which is an instance
# of a class
# that has an interface. This suggests that the code is part of a larger program that involves #
# user
# account creation or registration. After the `SIGNUP()` method is executed, the # #
#`accountsDB` database
# is closed.

start = interface()
start.SIGNUP()
accountsDB.close()
```