

RANDOM PASSWORD GENERATOR

AIM:

In this article, how to create a random password generator in Python. Using a strong password is necessary, rather recommended. As per Cyber Security Experts, a password must be a combination of alphabets, digits, and symbols, also making sure that the combination doesn't make a sensible word or combination which can be easily guessed using the social engineering method.

A random password on the other hand of a minimum of 8 characters in length even after applying advanced security breach methods like Brute Force attacking it is hard to breach or crack or may take a very long time to breach it.

SYSTEM REQUIREMENTS:

Before running the random password generator, ensure that your system meets the following requirements.

Python Installation:

Python 3.x should be installed on your local machine. You can download Python from the official website: [Python official website](https://www.python.org/).

Python Libraries:

The script relies on the following Python libraries, which can be installed using pip:

PIL (Python Image Library):

Used for image creation and manipulation. Used to generate Background image and icon image.

You Can Install These Libraries Using The Following Commands:

```
pip install tkinter
```

```
pip install pyperclip
```

```
pip install random
```

```
pip install PIL
```

PROCEDURE:

CODE FOR RANDOM PASSWORD GENERATOR:

Generating a strong random password of recommended length is a difficult task, obviously not more than remembering it. But here, we will code a python script to generate a random password.

1. Install And Import The Required Modules:

We begin with installing the required libraries using the pip package manager. Enter the below commands in your command line or terminal to install the modules.

We need to install Tkinter to make our password generator based pyperclip library to infuse the Copy to clipboard functionality.

We need to install PIL to make our background image fix based PIL library.

```
pip install tkinter
```

```
pip install pyperclip
```

```
pip install random
```

```
pip install PIL
```

After installing the required libraries from the terminal, we now move to our Python file to code. We start with importing the libraries as:

where random is used to generate random characters from a given list and Int is used to get characters/text.

Import Modules:

```
import random
```

```
import pyperclip
```

```
from tkinter import *
```

```
from tkinter.ttk import *
```

```
from PIL import ImageTk, Image
```

```
import tkinter as tk
```

```
import PIL
```

2. Initialize Tkinter Window:

As a next step, we initialize our window using the Tkinter module.

```
root = Tk()
```

```
var = IntVar()
```

```
var1 = IntVar()
```

```
root.geometry("1500x1500") #size of the window by default
```

title of our window:

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

3. Code The Elements:

To select the length of the password,

- We use the Label method to generate a label of text to define the purpose of the input field we wanted for the length of the password.
- The spinbox method is used to take the input against a value selector, ranging from 4 to 32, which you can change as per need, this defines the minimum and maximum length of the password.

We code a Generate Password Button, on which we click to generate a random password:

- We give our button some styling, along with the name – Generate Password. We use the command, which shows which function (here, generate function) would run on the click (key press) of the button.

After adding so, we now add the output elements in our code.

- We again add a label to show what is being displayed, we add a label of “Random Generated Password” with some styling.
- Yet again, we add an Entry widget to create an input field, this is intended to display our Randomly Generated Password
- The Generate Password widgets is used to show text from the value of Random_password Entry , which hold the randomly generated password and the password shows='*'.
- We now add the Add to copy button in our code to display, the command widget of it shows that the copy1 function would run upon click on this button.
- We now add the Add to view button in our code display, the command widgets of it shows that the view function would run upon click on this button then the password is shown in view entry.
- Then now add the Add to clear button in our code display, the command widgets of it shows that the clear function would run upon click on this button then the password is clear on view entry.

4. Random Password Function

Having finished the front-end part, we now move to our code's backend, where we add functionality to our buttons

- We code the most important function of this code, which is for Random Password Generation, we do so as in the code:

5. Copy Password Function

As a final step, we add the Copy Password function in our code as:

- We use the copy method of the pyperclip library to save the password as copied to our system. When will you use to click copy button then the copy1 function run and copy the password.

6. Background Image Function:

- The first step is to install the Pillow library, which is an extension of the Python Imaging Library (PIL). Pillow allows us to work with images and perform various operations such as resizing, cropping, and more. To install the Pillow library
- Next, we need to import Tkinter, along with Pillow's Image and ImageTk modules. To do this, add the following lines at the beginning of your Python script:
- To load and display an image on the window, follow these steps:
 - a. Open the image using the Image.open() function from the Pillow library.
 - b. Convert the opened image into a PhotoImage object using the ImageTk.PhotoImage() function.
 - c. Create a Label widget and set its image parameter to the PhotoImage object created in the previous step.
 - d. Add the Label widget to the window using the place methods.

7. Icon Image Function:

- The first step is to install the Pillow library, which is an extension of the Python Imaging Library (PIL). Pillow allows us to work with images and perform various operations such as resizing, cropping, and more. To install the Pillow library
- Next, we need to import Tkinter as tk, along with iconphoto and tk.PhotoImage modules. To do this, add the following lines at the beginning of your Python script:
- To load and display an image on the window.

Conclusion

With these steps, we have successfully created a 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

PROGRAM

```
import random
import pyperclip
from tkinter import *
from tkinter.ttk import *
from PIL import ImageTk, Image
import tkinter as tk
import PIL

def low():
    entry.delete(0, END)

    length = var1.get()

    lower = "abcdefghijklmnopqrstuvwxyz"
    upper =
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"
    digits =
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz01234
56789 !@#$%^&*()"
    password = ""

    if var.get() == 1:
        for i in range(0, length):
            password = password + random.choice(lower)
        return password

    elif var.get() == 0:
        for i in range(0, length):
            password = password + random.choice(upper)
        return password
```



```
elif var.get() == 3:
    for i in range(0, length):
        password = password + random.choice(digits)
    return password
else:
    print("Please choose an option")
```

```
def generate():
    password1 = low()
    entry.insert(10, password1)
```

```
def copy1():
    random_password = entry.get()
    pyperclip.copy(random_password)
```

```
def view():
    random_password = entry.get()
    view_entry.insert(10, random_password)
```

```
def clear():
    view_entry.delete(0, END)
```

```
root = Tk()
var = IntVar()
var1 = IntVar()

root.geometry("1500x1500")
root.title("Password Generator")

bg = ImageTk.PhotoImage(file="C:\\Users\\Lab pc- 1\\Pictures\\Saved
Pictures\\pexels-eberhard-grossgasteiger-691668.jpg")

canvas = Canvas(root, width=700, height=3500)
canvas.pack(fill=BOTH, expand=True)

canvas.create_image(0, 0, image=bg, anchor='nw')

def resize_image(e):
    global image, resized, image2

    image = Image.open("C:\\Users\\Lab pc- 1\\Pictures\\Saved
Pictures\\pexels-eberhard-grossgasteiger-691668.jpg")

    resized = image.resize((e.width, e.height),
PIL.Image.Resampling.LANCZOS)

    image2 = ImageTk.PhotoImage(resized)
    canvas.create_image(0, 0, image=image2, anchor='nw')

root.bind("<Configure>", resize_image)
```

```
root.iconphoto(False, tk.PhotoImage(file='C:\\Users\\Lab pc-1\\Pictures\\Saved Pictures\\UKESH PROJECT\\password-8-72964.png'))
```

```
title_label = Label(root, text="PASSWORD GENERATOR", font=("lucida 20 bold italic"))  
title_label.place(x=460,y=30)
```

```
Random_password = Label(root, text="Password",font=('Ubuntu Mono',12))  
Random_password.place(x=420,y=150)  
entry = Entry(root, show='*')  
entry.place(x=550,y=150)
```

```
c_label = Label(root, text="Length",font=('Ubuntu Mono',12))  
c_label.place(x=420,y=250)
```

```
copy_button = Button(root, text="Copy", command=copy1)  
copy_button.place(x=730,y=149)  
generate_button = Button(root, text="Generate Password",  
command=generate)  
generate_button.place(x=550,y=450)
```

```
view_button = Button(root, text="view",command=view)  
view_button.place(x=420,y=350)  
view_entry = Entry(root)  
view_entry.place(x=550,y=352)
```

```
clear_button = Button(root, text="clear",command=clear)  
clear_button.place(x=730,y=350)
```

```
radio_low = Radiobutton(root, text="Low", variable=var, value=1)
radio_low.place(x=730,y=250)
radio_middle = Radiobutton(root, text="Medium", variable=var,
value=0)
radio_middle.place(x=790,y=250)
radio_strong = Radiobutton(root, text="Strong", variable=var, value=3)
radio_strong.place(x=870,y=250)
combo = Combobox(root, textvariable=var1)
```

```
combo['values'] = (8, 9, 10, 11, 12, 13, 14, 15, 16,
                  17, 18, 19, 20, 21, 22, 23, 24, 25,
                  26, 27, 28, 29, 30, 31, 32, "Length")
combo.current(0)
combo.bind('<<ComboboxSelected>>')
combo.place(x=550,y=250)
```

```
root.mainloop()
```

PASSWORD GENERATOR WINDOW:

Windows taskbar: Type here to search, icons for File Explorer, Edge, Mail, etc., 11:01 AM 10/31/2023

PASSWORD GENERATOR

Password

Length ☐ Low ☒ Medium ☐ Strong

PASSWORD GENERATOR SELECT LENGTH OPTION:

Windows taskbar: Type here to search, icons for File Explorer, Edge, Mail, etc., 11:03 AM 10/31/2023

PASSWORD GENERATOR

Password

Length

8

8

9

10

11

12

13

14

15

16

17

☐ Low ☐ Medium ☒ Strong

GENERATE PASSWORD BUTTON:

Password Generator

PASSWORD GENERATOR

Password:

Length: ☐ Low ☒ Medium ☐ Strong

Type here to search 11:02 AM 10/31/2023

SELECT LOW PASSWORD GENERATE OPTION:

Password Generator

PASSWORD GENERATOR

Password:

Length: ☒ Low ☐ Medium ☐ Strong

Type here to search 11:02 AM 10/31/2023

SELECT MEDIUM PASSWORD GENERATE OPTION:

Password Generator

PASSWORD GENERATOR

Password:

Length: ☐ Low ☒ Medium ☐ Strong

Type here to search 11:02 AM 10/31/2023

SELECT STRONG PASSWORD GENERATE OPTION:

Password Generator

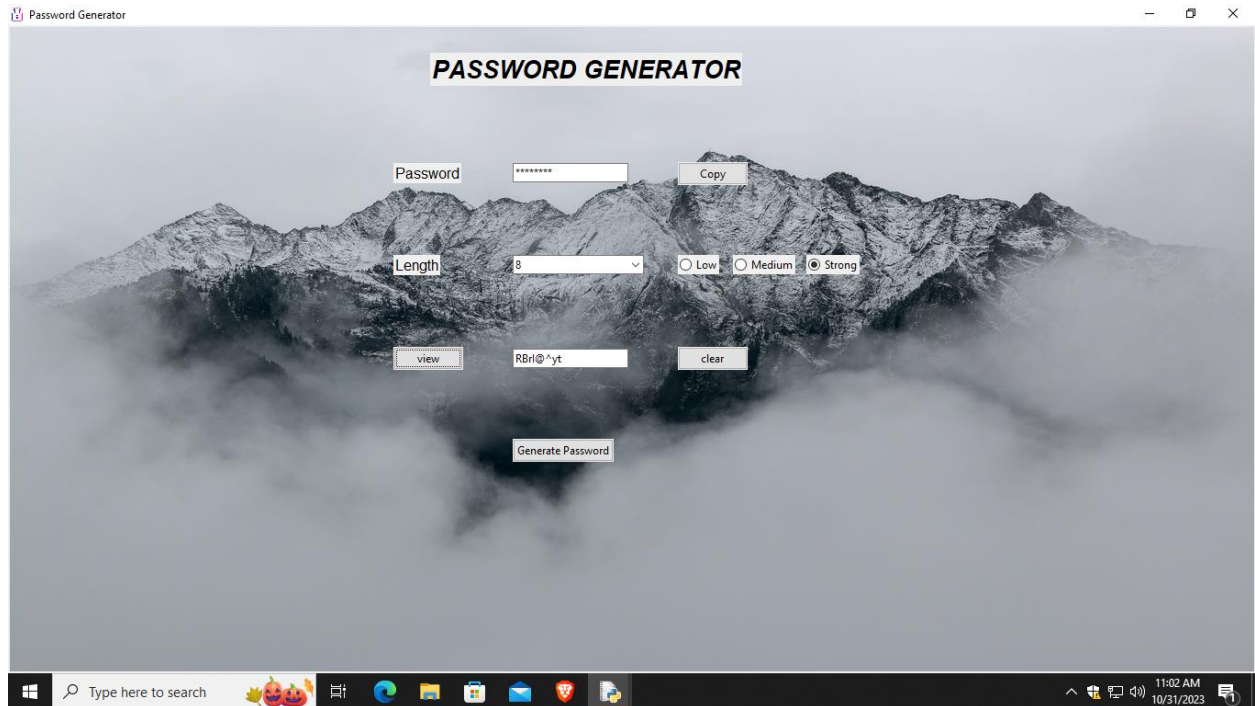
PASSWORD GENERATOR

Password:

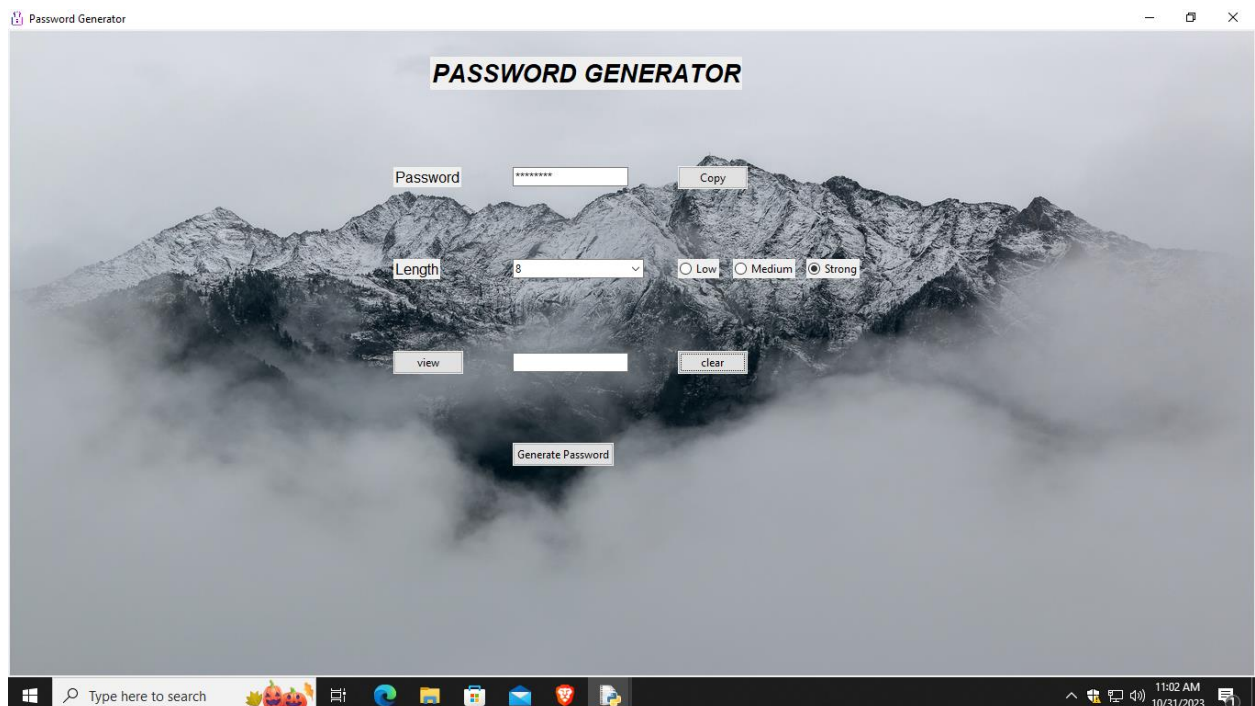
Length: ☐ Low ☐ Medium ☒ Strong

Type here to search 11:02 AM 10/31/2023

PASSWORD GENERATOR VIEW BUTTON:



PASSWORD GENERATOR CLEAR BUTTON:



PASSWORD GENERATOR COPY BUTTON:

