PROJECT REPORT

on

Unique password generator GUI

Submitted by

Sneha
Roll No: 201510022

Tushar Upadhyay
Roll No: 201510023

Vaibhav Sharma
Roll No: 201510024

Vishal Rawat
Roll No: 201510025

Department of Computer Engineering & Applications

Institute of Engineering & Technology

GLA University

Mathura- 281406, INDIA

Department of Computer Engineering and Applications
GLA University, Mathura
17 km. Stone NH#2, Mathura-Delhi Road, P.O. –

Declaration

I hereby declare that the work which is being presented in the Minor Project Report **“Unique password generator GUI”,** in partial fulfillment of the requirements for Project is an authentic record of my own work carried under the supervision of **Mr. Shabir Ali, Assistant Professor, GLA University, Mathura**.

Sign ________________________

Name of Candidate: Sneha

University Roll No.: 201510022

Sign ________________________

Name of Candidate: Tushar Upadhyay

University Roll No.: 201510023

Sign ________________________

Name of Candidate: Vaibhav Sharma

University Roll No.: 201510024

Sign ________________________

Name of Candidate: Vishal Rawat

University Roll No.: 201510025

Certificate

This is to certify that the above statements made by the candidate are correct to the best of my/our knowledge and belief.

____

Project Supervisor
Mr. Shabir Ali
Assistant Professor

Date: 09-12-2021

Table of Contents

Introduction

Overview

A random password generator is software program or hardware device that takes input from a random or pseudo-random number generator and automatically generates a password. Random passwords can be generated manually, using simple sources of randomness such as dice or coins, or they can be generated using a computer.

While there are many examples of “random” password generator programs available on the Internet, generating randomness can be tricky and many programs do not generate random characters in a way that ensures strong security. A common recommendation is to use open source security tools where possible since they allow independent checks on the quality of the methods used. Note that simply generating a password at random does not ensure the password is a strong password, because it is possible, although highly unlikely, to generate an easily guessed or cracked password. In fact, there is no need at all for a password to have been produced by a perfectly random process: it just needs to be sufficiently difficult to guess.

Objective

With growing technology, everything has relied on data and securing these data is the main concern. Passwords are meant to keep the data safe that we upload on the Internet.

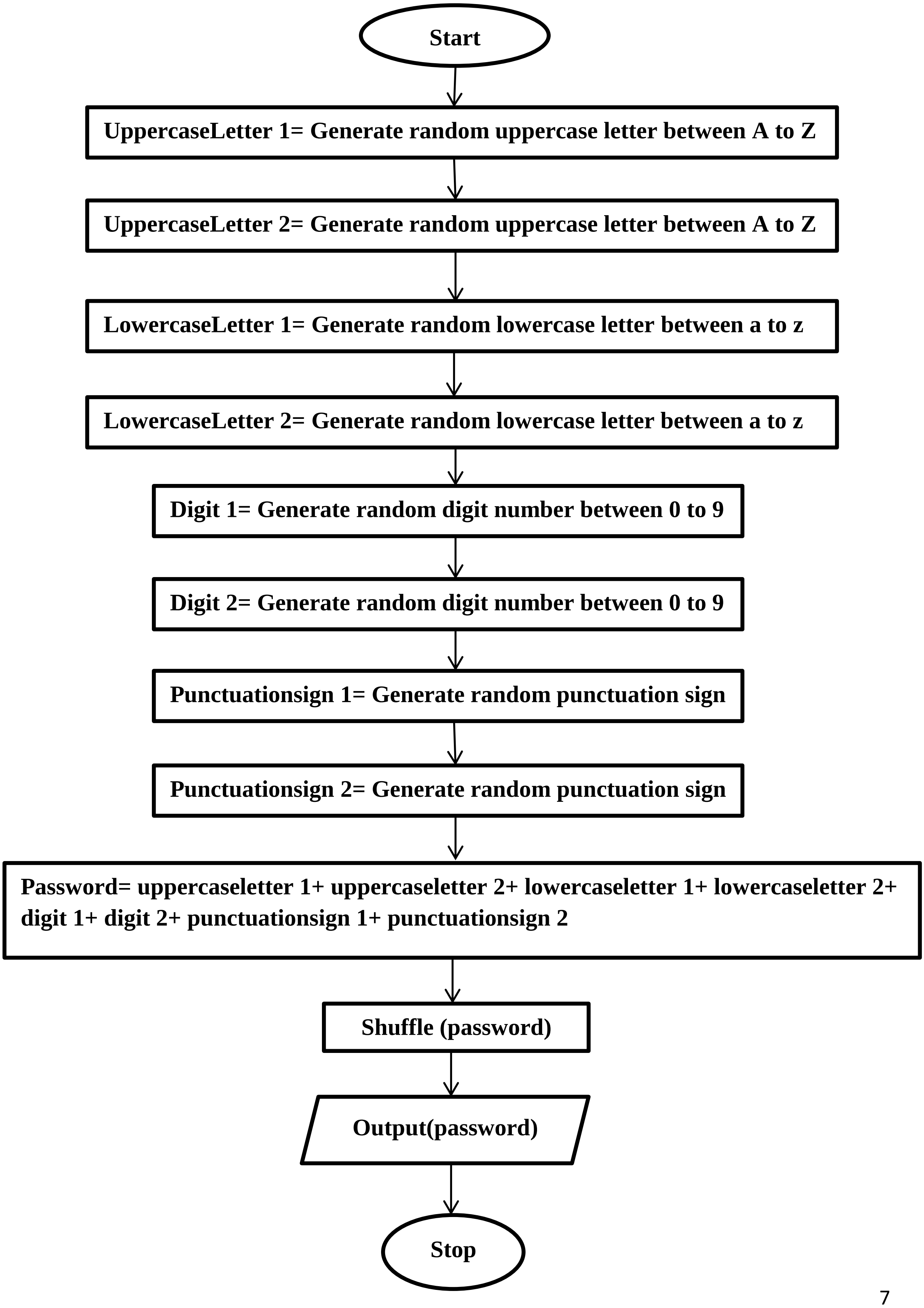
An easy password can be hacked easily and all the personal information can be misused. In order to prevent such things and keep the data safe, it is quite necessary to keep our passwords very strong.

Let’s create a simple application which can randomly generate strong passwords using Python Tkinter module.

This application can generate random password, with the combination of letters, numeric, and special characters. One can mention length of the password based on requirement and can also select the strength of the password.

Project design

Flow chart


Start
UppercaseLetter 1= Generate random uppercase letter between A to Z
UppercaseLetter 2= Generate random uppercase letter between A to Z
LowercaseLetter 1= Generate random lowercase letter between a to z
LowercaseLetter 2= Generate random lowercase letter between a to z
Digit 1= Generate random digit number between 0 to 9
Digit 2= Generate random digit number between 0 to 9
Punctuationsign 1= Generate random punctuation sign
Punctuationsign 2= Generate random punctuation sign
Password= uppercaseletter 1+ uppercaseletter 2+ lowercaseletter 1+ lowercaseletter 2+ digit 1+ digit 2+ punctuationsign 1+ punctuationsign 2
Shuffle (password)
Output(password)
Stop

Project implementation

Explanation

So before starting the actual coding let's first understand some theory. In this password generator GUI application, we are going to use two 3 modules. Firstly, the Tkinter module for creating an application window. Secondly, pyperclip module for copying the generated password. Now, remember these two modules are not inbuilt modules. So you have to install them using the pip install command. The third and final module we are going to use is a random module to generate the random password finally. The quick logic for this GUI application is going to be like this: Firstly, We will create import the necessary modules. Then we will create the application window. After that, we will generate a random password and copy it. Afterward, In the end, we will create some buttons to make our password generator application more interactive for the user. Because after all, it's a GUI.

Understanding the Code of GUI Application

Firstly, Let's understand the coding now with some technical terms. Here, we are going to explain the code stepwise. So that if you have some knowledge of python you will be able to make this password generator application on your own by just reading this section.

Step1: Firstly, Import necessary modules (Tkinter, pyperclip and random)

Step2: Then, Initialize Tkinter using Tk() method

Step3: After that, Create an application window for GUI application using the geometry method of Tkinter

Step4: Declare 2 variable. One for storing the generated password and another one for taking password length input from user.

Step5: Then, Define the generate password method.

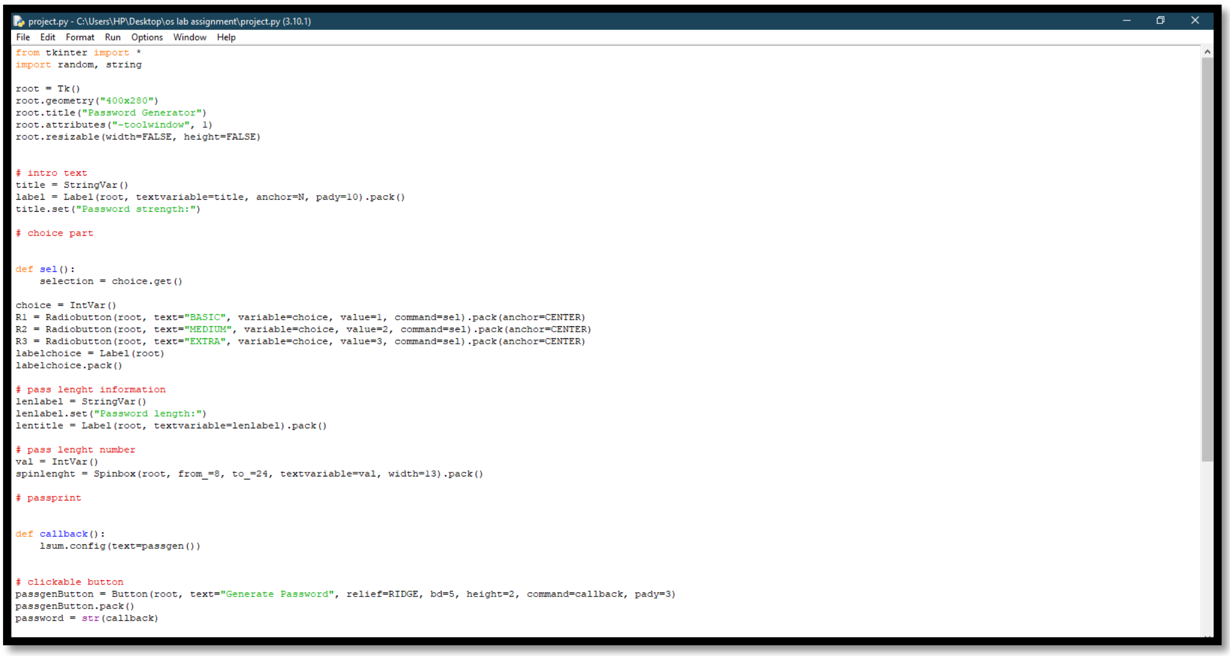
Step6: Define the copy to clipboard method.

Step7: Then, Create Button widgets and set commands as per the above defined methods.

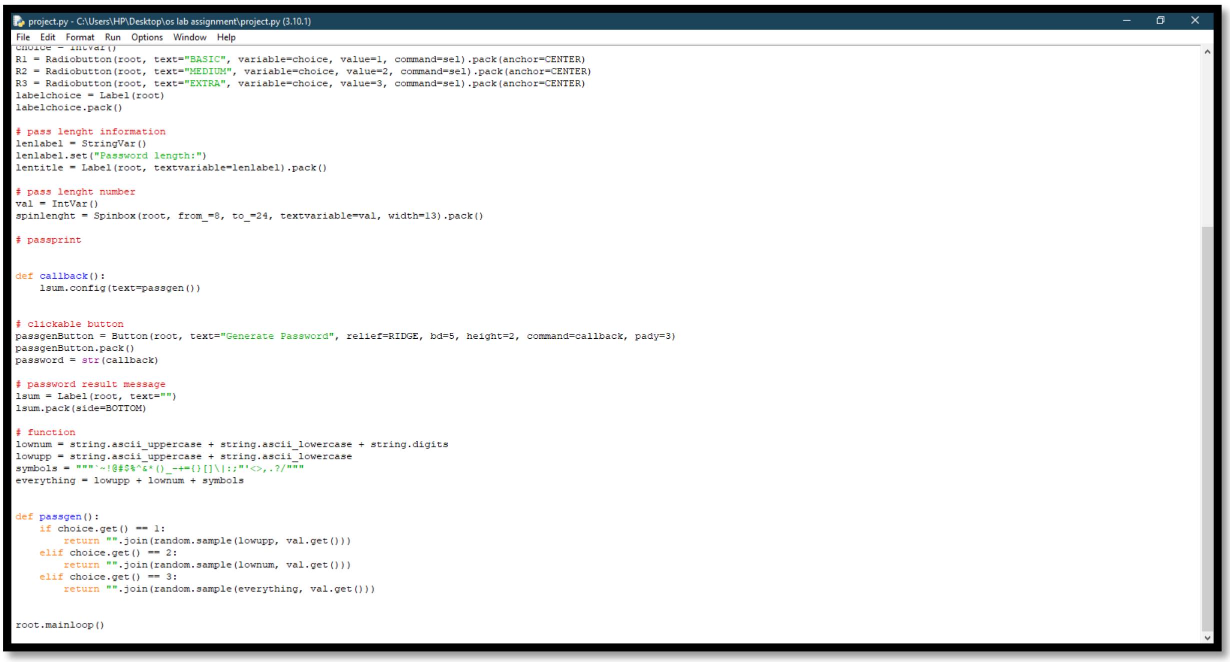
Step8: Finally, Run infinite mainloop to run the application.

Project implementation and user interface

Source code

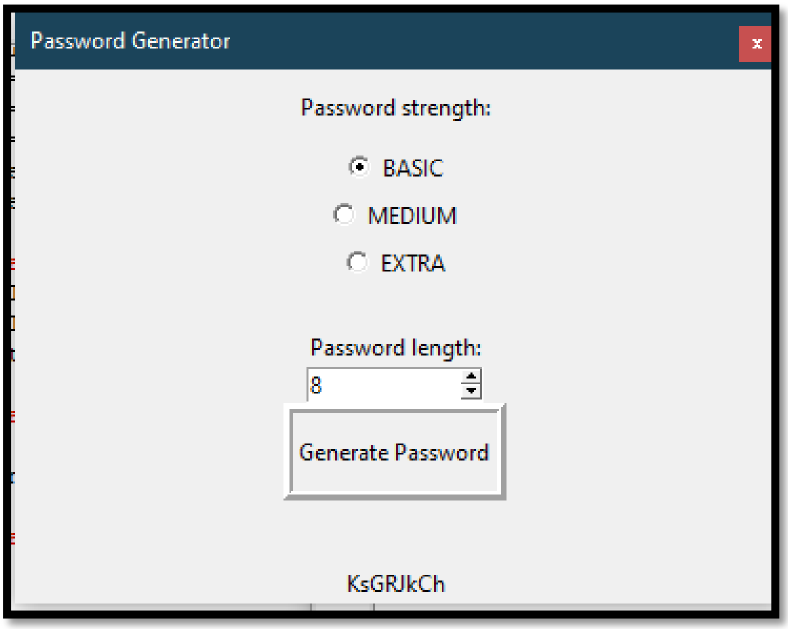

project.py - C:\Users\HP\Desktop\os lab assignment\project.py (3.10.1)
File Edit Format Run Options Window Help

```
from tkinter import *
import random, string

root = Tk()
root.geometry("400x280")
root.title("Password Generator")
root.attributes("-toolwindow", 1)
root.resizable(width=FALSE, height=FALSE)


# intro text
title = StringVar()
label = Label(root, textvariable=title, anchor=N, pady=10).pack()
title.set("Password strength:")

# choice part


def sel():
    selection = choice.get()

choice = IntVar()
R1 = Radiobutton(root, text="BASIC", variable=choice, value=1, command=sel).pack(anchor=CENTER)
R2 = Radiobutton(root, text="MEDIUM", variable=choice, value=2, command=sel).pack(anchor=CENTER)
R3 = Radiobutton(root, text="EXTRA", variable=choice, value=3, command=sel).pack(anchor=CENTER)
labelchoice = Label(root)
labelchoice.pack()

# pass lenght information
lenlabel = StringVar()
lenlabel.set("Password length:")
lentitle = Label(root, textvariable=lenlabel).pack()

# pass lenght number
val = IntVar()
spinlenght = Spinbox(root, from_=8, to_=24, textvariable=val, width=13).pack()

# passprint


def callback():
    lsum.config(text=passgen())


# clickable button
passgenButton = Button(root, text="Generate Password", relief=RIDGE, bd=5, height=2, command=callback, pady=3)
passgenButton.pack()
password = str(callback)
```


project.py - C:\Users\HP\Desktop\os lab assignment\project.py (3.10.1)
File Edit Format Run Options Window Help

```
choice = IntVar()
R1 = Radiobutton(root, text="BASIC", variable=choice, value=1, command=sel).pack(anchor=CENTER)
R2 = Radiobutton(root, text="MEDIUM", variable=choice, value=2, command=sel).pack(anchor=CENTER)
R3 = Radiobutton(root, text="EXTRA", variable=choice, value=3, command=sel).pack(anchor=CENTER)
labelchoice = Label(root)
labelchoice.pack()

# pass lenght information
lenlabel = StringVar()
lenlabel.set("Password length:")
lentitle = Label(root, textvariable=lenlabel).pack()

# pass lenght number
val = IntVar()
spinlenght = Spinbox(root, from_=8, to_=24, textvariable=val, width=13).pack()

# passprint


def callback():
    lsum.config(text=passgen())


# clickable button
passgenButton = Button(root, text="Generate Password", relief=RIDGE, bd=5, height=2, command=callback, pady=3)
passgenButton.pack()
password = str(callback)

# password result message
lsum = Label(root, text="")
lsum.pack(side=BOTTOM)

# function
lownum = string.ascii_uppercase + string.ascii_lowercase + string.digits
lowupp = string.ascii_uppercase + string.ascii_lowercase
symbols = """`~!@#$%^&*()_-+={}[]\|:;"'<>,.?/"""
everything = lowupp + lownum + symbols


def passgen():
    if choice.get() == 1:
        return "".join(random.sample(lowupp, val.get()))
    elif choice.get() == 2:
        return "".join(random.sample(lownum, val.get()))
    elif choice.get() == 3:
        return "".join(random.sample(everything, val.get()))


root.mainloop()
```

Output



Password Generator

Password strength:

● BASIC

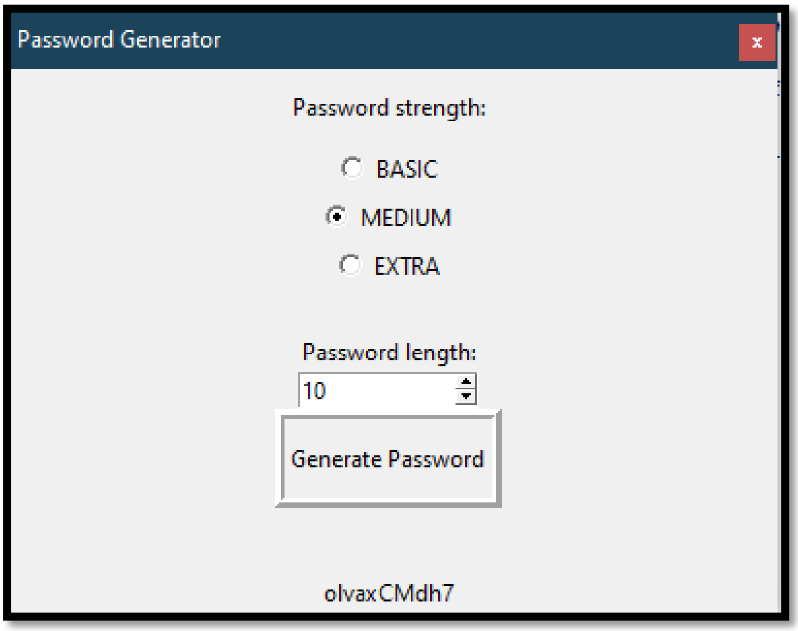
○ MEDIUM

○ EXTRA

Password length:

8

KsGRJkCh



Password Generator

Password strength:

○ BASIC

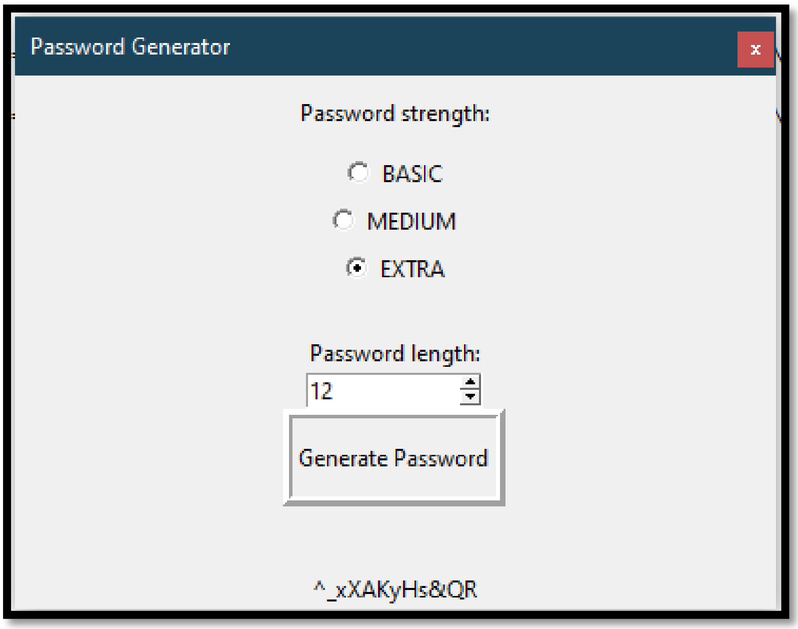
● MEDIUM

○ EXTRA

Password length:

10

olvaxCMdh7



Password strength:

○ BASIC

○ MEDIUM

● EXTRA

Password length:

12

^_xXAKyHs&QR

Code working

```
from tkinter import *
from tkinter.ttk import Combobox
import random
```

Tkinter is the name of the GUI library in Python.

The tkinter.ttk module provides access to the Tk themed widget set and import Combobox. A combo box is a GUI feature and combo box combines a drop-down box, list box or provide the user to select the input.

```
Screen = Tk()
screen.title(“Password Generator”)
screen.geometry(‘600x400’)
screen.configure(background =”bisque”)
```

Tk class is used to create a root window. The Tk function provides a window of GUI as well as provides so many features like setting the title, set the geometry of the GUI window.

In these steps, we are giving the title of the GUI root window. The title is defined about the project in one line. Tkinter provides many methods, one of them is the geometry() method.

Here in this line of code, we are working on the background color of the Tkinter GUI window.

```
Def gen():
global sc1
sc1.set(“”)
passw=””
length=int(c1.get())
lowercase=’abcdefghijklmnopqrstuvwxyz’
uppercase=’ABCDEFGHIJKLMNOPQRSTUVWXYZ’+lowercase
mixs=’0123456789’+lowercase+uppercase+’@#$%&*’
```

Now coming to the next line of the code where we define a function. Here we define the global variable sc1. There are two types of variables including in the programming, the first is the local variable and the second is the global variable.

In this line of code, we used the set method on the sc1. Here we set the sc1 value as null. Then we define the passw variable with a null value. In this line of code, we define the length variable to hold the value of c1 and c1 value type as integer. Now we define the lowercase variable to hold some string value. Next, we define the uppercase variable to hold some string value. And at the last line of code, we define the mix.

```
If c2.get()==’Low Strength’:
for i in range(0,length):
```

```
passw=passw+random.choice(lowercase)
sc1.set(passw)
elif c2.get()==’Medium Strength’:
for i in range(0,length):
passw=passw+random.choice(uppercase)
sc1.set(passw)
elif c2.get()==’High Strength’:
for i in range(0,length):
passw=passw+random.choice(mixs)
sc1.set(passw)
```

In this line of code, we define the if condition for assigning some condition in our project.

So coming to the code we define the if condition where the condition is: c2.get()==’Low Strength’. Here the condition goes true then the program jumps to the next step of the code. Then we define the for a loop. Next, we define the passw variable to hold passw + random choice value from lowercase. Again we use the set function and assign the new value of the sc1 variable from the passw variable.

Now we use the elif condition in the code. First comprehend, what is elif? So the elif is the keyword in a python programming language. Elif is used when the previous if conditions are not true, the program jumps to the elif condition”. Here we define the elif condition where the condition is: c2.get()==’Medium Strength’.

Define the for a loop. Now again we use the set function and set the value of the sc1 variable from the passw variable.

In this line of code, we again define the elif condition where the condition is: c2.get()==’High Strength’. When the condition goes true then the program jumps to the next line of the code. Here we define the for a loop. Now again when the loop is completed the program comes to the next line of code. Here we define the passw variable to hold passw + random choice value from the mix variable. Now again we use the set function to set the sc1 variable value from the passw variable.

```
Sc1=StringVar(‘’)
t1=Label(screen,text=’Automatic Password
Generator’,font=(‘Arial’,25),fg=’red’,background =”bisque”)
t1.place(x=60,y=0)
t2=Label(screen,text=’password:’,font=(‘Arial’,14),background
=”bisque”)
t2.place(x=145,y=90)
```

Now in this line of code, we define the sc1 variable.

Now come to the Designing Part of the GUI window. Here we are using some Label, Entry Box and Button in a project. So let’s start to comprehend the working.

Here we define a Label t1 on the GUI window, where t1 is the name of the Label object. Here we write the text on the label and set font and set bg of Label.

Now we again take one more Label and define a Label t2 and vice-versa. We assign the place of the t1 Label using x or y coordinates values.

```
Il=Entry(screen,font=(‘Arial’,14),textvariable=sc1)
il.place(x=270,y=90)
t3=Label(screen,text=’Length: ‘,font=(‘Arial’,14),background
=”bisque”)
t3.place(x=145,y=120)
t4=Label(screen,text=’Strength:’,font=(‘Arial’,14),background
=”bisque”)
t4.place(x=145,y=155)
c1=Entry(screen,font=(‘Arial’,14),width=10)
c1.place(x=230,y=120)
```

Here we define the Entry Box with the object il on the GUI window and set the font of il Entry and the text variable holds the string value of sc1. Now in this line, we assign the place of the il Entry using x or y coordinates values.

Again we define the Label t3. Next, we assign the place by coordinates values. We define the Label t4. Next, we assign the place by coordinates values. Now again we define Entry c1. Assign the place.

```
C2=Combobox(screen,font=(‘Arial’,14),width=15)
c2[‘values’]=(‘Low Strength’,’Medium Strength’,’High Strength’)
c2.current(1)
c2.place(x=237,y=155)
b=Button(screen,text=’Generate’,font=(‘Arial’,14),fg=’red’,backg
round =”white”,command=gen)
b.place(x=230,y=195)
screen.mainloop()
```

In this line of the code, we are working on the Combobox. Here we define the Combobox c2. Define the values of the Combobox c2 as Low Strength, Medium Strength, High Strength. Here we set the c2 value with the current function. We can also give the place of the Combobox and can assign the place of the c2 Combobox using x or y coordinate values.

Conclusion

Password is used in any application for authentication. Creating a strong password is very important to keep the user's account secure. Any account can be hacked easily if a very simple password is used for the account, such as 12345 or the user's name. A strong password can be created by combining an uppercase letter, lowercase letter, digits, and special symbols. A password generator is a program that is used to generate random passwords. The Password generated by this application is very strong, and it can't be guessed easily by the hacker. It is better to use the Password generated from the password generator for any normal or administrative account to keep the account safe. This password generator program can be implemented in different ways using the python script shown in this project.

References

- **Geeksforgeeks.com**
- **Cleverprogrammer.com**
- **Github.com**
- **Stackoverflow.com**