**RANDOM PASSWORD GENERATOR**

|  |  |
| --- | --- |
| **Name:** | **Aarush Singh Sabharwal**  **(11916973, RK19KBB47)**  **Vasabathula Venkataratnam**  **(11917311, RK19KBB46)** |
| **Subject:** | **INT213** |
| **Group no:** | **16** |

**Table of contents:**

* Acknowledgement
* Declaration
* Introduction (need for a random password generator)
* Data Flow diagram and Github repository of the project
* Description of the project and supporting images
* Results and some tips for a secure password

**Acknowledgement:**

This project is done by us as a partial fulfillment of the CA work assigned to us as a part of the course INT213. We thank our venerable teacher for giving us an opportunity to present our work on the project entitled: ‘RANDOM PASSWORD GENERATOR’.

**Declaration:**

We thereby declare that this work of ours is original and in its true form. This has been carried out under the guidance of our respected teacher, Madam Ankita Wadhawan.

Aarush Singh Sabharwal

Vasabathula Venkataratnam

(Students)

(Lovely Professional University)

**Introduction (need for a random password generator):**

In today’s world, cyber crimes are on a rise. The people with malicious intents are always there, busy finding their ways into systems so as to fulfill their needs. So, what actually is the main reason for our accounts being hijacked? Social engineering attacks are one of the common reasons as we all know. But, what else makes our accounts vulnerable? A weak password! A password that can be brute forced or a password that might be subject to a dictionary/wordlist attack. There are plenty of tools to do so, John the Ripper, Hash cat, aircrack-ng, hydra to name a few. So what can we do? Generate a strong password!

So, what is a strong password? First of all, a strong password should be atleast 8 characters long. Secondly, it should be random. You can even increase the security of a password by using combinations of letters, numbers and special characters. This way, you can give a hard time to the crackers trying to take over your account. There are plenty of tools online to generate strong passwords. You can easily do so, not a big deal! But, what if that website (generating random passwords) stores your password after generating? A data breach on that website would be even worse. Hackers can get all the passwords, use tools to target multiple usernames on social networking websites. Your username and password might be there! So, the best way is to make a Random password Generator yourself and run it on your own machine. The only problem ☹ now is, to remember that password ☺

**Data Flow diagram of the project and supporting images:**

0

Random Password Generator

User inputs password length

Password length module

User

User selects password strength

Result

LEVEL 0

Error

Result

0.4

Length of password

User inputs password length

Could not validate passwd strength

0.1

Validated passwd strength

0.6

Output

0.3

Password generation module

0.5

Success

0.2

Strength of password

LEVEL 1

Password strength module

User selects password strength

Could not validate passwd

Length of password

Processing the request

Validating the password

0.3.1

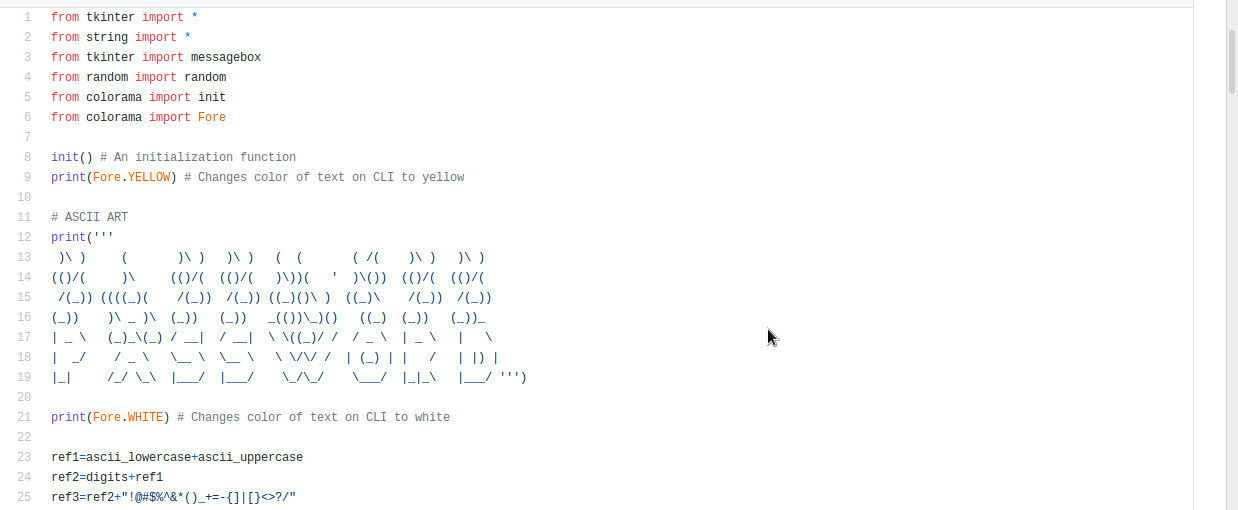
0.3.2

Strength of password

Validated passwd

LEVEL 2

*Github repository* <https://github.com/Aarushsinghsabharwal/passwd_generator.git>

****

Some useful external packages are imported. The purpose of ASCII art is just for beautification. The reference strings for generating passwords of varying strengths are defined.

****

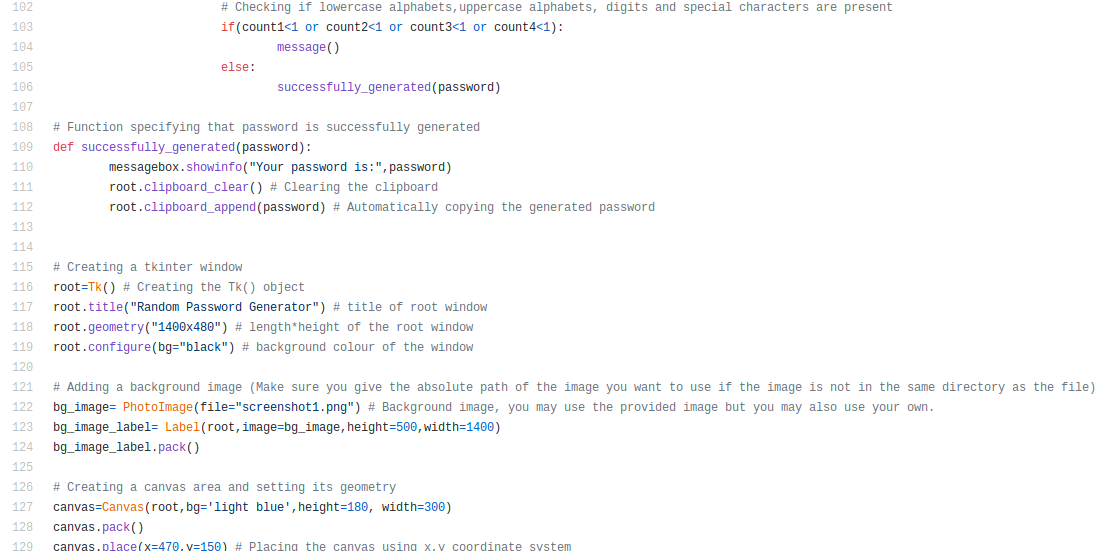
The message function is defined to show a message box when there is an error in generating password. The generate function generates password as per user requirements (default is a strong password). The min password length should be 8 characters (otherwise exception is thrown). Also, there is exception thrown if anything other than a whole number is fed as password length.

****

If everything is correct, then the password is generated. After that, the password is validated to make sure if it is really of the desired security. If the password does not pass the validity test, an error is shown (the message () function defined earlier). E.g. suppose we wanted a medium security password, but the random module picked up only the alphabets from the reference string. In this way, it will become a weak password and not really a medium security password. Validation ensures that this does not happen.

****

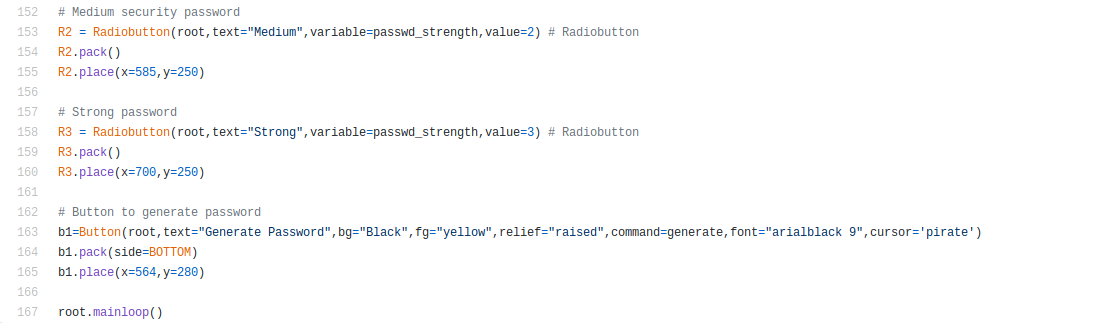
This is the continuation of validation. (In this case, the validity of a strong password is taken care of)

****

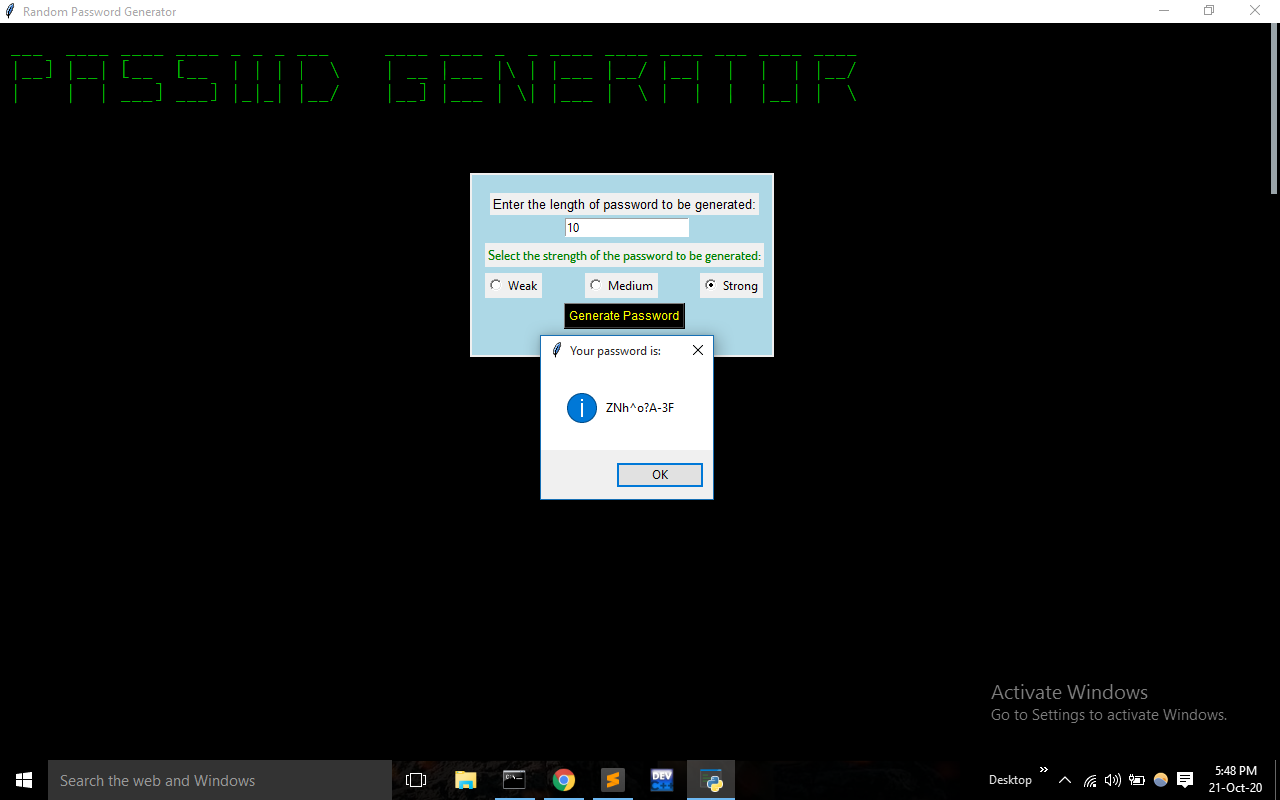
Function specifying that password is successfully generated is shown here. Also, the GUI root window, its title, geometry, background color, the background image and its characteristics and the canvas are shown in this screenshot.

****

The label and entry for password length are shown here. Also, there is label and radio buttons for the user who selects the desired password strength. The radio buttons are assigned with their respective integer values which helps the program know which option is selected.

****

Here the labels and radio buttons for medium and strong passwords are shown. Also there is a button which generates password as soon as it is pressed (unless an exception occurs as described previously). This is the end of the code.

**Results: **A GUI interface for random password generation running as a python script.

**Some tips for a secure password:**

* Always try to use different passwords for different accounts
* Never use something that is as easy as Name@Date\_of\_birth, Name@Year\_of\_birth, Name@current\_year, etc. For safety, don’t even use your pet’s name, house address, etc.
* Don’t store your passwords and usernames in the browser (although we all do it).
* For extraordinary security (even with alphabets only), use something like, ‘Maryhadalittlelamb’ (Mary had a little lamb), ‘Twinkletwinklelittlestar’ (Twinkle twinkle little star), etc. but not as common as these. To put it simply, use phrases/sentences.
* If you want to check your password strength, use secure websites like: <https://www.security.org/how-secure-is-my-password/>