

Python Project Report

CSE106L

Unique Password Generator

With GUI Using Python

This Unique Password Generator can generate endless unique highly secure passwords. It has many features like customizing the password, encrypting and decrypting any text, saving the password and many more. This application is written in python language is made using the latest python GUI package called PySimpleGUIQt. One can organize his/her passwords very comfortably using this application.

About PySimpleGUI:

PySimpleGUI was launched in 2018, so it's a relatively new package compared with the likes of wxPython or PyQt.

PySimpleGUI has four ports:

1. Tkinter
2. PyQt
3. wxPython
4. Remi

PySimpleGUI wraps portions of each of these other packages and makes them easier to use. However, each of the ports has to be installed separately.

PySimpleGUI wraps the entirety of Tkinter, which comes with Python. PySimpleGUI has wrapped most of PySide2, but only a small portion of wxPython. When you install PySimpleGUI, you get the Tkinter variant by default. For more information about Tkinter, check out Python GUI Programming With Tkinter.

Depending on which variant of PySimpleGUI you use, applications that you create with PySimpleGUI may not look native to their platform. But don't let this stop you from giving PySimpleGUI a try. PySimpleGUI is still quite powerful and can get most things done with a little work.

Imported Modules:

PySimpleGUIQt:

PySimpleGUIQt is the PyQt version of PySimpleGUI Module.

random:

Python Random module is an in-built module of Python which is used to generate random numbers. These are pseudo-random numbers means these are not truly random. This module can be used to perform random actions such as generating random numbers, printing random a value for a list or string, etc.

pyperclip:

Pyperclip is a cross-platform Python module for copy and pasting clipboard functions. It works with both Python 2 and 3. This module was created to enable cross-platform copy-pasting in Python which was earlier absent. The pyperclip module has copy() and paste() functions that can send text to and receive a text from your computer's clipboard. Sending the output of your program to the clipboard will make it easy to paste it on an email, word processor, or some other software.

cryptography.fernet:

Cryptography is the practice of securing useful information while transmitting from one computer to another or storing data on a computer. Cryptography deals with the encryption of plaintext into ciphertext and decryption of ciphertext into plaintext. Python supports a cryptography package that helps us encrypt and decrypt data. The fernet module of the cryptography package has inbuilt functions for the generation of the key, encryption of plaintext into ciphertext, and decryption of ciphertext into plaintext using the encrypt and decrypt methods respectively. The fernet module guarantees that data encrypted using it cannot be further manipulated or read without the key.

os:

The OS module in Python provides functions for interacting with the operating system. OS comes under Python's standard utility modules. This module provides a portable way of using operating system dependent functionality. The *os* and *os.path* modules include many functions to interact with the file system.

Application Theme:

PySimpleGUI Module provides many colour themes for the application. We can set a theme using PySimpleGUI.theme() method that takes the theme name as a parameter.

Constants:

BTN_SIZE - Size of the Button widget.

MLTLN_SIZE - Size of the Multiline widget.

NLN_SIZE - Height of the new line character.

POPUP_LCTN - Position of popups from the application on the screen.

Global Variables:

lower - String with all lower case alphabets.
lower_list - List of all lower case alphabets.
upper - String with all upper case alphabets.
upper_list - List of all upper case alphabets.
alpha_list - String with all lower and upper case alphabets.
digits - String with all digits.
digits_list - List of all digits.
alnum_list - List of all alphanumeric characters.
special - String with all special characters.
special_list - List of all special characters.
all_list - List of all alphanumeric and special characters.

Classes:

Password:

Docstring - Password class to represent a password. It generates and stores all the characteristics of a password.

Attributes:

self.password - Stores the generated or user given password.
self.length - Length of the generated or user given password.
self.contains - List of 4 boolean values.
 1st value is true if the password contains Uppercase letters else false.
 2nd value is true if the password contains lowercase letters else false.
 3rd value is true if the password contains digits else false.
 4th value is true if the password contains special characters else false.
self.starts_with - List of 5 boolean values.
 1st value is true if the password starts with the Uppercase letter else false.
 2nd value is true if the password starts with a lowercase letter else false.
 3rd value is true if the password starts with a digit else is false.
 4th value is true if the password starts with a special character else false.
 5th value is true if the password starts with a custom user given string.
self.ends_with - List of 5 boolean values.
 1st value is true if the password ends with an Uppercase letter else false.
 2nd value is true if the password ends with a lowercase letter else false.
 3rd value is true if the password ends with a digit else is false.
 4th value is true if the password ends with a special character else false.
 5th value is true if the password ends with a custom user given string.
self.custom_starts_with - If the 5th value of self.starts_with is true, this attribute holds the custom string.

`self.custom_ends_with` - If the 5th value of `self.ends_with` is true, this attribute holds the custom string.

Constructor:

-> Parametrized Constructor with all default values.

-> Initializes all attributes.

-> If the password is NoneType, `self.generate` method is called that generates a new password.

Methods:

`--str--` - Returns the password.

`--len--` - Returns the length of the password.

`generate` - Generates a new password according to the customizations given by the user.

Layouts in PySimpleGUI:

Layouts in PySimpleGUI are represented using nested lists. Each of the nested lists contains widgets like Buttons, Texts, etc. Widgets in the same list are aligned side by side. Widgets in the different lists are aligned in different lines.

Frames in PySimpleGUI:

The frame is a widget that is a container of widgets. It stores the widgets as a layout. Frames have separate titles.

Widgets in PySimpleGUI:

A widget is an element of a graphical user interface (GUI) that displays information or provides a specific way for a user to interact with the operating system or an application.

Widgets include icons, pull-down menus, buttons, selection boxes, progress indicators, on-off checkmarks, scroll bars, windows, window edges (that let you resize the window), toggle buttons, form, and many other devices for displaying information and for inviting, accepting, and responding to user actions.

Window in PySimpleGUI:

PySimpleGUI provides a Window Element that you use to display other Elements in, such as buttons, text, images, and more. These windows take layout as the parameter and display the widgets in the layout. These Windows can be made Modal. A Modal Window won't let you interact with any other Windows in your program until you exit it. This is useful when you want to force the user to read something or ask the user a question.

Event Loop:

A graphical user interface needs to run inside a loop and wait for the user to do something. For example, the user might need to press a button in your UI or type something with their keyboard. When they do that, those events are processed by the event loop.

When you use PySimpleGUI, you make an event loop by creating an infinite while loop that reads events from the window object. If the user presses the OK button or the Exit button, then you want the program to end. To accomplish that, you break out of the loop and close() the window.

Window.read():

This method returns two values: a string and a dictionary. The string is the event and the dictionary is the values.

Event:

Variable Type - String

Every widget is identified using a key. Whenever the user interacts with a widget, the read method returns the key of the widget as a string indicating the user interaction.

Values:

Variable Type - Dictionary

The keys of this dictionary are the key of every widget of the application. The value of the corresponding key is the data stored in the widget with that key. If the widget does not contain any data, the value is set to None.

Exceptions Handled:

1. Invalid open file path
2. Invalid file (file without .password extension)
3. Invalid save location
4. Invalid decrypt key
5. Invalid encrypt text

Designed and Developed by Rajesh Yemineni, Student, SRM University, AP.

Contact: rajesh_y@srmap.edu.in

Thank you for using this software.

Rajesh Yemineni

AP20110010005

CSE-A

Sample Screenshots:

The screenshot shows the 'Unique Password Generator' application window. It is divided into three main sections: 'Open/Save', 'Generate', and 'Encrypt/Decrypt'.

- Open/Save:** Includes fields for 'Open' (with a 'Browse' button), 'Password' (with 'Open' and 'Copy' buttons), 'Save' (with a dropdown menu and 'Encrypt' checkbox), 'Save location' (with a 'Browse' button), and 'File name' (with a 'Save' button).
- Generate:** Includes a 'Length' field (set to 8), 'Contains' checkboxes (all checked: 'Uppercase Characters (A-Z)', 'Lowercase Characters (a-z)', 'Digits (0-9)', 'Special Characters (!#\$%&'()*+,-./:;<=>?@[\]^_`{|}~)'), 'Starts with' (radio buttons for 'Uppercase Character (A-Z)', 'Lowercase Character (a-z)', 'Digit (0-9)', 'Special Character (!#\$%&'()*+,-./:;<=>?@[\]^_`{|}~)', and 'Custom'), 'Ends with' (radio buttons for 'Uppercase Character (A-Z)', 'Lowercase Character (a-z)', 'Digit (0-9)', 'Special Character (!#\$%&'()*+,-./:;<=>?@[\]^_`{|}~)', and 'Custom'), and a 'Generated Password' field (empty). 'Generate' and 'Copy' buttons are at the bottom.
- Encrypt/Decrypt:** Includes an 'Encrypt' section with 'Input text to be encrypted' (empty), an 'Encrypt' button, a 'Key' field (empty), a 'Copy' button, and an 'Encrypted Text' field (empty) with a 'Copy' button. It also has a 'Decrypt' section with 'Input encrypted text' (empty), a 'Decrypt' button, an 'Input key' field (empty), and a 'Decrypted Text' field (empty) with a 'Copy' button.

At the bottom of the 'Generate' section, it says 'Designed and Developed by AP20110010005, SRM University, AP.'

This screenshot shows the same application window, but with a 'Quick Generate' section and a 'Generated Password' field. The 'Quick Generate' section has a 'Length' field (set to 8) and a 'Quick Generated Password' field (containing '| = , ; . 2 f I'). The 'Generated Password' field (containing 'Sam1667p1e') has 'Generate' and 'Copy' buttons. A small dialog box is open over the 'Generated Password' field, displaying the message: 'Password saved successfully at /home/rajesh/Desktop/Generated_Password_VIAUg.password' with an 'OK' button. The 'Encrypt/Decrypt' section is also visible, showing 'Sample text.' in the 'Input text to be encrypted' field, and the 'Encrypted Text' field containing 'WN-8J0xhM0iIXSEGGLB4NwZBayQVrkjI0KIA=='. The 'Decrypted Text' field also contains 'Sample text.'

End Of File