**PYTHON GUI BASED PROJECT**

**Project to create random password generator in python using Tkinter.**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **TITLE** | **PAGE NO.** |
| **1.**  **2.**  **3.**  **4.**  **5.** | **INTRODUCTION**  **DESIGN/FRAMEWORK**  **ALGORITHM**  **CODE**  **OUTPUT** | **03**  **04-07**  **08-09**  **10-13**  **14** |

**INTRODUCTION**

Creating a strong password and remembering it is a tedious task. So, we built a program that intakes some words from the user and then generates a random password using those words and copy to the clipboard. The user can remember the password with the help of the words the user gave as an input in that particular order.

**Project features:**

1. For creating a stronger password, password is composed of randomly selected uppercase and lowercase letters of input words and last two letters are randomly selected digit and randomly selected special character and length is above 8 ( Password = randomly arranged letter of words entered + randomly selected digit + randomly selected special character ).

2. User can select different length of password (any number). User can copy password to the clipboard with COPY button.

2. DATABASE (spreadsheet) to remember all passwords along with the words of user.

3. An ERROR POP-UP MESSAGE BOX will appear on directly clicking generate button or remember button without entering any words.

4. Length of password matters: Same words in same order but different length of password will generate different password (because length is different).

5. It remembers all passwords generated using those words and will be shown in the order of their generation.

6. IMPORTANT NOTES and STEPS are shown in the window for user’s convenience and understanding.

**DESIGN/FRAMEWORK**

1. **Tkinter** - **Tkinter** is the most commonly used library for developing GUI (Graphical User Interface) in Python. It is a standard Python interface to the Tk GUI toolkit shipped with Python. As Tk and Tkinter are available on most of the Unix platforms as well as on the Windows system, developing GUI applications with Tkinter becomes the fastest and easiest. **Tkinter** provides us with a variety of common GUI elements which we can use to build out interface – such as buttons, menus and various kind of entry fields and display areas. We call these elements **Widgets**.

* In general, **Widget** is an element of Graphical User Interface (GUI) that displays/illustrates information or gives a way for the user to interact with the OS. In **Tkinter**, **Widgets** are objects; instances of classes that represent buttons, frames, and so on.

Each separate widget is a Python object. When creating a widget, you must pass its parent as a parameter to the widget creation function. The only exception is the “root” window, which is the top-level window that will contain everything else and it does not have a parent.

|  |  |
| --- | --- |
| **WIDGETS** | **DESCRIPTION** |
| **Label** | **It is used to display text or image on the screen** |
| **Button** | **It is used to add buttons to your application** |
| **Canvas** | **It is used to draw pictures and others layouts like texts, graphics etc.** |
| **ComboBox** | **It contains a down arrow to select from list of available options** |
| **CheckButton** | **It displays a number of options to the user as toggle buttons from which user can select any number of options.** |
| **RadiButton** | **It is used to implement one-of-many selection as it allows only one option to be selected** |
| **Entry** | **It is used to input single line text entry from user** |
| **Frame** | **It is used as container to hold and organize the widgets** |
| **Message** | **It works same as that of label and refers to multi-line and non-editable text** |
| **Scale** | **It is used to provide a graphical slider which allows to select any value from that scale** |
| **Scrollbar** | **It is used to scroll down the contents. It provides a slide controller.** |
| **SpinBox** | **It is allows user to select from given set of values** |
| **Text** | **It allows user to edit multiline text and format the way it has to be displayed** |
| **Menu** | **It is used to create all kinds of menu used by an application** |

**Geometry management:**

| **METHOD** | **DESCRIPTION** |
| --- | --- |
| **pack()** | **The Pack geometry manager packs widgets in rows or columns.** |
| **grid()** | **The Grid geometry manager puts the widgets in a 2-dimensional table.  The master widget is split into a number of rows and columns, and each “cell” in the resulting table can hold a widget.** |
| **place()** | **The Place geometry manager is the simplest of the three general geometry managers provided in Tkinter.  It allows you explicitly set the position and size of a window, either in absolute terms, or relative to another window.** |

2. **tkinter.ttk** - It is used to create the GUI applications with the effects of modern graphics which cannot be achieved using only *tkinter*. Checkbutton is used to select multiple options.

Checkbuttons can be created using following snippet.

chkbtn = ttk.Checkbutton(master, option=value, ...)

3. **Pyperclip** - **Pyperclip** is a cross-platform Python module for copy and paste 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 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.

**Installing pyperclip:**

pip install pyperclip

4. **random module** : The Python random module functions depend on a pseudo-random number generator function random(), which generates the float number between 0.0 and 1.0.

5. **CSV module**: CSV stands for comma separated values. This file format is a commonly used data format while exporting/importing data to/from spreadsheets and data tables in databases. The CSV module was incorporated in Python’s standard library as a result of PEP 305. It presents classes and methods to perform read/write operations on CSV file as per recommendations of PEP 305.

CSV is a preferred export data format by Microsoft’s Excel spreadsheet software. However, CSV module can handle data represented by other dialects also.

6. **OS module**: 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.

**ALGORITHM**

**Steps:**

1.Start

2.First of all, we import all the package required.

3.We create a new/ clear the old CSV file.

4.Create GUI window with (550x230) geometry, lawn green color, giving it a title.

5. Include three buttons and three label fields in the GUI.

6.For GENERATE button:

1. Control goes to generate function and from there it goes to low function.
2. If there are no words in input then go to step 3 else go to step 4.
3. An error message box will appear and return to generate function.
4. Passwords will be generated in passwords variable.
5. Open the CSV file and append a row containing words and password then return back to generate function.

7.For REMEMBER button:

1. Open the file to read and read all first column of every row and check if the input words are there or not.
2. If we found those words in that order then go to step 3 else go to step 5.
3. Declare counter to 0.
4. Loop through each row and where we find those words, we will increase the counter by one and display its password along with the counter number.
5. An error message box will appear and return to generate function.

8.For COPY button:

1. Get the password grid text in variable random\_password.
2. Use copy( function of pyperclip module to copy random\_password to the clipboard.

9. End

**CODE**

# Random Password Generator Program using Tkinter in python.

# (Creating a strong password and remembering it is a tedious task.

# You can build a program that intakes some words from the user and then generates a random password using those words.

# The user can remember the password with the help of the words he gave as an input).

# frameworks used

import pyperclip

import random

import csv

import os

from tkinter import \*

from tkinter.ttk import \*

from tkinter import messagebox

# Function for calculation of password

def low():

    entry.delete(0, END)

    length = var1.get() # Get the length of passowrd

    words = var2.get() # Get words from user

    words\_without\_space = words.replace(" ","") # remove all spaces from words input so we don't have space in our strong password.

    #string of all letters of words in upper and lower case

    words\_without\_space = words\_without\_space.upper() + words\_without\_space.lower()

    digits ="0123456789"

    special = "!@#$%^&\*(){}[]<>,."

    password = ""

    if words\_without\_space == "":

        # If user does not entered any words, an error message will be shown.

        messagebox.showerror("Warning","Error: Please enter some words.")

        return ""

    else:

        # for strong password

        # password = randomly arranged letters of word(of upper and lower case) + randomly selected digit + randomly selected special character

        for i in range(0, length-2):

            password = password + random.choice(words\_without\_space)

        password = password + random.choice(digits)

        password = password + random.choice(special)

        with open('passwords.csv', 'a', newline = '') as file:

            writer = csv.writer(file)

            words = words + str(length)

            writer.writerow([words, password])

        file.close()

        return password

# Function for GENERATE button to generate password

def generate():

    password1 = low()

    entry.insert(10, password1)

# Function for COPY button to copy password to clipboard

def copy():

    random\_password = entry.get()

    pyperclip.copy(random\_password)

# Function for REMEMBER button to remember password

def remember():

    entry.delete(0, END)

    data = []

    with open("passwords.csv") as csvfile:

        reader = csv.reader(csvfile)

        for row in reader:

            data.append(row)

    length = var1.get()

    words = var2.get()

    words = words + str(length)

    column = [i[0] for i in data]

    counter = 0

    if words in column:

        for i in range(0,len(data)):

            if words == data[i][0]:

                # for printing all passwords generated by those arrangement of words entered by user.

                counter += 1

                p = str(counter) + ". " + data[i][1] + " " # printing all passwords in order of their generation.

                entry.insert(20, p)

    else:

        # If enter words which was not entered earlier, an error message will be shown.

        messagebox.showerror("Warning","These words not entered in this order before.\nPlease enter words in the order entered before.")

# Main Function

# clear file data

f = open("passwords.csv", "w") # a csv file is created.

f.truncate()

f.close()

# create GUI window

window = Tk()

window.geometry('550x230')

window.configure(background="lawngreen")

var = IntVar()

var1 = IntVar()

var2 = StringVar()

# Title of your GUI window

window.title("Random Password Generator")

# Create label and entry for WORDS from user

radio\_strong = Label(window, text="WORDS",background="lawngreen",font="Bold 10")

radio\_strong.grid(row=0)

enter = Entry(window, textvariable=var2)

enter.grid(row=0, column=1,ipadx = 8)

# create label for LENGTH of password

c\_label = Label(window, text="LENGTH",background="lawngreen",font="Bold 10")

c\_label.grid(row=2)

# create label and entry to show user the PASSWORD generated

Random\_password = Label(window, text="PASSWORD",background="lawngreen",font="Bold 10 ")

Random\_password.grid(row=2, column = 2)

entry = Entry(window)

entry.grid(row=2, column=3,ipadx = 27)

# create Buttons

# GENERATE button which will generate the password

# COPY button which will copy password to clipboard

# REMEMBER button to remember the password

generate\_button = Button(window, text="GENERATE", command=generate)

generate\_button.grid(row=0, column=2)

copy\_button = Button(window, text="COPY", command=copy)

copy\_button.grid(row=0, column=3)

remember\_button = Button(window, text="REMEMBER", command=remember)

remember\_button.grid(row=1, column=3)

# display all the messages for the user to read and use this GUI properly

radio\_strong = Label(window, text="",background="lawngreen",font="Bold 10")

radio\_strong.grid(row=3)

radio\_strong = Label(window, text="NOTE:",background="lawngreen",font="Bold 10")

radio\_strong.grid(row=4)

radio\_strong = Label(window, text="ORDER of words and",background="lawngreen",font="Bold 10")

radio\_strong.grid(row=4,column = 1)

radio\_strong = Label(window, text="LENGTH of password",background="lawngreen",font="Bold 10")

radio\_strong.grid(row=4,column = 2)

radio\_strong = Label(window, text="are very crucial for",background="lawngreen",font="Bold 10")

radio\_strong.grid(row=5,column = 1)

radio\_strong = Label(window, text="remembering password.",background="lawngreen",font="Bold 10")

radio\_strong.grid(row=5,column = 2)

radio\_strong = Label(window, text="STEPS:",background="lawngreen",font="Bold 10")

radio\_strong.grid(row=6)

radio\_strong = Label(window, text="To Generate:",background="lawngreen",font="Bold 10")

radio\_strong.grid(row=6,column = 1)

radio\_strong = Label(window, text="1. Enter the words and\nchoose length.",background="lawngreen",font="Bold 10")

radio\_strong.grid(row=6,column = 2)

radio\_strong = Label(window, text="2. Click on GENERATE button.",background="lawngreen",font="Bold 10")

radio\_strong.grid(row=6,column = 3)

radio\_strong = Label(window, text="To Remember:",background="lawngreen",font="Bold 10")

radio\_strong.grid(row=7,column = 1)

radio\_strong = Label(window, text="1. Enter the words and\nchoose length.",background="lawngreen",font="Bold 10")

radio\_strong.grid(row=7,column = 2)

radio\_strong = Label(window, text="2. Click on REMEMBER button.",background="lawngreen",font="Bold 10")

radio\_strong.grid(row=7,column = 3)

radio\_strong = Label(window, text="To Copy:",background="lawngreen",font="Bold 10")

radio\_strong.grid(row=8,column = 1)

radio\_strong = Label(window, text="1. Click on COPY button.",background="lawngreen",font="Bold 10")

radio\_strong.grid(row=8,column = 2)

radio\_strong = Label(window, text="2. Paste where needed.",background="lawngreen",font="Bold 10")

radio\_strong.grid(row=8,column = 3)

combo = Combobox(window, textvariable=var1)

# Combo Box for length of password to be choosed by user

combo['values'] = (8, 9, 10, 11, 12, 13, 14, 15, 16, "Enter any Length") # password can be of any length

combo.current(0)

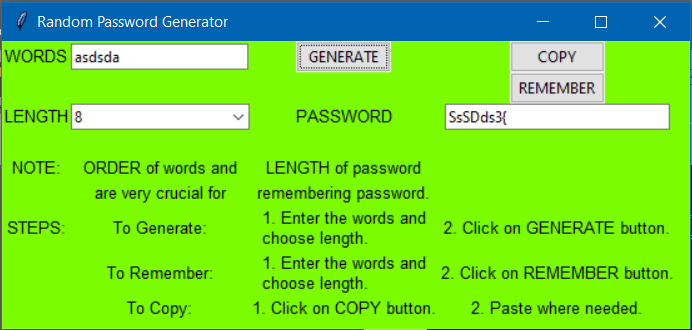
combo.bind('<<ComboboxSelected>>')

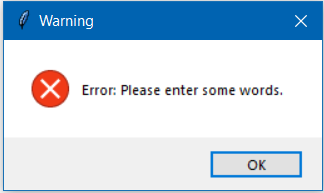
combo.grid(column=1, row=2)

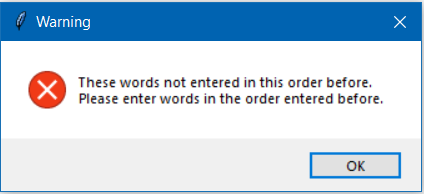
# start the GUI

window.mainloop()

**OUTPUT**







Thank you