**[Generate Password with Python]**

**Mini Project “Generate Password” Report**

Submitted in partial fulfillment of the requirements for the degree of

**Bachelor of Engineering (EXTC Engineering)**

**Second year EXTC Engineering**

by:

|  |  |
| --- | --- |
| ANIKET MAHADEV GAWADE | TU2F1920082 |

**Under the Guidance of**

**Prof. [Nilesh sir]**



**Department of EXTC Engineering**

**TERNA ENGINEERING COLLEGE**

Nerul (W), Navi Mumbai 400706

**(University of Mumbai)**

(2020-21)

****

**TERNA ENGINEERING COLLEGE, NERUL**

**Department of EXTC Engineering**

Academic Year 2020-21

**CERTIFICATE**

This is to certify that the mini project of the subject”**Python Programing Lab**” entitled **“****Generate Password with Python”** is a bonafide work of

|  |  |
| --- | --- |
| ANIKET MAHADEV GAWADE | TU2F1920082 |

submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the Bachelor of Engineering (EXTC Engineering).

**Guide Head of Department Principal**

**Project Report Approval**

This Mini Project Report of the subject “**Python Programming Lab** ” Report – entitled “**Generate Password with Python**” by following students is approved for the degree of ***B.E. in "EXTC Engineering"***.

**Submitted by:**

|  |  |
| --- | --- |
| ANIKET MAHADEV GAWADE | TU2F1920082 |

Examiners Name & Signature:

1.---------------------------------------------------------

2.----------------------------------------------------------

Date: ---------------------------------

Place: ---------------------------------

**Declaration**

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

|  |  |  |
| --- | --- | --- |
| ANIKET MAHADEV GAWADE | TU2F1920082 | --------------------------- |

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Place: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Acknowledgement**

We would like to express our sincere gratitude towards our guide **Prof. Nilesh Kulal,** for their help, guidance and encouragement, they provided during the project development. This work would have not been possible without their valuable time, patience and motivation. We thank them for making our stint thoroughly pleasant and enriching. It was great learning and an honor being their student.

We are deeply thankful to **Dr. Jyothi Digge (H.O.D EXTC Department)** and entire team in the EXTC Department. They supported us with scientific guidance, advice and encouragement, they were always helpful and enthusiastic and this inspired us in our work.

We take the privilege to express our sincere thanks **to Dr. L. K. Ragha** our Principal for providing the encouragement and much support throughout our work.

|  |  |  |
| --- | --- | --- |
| ANIKET MAHADEV GAWADE | TU2F1920082 | --------------------------- |

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Place: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Abstract**

It is a tool that generates passwords based on the given guidelines that you set to create an unpredictable strong password for your accounts.

The Password generator tool creates a random and customized password for users that helps them to create a strong password which provides greater security.

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Table of Contents** | | |
| caption |  | Page no. |
| Chapter 1 | Introduction | 07 |
|  | * 1. Aim And Objective of project   2. Scope | 07  08 |
| Chapter 2 | Literature Survey | 08 |
| Chapter 3 | Software Analysis | 09 |
| Chapter 4 | Design And implementation  1.1 code | 09  10 |
| Chapter 5 | Implementation Details | 12 |
| Chapter 6 | Results/Project Screenshots | 15 |
| Chapter 7 | Conclution | 17 |

**CHAPTER - 1**

**Introduction**

* Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.[29]
* Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library

**1.1 Aim and Objectives of Project**

The objective of this project is to create a password generator using python. The password generator project will be build using python modules like Tkinter, random, string, pyperclip.

In this project, the user has to select the password length and then click on the “**Generate Password**” button. It will show the generated password below. If the user clicks on the “**Copy To Clipboard**” button, then it will copy the password automatically.

**1.2 Scope**

The scope of the project is to A random password generator is a software program, hardware device, or online tool that automatically generates a password using parameters that a user sets, including mixed-case letters, numbers, symbols, pronounceability, length, and strength.

**Chapter 2**

**Literature Survey**

* 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.

**Chapter 3**

**Software Analysis**

To build this project we will use the basic concept of python and libraries – Tkinter, pyperclip, random, string.

* **Tkinter** is a standard GUI library and is one of the easiest ways to build a GUI application.
* **pyperclip** module allows us to copy and paste text to and from the clipboard to your computer
* **The random** module can generate random numbers
* **string** module contains a number of functions to process the standard python string.

**Chapter 4**

**Design And Implementation**

Let’s check the step to build a Password Generator using Python

* Import modules
* Initialized Window
* Select Password Length
* Define Functions

**1.1 Generate Password with Python code:-**

#importing Libraries

from tkinter import \*

import random, string

import pyperclip

###initialize window

root =Tk()

root.geometry("400x400")

root.resizable(0,0)

root.title(" PASSWORD GENERATOR")

#heading

heading = Label(root, text = 'PASSWORD GENERATOR' , font ='arial 15 bold').pack()

Label(root, text ='box', font ='arial 15 bold').pack(side = BOTTOM)

###select password length

pass\_label = Label(root, text = 'PASSWORD LENGTH', font = 'arial 10 bold').pack()

pass\_len = IntVar()

length = Spinbox(root, from\_ = 8, to\_ = 32 , textvariable = pass\_len , width = 15).pack()

#####define function

pass\_str = StringVar()

def Generator():

password = ''

for x in range (0,4):

password = random.choice(string.ascii\_uppercase)+random.choice(string.ascii\_lowercase)+random.choice(string.digits)+random.choice(string.punctuation)

for y in range(pass\_len.get()- 4):

password = password+random.choice(string.ascii\_uppercase + string.ascii\_lowercase + string.digits + string.punctuation)

pass\_str.set(password)

###button

Button(root, text = "GENERATE PASSWORD" , command = Generator ).pack(pady= 5)

Entry(root , textvariable = pass\_str).pack()

########function to copy

def Copy\_password():

pyperclip.copy(pass\_str.get())

Button(root, text = 'COPY TO CLIPBOARD', command = Copy\_password).pack(pady=5)

# loop to run program

root.mainloop()

**Chapter 5**

**Implementation Details**

1. Import Libraries

The first step is to import libraries

from tkinter import \*

import random, string

import pyperclip

2. Initialize Window

root = Tk()

root.geometry("400x400")

root.resizable(0,0)

root.title("DataFlair - PASSWORD GENERATOR")

* **Tk()** initialized tkinter which means window created
* **geometry()** set the width and height of the window
* **resizable(0,0)** set the fixed size of the window
* **title()** set the title of the window

Label(root, text = 'PASSWORD GENERATOR' , font ='arial 15 bold').pack()

Label(root, text ='DataFlair', font ='arial 15 bold').pack(side = BOTTOM)

**Label()** widget use to display one or more than one line of text that users can’t able to modify.

* **root** is the name which we refer to our window
* **text** which we display on the label
* **font** in which the text is written
* **pack** organized widget in block

3. Select Password Length

pass\_label = Label(root, text = 'PASSWORD LENGTH', font = 'arial 10 bold').pack()

pass\_len = IntVar()

length = Spinbox(root, from\_ = 8, to\_ = 32 , textvariable = pass\_len , width = 15).pack()

* **pass\_len** is an integer type variable that stores the length of a password.
* To select the password length we use **Spinbox()** widget.
* **Spinbox()** widget is used to select from a fixed number of values. Here the value from 8 to 32

4. Function to Generate Password

pass\_str = StringVar()

**def** Generator():

password = ''

**for** x **in** range (0,4):

Password = random.choice(string.ascii\_uppercase) +

random.choice(string.ascii\_lowercase) + random.choice(string.digits) +

random.choice(string.punctuation)

**for** y **in** range(pass\_len.get()- 4):

password = password + random.choice(string.ascii\_uppercase +

string.ascii\_lowercase + string.digits + string.punctuation)

pass\_str.set(password)

**pass\_str** is a string type variable that stores the generated password

* **password** = “” is the empty string
* First loop will generate a string of length 4 which is a combination of an uppercase letter, a lowercase letter, digits, and a special symbol and that string will store in password variable.
* The second loop will generate a random string of length entered by the user – 4 and add to the password variable. Here we minus 4 to the length of the user because we already generate the string of length 4.

We have done this because we want a password which must contain an uppercase, a lowercase, a digit, and a special symbol.

Now the password is set to the **pass\_str()** variable.

Button(root, text = "GENERATE PASSWORD" , command = Generator ).pack(pady= 5)

Entry(root , textvariable = pass\_str).pack()

* **Button()** widget used to display button on our window
* **command** is called when the button is click
* **Entry()** widget used to create an input text field
* **textvariable** used to retrieve the current text to the entry widget

5. Function to Copy Password

def Copy\_password():

pyperclip.copy(pass\_str.get())

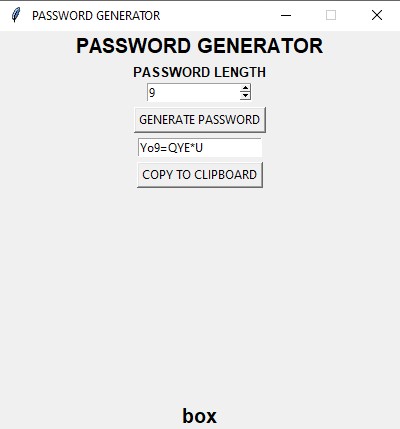
Button(root, text = 'COPY TO CLIPBOARD', command =

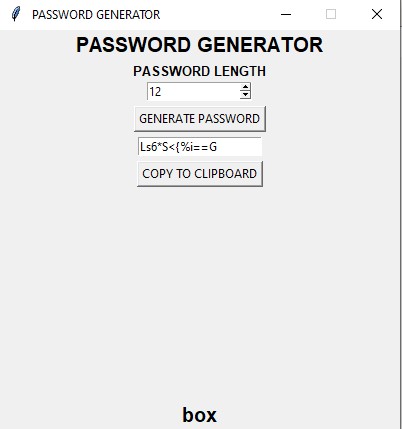
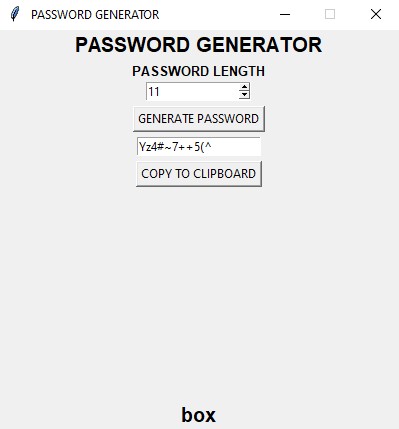
Copy\_password).pack(pady=5)

**pyperclip.copy()** used to copy the text to clipboard

**Chapter 6**

**Results/ Project Screenshots**

****

****

**Chapter 7**

**Conclusion.**

With these steps, we have successfully created a random 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 textfield, labels, and spinbox. In this way, we successfully created our password generator python project. Hope you enjoyed it.

**References**

Google:-

https://data-flair.training/