Index

Annexure I

Micro Project Proposal

Random Password Generator

1.Aims/Benefits of the Micro-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.

A strong password generated online can help you protect the security of your personal and professional email accounts, social network accounts, WiFi encryption, banking and financial and savings accounts.

2. Course Outcome Addressed:

a. Display message on screen using python script on IDE.

b. Develop python program to demonstrate use of operators

c. Perform operations on data structures in python.

d. Develop function for given problem.

e .Handle exceptions.

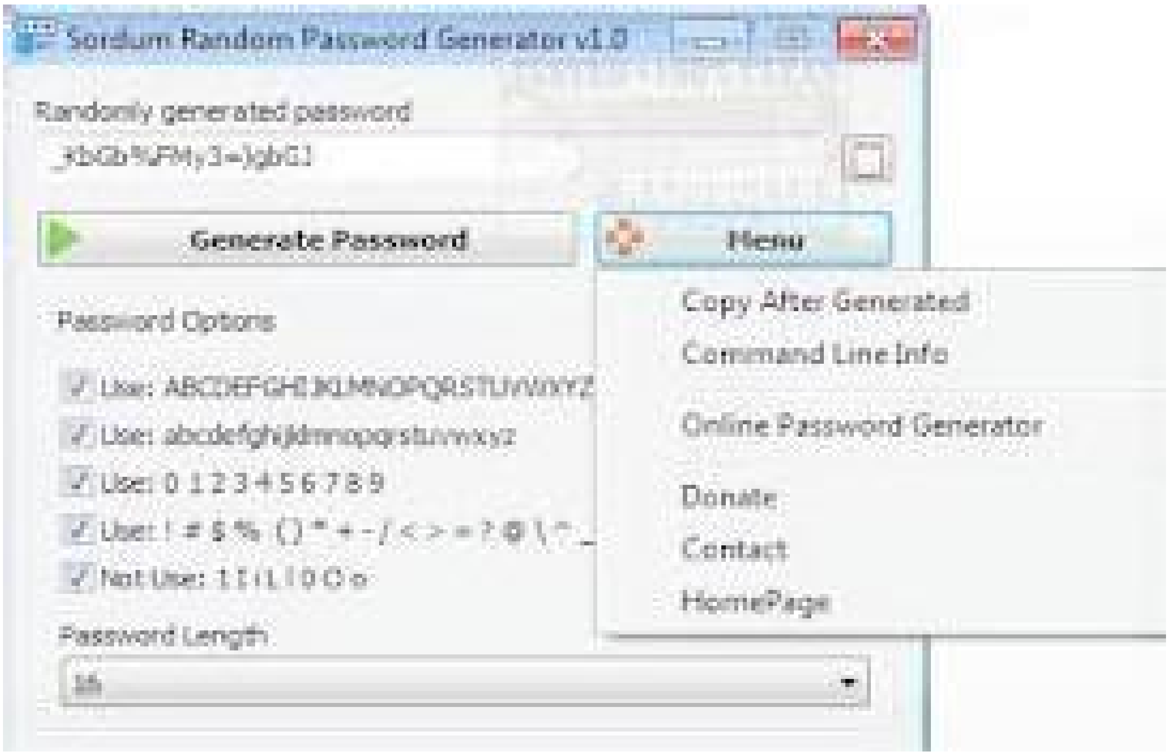
3. Proposed Methodology:

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.

A password generator can be part of a password manager. When a password policy enforces complex rules, it can be easier to use a password generator based on that set of rules than to manually create passwords.

Long strings of random characters are difficult for most people to memorize. Mnemonic hashes, which reversibly convert random strings into more memorable passwords, can substantially improve the ease of memorization. As the hash can be processed by a computer to recover the original 60-bit string, it has at least as much information content as the original string. Similar techniques are used in memory sport.



Randomly generated password

_KbGb%FMy3=}gbGJ

Password Options

☑ Use: ABCDEFGHIJKLMNOPQRSTUVWXYZ

☑ Use: abcdefghijklmnopqrstuvwxyz

☑ Use: 0 1 2 3 4 5 6 7 8 9

☑ Use: ! # $ % ( ) * + - / < > = ? @ \ ^ _

☑ Not Use: 1 I i L l 0 O o

Password Length

16

4.Action Plan:

Sr. No.	Details of Activity	Planned Start date	Planned Finish date	Name of responsible Team Member
1	Search the topic	25/03/2021 2:00pm-4:00pm	01/04/2021 2:00pm-4:00pm	Shraddha Surykant Biradar
2	Search the information	08/04/2021 2:00pm-4:00pm	15/04/2021 2:00pm-4:00pm	
3	Algorithm developing	22/04/2021 2:00pm-4:00pm	29/04/2021 2:00pm-4:00pm	
4	Flowchart developing	06/05/2021 2:00pm-4:00pm	13/05/2021 2:00pm-4:00pm	
5	Function making	20/05/2021 2:00pm-4:00pm	27/05/2021 2:00pm-4:00pm	
6	Coding developing	29/05/2021 2:00pm-4:00pm	01/06/2021 2:00pm-4:00pm	
7	Debugging	03/06/2021 2:00pm-4:00pm	07/06/2021 2:00pm-4:00pm	
8	Finalizing Project with its report	08/06/2020 2:00pm-4:00pm	11/06/2021 2:00pm-4:00pm	

5. Resources Required:

Sr. No.	Name of resource / material	Specification	Quantity	Remarks
1	Computer	WINDOWS 7,2GB RAM, 160GB HDD	1	
2	Operating System	WINDOWS 7	1	
3	Compiler	Turbo C/GCC	1	
4	Browser	Chrome	1	

Names of Team Members with Roll No.’s:

Sr. No.	Enrollment No.	Name of Team Member	Roll No.
1		Shraddha Surykant Biradar	24

Mr. Lokare A.P.

Name and Signature of the Teacher

Annexure – II

Micro Project Proposal

Random Password Generator

1. Rationale:

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. Mnemonic hashes, which reversibly convert random strings into more memorable passwords, can substantially improve the ease of memorization.

2. Aims/Benefits of the Micro-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.

A strong **password** generated online can help you protect the security of your personal and professional email accounts, social network accounts, WiFi encryption, banking and financial and savings accounts

3. Course Outcomes Achieved:

a. Display message on screen using python script on IDE.

b. Develop python program to demonstrate use of operators

c. Perform operations on data structures in python.

d. Develop function for given problem.

e .Handle exceptions.

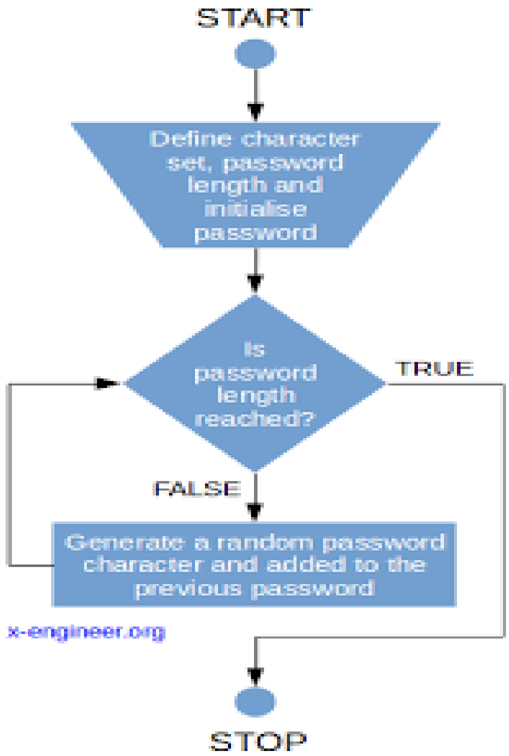
4.Literature Review:

A variety of methods exist for generating strong, cryptographically secure random passwords. On Unix platforms /dev/random and /dev/urandom are commonly used, either programmatically or in conjunction with a program such as makepasswd.[7] Windows programmers can use the Cryptographic Application Programming Interface function CryptGenRandom. The Java programming language includes a class called *SecureRandom*. Another possibility is to derive randomness by measuring some external phenomenon, such as timing user keyboard input.

Many computer systems already have an application (typically named "apg") to implement FIPS 181. FIPS 181—Automated Password Generator—describes a standard process for converting random bits (from a hardware random number generator) into somewhat pronounceable "words" suitable for a passphrase. However, in 1994 an attack on the FIPS 181 algorithm was discovered, such that an attacker can expect, on average, to break into 1% of accounts that have passwords based on the algorithm, after searching just 1.6 million passwords. This is due to the non-uniformity in the distribution of passwords generated, which can be addressed by using longer passwords or by modifying the algorithm

5 Actual MethodologyFollowed:

5.1 Flow Chart:


START
Define character set, password length and initialise password
Is password length reached?
TRUE
FALSE
Generate a random password character and added to the previous password
x-engineer.org
STOP

5.2 Source Code:

```
# Python program to generate random
# password using Tkinter module
import random
import pyperclip
from tkinter import *
from tkinter.ttk import *
```

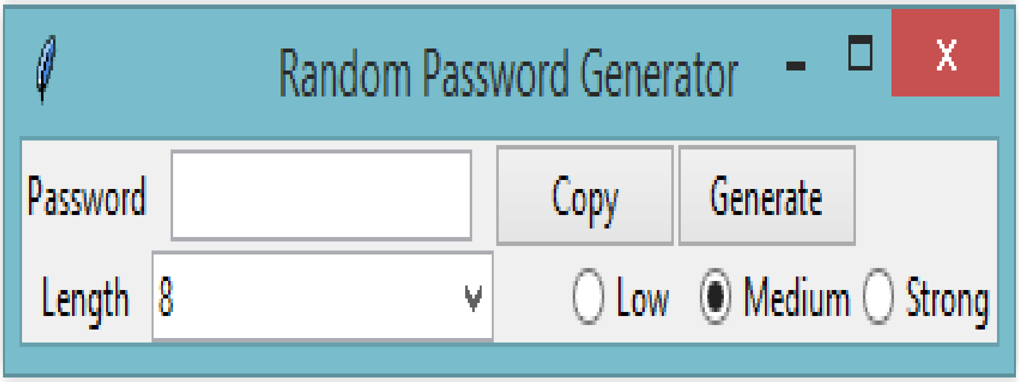
```
# Function for calculation of password
def low():
    entry.delete(0, END)

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

    lower = "abcdefghijklmnopqrstuvwxyz"
    upper = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"
    digits = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789
!@#$%^&*()"
    password = ""

    # if strength selected is low
    if var.get() == 1:
        for i in range(0, length):
            password = password + random.choice(lower)
        return password

    # if strength selected is medium
    elif var.get() == 0:
        for i in range(0, length):
            password = password + random.choice(upper)
        return password

    # if strength selected is strong
    elif var.get() == 3:
        for i in range(0, length):
            password = password + random.choice(digits)
        return password
    else:
        print("Please choose an option")


# Function for generation of password
def generate():
    password1 = low()
    entry.insert(10, password1)


# Function for copying password to clipboard
def copy1():
    random_password = entry.get()
    pyperclip.copy(random_password)


# Main Function

# create GUI window
root = Tk()
var = IntVar()
var1 = IntVar()
```

```
# Title of your GUI window
root.title("Random Password Generator")

# create label and entry to show
# password generated
Random_password = Label(root, text="Password")
Random_password.grid(row=0)
entry = Entry(root)
entry.grid(row=0, column=1)

# create label for length of password
c_label = Label(root, text="Length")
c_label.grid(row=1)

# create Buttons Copy which will copy
# password to clipboard and Generate
# which will generate the password
copy_button = Button(root, text="Copy", command=copy1)
copy_button.grid(row=0, column=2)
generate_button = Button(root, text="Generate", command=generate)
generate_button.grid(row=0, column=3)

# Radio Buttons for deciding the
# strength of password
# Default strength is Medium
radio_low = Radiobutton(root, text="Low", variable=var, value=1)
radio_low.grid(row=1, column=2, sticky='E')
radio_middle = Radiobutton(root, text="Medium", variable=var, value=0)
radio_middle.grid(row=1, column=3, sticky='E')
radio_strong = Radiobutton(root, text="Strong", variable=var, value=3)
radio_strong.grid(row=1, column=4, sticky='E')
combo = Combobox(root, textvariable=var1)

# Combo Box for length of your password
combo['values'] = (8, 9, 10, 11, 12, 13, 14, 15, 16,
                   17, 18, 19, 20, 21, 22, 23, 24, 25,
                   26, 27, 28, 29, 30, 31, 32, "Length")
combo.current(0)
combo.bind('<<ComboboxSelected>>')
combo.grid(column=1, row=1)

# start the GUI
root.mainloop()
```

6. Actual Resources Used:

Sr. No.	Name of resource / material	Specification	Quantity	Remarks
1	Computer	WINDOWS 7,2GB RAM, 160GB HDD	1	
2	Operating System	WINDOWS 7	1	
3	Compiler	Turbo C/GCC	1	
4	Browser	Chrome	1	

7.Outputs of Micro-Projects:



Random Password Generator

Password

Length 8

○ Low ● Medium ○ Strong

8.Skill developed / Learning out of this Micro-Project:

There are so many thing that we learn from this project of

1. We learn that how to make the project in PHP.
2. How to do the testing of program in Visual Studio.
3. How to collect the information and how to make the presentation that we learn from this project.
4. We develop our logic implementation for programing and coding.
5. We learn to Random Password Generator.
6. We learn how to use correct data type in program.
8. This all thing we learn from this project.

9. Applications of this Micro-Project:

1. It can also be used to radio Button, Lable, CheckBox,RadioButton etc.
2. It can also be used to PHP .

****