UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

NAAC GRADE A+ ACCREDITED UNIVERSITY

Student Name: Aakash Thapliyal
UID: 24MCC20027
Branch: MCA(ccd)
Section/Group: A 1
Semester: 1st
Date of Performance: 22-10-2024
Subject Name: Python Programming
Subject Code: 24CAH-606

Q. Title of Project. Password Generator Using Python

Aim/Overview of the practical: In today's digital world, ensuring secure authentication is of utmost importance. One of the primary ways to protect data is by using strong and secure passwords. This project involves creating a simple yet effective password generator using Python that generates random passwords based on a combination of lowercase letters, uppercase letters, and digits. The user is prompted to input the desired password length, and the program generates a random password accordingly.

1. Task to be done: The objective of this project is to develop a Python-based tool to generate random and secure passwords, enhancing the security of user accounts by providing unique and unpredictable passwords.

2. Code for experiment/practical:

```
import string
import random
s1 = string.ascii_lowercase #(a-z)
s2 = string.ascii_uppercase #(A-Z)
s3 = string.digits #(0-9)
plen = int(input("Enter password length\n"))
s = []
s.extend(list(s1))
s.extend(list(s2))
s.extend(list(s3))
print("Your password is: ")
print("".join(random.sample(s, plen)))
```

3. Result/Output/Writing Summary:

```
Enter password length
 8
Your password is:
YJNwOCsp
```

4. Future Scope:

This project can be enhanced by:

1. Adding special characters to increase password complexity.
2. Creating a graphical user interface (GUI) for ease of use.

5. Modules Used:

string: To access pre-defined character sets like lowercase letters, uppercase letters, and digits.
random: To shuffle and generate a random selection of characters for the password.

6. Conclusion:

The password generator is an efficient tool that provides users with a strong and random password, significantly reducing the chances of security breaches. This project demonstrates the importance of generating complex passwords and how Python can be effectively used to solve real-world security challenges.

Learning outcomes (What I have learnt):

1. **Understanding of Python Modules**: Gained proficiency in using Python's built-in string and random modules to access character sets and generate random sequences. This provided insight into how Python can handle string manipulation and randomization.
2. **Knowledge of String Operations**: Learned how to work with string data types and perform operations such as extending lists with multiple character sets (lowercase, uppercase, digits), and joining strings to create a final password.
3. **Randomization Techniques**: Gained insight into randomization techniques by learning how to shuffle and sample elements randomly using the random.sample() function, which is essential in generating unpredictable passwords.