Project Title: Password Generator

Introduction:

The Password Generator project aims to create a tool that enables users to generate secure and customizable passwords. The Python programming language is utilized for this purpose, and the project is implemented in a Jupyter Notebook environment, offering an interactive and user-friendly interface.

Objective:

A password generator is a useful tool that generates strong and random passwords for users. This project aims to create a password generator application using Python, allowing users to specify the length and complexity of the password.

- User Input: Prompt the user to specify the desired length of the password.
- Generate Password: Use a combination of random characters to generate a password of the specified length.
- Display the Password: Print the generated password on the screen.

The primary objective of this project is to provide users with a reliable tool for generating strong and unique passwords tailored to their specific requirements. By allowing users to specify the desired length and composition of the password, the generator ensures that the generated passwords meet individual security needs.

Implementation:

The implementation of the Password Generator project involves the following steps:

1. **Importing Required Libraries:**

The project utilizes the `random` and `string` libraries in Python to generate random characters for the password. These libraries offer functions to generate random letters, digits, and special characters.

2. **User Input:**

The program prompts the user to input the desired lengths of letters, digits, and special characters for the password. This input enables users to customize the password according to their preferences.

3. **Generating Password:**

Based on the user input, the program generates random characters for each category (letters, digits, and special characters) using list comprehensions and the `random.choice()` function. The generated characters are then concatenated and shuffled to create a randomized password.

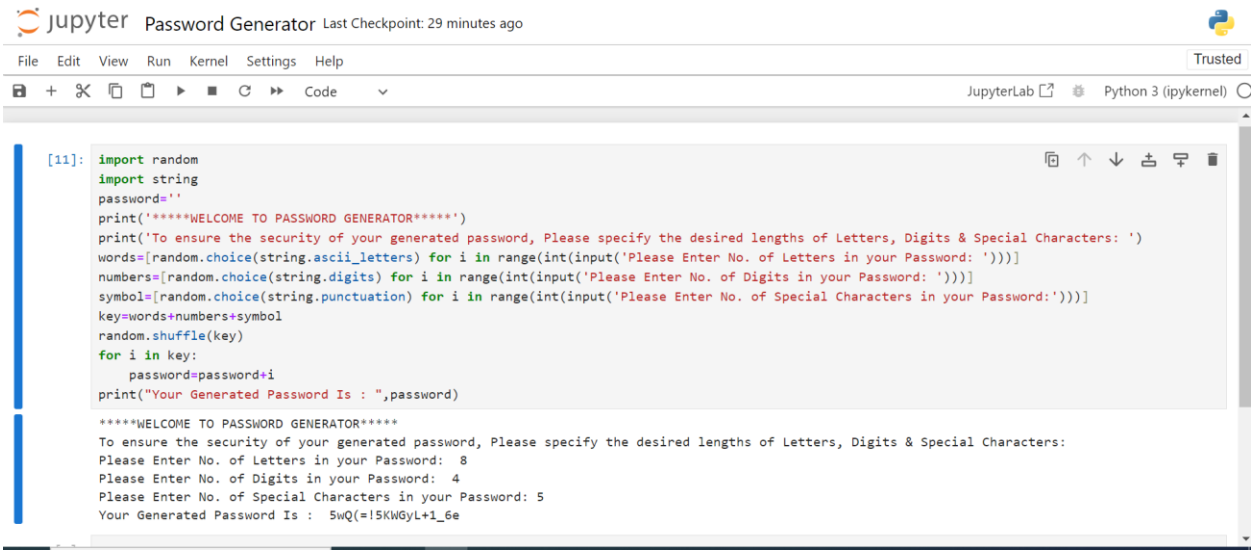
4. **Output:**

Finally, the program displays the generated password to the user, ensuring that it meets the specified criteria and is suitable for use.

Code Implementation:

```
import random

import string

password=''

print('*****WELCOME TO PASSWORD GENERATOR*****')

print('To ensure the security of your generated password, Please specify the desired lengths of Letters, Digits & Special Characters: ')

words=[random.choice(string.ascii_letters) for i in range(int(input('Please Enter No. of Letters in your Password: ')))]

numbers=[random.choice(string.digits) for i in range(int(input('Please Enter No. of Digits in your Password: ')))]

symbol=[random.choice(string.punctuation) for i in range(int(input('Please Enter No. of Special Characters in your Password:')))]

key=words+numbers+symbol

random.shuffle(key)

for i in key:

    password=password+i

print("Your Generated Password Is : ",password)
```

Output Snips:



Conclusion:

The Password Generator project provides a practical solution for generating strong and customizable passwords using Python within a Jupyter Notebook environment. By allowing users to specify the length and composition of the password, the generator ensures that the generated passwords meet individual security requirements. This project demonstrates the effective utilization of Python programming concepts and libraries to create a valuable tool for enhancing cybersecurity.