PASSWORD GENERATOR

PROJECT DOCUMENTATION

Project Title : Password Generator

Submitted by : Sadhasri Kathraji (258575)

Submitted to : Mrs. Gouthami Shyam

Coordinator : Mr. Inba

Institution : Besant Technologies

Date of Submission : 26 December 2024

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to the following individuals who have contributed to the successful completion of this project:

- Mrs. Gouthami Shyam, Trainer, Besant Technologies, for her guidance, support, and valuable feedback throughout the project.
- Mr. Inba, Co-ordinator, Besant Technologies, for his co-ordination and support.

I would also like to thank my institution, Besant Technologies, for providing me with the necessary resources and opportunities to complete this project.

Thank you.

Sadhasri K

ID : 258575

CONCEPTS USED IN THE PROJECT

The Password Generator project employs a combination of programming concepts and techniques to generate secure and unique passwords. The following concepts were utilized:

1. Python Programming Language: Python's simplicity, readability, and extensive libraries make it an ideal choice for this project. The project leverages Python's syntax and semantics to create a robust and efficient password generator.

2. Random Module: The random module is a built-in Python library that provides functionality for generating random numbers. In this project, the random module is used to select random characters from predefined character sets (lowercase letters, uppercase letters, digits, and symbols).

3. Conditional Statements (If-Else): Conditional statements are used to validate user input and ensure that the password meets the specified criteria. For example, the project checks if the user-input password length is within the allowed range.

4. Loops (For Loop): For loops are used to iterate through the password length and generate the password. The loop iterates for the specified number of times, selecting and appending random characters to the password string.

5. User Input: The project utilizes user input to specify the password length and character composition. The user is prompted to enter the desired password length, as well as the number of lowercase letters, uppercase letters, digits, and symbols to include in the password.

6. String Manipulation: The project involves string manipulation techniques, such as concatenation, to generate and modify the password string.

7. Python Lists: Python lists are used to store and manipulate the character sets, including lowercase letters, uppercase letters, numbers, and symbols.

By incorporating these concepts, the Password Generator project demonstrates a solid understanding of programming fundamentals and applies them to create a practical and useful application.

SOURCE CODE

```
import random
alpha_lower = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't',
'u', 'v', 'w', 'x', 'y', 'z']
alpha_upper = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q',
'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']
numbers = ['0','1','2','3','4','5','6','7','8','9']
symbols = ['@','#','$','%','&','*','9','0']

print("Welcome to the Password Generator!!")

l = int(input("Enter the no. of lower case letters to include in your password:"))
u = int(input("Enter the no. of upper case letters to include in your password:"))
n = int(input("Enter the no. of numbers to include in your password:"))
s = int(input("Enter the no. of symbols to include in your password:"))

password = ""
for i in range(l):
    a = random.choice(alpha_lower)
    password += a

for i in range(u):
    b = random.choice(alpha_upper)
    password += b
```

```
for i in range(n):
    c = random.choice(numbers)
    password += c

for i in range(s):
    d = random.choice(symbols)
    password += d

if len(password) > 6:
    print("Total password length exceeds 6 characters.")
else:
    print(f"Your password is: {password}")
```

DESCRIPTION OF SOURCE CODE

Importing Modules

The project begins by importing the necessary modules. In this case, the random module is imported to generate random numbers and characters.

```
import random
```

Defining Character Sets

The project defines character sets for lowercase letters, uppercase letters, numbers, and symbols.

```
alpha_lower = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z']

alpha_upper = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']

numbers = ['0','1','2','3','4','5','6','7','8','9']

symbols = ['@','#','$','%','&','*','9','0']
```

User Input

The project prompts the user to input the number of lowercase letters, uppercase letters, numbers, and symbols to include in the password.

```
l = int(input("enter the no. of lower case letters to include in your password:"))

u = int(input("enter the no. of upper case letters to include in your password:"))

n = int(input("enter the no. of numbers to include in your password:"))

s = int(input("enter the no. of symbols to include in your password:"))
```

Password Generation

The project generates a password by randomly selecting characters from the defined character sets based on the user's input.

```
password = ""
for i in range(l):
    a = random.choice(alpha_lower)
    password += a
for i in range(u):
    b = random.choice(alpha_upper)
    password += b
for i in range(n):
    c = random.choice(numbers)
    password += c
for i in range(s):
    d = random.choice(symbols)
    password += d
```

Printing the Password

Finally, the project prints the generated password.

```
print(password)
```

OUTPUT

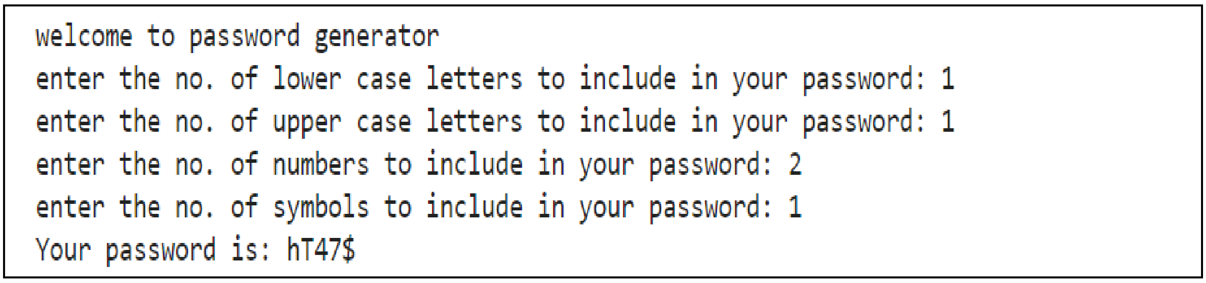
```
Welcome to Password Generator!!
Enter the no. of lower case letters to include in your password: 2
Enter the no. of upper case letters to include in your password: 2
Enter the no. of numbers to include in your password: 1
Enter the no. of symbols to include in your password: 1
F4bA@p
```

In this example, the user requested a password with:

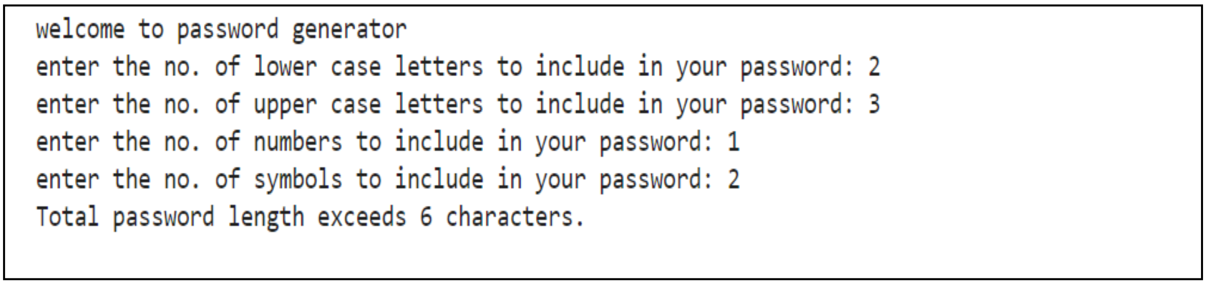
- 2 lowercase letters (b, p)
- 2 uppercase letters (F, A)
- 1 number (4)
- 1 symbol (@)

The generated password is a random combination of these characters.

OUTPUT SCREENS



```
welcome to password generator
enter the no. of lower case letters to include in your password: 1
enter the no. of upper case letters to include in your password: 1
enter the no. of numbers to include in your password: 2
enter the no. of symbols to include in your password: 1
Your password is: hT47$
```



```
welcome to password generator
enter the no. of lower case letters to include in your password: 2
enter the no. of upper case letters to include in your password: 3
enter the no. of numbers to include in your password: 1
enter the no. of symbols to include in your password: 2
Total password length exceeds 6 characters.
```

CONCLUSION & BIBLIOGRAPHY

The Password Generator project demonstrates the application of Python programming concepts to create a secure and efficient password generator. Through this project, we have explored the use of Python lists, random module, conditional statements, loops, and string manipulation techniques. The project provides a user-friendly interface for generating passwords based on user-specified criteria.

Future Enhancements

Potential future enhancements to the project include:

- Implementing additional password complexity rules

- Integrating a password storage mechanism

- Developing a graphical user interface (GUI) for the password generator

Bibliography

No external references were used