Password Generator

[Narayan Joshi, Python, Internship]

[Python]

Author Note

[Include any grant/funding information and a complete correspondence address.]

[Password Generator which generates password according to length]

Introduction

This report describes a Python program for generating passwords of varying lengths and complexities. The program utilizes the `random` and `string` modules to create passwords consisting of lowercase letters, uppercase letters, digits, and special characters.

Method

The program defines a function `generate\_password(length)` that takes a parameter `length` to specify the desired length of the password. It then defines character sets for different complexity levels, including lowercase letters, uppercase letters, digits, and special characters. These character sets are combined to form a single set, `all\_chars`, from which the password is generated using a loop that selects characters randomly. The generated password is then returned by the function.

Results

The program prompts the user to enter the desired length of the password and generates a password based on the input length. If the user enters a non-integer value, the program displays an error message and prompts the user to enter a valid length.

Discussion

The program provides a simple and effective way to generate passwords with varying levels of complexity. By allowing the user to specify the length of the password, it offers flexibility in creating passwords for different security requirements. However, the program does not include a check for password strength, which could be a potential improvement.

Conclusion

In conclusion, the Python program presented in this report demonstrates a basic approach to generating passwords of varying lengths and complexities. While it provides a useful tool for creating passwords, further enhancements could be made to improve its security and usability.

References

References are taken from CHATGPT and CodeWithHarry.

Weekly Progress Report

Prepared by: Narayan Joshi

Introduction:

This report outlines the progress made during the week 3 on the project Password Generator. It highlights completed tasks, milestones achieved, challenges faced, and lessons learned.

Completed Tasks:

- Developed a Python program for generating passwords of varying lengths and complexities.

- Implemented a function to generate passwords based on user-specified length.

- Tested the program for functionality and correctness.

Milestones Achieved:

- Successfully generated passwords containing lowercase letters, uppercase letters, digits, and special characters.

- Ensured user input validation for the password length.

Significant Contributions:

- Provided insights into the design and implementation of the password generation algorithm.

- Collaborated with team members to ensure the program meets project requirements.

Challenges and Hurdles:

- One of the challenges faced was validating user input for the password length, especially handling non-integer inputs.

- Overcame this hurdle by implementing a try-except block to catch ValueError and prompt the user to enter a valid length.

Lessons Learned:

- Learned the importance of user input validation in ensuring program robustness.

- Acquired knowledge in using the `random` and `string` modules for generating random passwords.

- Developed problem-solving skills by overcoming challenges in the implementation process.

Conclusion:

Overall, the week was productive, with significant progress made on the password generation program. The challenges encountered provided valuable learning experiences that will be beneficial for future projects.