**Password Strength Checker - Project Documentation**

**1. Project Overview**

The **Password Strength Checker** is a Python-based command-line tool that evaluates the strength of a user-provided password. It analyzes the password based on character diversity, including uppercase letters, lowercase letters, numbers, whitespace, and special characters. Based on the composition, it assigns a strength score and provides feedback on improving password security.

**2. Objectives**

* To create a simple yet effective tool for assessing password security.
* To educate users on strong password creation.
* To implement basic cybersecurity best practices using Python.

**3. Features**

* **Password Input Handling**: Accepts a user-inputted password securely.
* **Character Analysis**: Counts different types of characters (uppercase, lowercase, digits, whitespace, special characters).
* **Strength Scoring System**:
  + Assigns a score from 1 to 5 based on password complexity.
  + Provides feedback and suggestions to improve password security.
* **User Interaction**:
  + Allows users to check multiple passwords in one session.
  + Handles incorrect inputs gracefully.

**4. Technology Stack**

* **Programming Language**: Python
* **Libraries Used**:
  + string (for character validation)
  + getpass (for secure password input, but can be replaced with input() in certain environments)

**5. How It Works**

1. The program starts with a welcome message.
2. The user is prompted to check a password.
3. The password is analyzed based on the following criteria:
   * Presence of lowercase letters
   * Presence of uppercase letters
   * Presence of numbers
   * Presence of special characters
   * Presence of whitespace
4. A strength score is assigned:
   * **1/5**: Very weak
   * **2/5**: Weak
   * **3/5**: Moderate
   * **4/5**: Strong
   * **5/5**: Very strong
5. The program provides feedback and displays the character breakdown.
6. The user is asked whether they want to check another password.
7. The program repeats or exits based on user input.

**6. Use Cases**

**6.1. Personal Security Awareness**

Users can test the strength of their passwords before using them for accounts, ensuring they are strong enough to resist attacks.

**6.2. Cybersecurity Training**

This project can be used in cybersecurity awareness programs to educate users on password best practices.

**6.3. Integration with Security Systems**

This logic can be integrated into applications requiring password validation before allowing user account creation.

**7. Sample Output**

+++ Welcome to Password Checker +++

Do you want to check a password? (y/n): y

Enter Password: \*\*\*\*\*\*\*\*

Your password has:

2 lowercase characters

2 uppercase characters

3 numeric characters

0 whitespace characters

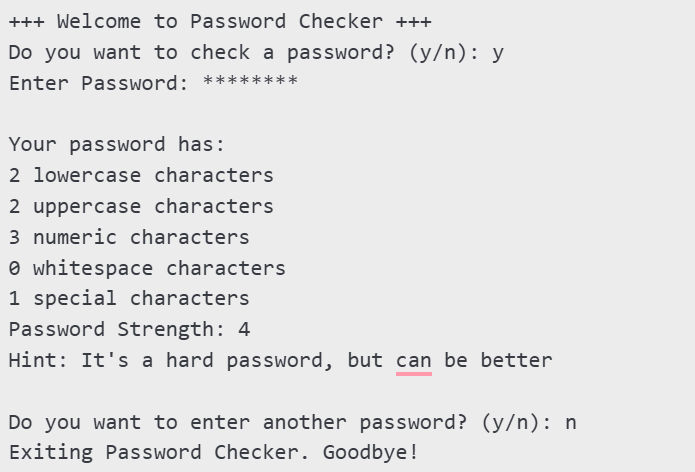
1 special characters

Password Strength: 4

Hint: It's a hard password, but can be better

Do you want to enter another password? (y/n): n

Exiting Password Checker. Goodbye!



**8. Future Improvements**

* Implement GUI version using Tkinter or PyQt.
* Use a database to store and analyze password trends.
* Add dictionary-based password checks to warn users about common passwords.
* Introduce entropy calculations to measure password randomness.

**9. Conclusion**

The Password Strength Checker is a useful tool for evaluating password security. By encouraging users to create stronger passwords, it helps in reducing cybersecurity risks. With future improvements, it can be expanded into a more robust security tool.