Python Password Strength Checker Project

Altolane Jackson (AJ)

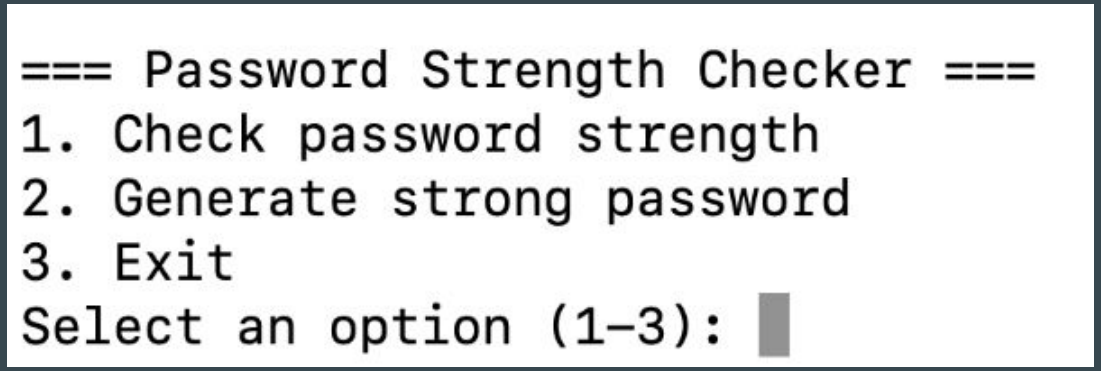
Password Strength Checker (Python CLI Project)

Project Type: Python Terminal-Based App

- helps users evaluate their password security and generate strong passwords.
- It runs in the terminal and follows key software development principles.

Why this project?

- Strong passwords are a crucial part of cybersecurity.
- This tool helps educate users about what makes a password strong.
- It also shows how simple logic and Python structures can create useful tools.



Features Overview

- There's an interactive menu: check a password, generate one, or exit.
- If a password is weak, the app gives improvement suggestions.
- It uses a custom scoring system based on length and character variety.

```
=== Password Strength Checker ===
1. Check password strength
2. Generate strong password
3. Exit
Select an option (1-3): 2
```

```
Select an option (1-3): 1
Enter your password: fycug

Strength Score: 1 / 6
Feedback: ❌ Weak Password

Suggestions to improve:
- Password is too short. Use at least 8 characters.
- Add uppercase letters.
- Include numbers.
- Include special characters (e.g. !@#$%).
```

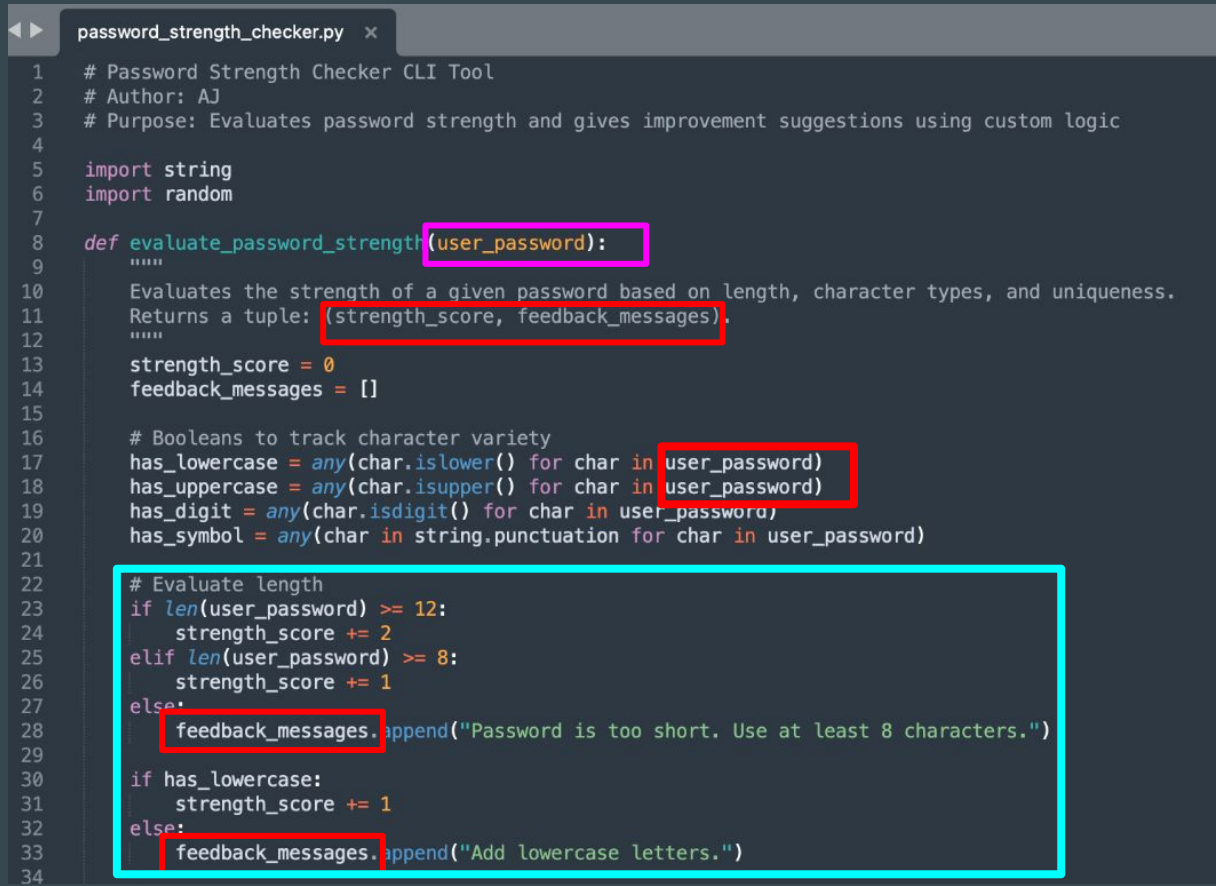
```
Strength Score: 3 / 6
Feedback: ⚠️Moderate Password

Suggestions to improve:
- Add lowercase letters.
- Include special characters (e.g. !@#$%).
```

```
Enter your password: AJwnvvownv234567!@#$

Strength Score: 6 / 6
```

```
Enter your password: Aj12345!@#

Strength Score: 5 / 6
Feedback: ✅ Strong Password
```

Programming Requirements

- **Descriptive Variable Names:** Used clear, meaningful names like user_password, strength_score, feedback_messages
- **Three Data Types:** Included strings (password input), integers (score values), and booleans (character type checks)*
- **Decision-Making Structures:** Used if, elif, and else to score strength and guide user interaction



- **Loops for Repeated Tasks:**
  while True loop enables continuous user access without restarting the program

- **Reusable Functions:**
  Defined evaluate_password_strength() and generate_strong_password() for clean, modular logic

- **List (Sequence) Use:**
  Stored improvement tips in feedback_messages[] and iterated through them with a for loop

- **Code Documentation:**
  Docstrings and inline comments throughout explain each function's purpose and logic