Title: Password Strength Analyzer using Python

**1. Introduction**

In today’s digital world, securing user accounts with strong passwords is crucial. Weak and predictable passwords are the most common cause of security breaches. To address this, I developed a **Password Strength Analyzer** in Python. This tool helps users test their password strength and provides feedback to create more secure password.

**2. Abstract**

The project implements a **command-line password strength analyzer** using Python and the **zxcvbn library**, which estimates how easily a password can be cracked. The analyzer evaluates password complexity, estimates cracking time, and provides suggestions for improvement. The tool allows multiple password inputs and continues until the user exits. This project demonstrates secure programming practices, user interaction, and use of external security libraries.

**3. Tools Used**

* **Python 3.13** – Core programming language
* **zxcvbn-python** – Library for password strength estimation
* **Virtual Environment (venv)** – For dependency management
* **Command Prompt (Windows)** – For running the project

**4. Steps Involved**

1. **Environment Setup**

* Installed Python 3.13 and created a virtual environment.
* Installed the zxcvbn-python library using pip.

1. **Code Development**
   * Imported zxcvbn library in the script.
   * Created a loop to accept user passwords until exit is typed.
   * Analyzed password strength, crack time, and feedback.
   * Displayed results in a clear, user-friendly format.
2. **Testing**
   * Tested weak passwords (e.g., 12345, maya@2004#).
   * Tested strong passwords (e.g., MyPassword@2025!, 1268Maya@5051#!).
   * Verified program output against expected results.

**5. Testing & Results**

Below are some sample outputs from the program:

Password: 12345

Strength score: 0

Estimated crack time: less than a second

Feedback: Top-10 common password.

Password: maya@2004#

Strength score: 3

Estimated crack time: less than a second

Password: MyPassword@2025!

Strength score: 4

Estimated crack time: 2 seconds

Password: 1268Maya@5051#!

Strength score: 4

Estimated crack time: 11 hours

Password: Maya1265@5051#!

Strength score: 4

Estimated crack time: 5 hours

These results confirm that the analyzer can detect **very weak**, **moderately strong**, and **very strong** passwords, while providing suggestions for improvement.

**6. Conclusion**

The Password Strength Analyzer successfully demonstrates how Python can be used to build a security-focused application. The tool educates users about the importance of strong passwords and provides actional libraries for Future improvements could include a graphical user interface (GUI) and saving results for reporting.