**Project Plan and Scope for Strong Password Detection**

**Project Repository Link** - [Strong Password Detection - GitHub Repository](https://github.com/donaldreginepo/Strong-Password-Detection.git)

**Project Goals**

The primary objective of this project is to develop a Python tool that helps users determine the strength of their passwords. This tool aims to promote enhanced security practices by ensuring passwords meet defined robustness criteria, thereby contributing to safer digital environments.

**Problem Statement**

Weak passwords are a significant vulnerability in securing online information. This project addresses the need for automated strong password verification, aiding users in creating passwords that are more resistant to cyberattacks, including those discussed in network and Windows environments.

**Expected Outcomes and Deliverables**

* **Python Script:** A script that checks passwords against specified criteria for strength.
* **Command-Line Interface:** An interface for users to easily test their passwords.
* **Documentation and User Guide:** Detailed guidelines on how the tool works and how to deploy it.
* **Demonstration Video:** A video tutorial demonstrating the tool in action.
* **Integration of Security Insights:** Incorporating best practices and current trends from security literature.

**Timeline**

**Week 1: Project Initiation and Rapid Literature Review**

* Project Setup: Set up the GitHub repository and prepare the development environment.
* Literature Review: Conduct a swift yet thorough review of relevant security literature to establish a solid foundation for defining password strength criteria. Key resources will include Sandbu's insights on ransomware protection, Bijalwan's network forensics, and contemporary articles on password security measures.
* Define Security Criteria: By the end of the week, finalize the security criteria for password strength based on reviewed literature and current best practices.

**Week 2: Development Sprint**

* Script Development: Begin coding the Python script with an initial focus on core functionalities—regular expression checks for password strength criteria (length, character variety, digit inclusion).
* User Interface Development: Start the design and implementation of a simple, user-friendly command-line interface that allows users to input passwords and receive feedback.
* Initial Testing: Conduct basic tests to ensure the script's core functionalities are working as intended.

**Week 3: Refinement and Enhanced Testing**

* Refinement of Features: Refine and expand the script and interface based on initial testing feedback. Enhancements might include adding more detailed user feedback for weak passwords.
* Comprehensive Testing: Implement thorough testing scenarios to identify any bugs and ensure the tool meets all defined criteria effectively. Test various password combinations extensively to validate all aspects of the tool.
* Start Documentation: Begin drafting documentation and user guidelines detailing how to install, use, and understand the tool’s password evaluations.

**Week 4: Finalization and Launch**

* Complete Documentation: Finalize the documentation, ensuring it's clear, informative, and helpful for end-users.
* Video Creation: Produce a concise, informative demonstration video that showcases how to use the tool, its features, and example outputs.
* Project Review and Launch: Review the entire project for any last-minute tweaks and ensure all elements are polished. Release the final version on the GitHub repository.
* Project Submission: Submit the project deliverables including the GitHub link, documentation, and demonstration video.

**Integration of Security Literature**

To ensure the tool aligns with modern security practices and provides substantial educational value, it will incorporate insights from the following references:

1. **Sandbu, M. (2023).** This reference will help in understanding the measures necessary for ransomware protection and the role of strong passwords in securing Windows endpoints.
2. **Bijalwan, A. (2022).** This book offers comprehensive knowledge on network forensics and security which will be used to contextualize the importance of password strength within overall network security frameworks.
3. **Gibner, J. (2004).** Insights from this article will be used to define what constitutes a strong password and how tools like **passprop.exe** can be leveraged to improve password security.
4. [Nulab's Guide on Password Strength](https://nulab.com/learn/software-development/password-strength/)**:** This online resource will provide contemporary understandings and methods for assessing password strength, which will inform the criteria used in the tool.
5. [IEEE INFOCOM 2010 Proceedings](https://ieeexplore.ieee.org/document/5461951)**:** This conference paper provides relevant technological context and latest research findings that can influence the development and effectiveness of security tools like the one being developed.

**The Tool**

**Description**

The Strong Password Detection tool utilizes regular expressions to verify if a user's password meets the following criteria:

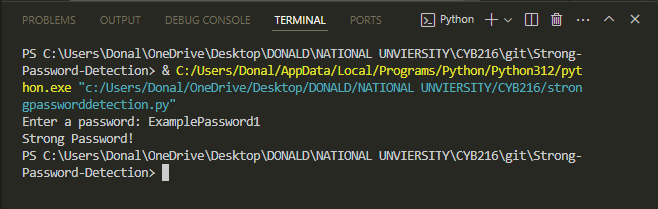
* Minimum of eight characters.
* Inclusion of both uppercase and lowercase letters.
* Presence of at least one numerical digit.

**Demonstration of the Tool**

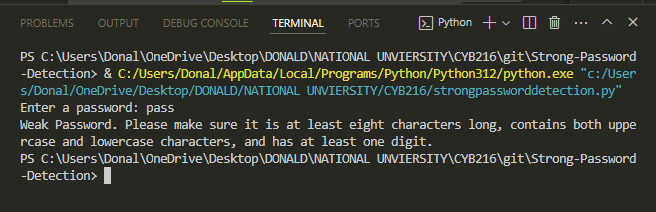
Users input their password into the command-line interface, and the tool evaluates it against the security criteria, providing immediate feedback on the password's strength.

**Example of Usage**

Strong password feedback:



Feedback for insufficient strength:



**Conclusion**

Through rigorous documentation and adherence to coding best practices, this project not only meets the academic requirements but also provides a real-world application that enhances users' digital security. This approach ensures the longevity and utility of the project, making it a valuable resource for anyone interested in learning about or improving password security.

**Recommendations**

To further enhance the tool, future developments could integrate multi-factor authentication checks and provide more granular feedback on password weaknesses.