**🔐 Proposal for Password Strength & Validity Checker**

**Project Title:**

🚀 **Advanced Password Strength & Security Validator**

**Team Members:**

1. **Irum Iman**
2. **Rania Khan**
3. **Memoona Shafiq**

**Introduction:**

In today’s digital era, weak passwords pose a serious security risk, making users vulnerable to hacking, identity theft, and data breaches. Many users still rely on predictable passwords, which cybercriminals can easily exploit.

Our project, **Password Strength & Validity Checker**, is designed to evaluate the security of a password based on **length, complexity, entropy, and blacklist detection**. This tool will provide **real-time feedback** to users, guiding them toward creating **stronger, more secure passwords** while also ensuring password validity through confirmation mechanisms.

**Objectives:**

✔️ Develop a **command-line-based** password strength and validity checker using **Assembly Language (MASM & Irvine32).**  
✔️ **Classify passwords** into **Weak, Medium, or Strong** based on security parameters.  
✔️ **Detect commonly used passwords** by implementing a blacklist feature.  
✔️ **Integrate entropy analysis** to measure randomness and security strength.  
✔️ **Ensure password validity** by requiring re-entry for confirmation.  
✔️ Provide an **interactive user experience** with prompts and suggestions.  
✔️ Implement a **password generator** to suggest strong alternatives for weak passwords.

**Methodology:**

**📌 Password Input & Processing:**

* The user **enters a password**, which is analyzed character by character.
* The system checks for **uppercase letters, lowercase letters, numbers, and special characters** to assess complexity.

**📌 Classification Logic:**

🔴 **Weak Passwords:**

* Less than **8 characters**
* Commonly used passwords (e.g., "password123", "qwerty")
* Lacks a mix of different character types

🟡 **Medium Passwords:**

* At least **8 characters**
* Contains a mix of letters and numbers but lacks special characters

🟢 **Strong Passwords:**

* At least **8 characters**
* **Contains uppercase letters, numbers, and special characters**
* High entropy value, making it hard to guess

**📌 Advanced Features:**

✅ **Blacklist Check:**

* Compares the entered password against a **list of commonly hacked passwords**.
* Prevents users from using predictable passwords.

✅ **Entropy Calculation:**

* Uses **Shannon Entropy** to measure the **randomness** and **strength** of a password.
* Helps determine how resistant a password is to brute-force attacks.

✅ **Password Confirmation:**

* Users must **re-enter** their password to confirm and avoid typographical errors.

✅ **Password Suggestion & Generator:**

* If a password is weak, the system **suggests stronger alternatives**.
* Option to **automatically generate a secure password** based on best security practices.

**Tools & Technologies:**

🛠 **Programming Language:** Assembly (MASM, Irvine32)  
🖥 **Platform:** Windows Console Application  
🔍 **Security Algorithms:** Entropy Calculation, Blacklist Checking, Character-Based Analysis

**Expected Outcome:**

🔹 A **fully functional password security tool** that enhances user security awareness.  
🔹 A system that **prevents weak passwords** and helps users create **stronger, more secure credentials**.  
🔹 A practical implementation of **cybersecurity principles** using **low-level programming**.

**Conclusion:**

With the growing number of **cyberattacks**, ensuring **password security** is more critical than ever. By developing this **Password Strength & Validity Checker**, we aim to provide a **reliable, efficient, and interactive tool** to help users create **stronger, more secure passwords**.

Additionally, this project will serve as a **practical demonstration of Assembly language programming**, incorporating **security best practices** in a **low-level computing environment**.

🔒 **Stronger Passwords. Better Security. Safer Digital World.**