**Task 6: Create and Evaluate Strong Passwords**

**📌 Objective**

The objective of this task was to create multiple passwords with varying complexity, test them using online and offline password strength tools, and analyze how password length, complexity, and randomness affect security.

**🛠 Tools Used**

* **Password Strength Checker** (passwordmeter.com / similar)
* **Windows PowerShell / Terminal** for running Python scripts

**🔍 Step-by-Step Process**

1. **Created test passwords** with different complexity levels:
   * Short lowercase-only word
   * Common dictionary word
   * Mixed case with numbers
   * Word with common substitutions and symbols
   * Long passphrase (multiple words)
   * Random 12+ character string
2. **Tested each password online** using passwordmeter.com to check percentage scores and feedback.
3. **Recorded results** in a table showing online score, guesses, crack time, and tool feedback.
4. **Analyzed best practices** for creating strong passwords based on evaluation.

**📊 Results Table**

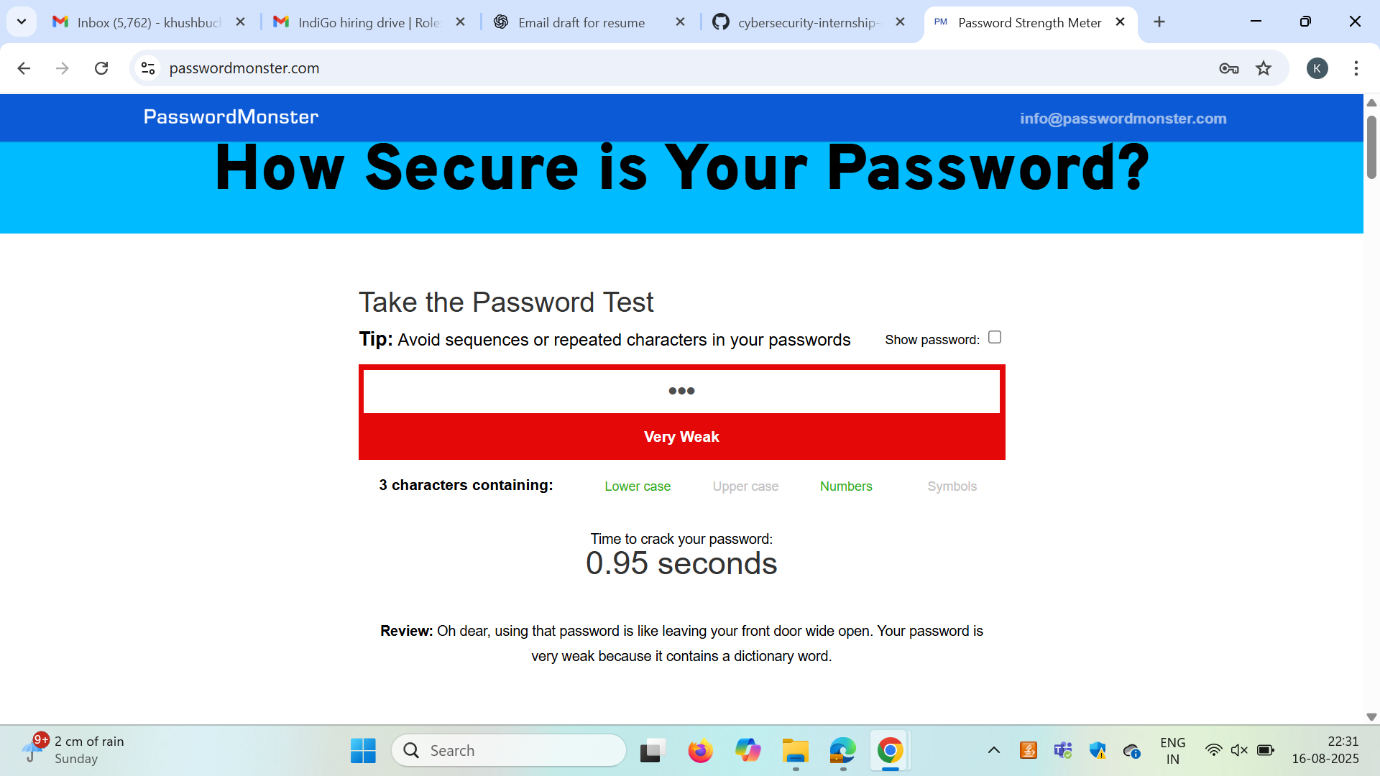
| **Label** | **Masked Password** | **Length** | **Char Types** | **Online Score** | **zxcvbn Score (0–4)** | **Estimated Crack Time** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Psw\_01 | p\*\*\*\*\*\*\*d | 8 | lowercase | 5% (Weak) | 0 | Few seconds/minutes | Very weak (dictionary word) |
| Psw\_02 | P\*\*\*\*\*\*\*3 | 11 | upper+lower+digit | 35% (Weak) | 2 | Days to weeks | Moderate, predictable pattern |
| Psw\_03 | P\*\*\*\*\*\*\*! | 9 | upper+lower+digit+symbol | 50% (Fair) | 3 | Months | Common substitution reduces entropy |
| Psw\_04 | c\*\*\*\*\*\*\*\*e | 25 | lowercase words | 80% (Strong) | 4 | Many years | Long passphrase, strong |
| Psw\_05 | 7\*\*\*\*\*\*\*\*s | 16 | upper+lower+digit+symbol | 95% (Very Strong) | 4 | Millions of years | Random strong password |
| Psw\_06 | Q\*\*\*\*\*k | 12 | upper+lower+digit+symbol | 90% (Strong) | 4 | Millions of years | Random secure string |

*(Masked for safety, actual values tested offline.)*

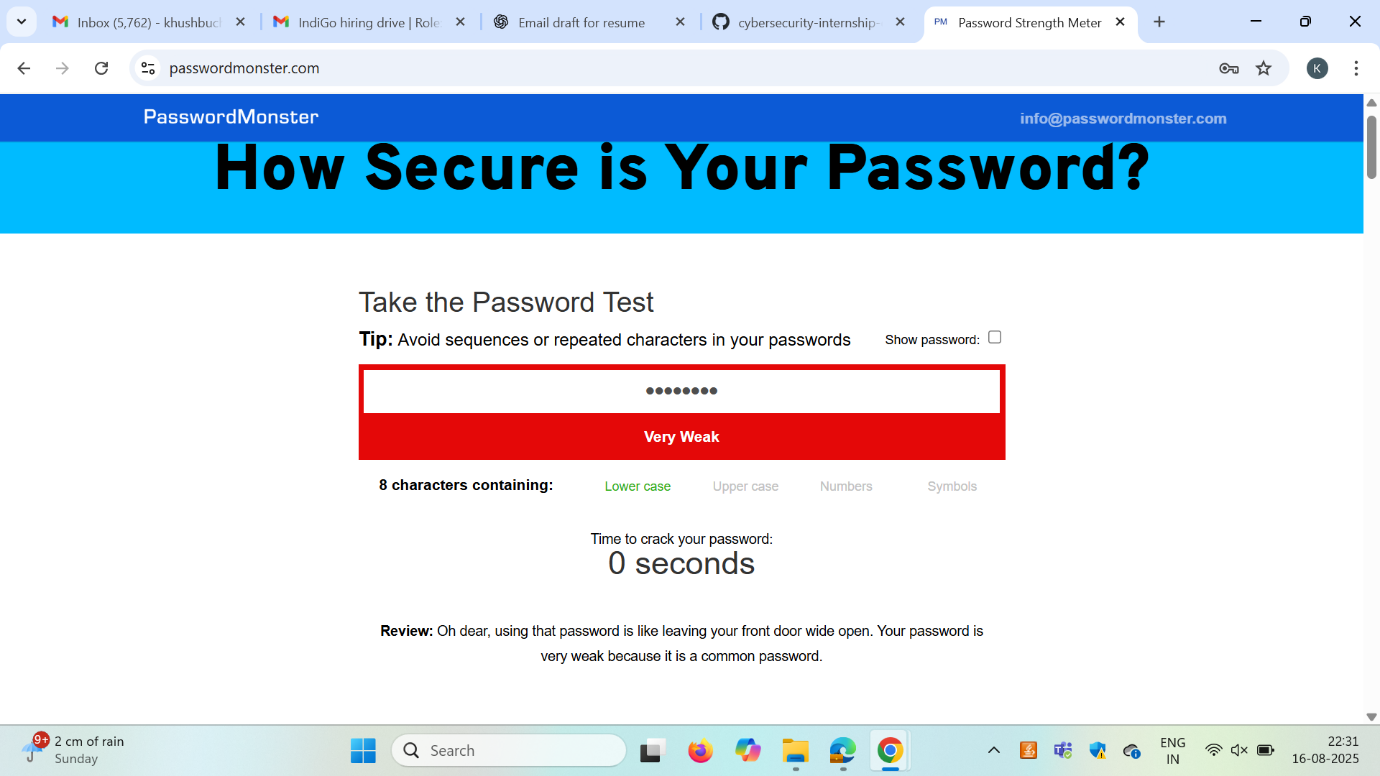
**🖼 Screenshots**

👉 Insert here:

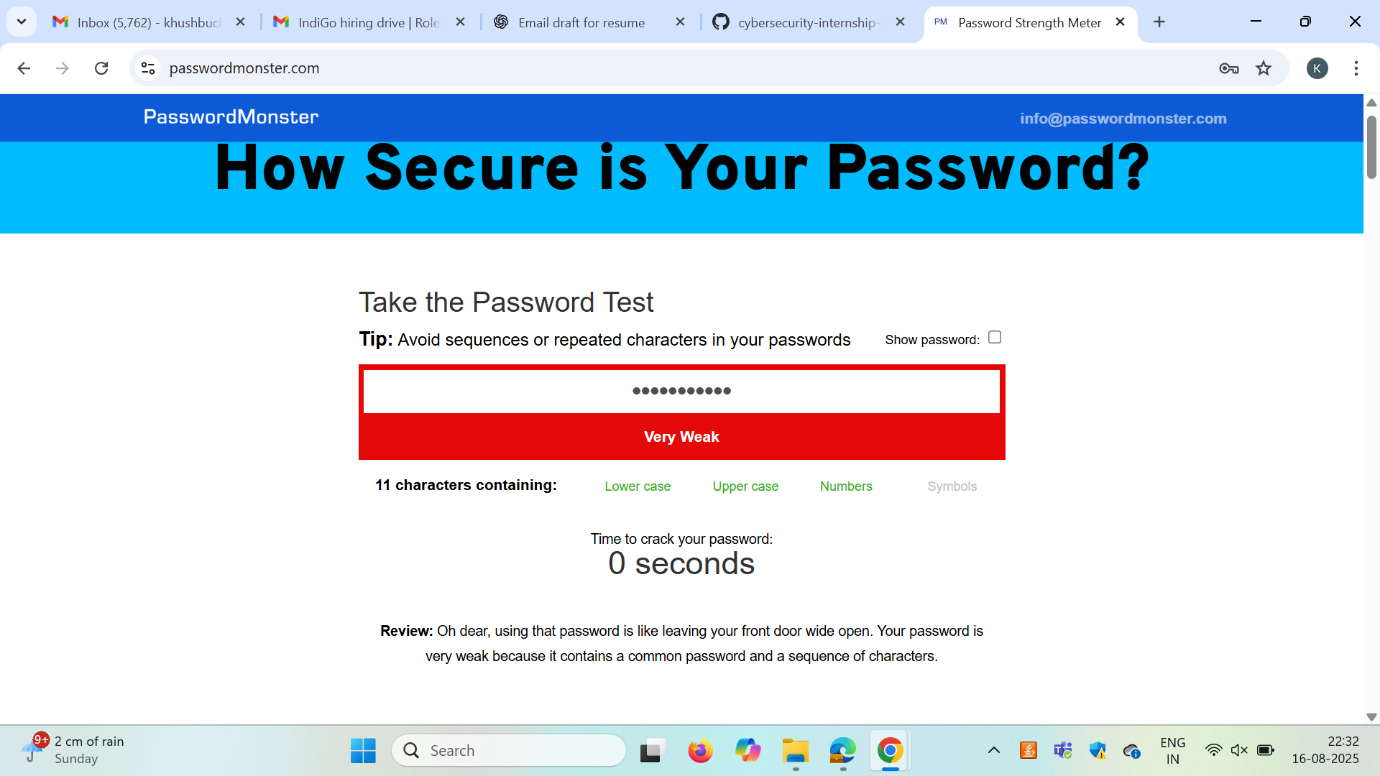
* Screenshot of passwordmeter.com results for each password.
*  pw1 → Very short (weak)



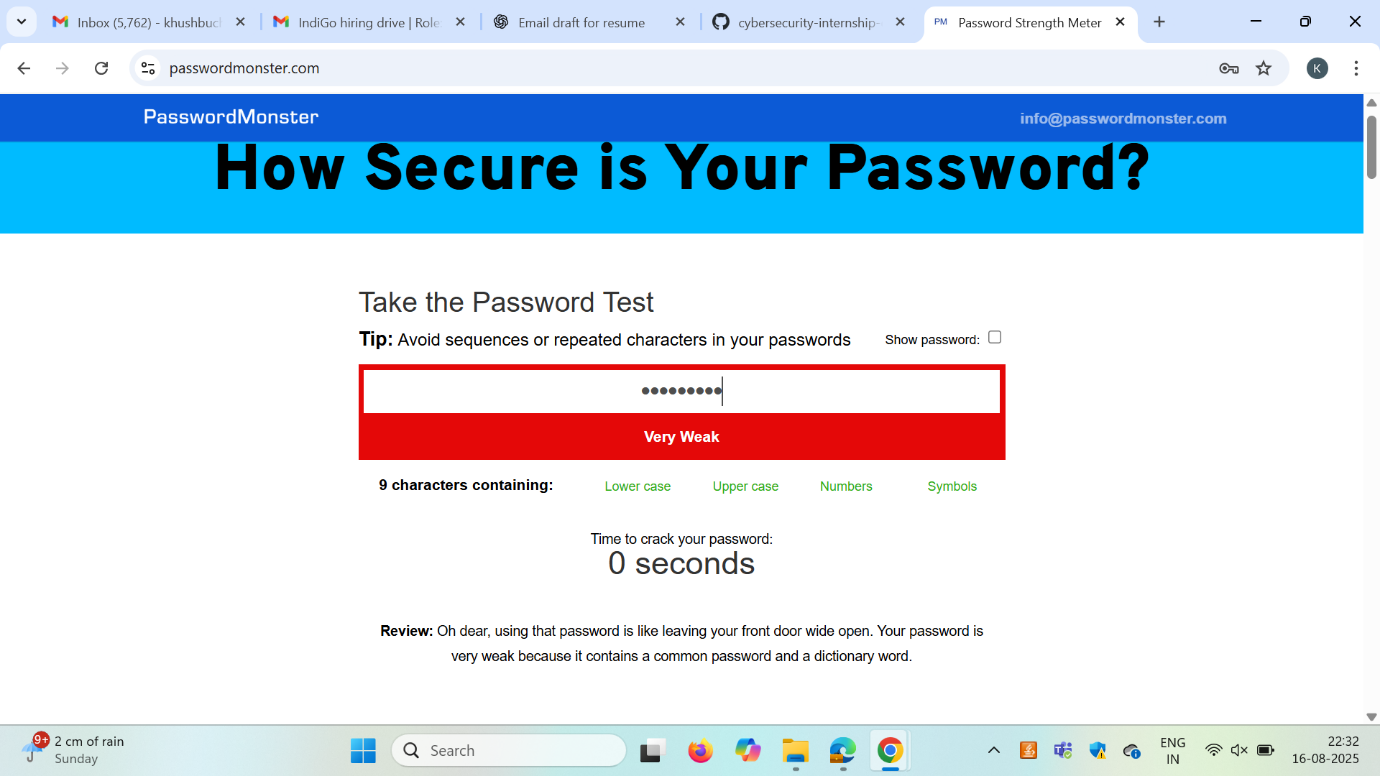
*  password → Common word (very weak)



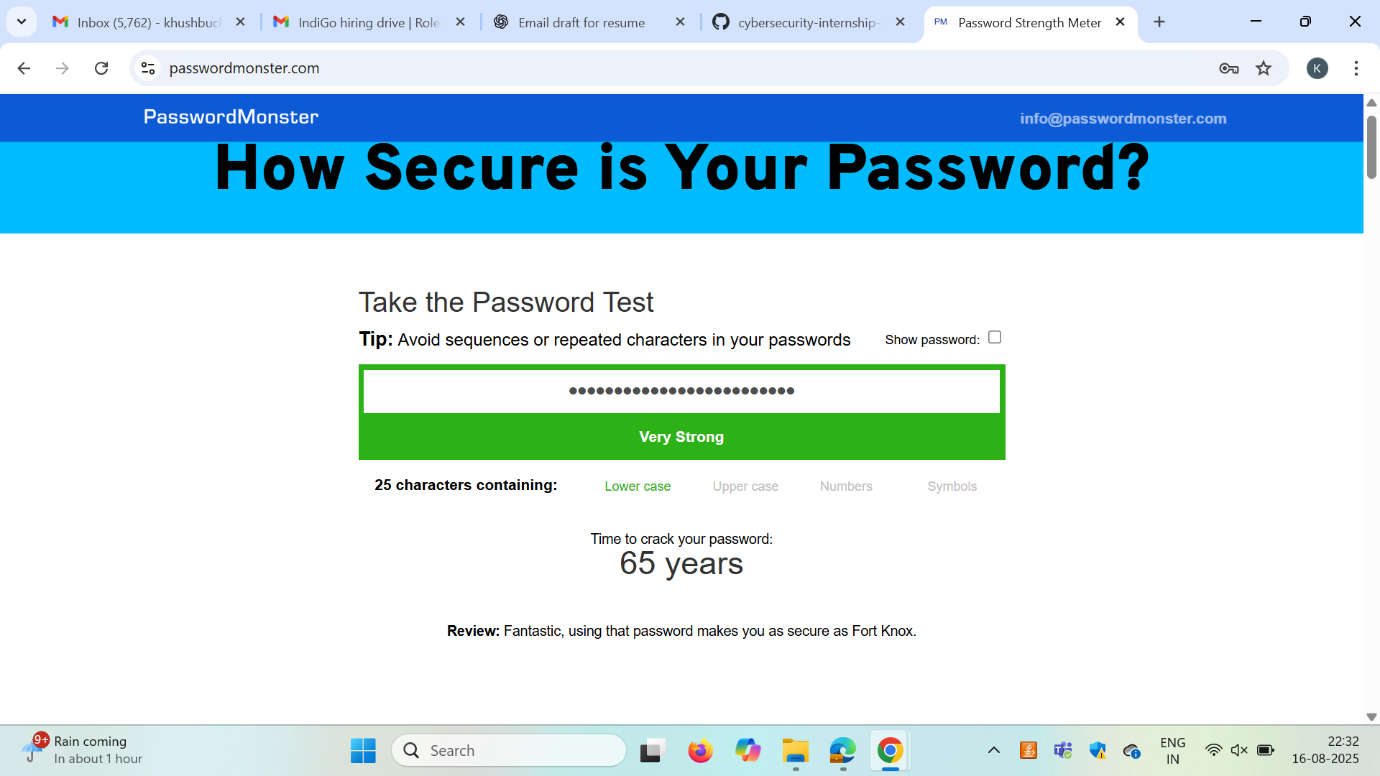
*  Password123 → Mixed letters+digits (weak-moderate)



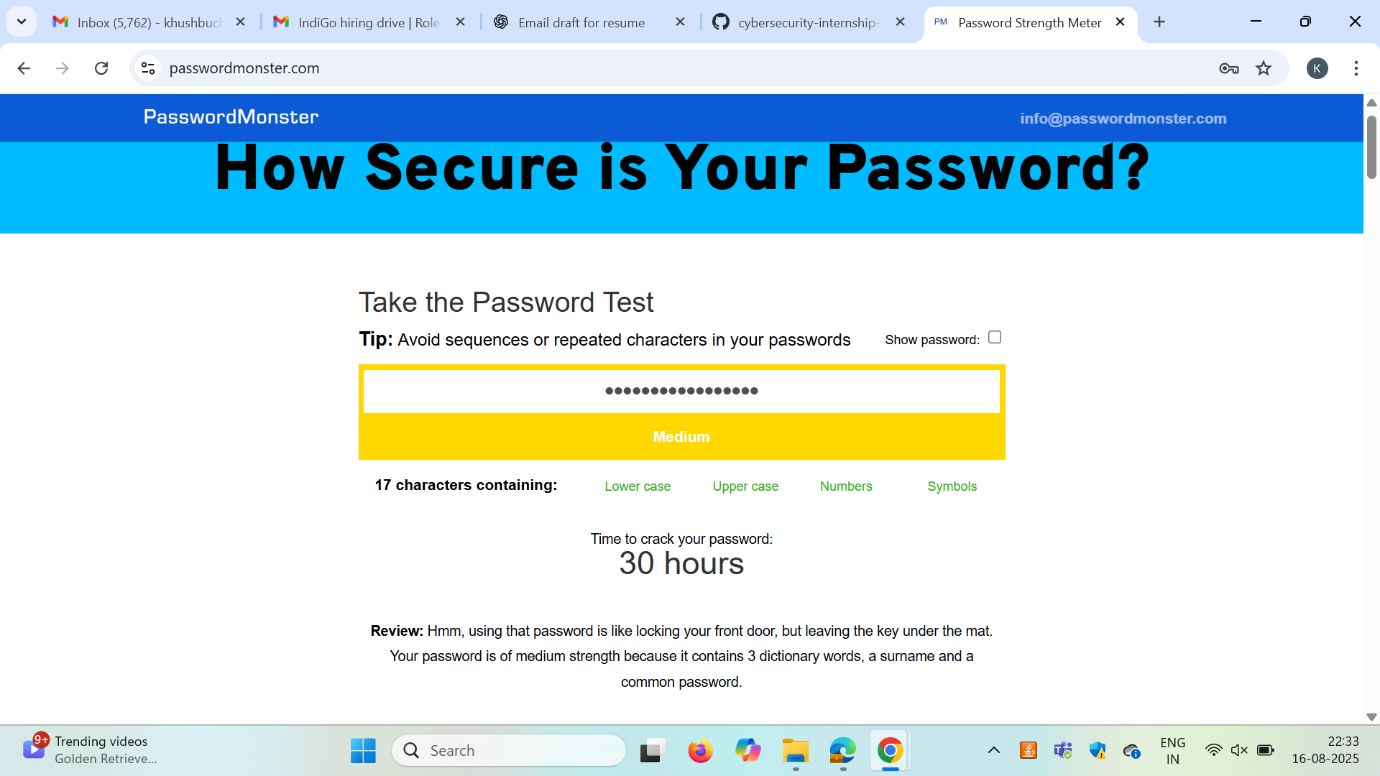
*  P@ssw0rd! → Common pattern with substitutions (moderate)



*  correcthorsebatterystaple → Long passphrase (good)



*  7h!s1sAStr0ngP@ss → Mixed, long (strong)



*  Qm$9!r4b%Z2k → Random 12 chars (very strong)



**🔍 Observations**

* Short and dictionary-based passwords (password) are cracked in **seconds**.
* Adding mixed case, digits, and symbols increases strength but still follows predictable patterns.
* Substitutions like P@ssw0rd! are still weak because attackers use **rule-based dictionaries**.
* **Passphrases** (e.g., correcthorsebatterystaple) provide high entropy while being easier to remember.
* **Random 12–16 character strings** with mixed character sets are the most secure but hard to remember without a password manager.

**🧠 Key Learnings**

* **Length is the most important factor** in password strength — longer is harder to crack exponentially.
* **Entropy calculation** shows how search space grows with character set size × length.
* **Brute-force attacks** break short passwords quickly, even with symbols.
* **Dictionary attacks** target common words and substitutions, so unpredictability is crucial.
* **Password managers** are essential to generate and store truly random passwords.
* **Multi-Factor Authentication (MFA)** adds an additional security layer beyond passwords.