**Password strength analysis**

**Internship Task – Elevate Labs**

**Submitted By:**  
Aditya Tiwari

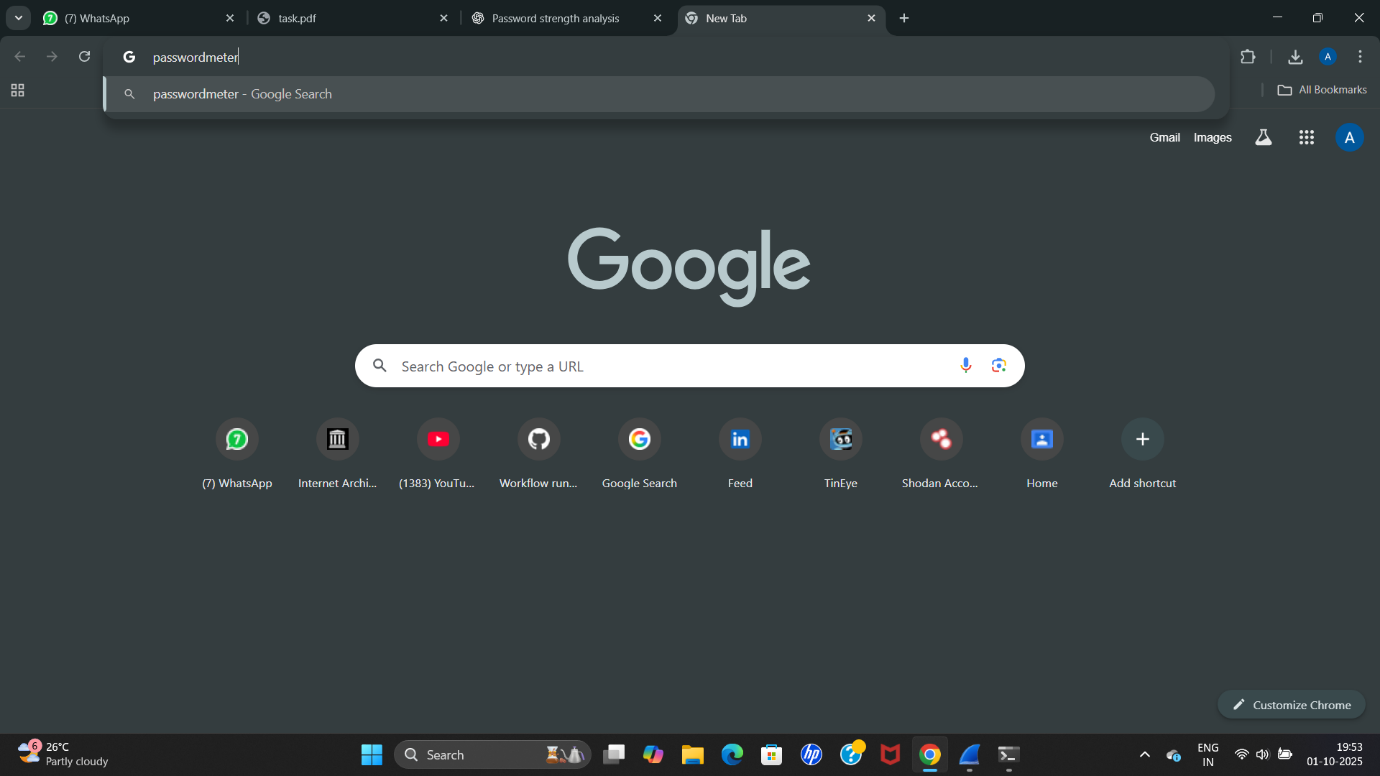
**Internship Mentor / Company:**  
Elevate Labs

**Tools Used:**

 [PasswordMeter](https://www.passwordmeter.com/)

 Any other free online password strength checker

To check the strength of the password, first go to password meter



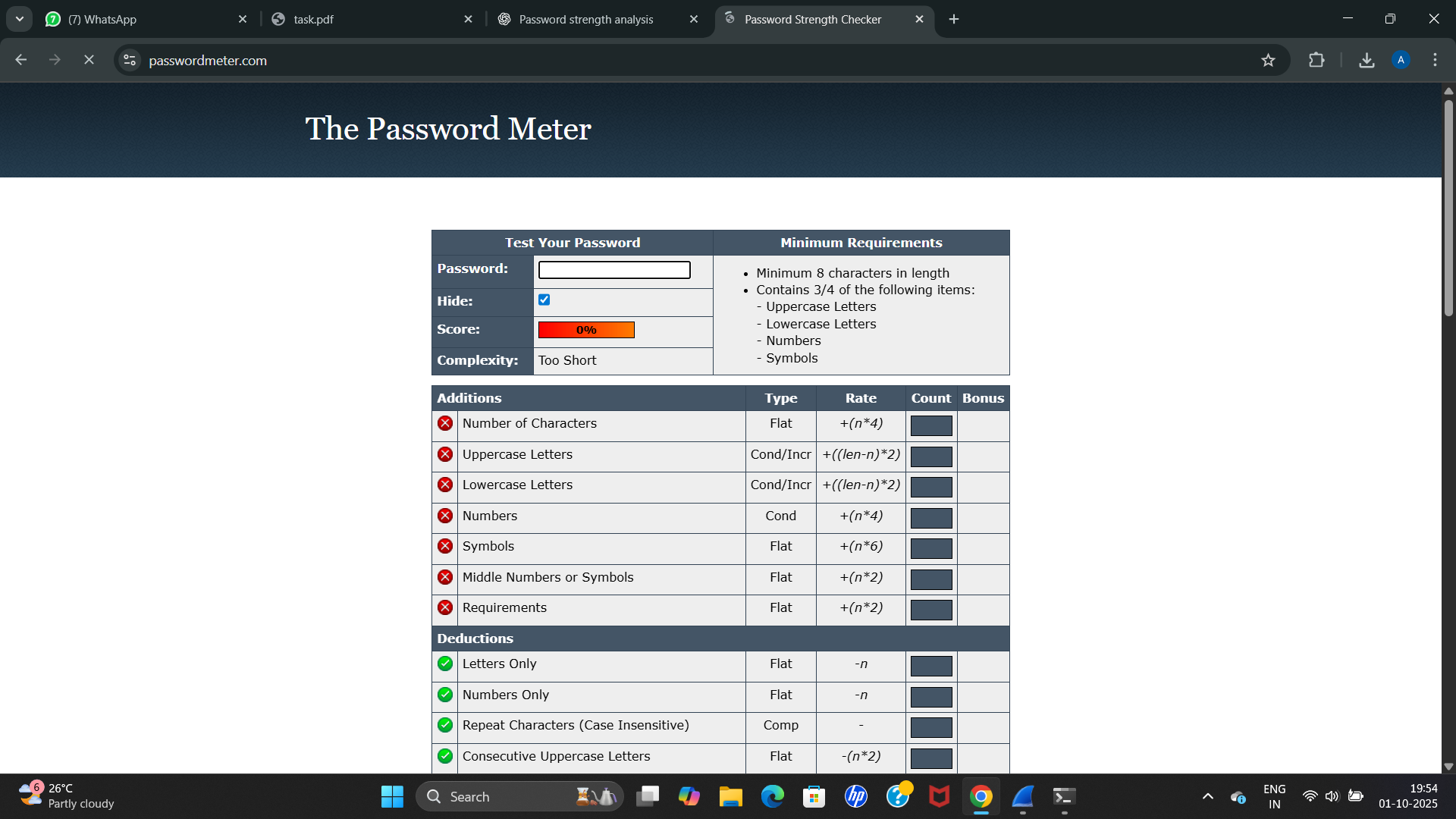
**1. Password Creation table**

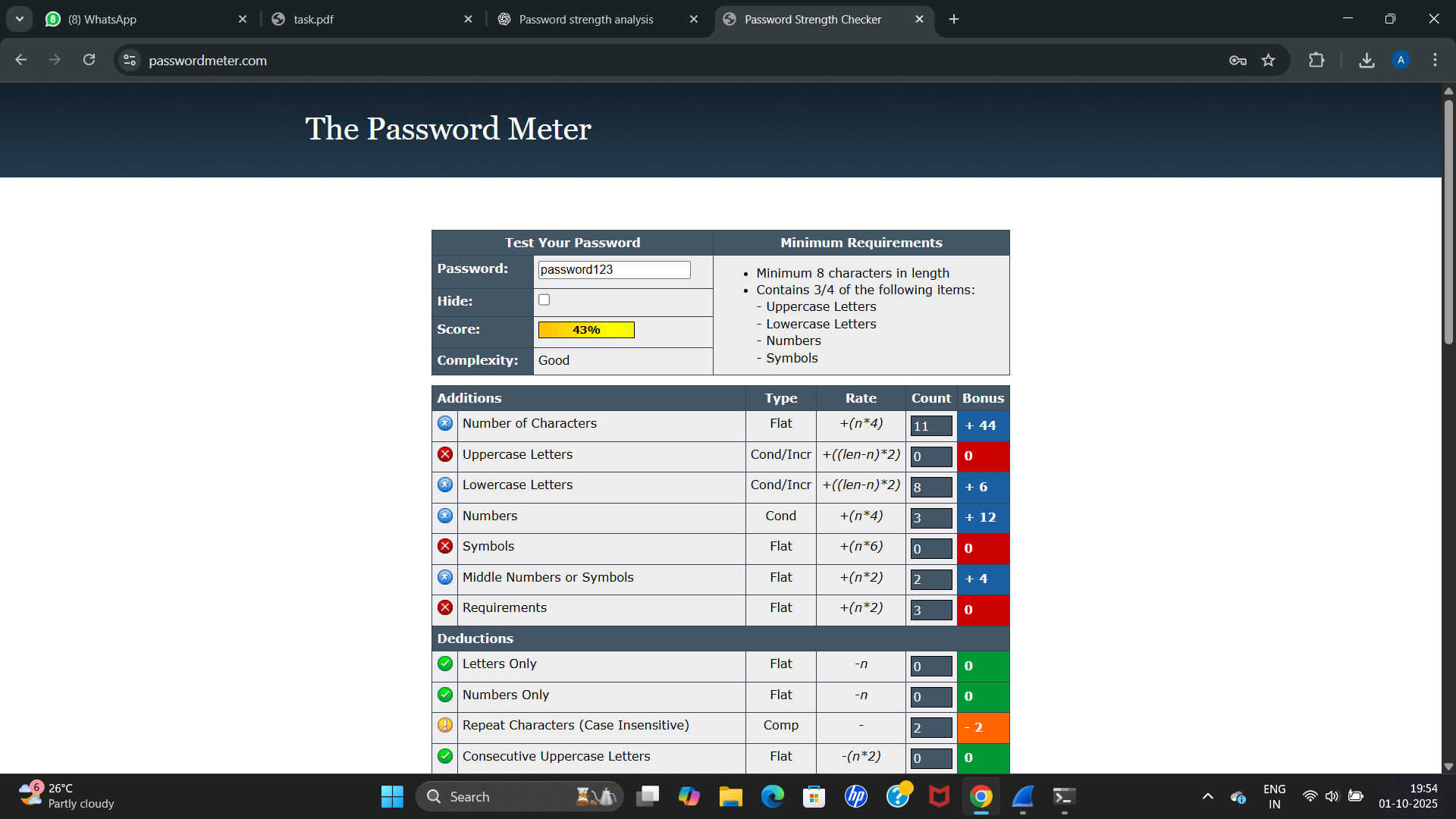
created multiple passwords with varying complexity:

| **Password** |  |  |
| --- | --- | --- |
| password123 |  |  |
| PaSs1234 |  |  |
| P@ssw0rd! |  |  |
| 7r0ubL3#Secur3! |  |  |
| 123456 |  |  |

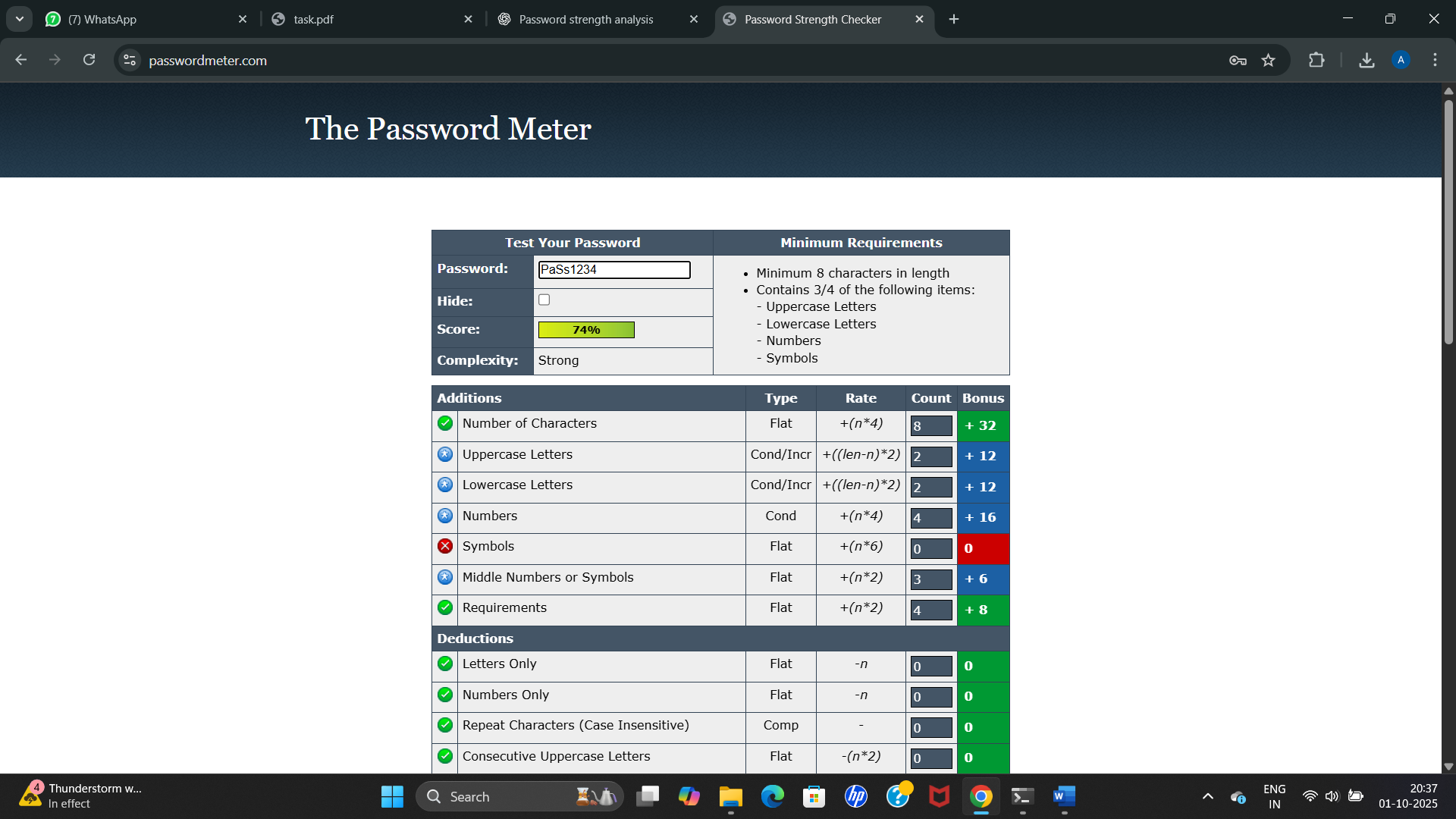
To check the each password’s strength we will type in the password,

First, lets use the first combination password123



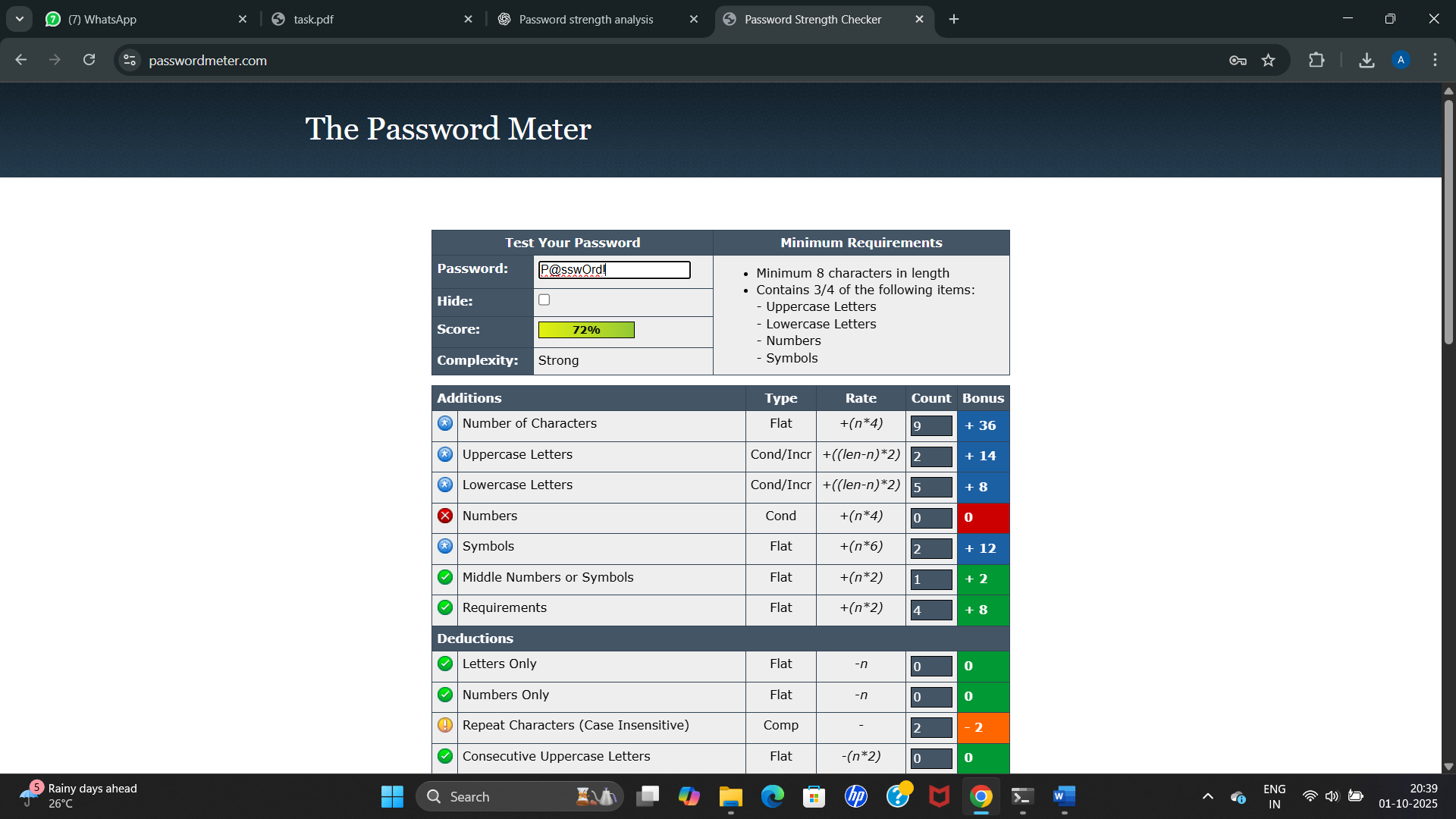


The second combination, PasS1234



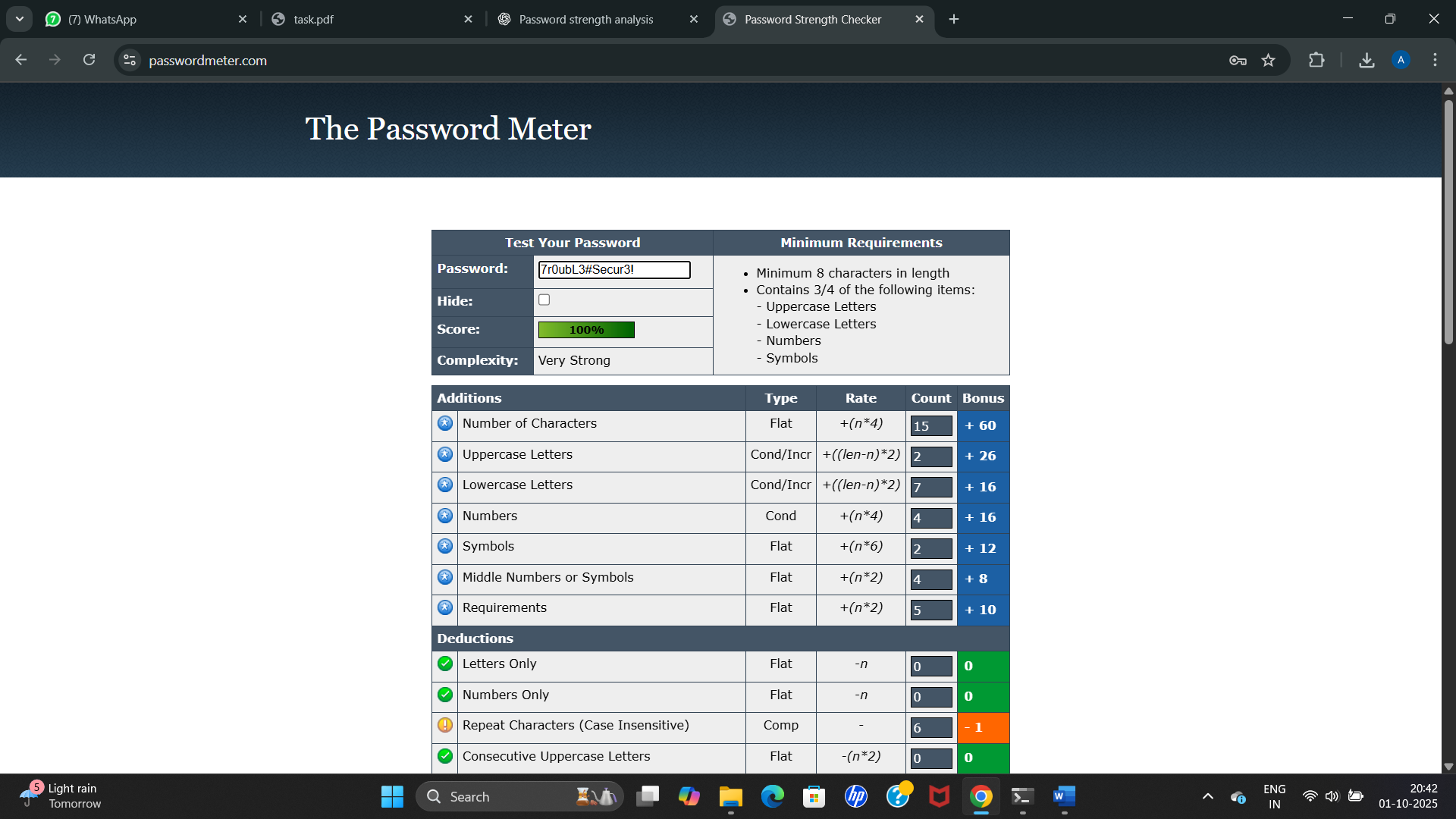
The complexity came as strong while the score was 74%

The third combination P@ssw0rd! is shown below



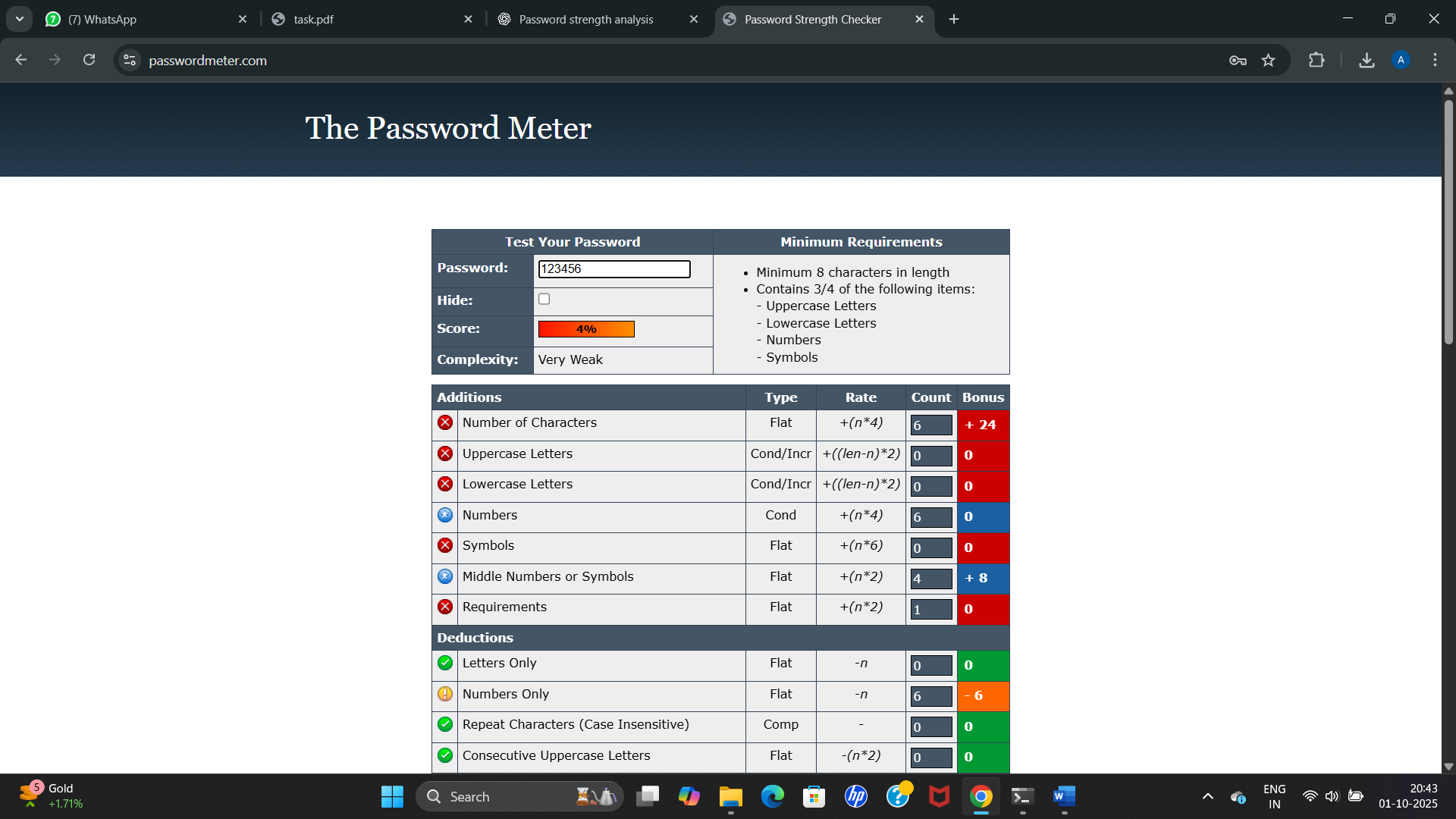
Complexity came as strong with a score of 72%

The fourth combination 7r0ubL3#Secur3! Is shown below



Complexity came as very strong with a score of 100%

Now the fifth combination 123456 is shown below



The result came as weak with score of just 4%

| **Password** | **Complexity** | **Score (%)** | **Feedback / Notes** |
| --- | --- | --- | --- |
| password123 | Good | 43% | Simple and predictable; easy to guess |
| PaSs1234 | Strong | 74% | Mixed case and numbers improve strength |
| P@ssw0rd! | Strong | 72% | Includes symbols and numbers; reasonably secure |
| 7r0ubL3#Secur3! | Very Strong | 100% | Long, complex, highly secure |
| 123456 | Too Weak | 4% | Extremely common; very easy to crack |

**4. Tips Learned from Evaluation (Detailed)**

1. **Password Length Matters:**  
   Longer passwords are significantly harder to crack. Passwords with 12 or more characters exponentially increase the number of possible combinations, making brute-force attacks much slower and less likely to succeed.
2. **Mix of Characters Increases Security:**  
   Using a combination of uppercase letters, lowercase letters, numbers, and symbols strengthens passwords. Each additional character type adds complexity, making it harder for attackers to guess using dictionary or brute-force methods.
3. **Avoid Common Words and Patterns:**  
   Passwords containing common words (like “password”), sequential numbers (like “1234”), or predictable patterns are very weak. These are the first things password-cracking tools try.
4. **Uniqueness is Key:**  
   Reusing passwords across multiple accounts is risky. If one account is compromised, all accounts with the same password become vulnerable. Each account should have a unique, strong password.
5. **Complexity Alone Isn’t Enough:**  
   Even a complex password can be weak if it’s short or predictable. Combining **length, complexity, and unpredictability** is essential for strong security.
6. **Passphrases Can Be Effective:**  
   Using multiple unrelated words to create a passphrase (e.g., BlueTiger!7Coffee$) can make passwords both strong and memorable. Passphrases are longer and harder to crack while easier for users to remember than random characters.
7. **Regular Updates Help:**  
   Updating passwords periodically, especially for sensitive accounts, reduces the risk of long-term exposure if credentials are leaked.
8. **Password Manager Use:**  
   Managing complex, unique passwords manually is difficult. Using a reputable password manager helps generate and store strong passwords safely.

**Best Practices for Creating Strong Passwords**

1. **Use a Mix of Character Types:**  
   Include **uppercase letters, lowercase letters, numbers, and symbols** in your passwords. The more variety, the harder it is to guess or crack.
2. **Make Passwords Long:**  
   Passwords should be **at least 12 characters long**. Longer passwords increase the number of possible combinations, making brute-force attacks more difficult.
3. **Avoid Common Words and Predictable Patterns:**  
   Do **not** use easily guessable words, phrases, or patterns like password, 123456, qwerty, or abcd. Hackers often try these first using dictionary attacks.
4. **Use Unique Passwords for Every Account:**  
   Never reuse passwords across multiple accounts. If one account is compromised, reused passwords put all other accounts at risk.
5. **Consider Passphrases:**  
   Use **multiple unrelated words** to create a passphrase, e.g., Sun$Cake7Rain!Tiger. Passphrases are **memorable yet secure**.
6. **Regularly Update Passwords:**  
   Change passwords periodically, especially for sensitive accounts, to reduce long-term exposure from potential leaks.
7. **Use a Password Manager:**  
   Password managers can **generate and securely store complex passwords**, reducing the risk of weak or reused passwords.
8. **Enable Multi-Factor Authentication (MFA):**  
   Whenever possible, use **MFA** (e.g., OTPs, authentication apps) to add an extra layer of security even if the password is compromised.

**Common Password Attacks**

**1. Brute‑Force Attack**

**What it is:**  
An attacker tries every possible combination of characters until the correct password is found.

**How it works:**

* Automations try all possible passwords (e.g., a, b, c … aa, ab, …) or all combinations up to a certain length.
* Complexity grows exponentially with length and character set (lowercase, uppercase, digits, symbols).

**Example:**  
A 6‑character lowercase password has 26⁶ ≈ 308 million possibilities. Add uppercase + digits and the space explodes.

**Defenses:**

* Use long passwords (≥12–16 chars).
* Enforce rate limits / lockouts after failed attempts.
* Use account lockout policies and CAPTCHA on repeated failures.
* Monitor and block suspicious IPs and use MFA.

**2. Dictionary Attack**

**What it is:**  
An attacker uses a large list (dictionary) of likely passwords: common words, leaked passwords, phrases, or variations.

**How it works:**

* Tries words from curated lists (e.g., “password,” “123456,” “qwerty”), and common substitutions (p@ssw0rd).
* Faster than full brute force because attacker only tests likely candidates.

**Example:**  
If the target uses Football2024!, a dictionary that includes “football” and common year additions can guess it quickly.

**Defenses:**

* Avoid common words or predictable substitutions.
* Add length and randomness (passphrases, uncommon combinations).
* Use password strength checks and block commonly used passwords.
* Enforce unique passwords across accounts and MFA.

**3. Credential Stuffing**

**What it is:**  
Attackers take username/password pairs leaked from one breach and try them on many other sites and services.

**How it works:**

* Uses automated tools to test large breached credential lists against target sites.
* Works because many users reuse passwords across services.

**Example:**  
If alice@gmail.com : Summer2020 leaked from one site, attackers try the same pair on banks, email, social networks.

**Defenses:**

* Encourage/require unique passwords (password managers help).
* Detect and block rapid, repeated logins from same IPs or credential combos.
* Use MFA (stops login even if password is correct).
* Implement login throttling, device fingerprinting, and anomaly detection.

**4. Rainbow Table Attack**

**What it is:**  
An attacker uses precomputed tables mapping plaintext passwords to hash values to reverse weakly protected password hashes quickly.

**How it works:**

* Instead of recomputing hashes, the attacker looks up the hash in a rainbow table to find the original password.
* Effective when passwords are hashed with fast, unsalted hash functions (e.g., plain MD5/SHA1) and when no unique salt is used.

**Example:**  
If a stored hash 5f4dcc3b5aa765d61d8327deb882cf99 corresponds to password in a table, an attacker can instantly reverse it.

**Defenses:**

* Use a unique **salt** per password (random data added before hashing).
* Use slow, memory-hard hashing algorithms designed for passwords: bcrypt, Argon2, scrypt, or PBKDF2 with many iterations.
* Use sufficiently long, random passwords so they won’t appear in precomputed tables.

**How Password Complexity Affects Security**

* **Length**: Single most effective factor — each added character multiplies the search space.
* **Character set variety**: Uppercase, lowercase, digits, and symbols multiply combinations per character.
* **Unpredictability**: Randomness or unrelated words prevents dictionary hits.
* **Uniqueness**: Prevents credential stuffing damage across accounts.
* **Storage practices (server‑side)**: Salting + slow hashing prevents rainbow tables and slows brute-force attacks on stolen hashes.

**Conclusion**

The evaluation of multiple passwords clearly demonstrates that **password strength depends on length, complexity, and unpredictability**. Simple or common passwords, even with numbers, are highly vulnerable to attacks like brute force, dictionary, credential stuffing, and rainbow tables.

Passwords that incorporate a **mix of uppercase and lowercase letters, numbers, symbols, and sufficient length** are significantly more secure. Using passphrases and unique passwords for each account further reduces the risk of compromise.

Additionally, the assessment highlights the importance of **best practices**, including avoiding common patterns, regularly updating passwords, and using multi-factor authentication. Strong passwords, when combined with proper storage techniques (like salting and hashing) and user awareness, provide robust protection against modern cyberattacks.

In summary, **password complexity directly improves security**, and adopting strong, unique passwords is a critical step in safeguarding personal and organizational data.