**Task 10 : Password Strength Analyzer**

**1. Objective**

The main objective of this task is to develop a **Password Strength Analyzer** using Python. The tool is designed to check and evaluate the strength of user-provided passwords based on a set of predefined security rules.  
This helps users understand the quality of their passwords and promotes good cybersecurity practices by encouraging the use of complex and hard-to-guess passwords.

**2. Description of the Task**

Passwords are the first line of defense against unauthorized access to accounts and systems. Weak passwords can easily be cracked through brute-force or dictionary attacks, leading to severe data breaches or identity theft.  
To prevent such issues, this task involves building a **Python-based password validation program** that can classify passwords as **Weak**, **Moderate**, or **Strong** depending on their compliance with security standards.

The program checks whether a password:

* Has a minimum required length (commonly 8 characters or more)
* Includes at least one uppercase letter
* Includes at least one lowercase letter
* Contains at least one digit (0–9)
* Includes at least one special character (e.g., @, #, $, %, &)

Based on these checks, the program provides feedback indicating how strong or weak the password is and suggests improvements if needed.

**3. Tools and Environment**

* **Programming Language:** Python
* **Editors:** VS Code / Jupyter Notebook / Google Colab
* **Deliverables:**
  + password\_checker.py (Python script)
  + Screenshots of input/output results

**4. Implementation Steps**

1. **Import Functions and Define Logic**  
   Use Python’s built-in string functions such as any(), isupper(), islower(), isdigit(), and len() to evaluate each rule.
2. **Password Input**  
   The user is prompted to enter a password for analysis.
3. **Validation**  
   Each rule is checked one by one. If any rule fails, the password is marked as weak or moderate.
4. **Output Result**  
   Based on the conditions satisfied, the program prints the password strength along with a message such as:
   * *Weak: Password too short or missing key elements.*
   * *Moderate: Add special characters or uppercase letters.*
   * *Strong: Great! Your password meets all security requirements.*
5. **Testing**  
   Run the program with different passwords to verify that it correctly identifies weak, moderate, and strong passwords.

**5. Sample Execution**

**Input:**

Enter your password to check strength: S@fep4ss123

**Output:**

Password Strength Result → Strong: Great!

Your password meets all security requirements.

**Example of Weak Password:**

Input: hello123

Output: Moderate: Try adding uppercase, lowercase, numbers, and special characters.

**6. Expected Deliverables**

* Python source file: password\_checker.py
* Screenshot of program execution (input and output)
* README file with project explanation