# Prodigy Infotech Internship

**Task NO_3**

**Password Complexity Checker**

Build a tool that assesses the strength of a password based on criteria such as length, presence of uppercase and lowercase letters, numbers, and special characters. Provide feedback to users on the password's strength.

**Prodigy Infotech Ltd**

Submitted by: Ahmad Ali

Submission date: October 05,2025

Task 3

## Password Complexity Checker Tool

- **Code**

```
def assess_password_strength(password):
    # Criteria weights
    length_criteria = len(password) >= 8
    upper_criteria = re.search(r'[A-Z]', password) is not None
    lower_criteria = re.search(r'[a-z]', password) is not None
    number_criteria = re.search(r'\d', password) is not None
    special_criteria = re.search(r'[^A-Za-z0-9]', password) is
        not None

    # Calculate score
    score = sum([length_criteria, upper_criteria, lower_criteria
        , number_criteria, special_criteria])

    # Provide feedback
    if score == 5:
        strength = "Excellent"
        feedback = "Your password is strong. Great job!"
    elif score == 4:
        strength = "Good"
```

```
    if score == 5:
        strength = "Excellent"
        feedback = "Your password is strong. Great job!"
    elif score == 4:
        strength = "Good"
        feedback = "Add more variety or increase length for even
            stronger security."
    elif score == 3:
        strength = "Fair"
        feedback = "Consider adding special characters and
            numbers."
    elif score == 2:
        strength = "Weak"
        feedback = "Add uppercase, numbers, and special
            characters. Increase the length."
    else:
        strength = "Very Weak"
        feedback = "Use at least 8 characters with a mix of
            upper/lowercase, numbers, and symbols."
```

Task 3

```
    details = {
        "Length >= 8": length_criteria,
        "Has Uppercase": upper_criteria,
        "Has Lowercase": lower_criteria,
        "Has Numbers": number_criteria,
        "Has Special Characters": special_criteria
    }

    return strength, feedback, details


# === Test the Tool ===
if __name__ == "__main__":
    password = input("Enter your password: ")
    strength, feedback, details = assess_password_strength
        (password)

    print("\n=== Password Strength Report ===")
    print(f"Strength: {strength}")
```

```
    print("\n=== Password Strength Report ===")
    print(f"Strength: {strength}")
    print(f"Feedback: {feedback}\n")

    print("Criteria Check:")
    for criterion, passed in details.items():
        print(f" - {criterion}: {'✅ Passed' if passed else '❌
            Failed'}")
```

Task 3

## Output/Result

```
if score == 5:
    strength = "Excellent"
    feedback = "Your password is strong. Great job!"
elif score == 4:
    strength = "Good"
    feedback = "Add more variety or increase length for even
        stronger security."
elif score == 3:
    strength = "Fair"
    feedback = "Consider adding special characters and
        numbers."
elif score == 2:
    strength = "Weak"
    feedback = "Add uppercase, numbers, and special
        characters. Increase the length."
else:
    strength = "Very Weak"
    feedback = "Use at least 8 characters with a mix of
        upper/lowercase, numbers, and symbols."
```

```
Enter your password: @EconO_p!u$

=== Password Strength Report ===
Strength: Excellent
Feedback: Your password is strong. Great job!

Criteria Check:
 - Length >= 8: ✅ Passed
 - Has Uppercase: ✅ Passed
 - Has Lowercase: ✅ Passed
 - Has Numbers: ✅ Passed
 - Has Special Characters: ✅ Passed

=== Code Execution Successful ===
```