Project 2:

Password Strength Checker :

A password strength checker is a tool or function designed to evaluate the security level of a given password. It checks various criteria such as length, the inclusion of different character types (uppercase, lowercase, digits, and special characters), and the overall complexity to determine whether the password is weak, moderate, or strong.

1. What is a Password Strength Checker?

A password strength checker is an algorithm or software application that evaluates the strength of passwords. It provides feedback on the password's robustness and suggests improvements if necessary. This tool helps users create stronger passwords, enhancing their security against unauthorized access.

2. Problem Statement

Weak passwords are a significant security risk, making it easy for attackers to gain unauthorized access to systems and personal accounts. Users often create simple passwords for convenience, which compromises security. Therefore, there is a need for a tool that can evaluate the strength of passwords and provide feedback to help users create more secure passwords.

3. Goal

The goal of a password strength checker is to:

- Evaluate the strength of a given password based on predefined criteria.
- Provide feedback on the strength level (e.g., weak, moderate,

strong).

- Encourage users to create stronger, more secure passwords.

4. Constraints

- Criteria Coverage: The checker must evaluate passwords based on various criteria, including length, character variety, and complexity.
- Performance: The checker should provide instant feedback without significant delays.
- User-Friendliness: The feedback provided should be clear and easy to understand.
- Security: The checker must not store or transmit passwords insecurely.

5. Solution Approach

To implement a password strength checker, the following steps should be taken:

1. Define Criteria: Establish the criteria for evaluating password strength (e.g., length, uppercase letters, lowercase letters, digits, special characters).
2. Calculate Strength: Implement a function that checks the password against each criterion and calculates a strength score.
3. Provide Feedback: Based on the calculated score, classify the password as weak, moderate, or strong and provide appropriate feedback.

6. **Algorithmic Solution Step 1: Define Criteria**

- Minimum length (e.g., 8 characters).
- Presence of lowercase letters.
- Presence of uppercase letters.
- Presence of digits.
- Presence of special characters.

Step 2: Calculate Strength

- Initialize a strength score to 0.
- Increment the score for each criterion met.

Step 3: Provide Feedback

- Classify the password as weak, moderate, or strong based on the strength score.
- Return the classification as feedback.

Python code:

```
import re

def check_password_strength(password): strength_score = 0

# Check the length of the password if len(password) >= 8:

strength_score += 1


# Check for lowercase letters if re.search("[a-z]", password):

strength_score += 1

# Check for uppercase letterif re.search("[A-Z]", password): strength_score += 1
```

```
# Check for digits
if re.search("[0-9]", password): strength_score += 1

# Check for special characters
if re.search("[!@#$%^&*(),.?\":{}|<>]", password): strength_score += 1

# Determine strength based on the score if strength_score <= 2:
return "Weak"
elif strength_score == 3 or strength_score == 4: return "Moderate"
else:
return "Strong"

# Test the function
password = input("Enter a password to check its strength: ") print(f"Password
strength: {check_password_strength(password)}")
```

Output:1

Enter a password to check its strength: P@ssw0rd!

Password strength: Strong

Screeenshots