**CODE:**

def assess\_password\_strength(password):

# Initialize criteria checks

length\_criteria = len(password) >= 8

uppercase\_criteria = any(c.isupper() for c in password)

lowercase\_criteria = any(c.islower() for c in password)

number\_criteria = any(c.isdigit() for c in password)

special\_char\_criteria = any(c in '!@#$%^&\*(),.?":{}|<>' for c in password)

# Calculate strength score

score = sum([length\_criteria, uppercase\_criteria, lowercase\_criteria, number\_criteria, special\_char\_criteria])

# Determine strength level

strength\_levels = {

5: 'Very Strong',

4: 'Strong',

3: 'Moderate',

2: 'Weak',

1: 'Very Weak',

0: 'Very Weak'

}

strength = strength\_levels[score]

# Feedback messages

feedback = []

if not length\_criteria:

feedback.append('Password should be at least 8 characters long.')

if not uppercase\_criteria:

feedback.append('Password should include at least one uppercase letter.')

if not lowercase\_criteria:

feedback.append('Password should include at least one lowercase letter.')

if not number\_criteria:

feedback.append('Password should include at least one number.')

if not special\_char\_criteria:

feedback.append('Password should include at least one special character.')

return strength, feedback

# Example usage

password = input("Enter your password: ")

strength, feedback = assess\_password\_strength(password)

print(f"Password strength: {strength}")

if feedback:

print("Feedback:")

for item in feedback:

print(f"- {item}")

**OUTPUT:**

**Enter your password: hsjahd/D32**

**Password strength: Strong**

**Feedback:**

**- Password should include at least one special character.**

**Enter your password: djcnhdk**

**Password strength: Very Weak**

**Feedback:**

**- Password should be at least 8 characters long.**

**- Password should include at least one uppercase letter.**

**- Password should include at least one number.**

**- Password should include at least one special character.**