THADOMAL SHAHANI ENGINEERING COLLEGE

DEPARTMENT OF INFORMATION TECHNOLOGY

Roll no:I-12 **11/4/25**

8. Pattern Matching: LO5

Aim:

Write a Python script that prompts the user to enter a password. Use regular expressions to validate the password based on these criteria: At least 8 characters long, Contains at least one uppercase letter, one lowercase letter, one digit, and one special character.

Theory:

- This program is built using Python to validate the strength of a user-entered password.
- It uses the re module (regular expressions) to perform pattern matching on the input string.
- The goal of the program is to ensure the password is strong and secure, meeting common security standards.

- The password must satisfy the following conditions:

1. At least 8 characters long
2. At least one uppercase letter (A-Z)
3. At least one lowercase letter (a-z)
4. At least one digit (0-9)
5. At least one special character (e.g., @, #, $, %, etc.)

- Each of these conditions is checked using re.search():

1. re.search(r"[A-Z]", password) checks for uppercase letters.
2. re.search(r"[a-z]", password) checks for lowercase letters.
3. re.search(r"\d", password) checks for numeric digits.
4. re.search(r"[!@#$%^&*(),.?\":{}|<>]", password) checks for special characters.

- If any condition is not met, the function returns a specific error message.
- If all conditions are satisfied, it returns “Password is valid.”
- The script then prompts the user for input and displays the validation result.

- This approach ensures strong password creation, which helps protect user accounts from being compromised.

The use of regular expressions makes the validation efficient, compact, and readable, ideal for real-world applications involving authentication and form validation.

Program :

```
import re


def validate_password(password):
    if len(password) < 8:
        return "Password must be at least 8 characters long."
    if not re.search(r"[A-Z]", password):
        return "Password must contain at least one uppercase letter."
    if not re.search(r"[a-z]", password):
        return "Password must contain at least one lowercase letter."
    if not re.search(r"\d", password):
        return "Password must contain at least one digit."
    if not re.search(r"[!@#$%^&*(),.?\":{}|<>]", password):
        return "Password must contain at least one special character."
    return "Password is valid."


user_password = input("Enter your password: ")
result = validate_password(user_password)
print(result)
```

Output:

```
Enter your password: star@lord123
Password must contain at least one uppercase letter.
PS D:\Assignments\Python Assignments> python -u "d:\Assignments\Python Assignments\Assignment 8\pattern.py"
Enter your password: StarLord@123
Password is valid.
PS D:\Assignments\Python Assignments> python -u "d:\Assignments\Python Assignments\Assignment 8\pattern.py"
Enter your password: 12345
Password must be at least 8 characters long.
PS D:\Assignments\Python Assignments> python -u "d:\Assignments\Python Assignments\Assignment 8\pattern.py"
Enter your password: Sohamsvk
Password must contain at least one digit.
PS D:\Assignments\Python Assignments> python -u "d:\Assignments\Python Assignments\Assignment 8\pattern.py"
Enter your password: 12345678
Password must contain at least one uppercase letter.
PS D:\Assignments\Python Assignments> python -u "d:\Assignments\Python Assignments\Assignment 8\pattern.py"
Enter your password: STARKINDUSTRIES
Password must contain at least one lowercase letter.
```

Conclusion:

The above Python program effectively validates a user's password using the re (regular expressions) module. It ensures the password meets essential security criteria: a minimum length of 8 characters, and at least one uppercase letter, one lowercase letter, one digit, and one special character.