A new project is allocated and the project name is "Password-Generator ",

You have to add some functionalities which are given below:

1. Generating a random password

• The Password must have the desired length (i.e. Contains 8-16 Characters).

• The Password might use Uppercase/Lowercase letters, Numbers, or Symbols to generate.

• The Randomly generated password is displayed on the console.

• Also, display the length of the password on the console.

2. Check Password Strength:

• The password is at least 8 characters long (8 is often the minimum required length for a decent password).

• The password is at most 16 characters long (16 is considered the maximum length for a good password).

• You have to display the categorize the password on the basis of strength (For Ex. Very weak, Weak, Medium, Strong, And Very Strong).

Hint: Depending on the length, random characters from the password alphabet are selected and combined to form a completely random password based on the user's requirements.

Submission:- Make a repository on GitHub named 'Internship', You have to upload all the projects into it. For this worksheet:- Make a new folder "Password-Generator project" in this repository, upload all the files there, and share the Github link with me through messages.

Submission Deadline: The deadline for the submission is 27-09-2023, at 11:59 PM.

Solution- Password-Generator

```
import java.security.SecureRandom;


public class PasswordGenerator {
   // Define character sets for uppercase letters, lowercase letters, numbers, and
   // symbols
   private static final String LOWERCASE_LETTERS = "abcdefghijklmnopqrstuvwxyz";
   private static final String UPPERCASE_LETTERS = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
   private static final String NUMBERS = "0123456789";
   private static final String SYMBOLS = "!@#$%^&*()-_=+[]{}|;:'\",.<>?";

   // Generate a random password of the desired length with given character sets
```

```
    public static String generatePassword(int length, boolean useLowercase, boolean useUppercase,
boolean useNumbers,
            boolean useSymbols) {
        SecureRandom random = new SecureRandom();
        StringBuilder password = new StringBuilder();

        // Create a character set based on user requirements
        String characters = "";
        if (useLowercase)
            characters += LOWERCASE_LETTERS;
        if (useUppercase)
            characters += UPPERCASE_LETTERS;
        if (useNumbers)
            characters += NUMBERS;
        if (useSymbols)
            characters += SYMBOLS;

        // Check if at least one character set is selected
        if (characters.isEmpty()) {
            throw new IllegalArgumentException("At least one character set must be selected.");
        }

        // Generate the random password
        for (int i = 0; i < length; i++) {
            int randomIndex = random.nextInt(characters.length());
            password.append(characters.charAt(randomIndex));
        }

        return password.toString();
    }
```

```
    // Check the strength of a password
    public static String checkPasswordStrength(String password) {
        int length = password.length();

        if (length < 8) {
            return "Very Weak";
        } else if (length <= 10) {
            return "Weak";
        } else if (length <= 12) {
            return "Medium";
        } else if (length <= 16) {
            return "Strong";
        } else {
            return "Very Strong";
        }
    }

    public static void main(String[] args) {
        int passwordLength = 12; // Define the desired password length

        // Set the character set preferences (true for use, false for not using)
        boolean useLowercase = true;
        boolean useUppercase = true;
        boolean useNumbers = true;
        boolean useSymbols = true;

        String generatedPassword = generatePassword(passwordLength, useLowercase, useUppercase,
useNumbers, useSymbols);

        System.out.println("Generated Password: " + generatedPassword);
        System.out.println("Password Length: " + generatedPassword.length());
```

```
        String strength = checkPasswordStrength(generatedPassword);
        System.out.println("Password Strength: " + strength);
    }
}
```