Assignment of Generating a random password

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.

Solution is here just below.

```
import java.util.Collections;

import java.util.List;

import java.util.stream.Collectors;

import org.apache.commons.text.RandomStringGenerator;

import static org.apache.commons.text.CharacterPredicates.DIGITS;

//create class GeneratePasswordExample2 to generate a random and secure password

public class GeneratePasswordExample2 {

   // main() method start

   public static void main(String args[]) {

      // call generateSecurePassword() method to generate random password using RandomStringGenerator

      String pass = generateSecurePassword();

      System.out.println("Password generated by RandomStringGenerator is:"+pass);

   }

   // create generateSecurePassword() method that find the secure 8 digit password and returns it to the
main() method

   public static String generateSecurePassword() {

      // generate a string that contains 2 special chars, 2 numbers, 2 uppercase and 2 lower case

      String demoPassword = generateRandomSpecialCharacters(2)
```

```
            .concat(generateRandomNumbers(2))
            .concat(generateRandomAlphabet(2, true))
            .concat(generateRandomAlphabet(2, false));
        // create a list of Char that stores all the characters, numbers and special characters
        List<Character> listOfChar = demoPassword.chars()
            .mapToObj(data -> (char) data)
            .collect(Collectors.toList());
        // use shuffle() method of the Collections to shuffle the list elements
        Collections.shuffle(listOfChar);
        //generate a random string(secure password) by using list stream() method and collect() method
        String password = listOfChar.stream()
    .collect(StringBuilder::new, StringBuilder::append, StringBuilder::append)
            .toString();
        // return RandomStringGenerator password to the main() method
        return password;
    }
    // create generateRandomSpecialCharacters() method that returns a string of special chars of the specified
length
    public static String generateRandomSpecialCharacters(int length) {
        // generate special string of specials chars by using Builder(), withinRange() and build() methods
        RandomStringGenerator generator = new RandomStringGenerator.Builder().withinRange(33, 45)
            .build();
        return generator.generate(length);
    }
    // create generateRandomNumbers() method that returns a string of numbers of the specified length
    public static String generateRandomNumbers(int length) {
        // generate special string of numbers by using Builder(), withinRange(), filteredBy() and build() methods
        RandomStringGenerator generator = new RandomStringGenerator.Builder()
            .withinRange('0', 'z')
            .filteredBy(DIGITS)
            .build();
        return generator.generate(length);
    }
```

```
    // create generateRandomAlphabet () method that returns either a upper case string or lower case string
    // of the specified length based on the boolean variable check
    public static String generateRandomAlphabet(int length, boolean check) {
        String password;
        // for lower case string
        if(check == true) {
            RandomStringGenerator generator = new RandomStringGenerator.Builder().withinRange('a', 'z')
                    .build();
            password = generator.generate(length);
        }
        // for upper case string
        else {
            RandomStringGenerator generator = new RandomStringGenerator.Builder().withinRange('A', 'Z')
                    .build();
            password = generator.generate(length);
        }0020
        return password;
    }
}
```