JAVA MINI-PROJECT

ON

PASSWORD GENERATOR

Submitted By

Shravani Gharat

Department of Computer Science & Engineering (Data Science)

SARASWATI COLLEGE OF ENGINEERING

Kharghar,Navi Mumbai

(Affiliated to University of Mumbai)

Academic Year :- 2025-26

INDEX

Sr.no	**CONTENT**	**Page no**
1.	Introduction	1
2.	Objectives	2
3.	Code	3-6
4.	Result	7-8
5.	Conclusion	9
6.	Reference	10

INTRODUCTION

The Password Generator is a software application designed to create secure and random passwords for users. By taking a user’s name as input and combining it with random letters, numbers, and symbols, the program generates strong passwords that are difficult to guess. This project is implemented using Java and can be run as a console-based program or a GUI-based application with a Submit button. It demonstrates core programming concepts like user input, randomization, string manipulation, and basic GUI development.

In the current digital age, cybersecurity is a major concern. Weak passwords can easily be hacked, leading to data theft and privacy breaches. This project is highly relevant because it helps users generate strong and unique passwords easily, thereby improving security. It also gives insight into how programming can solve practical, real-world problems.

This project was chosen because it is useful, practical, and beginner- friendly. It allows learners to combine programming logic with creativity and understand concepts like random number generation, string manipulation, and GUI design. Moreover, password security is a real- world issue, so creating a tool that helps users generate strong passwords adds value and practical significance to the project.

OBJECTIVES

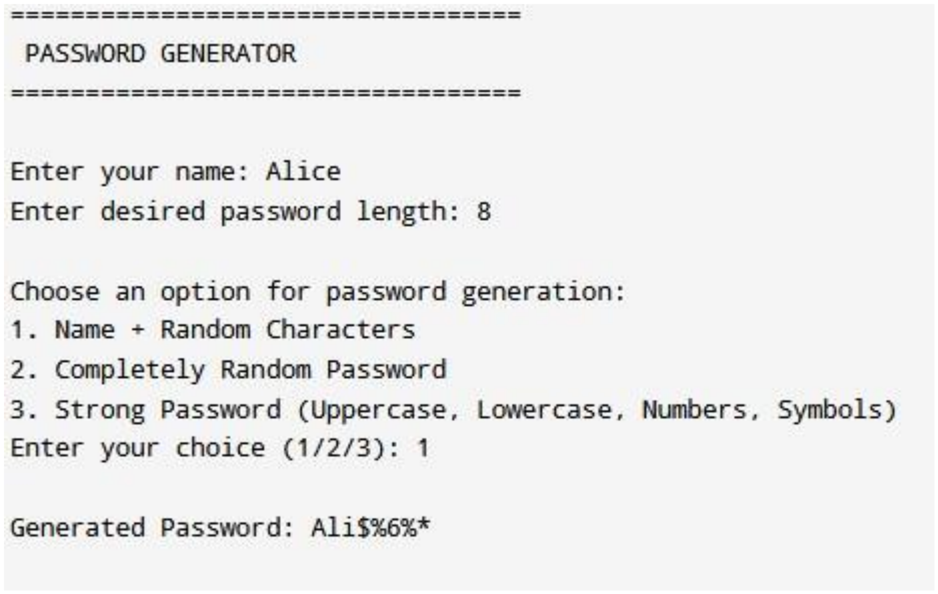
1. To create strong and secure passwords.
2. To simplify password creation for users.
3. To demonstrate programming concepts in Java.
4. 4. To develop a GUI-based application .
5. To promote cybersecurity awareness.

IMPLEMENTATION CODE

```
import java.util.Random; import
java.util.Scanner;

public class PasswordGenerator {

   // Option 1: Use part of name + random characters     public
static String generateWithName(String name, int length) {
        String characters = "0123456789!@#$%^&*()";
        Random random = new Random();
        StringBuilder password = new StringBuilder();

        // Add first 3 letters of name
        password.append(name.substring(0, Math.min(3, name.length())));

        // Fill remaining length with random characters
for (int i = password.length(); i < length; i++) {          int
index        =        random.nextInt(characters.length());
password.append(characters.charAt(index));
        }

        return password.toString();
   }

   // Option 2: Completely random password
public static String generateRandom(int length) {
        String characters =
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789!@#$%^&*()";
Random random = new Random();
```

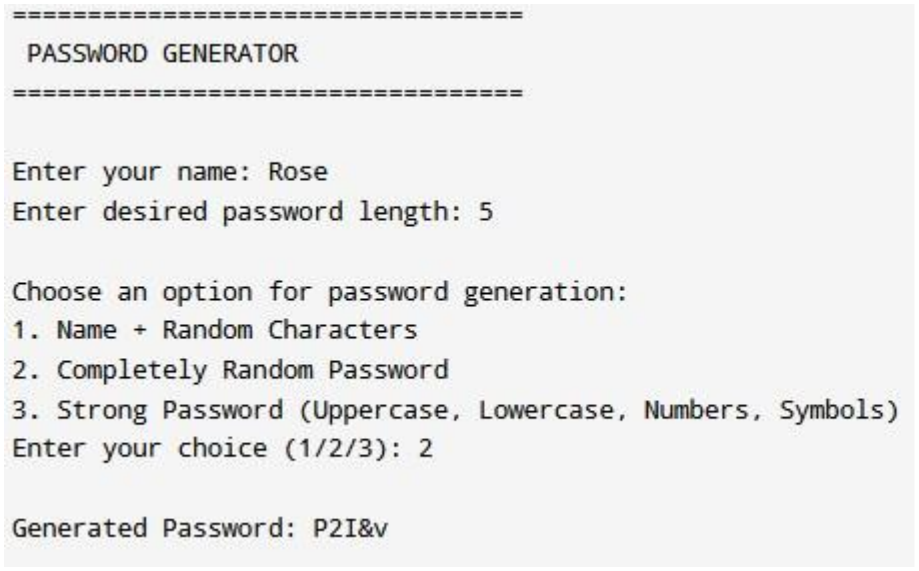
```
        StringBuilder password = new StringBuilder();

        for (int i = 0; i < length; i++) {             int index =
random.nextInt(characters.length());
password.append(characters.charAt(index));
        }
        return password.toString();
    }

    // Option 3: Strong password with all character types
public static String generateStrong(int length) {
        String upper = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
        String lower = "abcdefghijklmnopqrstuvwxyz";
        String digits = "0123456789";
        String special = "!@#$%^&*()";
        String all = upper + lower + digits + special;

        Random random = new Random();
        StringBuilder password = new StringBuilder();

        // Ensure at least one of each type
        password.append(upper.charAt(random.nextInt(upper.length())));
password.append(lower.charAt(random.nextInt(lower.length())));
password.append(digits.charAt(random.nextInt(digits.length())));
password.append(special.charAt(random.nextInt(special.length())));

        // Fill remaining length
        for (int i = password.length(); i < length; i++) {
password.append(all.charAt(random.nextInt(all.length())));
        }

        return password.toString();
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("===================================");
        System.out.println("        PASSWORD GENERATOR        ");
        System.out.println("===================================\n");

        System.out.print("Enter your name: ");
        String name = sc.nextLine();

        System.out.print("Enter desired password length: ");
int length = sc.nextInt();

        if (length < 4) {
            System.out.println("Password length should be at least 4.");
        } else {
            System.out.println("\nChoose an option for password generation:");
            System.out.println("1. Name + Random Characters");
            System.out.println("2. Completely Random Password");
            System.out.println("3. Strong Password (Uppercase, Lowercase, Numbers, Symbols)");
            System.out.print("Enter your choice (1/2/3): ");
            int choice = sc.nextInt();

            String password = "";

            switch (choice) {
                case 1:
                    password = generateWithName(name, length);
                    break;
case 2:
                    password = generateRandom(length);
                    break;
case 3:
                    password = generateStrong(length);
                    break;
default:
                    System.out.println("Invalid choice!");
                    return;
            }

            System.out.println("\nGenerated Password: " + password);
        }

        sc.close();
    }
}
```

RESULT



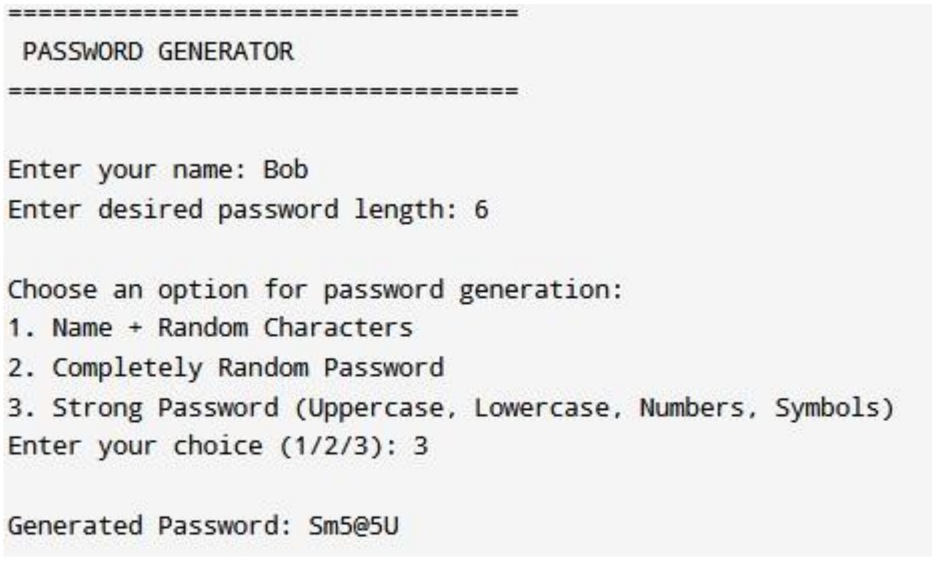
```
===================================
 PASSWORD GENERATOR
===================================

Enter your name: Alice
Enter desired password length: 8

Choose an option for password generation:
1. Name + Random Characters
2. Completely Random Password
3. Strong Password (Uppercase, Lowercase, Numbers, Symbols)
Enter your choice (1/2/3): 1

Generated Password: Ali$%6%*
```

Fig: 1 Random characters



```
===================================
 PASSWORD GENERATOR
===================================

Enter your name: Rose
Enter desired password length: 5

Choose an option for password generation:
1. Name + Random Characters
2. Completely Random Password
3. Strong Password (Uppercase, Lowercase, Numbers, Symbols)
Enter your choice (1/2/3): 2

Generated Password: P2I&v
```

Fig: 2 Complete random password



```
==================================
 PASSWORD GENERATOR
==================================

Enter your name: Bob
Enter desired password length: 6

Choose an option for password generation:
1. Name + Random Characters
2. Completely Random Password
3. Strong Password (Uppercase, Lowercase, Numbers, Symbols)
Enter your choice (1/2/3): 3

Generated Password: Sm5@5U
```

Fig: 3 Strong password

CONCLUSION

The Password Generator project demonstrates the practical application of programming to solve real-world problems, particularly in the area of cybersecurity. By generating strong, random passwords using a user's name and additional characters, the project provides a simple yet effective way to enhance online security.

Through this project, we learned how to implement user input handling, randomization, string manipulation, and conditional statements in Java. The project also highlights the importance of creating passwords that are difficult to guess, emphasizing the role of secure password practices in protecting personal and sensitive information.

Overall, this project is useful, easy to understand, and can be further enhanced with features like GUI interfaces, copy-to-clipboard functionality, or more complex password rules. It serves as a practical example of how programming can directly contribute to improving digital safety.

REFERENCES

- Websites:
  - Oracle Java Documentation: https://docs.oracle.com/en/java/
  - GeeksforGeeks – Java Random Class: https://www.geeksforgeeks.org/java-util-random-nextintmethod-java/
  - W3Schools Java Tutorial: https://www.w3schools.com/java/
- Online IDEs / Tutorials Used for Reference:
  - JDoodle Online Java Compiler: https://www.jdoodle.com/online-java-compiler
  - Programiz Java Tutorial: https://www.programiz.com/javaprogramming
- Tutorial/Article:
  - JournalDev – Java String and Random Class Tutorial: https://www.journaldev.com/