****

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

SUBMISSION REPORT OF JAVA PROJECT

**PASSWORD GENERATOR**

**Submitted By**

**Name Registration No. Roll no.**

S.Ashish Prakash 12105176 RK21QAA19

Debangshu Goswami 12105281 RK21QAB45

Nitish Kumar Dungdung 12105697 RK21QAA09

**Submitted to**

**DR. BHIMASEN MOHARANA**

UID: 28066

**Name Registration No. Role in project**

S.Ashish Prakash 12105176 Source code

Debangshu Goswami 12105281 Source code

Nitish Kumar Dungdung 12105697 Final report

**INTRODUCTION**

It’s a GUI-based project used with the swing library to organize all the elements that work under the **Password Generator Project**.

A password generator is a software tool or application that creates unique and secure passwords for users. These passwords are designed to be difficult to guess or crack, and are typically composed of a combination of letters, numbers, and symbols. Password generators are commonly used to protect sensitive information, such as bank accounts, email accounts, and social media profiles, from unauthorized access. By using a password generator, users can create complex passwords that are difficult to guess, making it less likely that their accounts will be compromised by hackers. Additionally, many password generators include features such as password strength meters and password managers, which can help users keep track of their passwords and ensure that they are using secure passwords across all of their accounts.

Java is also commonly used for desktop computing, other mobile computing, and numerical computing.

The popularity of Java is further evidenced with 90% of all Fortune 500 companies using Java.

**OBJECTIVE**

1.The objective of a password generator project is to create a software tool or application that can generate secure and unique passwords for users.

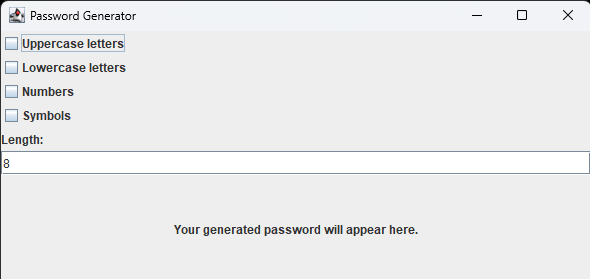
2.The primary goal is to enhance the security of user accounts by creating complex passwords that are difficult for hackers to guess or crack.

3.In addition, password generators may aim to provide users with features such as password strength meters, password managers, and password policies to further strengthen the security of their accounts.

4.The project may also focus on developing a user-friendly interface that allows users to easily generate and manage their passwords.

5. Overall, the objective of a password generator project is to help users protect their sensitive information by creating and managing strong and secure passwords.

**DESIGN**



The **Password Generator** is been designed as per the requirements, the design of the password generator window is user friendly as well as the design is easy to understand for any person.

In the centre of the window it contains different bullet options.

In the bottom of the window there is a message which shows the generated password.

The user has to use the different options to create the password like, if the user wants to generate the password using only Uppercase letters the user has to select the Uppercase option to generate the password.

Just below the options the user can select the length of the password.

**LIBRARY USED**

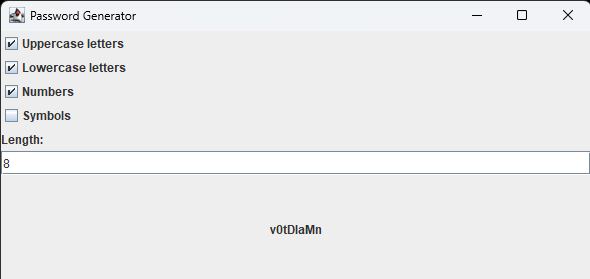
1. **UTIL**: - Package java. util. Contains the collections framework, some internationalization support classes, a service loader, properties, random number generation, string parsing and scanning classes, base64 encoding and decoding, a bit array, and several miscellaneous utility classes.
2. **SWING: -** Swing in Java is a lightweight GUI toolkit which has a wide variety of widgets for building optimized window based applications.

It is a part of the JFC (Java Foundation Classes). It is build on top of the AWT API and entirely written in java.

1. **AWT: -** AWT stands for Abstract window toolkit is an Application programming interface (API) for creating Graphical User Interface (GUI) in Java. It allows Java programmers to develop window-based applications. AWT provides various components like button, label, checkbox, etc. used as objects inside a Java Program.

**Result Screenshot**

* When the code is been run the window visible on the screen would be the same as the below screenshot.
* On the very top of the window (task bar) the name that is the title is been visible.



* In the above window we can see the different options to generate a password for example we can generate the password by only using Uppercase letters.
* And we can similarly use different type of options to generate the password.
* After selecting the options we can see a randomly password is generated in the bottom of the window.

**JAVA CODE USED**

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class PasswordGeneratorGUI extends JFrame implements ActionListener {

    private JLabel passwordLabel;

    private JButton generateButton;

    private JCheckBox uppercaseBox, lowercaseBox, numbersBox, symbolsBox;

    private JTextField lengthField;

    public PasswordGeneratorGUI() {

        super("Password Generator");

        // Create UI elements

        passwordLabel = new JLabel("Your generated password will appear here.");

        passwordLabel.setHorizontalAlignment(JLabel.CENTER);

        generateButton = new JButton("Generate");

        generateButton.addActionListener(this);

        uppercaseBox = new JCheckBox("Uppercase letters");

        lowercaseBox = new JCheckBox("Lowercase letters");

        numbersBox = new JCheckBox("Numbers");

        symbolsBox = new JCheckBox("Symbols");

        lengthField = new JTextField("8"); import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class PasswordGeneratorGUI extends JFrame implements ActionListener {

    private JLabel passwordLabel;

    private JButton generateButton;

    private JCheckBox uppercaseBox, lowercaseBox, numbersBox, symbolsBox;

    private JTextField lengthField;

    public PasswordGeneratorGUI() {

        super("Password Generator");

        // Create UI elements

        passwordLabel = new JLabel("Your generated password will appear here.");

        passwordLabel.setHorizontalAlignment(JLabel.CENTER);

        generateButton = new JButton("Generate");

        generateButton.addActionListener(this);

        uppercaseBox = new JCheckBox("Uppercase letters");

        lowercaseBox = new JCheckBox("Lowercase letters");

        numbersBox = new JCheckBox("Numbers");

        symbolsBox = new JCheckBox("Symbols");

        lengthField = new JTextField("8");

        // Create UI layout

        JPanel panel = new JPanel();

        panel.setLayout(new GridLayout(6, 1));

        panel.add(uppercaseBox);

        panel.add(lowercaseBox);

        panel.add(numbersBox);

        panel.add(symbolsBox);

        panel.add(new JLabel("Length:"));

        panel.add(lengthField);

        add(panel, BorderLayout.NORTH);

        add(passwordLabel, BorderLayout.CENTER);

        add(generateButton, BorderLayout.SOUTH);

        // Set window properties

        setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        setSize(300, 200);

        setLocationRelativeTo(null);

        setVisible(true);

    }

    public static void main(String[] args) {

        new PasswordGeneratorGUI();

    }

    public void actionPerformed(ActionEvent e) {

        if (e.getSource() == generateButton) {

            boolean useUppercase = uppercaseBox.isSelected();

            boolean useLowercase = lowercaseBox.isSelected();

            boolean useNumbers = numbersBox.isSelected();

            boolean useSymbols = symbolsBox.isSelected();

            int length = Integer.parseInt(lengthField.getText());

            // Generate password

            String password = generatePassword(useUppercase, useLowercase, useNumbers, useSymbols, length);

            passwordLabel.setText(password);

        }

    }

    private String generatePassword(boolean useUppercase, boolean useLowercase, boolean useNumbers, boolean useSymbols, int length) {

        String uppercaseChars = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

        String lowercaseChars = "abcdefghijklmnopqrstuvwxyz";

        String numberChars = "0123456789";

        String symbolChars = "!@#$%^&\*()\_+-=[]{}\\|;:'\",.<>/?";

        String allChars = "";

        if (useUppercase) allChars += uppercaseChars;

        if (useLowercase) allChars += lowercaseChars;

        if (useNumbers) allChars += numberChars;

        if (useSymbols) allChars += symbolChars;

        StringBuilder password = new StringBuilder();

        for (int i = 0; i < length; i++) {

            int index = (int) (Math.random() \* allChars.length());

            password.append(allChars.charAt(index));

        }

        return password.toString();

    }

}

        // Create UI layout

        JPanel panel = new JPanel();

        panel.setLayout(new GridLayout(6, 1));

        panel.add(uppercaseBox);

        panel.add(lowercaseBox);

        panel.add(numbersBox);

        panel.add(symbolsBox);

        panel.add(new JLabel("Length:"));

        panel.add(lengthField);

        add(panel, BorderLayout.NORTH);

        add(passwordLabel, BorderLayout.CENTER);

        add(generateButton, BorderLayout.SOUTH);

        // Set window properties

        setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        setSize(300, 200);

        setLocationRelativeTo(null);

        setVisible(true);

    }

    public static void main(String[] args) {

        new PasswordGeneratorGUI();

    }

    public void actionPerformed(ActionEvent e) {

        if (e.getSource() == generateButton) {

            boolean useUppercase = uppercaseBox.isSelected();

            boolean useLowercase = lowercaseBox.isSelected();

            boolean useNumbers = numbersBox.isSelected();

            boolean useSymbols = symbolsBox.isSelected();

            int length = Integer.parseInt(lengthField.getText());

            // Generate password

            String password = generatePassword(useUppercase, useLowercase, useNumbers, useSymbols, length);

            passwordLabel.setText(password);

        }

    }

    private String generatePassword(boolean useUppercase, boolean useLowercase, boolean useNumbers, boolean useSymbols, int length) {

        String uppercaseChars = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

        String lowercaseChars = "abcdefghijklmnopqrstuvwxyz";

        String numberChars = "0123456789";

        String symbolChars = "!@#$%^&\*()\_+-=[]{}\\|;:'\",.<>/?";

        String allChars = "";

        if (useUppercase) allChars += uppercaseChars;

        if (useLowercase) allChars += lowercaseChars;

        if (useNumbers) allChars += numberChars;

        if (useSymbols) allChars += symbolChars;

        StringBuilder password = new StringBuilder();

        for (int i = 0; i < length; i++) {

            int index = (int) (Math.random() \* allChars.length());

            password.append(allChars.charAt(index));

        }

        return password.toStr import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class PasswordGeneratorGUI extends JFrame implements ActionListener {

    private JLabel passwordLabel;

    private JButton generateButton;

    private JCheckBox uppercaseBox, lowercaseBox, numbersBox, symbolsBox;

    private JTextField lengthField;

    public PasswordGeneratorGUI() {

        super("Password Generator");

        // Create UI elements

        passwordLabel = new JLabel("Your generated password will appear here.");

        passwordLabel.setHorizontalAlignment(JLabel.CENTER);

        generateButton = new JButton("Generate");

        generateButton.addActionListener(this);

        uppercaseBox = new JCheckBox("Uppercase letters");

        lowercaseBox = new JCheckBox("Lowercase letters");

        numbersBox = new JCheckBox("Numbers");

        symbolsBox = new JCheckBox("Symbols");

        lengthField = new JTextField("8");

        // Create UI layout

        JPanel panel = new JPanel();

        panel.setLayout(new GridLayout(6, 1));

        panel.add(uppercaseBox);

        panel.add(lowercaseBox);

        panel.add(numbersBox);

        panel.add(symbolsBox);

        panel.add(new JLabel("Length:"));

        panel.add(lengthField);

        add(panel, BorderLayout.NORTH);

        add(passwordLabel, BorderLayout.CENTER);

        add(generateButton, BorderLayout.SOUTH);

        // Set window properties

        setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        setSize(300, 200);

        setLocationRelativeTo(null);

        setVisible(true);

    }

    public static void main(String[] args) {

        new PasswordGeneratorGUI();

    }

    public void actionPerformed(ActionEvent e) {

        if (e.getSource() == generateButton) {

            boolean useUppercase = uppercaseBox.isSelected();

            boolean useLowercase = lowercaseBox.isSelected();

            boolean useNumbers = numbersBox.isSelected();

            boolean useSymbols = symbolsBox.isSelected();

            int length = Integer.parseInt(lengthField.getText());

            // Generate password

            String password = generatePassword(useUppercase, useLowercase, useNumbers, useSymbols, length);

            passwordLabel.setText(password);

        }

    }

    private String generatePassword(boolean useUppercase, boolean useLowercase, boolean useNumbers, boolean useSymbols, int length) {

        String uppercaseChars = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

        String lowercaseChars = "abcdefghijklmnopqrstuvwxyz";

        String numberChars = "0123456789";

        String symbolChars = "!@#$%^&\*()\_+-=[]{}\\|;:'\",.<>/?";

        String allChars = "";

        if (useUppercase) allChars += uppercaseChars;

        if (useLowercase) allChars += lowercaseChars;

        if (useNumbers) allChars += numberChars;

        if (useSymbols) allChars += symbolChars;

        StringBuilder password = new StringBuilder();

        for (int i = 0; i < length; i++) {

            int index = (int) (Math.random() \* allChars.length());

            password.append(import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class PasswordGeneratorGUI extends JFrame implements ActionListener {

private JLabel passwordLabel;

private JButton generateButton;

private JCheckBox uppercaseBox, lowercaseBox, numbersBox, symbolsBox;

private JTextField lengthField;

public PasswordGeneratorGUI() {

super("Password Generator");

// Create UI elements

passwordLabel = new JLabel("Your generated password will appear here.");

passwordLabel.setHorizontalAlignment(JLabel.CENTER);

generateButton = new JButton("Generate");

generateButton.addActionListener(this);

uppercaseBox = new JCheckBox("Uppercase letters");

lowercaseBox = new JCheckBox("Lowercase letters");

numbersBox = new JCheckBox("Numbers");

symbolsBox = new JCheckBox("Symbols");

lengthField = new JTextField("8");

// Create UI layout

JPanel panel = new JPanel();

panel.setLayout(new GridLayout(6, 1));

panel.add(uppercaseBox);

panel.add(lowercaseBox);

panel.add(numbersBox);

panel.add(symbolsBox);

panel.add(new JLabel("Length:"));

panel.add(lengthField);

add(panel, BorderLayout.NORTH);

add(passwordLabel, BorderLayout.CENTER);

add(generateButton, BorderLayout.SOUTH);

// Set window properties

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setSize(300, 200);

setLocationRelativeTo(null);

setVisible(true);

}

public static void main(String[] args) {

new PasswordGeneratorGUI();

}

public void actionPerformed(ActionEvent e) {

if (e.getSource() == generateButton) {

boolean useUppercase = uppercaseBox.isSelected();

boolean useLowercase = lowercaseBox.isSelected();

boolean useNumbers = numbersBox.isSelected();

boolean useSymbols = symbolsBox.isSelected();

int length = Integer.parseInt(lengthField.getText());

// Generate password

String password = generatePassword(useUppercase, useLowercase, useNumbers, useSymbols, length);

passwordLabel.setText(password);

}

}

private String generatePassword(boolean useUppercase, boolean useLowercase, boolean useNumbers, boolean useSymbols, int length) {

String uppercaseChars = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

String lowercaseChars = "abcdefghijklmnopqrstuvwxyz";

String numberChars = "0123456789";

String symbolChars = "!@#$%^&\*()\_+-=[]{}\\|;:'\",.<>/?";

String allChars = "";

if (useUppercase) allChars += uppercaseChars;

if (useLowercase) allChars += lowercaseChars;

if (useNumbers) allChars += numberChars;

if (useSymbols) allChars += symbolChars;

StringBuilder password = new StringBuilder();

for (int i = 0; i < length; i++) {

int index = (int) (Math.random() \* allChars.length());

password.append(allChars.charAt(index));

}

return password.toString();

}

}

**CONCLUSION**

In conclusion, the Java password generator project is a useful tool for creating strong and secure passwords. It allows users to generate random passwords of varying lengths and complexity, and provides options for including different types of characters such as uppercase letters, lowercase letters, digits, and symbols. The project also includes error handling to prevent invalid inputs and ensure proper functionality. Overall, the password generator project is a valuable addition to any developer's toolkit and can be easily customized and integrated into other applications.