

# JAVA MINI PROJECT

Project Title

## **Strong Password Generator using JAVA**

### **Members:**

Aditya Rakshe	60004200094
Yogesh Jha	60004200123
Ayush Jain	60004200132
Preet Gada	60004200101

## Problem Statement

Java desktop application to generate random secure and strong password with lowercase & uppercase letters, symbols and numbers. This application let user to define any length for the password ranging between 8 and 25. It is built using Java Swing.

## Code:

```
package passwordgenerator;

import java.awt.Color;
import java.awt.Font;
import java.awt.Image;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JCheckBox;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JTextArea;
import javax.swing.SwingConstants;
import javax.swing.UIManager;

public class PasswordGenerator implements ActionListener{
    JFrame jframe;
    JComboBox passlen;
    JCheckBox lowercaselett,uppercaselett,numbers,symbols;
    JLabel plen;
    JButton btngen;
    JTextArea ta;
    String[] lengths;
    PasswordGenerator(){
        jframe = new JFrame("Nimbus Look and Feel");
        Color c = new Color(218, 247, 166);
        jframe.getContentPane().setBackground(c);
        Image icon = new
        ImageIcon(this.getClass().getResource("/icons/passwordGeneratorIcon.png")).getImage();
        lengths = new String[]{"8","9","10","11","12","13","14","15","16","17","18","19","20",
            "21","22","23","24","25"};
        plen = new JLabel("Select password length");
        plen.setFont(new Font(Font.SANS_SERIF,Font.PLAIN,13));
        plen.setBounds(90,40,150,20);
        passlen = new JComboBox(lengths);
        passlen.setBounds(230,40,50,20);
        lowercaselett = new JCheckBox("Include lowercase letters");
        lowercaselett.setBounds(110,70,170,30);
        lowercaselett.setBackground(c);
        uppercaselett = new JCheckBox("Include uppercase letters");
        uppercaselett.setBounds(110,100,170,30);
        uppercaselett.setBackground(c);
        numbers = new JCheckBox("Include numbers");
        numbers.setBounds(110,130,150,30);
        numbers.setBackground(c);
        symbols = new JCheckBox("Include symbols");
        symbols.setBounds(110,160,150,30);
    }
}
```

```

symbols.setBackground(c);
btnngen = new JButton("Generate");
btnngen.addActionListener(this);
btnngen.setBounds(130,195,100,30);
btnngen.setBackground(Color.RED);
btnngen.setForeground(Color.white);
ta = new JTextArea();
ta.setBounds(40,240,310,35);
ta.setAlignmentX(SwingConstants.CENTER);
ta.setEditable(false);
jframe.add(p1en);
jframe.add(passlen);
jframe.add(lowercaselett);
jframe.add(uppercaselett);
jframe.add(numbers);
jframe.add(symbols);
jframe.add(btnngen);
jframe.add(ta);
jframe.setLayout(null);
jframe.setTitle("Strong Password Generator");
jframe.setIconImage(icon);
jframe.setSize(400,360);
jframe.setVisible(true);
jframe.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
jframe.setResizable(false);
}

public String generateLCLetters(int len){
    String lc="";
    for(int i=1;i<=len;i++){
        double d = (Math.random()*(122-97))+97;    // ASCII values between 97 to 122
        int num = (int)d;
        String s = Character.toString((char)num);
        lc += s;
    }
    return lc;
}

public String generateUCLetters(int len){
    String uc="";
    for(int i=1;i<=len;i++){
        double d = (Math.random()*(91-65))+65;    // ASCII values between 65 to 91
        int num = (int)d;
        String s = Character.toString((char)num);
        uc += s;
    }
    return uc;
}

public String generateNumbers(int len){
    String numletters="";
    for(int i=1;i<=len;i++){
        double d = (Math.random()*(57-48))+48;    // ASCII values between 48 to 57
        int num = (int)d;

```

```

        String s = Character.toString((char)num);
        numletters += s;
    }
    return numletters;
}

public String generateSymbols(int len){
    String symbletters="";
    for(int i=1;i<=len;i++){
        double choice = (Math.random()*(5-1))+1;    // Random numbers between 1 to 4
        int choicenum = (int)choice;
        String symbolchoice="";
        double d=0;
        switch(choicenum){
            case 1:
                d = (Math.random()*(47-33))+33;    // ASCII values between 33 to 47
                break;
            case 2:
                d = (Math.random()*(64-58))+58;    // ASCII values between 58 to 64
                break;
            case 3:
                d = (Math.random()*(95-91))+91;    // ASCII values between 91 to 95
                break;
            case 4:
                d = (Math.random()*(126-123))+123;    // ASCII values between 123 to 126
                break;
            default:
                break;
        }
        int num = (int)d;
        String s = Character.toString((char)num);
        symbletters += s;
    }
    return symbletters;
}

public String lowUpSymNum(int len){
    String password = "";
    for(int i=0;i<len;i++){
        double d = (Math.random()*(5-1))+1;    // Random numbers between 1 to 4
        int num = (int)d;
        String choice = "";
        switch(num){
            case 1:
                choice = generateLCLetters(1);
                break;
            case 2:
                choice = generateUCLetters(1);
                break;
            case 3:
                choice = generateSymbols(1);
                break;
            case 4:
                choice = generateNumbers(1);
                break;
        }
    }
}

```

```

        default:
            break;
    }
    password += choice;
}
return password;
}

public String twoChoices(String code,int len){
    String password = "";
    for(int i=0;i<len;i++){
        double binar = (Math.random()*(2-0))+0;    // Random number 1 or 2
        double one = (Math.random()*(2-1))+1;    // Random number 1
        double two = (Math.random()*(3-2))+2;    // Random number 2
        double three = (Math.random()*(4-3))+3;    // Random number 3
        double four = (Math.random()*(5-4))+4;    // Random number 4
        double finalans=0;
        if(code.equals("LS") || code.equals("SL")){
            finalans = (int)binar==0?one:three;
        }
        // When lowercase letters checkbox & numbers checkbox are selected
        else if(code.equals("LN") || code.equals("NL")){
            finalans = (int)binar==0?one:four;
        }
        else if(code.equals("LU") || code.equals("UL")){
            finalans = (int)binar==0?one:two;
        }

        else if(code.equals("UN") || code.equals("NU")){
            finalans = (int)binar==0?two:four;
        }
        else if(code.equals("SU") || code.equals("US")){
            finalans = (int)binar==0?three:two;
        }
        else if(code.equals("NS") || code.equals("SN")){
            finalans = (int)binar==0?four:three;
        }
        int num = (int)finalans;
        String choice = "";
        switch(num){
            case 1:
                choice = generateLCLetters(1);
                break;
            case 2:
                choice = generateUCLetters(1);
                break;
            case 3:
                choice = generateSymbols(1);
                break;
            case 4:
                choice = generateNumbers(1);
                break;
            default:
                break;
        }
    }
}

```

```

    }
    password += choice;
}
return password;
}
public String threeChoices(String code,int len){
    String password = "";
    for(int i=0;i<len;i++){
        double trio = (Math.random()*(3-0))+0;
        double one = (Math.random()*(2-1))+1;
        double two = (Math.random()*(3-2))+2;
        double three = (Math.random()*(4-3))+3;
        double four = (Math.random()*(5-4))+4;
        double finalans=0;
        if(code.equals("LUN")|code.equals("ULN")|code.equals("NLU")){
            if((int)trio%2==0){
                finalans = (int)trio==0?one:two;
            } else{
                finalans = (int)trio==0?one:four;
            }
        }
        else if(code.equals("LUS")|code.equals("USL")|code.equals("SLU")){
            if((int)trio%2==0){
                finalans = (int)trio==0?one:two;
            } else{
                finalans = (int)trio==0?one:three;
            }
        }
        else if(code.equals("LNS")|code.equals("NLS")|code.equals("SNL")){
            if((int)trio%2==0){
                finalans = (int)trio==0?one:three;
            } else{
                finalans = (int)trio==0?one:four;
            }
        }
        else if(code.equals("UNS")|code.equals("SUN")|code.equals("NUS")){
            if((int)trio%2==0){
                finalans = (int)trio==0?two:three;
            } else{
                finalans = (int)trio==0?three:four;
            }
        }
        int num = (int)finalans;
        String choice = "";
        switch(num){
            case 1:
                choice = generateLCLetters(1);
                break;
            case 2:
                choice = generateUCLetters(1);
                break;
            case 3:
                choice = generateSymbols(1);

```

```

        break;
    case 4:
        choice = generateNumbers(1);
        break;
    default:
        break;
    }
    password += choice;
}
return password;
}

public static void main(String[] args) {
    // TODO code application logic here
    try {
        UIManager.setLookAndFeel("javax.swing.plaf.nimbus.NimbusLookAndFeel");    // Look and feel
    }
    catch (Exception e) {
        System.out.println("Look and Feel not set");
    }
    PasswordGenerator spgobj = new PasswordGenerator();
}

@Override
public void actionPerformed(ActionEvent e) {
    int len = passlen.getSelectedIndex();
    int passwordlen = Integer.parseInt(lengths[len]);
    String ans="";
    if(symbols.isSelected() && !lowercaselett.isSelected() && !uppercaselett.isSelected() &&
!numbers.isSelected()){
        ans = generateSymbols(passwordlen);
    }
    else if(lowercaselett.isSelected() && !uppercaselett.isSelected() && !numbers.isSelected() &&
!symbols.isSelected()){
        ans = generateLCLetters(passwordlen);
    }
    else if(uppercaselett.isSelected() && !lowercaselett.isSelected() && !numbers.isSelected() &&
!symbols.isSelected()){
        ans = generateUCLetters(passwordlen);
    }
    else if(numbers.isSelected() && !lowercaselett.isSelected() && !uppercaselett.isSelected() &&
!symbols.isSelected()){
        ans = generateNumbers(passwordlen);
    }
    else if(lowercaselett.isSelected() && uppercaselett.isSelected() && numbers.isSelected() &&
symbols.isSelected()){
        ans = lowUpSymNum(passwordlen);
    }
    else if(lowercaselett.isSelected() && uppercaselett.isSelected() && !numbers.isSelected() &&
!symbols.isSelected()){
        ans = twoChoices("LU",passwordlen);
    }
}

```

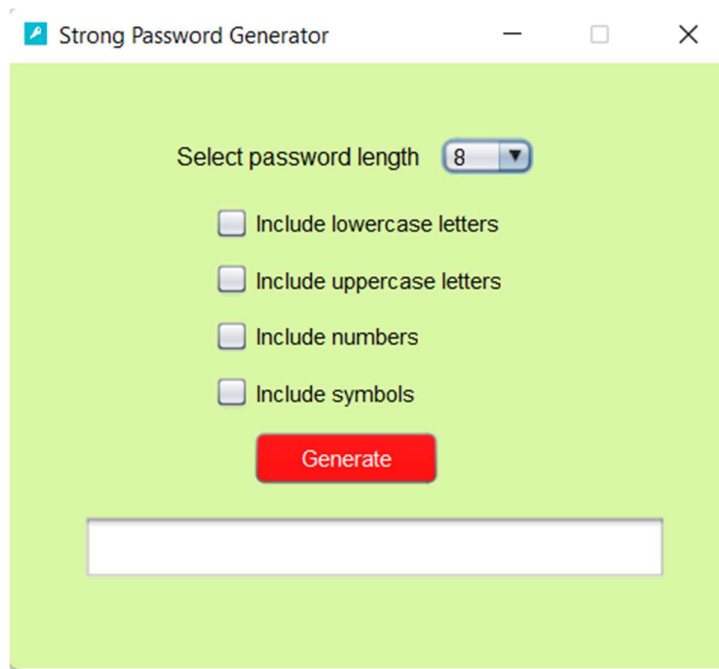


```

        else if(lowercaselett.isSelected() && !uppercaselett.isSelected() && !numbers.isSelected() &&
symbols.isSelected()){
            ans = twoChoices("LS",passwordlen);
        }
        else if(lowercaselett.isSelected() && !uppercaselett.isSelected() && numbers.isSelected() &&
!symbols.isSelected()){
            ans = twoChoices("LN",passwordlen);
        }
        else if(!lowercaselett.isSelected() && uppercaselett.isSelected() && numbers.isSelected() &&
!symbols.isSelected()){
            ans = twoChoices("UN",passwordlen);
        }
        else if(!lowercaselett.isSelected() && uppercaselett.isSelected() && !numbers.isSelected() &&
symbols.isSelected()){
            ans = twoChoices("SU",passwordlen);
        }
        else if(!lowercaselett.isSelected() && !uppercaselett.isSelected() && numbers.isSelected() &&
symbols.isSelected()){
            ans = twoChoices("NS",passwordlen);
        }
        else if(lowercaselett.isSelected() && uppercaselett.isSelected() && numbers.isSelected() &&
!symbols.isSelected()){
            ans = threeChoices("LUN",passwordlen);
        }
        else if(lowercaselett.isSelected() && uppercaselett.isSelected() && !numbers.isSelected() &&
symbols.isSelected()){
            ans = threeChoices("LUS",passwordlen);
        }
        else if(lowercaselett.isSelected() && !uppercaselett.isSelected() && numbers.isSelected() &&
symbols.isSelected()){
            ans = threeChoices("LNS",passwordlen);
        }
        else if(!lowercaselett.isSelected() && uppercaselett.isSelected() && numbers.isSelected() &&
symbols.isSelected()){
            ans = threeChoices("UNS",passwordlen);
        }
        ta.setFont(new Font("Serif",Font.PLAIN,20));
        ta.setText(ans);
    }
}

```

## Output:



Strong Password Generator

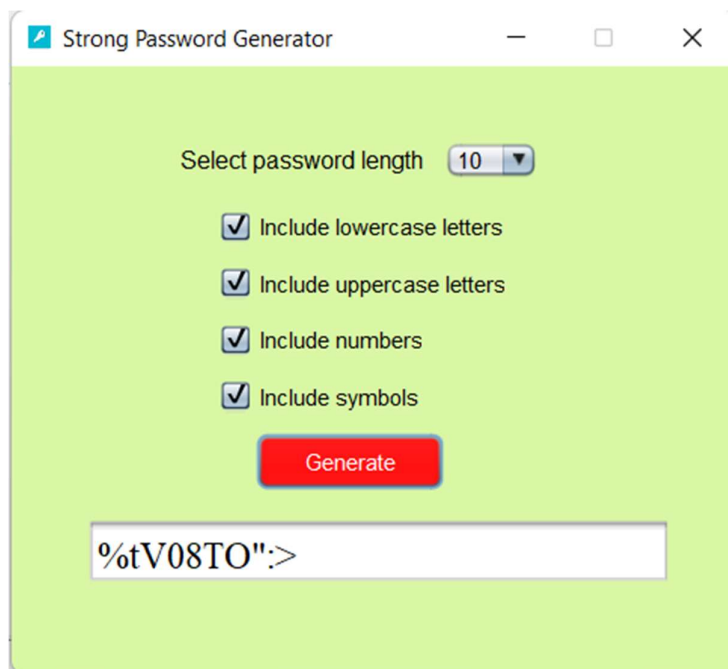
Select password length

☐ Include lowercase letters

☐ Include uppercase letters

☐ Include numbers

☐ Include symbols



Strong Password Generator

Select password length

☒ Include lowercase letters

☒ Include uppercase letters

☒ Include numbers

☒ Include symbols

## Conclusion:

We have learnt and made a working java project to demonstrate how to generate a Strong Password Generator.

Where in we have used Swings and awt to perform certain actions and creating the JFrame.

Also using random function to generate a random password.