**2016103604**

**2016103019**

**Varadharajan R**

**Dhana Pradeep**

**Introduction**

The main idea beyond developing this project is to create a user friendly application that will help any course instructor taking the daily attendance of the students very easily.

**Front End Source Code Java☺ :**

package mywork;

import java.awt.\*;

import java.awt.event.\*;

import java.util.ArrayList;

import java.util.List;

import javax.swing.\*;

import javax.sound.sampled.AudioInputStream;

import javax.sound.sampled.AudioSystem;

import javax.sound.sampled.Clip;

import javax.sound.sampled.LineUnavailableException;

import javax.sound.sampled.UnsupportedAudioFileException;

import java.io.\*;

import java.net.MalformedURLException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.logging.Level;

import java.util.logging.Logger;

class Retrieve {

/\*\*

\* @param args the command line arguments

\*/

void runn(String uname) {

EventQueue.invokeLater(new Runnable() {

@Override

public void run() {

try {

UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());

} catch (ClassNotFoundException | InstantiationException | IllegalAccessException | UnsupportedLookAndFeelException ex) {

ex.printStackTrace();

}

try{

//step1 load the driver class

Class.forName("oracle.jdbc.driver.OracleDriver");

//step2 create the connection object

Connection con=DriverManager.getConnection(

"jdbc:oracle:thin:@localhost:1521:XE","system","batman");

//step3 create the statement object

Statement stmt=con.createStatement();

ResultSet rs=stmt.executeQuery("select sum(s1),sum(s2),sum(s3),sum(s4),sum(s5),sum(s6),sum(s7),sum(s8),sum(s9),sum(s10),sum(s11),sum(s12),sum(s13),sum(s14),sum(s15),sum(s16),sum(s17),sum(s18),sum(s19),sum(s20),sum(s21),sum(s22),sum(s23),sum(s24),sum(s25),sum(s26),sum(s27),sum(s28),sum(s29),sum(s30),sum(s31),sum(s32),sum(s33),sum(s34),sum(s35),sum(s36),sum(s37),sum(s38),sum(s39),sum(s40),sum(s41),sum(s42),sum(s43),sum(s44),sum(s45),sum(S46),sum(s47),sum(s48),sum(S49),sum(s50),sum(s51),sum(s52),sum(s53),sum(s54),sum(s55),sum(s56),sum(s57),sum(s58),sum(s59),sum(s60),count(s1) from attendance where staffid='"+uname+"'");

int[] b = new int[100];

int[] per = new int[100];

while(rs.next())

for(int i=1;i<=61;i++)

{

b[i]=rs.getInt(i);

}

for(int i=1;i<61;i++)

{

per[i]=(b[i]\*100)/b[61];

// System.out.println(per[i]);

}

// TODO code application logic here

JFrame f= new JFrame("Label Example");

f.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

f.setLayout(new GridLayout(0, 10));

JLabel l1,l2,l3,l4,l5,l6,l7,l8,l9,l10,l11,l12,l13,l14,l15,l16,l17,l18,l19,l20,l21,l22,l23,l24,l25,l26,l27,l28,l29,l30,l31,l32,l33,l34,l35,l36,l37,l38,l39,l40,l41,l42,l43,l44,l45,l46,l47,l48,l49,l50,l51,l52,l53,l54,l55,l56,l57,l58,l59,l60;

l1=new JLabel("S1");

l1.setBounds(50,20, 100,30);

l2=new JLabel("S2");

l2.setBounds(50,60, 100,30);

l3=new JLabel("S3");

l3.setBounds(50,100, 100,30);

l4=new JLabel("S4");

l4.setBounds(50,140, 100,30);

l5=new JLabel("S5");

l5.setBounds(50,180, 100,30);

l6=new JLabel("S6");

l6.setBounds(50,220, 100,30);

l7=new JLabel("S7");

l7.setBounds(50,260, 100,30);

l8=new JLabel("S8");

l8.setBounds(50,300, 100,30);

l9=new JLabel("S9");

l9.setBounds(50,340, 100,30);

l10=new JLabel("S10");

l10.setBounds(50,380, 100,30);

l11=new JLabel("S11");

l11.setBounds(50,420, 100,30);

l12=new JLabel("S12");

l12.setBounds(50,460, 100,30);

l13=new JLabel("S13");

l13.setBounds(50,500, 100,30);

l14=new JLabel("S14");

l14.setBounds(50,540, 100,30);

l15=new JLabel("S15");

l15.setBounds(50,580, 100,30);

l16=new JLabel("S16");

l16.setBounds(50,620, 100,30);

l17=new JLabel("S17");

l17.setBounds(50,660, 100,30);

l18=new JLabel("S18");

l18.setBounds(50,700, 100,30);

l19=new JLabel("S19");

l19.setBounds(50,740, 100,30);

l20=new JLabel("S20");

l20.setBounds(50,780, 100,30);

l21=new JLabel("S21");

l21.setBounds(50,820, 100,30);

l22=new JLabel("S22");

l22.setBounds(50,860, 100,30);

l23=new JLabel("S23");

l23.setBounds(50,900, 100,30);

l24=new JLabel("S24");

l24.setBounds(50,940, 100,30);

l25=new JLabel("S25");

l25.setBounds(50,980, 100,30);

l26=new JLabel("S26");

l26.setBounds(50,1020, 100,30);

l27=new JLabel("S27");

l27.setBounds(50,1060, 100,30);

l28=new JLabel("S28");

l28.setBounds(50,1100, 100,30);

l29=new JLabel("S29");

l29.setBounds(50,1140, 100,30);

l30=new JLabel("S30");

l30.setBounds(50,1180, 100,30);

l31=new JLabel("S31");

l31.setBounds(50,1220, 100,30);

l32=new JLabel("S32");

l32.setBounds(50,1260, 100,30);

l33=new JLabel("S33");

l33.setBounds(50,1300, 100,30);

l34=new JLabel("S34");

l34.setBounds(50,1340, 100,30);

l35=new JLabel("S35");

l35.setBounds(50,1380, 100,30);

l36=new JLabel("S36");

l36.setBounds(50,1420, 100,30);

l37=new JLabel("S37");

l37.setBounds(50,1460, 100,30);

l38=new JLabel("S38");

l38.setBounds(50,1500, 100,30);

l39=new JLabel("S39");

l39.setBounds(50,1540, 100,30);

l40=new JLabel("S40");

l40.setBounds(50,1580, 100,30);

l41=new JLabel("S41");

l41.setBounds(50,1620, 100,30);

l42=new JLabel("S42");

l42.setBounds(50,1660, 100,30);

l43=new JLabel("S43");

l43.setBounds(50,1700, 100,30);

l44=new JLabel("S44");

l44.setBounds(50,1740, 100,30);

l45=new JLabel("S45");

l45.setBounds(50,1780, 100,30);

l46=new JLabel("S46");

l46.setBounds(50,1820, 100,30);

l47=new JLabel("S47");

l47.setBounds(50,1860, 100,30);

l48=new JLabel("S48");

l48.setBounds(50,1900, 100,30);

l49=new JLabel("S49");

l49.setBounds(50,1940, 100,30);

l50=new JLabel("S50");

l50.setBounds(50,1980, 100,30);

l51=new JLabel("S51");

l51.setBounds(50,2020, 100,30);

l52=new JLabel("S52");

l52.setBounds(50,2060, 100,30);

l53=new JLabel("S53");

l53.setBounds(50,2100, 100,30);

l54=new JLabel("S54");

l54.setBounds(50,2140, 100,30);

l55=new JLabel("S55");

l55.setBounds(50,2180, 100,30);

l56=new JLabel("S56");

l56.setBounds(50,2220, 100,30);

l57=new JLabel("S57");

l57.setBounds(50,2240, 100,30);

l58=new JLabel("S58");

l58.setBounds(50,2280, 100,30);

l59=new JLabel("S59");

l59.setBounds(50,2320, 100,30);

l60=new JLabel("S60");

l60.setBounds(50,2360, 100,30);

f.add(l1); f.add(l2);f.add(l3);f.add(l4);f.add(l5);

f.add(l6);f.add(l7);f.add(l8);f.add(l9);f.add(l10);

f.add(l11);f.add(l12);f.add(l13);f.add(l14);f.add(l15);

f.add(l16);f.add(l17);f.add(l18);f.add(l19);f.add(l20);

f.add(l21);f.add(l22);f.add(l23);f.add(l24);f.add(l25);

f.add(l26);f.add(l27);f.add(l28);f.add(l29);f.add(l30);

f.add(l31);f.add(l32);f.add(l33);f.add(l34);f.add(l35);

f.add(l36);f.add(l37);f.add(l38);f.add(l39);f.add(l40);

f.add(l41);f.add(l42);f.add(l43);f.add(l44);f.add(l45);

f.add(l46);f.add(l47);f.add(l48);f.add(l49);f.add(l50);

f.add(l51);f.add(l52);f.add(l53);f.add(l54);f.add(l55);

f.add(l56);f.add(l57);f.add(l58);f.add(l59);f.add(l60);

JTextField t1,t2,t3,t4,t5,t6,t7,t8,t9,t10,t11,t12,t13,t14,t15,t16,t17,t18,t19,t20,t21,t22,t23,t24,t25,t26,t27,t28,t29,t30,t31,t32,t33,t34,t35,t36,t37,t38,t39,t40,t41,t42,t43,t44,t45,t46,t47,t48,t49,t50,t51,t52,t53,t54,t55,t56,t57,t58,t59,t60;

t1=new JTextField(per[1]+"%");

t1.setBounds(100,20, 100,30);

t2=new JTextField(per[2]+"%");

t2.setBounds(100,60, 100,30);

t3=new JTextField(per[3]+"%");

t3.setBounds(100,100, 100,30);

t4=new JTextField(per[4]+"%");

t4.setBounds(100,140, 100,30);

t5=new JTextField(per[5]+"%");

t5.setBounds(100,140, 100,30);

t6=new JTextField(per[6]+"%");

t6.setBounds(100,180, 100,30);

t7=new JTextField(per[7]+"%");

t7.setBounds(100,220, 100,30);

t8=new JTextField(per[8]+"%");

t8.setBounds(100,260, 100,30);

t9=new JTextField(per[9]+"%");

t9.setBounds(100,260, 100,30);

t10=new JTextField(per[10]+"%");

t10.setBounds(100,300, 100,30);

t11=new JTextField(per[11]+"%");

t11.setBounds(100,340, 100,30);

t12=new JTextField(per[12]+"%");

t12.setBounds(100,380, 100,30);

t13=new JTextField(per[13]+"%");

t13.setBounds(100,420,100,30);

t14=new JTextField(per[14]+"%");

t14.setBounds(100,460, 200,30);

t15=new JTextField(per[15]+"%");

t15.setBounds(50,150, 200,30);

t16=new JTextField(per[16]+"%");

t16.setBounds(50,150, 200,30);

t17=new JTextField(per[17]+"%");

t17.setBounds(50,150, 200,30);

t18=new JTextField(per[18]+"%");

t18.setBounds(50,150, 200,30);

t19=new JTextField(per[19]+"%");

t19.setBounds(50,150, 200,30);

t20=new JTextField(per[20]+"%");

t20.setBounds(50,150, 200,30);

t21=new JTextField(per[21]+"%");

t21.setBounds(50,150, 200,30);

t22=new JTextField(per[22]+"%");

t22.setBounds(50,150, 200,30);

t23=new JTextField(per[23]+"%");

t23.setBounds(50,150, 200,30);

t24=new JTextField(per[24]+"%");

t24.setBounds(50,150, 200,30);

t25=new JTextField(per[25]+"%");

t25.setBounds(50,150, 200,30);

t26=new JTextField(per[26]+"%");

t26.setBounds(50,150, 200,30);

t27=new JTextField(per[27]+"%");

t27.setBounds(50,150, 200,30);

t28=new JTextField(per[28]+"%");

t28.setBounds(50,150, 200,30);

t29=new JTextField(per[29]+"%");

t29.setBounds(50,150, 200,30);

t30=new JTextField(per[30]+"%");

t30.setBounds(50,150, 200,30);

t31=new JTextField(per[31]+"%");

t31.setBounds(50,150, 200,30);

t32=new JTextField(per[32]+"%");

t32.setBounds(50,150, 200,30);

t33=new JTextField(per[33]+"%");

t33.setBounds(50,150, 200,30);

t34=new JTextField(per[34]+"%");

t34.setBounds(50,150, 200,30);

t35=new JTextField(per[35]+"%");

t35.setBounds(50,150, 200,30);

t36=new JTextField(per[36]+"%");

t36.setBounds(50,150, 200,30);

t37=new JTextField(per[37]+"%");

t37.setBounds(50,150, 200,30);

t38=new JTextField(per[38]+"%");

t38.setBounds(50,150, 200,30);

t39=new JTextField(per[39]+"%");

t39.setBounds(50,150, 200,30);

t40=new JTextField(per[40]+"%");

t40.setBounds(50,150, 200,30);

t41=new JTextField(per[41]+"%");

t41.setBounds(50,150, 200,30);

t42=new JTextField(per[42]+"%");

t42.setBounds(50,150, 200,30);

t43=new JTextField(per[43]+"%");

t43.setBounds(50,150, 200,30);

t44=new JTextField(per[44]+"%");

t44.setBounds(50,150, 200,30);

t45=new JTextField(per[45]+"%");

t45.setBounds(50,150, 200,30);

t46=new JTextField(per[46]+"%");

t46.setBounds(50,150, 200,30);

t47=new JTextField(per[47]+"%");

t47.setBounds(50,150, 200,30);

t48=new JTextField(per[48]+"%");

t48.setBounds(50,150, 200,30);

t49=new JTextField(per[49]+"%");

t49.setBounds(50,150, 200,30);

t50=new JTextField(per[50]+"%");

t50.setBounds(50,150, 200,30);

t51=new JTextField(per[51]+"%");

t51.setBounds(50,150, 200,30);

t52=new JTextField(per[52]+"%");

t52.setBounds(50,150, 200,30);

t53=new JTextField(per[53]+"%");

t53.setBounds(50,150, 200,30);

t54=new JTextField(per[54]+"%");

t54.setBounds(50,150, 200,30);

t55=new JTextField(per[55]+"%");

t55.setBounds(50,150, 200,30);

t56=new JTextField(per[56]+"%");

t56.setBounds(50,150, 200,30);

t57=new JTextField(per[57]+"%");

t57.setBounds(50,150, 200,30);

t58=new JTextField(per[58]+"%");

t58.setBounds(50,150, 200,30);

t59=new JTextField(per[59]+"%");

t59.setBounds(50,150, 200,30);

t60=new JTextField(per[60]+"%");

t60.setBounds(50,150, 200,30);

f.add(t1); f.add(t2);f.add(t3);f.add(t4);f.add(t5);

f.add(t6);f.add(t7);f.add(t8);f.add(t9);f.add(t10);

f.add(t11);f.add(t12);f.add(t13);f.add(t14);f.add(t15);

f.add(t16);f.add(t17);f.add(t18);f.add(t19);f.add(t20);

f.add(t21);f.add(t22);f.add(t23);f.add(t24);f.add(t25);

f.add(t26);f.add(t27);f.add(t28);f.add(t29);f.add(t30);

f.add(t31);f.add(t32);f.add(t33);f.add(t34);f.add(t35);

f.add(t36);f.add(t37);f.add(t38);f.add(t39);f.add(t40);

f.add(t41);f.add(t42);f.add(t43);f.add(t44);f.add(t45);

f.add(t46);f.add(t47);f.add(t48);f.add(t49);f.add(t50);

f.add(t51);f.add(t52);f.add(t53);f.add(t54);f.add(t55);

f.add(t56);f.add(t57);f.add(t58);f.add(t59);f.add(t60);

f.setSize(4000,4000);

f.setVisible(true);

JFrame f1= new JFrame("DEFAULTERS\_\_LIST");

f1.setSize(500, 500);

f1.setVisible(true);

f1.getContentPane().setLayout(new FlowLayout());

JTextArea textArea = new JTextArea(20, 20);

JScrollPane scrollableTextArea = new JScrollPane(textArea);

scrollableTextArea.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL\_SCROLLBAR\_ALWAYS);

scrollableTextArea.setVerticalScrollBarPolicy(JScrollPane.VERTICAL\_SCROLLBAR\_ALWAYS);

try {

for(int i=1;i<=60;i++)

{

if(per[i]<75)

textArea.getDocument().insertString(0, "STUDENT ROLL NO : "+i+" and his percentage :"+per[i] + "\n", null);

}

}catch (Exception e1) {

// TODO Auto-generated catch block

System.out.println(e1);

}

f1.getContentPane().add(scrollableTextArea);

con.close();

}catch(Exception e){ System.out.println(e);}

}

});

}

}

class SimpleAudioPlayer extends JFrame {

JButton btn = new JButton("Play Sound");

File wavFile;

URL defaultSound;

public static Clip clip;

public static AudioInputStream audioInputStream;

public SimpleAudioPlayer(String url) {

try {

setSize(300, 100);

setLocation(400, 300);

JPanel jp = new JPanel();

defaultSound = new URL (url);

jp.add(btn);

getContentPane().add(jp);

pack();

btn.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

play();

}

});

} catch (MalformedURLException ex) {

Logger.getLogger(WavPlayer.class.getName()).log(Level.SEVERE, null, ex);

}

}

public void play() {

try {

audioInputStream = AudioSystem.getAudioInputStream(defaultSound);

try {

clip = AudioSystem.getClip();

clip.open(audioInputStream);

clip.loop(20000);

clip.start();

} catch (LineUnavailableException e) {

}

} catch (UnsupportedAudioFileException | IOException e) {

}

}

public void stop() {

clip.stop();

}

}

class Sqlexamp{

public int dbman(String uname,String pwd){

try{

//step1 load the driver class

Class.forName("oracle.jdbc.driver.OracleDriver");

//step2 create the connection object

Connection con=DriverManager.getConnection(

"jdbc:oracle:thin:@localhost:1521:XE","system","batman");

//step3 create the statement object

Statement stmt=con.createStatement();

ResultSet rs=stmt.executeQuery("select \* from passtab");

while(rs.next())

{

String x=rs.getString(1);

String y=rs.getString(2);

if(x.equals(uname) && y.equals(pwd))

{

System.out.println("DATA FOUND");

return 1;

}

else

{

System.out.println("DATA NOT FOUND");

}

}

con.close();

}

catch(Exception e){ System.out.println(e);}

return 0;

}

}

public class Mywork {

public static void main(String[] args) {

int[] a = new int[100];

for(int i=1;i<=60;i++)

{

a[i]=1;

}

JFrame f= new JFrame("TextField Example");

f.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

JTextField t1,t2;

t1=new JTextField("");

t1.setBounds(50,100, 200,30);

t2=new JTextField("");

t2.setBounds(50,200, 200,30);

f.add(t1); f.add(t2);

JLabel l1,l2;

l1=new JLabel("ENTER UR STAFFID");

l1.setBounds(50,50, 150,200);

l2=new JLabel("ENTER UR PASSWORD");

l2.setBounds(50,150, 150,200);

f.add(l1); f.add(l2);

Button b=new Button("LOG\_IN");

b.setBounds(50,400,60,30);

b.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

Sqlexamp ex=new Sqlexamp();

String uname = t1.getText();

String pwd = t2.getText();

int c=ex.dbman(uname,pwd);

if(c==1)

{

new Mywork(a,uname);

}

}

});

f.add(b);

JButton b1=new JButton("SIGN\_UP");

b1.setBounds(500,100,100,100);

f.add(b1);

b1.addActionListener(new ActionListener(){

@Override

public void actionPerformed(ActionEvent e){

try{

//step1 load the driver class

Class.forName("oracle.jdbc.driver.OracleDriver");

//step2 create the connection object

Connection con=DriverManager.getConnection(

"jdbc:oracle:thin:@localhost:1521:XE","system","batman");

//step3 create the statement object

Statement stmt=con.createStatement();

String uname=t1.getText();

String pwd=t2.getText();

String sql = "INSERT INTO PASSTAB " +

"VALUES ('"+uname+"', '"+pwd+"')";

stmt.executeUpdate(sql);

System.out.println("REGISTRATION DONE SUCCESSFULLY");

}

catch(Exception el){ System.out.println(el);}

}

});

f.setSize(1000,1000);

f.setLayout(null);

f.setVisible(true);

}

public Mywork(int a[],String uname) {

EventQueue.invokeLater(new Runnable() {

@Override

public void run() {

try {

UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());

} catch (ClassNotFoundException | InstantiationException | IllegalAccessException | UnsupportedLookAndFeelException ex) {

ex.printStackTrace();

}

JFrame frame = new JFrame("Testing");

frame.setLayout(new GridLayout(0, 2));

//frame.add(new CheckBoxGroup(new String[]{"S1", "S2","S3","S4","S5","S6","S7","S8","S9","S10","S11","S12","S13","S14","S15","S16","S17","S18","S19","S20","S21","S22","S23","S24","S25","S26","S27","S28","S29","S30"}));

//frame.add(new CheckBoxGroup(new String[]{"S31", "S32", "S33", "S34", "S35", "S36", "S37","S38","S39","S40","S41","S42","S43","S44","S45","S46","S47","S48","S49","S50","S51","S52","S53","S54","S55","S56","S57","S58","S59","S60"}));

JCheckBox checkBox1 = new JCheckBox("1",true);

checkBox1.setBounds(100,100, 50,50);

JCheckBox checkBox2 = new JCheckBox("2", true);

checkBox2.setBounds(100,150, 50,50);

JCheckBox checkBox3 = new JCheckBox("3", true);

checkBox3.setBounds(100,200, 50,50);

JCheckBox checkBox4 = new JCheckBox("4", true);

checkBox4.setBounds(100,250, 50,50);

JCheckBox checkBox5 = new JCheckBox("5", true);

checkBox5.setBounds(100,300, 50,50);

JCheckBox checkBox6 = new JCheckBox("6", true);

checkBox6.setBounds(100,350, 50,50);

JCheckBox checkBox7 = new JCheckBox("7", true);

checkBox7.setBounds(100,400, 50,50);

JCheckBox checkBox8 = new JCheckBox("8", true);

checkBox8.setBounds(100,450, 50,50);

JCheckBox checkBox9 = new JCheckBox("9", true);

checkBox9.setBounds(100,500, 50,50);

JCheckBox checkBox10 = new JCheckBox("10", true);

checkBox10.setBounds(100,550, 50,50);

JCheckBox checkBox11 = new JCheckBox("11", true);

checkBox11.setBounds(100,600, 50,50);

JCheckBox checkBox12 = new JCheckBox("12", true);

checkBox12.setBounds(100,650, 50,50);

JCheckBox checkBox13 = new JCheckBox("13", true);

checkBox13.setBounds(100,700, 50,50);

JCheckBox checkBox14 = new JCheckBox("14", true);

checkBox14.setBounds(100,750, 50,50);

JCheckBox checkBox15 = new JCheckBox("15", true);

checkBox15.setBounds(100,800, 50,50);

JCheckBox checkBox16 = new JCheckBox("16", true);

checkBox16.setBounds(100,850, 50,50);

JCheckBox checkBox17 = new JCheckBox("17", true);

checkBox17.setBounds(100,900, 50,50);

JCheckBox checkBox18 = new JCheckBox("18", true);

checkBox18.setBounds(100,950, 50,50);

JCheckBox checkBox19 = new JCheckBox("19", true);

checkBox19.setBounds(100,1000, 50,50);

JCheckBox checkBox20 = new JCheckBox("20", true);

checkBox20.setBounds(100,1050, 50,50);

JCheckBox checkBox21 = new JCheckBox("21", true);

checkBox21.setBounds(100,1100, 50,50);

JCheckBox checkBox22 = new JCheckBox("22", true);

checkBox22.setBounds(100,1150, 50,50);

JCheckBox checkBox23 = new JCheckBox("23", true);

checkBox23.setBounds(100,1200, 50,50);

JCheckBox checkBox24 = new JCheckBox("24", true);

checkBox24.setBounds(100,1250, 50,50);

JCheckBox checkBox25 = new JCheckBox("25", true);

checkBox25.setBounds(100,1300, 50,50);

JCheckBox checkBox26 = new JCheckBox("26", true);

checkBox26.setBounds(100,1350, 50,50);

JCheckBox checkBox27 = new JCheckBox("27", true);

checkBox27.setBounds(100,1400, 50,50);

JCheckBox checkBox28 = new JCheckBox("28", true);

checkBox28.setBounds(100,1450, 50,50);

JCheckBox checkBox29 = new JCheckBox("29", true);

checkBox29.setBounds(100,1500, 50,50);

JCheckBox checkBox30 = new JCheckBox("30", true);

checkBox30.setBounds(100,1550, 50,50);

JCheckBox checkBox31 = new JCheckBox("31", true);

checkBox31.setBounds(100,1600, 50,50);

JCheckBox checkBox32 = new JCheckBox("32", true);

checkBox32.setBounds(100,1650, 50,50);

JCheckBox checkBox33 = new JCheckBox("33", true);

checkBox33.setBounds(100,1700, 50,50);

JCheckBox checkBox34 = new JCheckBox("34", true);

checkBox34.setBounds(100,1750, 50,50);

JCheckBox checkBox35 = new JCheckBox("35", true);

checkBox35.setBounds(100,1800, 50,50);

JCheckBox checkBox36 = new JCheckBox("36", true);

checkBox36.setBounds(100,1850, 50,50);

JCheckBox checkBox37 = new JCheckBox("37", true);

checkBox37.setBounds(100,1900, 50,50);

JCheckBox checkBox38 = new JCheckBox("38", true);

checkBox38.setBounds(100,1950, 50,50);

JCheckBox checkBox39 = new JCheckBox("39", true);

checkBox39.setBounds(100,2000, 50,50);

JCheckBox checkBox40 = new JCheckBox("40", true);

checkBox40.setBounds(100,2050, 50,50);

JCheckBox checkBox41 = new JCheckBox("41", true);

checkBox41.setBounds(100,2100, 50,50);

JCheckBox checkBox42 = new JCheckBox("42", true);

checkBox42.setBounds(100,2150, 50,50);

JCheckBox checkBox43 = new JCheckBox("43", true);

checkBox43.setBounds(100,2200, 50,50);

JCheckBox checkBox44 = new JCheckBox("44", true);

checkBox44.setBounds(100,2250, 50,50);

JCheckBox checkBox45 = new JCheckBox("45", true);

checkBox45.setBounds(100,2300, 50,50);

JCheckBox checkBox46 = new JCheckBox("46", true);

checkBox46.setBounds(100,2350, 50,50);

JCheckBox checkBox47 = new JCheckBox("47", true);

checkBox47.setBounds(100,2400, 50,50);

JCheckBox checkBox48 = new JCheckBox("48", true);

checkBox48.setBounds(100,2450, 50,50);

JCheckBox checkBox49 = new JCheckBox("49", true);

checkBox49.setBounds(100,2500, 50,50);

JCheckBox checkBox50 = new JCheckBox("50", true);

checkBox50.setBounds(100,2550, 50,50);

JCheckBox checkBox51 = new JCheckBox("51", true);

checkBox51.setBounds(100,2600, 50,50);

JCheckBox checkBox52 = new JCheckBox("52", true);

checkBox52.setBounds(100,2650, 50,50);

JCheckBox checkBox53 = new JCheckBox("53", true);

checkBox53.setBounds(100,2700, 50,50);

JCheckBox checkBox54 = new JCheckBox("54", true);

checkBox54.setBounds(100,2750, 50,50);

JCheckBox checkBox55 = new JCheckBox("55", true);

checkBox55.setBounds(100,2800, 50,50);

JCheckBox checkBox56 = new JCheckBox("56", true);

checkBox56.setBounds(100,2850, 50,50);

JCheckBox checkBox57 = new JCheckBox("57", true);

checkBox57.setBounds(100,2900, 50,50);

JCheckBox checkBox58 = new JCheckBox("58", true);

checkBox58.setBounds(100,2950, 50,50);

JCheckBox checkBox59 = new JCheckBox("59", true);

checkBox59.setBounds(100,3000, 50,50);

JCheckBox checkBox60 = new JCheckBox("60", true);

checkBox60.setBounds(100,3050, 50,50);

frame.add(checkBox1);

frame.add(checkBox2);

frame.add(checkBox3);

frame.add(checkBox4);

frame.add(checkBox5);

frame.add(checkBox6);

frame.add(checkBox7);

frame.add(checkBox8);

frame.add(checkBox9);

frame.add(checkBox10);

frame.add(checkBox11);

frame.add(checkBox12);

frame.add(checkBox13);

frame.add(checkBox14);

frame.add(checkBox15);

frame.add(checkBox16);

frame.add(checkBox17);

frame.add(checkBox18);

frame.add(checkBox19);

frame.add(checkBox20);

frame.add(checkBox21);

frame.add(checkBox22);

frame.add(checkBox23);

frame.add(checkBox24);

frame.add(checkBox25);

frame.add(checkBox26);

frame.add(checkBox27);

frame.add(checkBox28);

frame.add(checkBox29);

frame.add(checkBox30);

frame.add(checkBox31);

frame.add(checkBox32);

frame.add(checkBox33);

frame.add(checkBox34);

frame.add(checkBox35);

frame.add(checkBox36);

frame.add(checkBox37);

frame.add(checkBox38);

frame.add(checkBox39);

frame.add(checkBox40);

frame.add(checkBox41);

frame.add(checkBox42);

frame.add(checkBox43);

frame.add(checkBox44);

frame.add(checkBox45);

frame.add(checkBox46);

frame.add(checkBox47);

frame.add(checkBox48);

frame.add(checkBox49);

frame.add(checkBox50);

frame.add(checkBox51);

frame.add(checkBox52);

frame.add(checkBox53);

frame.add(checkBox54);

frame.add(checkBox55);

frame.add(checkBox56);

frame.add(checkBox57);

frame.add(checkBox58);

frame.add(checkBox59);

frame.add(checkBox60);

checkBox1.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[1]=1;

else

a[1]=0;

}

});

checkBox2.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[2]=1;

else

a[2]=0;

}

});

checkBox3.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[3]=1;

else

a[3]=0;

}

});

checkBox4.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[4]=1;

else

a[4]=0;

}

});

checkBox5.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[5]=1;

else

a[5]=0;

}

});

checkBox6.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[6]=1;

else

a[6]=0;

}

});

checkBox7.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[7]=1;

else

a[7]=0;

}

});

checkBox8.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[8]=1;

else

a[8]=0;

}

});

checkBox9.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[9]=1;

else

a[9]=0;

}

});

checkBox10.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[10]=1;

else

a[10]=0;

}

});

checkBox11.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[11]=1;

else

a[11]=0;

}

});

checkBox12.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[12]=1;

else

a[12]=0;

}

});

checkBox13.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[13]=1;

else

a[13]=0;

}

});

checkBox14.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[14]=1;

else

a[14]=0;

}

});

checkBox15.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[15]=1;

else

a[15]=0;

}

});

checkBox16.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[16]=1;

else

a[16]=0;

}

});

checkBox17.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[17]=1;

else

a[17]=0;

}

});

checkBox18.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[18]=1;

else

a[18]=0;

}

});

checkBox19.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[19]=1;

else

a[19]=0;

}

});

checkBox20.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[20]=1;

else

a[20]=0;

}

});

checkBox21.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[21]=1;

else

a[21]=0;

}

});

checkBox22.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[22]=1;

else

a[22]=0;

}

});

checkBox23.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[23]=1;

else

a[23]=0;

}

});

checkBox24.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[24]=1;

else

a[24]=0;

}

});

checkBox25.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[25]=1;

else

a[25]=0;

}

});

checkBox26.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[26]=1;

else

a[26]=0;

}

});

checkBox27.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[27]=1;

else

a[27]=0;

}

});

checkBox28.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[28]=1;

else

a[28]=0;

}

});

checkBox29.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[29]=1;

else

a[29]=0;

}

});

checkBox30.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[30]=1;

else

a[30]=0;

}

});

checkBox31.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[31]=1;

else

a[31]=0;

}

});

checkBox32.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[32]=1;

else

a[32]=0;

}

});

checkBox33.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[33]=1;

else

a[33]=0;

}

});

checkBox34.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[34]=1;

else

a[34]=0;

}

});

checkBox35.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[35]=1;

else

a[35]=0;

}

});

checkBox36.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[36]=1;

else

a[36]=0;

}

});

checkBox37.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[37]=1;

else

a[37]=0;

}

});

checkBox38.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[38]=1;

else

a[38]=0;

}

});

checkBox39.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[39]=1;

else

a[39]=0;

}

});

checkBox40.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[40]=1;

else

a[40]=0;

}

});

checkBox41.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[41]=1;

else

a[41]=0;

}

});

checkBox42.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[42]=1;

else

a[42]=0;

}

});

checkBox43.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[43]=1;

else

a[43]=0;

}

});

checkBox44.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[44]=1;

else

a[44]=0;

}

});

checkBox45.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[45]=1;

else

a[45]=0;

}

});

checkBox46.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[46]=1;

else

a[46]=0;

}

});

checkBox47.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[47]=1;

else

a[47]=0;

}

});

checkBox48.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[48]=1;

else

a[48]=0;

}

});

checkBox49.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[49]=1;

else

a[49]=0;

}

});

checkBox50.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[50]=1;

else

a[50]=0;

}

});

checkBox51.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[51]=1;

else

a[51]=0;

}

});

checkBox52.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[52]=1;

else

a[52]=0;

}

});

checkBox53.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[53]=1;

else

a[53]=0;

}

});

checkBox54.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[54]=1;

else

a[54]=0;

}

});

checkBox55.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[55]=1;

else

a[55]=0;

}

});

checkBox56.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[56]=1;

else

a[56]=0;

}

});

checkBox57.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[57]=1;

else

a[57]=0;

}

});

checkBox58.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[58]=1;

else

a[58]=0;

}

});

checkBox59.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[59]=1;

else

a[59]=0;

}

});

checkBox60.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED)

a[60]=1;

else

a[60]=0;

}

});

final JTextField tf=new JTextField();

tf.setBounds(50,50, 150,20);

frame.add(tf);

final JTextField tf1=new JTextField();

tf1.setBounds(50,50, 150,20);

frame.add(tf1);

JButton b=new JButton("LOCK FOR TODAY");

b.setBounds(50,100,95,30);

frame.add(b);

frame.setSize(400,400);

frame.setVisible(true);

JButton b1=new JButton("LOCK PERMANENTLY AND VIEW ATTENDANCE SHEET");

b1.setBounds(50,100,95,30);

frame.add(b1);

frame.setSize(400,400);

frame.setVisible(true);

JButton b2=new JButton("Start Audio Clip");

b1.setBounds(50,100,95,30);

frame.add(b2);

frame.setSize(4000,4000);

frame.setVisible(true);

frame.pack();

frame.setLocationRelativeTo(null);

frame.setVisible(true);

//first button event

b.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

tf.setText("Welcome to Javatpoint.");

try{

//step1 load the driver class

Class.forName("oracle.jdbc.driver.OracleDriver");

//step2 create the connection object

Connection con=DriverManager.getConnection(

"jdbc:oracle:thin:@localhost:1521:XE","system","batman");

//step3 create the statement object

Statement stmt=con.createStatement();

DateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");

Date date = new Date();

System.out.println(dateFormat.format(date)); //2016/11/16 12:08:43

String sql = "INSERT INTO attendance " +

"VALUES ('"+uname+"', '"+date+"','"+a[1]+"','"+a[2]+"','"+a[3]+"','"+a[4]+"','"+a[5]+"','"+a[6]+"','"+a[7]+"','"+a[8]+"','"+a[9]+"','"+a[10]+"','"+a[11]+"','"+a[12]+"','"+a[13]+"','"+a[14]+"','"+a[15]+"','"+a[16]+"','"+a[17]+"','"+a[18]+"','"+a[19]+"','"+a[20]+"','"+a[21]+"','"+a[22]+"','"+a[23]+"','"+a[24]+"','"+a[25]+"','"+a[26]+"','"+a[27]+"','"+a[28]+"','"+a[29]+"','"+a[30]+"','"+a[31]+"','"+a[32]+"','"+a[33]+"','"+a[34]+"','"+a[35]+"','"+a[36]+"','"+a[37]+"','"+a[38]+"','"+a[39]+"','"+a[40]+"','"+a[41]+"','"+a[42]+"','"+a[43]+"','"+a[44]+"','"+a[45]+"','"+a[46]+"','"+a[47]+"','"+a[48]+"','"+a[49]+"','"+a[50]+"','"+a[51]+"','"+a[52]+"','"+a[53]+"','"+a[54]+"','"+a[55]+"','"+a[56]+"','"+a[57]+"','"+a[58]+"','"+a[59]+"','"+a[60]+"')";

stmt.executeUpdate(sql);

/\*String sqll="INSERT INTO hihi " + "Values ('"+date+"')";

stmt.executeUpdate(sqll);\*/

//step5 close the connection object

con.close();

}catch(Exception exx){ System.out.println(exx);}

}

});

//second button event

b1.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

tf1.setText("Welcome to Javatpoint.");

Retrieve r=new Retrieve();

r.runn(uname);

}

});

//audio clip button event

b2.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

SimpleAudioPlayer t = new SimpleAudioPlayer("file:C:/image/song.wav");

t.setVisible(true);

}

});

}

});

}

public class CheckBoxGroup extends JPanel {

private JCheckBox all;

private List<JCheckBox> checkBoxes;

public CheckBoxGroup(String... options) {

checkBoxes = new ArrayList<>(25);

setLayout(new BorderLayout());

JPanel header = new JPanel(new FlowLayout(FlowLayout.LEFT, 1, 1));

all = new JCheckBox("Select All...");

all.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

for (JCheckBox cb : checkBoxes) {

cb.setSelected(all.isSelected());

}

}

});

header.add(all);

add(header, BorderLayout.NORTH);

JPanel content = new ScrollablePane(new GridBagLayout());

content.setBackground(UIManager.getColor("List.background"));

if (options.length > 0) {

GridBagConstraints gbc = new GridBagConstraints();

gbc.gridwidth = GridBagConstraints.REMAINDER;

gbc.anchor = GridBagConstraints.NORTHWEST;

gbc.weightx = 1;

for (int index = 0; index < options.length - 1; index++) {

JCheckBox cb = new JCheckBox(options[index]);

cb.setOpaque(false);

checkBoxes.add(cb);

content.add(cb, gbc);

}

JCheckBox cb = new JCheckBox(options[options.length - 1]);

cb.setOpaque(false);

checkBoxes.add(cb);

gbc.weighty = 1;

content.add(cb, gbc);

}

add(new JScrollPane(content));

}

public class ScrollablePane extends JPanel implements Scrollable {

public ScrollablePane(LayoutManager layout) {

super(layout);

}

public ScrollablePane() {

}

@Override

public Dimension getPreferredScrollableViewportSize() {

return new Dimension(100, 100);

}

@Override

public int getScrollableUnitIncrement(Rectangle visibleRect, int orientation, int direction) {

return 32;

}

@Override

public int getScrollableBlockIncrement(Rectangle visibleRect, int orientation, int direction) {

return 32;

}

@Override

public boolean getScrollableTracksViewportWidth() {

boolean track = false;

Container parent = getParent();

if (parent instanceof JViewport) {

JViewport vp = (JViewport) parent;

track = vp.getWidth() > getPreferredSize().width;

}

return track;

}

@Override

public boolean getScrollableTracksViewportHeight() {

boolean track = false;

Container parent = getParent();

if (parent instanceof JViewport) {

JViewport vp = (JViewport) parent;

track = vp.getHeight() > getPreferredSize().height;

}

return track;

}

}

}

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

/\*\*

\*

\* @author VARADHARAJAN

\*/

**Back End Triggers and Procedures SQL PLUS ORACLE DB ☺**

**Table Structures:**

SQL> desc passtab;

Name Null? Type

----------------------------------------- -------- ----------------------------

STAFFID NOT NULL VARCHAR2(30)

PASSWORD NOT NULL VARCHAR2(30)

SQL> desc attendance;

Name Null? Type

----------------------------------------- -------- ----------------------------

STAFFID NOT NULL VARCHAR2(15)

DATES VARCHAR2(50)

S1 NUMBER(38)

S2 NUMBER(38)

S3 NUMBER(38)

S4 NUMBER(38)

S5 NUMBER(38)

S6 NUMBER(38)

S7 NUMBER(38)

S8 NUMBER(38)

S9 NUMBER(38)

S10 NUMBER(38)

S11 NUMBER(38)

S12 NUMBER(38)

S13 NUMBER(38)

S14 NUMBER(38)

S15 NUMBER(38)

S16 NUMBER(38)

S17 NUMBER(38)

S18 NUMBER(38)

S19 NUMBER(38)

S20 NUMBER(38)

S21 NUMBER(38)

S22 NUMBER(38)

S23 NUMBER(38)

S24 NUMBER(38)

S25 NUMBER(38)

S26 NUMBER(38)

S27 NUMBER(38)

S28 NUMBER(38)

S29 NUMBER(38)

S30 NUMBER(38)

S31 NUMBER(38)

S32 NUMBER(38)

S33 NUMBER(38)

S34 NUMBER(38)

S35 NUMBER(38)

S36 NUMBER(38)

S37 NUMBER(38)

S38 NUMBER(38)

S39 NUMBER(38)

S40 NUMBER(38)

S41 NUMBER(38)

S42 NUMBER(38)

S43 NUMBER(38)

S44 NUMBER(38)

S45 NUMBER(38)

S46 NUMBER(38)

S47 NUMBER(38)

S48 NUMBER(38)

S49 NUMBER(38)

S50 NUMBER(38)

S51 NUMBER(38)

S52 NUMBER(38)

S53 NUMBER(38)

S54 NUMBER(38)

S55 NUMBER(38)

S56 NUMBER(38)

S57 NUMBER(38)

S58 NUMBER(38)

S59 NUMBER(38)

S60 NUMBER(38)

**PROCEDURE FOR IDENTIFYING THE STAFFS WHO LOGGED IN ON A PARTICULAR DAY:**

CREATE PROCEDURE a\_proc(dt in varchar2)

AS

CURSOR names\_cur IS

SELECT staffid

FROM attendance

WHERE dates=dt;

names\_t names\_cur%ROWTYPE;

TYPE names\_ntt IS TABLE OF names\_t%TYPE; -- must use type

l\_names names\_ntt;

BEGIN

OPEN names\_cur;

FETCH names\_cur BULK COLLECT INTO l\_names;

CLOSE names\_cur;

FOR indx IN 1..l\_names.COUNT LOOP

DBMS\_OUTPUT.PUT\_LINE(l\_names(indx).staffid);

END LOOP;

END a\_proc;

**PROCEDURE FOR TAKING STUDENTCOUNT ON A PARTICULAR DAY FOR A PARTICULAR STAFF:**

create or replace procedure studentcount(dt in varchar2,id in varchar2,c out int)

as

begin

select s1+s2+s3+s4+s5+s6+s7+s8+s9+s10+s11+s12+s13+s14+s15+s16+s17+s18+s19+s20+s21+s22+s23+s24+s25+S26+s27+s28

+s29+s30+s31+s32+s33+S34+s35+s36+s37+s38+s39+s40+s41+s42+s43+s44+s45+s46+s47+s48+s49+s50+s51+s52+s53+s54+s55+s56+S57+s58+s59+s60 into c from

attendance where(staffid=id and dates=dt);

dbms\_output.put\_line("students present today"||c);

end;

/

**TRIGGER FOR BACKINGUP OF PASSTAB TABLE INTO PASSTAB1 TABLE:**

create or replace trigger t1 after insert on passtab

for eeach row

begin

insert into passtab1 values(:new.staffid,:new.password);

end;

/

**TRIGGER FOR BACKINGUP OF ATTENDANCE TABLE INTO ATTENDACNE1 TABLE:**

create or replace trigger t11 after insert on attendance

for each row

begin

insert into attedance1 values(:new.staffid,:new.dates,:new.s1,:new.s2,:new.s3,:new.s4,:new.s5

,:new.s6,:new.s7,:new.s8,:new.s9,:new.s10,:new.s11,:new.s12,:new.s13,:new.s14,:new.s15,:new.s16,:new.s17

,:new.s18,:new.s19,:new.s20,:new.s21,:new.s22,:new.s23,:new.s24,:new.s25,:new.s26,:new.s27,:new.s28

,:new.s29,:new.s30,:new.s31,:new.s32,:new.s33,:new.s34,:new.s35,:new.s36,:new.s37,:new.s38,:new.s39

,:new.s40,:new.s41,:new.s42,:new.s43,:new.s44,:new.s45,:new.s46,:new.s47,:new.s48,:new.s49

,:new.s50,:new.s51,:new.s52,:new.s53,:new.s54,:new.s55,:new.s56,:new.s57,:new.s58,:new.s59,:new.s60);

end;

/

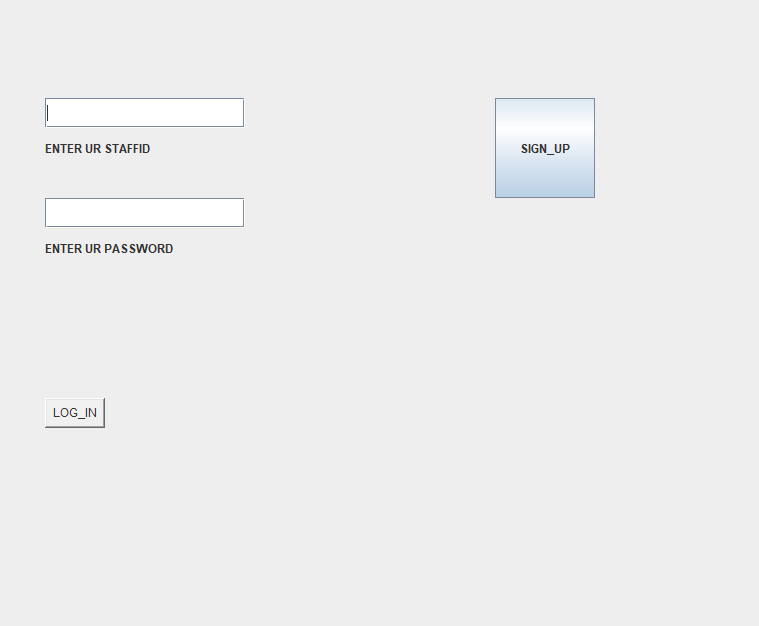
**Abstract**

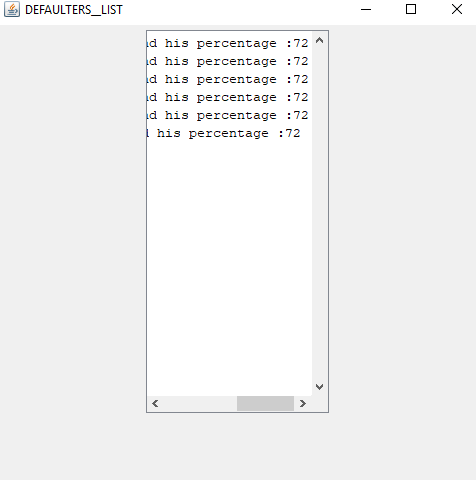
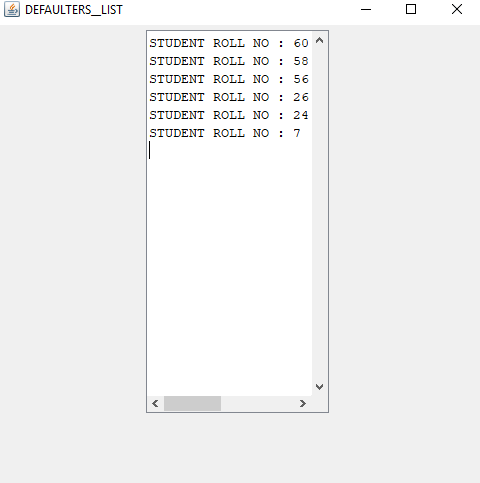
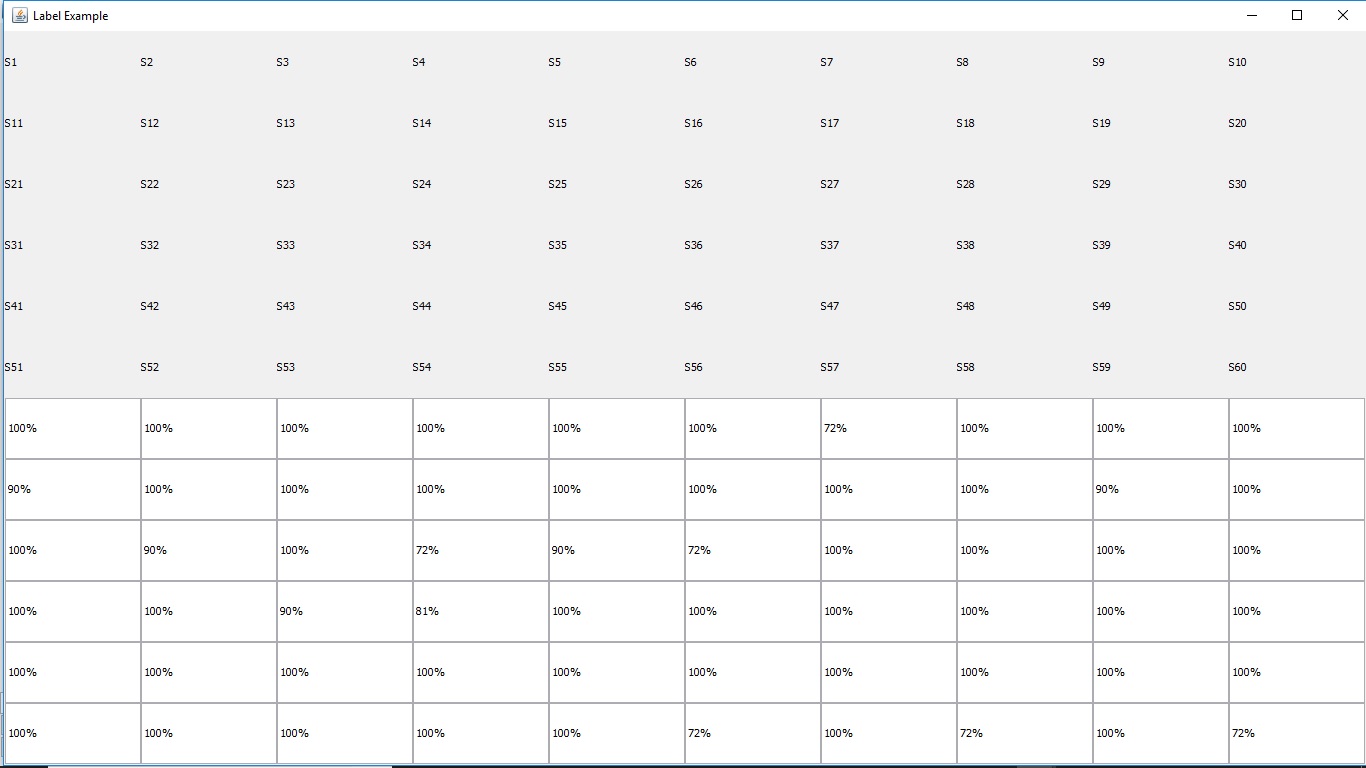
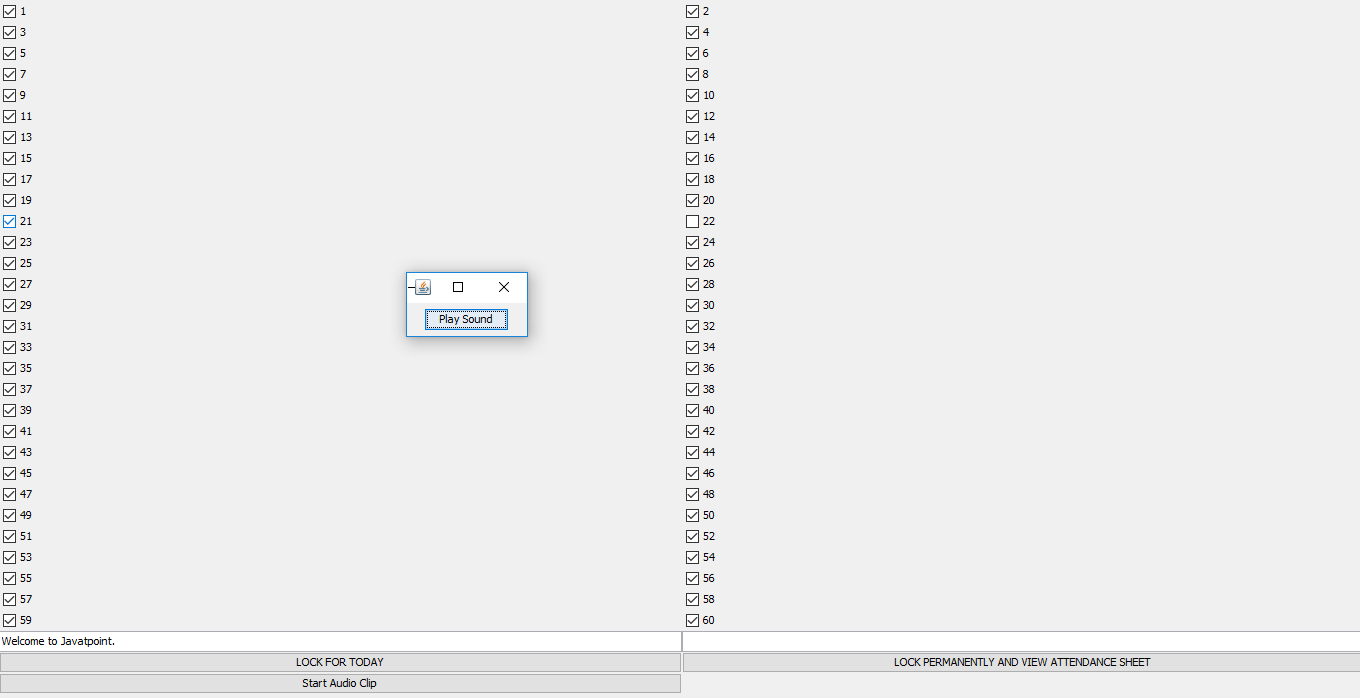
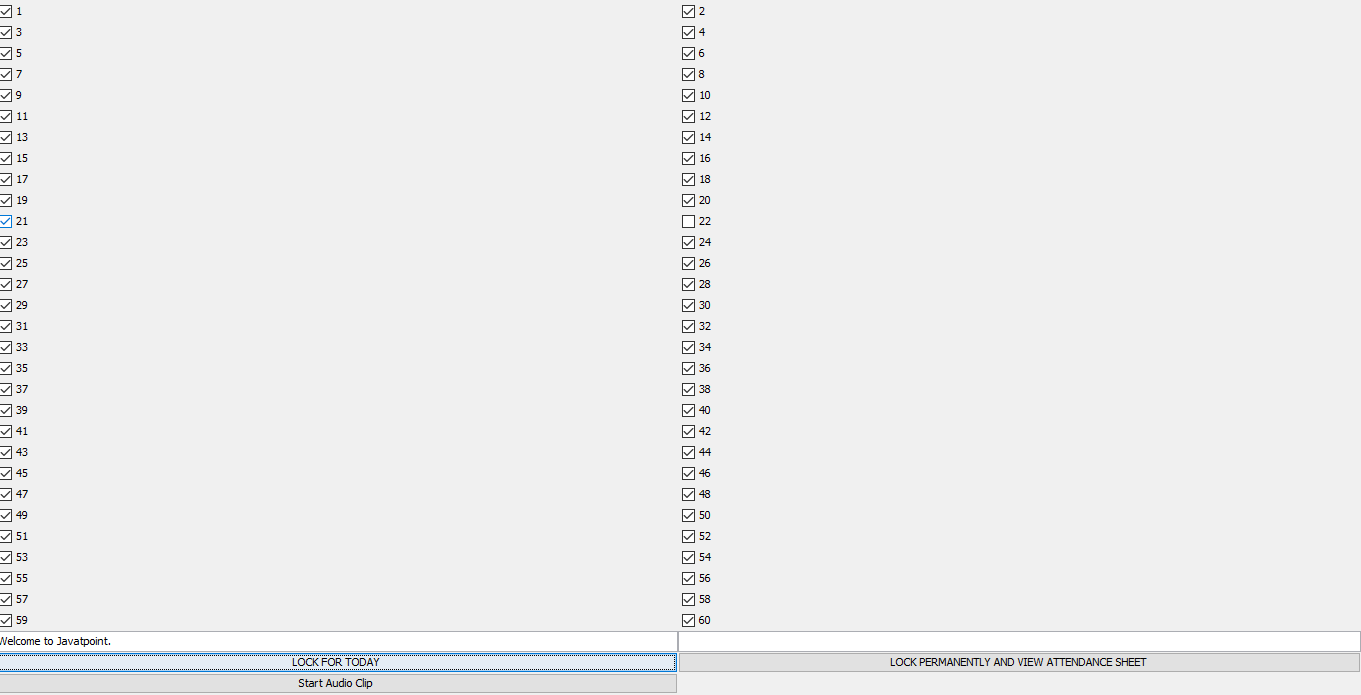
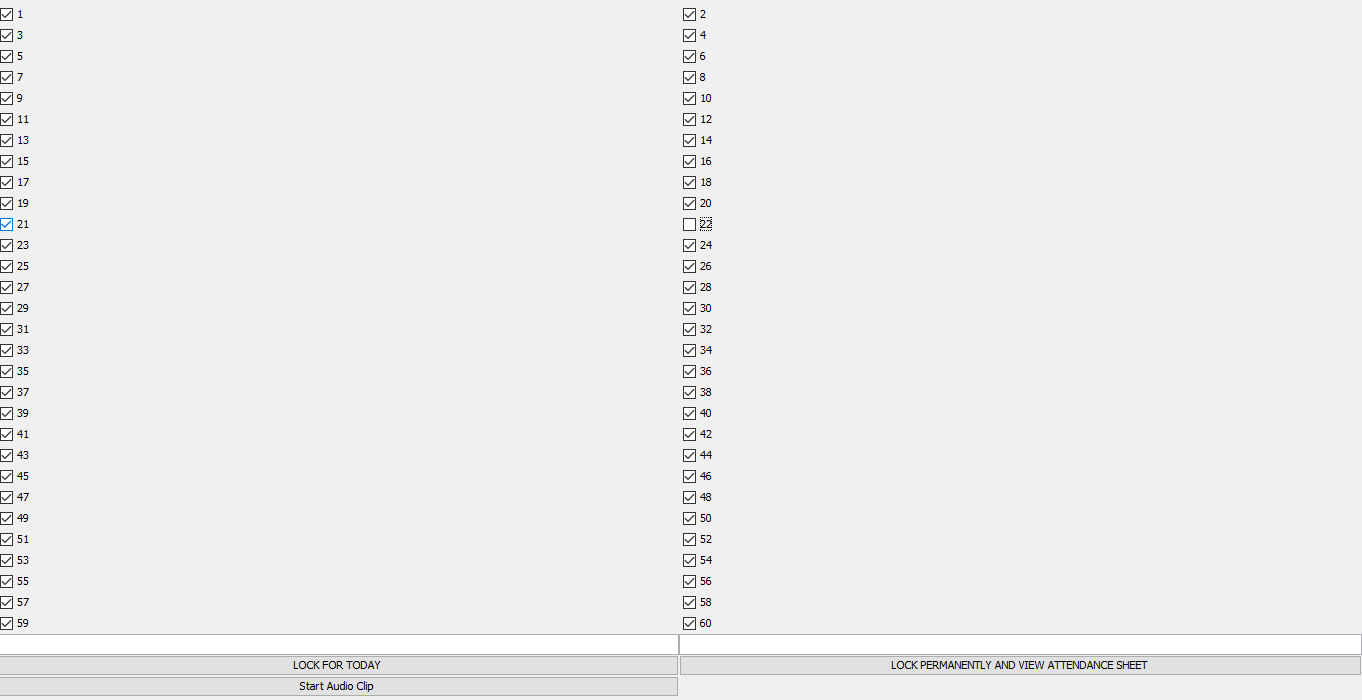
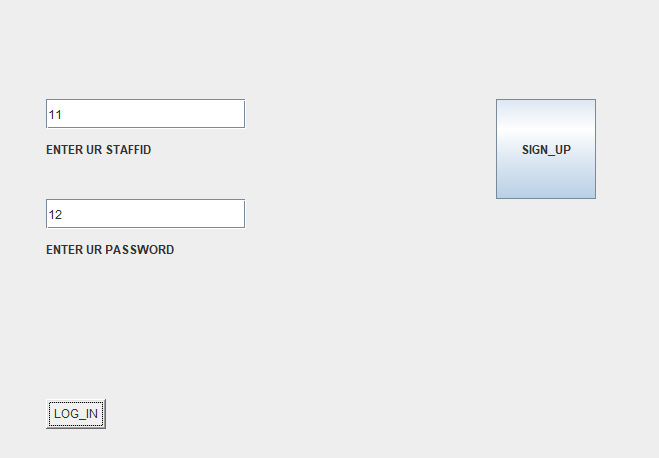
The main idea beyond developing this project is to create a user friendly application that will help any course instructor taking the daily attendance of the students very easily .It extends its application also in generating the defaulters list who generally have their attendance less than 75%.The application prototype is constructed in a manner such its user interface (front) end being developed with Java Virtual Machine (Netbeans IDE) with Oracle 11g Database as its back (storage) end. In this modern era the very classical way of taking attendances may not be comfortable for everyone. So, it’s essential to build such android applications with the ease of marking daily attendances and in a way helpful to the higher authorities of educat ion division of any government or private institutions have a watch over the particular instructor.

The hierarchical control of the attendance database can be maintained through a separate administrator only granting particular permissions to the users according to their designation. This when further extended as a web supporting app will help the education department have control over the progress of every staff in instruction of the courses they have taken charge of. Instead of a classical way of accessing the app through username and password the parameters can be set like the staff’s fingerprint that would serve as the staff’s username and the password is a location of the classroom where they are supposed to instruct the course.

This application when build with android tools like Speech recognition and Artificial Intelligence software will tend to reduce the workload of staffs too by calling out the student names itself and when voice recognition and face detection methods are applied in avoiding the mismatches of student ID’s by positioning and searching the student on a particular axis where the sound has arouse from and thermal image processing whether the student face matches with his existing face in its Database, will yield better results thereby reducing the workload of the staff. In this world which is tending now towards job automation this application will have a major impact thereby helping the instructed machines to function like a staff obliged to rules and regulations.

**Front end Images:**

****

****

**Conclusion:**

If this project is enhanced with add-on features such as API for voice recognition of students with target identification camera hardware and other such full automation features it will be a very great one for the staff and the administration side to avoid the errors that may arise in attendance calculation.