

Table of Contents:

Table of Contents:	1
Login Page:	2
Sign Up Page:	13
Salary Generator Page:	28
Manipulate Employee Records Page:	110
Salary History Page:	164
Loan History Page:	218
Manipulate SSS Allocations Page:	257
Manipulate PhilHealth Allocations:	294
Manipulate HDMF Allocations Page:	331
EmailSender.java	368
NumericDocumentFilter.java	371
IntegerDocumentFilter.java	373

Login Page:

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
 license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */
```

```
import java.awt.Color;
import java.io.File;
import java.sql.Connection;
import java.sql.DatabaseMetaData;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
```

```
/**
 *
 * @author User
 */
public class Login extends javax.swing.JFrame {
```

```
    /**
     * Creates new form Login
     */
    private void initComponents2(){
        //Overwrite Java Driver Code
        //Set textboxes to clear background
        txtEmail.setBackground(new Color(0, 0, 0, 0));
        txtPassword.setBackground(new Color(0, 0, 0, 0));
    }
```

```
    public static boolean doesTableExists(String tableName, Connection conn)
        throws SQLException
    {
        DatabaseMetaData meta = conn.getMetaData();
        ResultSet result = meta.getTables(null, null, tableName.toUpperCase(), null);

        return result.next();
    }
```

```

    }

    public Login() {
        initComponents();
        try{
            initComponents2();
            String currentURL = "jdbc:derby:SALARY_GENERATOR;create=true";
            Connection con = DriverManager.getConnection(currentURL);

            Statement st = con.createStatement();

            if (!doesTableExists("users", con)){
                String sql = "CREATE TABLE users (eid integer PRIMARY KEY, "
                    + "email varchar(100), password varchar(100))";
                st.execute(sql);

                //      System.out.println("created");
            }
            con.close();

        }
        catch (Exception e){
            e.printStackTrace();
        }
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        pnlBackground = new com.k33ptoo.components.KGradientPanel();
        pnlLogin = new javax.swing.JPanel();
        txtEmail = new javax.swing.JTextField();
        lblWelcome = new javax.swing.JLabel();
        lblEmail = new javax.swing.JLabel();
        lblPassword = new javax.swing.JLabel();
        txtPassword = new javax.swing.JPasswordField();
        btnShowPassword = new javax.swing.JCheckBox();

```

```

kbtnSignIn = new com.k33ptoo.components.KButton();
kbtnRegister = new com.k33ptoo.components.KButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setBackground(new java.awt.Color(255, 255, 255));
setLocationByPlatform(true);
setResizable(false);

pnlBackground.setkEndColor(new java.awt.Color(194, 233, 251));
pnlBackground.setkGradientFocus(0);
pnlBackground.setkStartColor(new java.awt.Color(161, 196, 253));

pnlLogin.setBackground(new java.awt.Color(55, 71, 133));
pnlLogin.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
pnlLogin.setForeground(new java.awt.Color(36, 48, 94));

txtEmail.setBackground(new java.awt.Color(0, 0, 0));
txtEmail.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtEmail.setForeground(new java.awt.Color(250, 252, 255));
txtEmail.setBorder(javax.swing.BorderFactory.createMatteBorder(0, 0, 2, 0, new
java.awt.Color(252, 250, 255)));
txtEmail.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtEmail.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtEmailActionPerformed(evt);
    }
});

lblWelcome.setBackground(new java.awt.Color(252, 250, 255));
lblWelcome.setFont(new java.awt.Font("Segoe UI Light", 0, 36)); // NOI18N
lblWelcome.setForeground(new java.awt.Color(252, 250, 255));
lblWelcome.setText("Welcome Back");
lblWelcome.setToolTipText("");
lblWelcome.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

lblEmail.setBackground(new java.awt.Color(252, 250, 255));
lblEmail.setFont(new java.awt.Font("Segoe UI Light", 0, 24)); // NOI18N
lblEmail.setForeground(new java.awt.Color(252, 250, 255));
lblEmail.setText("Email:");

lblPassword.setBackground(new java.awt.Color(252, 250, 255));
lblPassword.setFont(new java.awt.Font("Segoe UI Light", 0, 24)); // NOI18N
lblPassword.setForeground(new java.awt.Color(252, 250, 255));

```

```

lblPassword.setText("Password:");

txtPassword.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtPassword.setForeground(new java.awt.Color(250, 252, 255));
txtPassword.setBorder(javax.swing.BorderFactory.createMatteBorder(0, 0, 2, 0, new
java.awt.Color(250, 252, 255)));
txtPassword.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtPassword.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtPasswordActionPerformed(evt);
    }
});

btnShowPassword.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
btnShowPassword.setForeground(new java.awt.Color(252, 250, 255));
btnShowPassword.setText("Show Password");
btnShowPassword.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnShowPasswordActionPerformed(evt);
    }
});

kbtnSignIn.setBorder(null);
kbtnSignIn.setText("Sign In");
kbtnSignIn.setFont(new java.awt.Font("Segoe UI Light", 1, 18)); // NOI18N
kbtnSignIn.setkBorderRadius(50);
kbtnSignIn.setkEndColor(new java.awt.Color(36, 48, 94));
kbtnSignIn.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnSignIn.setkHoverForeGround(new java.awt.Color(255, 255, 255));
kbtnSignIn.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnSignIn.setkIndicatorThickness(5);
kbtnSignIn.setkStartColor(new java.awt.Color(36, 48, 94));
kbtnSignIn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnSignInActionPerformed(evt);
    }
});

kbtnRegister.setBorder(null);
kbtnRegister.setText("Register");
kbtnRegister.setFont(new java.awt.Font("Segoe UI Light", 1, 18)); // NOI18N
kbtnRegister.setkBorderRadius(50);
kbtnRegister.setkEndColor(new java.awt.Color(36, 48, 94));
kbtnRegister.setkHoverEndColor(new java.awt.Color(76, 111, 167));

```

```

kbtnRegister.setkHoverForeGround(new java.awt.Color(255, 255, 255));
kbtnRegister.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnRegister.setkIndicatorThickness(5);
kbtnRegister.setkStartColor(new java.awt.Color(36, 48, 94));
kbtnRegister.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnRegisterActionPerformed(evt);
    }
});

javax.swing.GroupLayout pnlLoginLayout = new javax.swing.GroupLayout(pnlLogin);
pnlLogin.setLayout(pnlLoginLayout);
pnlLoginLayout.setHorizontalGroup(
    pnlLoginLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlLoginLayout.createSequentialGroup()
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGroup(pnlLoginLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(btnShowPassword)
        .addComponent(lblEmail, javax.swing.GroupLayout.PREFERRED_SIZE, 159,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(pnlLoginLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING,
false)
        .addComponent(txtEmail)
        .addComponent(lblPassword, javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.PREFERRED_SIZE, 159,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(txtPassword, javax.swing.GroupLayout.PREFERRED_SIZE,
260, javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGroup(pnlLoginLayout.createSequentialGroup()
        .addGap(53, 53, 53))
        .addGroup(pnlLoginLayout.createSequentialGroup()

.addGroup(pnlLoginLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlLoginLayout.createSequentialGroup()
        .addGap(37, 37, 37)
        .addComponent(kbtnSignIn, javax.swing.GroupLayout.PREFERRED_SIZE, 140,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(kbtnRegister, javax.swing.GroupLayout.PREFERRED_SIZE,
140, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(pnlLoginLayout.createSequentialGroup()
        .addGap(73, 73, 73)

```

```

        .addComponent(lblWelcome)))
        .addContainerGap(39, Short.MAX_VALUE))
    );
    pnlLoginLayout.setVerticalGroup(
        pnlLoginLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlLoginLayout.createSequentialGroup()
            .addGap(18, 18, 18)
            .addComponent(lblWelcome, javax.swing.GroupLayout.PREFERRED_SIZE, 68,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(lblEmail, javax.swing.GroupLayout.PREFERRED_SIZE, 32,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(txtEmail, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(30, 30, 30)
            .addComponent(lblPassword, javax.swing.GroupLayout.PREFERRED_SIZE, 32,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(txtPassword, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(btnShowPassword)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 69,
Short.MAX_VALUE)

        .addGroup(pnlLoginLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        )
            .addComponent(kbtnRegister, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(kbtnSignIn, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(25, 25, 25))
    );

    javax.swing.GroupLayout pnlBackgroundLayout = new
javax.swing.GroupLayout(pnlBackground);
    pnlBackground.setLayout(pnlBackgroundLayout);
    pnlBackgroundLayout.setHorizontalGroup(

    pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlBackgroundLayout.createSequentialGroup()
            .addGap(398, 398, 398)

```

```

        .addComponent(pnlLogin, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(422, Short.MAX_VALUE))
    );
    pnlBackgroundLayout.setVerticalGroup(

pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlBackgroundLayout.createSequentialGroup()
    .addContainerGap(98, Short.MAX_VALUE)
    .addComponent(pnlLogin, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(129, 129, 129))
    );

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(pnlBackground, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(pnlBackground, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );

    pack();
    setLocationRelativeTo(null);
} // </editor-fold>

private void txtEmailActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void btnShowPasswordActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        if (btnShowPassword.isSelected()==false){//Hide password
            txtPassword.setEchoChar('*');
        }
        else if (btnShowPassword.isSelected()==true){//Show password
            txtPassword.setEchoChar((char)0);
        }
    }
}

```



```

    }
    catch (Exception e){

    }
}

private void txtPasswordActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void kbtnRegisterActionPerformed(java.awt.event.ActionEvent evt) {
    try{//Go to Register Page
        new Register().setVisible(true);
        this.setVisible(false);
    }
    catch (Exception e){

    }

}

private void kbtnSignInActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        String email = txtEmail.getText().trim().toUpperCase();

        //Email Error Trapping
        if (email.length()==0){
            JOptionPane.showMessageDialog(null, "Email Field Empty. Please enter an email.");
        }
        else if (email.length(>100){
            JOptionPane.showMessageDialog(null, "Invalid Email. The email is longer than 100
characters.");
        }
        else{
            char chars[] = txtPassword.getPassword();
            String password = "";
            for (int i = 0; i<chars.length; i++){
                password = password + chars[i];
            }
            //System.out.println("Email: " + email + " Password: " + password);

            //Check if username exists
            String databaseURL = "jdbc:derby:SALARY_GENERATOR";

```

```

Connection con = DriverManager.getConnection(databaseURL);

String sql;
Statement st = con.createStatement();

sql = "SELECT * FROM users WHERE email='" + email + "'";
ResultSet rs = st.executeQuery(sql);

//db stands for database
String dbid = "", dbemail = "", dbpassword = "";

while (rs.next()==true){
    System.out.println("Reaches here");
    dbid = rs.getString(1);
    dbemail = rs.getString(2);
    dbpassword = rs.getString(3);
}

//      System.out.println("Dbid: " + dbid + " dbemail: " + dbemail + " dbpassword: " +
dbpassword);
    if (password.length()==0){
        JOptionPane.showMessageDialog(null, "Password field empty. Please enter a
password.");
    }
    else if ("".equals(dbemail)){
        JOptionPane.showMessageDialog(null, "Email does not exist or was typed
incorrectly. If it's the former, please register.");
    }
    else{
        if (!(dbpassword.equals(password))){
            JOptionPane.showMessageDialog(null, "Password is incorrect.");
        }
        else{
            //Continue to the main menu
            new SalaryGenerator(dbid,dbemail).setVisible(true);
            this.setVisible(false);
        }
    }
}

}

}

}
catch (Exception e){
    e.printStackTrace();
}

```

```

    }
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    }
}
//</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Login().setVisible(true);
        }
    }
}

```

```
    });  
}  
  
// Variables declaration - do not modify  
private javax.swing.JCheckBox btnShowPassword;  
private com.k33ptoo.components.KButton kbtnRegister;  
private com.k33ptoo.components.KButton kbtnSignIn;  
private javax.swing.JLabel lblEmail;  
private javax.swing.JLabel lblPassword;  
private javax.swing.JLabel lblWelcome;  
private com.k33ptoo.components.KGradientPanel pnlBackground;  
private javax.swing.JPanel pnlLogin;  
private javax.swing.JTextField txtEmail;  
private javax.swing.JPasswordField txtPassword;  
// End of variables declaration  
}
```

Sign Up Page:

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */

/**
 *
 * @author User
 */
import java.awt.Color;
import java.io.File;
import java.sql.Connection;
import java.sql.DatabaseMetaData;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

public class Register extends javax.swing.JFrame {

    /**
     * Creates new form Register
     */
    private void initComponents2(){
        //Overwrite Java Driver Code
        txtEmail.setBackground(new Color(0, 0, 0, 0));
        txtPassword.setBackground(new Color(0, 0, 0, 0));
        txtConfirm.setBackground(new Color(0, 0, 0, 0));
    }

    int max = 0;

    private void maxfinder(){ //figure out the highest eid
        try{
            String databaseURL = "jdbc:derby:SALARY_GENERATOR";
            Connection con = DriverManager.getConnection(databaseURL);
```

```

String sql;
Statement st = con.createStatement();

sql = "SELECT * FROM users";
ResultSet rs = st.executeQuery(sql);

//db stands for database

int max = 0;
while (rs.next()==true){
    String dbeid = rs.getString(1);
    int eid = Integer.parseInt(dbeid);
    if (eid>max){
        max = eid;
    }
}
this.max = max;
}
catch (Exception e){

}
}

```

```

public Register() {
    initComponents();

    try{
        initComponents2();
        maxfinder();
    }
    catch (Exception e){

    }

}

```

```

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")

```

```
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel1 = new javax.swing.JLabel();
    pnlBackground = new com.k33ptoo.components.KGradientPanel();
    pnlRegister = new javax.swing.JPanel();
    txtEmail = new javax.swing.JTextField();
    lblRegister = new javax.swing.JLabel();
    lblEmail = new javax.swing.JLabel();
    lblPassword = new javax.swing.JLabel();
    txtPassword = new javax.swing.JPasswordField();
    btnShowPassword = new javax.swing.JCheckBox();
    kbtnRegister = new com.k33ptoo.components.KButton();
    kbtnBack = new com.k33ptoo.components.KButton();
    lblConfirm = new javax.swing.JLabel();
    btnShowConfirm = new javax.swing.JCheckBox();
    txtConfirm = new javax.swing.JPasswordField();

    jLabel1.setText("jLabel1");

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setBackground(new java.awt.Color(255, 255, 255));

    pnlBackground.setkEndColor(new java.awt.Color(29, 38, 113));
    pnlBackground.setkGradientFocus(0);
    pnlBackground.setkStartColor(new java.awt.Color(161, 196, 253));

    pnlRegister.setBackground(new java.awt.Color(248, 233, 161));
    pnlRegister.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
    pnlRegister.setForeground(new java.awt.Color(36, 48, 94));

    txtEmail.setBackground(new java.awt.Color(0, 0, 0));
    txtEmail.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
    txtEmail.setForeground(new java.awt.Color(33, 30, 39));
    txtEmail.setBorder(javax.swing.BorderFactory.createMatteBorder(0, 0, 2, 0, new
java.awt.Color(33, 30, 39)));
    txtEmail.setMargin(new java.awt.Insets(2, 2, 2, 2));
    txtEmail.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            txtEmailActionPerformed(evt);
        }
    });
}
```

```

lblRegister.setBackground(new java.awt.Color(252, 250, 255));
lblRegister.setFont(new java.awt.Font("Segoe UI Light", 0, 36)); // NOI18N
lblRegister.setForeground(new java.awt.Color(33, 30, 39));
lblRegister.setText("Register");
lblRegister.setToolTipText("");
lblRegister.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);

lblEmail.setBackground(new java.awt.Color(252, 250, 255));
lblEmail.setFont(new java.awt.Font("Segoe UI Light", 0, 24)); // NOI18N
lblEmail.setForeground(new java.awt.Color(33, 30, 39));
lblEmail.setText("Email:");

lblPassword.setBackground(new java.awt.Color(252, 250, 255));
lblPassword.setFont(new java.awt.Font("Segoe UI Light", 0, 24)); // NOI18N
lblPassword.setForeground(new java.awt.Color(33, 30, 39));
lblPassword.setText("Password:");

txtPassword.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtPassword.setForeground(new java.awt.Color(33, 30, 39));
txtPassword.setBorder(javax.swing.BorderFactory.createMatteBorder(0, 0, 2, 0, new
java.awt.Color(33, 30, 39)));
txtPassword.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtPassword.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtPasswordActionPerformed(evt);
    }
});

btnShowPassword.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
btnShowPassword.setForeground(new java.awt.Color(33, 30, 39));
btnShowPassword.setText("Show Password");
btnShowPassword.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnShowPasswordActionPerformed(evt);
    }
});

kbtnRegister.setBorder(null);
kbtnRegister.setText("Register");
kbtnRegister.setFont(new java.awt.Font("Segoe UI Light", 1, 18)); // NOI18N
kbtnRegister.setkBorderRadius(50);
kbtnRegister.setkEndColor(new java.awt.Color(36, 48, 94));
kbtnRegister.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnRegister.setkHoverForeGround(new java.awt.Color(255, 255, 255));

```



```

kbtnRegister.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnRegister.setkIndicatorThickness(5);
kbtnRegister.setkStartColor(new java.awt.Color(36, 48, 94));
kbtnRegister.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnRegisterActionPerformed(evt);
    }
});

kbtnBack.setBorder(null);
kbtnBack.setText("Back");
kbtnBack.setFont(new java.awt.Font("Segoe UI Light", 1, 18)); // NOI18N
kbtnBack.setkBorderRadius(50);
kbtnBack.setkEndColor(new java.awt.Color(36, 48, 94));
kbtnBack.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnBack.setkHoverForeground(new java.awt.Color(255, 255, 255));
kbtnBack.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnBack.setkIndicatorThickness(5);
kbtnBack.setkStartColor(new java.awt.Color(36, 48, 94));
kbtnBack.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnBackActionPerformed(evt);
    }
});

lblConfirm.setBackground(new java.awt.Color(252, 250, 255));
lblConfirm.setFont(new java.awt.Font("Segoe UI Light", 0, 24)); // NOI18N
lblConfirm.setForeground(new java.awt.Color(33, 30, 39));
lblConfirm.setText("Confirm:");

btnShowConfirm.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
btnShowConfirm.setForeground(new java.awt.Color(33, 30, 39));
btnShowConfirm.setText("Show Password");
btnShowConfirm.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnShowConfirmActionPerformed(evt);
    }
});

txtConfirm.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtConfirm.setForeground(new java.awt.Color(33, 30, 39));
txtConfirm.setBorder(javax.swing.BorderFactory.createMatteBorder(0, 0, 2, 0, new
java.awt.Color(33, 30, 39)));
txtConfirm.setMargin(new java.awt.Insets(2, 2, 2, 2));

```

```
txtConfirm.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        txtConfirmActionPerformed(evt);  
    }  
});  
  
javax.swing.GroupLayout pnlRegisterLayout = new javax.swing.GroupLayout(pnlRegister);  
pnlRegister.setLayout(pnlRegisterLayout);  
pnlRegisterLayout.setHorizontalGroup(  
    pnlRegisterLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
        .addGroup(pnlRegisterLayout.createSequentialGroup()  
.addGroup(pnlRegisterLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
G)  
            .addGroup(pnlRegisterLayout.createSequentialGroup()  
                .addGap(37, 37, 37)  
                .addComponent(btnRegister, javax.swing.GroupLayout.PREFERRED_SIZE,  
140, javax.swing.GroupLayout.PREFERRED_SIZE)  
                .addPreferredGap(LayoutStyle.ComponentPlacement.RELATED)  
                .addComponent(btnBack, javax.swing.GroupLayout.PREFERRED_SIZE, 140,  
javax.swing.GroupLayout.PREFERRED_SIZE))  
            .addGroup(pnlRegisterLayout.createSequentialGroup()  
                .addGap(53, 53, 53)  
.addGroup(pnlRegisterLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING  
G)  
            .addComponent(txtEmail, javax.swing.GroupLayout.PREFERRED_SIZE, 260,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
            .addComponent(lblEmail, javax.swing.GroupLayout.PREFERRED_SIZE, 159,  
javax.swing.GroupLayout.PREFERRED_SIZE))  
.addGroup(pnlRegisterLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING  
NG, false)  
            .addGroup(javax.swing.GroupLayout.Alignment.LEADING,  
pnlRegisterLayout.createSequentialGroup()  
                .addComponent(lblConfirm)  
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,  
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)  
                .addComponent(btnShowConfirm))  
            .addComponent(txtConfirm, javax.swing.GroupLayout.Alignment.LEADING,  
javax.swing.GroupLayout.PREFERRED_SIZE, 260,  
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```

.addGroup(pnlRegisterLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
    .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
pnlRegisterLayout.createSequentialGroup()
    .addComponent(lblPassword)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(btnShowPassword))
    .addComponent(txtPassword,
javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED_SIZE,
260, javax.swing.GroupLayout.PREFERRED_SIZE))))
    .addContainerGap(45, Short.MAX_VALUE))
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlRegisterLayout.createSequentialGroup()
    .addGap(0, 0, Short.MAX_VALUE)
    .addComponent(lblRegister)
    .addGap(119, 119, 119))
);
pnlRegisterLayout.setVerticalGroup(
pnlRegisterLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(pnlRegisterLayout.createSequentialGroup()
    .addGap(12, 12, 12)
    .addComponent(lblRegister)
    .addGap(18, 18, 18)
    .addComponent(lblEmail, javax.swing.GroupLayout.PREFERRED_SIZE, 32,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(txtEmail, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(pnlRegisterLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblPassword, javax.swing.GroupLayout.PREFERRED_SIZE, 32,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(btnShowPassword))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(txtPassword, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

```

```

.addGroup(pnlRegisterLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblConfirm, javax.swing.GroupLayout.PREFERRED_SIZE, 32,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(btnShowConfirm))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(txtConfirm, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 69,
Short.MAX_VALUE)

.addGroup(pnlRegisterLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(kbtnBack, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(kbtnRegister, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(25, 25, 25))
);

javax.swing.GroupLayout pnlBackgroundLayout = new
javax.swing.GroupLayout(pnlBackground);
pnlBackground.setLayout(pnlBackgroundLayout);
pnlBackgroundLayout.setHorizontalGroup(

pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(pnlBackgroundLayout.createSequentialGroup())
    .addContainerGap(393, Short.MAX_VALUE)
    .addComponent(pnlRegister, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(427, 427, 427))
);
pnlBackgroundLayout.setVerticalGroup(

pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(pnlBackgroundLayout.createSequentialGroup())
    .addGap(93, 93, 93)
    .addComponent(pnlRegister, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addContainerGap(134, Short.MAX_VALUE))
);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

```

```

        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(pnlBackground, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(pnlBackground, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        );

        pack();
        setLocationRelativeTo(null);
    }// </editor-fold>

```

```

private void txtEmailActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

```

```

private void txtPasswordActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

```

```

private void btnShowPasswordActionPerformed(java.awt.event.ActionEvent evt) {
    try{ //Show Password
        if (btnShowPassword.isSelected()==false){
            txtPassword.setEchoChar('*');
        }
        else if (btnShowPassword.isSelected()==true){
            txtPassword.setEchoChar((char)0);
        }
    }
    catch (Exception e){

    }
}

```

```

private void kbtnRegisterActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        String email = txtEmail.getText().trim().toUpperCase();
        //Initializing variables
        int atcount = 0;
        int atpos = 0;
    }
}

```

```

int firstdotpos = -1;
int dotpos = 0;
int finaldotpos = 0;
boolean invalidchars = false;

//Figuring out the positions of important characters
for (int i = 0; i<email.length(); i++){
    char character = email.charAt(i);
    //find position of the first at symbol
    if (character=='@'){
        atcount++; //count the nubmer of @ symbols
        atpos = i;
    }
    if (character=='.' && atpos!=0 && dotpos==0){
        dotpos = i; //the first . after the email
    }
    if (character=='.' && atpos==0 && firstdotpos==-1){
        firstdotpos = i; //the first . before the email
    }
    if (character=='.' && atpos!=0 && dotpos!=0){
        finaldotpos = i; //the final . symbol
    }
    int ascii = (int)character;
    if (ascii<=45 || ascii==47 || (ascii>=58 && ascii<=63) || ascii>=91){
        invalidchars = true; //check that all the ascii characters are valid
    }
}

//Email Error Trapping
if (email.length()==0){
    JOptionPane.showMessageDialog(null, "Email Field Empty. Please enter an email.");
}
else if (firstdotpos==0){
    JOptionPane.showMessageDialog(null, "Invalid Email. The '.' cannot be at the first
position.");
}
else if (finaldotpos == email.length()-1){
    JOptionPane.showMessageDialog(null, "Invalid Email. A '.' cannot be at the final
position.");
}
else if (atcount==0){
    JOptionPane.showMessageDialog(null, "Invalid Email. The email is missing an '@'
symbol.");
}

```

```

        else if (atcount>=2){
            JOptionPane.showMessageDialog(null, "Invalid Email. The email has too many '@'
symbols.");
        }
        else if (atpos==0){
            JOptionPane.showMessageDialog(null, "Invalid Email. The email is lacking a prefix
before the '@' symbol.");
        }
        else if (dotpos - atpos <=1){
            JOptionPane.showMessageDialog(null, "Invalid Email. The email is lacking a valid
domain name.");
        }
        else if (invalidchars==true){
            JOptionPane.showMessageDialog(null, "Invalid Email. The email contains invalid
characters.");
        }
        else if (email.length()>100){
            JOptionPane.showMessageDialog(null, "Invalid Email. The email is longer than 100
characters.");
        }
        else{
            char chars[] = txtPassword.getPassword();
            String password = "";
            for (int i = 0; i<chars.length; i++){
                password = password + chars[i];
            }

            chars = txtConfirm.getPassword();
            String confirm = "";
            for (int i = 0; i<chars.length; i++){
                confirm = confirm + chars[i];
            }

//            System.out.println("Email: " + email + " Password: " + password);

            //Check if username exists
            String databaseURL = "jdbc:derby:SALARY_GENERATOR";
            Connection con = DriverManager.getConnection(databaseURL);

            String sql;
            Statement st = con.createStatement();

            sql = "SELECT * FROM users WHERE email='" + email + "'";

```

```

ResultSet rs = st.executeQuery(sql);

//db stands for database
String dbid = "", dbemail = "", dbpassword = "";

while (rs.next()==true){
    dbid = rs.getString(1);
    dbemail = rs.getString(2);
    dbpassword = rs.getString(3);
}

con.close();

if (password.length()==0){
    JOptionPane.showMessageDialog(null, "Password field empty. Please enter a
password.");
}
else if (confirm.length()==0){
    JOptionPane.showMessageDialog(null, "Confirm field empty. Please confirm the
password.");
}
else if (password.length()>100){
    JOptionPane.showMessageDialog(null, "Chosen password must not exceed 100
characters.");
}
else if ("".equals(dbemail)){
    //Email not in database yet
    if (!(password.equals(confirm))){
        JOptionPane.showMessageDialog(null, "Passwords do not match.");
    }
    else{
        //Continue to Login page
        new Login().setVisible(true);
        this.setVisible(false);

        databaseURL = "jdbc:derby:SALARY_GENERATOR";
        con = DriverManager.getConnection(databaseURL);
        st = con.createStatement();

        int max = this.max;
        max++;
        this.max = max;
    }
}

```



```

        sql = "INSERT INTO users(eid,email,password)
VALUES("+Integer.toString(max)+"," + email + "," + password + ")";
        st.execute(sql);
        con.close();

    }

}

else{
    JOptionPane.showMessageDialog(null, "Email already exists in the database.
Please sign up instead.");
}
}

}

catch (Exception e){
    e.printStackTrace();
}
}

private void kbtnBackActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        new Login().setVisible(true);
        this.setVisible(false);
    }
    catch (Exception e){

    }

}

private void btnShowConfirmActionPerformed(java.awt.event.ActionEvent evt) {
    try{ //Show Password
        if (btnShowConfirm.isSelected()==false){
            txtConfirm.setEchoChar('*');
        }
        else if (btnShowConfirm.isSelected()==true){
            txtConfirm.setEchoChar((char)0);
        }
    }
    catch (Exception e){

    }
}

```

```

private void txtConfirmActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {

```

```

        new Register().setVisible(true);
    }
    });
}

// Variables declaration - do not modify
private javax.swing.JCheckBox btnShowConfirm;
private javax.swing.JCheckBox btnShowPassword;
private javax.swing.JLabel jLabel1;
private com.k33ptoo.components.KButton kbtnBack;
private com.k33ptoo.components.KButton kbtnRegister;
private javax.swing.JLabel lblConfirm;
private javax.swing.JLabel lblEmail;
private javax.swing.JLabel lblPassword;
private javax.swing.JLabel lblRegister;
private com.k33ptoo.components.KGradientPanel pnlBackground;
private javax.swing.JPanel pnlRegister;
private javax.swing.JPasswordField txtConfirm;
private javax.swing.JTextField txtEmail;
private javax.swing.JPasswordField txtPassword;
// End of variables declaration
}

```

Salary Generator Page:

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */

/**
 *
 * @author User
 */
import java.awt.Color;
import java.io.File;
import java.io.IOException;
import java.sql.Connection;
import java.sql.DatabaseMetaData;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.PreparedStatement;
import java.text.MessageFormat;
import java.time.LocalDate;
import javax.swing.JOptionPane;
import javax.swing.RowFilter;
import javax.swing.SwingUtilities;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableModel;
import javax.swing.table.TableRowSorter;

import java.util.*;
import javax.mail.*;
import javax.mail.internet.*;

import javax.swing.text.*;

import org.apache.pdfbox.pdmodel.PDDocument;
import org.apache.pdfbox.pdmodel.PDPage;
import org.apache.pdfbox.pdmodel.PDPageContentStream;
import org.apache.pdfbox.pdmodel.font.PDType1Font;
import static org.apache.pdfbox.pdmodel.font.Standard14Fonts.FontName.HELVETICA;
```

```

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileWriter;
import java.io.InputStream;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.Scanner;
import java.util.logging.Level;
import java.util.logging.Logger;

import javax.print.Doc;
import javax.print.DocFlavor;
import javax.print.DocPrintJob;
import javax.print.PrintException;
import javax.print.PrintService;
import javax.print.PrintServiceLookup;
import javax.print.SimpleDoc;
import javax.print.attribute.HashPrintRequestAttributeSet;
import javax.print.attribute.PrintRequestAttributeSet;
import javax.print.attribute.standard.Copies;
import javax.print.attribute.standard.Finishings;
import javax.print.attribute.standard.MediaSizeName;
import javax.print.attribute.standard.NumberUp;
import javax.print.attribute.standard.OrientationRequested;
import javax.print.attribute.standard.Sides;
import javax.print.event.PrintJobAdapter;
import javax.print.event.PrintJobEvent;
import javax.swing.JTable;

```

```

public class SalaryGenerator extends javax.swing.JFrame {

```

```

    /**
     * Creates new form Register
     */

```

```

    //Global Variables
    int max = 0;
    int payperiod = -1;
    String email = "", uid = "";
    String selectedroweid = "-1";
    //Setting up the regexfilter
    TableRowSorter sorter;

```

```

private void calcFinalWage(){
    //Try-catch makes it so that if there's any mistake, the final wage doesn't update its textfield
    try{
        //retrieves necessary values from the textboxes
        double wage = Double.parseDouble(txtWage.getText());
        double sssee = Double.parseDouble(txtSSSEE.getText());
        double phее = Double.parseDouble(txtPHEE.getText());
        double hdmfee = Double.parseDouble(txtHDMFEE.getText());
        double sub = Double.parseDouble(txtLoanSub.getText());
        double deduction = Double.parseDouble(txtDeduction.getText());
        double bonus = Double.parseDouble(txtBonus.getText());

        //applying the calculation dynamically
        double n = wage - sssee - (phее*wage/100) - (hdmfee*wage/100) - sub - deduction +
bonus;
        String strn = Double.toString(n);

        //Updating the textbox
        txtFinalWage.setText(strn);
    }
    catch(Exception e){
        e.printStackTrace();
    }
}

private void payperiodfinder(){ //figure out the highest pay period
    try{
        String databaseURL = "jdbc:derby:SALARY_GENERATOR";
        Connection con = DriverManager.getConnection(databaseURL);

        String sql;
        Statement st = con.createStatement();

        sql = "SELECT * FROM salary_allocations";
        ResultSet rs = st.executeQuery(sql);

        //db stands for database

        int max = 0;
        while (rs.next()==true){
            String dbeid = rs.getString(3);
            int eid = Integer.parseInt(dbeid);

```

```

        if (eid>max){
            max = eid;
        }
    }
    this.payperiod = max;
    con.close();
}
catch (Exception e){
    e.printStackTrace();
}
}

private void initComponents2(){
    try {
        //Overwrite Java Driver Code

        //Display the email used to login, truncated if too long
        if (email.length()<25){
            lblEmail.setText(email);
        }
        else{
            lblEmail.setText(email.substring(0,20)+"...");
        }

        //Initializing the table with specific settings
        //For the regexfilter later on
        TableRowSorter sorter = new
TableRowSorter<>((DefaultTableModel)tblEmployees.getModel());
        this.sorter = sorter;
        tblEmployees.setRowSorter(sorter);
        //Disables table sorting
        sorter.setSortable(0, false);
        sorter.setSortable(1, false);
        sorter.setSortable(2, false);
        sorter.setSortable(3, false);
        //Disables rearranging table columns
        tblEmployees.getTableHeader().setReorderingAllowed(false);

        //Textboxes set to a clear background
        /*txtFullName.setBackground(new Color(0, 0, 0, 0));
        txtEmail.setBackground(new Color(0, 0, 0, 0));
        txtPhoneNumber.setBackground(new Color(0, 0, 0, 0));*/
    }
}

```

```

//Numeric text fields
((AbstractDocument) txtEID.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
((AbstractDocument) txtWage.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
((AbstractDocument) txtSSSEE.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
((AbstractDocument) txtSSSER.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
((AbstractDocument) txtPHEE.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
((AbstractDocument) txtPHER.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
((AbstractDocument) txtHDMFEE.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
((AbstractDocument) txtHDMFER.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
((AbstractDocument) txtLoan.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
((AbstractDocument) txtLoanSub.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
((AbstractDocument) txtLoanAdd.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
((AbstractDocument) txtBonus.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
((AbstractDocument) txtDeduction.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
// ((AbstractDocument) txtFinalWage.getDocument()).setDocumentFilter(new
NumericDocumentFilter());

```

```

String maxeid = Integer.toString(this.max);
txtEID.setText(maxeid);

```

```

txtSSSEE.setText("0");
txtSSSER.setText("0");
txtPHEE.setText("0");
txtPHER.setText("0");
txtHDMFEE.setText("0");
txtHDMFER.setText("0");

```



```

//Initialize comboboxes
String databaseURL = "jdbc:derby:SALARY_GENERATOR";
Connection con = DriverManager.getConnection(databaseURL);

//Pulling data from SALARY_GENERATOR Database based on the added SSS Values
String sql;
Statement st = con.createStatement();
ResultSet rs = st.executeQuery("SELECT bracketlowerbound, bracketupperbound
FROM sss");

//Iterating through the resultset and adding the choices sequentially
cmbSSSBracket.addItem("-SELECT SSS-");
while (rs.next()) {
    double lowerBound = rs.getDouble("bracketlowerbound");
    double upperBound = rs.getDouble("bracketupperbound");
    cmbSSSBracket.addItem(lowerBound + " - " + upperBound);
}

//Pulling data from SALARY_GENERATOR Database based on the added PH Values
rs = st.executeQuery("SELECT bracketlowerbound, bracketupperbound FROM
philhealth");

//Iterating through the resultset and adding the choices sequentially
cmbPHBracket.addItem("-SELECT PH-");
while (rs.next()) {
    double lowerBound = rs.getDouble("bracketlowerbound");
    double upperBound = rs.getDouble("bracketupperbound");
    cmbPHBracket.addItem(lowerBound + " - " + upperBound);
}

//Pulling data from SALARY_GENERATOR Database based on the added HDMF Values
rs = st.executeQuery("SELECT bracketlowerbound, bracketupperbound FROM hdmf");

//Iterating through the resultset and adding the choices sequentially
cmbHDMFBracket.addItem("-SELECT HDMF-");
while (rs.next()) {
    double lowerBound = rs.getDouble("bracketlowerbound");
    double upperBound = rs.getDouble("bracketupperbound");
    cmbHDMFBracket.addItem(lowerBound + " - " + upperBound);
}
con.close();
}

```

```

    catch (Exception e) {
        e.printStackTrace();
    }
}

public static boolean doesTableExists(String tableName, Connection conn)
    throws SQLException
{
    DatabaseMetaData meta = conn.getMetaData();
    ResultSet result = meta.getTables(null, null, tableName.toUpperCase(), null);

    return result.next();
}

private void maxfinder(){ //figure out the highest eid
    try{
        //Connect to the database
        String databaseURL = "jdbc:derby:SALARY_GENERATOR";
        Connection con = DriverManager.getConnection(databaseURL);

        String sql;
        Statement st = con.createStatement();

        sql = "SELECT * FROM salary_allocations";
        ResultSet rs = st.executeQuery(sql);

        //db stands for database

        //Have a max variable begin at 0
        int max = 0;
        while (rs.next()==true){
            String dbeid = rs.getString(1);
            int eid = Integer.parseInt(dbeid);
            if (eid>max){ //change max with the value in the ResultSet if the value is larger
                max = eid;
            }
        }
        //assign it to a global max variable to be used later.
        //Incrementing this.max by 1 guarantees a unique max value
        this.max = max;
        con.close();
    }
    catch (Exception e){

```

```

    }
}

private void search(){
    try{
        int in = cmbIn.getSelectedIndex();

        String search = txtSearch.getText().trim();

        //Search using RowFilter and regex (regular expression)
        RowFilter<DefaultTableModel, Object> rf = null;
        try {
            if (in == 0){
                rf = RowFilter.regexFilter("(?i)" + search); //("(?i)" for case insensitive filter
            }
            else{
                //to search a specific column
                rf = RowFilter.regexFilter("(?i)" + search, in - 1);
            }
        }
        catch (java.util.regex.PatternSyntaxException e) {
            //      System.out.println("Error Found");
            return;
        }
        //The set RowFilter only shows table values that satisfy the search parameters
        sorter.setRowFilter(rf);
    }
    catch (Exception e){
        e.printStackTrace();
    }
}

private void refreshtable(String sql){//reset table contents, works for SELECT queries only
    try{
        //Creating a defaulttablemodel for tblEmployees
        DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();

        //Connect to SALARY_GENERATOR database
        String databaseURL = "jdbc:derby:SALARY_GENERATOR";
        Connection con = DriverManager.getConnection(databaseURL);
        Statement st = con.createStatement();

        //Execute the SQL query
        ResultSet rs = st.executeQuery(sql);
    }
}

```

```

model.setRowCount(0); //reset the table
int i = 0;

Double total_wages = 0.0;
Double total_er = 0.0;
//Iterate through resultset like it's a collection
while (rs.next()==true){
    i++;
    String eid = rs.getString(2); // EID
    String wage = rs.getString(4); // wage
    String sss_ee = rs.getString(5); // sss_ee
    String sss_er = rs.getString(6); // sss_er
    String ph_ee_p = rs.getString(7); // ph_ee_p
    String ph_er_p = rs.getString(8); // ph_er_p
    String hdmf_ee_p = rs.getString(9); // hdmf_ee_p
    String hdmf_er_p = rs.getString(10); // hdmf_er_p
    String loan = rs.getString(11); // loan
    String loan_add = rs.getString(12); // loan_add
    String loan_sub = rs.getString(13); // loan_sub
    String deduction = rs.getString(14); // deduction
    String bonus = rs.getString(15); // bonus
    String fullname = rs.getString(16); //fullname

    Double dwage = Double.parseDouble(wage);
    Double dsss_ee = Double.parseDouble(sss_ee);
    Double dsss_er = Double.parseDouble(sss_er);
    Double dph_ee_p = Double.parseDouble(ph_ee_p);
    Double dph_er_p = Double.parseDouble(ph_er_p);
    Double dhdmf_ee_p = Double.parseDouble(hdmf_ee_p);
    Double dhdmf_er_p = Double.parseDouble(hdmf_er_p);
    Double dloan = Double.parseDouble(loan);
    Double dloan_add = Double.parseDouble(loan_add);
    Double dloan_sub = Double.parseDouble(loan_sub);
    Double ddeduction = Double.parseDouble(deduction);
    Double dbonus = Double.parseDouble(bonus);

    //Calculate dynamic values
    Double final_wage = dwage - dsss_ee - (dph_ee_p/100*dwage) -
(dhdmf_ee_p/100*dwage) - dloan_sub - ddeduction + dbonus;
    Double er = dsss_er + (dph_er_p/100*dwage) + (dhdmf_er_p/100*dwage);
    total_wages+=final_wage;
    total_er+=er;
}

```

```

        //Add values to the jTable
        model.addRow(new Object[]{eid, fullname, wage, sss_ee, sss_er,
            ph_ee_p, ph_er_p, hdmf_ee_p, hdmf_er_p, loan, loan_add, loan_sub, deduction,
bonus, final_wage});
    }

    //Update the appropriate text boxes with calculated dynamic values
    String strTotalWages = total_wages.toString();
    txtTotalWages.setText(strTotalWages);

    String strTotalER = total_er.toString();
    txtTotalER.setText(strTotalER);

    con.close();

    String stri = Integer.toString(i);
    txtActiveEmployees.setText(stri);

}
catch (Exception e){
    e.printStackTrace();
}

}

private static void writeDateToFile(String filename, String date) {
    try {
        FileWriter writer = new FileWriter(filename);
        writer.write(date);
        writer.close();
    } catch (IOException e) {
        System.out.println("An error occurred while writing to the file.");
        e.printStackTrace();
    }
}

public SalaryGenerator() {
    initComponents();
    try{
        this.email = email;
        String currentURL = "jdbc:derby:SALARY_GENERATOR;create=true";
        Connection con = DriverManager.getConnection(currentURL);

```

```

Statement st = con.createStatement();

//      System.out.println("Reaches here");

uid = "salary_allocation"; //table names can't start with a number
this.uid = uid;
//      System.out.println(uid);

if (!doesTableExists("info", con)){
    String sql = "CREATE TABLE info "
        + "(eid integer PRIMARY KEY, userid integer DEFAULT NULL, hired boolean, "
        + "first_name varchar(50), middle_name varchar(50), "
        + "last_name varchar(50), age integer, birthdate date, "
        + "address varchar(100), phonenumber varchar(50), email varchar(50) DEFAULT
NULL,"
        + "height integer DEFAULT NULL, weight integer DEFAULT NULL, notes
varchar(200) DEFAULT NULL)";
    st.execute(sql);
}

if (!doesTableExists("salary_allocations", con)){
    String sql = "CREATE TABLE salary_allocations "
        + "(sid integer PRIMARY KEY, eid integer, pay_period integer,"
        + "monthly_wage_rate decimal(8,2), sss_ee decimal(8,2)," //maximum bounds
for currency values
        + "sss_er decimal(8,2), ph_ee_p decimal(8,2), ph_er_p decimal(8,2)," //8 whole
number digits, 2 decimal places
        + "hdmf_ee_p decimal(8,2), hdmf_er_p decimal(8,2), loan decimal(8,2), "
        + "loan_add decimal(8,2), loan_sub decimal(8,2), deduction decimal(8,2),"
        + "bonus decimal(8,2))";
    st.execute(sql);
}

if (!doesTableExists("sss", con)){
    String sql = "CREATE TABLE sss (bid integer PRIMARY KEY, bracketlowerbound
decimal(8,2), bracketupperbound decimal(8,2), sss_ee decimal(8,2), sss_er decimal(8,2))";
    st.execute(sql);
}

if (!doesTableExists("philhealth", con)){
    String sql = "CREATE TABLE philhealth (bid integer PRIMARY KEY,
bracketlowerbound decimal(8,2), bracketupperbound decimal(8,2), ph_ee_p decimal(8,2),
ph_er_p decimal(8,2))";

```

```

        st.execute(sql);
    }

    if (!doesTableExists("hdmf", con)){
        String sql = "CREATE TABLE hdmf (bid integer PRIMARY KEY, bracketlowerbound
decimal(8,2), bracketupperbound decimal(8,2), hdmf_ee_p decimal(8,2), hdmf_er_p
decimal(8,2))";
        st.execute(sql);
    }

    if (!doesTableExists("salary_history", con)){
        String sql = "CREATE TABLE salary_history (shid integer PRIMARY KEY, pay_period
integer, eid integer, datesaved date, monthly_wage_rate decimal(8,2),"
+ "sss_ee decimal(8,2), sss_er decimal(8,2), ph_ee_p decimal(8,2), ph_er_p
decimal(8,2), hdmf_ee_p decimal(8,2), hdmf_er_p decimal(8,2), "
+ "deduction decimal(8,2), bonus decimal(8,2), final_wage decimal(8,2), final_er
decimal(8,2))";
        st.execute(sql);
    }

    if (!doesTableExists("loan_history", con)){
        String sql = "CREATE TABLE loan_history (lhid integer PRIMARY KEY, pay_period
integer, eid integer, datesaved date, loan decimal(8,2), "
+ "loan_add decimal(8,2), loan_sub decimal(8,2), final_loan decimal(8,2))";
        st.execute(sql);
    }
    con.close();

    maxfinder();

    refreshtable("SELECT sa.*, " +
        "i.last_name || ' ' || i.first_name || ' ' || COALESCE(i.middle_name, '') AS full_name\n"
+
        "FROM salary_allocations sa\n" +
        "JOIN info i ON sa.eid = i.eid\n" +
        "WHERE i.hired = TRUE");

    initComponents2();

}
catch (Exception e){
    e.printStackTrace();
    JOptionPane.showMessageDialog(null, "Error: " + e.getMessage());
}

```

```

    }
}

// Email credentials

public SalaryGenerator(String uid, String email){ //CATCH ALL INCOMING DATA
    initComponents();
    try{
        this.email = email;
        String currentURL = "jdbc:derby:SALARY_GENERATOR;create=true";
        Connection con = DriverManager.getConnection(currentURL);

        Statement st = con.createStatement();

        uid = "salary_allocation";
        this.uid = uid;

        if (!doesTableExists("info", con)){
            String sql = "CREATE TABLE info "
                + "(eid integer PRIMARY KEY, userid integer DEFAULT NULL, hired boolean, "
                + "first_name varchar(50), middle_name varchar(50), "
                + "last_name varchar(50), age integer, birthdate date, "
                + "address varchar(100), phonenummer varchar(50), email varchar(50) DEFAULT
NULL,"
                + "height integer DEFAULT NULL, weight integer DEFAULT NULL, notes
varchar(200) DEFAULT NULL)";
            st.execute(sql);
        }

        if (!doesTableExists("salary_allocations", con)){
            String sql = "CREATE TABLE salary_allocations "
                + "(sid integer PRIMARY KEY, eid integer, pay_period integer,"
                + "monthly_wage_rate decimal(8,2), sss_ee decimal(8,2),"
                + "sss_er decimal(8,2), ph_ee_p decimal(8,2), ph_er_p decimal(8,2),"
                + "hdmf_ee_p decimal(8,2), hdmf_er_p decimal(8,2), loan decimal(8,2), "
                + "loan_add decimal(8,2), loan_sub decimal(8,2), deduction decimal(8,2),"
                + "bonus decimal(8,2))";
            st.execute(sql);
        }

        if (!doesTableExists("sss", con)){

```



```

        String sql = "CREATE TABLE sss (bid integer PRIMARY KEY, bracketlowerbound
decimal(8,2), bracketupperbound decimal(8,2), sss_ee decimal(8,2), sss_er decimal(8,2))";
        st.execute(sql);
    }

    if (!doesTableExists("philhealth", con)){
        String sql = "CREATE TABLE philhealth (bid integer PRIMARY KEY,
bracketlowerbound decimal(8,2), bracketupperbound decimal(8,2), ph_ee_p decimal(8,2),
ph_er_p decimal(8,2))";
        st.execute(sql);
    }

    if (!doesTableExists("hdmf", con)){
        String sql = "CREATE TABLE hdmf (bid integer PRIMARY KEY, bracketlowerbound
decimal(8,2), bracketupperbound decimal(8,2), hdmf_ee_p decimal(8,2), hdmf_er_p
decimal(8,2))";
        st.execute(sql);
    }

    if (!doesTableExists("salary_history", con)){
        String sql = "CREATE TABLE salary_history (shid integer PRIMARY KEY, pay_period
integer, eid integer, datesaved date, monthly_wage_rate decimal(8,2),"
        + "sss_ee decimal(8,2), sss_er decimal(8,2), ph_ee_p decimal(8,2), ph_er_p
decimal(8,2), hdmf_ee_p decimal(8,2), hdmf_er_p decimal(8,2), "
        + "deduction decimal(8,2), bonus decimal(8,2), final_wage decimal(8,2), final_er
decimal(8,2))";
        st.execute(sql);
    }

    if (!doesTableExists("loan_history", con)){
        String sql = "CREATE TABLE loan_history (lhid integer PRIMARY KEY, pay_period
integer, eid integer, datesaved date, loan decimal(8,2), "
        + "loan_add decimal(8,2), loan_sub decimal(8,2), final_loan decimal(8,2))";
        st.execute(sql);
    }

    //Email sending code
    String lastSentDateString = null;
    try {
        // Read the last sent date from the file
        File file = new File("filename.txt");
        if (file.exists()) {
            Scanner myReader = new Scanner(file);

```

```

        if (myReader.hasNextLine()) {
            lastSentDateString = myReader.nextLine();
        }
        myReader.close();
    }
} catch (FileNotFoundException e) {
    e.printStackTrace();
}

if (lastSentDateString == null) {
    // File doesn't contain any date, initialize with current date
    lastSentDateString = new SimpleDateFormat("yyyy-MM-dd").format(new Date(0));
    //if it's equal to null, write the current date to the file so an email doesn't get sent again
    String lastSentDateString2 = new SimpleDateFormat("yyyy-MM-dd").format(new
Date());
    writeDateToFile("filename.txt", lastSentDateString2);
}

try {
    // Query the database for the most recent date salaries were saved
    ResultSet rs = st.executeQuery("SELECT MAX(datesaved) FROM salary_history");
    String mostRecentSalaryDate = null;
    if (rs.next()) { // Move to the first (and only) row of the result
        mostRecentSalaryDate = rs.getString(1);
    }

    SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd");
    Date lastSentDate = new Date();
    Date mostRecentDate = new Date();

    if (lastSentDateString == null){
        lastSentDateString = new SimpleDateFormat("yyyy-MM-dd").format(new Date());
    }
    else{
        lastSentDate = formatter.parse(lastSentDateString);
    }

    if (mostRecentSalaryDate == null) {
        mostRecentSalaryDate = new SimpleDateFormat("yyyy-MM-dd").format(new
Date()); // Fallback to epoch date
    }
    else{
        mostRecentDate = formatter.parse(mostRecentSalaryDate);
    }
}

```

```

// Parse dates for comparison
// Check if salaries are up-to-date for this month
Calendar cal = Calendar.getInstance();
cal.setTime(lastSentDate);
int currentDay = cal.get(Calendar.DAY_OF_MONTH);
int currentMonth = cal.get(Calendar.MONTH) + 1; // Calendar months are 0-based
int currentYear = cal.get(Calendar.YEAR);

// Check if mostRecentDate is in the current month and year
cal.setTime(mostRecentDate);
int salaryDay = cal.get(Calendar.DAY_OF_MONTH);
int salaryMonth = cal.get(Calendar.MONTH) + 1;
int salaryYear = cal.get(Calendar.YEAR);

// Determine if salaries have been done for the first and second halves of the month
boolean salaryDoneFirstHalf = salaryYear == currentYear && salaryMonth ==
currentMonth && salaryDay <= 15;
boolean salaryDoneSecondHalf = salaryYear == currentYear && salaryMonth ==
currentMonth && salaryDay > 15;

// Trigger the email if salaries for both halves of the current month are not done
if (!(salaryDoneFirstHalf && salaryDoneSecondHalf)) {
    // Ensure dates are compared without time
    SimpleDateFormat dateOnlyFormatter = new SimpleDateFormat("yyyy-MM-dd");
    Date truncatedLastSentDate =
dateOnlyFormatter.parse(dateOnlyFormatter.format(lastSentDate));
    Date truncatedCurrentDate =
dateOnlyFormatter.parse(dateOnlyFormatter.format(mostRecentDate));

    // Check if the email has already been sent today
    boolean emailSentToday = truncatedLastSentDate.equals(truncatedCurrentDate);

    if (!emailSentToday) {
        // Send reminder email

        email = email.toLowerCase();

        EmailSender send = new EmailSender(email, "Wage Reminder " +
truncatedCurrentDate.toString(),
        "Please do the wages for this month.");

        lastSentDateString = new
SimpleDateFormat("yyyy-MM-dd").format(truncatedCurrentDate);

```

```

        writeDateToFile("filename.txt", lastSentDateString); //this doesn't work evidently
    }
}
} catch (SQLException | ParseException e) {
    System.out.println("An error occurred while processing the data.");
    e.printStackTrace();
}
con.close();

maxfinder();

refreshTable("SELECT sa.*, " +
    "i.last_name || ', ' || i.first_name || ' ' || COALESCE(i.middle_name, '') AS full_name\n"
+
    "FROM salary_allocations sa\n" +
    "JOIN info i ON sa.eid = i.eid\n" +
    "WHERE i.hired = TRUE");

initComponents2();

}
catch (Exception e){
    e.printStackTrace();
    JOptionPane.showMessageDialog(null, "Error: " + e.getMessage());
}
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    pnlBackground = new com.k33ptoo.components.KGradientPanel();
    pnlSearch = new javax.swing.JPanel();
    txtSearch = new javax.swing.JTextField();
    lblSearch = new javax.swing.JLabel();
    cmbIn = new javax.swing.JComboBox<>();
    txtIn = new javax.swing.JLabel();

```

```

txtSort = new javax.swing.JLabel();
cmbColumn = new javax.swing.JComboBox<>();
cmbOrder = new javax.swing.JComboBox<>();
kbtnSort = new com.k33ptoo.components.KButton();
kbtnReset = new com.k33ptoo.components.KButton();
pnlTable = new javax.swing.JPanel();
jScrollPane1 = new javax.swing.JScrollPane();
tblEmployees = new javax.swing.JTable();
pnlEmail = new javax.swing.JPanel();
lblEmail = new javax.swing.JLabel();
pnlDelete = new javax.swing.JPanel();
kbtnSalaryHistory = new com.k33ptoo.components.KButton();
kbtnLoanHistory = new com.k33ptoo.components.KButton();
kbtnEditEmpList = new com.k33ptoo.components.KButton();
pnlDelete1 = new javax.swing.JPanel();
txtEID = new javax.swing.JTextField();
lblEID = new javax.swing.JLabel();
txtWage = new javax.swing.JTextField();
lblWage = new javax.swing.JLabel();
lblSSSSBracket = new javax.swing.JLabel();
cmbSSSSBracket = new javax.swing.JComboBox<>();
lblSSSEE = new javax.swing.JLabel();
txtSSSEE = new javax.swing.JTextField();
lblSSSER = new javax.swing.JLabel();
txtSSSER = new javax.swing.JTextField();
lblPHBracket = new javax.swing.JLabel();
cmbPHBracket = new javax.swing.JComboBox<>();
lblPHEE = new javax.swing.JLabel();
txtPHEE = new javax.swing.JTextField();
lblPHER = new javax.swing.JLabel();
txtPHER = new javax.swing.JTextField();
lblHDMFBracket = new javax.swing.JLabel();
cmbHDMFBracket = new javax.swing.JComboBox<>();
lblHDMFEE = new javax.swing.JLabel();
txtHDMFEE = new javax.swing.JTextField();
lblHDMFER = new javax.swing.JLabel();
txtHDMFER = new javax.swing.JTextField();
lblLoan = new javax.swing.JLabel();
txtLoan = new javax.swing.JTextField();
lblLoanSub = new javax.swing.JLabel();
txtLoanSub = new javax.swing.JTextField();
lblLoanAdd = new javax.swing.JLabel();
txtLoanAdd = new javax.swing.JTextField();
lblBonus = new javax.swing.JLabel();

```

```

txtBonus = new javax.swing.JTextField();
lblDeduction = new javax.swing.JLabel();
txtDeduction = new javax.swing.JTextField();
lblFinalWage = new javax.swing.JLabel();
txtFinalWage = new javax.swing.JTextField();
kbtnClearFields = new com.k33ptoo.components.KButton();
kbtnDeselect = new com.k33ptoo.components.KButton();
kbtnUpdate = new com.k33ptoo.components.KButton();
pnlEmail1 = new javax.swing.JPanel();
kbtnEditSSS = new com.k33ptoo.components.KButton();
kbtnEditPH = new com.k33ptoo.components.KButton();
kbtnEditHDMF = new com.k33ptoo.components.KButton();
kbtnLogout = new com.k33ptoo.components.KButton();
pnlSearch1 = new javax.swing.JPanel();
txtActiveEmployees = new javax.swing.JTextField();
lblTotalWages = new javax.swing.JLabel();
txtSort1 = new javax.swing.JLabel();
lblActiveEmployees = new javax.swing.JLabel();
txtTotalWages = new javax.swing.JTextField();
lblTotalER = new javax.swing.JLabel();
txtTotalER = new javax.swing.JTextField();
kbtnSave = new com.k33ptoo.components.KButton();
pnlSearch2 = new javax.swing.JPanel();
txtSort2 = new javax.swing.JLabel();
kbtnPrint = new com.k33ptoo.components.KButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setBackground(new java.awt.Color(255, 255, 255));
setResizable(false);

pnlBackground.setkEndColor(new java.awt.Color(66, 133, 244));
pnlBackground.setkGradientFocus(900);
pnlBackground.setkStartColor(new java.awt.Color(66, 133, 244));

pnlSearch.setBackground(new java.awt.Color(238, 238, 238));
pnlSearch.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
pnlSearch.setForeground(new java.awt.Color(36, 48, 94));

txtSearch.setBackground(new java.awt.Color(252, 250, 255));
txtSearch.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtSearch.setForeground(new java.awt.Color(8, 8, 8));
txtSearch.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(38, 117, 247)));

```

```

txtSearch.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtSearch.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtSearchActionPerformed(evt);
    }
});
txtSearch.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtSearchKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtSearchKeyTyped(evt);
    }
});

lblSearch.setIcon(new javax.swing.ImageIcon(getClass().getResource("/29x29
Search.png"))); // NOI18N
lblSearch.setText("jLabel2");

cmbIn.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "All", "EID",
"Full Name", "Wage", "SSS EE", "SSS ER", "PH EE", "PH ER", "HDMF EE", "HDMF ER",
"Loan", "Add", "Sub", "Bonus", "Deduction", "Final Wage" }));
cmbIn.addItemListener(new java.awt.event.ItemListener() {
    public void itemStateChanged(java.awt.event.ItemEvent evt) {
        cmbInItemStateChanged(evt);
    }
});
cmbIn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cmbInActionPerformed(evt);
    }
});
cmbIn.addPropertyChangeListener(new java.beans.PropertyChangeListener() {
    public void propertyChange(java.beans.PropertyChangeEvent evt) {
        cmbInPropertyChange(evt);
    }
});

txtIn.setFont(new java.awt.Font("Segoe UI Light", 0, 16)); // NOI18N
txtIn.setForeground(new java.awt.Color(8, 8, 8));
txtIn.setText("In:");

txtSort.setFont(new java.awt.Font("Segoe UI Light", 0, 16)); // NOI18N
txtSort.setForeground(new java.awt.Color(8, 8, 8));

```



```

        .addContainerGap()
        .addComponent(lblSearch, javax.swing.GroupLayout.PREFERRED_SIZE, 29,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(txtSearch)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(txtIn)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(cmbIn, javax.swing.GroupLayout.PREFERRED_SIZE, 74,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(24, 24, 24)
        .addComponent(txtSort)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(cmbColumn, javax.swing.GroupLayout.PREFERRED_SIZE, 106,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(cmbOrder, javax.swing.GroupLayout.PREFERRED_SIZE, 84,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(kbtnSort, javax.swing.GroupLayout.PREFERRED_SIZE, 55,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(kbtnReset, javax.swing.GroupLayout.PREFERRED_SIZE, 58,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap())
    );
    pnlSearchLayout.setVerticalGroup(
        pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlSearchLayout.createSequentialGroup()
            .addGap(5, 5, 5)

.addGroup(pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addGroup(pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
            .addComponent(kbtnSort, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, 0, Short.MAX_VALUE)

.addGroup(pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(cmbOrder, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

```



```

tblEmployees.setToolTipText("");
tblEmployees.setAutoResizeMode(javax.swing.JTable.AUTO_RESIZE_LAST_COLUMN);
tblEmployees.getTableHeader().setReorderingAllowed(false);
tblEmployees.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        tblEmployeesMouseClicked(evt);
    }
});
jScrollPane1.setViewportView(tblEmployees);
if (tblEmployees.getColumnModel().getColumnCount() > 0) {
    tblEmployees.getColumnModel().getColumn(0).setResizable(false);
    tblEmployees.getColumnModel().getColumn(1).setResizable(false);
    tblEmployees.getColumnModel().getColumn(2).setResizable(false);
    tblEmployees.getColumnModel().getColumn(3).setResizable(false);
    tblEmployees.getColumnModel().getColumn(4).setResizable(false);
    tblEmployees.getColumnModel().getColumn(5).setResizable(false);
    tblEmployees.getColumnModel().getColumn(6).setResizable(false);
    tblEmployees.getColumnModel().getColumn(7).setResizable(false);
    tblEmployees.getColumnModel().getColumn(8).setResizable(false);
    tblEmployees.getColumnModel().getColumn(9).setResizable(false);
    tblEmployees.getColumnModel().getColumn(10).setResizable(false);
    tblEmployees.getColumnModel().getColumn(11).setResizable(false);
    tblEmployees.getColumnModel().getColumn(12).setResizable(false);
    tblEmployees.getColumnModel().getColumn(13).setResizable(false);
}

javax.swing.GroupLayout pnlTableLayout = new javax.swing.GroupLayout(pnlTable);
pnlTable.setLayout(pnlTableLayout);
pnlTableLayout.setHorizontalGroup(
    pnlTableLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jScrollPane1)
);
pnlTableLayout.setVerticalGroup(
    pnlTableLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jScrollPane1)
);

pnlEmail.setBackground(new java.awt.Color(238, 238, 238));
pnlEmail.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
pnlEmail.setForeground(new java.awt.Color(36, 48, 94));

lblEmail.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblEmail.setForeground(new java.awt.Color(8, 8, 8));

```



```

.addGroup(pnlDeleteLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(kbtnSalaryHistory, javax.swing.GroupLayout.PREFERRED_SIZE,
30, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(kbtnLoanHistory, javax.swing.GroupLayout.PREFERRED_SIZE,
30, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(kbtnEditEmpList, javax.swing.GroupLayout.PREFERRED_SIZE,
30, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addContainerGap(10, Short.MAX_VALUE))
);

pnlDelete1.setBackground(new java.awt.Color(238, 238, 238));
pnlDelete1.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
pnlDelete1.setForeground(new java.awt.Color(36, 48, 94));

txtEID.setBackground(new java.awt.Color(252, 250, 255));
txtEID.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtEID.setForeground(new java.awt.Color(8, 8, 8));
txtEID.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtEID.setFocusable(false);
txtEID.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtEID.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtEIDActionPerformed(evt);
    }
});
txtEID.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtEIDKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtEIDKeyTyped(evt);
    }
});

lblEID.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblEID.setForeground(new java.awt.Color(8, 8, 8));
lblEID.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblEID.setText("EID:");

txtWage.setBackground(new java.awt.Color(252, 250, 255));

```

```

txtWage.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtWage.setForeground(new java.awt.Color(8, 8, 8));
txtWage.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtWage.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtWage.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtWageActionPerformed(evt);
    }
});
txtWage.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtWageKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtWageKeyTyped(evt);
    }
});

```

```

lblWage.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblWage.setForeground(new java.awt.Color(8, 8, 8));
lblWage.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblWage.setText("Wage:");

```

```

lblSSSBracket.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblSSSBracket.setForeground(new java.awt.Color(8, 8, 8));
lblSSSBracket.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblSSSBracket.setText("SSS Bracket:");

```

```

cmbSSSBracket.addItemListener(new java.awt.event.ItemListener() {
    public void itemStateChanged(java.awt.event.ItemEvent evt) {
        cmbSSSBracketItemStateChanged(evt);
    }
});
cmbSSSBracket.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cmbSSSBracketActionPerformed(evt);
    }
});

```

```

lblSSSEE.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblSSSEE.setForeground(new java.awt.Color(8, 8, 8));
lblSSSEE.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblSSSEE.setText("EE:");

```

```

txtSSSEE.setBackground(new java.awt.Color(252, 250, 255));
txtSSSEE.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtSSSEE.setForeground(new java.awt.Color(8, 8, 8));
txtSSSEE.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtSSSEE.setFocusable(false);
txtSSSEE.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtSSSEE.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtSSSEEActionPerformed(evt);
    }
});
txtSSSEE.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtSSSEEKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtSSSEEKeyTyped(evt);
    }
});

```

```

lblSSSER.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblSSSER.setForeground(new java.awt.Color(8, 8, 8));
lblSSSER.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblSSSER.setText("ER:");

```

```

txtSSSER.setBackground(new java.awt.Color(252, 250, 255));
txtSSSER.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtSSSER.setForeground(new java.awt.Color(8, 8, 8));
txtSSSER.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtSSSER.setFocusable(false);
txtSSSER.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtSSSER.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtSSSERActionPerformed(evt);
    }
});
txtSSSER.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtSSSERKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {

```



```

        txtSSSERKeyTyped(evt);
    }
});

lblPHBracket.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblPHBracket.setForeground(new java.awt.Color(8, 8, 8));
lblPHBracket.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblPHBracket.setText("PH Bracket:");

cmbPHBracket.addItemListener(new java.awt.event.ItemListener() {
    public void itemStateChanged(java.awt.event.ItemEvent evt) {
        cmbPHBracketItemStateChanged(evt);
    }
});

lblPHEE.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblPHEE.setForeground(new java.awt.Color(8, 8, 8));
lblPHEE.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblPHEE.setText("%EE:");

txtPHEE.setBackground(new java.awt.Color(252, 250, 255));
txtPHEE.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtPHEE.setForeground(new java.awt.Color(8, 8, 8));
txtPHEE.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtPHEE.setFocusable(false);
txtPHEE.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtPHEE.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtPHEEActionPerformed(evt);
    }
});
txtPHEE.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtPHEEKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtPHEEKeyTyped(evt);
    }
});

lblPHER.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblPHER.setForeground(new java.awt.Color(8, 8, 8));
lblPHER.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);

```

```

lblPHER.setText("%ER:");

txtPHER.setBackground(new java.awt.Color(252, 250, 255));
txtPHER.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtPHER.setForeground(new java.awt.Color(8, 8, 8));
txtPHER.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtPHER.setFocusable(false);
txtPHER.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtPHER.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtPHERActionPerformed(evt);
    }
});
txtPHER.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtPHERKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtPHERKeyTyped(evt);
    }
});

lblHDMFBracket.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblHDMFBracket.setForeground(new java.awt.Color(8, 8, 8));
lblHDMFBracket.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblHDMFBracket.setText("HDMF Bracket:");

cmbHDMFBracket.addItemListener(new java.awt.event.ItemListener() {
    public void itemStateChanged(java.awt.event.ItemEvent evt) {
        cmbHDMFBracketItemStateChanged(evt);
    }
});
cmbHDMFBracket.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cmbHDMFBracketActionPerformed(evt);
    }
});

lblHDMFEE.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblHDMFEE.setForeground(new java.awt.Color(8, 8, 8));
lblHDMFEE.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblHDMFEE.setText("%EE:");

```

```

txtHDMFEE.setBackground(new java.awt.Color(252, 250, 255));
txtHDMFEE.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtHDMFEE.setForeground(new java.awt.Color(8, 8, 8));
txtHDMFEE.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtHDMFEE.setFocusable(false);
txtHDMFEE.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtHDMFEE.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtHDMFEEActionPerformed(evt);
    }
});
txtHDMFEE.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtHDMFEEKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtHDMFEEKeyTyped(evt);
    }
});

```

```

lblHDMFER.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblHDMFER.setForeground(new java.awt.Color(8, 8, 8));
lblHDMFER.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblHDMFER.setText("%ER:");

```

```

txtHDMFER.setBackground(new java.awt.Color(252, 250, 255));
txtHDMFER.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtHDMFER.setForeground(new java.awt.Color(8, 8, 8));
txtHDMFER.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtHDMFER.setFocusable(false);
txtHDMFER.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtHDMFER.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtHDMFERActionPerformed(evt);
    }
});
txtHDMFER.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtHDMFERKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtHDMFERKeyTyped(evt);
    }
});

```

```

    }
});

lblLoan.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblLoan.setForeground(new java.awt.Color(8, 8, 8));
lblLoan.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblLoan.setText("Loan:");

txtLoan.setBackground(new java.awt.Color(252, 250, 255));
txtLoan.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtLoan.setForeground(new java.awt.Color(8, 8, 8));
txtLoan.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtLoan.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtLoan.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtLoanActionPerformed(evt);
    }
});
txtLoan.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtLoanKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtLoanKeyTyped(evt);
    }
});

lblLoanSub.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblLoanSub.setForeground(new java.awt.Color(8, 8, 8));
lblLoanSub.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblLoanSub.setText("Loan Sub:");

txtLoanSub.setBackground(new java.awt.Color(252, 250, 255));
txtLoanSub.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtLoanSub.setForeground(new java.awt.Color(8, 8, 8));
txtLoanSub.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtLoanSub.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtLoanSub.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtLoanSubActionPerformed(evt);
    }
});

```

```

txtLoanSub.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtLoanSubKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtLoanSubKeyTyped(evt);
    }
});

lblLoanAdd.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblLoanAdd.setForeground(new java.awt.Color(8, 8, 8));
lblLoanAdd.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblLoanAdd.setText("Loan Add:");

txtLoanAdd.setBackground(new java.awt.Color(252, 250, 255));
txtLoanAdd.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtLoanAdd.setForeground(new java.awt.Color(8, 8, 8));
txtLoanAdd.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtLoanAdd.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtLoanAdd.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtLoanAddActionPerformed(evt);
    }
});
txtLoanAdd.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtLoanAddKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtLoanAddKeyTyped(evt);
    }
});

lblBonus.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblBonus.setForeground(new java.awt.Color(8, 8, 8));
lblBonus.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblBonus.setText("Bonus:");

txtBonus.setBackground(new java.awt.Color(252, 250, 255));
txtBonus.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtBonus.setForeground(new java.awt.Color(8, 8, 8));
txtBonus.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));

```

```

txtBonus.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtBonus.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtBonusActionPerformed(evt);
    }
});
txtBonus.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtBonusKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtBonusKeyTyped(evt);
    }
});

lblDeduction.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblDeduction.setForeground(new java.awt.Color(8, 8, 8));
lblDeduction.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblDeduction.setText("Deduction:");

txtDeduction.setBackground(new java.awt.Color(252, 250, 255));
txtDeduction.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtDeduction.setForeground(new java.awt.Color(8, 8, 8));
txtDeduction.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtDeduction.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtDeduction.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtDeductionActionPerformed(evt);
    }
});
txtDeduction.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtDeductionKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtDeductionKeyTyped(evt);
    }
});

lblFinalWage.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblFinalWage.setForeground(new java.awt.Color(8, 8, 8));
lblFinalWage.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblFinalWage.setText("Final Wage:");

```

```

txtFinalWage.setBackground(new java.awt.Color(252, 250, 255));
txtFinalWage.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtFinalWage.setForeground(new java.awt.Color(8, 8, 8));
txtFinalWage.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtFinalWage.setFocusable(false);
txtFinalWage.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtFinalWage.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtFinalWageActionPerformed(evt);
    }
});
txtFinalWage.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtFinalWageKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtFinalWageKeyTyped(evt);
    }
});

```

```

kbtnClearFields.setText("Clear Fields");
kbtnClearFields.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnClearFields.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnClearFields.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnClearFields.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnClearFields.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnClearFields.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnClearFields.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnClearFieldsActionPerformed(evt);
    }
});

```

```

kbtnDeselect.setText("Deselect");
kbtnDeselect.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnDeselect.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnDeselect.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnDeselect.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnDeselect.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnDeselect.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnDeselect.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {

```

```

        kbtnDeselectActionPerformed(evt);
    }
});

kbtnUpdate.setText("Update");
kbtnUpdate.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnUpdate.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnUpdate.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnUpdate.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnUpdate.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnUpdate.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnUpdate.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnUpdateActionPerformed(evt);
    }
});

javax.swing.GroupLayout pnlDelete1Layout = new javax.swing.GroupLayout(pnlDelete1);
pnlDelete1.setLayout(pnlDelete1Layout);
pnlDelete1Layout.setHorizontalGroup(
    pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlDelete1Layout.createSequentialGroup()
            .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .add(pnlDelete1Layout.createSequentialGroup()
                    .addContainerGap()
                )
            )
        )

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .add(pnlDelete1Layout.createSequentialGroup()
        .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .add(pnlDelete1Layout.createSequentialGroup()
                .addGroup(pnlDelete1Layout.createSequentialGroup()
                    .addComponent(kbtnClearFields, javax.swing.GroupLayout.PREFERRED_SIZE,
80, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(36, 36, 36)
                    .addComponent(kbtnDeselect, javax.swing.GroupLayout.PREFERRED_SIZE,
80, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
33, Short.MAX_VALUE)
                    .addComponent(kbtnUpdate, javax.swing.GroupLayout.PREFERRED_SIZE, 80,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(15, 15, 15))
                .addGroup(pnlDelete1Layout.createSequentialGroup()

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                    .addComponent(lblSSSEE, javax.swing.GroupLayout.PREFERRED_SIZE, 76,
javax.swing.GroupLayout.PREFERRED_SIZE)

```



```

        .addComponent(lblSSSBracket,
javax.swing.GroupLayout.PREFERRED_SIZE, 76,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(lblEID, javax.swing.GroupLayout.PREFERRED_SIZE, 76,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(lblWage, javax.swing.GroupLayout.PREFERRED_SIZE, 76,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(pnlDelete1Layout.createSequentialGroup())

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
G)
        .addComponent(txtWage)
        .addComponent(txtEID))
        .addGap(18, 18, 18))
        .addGroup(pnlDelete1Layout.createSequentialGroup())

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
G)
        .addComponent(cmbSSSBracket,
javax.swing.GroupLayout.PREFERRED_SIZE, 146,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGroup(pnlDelete1Layout.createSequentialGroup())
        .addComponent(txtSSSEE,
javax.swing.GroupLayout.PREFERRED_SIZE, 61,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(lblSSSER)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(txtSSSER,
javax.swing.GroupLayout.PREFERRED_SIZE, 61,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlDelete1Layout.createSequentialGroup())
        .addComponent(lblBonus, javax.swing.GroupLayout.PREFERRED_SIZE, 76,
javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        .addGroup(pnlDelete1Layout.createSequentialGroup())

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
        .addGroup(pnlDelete1Layout.createSequentialGroup()
            .addComponent(lblPHEE, javax.swing.GroupLayout.PREFERRED_SIZE,
76, javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(txtPHEE, javax.swing.GroupLayout.PREFERRED_SIZE,
61, javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(lblPHER)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(txtPHER, javax.swing.GroupLayout.PREFERRED_SIZE,
61, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(pnlDelete1Layout.createSequentialGroup()
            .addComponent(lblHDMFEE,
javax.swing.GroupLayout.PREFERRED_SIZE, 76,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(txtHDMFEE,
javax.swing.GroupLayout.PREFERRED_SIZE, 61,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(lblHDMFER)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(txtHDMFER,
javax.swing.GroupLayout.PREFERRED_SIZE, 61,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(pnlDelete1Layout.createSequentialGroup()
            .addComponent(lblHDMFBracket)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(cmbHDMFBracket, 0,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addGroup(pnlDelete1Layout.createSequentialGroup())

```

```

        .addGap(9, 9, 9)
        .addComponent(lblPHBracket,
javax.swing.GroupLayout.PREFERRED_SIZE, 76,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(cmbPHBracket,
javax.swing.GroupLayout.PREFERRED_SIZE, 143,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGap(0, 0, Short.MAX_VALUE))))
        .addGroup pnlDelete1Layout.createSequentialGroup()

        .addGroup pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        {
            .addGroup pnlDelete1Layout.createSequentialGroup()
            {
                .addComponent(lblDeduction, javax.swing.GroupLayout.PREFERRED_SIZE, 76,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(txtDeduction))
            .addGroup pnlDelete1Layout.createSequentialGroup()
            {
                .addComponent(lblLoanAdd, javax.swing.GroupLayout.PREFERRED_SIZE, 76,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

            }
        }
        .addGroup pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        {
            .addComponent(txtBonus)
            .addComponent(txtLoanAdd)))
        .addGroup pnlDelete1Layout.createSequentialGroup()
        {
            .addComponent(lblLoan, javax.swing.GroupLayout.PREFERRED_SIZE, 76,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(txtLoan))
        .addGroup pnlDelete1Layout.createSequentialGroup()
        {
            .addComponent(lblLoanSub, javax.swing.GroupLayout.PREFERRED_SIZE, 76,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(txtLoanSub))
        .addGroup pnlDelete1Layout.createSequentialGroup()
        {
            .addComponent(lblFinalWage, javax.swing.GroupLayout.PREFERRED_SIZE,
76, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(txtFinalWage)))
        .addContainerGap()
    }

```

```

);
pnlDelete1Layout.setVerticalGroup(
    pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlDelete1Layout.createSequentialGroup()
            .addContainerGap()

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblEID)
    .addComponent(txtEID, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    .addComponent(txtWage, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(lblWage))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblSSSBracket)
    .addComponent(cmbSSSBracket, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblSSSEE)
    .addComponent(txtSSSEE, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(lblSSSER)
    .addComponent(txtSSSER, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblPHBracket)
    .addComponent(cmbPHBracket, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

```

```

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblPHEE)
    .addComponent(txtPHEE, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(lblPHER)
    .addComponent(txtPHER, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblHDMFBracket)
    .addComponent(cmbHDMFBracket, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblHDMFEE)
    .addComponent(txtHDMFEE, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(lblHDMFER)
    .addComponent(txtHDMFER, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblLoan)
    .addComponent(txtLoan, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblLoanSub)
    .addComponent(txtLoanSub, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

```

```

        .addComponent(lblLoanAdd)
        .addComponent(txtLoanAdd, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    NE)
        .addComponent(lblBonus)
        .addComponent(txtBonus, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    NE)
        .addComponent(lblDeduction)
        .addComponent(txtDeduction, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    NE)
        .addComponent(lblFinalWage)
        .addComponent(txtFinalWage, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 26,
Short.MAX_VALUE)

    .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    NE)
        .addComponent(kbtnClearFields, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(kbtnDeselect, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(kbtnUpdate, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(14, 14, 14))
    );

    pnlEmail1.setBackground(new java.awt.Color(238, 238, 238));
    pnlEmail1.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
    pnlEmail1.setForeground(new java.awt.Color(36, 48, 94));

    kbtnEditSSS.setText("Edit SSS");

```

```

kbtnEditSSS.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnEditSSS.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnEditSSS.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnEditSSS.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnEditSSS.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnEditSSS.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnEditSSS.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnEditSSSActionPerformed(evt);
    }
});

```

```

kbtnEditPH.setText("Edit PH");
kbtnEditPH.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnEditPH.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnEditPH.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnEditPH.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnEditPH.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnEditPH.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnEditPH.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnEditPHActionPerformed(evt);
    }
});

```

```

kbtnEditHDMF.setText("Edit HDMF");
kbtnEditHDMF.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnEditHDMF.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnEditHDMF.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnEditHDMF.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnEditHDMF.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnEditHDMF.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnEditHDMF.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnEditHDMFActionPerformed(evt);
    }
});

```

```

kbtnLogout.setText("Log Out");
kbtnLogout.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnLogout.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnLogout.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnLogout.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnLogout.setkSelectedColor(new java.awt.Color(252, 250, 255));

```

```

        kbtnLogout.setkStartColor(new java.awt.Color(28, 84, 140));
        kbtnLogout.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                kbtnLogoutActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout pnlEmail1Layout = new javax.swing.GroupLayout(pnlEmail1);
        pnlEmail1.setLayout(pnlEmail1Layout);
        pnlEmail1Layout.setHorizontalGroup(
            pnlEmail1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(pnlEmail1Layout.createSequentialGroup()
                    .addGroup(pnlEmail1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addGroup(pnlEmail1Layout.createSequentialGroup()
                            .addComponent(kbbtnEditSSS, javax.swing.GroupLayout.PREFERRED_SIZE,
                                67, javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                            .addComponent(kbbtnEditPH, javax.swing.GroupLayout.PREFERRED_SIZE, 67,
                                javax.swing.GroupLayout.PREFERRED_SIZE))
                        .addGroup(pnlEmail1Layout.createSequentialGroup()
                            .addComponent(kbbtnEditHDMF, javax.swing.GroupLayout.PREFERRED_SIZE,
                                67, javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                            .addComponent(kbbtnLogout, javax.swing.GroupLayout.PREFERRED_SIZE, 67,
                                javax.swing.GroupLayout.PREFERRED_SIZE)))
                    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                .addGroup(pnlEmail1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(pnlEmail1Layout.createSequentialGroup()
                        .addComponent(kbbtnEditSSS, javax.swing.GroupLayout.PREFERRED_SIZE, 36,
                            javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(kbbtnEditPH, javax.swing.GroupLayout.PREFERRED_SIZE, 36,
                            javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

```



```

.addGroup(pnlEmail1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(kbtnLogout, javax.swing.GroupLayout.PREFERRED_SIZE, 36,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(kbtnEditHDMF, javax.swing.GroupLayout.PREFERRED_SIZE, 36,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);

pnlSearch1.setBackground(new java.awt.Color(238, 238, 238));
pnlSearch1.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
pnlSearch1.setForeground(new java.awt.Color(36, 48, 94));

txtActiveEmployees.setBackground(new java.awt.Color(252, 250, 255));
txtActiveEmployees.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtActiveEmployees.setForeground(new java.awt.Color(8, 8, 8));
txtActiveEmployees.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2,
new java.awt.Color(38, 117, 247)));
txtActiveEmployees.setFocusable(false);
txtActiveEmployees.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtActiveEmployees.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtActiveEmployeesActionPerformed(evt);
    }
});
txtActiveEmployees.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtActiveEmployeesKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtActiveEmployeesKeyTyped(evt);
    }
});

lblTotalWages.setFont(new java.awt.Font("Segoe UI Light", 0, 16)); // NOI18N
lblTotalWages.setForeground(new java.awt.Color(8, 8, 8));
lblTotalWages.setText("Total Wages:");

txtSort1.setFont(new java.awt.Font("Segoe UI Light", 0, 16)); // NOI18N
txtSort1.setForeground(new java.awt.Color(8, 8, 8));

lblActiveEmployees.setFont(new java.awt.Font("Segoe UI Light", 0, 16)); // NOI18N

```

```

lblActiveEmployees.setForeground(new java.awt.Color(8, 8, 8));
lblActiveEmployees.setText("Active Employees:");

txtTotalWages.setBackground(new java.awt.Color(252, 250, 255));
txtTotalWages.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtTotalWages.setForeground(new java.awt.Color(8, 8, 8));
txtTotalWages.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(38, 117, 247)));
txtTotalWages.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtTotalWages.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtTotalWagesActionPerformed(evt);
    }
});
txtTotalWages.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtTotalWagesKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtTotalWagesKeyTyped(evt);
    }
});

lblTotalER.setFont(new java.awt.Font("Segoe UI Light", 0, 16)); // NOI18N
lblTotalER.setForeground(new java.awt.Color(8, 8, 8));
lblTotalER.setText("Total ER:");

txtTotalER.setBackground(new java.awt.Color(252, 250, 255));
txtTotalER.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtTotalER.setForeground(new java.awt.Color(8, 8, 8));
txtTotalER.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(38, 117, 247)));
txtTotalER.setFocusable(false);
txtTotalER.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtTotalER.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtTotalERActionPerformed(evt);
    }
});
txtTotalER.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtTotalERKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {

```



```

        pnlSearch1Layout.setVerticalGroup(
            pnlSearch1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(pnlSearch1Layout.createSequentialGroup())
                .addContainerGap()

        .addGroup(pnlSearch1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            G)
                .addGroup(pnlSearch1Layout.createSequentialGroup())
                .addGap(0, 0, Short.MAX_VALUE)
                .addComponent(txtSort1))
                .addGroup(pnlSearch1Layout.createSequentialGroup()

        .addGroup(pnlSearch1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            G)

        .addGroup(pnlSearch1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            NE)
                .addComponent(lblActiveEmployees)
                .addComponent(txtActiveEmployees,
                    javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(lblTotalWages)
                .addComponent(txtTotalWages,
                    javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(lblTotalER)
                .addComponent(txtTotalER, javax.swing.GroupLayout.PREFERRED_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addComponent(kbtnSave, javax.swing.GroupLayout.PREFERRED_SIZE, 29,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(0, 0, Short.MAX_VALUE))))
    );

    pnlSearch2.setBackground(new java.awt.Color(238, 238, 238));
    pnlSearch2.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
        java.awt.Color(255, 255, 255)));
    pnlSearch2.setForeground(new java.awt.Color(36, 48, 94));

    txtSort2.setFont(new java.awt.Font("Segoe UI Light", 0, 16)); // NOI18N
    txtSort2.setForeground(new java.awt.Color(8, 8, 8));

    kbtnPrint.setText("Print");
    kbtnPrint.setkEndColor(new java.awt.Color(28, 84, 140));
    kbtnPrint.setkHoverEndColor(new java.awt.Color(76, 111, 167));

```

```

ktbtnPrint.setkHoverForeGround(new java.awt.Color(252, 250, 255));
ktbtnPrint.setkHoverStartColor(new java.awt.Color(76, 111, 167));
ktbtnPrint.setkSelectedColor(new java.awt.Color(252, 250, 255));
ktbtnPrint.setkStartColor(new java.awt.Color(28, 84, 140));
ktbtnPrint.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ktbtnPrintActionPerformed(evt);
    }
});

javax.swing.GroupLayout pnlSearch2Layout = new javax.swing.GroupLayout(pnlSearch2);
pnlSearch2.setLayout(pnlSearch2Layout);
pnlSearch2Layout.setHorizontalGroup(
    pnlSearch2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlSearch2Layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addComponent(ktbtnPrint, javax.swing.GroupLayout.PREFERRED_SIZE, 55,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(233, 233, 233)
            .addComponent(txtSort2))
        );
pnlSearch2Layout.setVerticalGroup(
    pnlSearch2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlSearch2Layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addComponent(ktbtnPrint, javax.swing.GroupLayout.PREFERRED_SIZE, 29,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(10, 10, 10)
            .addComponent(txtSort2))
        );

javax.swing.GroupLayout pnlBackgroundLayout = new
javax.swing.GroupLayout(pnlBackground);
pnlBackground.setLayout(pnlBackgroundLayout);
pnlBackgroundLayout.setHorizontalGroup(
    pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlBackgroundLayout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addComponent(ktbtnPrint, javax.swing.GroupLayout.PREFERRED_SIZE, 55,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(233, 233, 233)
            .addComponent(txtSort2))
        );

```

```

.addGroup(pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(pnlDelete, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(pnlDelete1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addGroup(pnlBackgroundLayout.createSequentialGroup())
    .addComponent(pnlEmail, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(pnlEmail1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(pnlBackgroundLayout.createSequentialGroup())
    .addComponent(pnlSearch1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(pnlSearch2, javax.swing.GroupLayout.PREFERRED_SIZE, 76,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addComponent(pnlTable, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(pnlSearch, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    .addContainerGap()
);
pnlBackgroundLayout.setVerticalGroup(

pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(pnlBackgroundLayout.createSequentialGroup())
    .addContainerGap()

.addGroup(pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
    .addGroup(pnlBackgroundLayout.createSequentialGroup())

.addGroup(pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
    .addComponent(pnlSearch2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

```

```

        .addComponent(pnlSearch1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 9,
Short.MAX_VALUE)
        .addComponent(pnlSearch, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addComponent(pnlEmail1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(pnlEmail, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(pnlBackgroundLayout.createSequentialGroup()
        .addComponent(pnlDelete1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(pnlDelete, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addComponent(pnlTable, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addContainerGap())
    );

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(pnlBackground, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(pnlBackground, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );

    pack();
    setLocationRelativeTo(null);
} // </editor-fold>

private void kbbtnSortActionPerformed(java.awt.event.ActionEvent evt) {
    try{

```

```

String search = txtSearch.getText().trim();

int column = cmbColumn.getSelectedIndex();

String[] columnnames =
{"eid","monthly_wage_rate","sss_ee","sss_er","ph_ee_p","ph_er_p","hdmf_ee_p","hdmf_ee_p","l
oan","loan_add","loan_sub","deduction","bonus"};
String strcolumn = columnnames[column];

int sort = cmbOrder.getSelectedIndex();
String strsort = "";
switch (sort){
    case 0:
        strsort = "ASC";
        break;
    case 1:
        strsort = "DESC";
}

if (column==0 && sort==0){
    DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();

    // Connect to SALARY_GENERATOR database
    String databaseURL = "jdbc:derby:SALARY_GENERATOR";
    Connection con = DriverManager.getConnection(databaseURL);
    Statement st = con.createStatement();

    // Execute the SQL query
    ResultSet rs = st.executeQuery("SELECT sa.*, " +
        "i.last_name || ' ' || i.first_name || ' ' || COALESCE(i.middle_name, '') AS
full_name " +
        "FROM salary_allocations sa " +
        "JOIN info i ON sa.eid = i.eid " +
        "WHERE i.hired = TRUE");

    // Store result set in a list for sorting
    List<Object[]> rowDataList = new ArrayList<>();
    Double total_wages = 0.0;
    Double total_er = 0.0;

    while (rs.next()) {
        String eid = rs.getString(2); // EID
        String fullname = rs.getString(16); // Full name
        String wage = rs.getString(4); // Wage
    }
}

```



```

String sss_ee = rs.getString(5);
String sss_er = rs.getString(6);
String ph_ee_p = rs.getString(7);
String ph_er_p = rs.getString(8);
String hdmf_ee_p = rs.getString(9);
String hdmf_er_p = rs.getString(10);
String loan = rs.getString(11);
String loan_add = rs.getString(12);
String loan_sub = rs.getString(13);
String deduction = rs.getString(14);
String bonus = rs.getString(15);

Double dwage = Double.parseDouble(wage);
Double dsss_ee = Double.parseDouble(sss_ee);
Double dsss_er = Double.parseDouble(sss_er);
Double dph_ee_p = Double.parseDouble(ph_ee_p);
Double dph_er_p = Double.parseDouble(ph_er_p);
Double dhdmf_ee_p = Double.parseDouble(hdmf_ee_p);
Double dhdmf_er_p = Double.parseDouble(hdmf_er_p);
Double dloan = Double.parseDouble(loan);
Double dloan_add = Double.parseDouble(loan_add);
Double dloan_sub = Double.parseDouble(loan_sub);
Double ddeduction = Double.parseDouble(deduction);
Double dbonus = Double.parseDouble(bonus);

Double final_wage = dwage - dsss_ee - (dph_ee_p / 100 * dwage) - (dhdmf_ee_p /
100 * dwage) - dloan_sub - ddeduction + dbonus;
Double er = dsss_er + (dph_er_p / 100 * dwage) + (dhdmf_er_p / 100 * dwage);
total_wages += final_wage;
total_er += er;

rowDataList.add(new Object[]{Integer.parseInt(eid), fullname, wage, sss_ee,
sss_er, ph_ee_p, ph_er_p, hdmf_ee_p, hdmf_er_p, loan, loan_add, loan_sub, deduction,
bonus, final_wage});
}

// Bubble Sort to sort by EID (first column)
for (int j = 0; j < rowDataList.size() - 1; j++) {
    for (int k = 0; k < rowDataList.size() - j - 1; k++) {
        if ((int) rowDataList.get(k)[0] > (int) rowDataList.get(k + 1)[0]) {
            Object[] temp = rowDataList.get(k);
            rowDataList.set(k, rowDataList.get(k + 1));
            rowDataList.set(k + 1, temp);
        }
    }
}

```

```

    }
}

// Reset table model and add sorted rows
model.setRowCount(0);
for (Object[] rowData : rowDataList) {
    model.addRow(rowData);
}

// Update total wages and total employer contributions
String strTotalWages = total_wages.toString();
txtTotalWages.setText(strTotalWages);

String strTotalER = total_er.toString();
txtTotalER.setText(strTotalER);

con.close();

txtActiveEmployees.setText(Integer.toString(rowDataList.size()));
}
else if (column==0 && sort==1){
    //Iterate through the resultset backwards
    DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
    String databaseURL = "jdbc:derby:SALARY_GENERATOR";
    Connection con = DriverManager.getConnection(databaseURL);
    Statement st =
con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_READ_ONL
Y);

    String sql = "SELECT sa.*, " +
    "i.last_name || ', ' || i.first_name || ' ' || COALESCE(i.middle_name, '') AS full_name " +
    "FROM salary_allocations sa " +
    "JOIN info i ON sa.eid = i.eid " +
    "WHERE i.hired = TRUE";
    ResultSet rs = st.executeQuery(sql);

    //Moves the cursor to the end of the ResultSet object
    model.setRowCount(0); //reset the table
    int i = 0;

    Double total_wages = 0.0;
    Double total_er = 0.0;

    rs.afterLast();
    model.setRowCount(0);

```

```

while(rs.previous()){

    String eid = rs.getString(2); // EID
    String wage = rs.getString(4); // wage
    String sss_ee = rs.getString(5); // sss_ee
    String sss_er = rs.getString(6); // sss_er
    String ph_ee_p = rs.getString(7); // ph_ee_p
    String ph_er_p = rs.getString(8); // ph_er_p
    String hdmf_ee_p = rs.getString(9); // hdmf_ee_p
    String hdmf_er_p = rs.getString(10); // hdmf_er_p
    String loan = rs.getString(11); // loan
    String loan_add = rs.getString(12); // loan_add
    String loan_sub = rs.getString(13); // loan_sub
    String deduction = rs.getString(14); // deduction
    String bonus = rs.getString(15); // bonus
    String fullname = rs.getString(16); //fullname

    Double dwage = Double.parseDouble(wage);
    Double dsss_ee = Double.parseDouble(sss_ee);
    Double dsss_er = Double.parseDouble(sss_er);
    Double dph_ee_p = Double.parseDouble(ph_ee_p);
    Double dph_er_p = Double.parseDouble(ph_er_p);
    Double dhdmf_ee_p = Double.parseDouble(hdmf_ee_p);
    Double dhdmf_er_p = Double.parseDouble(hdmf_er_p);
    Double dloan = Double.parseDouble(loan);
    Double dloan_add = Double.parseDouble(loan_add);
    Double dloan_sub = Double.parseDouble(loan_sub);
    Double ddeduction = Double.parseDouble(deduction);
    Double dbonus = Double.parseDouble(bonus);

    Double final_wage = dwage - dsss_ee - (dph_ee_p/100*dwage) -
(dhdmf_ee_p/100*dwage) - dloan_sub - ddeduction + dbonus;
    Double er = dsss_er + (dph_er_p/100*dwage) + (dhdmf_er_p/100*dwage);
    total_wages+=final_wage;
    total_er+=er;

    model.addRow(new Object[]{eid, fullname, wage, sss_ee, sss_er, ph_ee_p,
ph_er_p, hdmf_ee_p, hdmf_er_p, loan, loan_add, loan_sub, deduction, bonus, final_wage});
}
rs.close();
st.close();
con.close();

```

```

        String strTotalWages = total_wages.toString();
        txtTotalWages.setText(strTotalWages);

        String strTotalER = total_er.toString();
        txtTotalER.setText(strTotalER);

        con.close();

        String stri = Integer.toString(i);
        txtActiveEmployees.setText(stri);
    }
    else{
        refreshTable("SELECT sa.*, " +
            "i.last_name || ', ' || i.first_name || ' ' || COALESCE(i.middle_name, '') AS full_name\n"
+
            "FROM salary_allocations sa\n" +
            "JOIN info i ON sa.eid = i.eid\n" +
            "WHERE i.hired = TRUE order by sa."+strcolumn+" "+strsort);
    }

}

catch (Exception e){
    e.printStackTrace();
}
}

private void cmbColumnActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void tblEmployeesMouseClicked(java.awt.event.MouseEvent evt) {
    try {
        // Prerequisite for the update function
        DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
        int selectedrow = tblEmployees.getSelectedRow();

        if (tblEmployees.getRowSorter() != null) {
            // Get the correct row for a filtered JTable (RowFilter)
            selectedrow = tblEmployees.getRowSorter().convertRowIndexToModel(selectedrow);
        }

        String eid = model.getValueAt(selectedrow, 0).toString();
        String fullname = model.getValueAt(selectedrow, 1).toString();
    }
}

```

```

String wage = model.getValueAt(selectedrow, 2).toString();
String sssee = model.getValueAt(selectedrow, 3).toString();
String ssser = model.getValueAt(selectedrow, 4).toString();
String phee = model.getValueAt(selectedrow, 5).toString();
String pher = model.getValueAt(selectedrow, 6).toString();
String hdmfee = model.getValueAt(selectedrow, 7).toString();
String hdmfer = model.getValueAt(selectedrow, 8).toString();
String loan = model.getValueAt(selectedrow, 9).toString();
String add = model.getValueAt(selectedrow, 10).toString();
String sub = model.getValueAt(selectedrow, 11).toString();
String bonus = model.getValueAt(selectedrow, 12).toString();
String deduction = model.getValueAt(selectedrow, 13).toString();
String finalwage = model.getValueAt(selectedrow, 14).toString();

```

```

this.selectedroweid = eid;

```

```

txtEID.setText("");
txtWage.setText("");
txtSSSEE.setText("");
txtSSSER.setText("");
txtPHEE.setText("");
txtPHER.setText("");
txtHDMFEE.setText("");
txtHDMFER.setText("");
txtLoan.setText("");
txtLoanAdd.setText("");
txtLoanSub.setText("");
txtDeduction.setText("");
txtBonus.setText("");

```

```

// Use SwingUtilities.invokeLater to ensure updates happen on EDT
// SwingUtilities.invokeLater(() -> {
//
//
//
//
// });

```

```

txtEID.setText(eid);
txtWage.setText(wage);
txtSSSEE.setText(sssee);
txtSSSER.setText(ssser);
txtPHEE.setText(phee);
txtPHER.setText(pher);
txtHDMFEE.setText(hdmfee);

```

```

        txtHDMFER.setText(hdmfer);
        txtLoan.setText(loan);
        txtLoanAdd.setText(add);
        txtLoanSub.setText(sub);
        txtBonus.setText(bonus);
        txtDeduction.setText(deduction);
        txtFinalWage.setText(finalwage);
        System.out.println("Selected Row: " + selectedrow);
    }
    catch (Exception e) {
        e.printStackTrace();
    }
}

private void cmbInPropertyChange(java.beans.PropertyChangeEvent evt) {

}

private void cmbInActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void cmbInItemStateChanged(java.awt.event.ItemEvent evt) {
    try{
        search();
    }
    catch (Exception e){

    }
}

private void txtSearchKeyTyped(java.awt.event.KeyEvent evt) {

}

private void txtSearchKeyReleased(java.awt.event.KeyEvent evt) {
    try{
        search();
    }
    catch (Exception e){

    }
}

```

```

private void txtSearchActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void kbtnResetActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        //Resets the search and sort functions; restores table to original
        refreshtable("SELECT sa.*, " +
            "i.last_name || ', ' || i.first_name || ' ' || COALESCE(i.middle_name, '') AS full_name\n"
+
            "FROM salary_allocations sa\n" +
            "JOIN info i ON sa.eid = i.eid\n" +
            "WHERE i.hired = TRUE");
        cmbColumn.setSelectedIndex(0);
        cmbOrder.setSelectedIndex(0);

        txtSearch.setText("");
        cmbIn.setSelectedIndex(0);

        search();
    }
    catch (Exception e){
        e.printStackTrace();
    }
}

private void kbtnEditSSSActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        new SSS(this.uid,this.email).setVisible(true);
        this.setVisible(false);
    }
    catch (Exception e){

    }
}

private void kbtnEditPHActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        new PhilHealth(this.uid,this.email).setVisible(true);
        this.setVisible(false);
    }
    catch (Exception e){

    }
}

```

```

}

private void kbtnEditHDMFActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        new HDMF(this.uid,this.email).setVisible(true);
        this.setVisible(false);
    }
    catch (Exception e){

    }
}

private void kbtnLogoutActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        int confirm = JOptionPane.showConfirmDialog(null,"Are you sure?");
        if (confirm==0){
            //Return to Login menu
            new Login().setVisible(true);
            this.setVisible(false);
        }
    }
    catch (Exception e){

    }
}

private void txtActiveEmployeesActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtActiveEmployeesKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtActiveEmployeesKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtTotalWagesActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

```



```

private void txtTotalWagesKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtTotalWagesKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtTotalERActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtTotalERKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtTotalERKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private int maxfinder2(){ //figure out the highest eid in salary history
    try{
        //establish connection to the database
        String databaseURL = "jdbc:derby:SALARY_GENERATOR";
        Connection con = DriverManager.getConnection(databaseURL);

        String sql;
        Statement st = con.createStatement();

        sql = "SELECT * FROM salary_history";
        ResultSet rs = st.executeQuery(sql);

        //db stands for database

        //Initialize a max variable
        int max = 0;
        while (rs.next()==true){
            String dbeid = rs.getString(1);
            int eid = Integer.parseInt(dbeid);
            if (eid>max){ //compare the max variable to the shid in the resultset
                max = eid;
            }
        }
    }
}

```

```

        con.close();

        //automatically returns the value of the highest record in salary history
        //incrementing this by 1 is a completely unique shid value.
        return max;
    }
    catch (Exception e){
        return -1;
    }
}

private int maxfinder3(){ //figure out the highest eid
    try{
        String databaseURL = "jdbc:derby:SALARY_GENERATOR";
        Connection con = DriverManager.getConnection(databaseURL);

        String sql;
        Statement st = con.createStatement();

        sql = "SELECT * FROM loan_history";
        ResultSet rs = st.executeQuery(sql);

        //db stands for database

        int max = 0;
        while (rs.next()==true){
            String dbeid = rs.getString(1);
            int eid = Integer.parseInt(dbeid);
            if (eid>max){
                max = eid;
            }
        }

        con.close();

        return max;
    }
    catch (Exception e){
        return -1;
    }
}

private void kbtnSaveActionPerformed(java.awt.event.ActionEvent evt) {

```

```

//must send all the data to salary history and loan history

try{

    int confirm = JOptionPane.showConfirmDialog(null,"Do you want to save ALL current
table contents to Salary and Loan History?");

    if (confirm==0){
        //Prerequisite for the save function
        DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();

        String eid = "";
        String fullname = "";
        String wage = "";
        String sssee = "";
        String ssser = "";
        String phee = "";
        String pher = "";
        String hdmfee="";
        String hdmfer="";
        String loan = "";
        String add = "";
        String sub = "";
        String bonus = "";
        String deduction = "";
        String finalwage = "";

        payperiodfinder();

        for (int row = 0; row < tblEmployees.getRowCount(); row++) {
            int selectedrow = row;
            if (tblEmployees.getRowSorter()!=null) {
                //Gets the correct row for a filtered jTable (RowFilter)
                selectedrow = tblEmployees.getRowSorter().convertRowIndexToModel(row);
            }

            eid = model.getValueAt(selectedrow,0).toString();
            fullname = model.getValueAt(selectedrow, 1).toString();
            wage = model.getValueAt(selectedrow,2).toString();
            sssee = model.getValueAt(selectedrow,3).toString();
            ssser = model.getValueAt(selectedrow,4).toString();
            phee = model.getValueAt(selectedrow,5).toString();

```

```

pher = model.getValueAt(selectedrow,6).toString();
hdmfee=model.getValueAt(selectedrow,7).toString();
hdmfer=model.getValueAt(selectedrow,8).toString();
loan = model.getValueAt(selectedrow,9).toString();
add = model.getValueAt(selectedrow,10).toString();
sub = model.getValueAt(selectedrow,11).toString();
bonus = model.getValueAt(selectedrow,12).toString();
deduction = model.getValueAt(selectedrow,13).toString();
finalwage = model.getValueAt(selectedrow,14).toString();

String databaseURL = "jdbc:derby:SALARY_GENERATOR";
Connection con = DriverManager.getConnection(databaseURL);
Statement st = con.createStatement();

Date d = new Date();
//      Date addeddate = new java.sql.Date(d.getTime());

int imax = maxfinder2();

if (imax==-1){
    throw new Exception("No connection to Salary history established");
}
String strmax = Integer.toString(imax+1);

Double dssser = Double.parseDouble(ssser);
Double dpher = Double.parseDouble(pher);
Double dhdmfer = Double.parseDouble(hdmfer);
Double dwage = Double.parseDouble(wage);
Double dfinaler = dssser + dpher*dwage/100 + dhdmfer*dwage/100;

String sql = "INSERT INTO salary_history (shid, pay_period, eid, datesaved,
monthly_wage_rate, sss_ee, sss_er, ph_ee_p, ph_er_p, hdmf_ee_p, hdmf_er_p, deduction,
bonus, final_wage, final_er) "
+ "VALUES (" +strmax+", "+payperiod+", "+eid+", "+new
java.sql.Timestamp(d.getTime())+", "+wage+", "+sssee+", "+ssser+", "+phee+", "+pher+", "
+
hdmfee+", "+hdmfer+", "+deduction+", "+bonus+", "+finalwage+", "+Double.toString(dfinaler)+)";

st.execute(sql);

imax = maxfinder3();
if (imax==-1){

```

```

        throw new Exception("No connection to Loan history established");
    }
    strmax = Integer.toString(imax+1);

    Double dloan = Double.parseDouble(loan);
    Double dadd = Double.parseDouble(add);
    Double dsub = Double.parseDouble(sub);

    Double dfinal_loan = dloan + dadd - dsub;

    sql = "INSERT INTO loan_history (lhid, pay_period, eid, datesaved, loan, loan_add,
loan_sub, final_loan) "
        + "VALUES (" + strmax + ", " + payperiod + ", " + eid + ", " + new
java.sql.Timestamp(d.getTime()) + ", " + loan + ", " + add + ", " + sub + ", " + Double.toString(dfinal_loan) + ")"
    ;

    st.execute(sql);

    //Update all the records in salary allocations

    sql = "UPDATE salary_allocations SET " +
        "monthly_wage_rate = " + wage + ", " +
        "pay_period = " + Integer.toString(payperiod+1) + ", " +
        "sss_ee = " + sssee + ", " +
        "sss_er = " + ssser + ", " +
        "ph_ee_p = " + phsee + ", " +
        "ph_er_p = " + pher + ", " +
        "hdmf_ee_p = " + hdmfee + ", " +
        "hdmf_er_p = " + hdmfer + ", " +
        "loan = " + Double.toString(dfinal_loan) + ", " +
        "loan_add = " + 0 + ", " +
        "loan_sub = " + 0 + ", " +
        "deduction = " + 0 + ", " +
        "bonus = " + 0 + " " +
        "WHERE eid = " + eid;
    st.execute(sql);

    con.close();
}

refreshtable("SELECT sa.*, " +
    "i.last_name || ', ' || i.first_name || ' ' || COALESCE(i.middle_name, '') AS full_name\n"
+
    "FROM salary_allocations sa\n" +

```

```

        "JOIN info i ON sa.eid = i.eid\n" +
        "WHERE i.hired = TRUE");

        JOptionPane.showMessageDialog(null, "Saved to Salary History and Loan History.");
    }
}
catch (Exception e){
    e.printStackTrace();
}
}

private void ktbtnPrintActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        // Create a print request attribute set to configure print settings
        PrintRequestAttributeSet set = new HashPrintRequestAttributeSet();

        // Get the current date and convert it to a string
        Date d = new Date();
        String sd = d.toString();

        // Create header and footer for the printout
        MessageFormat header = new MessageFormat("Employee Salaries");
        MessageFormat footer = new MessageFormat(sd);

        // Set the print orientation to portrait mode
        set.add(OrientationRequested.PORTRAIT);

        // Print the JTable with the given header and footer
        tblEmployees.print(JTable.PrintMode.FIT_WIDTH, header,
            footer, true, set, true);

        // Show a success message after printing
        JOptionPane.showMessageDialog(null, "\n Printed Successfully");
    }
    catch (java.awt.print.PrinterException e) {
        // Show an error message if printing fails
        JOptionPane.showMessageDialog(null, "\n Printing Failed");
    }
}

private void txtHDMFERKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

```

```

private void txtHDMFERKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtHDMFERActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtHDMFEEKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtHDMFEEKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtHDMFEEActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtPHERKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtPHERKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtPHERActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtPHEEKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtPHEEKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtPHEEActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

```

```

private void txtSSSERKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtSSSERKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtSSSERActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtSSSEEKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtSSSEEKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtSSSEEActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtWageKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtWageKeyReleased(java.awt.event.KeyEvent evt) {
    calcFinalWage();
}

private void txtWageActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtEIDKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtEIDKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

```



```

private void txtEIDActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtLoanActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtLoanKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtLoanKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtLoanSubActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtLoanSubKeyReleased(java.awt.event.KeyEvent evt) {
    calcFinalWage();
}

private void txtLoanSubKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtLoanAddActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtLoanAddKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtLoanAddKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtBonusActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

```

```

private void txtBonusKeyReleased(java.awt.event.KeyEvent evt) {
    calcFinalWage();
}

private void txtBonusKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtDeductionActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtDeductionKeyReleased(java.awt.event.KeyEvent evt) {
    calcFinalWage();
}

private void txtDeductionKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtFinalWageActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtFinalWageKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtFinalWageKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void kbtnClearFieldsActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        txtWage.setText("");
        txtSSSEE.setText("0");
        txtSSSER.setText("0");
        txtPHEE.setText("0");
        txtPHER.setText("0");
        txtHDMFEE.setText("0");
        txtHDMFER.setText("0");
        txtLoan.setText("");
        txtLoanAdd.setText("");
    }
}

```

```

        txtLoanSub.setText("");
        txtDeduction.setText("");
        txtBonus.setText("");
        txtFinalWage.setText("");
    }
    catch(Exception e){

    }
}

private void kbtnDeselectActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        //Deselects all rows
        //      DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
        //      tblEmployees.removeRowSelectionInterval(0, model.getRowCount()-1);
        tblEmployees.clearSelection();
        //resets selectedroweid
        this.selectedroweid = "-1";
    }
    catch (Exception e){

    }
}

private void kbtnUpdateActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        //Initializing Variables
        String sid = "";
        String eid = "";
        String pay_period = "";
        String monthly_wage_rate = "";
        String sssee = "";
        String sssee = "";
        String phee = "";
        String pher = "";
        String hdmfee = "";
        String hdmfer = "";
        String loan = "";
        String loanadd = "";
        String loansub = "";
        String deduction = "";
        String bonus = "";
        String finalwage = "";
    }
    catch (Exception e){

    }
}

```

```

//Get variables from user-input
eid = txtEID.getText();
monthly_wage_rate = txtWage.getText();
sssee = txtSSSEE.getText(); //SSS EE and ER must be validated on their respective
pages
ssser = txtSSSER.getText();
phee = txtPHEE.getText();
pher = txtPHER.getText();
hdmfee = txtHDMFEE.getText();
hdmfer = txtHDMFER.getText();
loan = txtLoan.getText();
loanadd = txtLoanAdd.getText();
loansub = txtLoanSub.getText();
deduction = txtDeduction.getText();
bonus = txtBonus.getText();
finalwage = txtFinalWage.getText();

//Error trapping
if ("".equals(monthly_wage_rate)){
    JOptionPane.showMessageDialog(null, "Wage field is empty.");
}
else if ("".equals(loan)){
    JOptionPane.showMessageDialog(null, "Loan field is empty.");
}
else if ("".equals(loanadd)){
    JOptionPane.showMessageDialog(null, "Loan Add field is empty.");
}
else if ("".equals(loansub)){
    JOptionPane.showMessageDialog(null, "Loans Sub field is empty.");
}
else if ("".equals(deduction)){
    JOptionPane.showMessageDialog(null, "Deduction field is empty.");
}
else if ("".equals(bonus)){
    JOptionPane.showMessageDialog(null, "Bonus field is empty.");
}
else if (Double.parseDouble(monthly_wage_rate)>99999999 ||
Double.parseDouble(monthly_wage_rate)!=((double)Math.round(Double.parseDouble(monthly_
wage_rate)*100)/100){
    JOptionPane.showMessageDialog(null, "Wage can be a maximum of 99,999,999.99.
Decimal precision is only 2 decimal places.");
}
else if (Double.parseDouble(loan)>999999999 ||
Double.parseDouble(loan)!=((double)Math.round(Double.parseDouble(loan)*100)/100){

```

```

        JOptionPane.showMessageDialog(null, "Loan can be a maximum of 99,999,999.99.
        Decimal precision is only 2 decimal places.");
    }
    else if (Double.parseDouble(loanadd)>999999999 ||
    Double.parseDouble(loanadd)!=((double)Math.round(Double.parseDouble(loanadd)*100)/100){
        JOptionPane.showMessageDialog(null, "Loan Add can be a maximum of
        99,999,999.99. Decimal precision is only 2 decimal places.");
    }
    else if (Double.parseDouble(loansub)>999999999 ||
    Double.parseDouble(loansub)!=((double)Math.round(Double.parseDouble(loansub)*100)/100){
        JOptionPane.showMessageDialog(null, "Loan Sub can be a maximum of
        99,999,999.99. Decimal precision is only 2 decimal places.");
    }
    else if (Double.parseDouble(deduction)>999999999 ||
    Double.parseDouble(deduction)!=((double)Math.round(Double.parseDouble(deduction)*100)/100
    ){
        JOptionPane.showMessageDialog(null, "Deduction can be a maximum of
        99,999,999.99. Decimal precision is only 2 decimal places.");
    }
    else if (Double.parseDouble(bonus)>999999999 ||
    Double.parseDouble(bonus)!=((double)Math.round(Double.parseDouble(bonus)*100)/100){
        JOptionPane.showMessageDialog(null, "Bonus can be a maximum of 99,999,999.99.
        Decimal precision is only 2 decimal places.");
    }
    else{
        maxfinder();
        this.max++;
        String strmax = Integer.toString(this.max);

        //Establish connection with the database
        String databaseURL = "jdbc:derby:SALARY_GENERATOR";
        Connection con = DriverManager.getConnection(databaseURL);
        Statement st = con.createStatement();

        //Prepare the SQL statement
        String sql = "UPDATE salary_allocations SET " +
            "monthly_wage_rate = " + monthly_wage_rate + ", " +
            "sss_ee = " + sssee + ", " +
            "sss_er = " + sssee + ", " +
            "ph_ee_p = " + phee + ", " +
            "ph_er_p = " + pher + ", " +
            "hdmf_ee_p = " + hdmfee + ", " +
            "hdmf_er_p = " + hdmfer + ", " +
            "loan = " + loan + ", " +

```

```

        "loan_add = " + loanadd + ", " +
        "loan_sub = " + loansub + ", " +
        "deduction = " + deduction + ", " +
        "bonus = " + bonus + " " +
//      "= " + bonus + ", " +
        "WHERE eid = " + eid;
st.execute(sql);
con.close();

System.out.println("Update successful.");

//No row currently selected.
selectedroweid = "-1";
tblEmployees.clearSelection();

refreshTable("SELECT sa.*, " +
    "i.last_name || ' ' || i.first_name || ' ' || COALESCE(i.middle_name, '') AS full_name\n"
+ //concatenate name columns from emp records
    "FROM salary_allocations sa\n" +
    "JOIN emp e ON sa.eid = e.eid\n" + //inner join statement to access data from
salary_allocations and emp
    "WHERE e.hired = TRUE"); //making sure this is only done for hired employees.
maxfinder();

JOptionPane.showMessageDialog(null, "Updated.");

txtEID.setText("");
txtWage.setText("");
txtSSSEE.setText("0");
txtSSSER.setText("0");
txtPHEE.setText("0");
txtPHER.setText("0");
txtHDMFEE.setText("0");
txtHDMFER.setText("0");
txtLoan.setText("");
txtLoanAdd.setText("");
txtLoanSub.setText("");
txtDeduction.setText("");
txtBonus.setText("");
txtFinalWage.setText("");

```

```

        cmbSSSBBracket.setSelectedIndex(0);
        cmbPHBracket.setSelectedIndex(0);
        cmbHDMFBracket.setSelectedIndex(0);
    }
}
catch (Exception e){
    e.printStackTrace();
}
}

private void kbtnSalaryHistoryActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        new SalaryHistory(this.uid,this.email).setVisible(true);
        this.setVisible(false);
    }
    catch (Exception e){

    }
}

private void kbtnLoanHistoryActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        new LoanHistory(this.uid,this.email).setVisible(true);
        this.setVisible(false);
    }
    catch (Exception e){

    }
}

private void kbtnEditEmpListActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        new EmployeeRecords(this.uid,this.email).setVisible(true);
        this.setVisible(false);
    }
    catch (Exception e){

    }
}

private void cmbSSSBBracketActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

```

```

private void cmbSSSBraKetItemStateChanged(java.awt.event.ItemEvent evt) {

    try{
        String selectedBracket = (String) cmbSSSBraKet.getSelectedItem();
        txtSSSEE.setText("");
        txtSSSER.setText("");
        if (selectedBracket.equals("-SELECT SSS-")) {
            txtSSSEE.setText("0"); //back to 0.
            txtSSSER.setText("0");

            return;
        }

        String[] bounds = selectedBracket.split(" - ");
        double lowerBound = Double.parseDouble(bounds[0]);
        double upperBound = Double.parseDouble(bounds[1]);

        String databaseURL = "jdbc:derby:SALARY_GENERATOR";
        Connection con = DriverManager.getConnection(databaseURL);

        String sql;
        Statement st = con.createStatement();
        PreparedStatement pstmt = con.prepareStatement("SELECT sss_ee, sss_er FROM sss
WHERE bracketlowerbound = ? AND bracketupperbound = ?");

        pstmt.setDouble(1, lowerBound);
        pstmt.setDouble(2, upperBound);

        try (ResultSet rs = pstmt.executeQuery()) {
            if (rs.next()) {
                txtSSSEE.setText(String.valueOf(rs.getDouble("sss_ee")));
                txtSSSER.setText(String.valueOf(rs.getDouble("sss_er")));
            }
        }

        calcFinalWage();
    }
    catch(SQLException e){
        e.printStackTrace();
        JOptionPane.showMessageDialog(this, "Error retrieving data: " + e.getMessage());
    }
}

```



```

private void cmbPHBracketItemStateChanged(java.awt.event.ItemEvent evt) {
    try{
        String selectedBracket = (String) cmbPHBracket.getSelectedItem();
        txtPHEE.setText("");
        txtPHER.setText("");
        if (selectedBracket.equals("-SELECT PH-")) {
            txtPHEE.setText("0"); //back to 0.
            txtPHER.setText("0");

            return;
        }

        String[] bounds = selectedBracket.split(" - ");
        double lowerBound = Double.parseDouble(bounds[0]);
        double upperBound = Double.parseDouble(bounds[1]);

        String databaseURL = "jdbc:derby:SALARY_GENERATOR";
        Connection con = DriverManager.getConnection(databaseURL);

        String sql;
        Statement st = con.createStatement();
        PreparedStatement pstmt = con.prepareStatement("SELECT ph_ee_p, ph_er_p FROM
philhealth WHERE bracketlowerbound = ? AND bracketupperbound = ?");

        pstmt.setDouble(1, lowerBound);
        pstmt.setDouble(2, upperBound);

        try (ResultSet rs = pstmt.executeQuery()) {
            if (rs.next()) {
                txtPHEE.setText(String.valueOf(rs.getDouble("ph_ee_p")));
                txtPHER.setText(String.valueOf(rs.getDouble("ph_er_p")));
            }
        }

        calcFinalWage();
    }
    catch(SQLException e){
        e.printStackTrace();
        JOptionPane.showMessageDialog(this, "Error retrieving data: " + e.getMessage());
    }
}

private void cmbHDMFBracketItemStateChanged(java.awt.event.ItemEvent evt) {

```

```

try{
    String selectedBracket = (String) cmbHDMFBracket.getSelectedItem();
    txtHDMFEE.setText("");
    txtHDMFER.setText("");
    if (selectedBracket.equals("-SELECT HDMF-")) {
        txtHDMFEE.setText("0"); //back to 0.
        txtHDMFER.setText("0");

        return;
    }

    String[] bounds = selectedBracket.split(" - ");
    double lowerBound = Double.parseDouble(bounds[0]);
    double upperBound = Double.parseDouble(bounds[1]);

    String databaseURL = "jdbc:derby:SALARY_GENERATOR";
    Connection con = DriverManager.getConnection(databaseURL);

    String sql;
    Statement st = con.createStatement();
    PreparedStatement pstmt = con.prepareStatement("SELECT hdmf_ee_p, hdmf_er_p
FROM hdmf WHERE bracketlowerbound = ? AND bracketupperbound = ?");

    pstmt.setDouble(1, lowerBound);
    pstmt.setDouble(2, upperBound);

    try (ResultSet rs = pstmt.executeQuery()) {
        if (rs.next()) {
            txtHDMFEE.setText(String.valueOf(rs.getDouble("hdmf_ee_p")));
            txtHDMFER.setText(String.valueOf(rs.getDouble("hdmf_er_p")));
        }
    }

    calcFinalWage();
}
catch(SQLException e){
    e.printStackTrace();
    JOptionPane.showMessageDialog(this, "Error retrieving data: " + e.getMessage());
}
}

private void cmbHDMFBracketActionPerformed(java.awt.event.ActionEvent evt) {

```

```

    // TODO add your handling code here:
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Register().setVisible(true);
        }
    }

```

```
});  
}
```

```
// Variables declaration - do not modify  
private javax.swing.JComboBox<String> cmbColumn;  
private javax.swing.JComboBox<String> cmbHDMFBracket;  
private javax.swing.JComboBox<String> cmbIn;  
private javax.swing.JComboBox<String> cmbOrder;  
private javax.swing.JComboBox<String> cmbPHBracket;  
private javax.swing.JComboBox<String> cmbSSSBracket;  
private javax.swing.JScrollPane jScrollPane1;  
private com.k33ptoo.components.KButton kbtnClearFields;  
private com.k33ptoo.components.KButton kbtnDeselect;  
private com.k33ptoo.components.KButton kbtnEditEmpList;  
private com.k33ptoo.components.KButton kbtnEditHDMF;  
private com.k33ptoo.components.KButton kbtnEditPH;  
private com.k33ptoo.components.KButton kbtnEditSSS;  
private com.k33ptoo.components.KButton kbtnLoanHistory;  
private com.k33ptoo.components.KButton kbtnLogout;  
private com.k33ptoo.components.KButton kbtnReset;  
private com.k33ptoo.components.KButton kbtnSalaryHistory;  
private com.k33ptoo.components.KButton kbtnSave;  
private com.k33ptoo.components.KButton kbtnSort;  
private com.k33ptoo.components.KButton kbtnUpdate;  
private com.k33ptoo.components.KButton kbtnPrint;  
private javax.swing.JLabel lblActiveEmployees;  
private javax.swing.JLabel lblBonus;  
private javax.swing.JLabel lblDeduction;  
private javax.swing.JLabel lblEID;  
private javax.swing.JLabel lblEmail;  
private javax.swing.JLabel lblFinalWage;  
private javax.swing.JLabel lblHDMFBracket;  
private javax.swing.JLabel lblHDMFEE;  
private javax.swing.JLabel lblHDMFER;  
private javax.swing.JLabel lblLoan;  
private javax.swing.JLabel lblLoanAdd;  
private javax.swing.JLabel lblLoanSub;  
private javax.swing.JLabel lblPHBracket;  
private javax.swing.JLabel lblPHEE;  
private javax.swing.JLabel lblPHER;  
private javax.swing.JLabel lblSSSBracket;  
private javax.swing.JLabel lblSSSEE;  
private javax.swing.JLabel lblSSSER;  
private javax.swing.JLabel lblSearch;
```

```

private javax.swing.JLabel lblTotalER;
private javax.swing.JLabel lblTotalWages;
private javax.swing.JLabel lblWage;
private com.k33ptoo.components.KGradientPanel pnlBackground;
private javax.swing.JPanel pnlDelete;
private javax.swing.JPanel pnlDelete1;
private javax.swing.JPanel pnlEmail;
private javax.swing.JPanel pnlEmail1;
private javax.swing.JPanel pnlSearch;
private javax.swing.JPanel pnlSearch1;
private javax.swing.JPanel pnlSearch2;
private javax.swing.JPanel pnlTable;
private javax.swing.JTable tblEmployees;
private javax.swing.JTextField txtActiveEmployees;
private javax.swing.JTextField txtBonus;
private javax.swing.JTextField txtDeduction;
private javax.swing.JTextField txtEID;
private javax.swing.JTextField txtFinalWage;
private javax.swing.JTextField txtHDMFEE;
private javax.swing.JTextField txtHDMFER;
private javax.swing.JLabel txtIn;
private javax.swing.JTextField txtLoan;
private javax.swing.JTextField txtLoanAdd;
private javax.swing.JTextField txtLoanSub;
private javax.swing.JTextField txtPHEE;
private javax.swing.JTextField txtPHER;
private javax.swing.JTextField txtSSSEE;
private javax.swing.JTextField txtSSSER;
private javax.swing.JTextField txtSearch;
private javax.swing.JLabel txtSort;
private javax.swing.JLabel txtSort1;
private javax.swing.JLabel txtSort2;
private javax.swing.JTextField txtTotalER;
private javax.swing.JTextField txtTotalWages;
private javax.swing.JTextField txtWage;
// End of variables declaration
}

```

Manipulate Employee Records Page:

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
 license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */
```

```
/**
 *
 * @author User
 */
import java.awt.Color;
import java.sql.Connection;
import java.sql.DatabaseMetaData;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.swing.JOptionPane;
import javax.swing.RowFilter;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableRowSorter;

import javax.swing.text.*;
```

```
public class EmployeeRecords extends javax.swing.JFrame {
```

```
    /**
     * Creates new form Register
     */
```

```
    //Global Variables
    int max = 0;
    int salarymax = 0;
    int historymax = 0;
    String email = "", uid = "";
    String selectedroweid = "-1";
```

```

//Setting up the regexfilter
TableRowSorter sorter;

private void initComponents2(){
    //Overwrite Java Driver Code

    //Initializing the table with specific settings
    //For the regexfilter later on
    TableRowSorter sorter = new
TableRowSorter<>((DefaultTableModel)tblEmployees.getModel());
    this.sorter = sorter;
    tblEmployees.setRowSorter(sorter);
    //Disables table sorting
    sorter.setSortable(0, false);
    sorter.setSortable(1, false);
    sorter.setSortable(2, false);
    sorter.setSortable(3, false);
    //Disables rearranging table columns
    tblEmployees.getTableHeader().setReorderingAllowed(false);
    //Textboxes set to a clear background
    /*txtFullName.setBackground(new Color(0, 0, 0, 0));
    txtEmail.setBackground(new Color(0, 0, 0, 0));
    txtPhoneNumber.setBackground(new Color(0, 0, 0, 0));*/

    //Adding numeric filters to necessary textboxes
    ((AbstractDocument) txtEID.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
    ((AbstractDocument) txtAge.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
    ((AbstractDocument) txtHeight.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
    ((AbstractDocument) txtWeight.getDocument()).setDocumentFilter(new
NumericDocumentFilter());

    String maxeid = Integer.toString(this.max+1);
    txtEID.setText(maxeid);

    jCalendarBirthdate.setDate(new Date(70,1,1));
}

public static boolean doesTableExists(String tableName, Connection conn)
    throws SQLException
{

```

```

        DatabaseMetaData meta = conn.getMetaData();
        ResultSet result = meta.getTables(null, null, tableName.toUpperCase(), null);

        return result.next();
    }

    private void maxfinder(){ //figure out the highest eid
        try{
            String databaseURL = "jdbc:derby:SALARY_GENERATOR";
            Connection con = DriverManager.getConnection(databaseURL);

            String sql;
            Statement st = con.createStatement();

            sql = "SELECT * FROM info";
            ResultSet rs = st.executeQuery(sql);

            //db stands for database

            int max = 0;
            while (rs.next()==true){
                String dbeid = rs.getString(1);
                int eid = Integer.parseInt(dbeid);
                if (eid>max){
                    max = eid;
                }
            }
            this.max = max;
            con.close();
        }
        catch (Exception e){
            e.printStackTrace();
        }
    }

    private void maxfinder2(){ //figure out the highest eid
        try{
            String databaseURL = "jdbc:derby:SALARY_GENERATOR";
            Connection con = DriverManager.getConnection(databaseURL);

            String sql;
            Statement st = con.createStatement();

            sql = "SELECT * FROM salary_allocations";

```



```

        ResultSet rs = st.executeQuery(sql);

        //db stands for database

        int max = 0;
        while (rs.next()==true){
            String dbeid = rs.getString(1);
            int eid = Integer.parseInt(dbeid);
            if (eid>max){
                max = eid;
            }
        }
        this.salarymax = max;
        con.close();
    }
    catch (Exception e){
        e.printStackTrace();
    }
}

private void maxfinder3(){ //figure out the highest eid
    try{
        String databaseURL = "jdbc:derby:SALARY_GENERATOR";
        Connection con = DriverManager.getConnection(databaseURL);

        String sql;
        Statement st = con.createStatement();

        sql = "SELECT * FROM history";
        ResultSet rs = st.executeQuery(sql);

        //db stands for database

        int max = 0;
        while (rs.next()==true){
            String dbeid = rs.getString(2); //finds the highest pay period
            int eid = Integer.parseInt(dbeid);
            if (eid>max){
                max = eid;
            }
        }
        this.historymax = max;
        con.close();
    }
}

```

```

    }
    catch (Exception e){
        e.printStackTrace();
    }
}

private void search(){
    try{
        int in = cmbIn.getSelectedIndex();

        String search = txtSearch.getText().trim();

        //Search using RowFilter and regex (regular expression) to find substrings
        RowFilter<DefaultTableModel, Object> rf = null;
        try {
            if (in == 0){
                rf = RowFilter.regexFilter("(?i)" + search); //("(?i)" for case insensitive filter
            }
            else{
                rf = RowFilter.regexFilter("(?i)" + search, in - 1);
            }
        }
        catch (java.util.regex.PatternSyntaxException e) {
            //      System.out.println("Error Found");
            return;
        }
        sorter.setRowFilter(rf);
    }
    catch (Exception e){
        e.printStackTrace();
    }
}

private void refreshtable(String sql){//reset table contents, works for SELECT queries only
    try{
        DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
        //Establish connection to the database
        String databaseURL = "jdbc:derby:SALARY_GENERATOR";
        Connection con = DriverManager.getConnection(databaseURL);

        Statement st = con.createStatement();

        //Place the results into a ResultSet
        ResultSet rs = st.executeQuery(sql);
    }
}

```

```

        model.setRowCount(0); //reset the table
        while (rs.next()==true){
            //Add the data into the jTable
            model.addRow(new
Object[]{rs.getString(1),rs.getBoolean(3),rs.getString(4),rs.getString(5),rs.getString(6),
            rs.getString(8),rs.getString(7),rs.getString(9),rs.getString(10),rs.getString(11),
            rs.getString(12),rs.getString(13),rs.getString(14)});
        }
        con.close();

    }
    catch (Exception e){
        System.out.println("Refresh table exception");
        e.printStackTrace();
    }
}

```

```

public EmployeeRecords() {
    initComponents();

    try{
//        this.email = email;
        String currentURL = "jdbc:derby:SALARY_GENERATOR;create=true";
        Connection con = DriverManager.getConnection(currentURL);

        Statement st = con.createStatement();

//        System.out.println("Reaches here");

        uid = "salary_allocation"; //table names can't start with a number
//        this.uid = uid;
//        System.out.println(uid);

//        System.out.println("went past if condition");
        con.close();

        maxfinder();

        refreshtable("select * from info");

        initComponents2();
    }
}

```

```

    }
    catch (Exception e){
        e.printStackTrace();
    }
}

public EmployeeRecords(String uid, String email){ //CATCH ALL INCOMING DATA
    initComponents();
    try{ //attempt to execute the code
        this.email = email;
        String currentURL = "jdbc:derby:SALARY_GENERATOR;create=true";
        Connection con = DriverManager.getConnection(currentURL);

        Statement st = con.createStatement();

        uid = "salary_allocation"; //table names can't start with a number
        this.uid = uid;

        con.close();

        maxfinder();

        refreshtable("select * from info");

        initComponents2();
    }
    catch (Exception e){ //catch any Exception, regardless of type
        e.printStackTrace(); //print the exact error
    }
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    pnlBackground = new com.k33ptoo.components.KGradientPanel();
    pnlSearch = new javax.swing.JPanel();
    txtSearch = new javax.swing.JTextField();

```

```

lblSearch = new javax.swing.JLabel();
cmbIn = new javax.swing.JComboBox<>();
txtIn = new javax.swing.JLabel();
txtSort = new javax.swing.JLabel();
cmbColumn = new javax.swing.JComboBox<>();
cmbOrder = new javax.swing.JComboBox<>();
kbtnSort = new com.k33ptoo.components.KButton();
kbtnReset = new com.k33ptoo.components.KButton();
pnlTable = new javax.swing.JPanel();
jScrollPane1 = new javax.swing.JScrollPane();
tblEmployees = new javax.swing.JTable();
pnlEmail = new javax.swing.JPanel();
lblEditPersonalRecords = new javax.swing.JLabel();
pnlDelete = new javax.swing.JPanel();
kbtnDeleteAll = new com.k33ptoo.components.KButton();
kbtnDelete = new com.k33ptoo.components.KButton();
pnlDelete1 = new javax.swing.JPanel();
txtEID = new javax.swing.JTextField();
lblEID = new javax.swing.JLabel();
txtFirstName = new javax.swing.JTextField();
lblFirstName = new javax.swing.JLabel();
txtHeight = new javax.swing.JTextField();
lblOptionalFields = new javax.swing.JLabel();
txtWeight = new javax.swing.JTextField();
lblNotes = new javax.swing.JLabel();
txtNotes = new javax.swing.JTextField();
kbtnClearFields = new com.k33ptoo.components.KButton();
kbtnDeselect = new com.k33ptoo.components.KButton();
kbtnUpdate = new com.k33ptoo.components.KButton();
kbtnAdd = new com.k33ptoo.components.KButton();
cbHired = new javax.swing.JCheckBox();
lblMiddleName = new javax.swing.JLabel();
txtMiddleName = new javax.swing.JTextField();
lblLastName = new javax.swing.JLabel();
txtLastName = new javax.swing.JTextField();
lblBirthdate = new javax.swing.JLabel();
lblAge = new javax.swing.JLabel();
txtAge = new javax.swing.JTextField();
txtAddress = new javax.swing.JTextField();
lblAddress = new javax.swing.JLabel();
txtPhoneNumber = new javax.swing.JTextField();
lblPhoneNumber = new javax.swing.JLabel();
txtEmail = new javax.swing.JTextField();
lblWeight = new javax.swing.JLabel();

```

```

lblHeight = new javax.swing.JLabel();
lblEmail = new javax.swing.JLabel();
jCalendarBirthdate = new de.wannawork.jcalendar.JCalendarComboBox();
pnlEmail1 = new javax.swing.JPanel();
kbtnSalaryGenerator = new com.k33ptoo.components.KButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setBackground(new java.awt.Color(255, 255, 255));
setResizable(false);

pnlBackground.setkEndColor(new java.awt.Color(66, 133, 244));
pnlBackground.setkGradientFocus(900);
pnlBackground.setkStartColor(new java.awt.Color(66, 133, 244));

pnlSearch.setBackground(new java.awt.Color(238, 238, 238));
pnlSearch.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
pnlSearch.setForeground(new java.awt.Color(36, 48, 94));

txtSearch.setBackground(new java.awt.Color(252, 250, 255));
txtSearch.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtSearch.setForeground(new java.awt.Color(8, 8, 8));
txtSearch.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(38, 117, 247)));
txtSearch.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtSearch.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtSearchActionPerformed(evt);
    }
});
txtSearch.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtSearchKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtSearchKeyTyped(evt);
    }
});

lblSearch.setIcon(new javax.swing.ImageIcon(getClass().getResource("/29x29
Search.png"))); // NOI18N
lblSearch.setText("jLabel2");

```

```

        cmbIn.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "All", "EID",
"Hired", "First Name", "Middle Name", "Last Name", "Birthdate", "Age", "Address", "Phone
Number", "Email", "Height", "Weight", "Notes" }));
        cmbIn.addItemListener(new java.awt.event.ItemListener() {
            public void itemStateChanged(java.awt.event.ItemEvent evt) {
                cmbInItemStateChanged(evt);
            }
        });
        cmbIn.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                cmbInActionPerformed(evt);
            }
        });
        cmbIn.addPropertyChangeListener(new java.beans.PropertyChangeListener() {
            public void propertyChange(java.beans.PropertyChangeEvent evt) {
                cmbInPropertyChange(evt);
            }
        });
    };

```

```

txtIn.setFont(new java.awt.Font("Segoe UI Light", 0, 16)); // NOI18N
txtIn.setForeground(new java.awt.Color(8, 8, 8));
txtIn.setText("In:");

```

```

txtSort.setFont(new java.awt.Font("Segoe UI Light", 0, 16)); // NOI18N
txtSort.setForeground(new java.awt.Color(8, 8, 8));
txtSort.setText("Sort By:");

```

```

        cmbColumn.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "EID",
"Hired", "First Name", "Middle Name", "Last Name", "Birthdate", "Age", "Address", "Phone
Number", "Email", "Height", "Weight", "Notes" }));
        cmbColumn.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                cmbColumnActionPerformed(evt);
            }
        });
    };

```

```

        cmbOrder.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {
"Ascending", "Descending" }));
    };

```

```

kbtnSort.setText("Sort");
kbtnSort.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnSort.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnSort.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnSort.setkHoverStartColor(new java.awt.Color(76, 111, 167));

```

```

kbtnSort.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnSort.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnSort.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnSortActionPerformed(evt);
    }
});

kbtnReset.setText("Reset");
kbtnReset.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnReset.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnReset.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnReset.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnReset.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnReset.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnResetActionPerformed(evt);
    }
});

javax.swing.GroupLayout pnlSearchLayout = new javax.swing.GroupLayout(pnlSearch);
pnlSearch.setLayout(pnlSearchLayout);
pnlSearchLayout.setHorizontalGroup(
    pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlSearchLayout.createSequentialGroup()
            .add(pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .add(pnlSearchLayout.createSequentialGroup()
                    .addContainerGap()
                    .addComponent(lblSearch, javax.swing.GroupLayout.PREFERRED_SIZE, 29,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(txtSearch, javax.swing.GroupLayout.PREFERRED_SIZE, 156,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(txtIn)
                    .addGap(18, 18, 18)
                    .addComponent(cmbIn, javax.swing.GroupLayout.PREFERRED_SIZE, 80,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(txtSort)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(cmbColumn, 0, 244, Short.MAX_VALUE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(cmbOrder, javax.swing.GroupLayout.PREFERRED_SIZE, 84,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

```



```

        .addComponent(kbtnSort, javax.swing.GroupLayout.PREFERRED_SIZE, 55,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(kbtnReset, javax.swing.GroupLayout.PREFERRED_SIZE, 58,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap())
    );
    pnlSearchLayout.setVerticalGroup(
        pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlSearchLayout.createSequentialGroup()
            .addGap(5, 5, 5)

.addGroup(pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addGroup(pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
            .addComponent(kbtnSort, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, 0, Short.MAX_VALUE)

.addGroup(pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(cmbOrder, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(cmbColumn, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(kbtnReset, javax.swing.GroupLayout.PREFERRED_SIZE,
27, javax.swing.GroupLayout.PREFERRED_SIZE)))

.addGroup(pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(lblSearch)

.addGroup(pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(txtSearch, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(cmbIn, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(txtIn)
            .addComponent(txtSort))))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );

```

```

pnlTable.setBackground(new java.awt.Color(55, 71, 133));
pnlTable.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
pnlTable.setForeground(new java.awt.Color(36, 48, 94));

tblEmployees.setAutoCreateRowSorter(true);
tblEmployees.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {

        },
    new String [] {
        "EID", "Hired", "First Name", "Middle Name", "Last Name", "Birthdate", "Age",
"Address", "Phone Number", "Email", "Height", "Weight", "Notes"
    }
) {
    boolean[] canEdit = new boolean [] {
        false, false, false, false, false, false, false, false, false, false, false, false
    };

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});
tblEmployees.setToolTipText("");
tblEmployees.setAutoResizeMode(javax.swing.JTable.AUTO_RESIZE_LAST_COLUMN);
tblEmployees.getTableHeader().setReorderingAllowed(false);
tblEmployees.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        tblEmployeesMouseClicked(evt);
    }
});
jScrollPane1.setViewportViewView(tblEmployees);
if (tblEmployees.getColumnModel().getColumnCount() > 0) {
    tblEmployees.getColumnModel().getColumn(0).setResizable(false);
    tblEmployees.getColumnModel().getColumn(1).setResizable(false);
    tblEmployees.getColumnModel().getColumn(2).setResizable(false);
    tblEmployees.getColumnModel().getColumn(3).setResizable(false);
    tblEmployees.getColumnModel().getColumn(4).setResizable(false);
    tblEmployees.getColumnModel().getColumn(5).setResizable(false);
    tblEmployees.getColumnModel().getColumn(6).setResizable(false);
    tblEmployees.getColumnModel().getColumn(7).setResizable(false);
    tblEmployees.getColumnModel().getColumn(8).setResizable(false);
    tblEmployees.getColumnModel().getColumn(9).setResizable(false);
    tblEmployees.getColumnModel().getColumn(10).setResizable(false);
}

```

```

tblEmployees.getColumnModel().getColumn(11).setResizable(false);
tblEmployees.getColumnModel().getColumn(12).setResizable(false);
}

javax.swing.GroupLayout pnlTableLayout = new javax.swing.GroupLayout(pnlTable);
pnlTable.setLayout(pnlTableLayout);
pnlTableLayout.setHorizontalGroup(
    pnlTableLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlTableLayout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 574,
Short.MAX_VALUE)
            .addGap(10, 10, 10)
        )
);

pnlEmail.setBackground(new java.awt.Color(238, 238, 238));
pnlEmail.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
pnlEmail.setForeground(new java.awt.Color(36, 48, 94));

lblEditPersonalRecords.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblEditPersonalRecords.setForeground(new java.awt.Color(8, 8, 8));
lblEditPersonalRecords.setText("Edit Personal Records");

javax.swing.GroupLayout pnlEmailLayout = new javax.swing.GroupLayout(pnlEmail);
pnlEmail.setLayout(pnlEmailLayout);
pnlEmailLayout.setHorizontalGroup(
    pnlEmailLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlEmailLayout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addComponent(lblEditPersonalRecords, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addGap(10, 10, 10)
        )
);

pnlEmailLayout.setVerticalGroup(
    pnlEmailLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlEmailLayout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addComponent(lblEditPersonalRecords, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addGap(10, 10, 10)
        )
);

```

```

        .addContainerGap()
        .addComponent(lblEditPersonalRecords,
javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );

    pnlDelete.setBackground(new java.awt.Color(238, 238, 238));
    pnlDelete.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
    pnlDelete.setForeground(new java.awt.Color(36, 48, 94));

    kbtnDeleteAll.setText("Delete All");
    kbtnDeleteAll.setEndColor(new java.awt.Color(28, 84, 140));
    kbtnDeleteAll.setHoverEndColor(new java.awt.Color(76, 111, 167));
    kbtnDeleteAll.setHoverForeGround(new java.awt.Color(252, 250, 255));
    kbtnDeleteAll.setHoverStartColor(new java.awt.Color(76, 111, 167));
    kbtnDeleteAll.setkSelectedColor(new java.awt.Color(252, 250, 255));
    kbtnDeleteAll.setkStartColor(new java.awt.Color(28, 84, 140));
    kbtnDeleteAll.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            kbtnDeleteAllActionPerformed(evt);
        }
    });

    kbtnDelete.setText("Delete");
    kbtnDelete.setEndColor(new java.awt.Color(28, 84, 140));
    kbtnDelete.setHoverEndColor(new java.awt.Color(76, 111, 167));
    kbtnDelete.setHoverForeGround(new java.awt.Color(252, 250, 255));
    kbtnDelete.setHoverStartColor(new java.awt.Color(76, 111, 167));
    kbtnDelete.setkSelectedColor(new java.awt.Color(252, 250, 255));
    kbtnDelete.setkStartColor(new java.awt.Color(28, 84, 140));
    kbtnDelete.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            kbtnDeleteActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout pnlDeleteLayout = new javax.swing.GroupLayout(pnlDelete);
    pnlDelete.setLayout(pnlDeleteLayout);
    pnlDeleteLayout.setHorizontalGroup(
        pnlDeleteLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(pnlDeleteLayout.createSequentialGroup()
                .addContainerGap()

```

```

        .addComponent(kbtnDeleteAll, javax.swing.GroupLayout.PREFERRED_SIZE, 78,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(kbtnDelete, javax.swing.GroupLayout.PREFERRED_SIZE, 78,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap()
    );
    pnlDeleteLayout.setVerticalGroup(
        pnlDeleteLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlDeleteLayout.createSequentialGroup()
        .addContainerGap()

.addGroup(pnlDeleteLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELIN
E)
        .addComponent(kbtnDeleteAll, javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(kbtnDelete, javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap(10, Short.MAX_VALUE))
    );

    pnlDelete1.setBackground(new java.awt.Color(238, 238, 238));
    pnlDelete1.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
    pnlDelete1.setForeground(new java.awt.Color(36, 48, 94));

    txtEID.setBackground(new java.awt.Color(252, 250, 255));
    txtEID.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
    txtEID.setForeground(new java.awt.Color(8, 8, 8));
    txtEID.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
    txtEID.setFocusable(false);
    txtEID.setMargin(new java.awt.Insets(2, 2, 2, 2));
    txtEID.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            txtEIDActionPerformed(evt);
        }
    });
    txtEID.addKeyListener(new java.awt.event.KeyAdapter() {
        public void keyReleased(java.awt.event.KeyEvent evt) {
            txtEIDKeyReleased(evt);
        }
        public void keyTyped(java.awt.event.KeyEvent evt) {

```

```

        txtEIDKeyTyped(evt);
    }
});

lblEID.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblEID.setForeground(new java.awt.Color(8, 8, 8));
lblEID.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblEID.setText("EID:");

txtFirstName.setBackground(new java.awt.Color(252, 250, 255));
txtFirstName.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtFirstName.setForeground(new java.awt.Color(8, 8, 8));
txtFirstName.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtFirstName.setFocusTraversalPolicyProvider(true);
txtFirstName.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtFirstName.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtFirstNameActionPerformed(evt);
    }
});
txtFirstName.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtFirstNameKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtFirstNameKeyTyped(evt);
    }
});

lblFirstName.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblFirstName.setForeground(new java.awt.Color(8, 8, 8));
lblFirstName.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblFirstName.setText("First Name:");

txtHeight.setBackground(new java.awt.Color(252, 250, 255));
txtHeight.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtHeight.setForeground(new java.awt.Color(8, 8, 8));
txtHeight.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtHeight.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtHeight.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtHeightActionPerformed(evt);
    }
});

```

```

    }
});
txtHeight.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtHeightKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtHeightKeyTyped(evt);
    }
});

lblOptionalFields.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblOptionalFields.setForeground(new java.awt.Color(8, 8, 8));
lblOptionalFields.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblOptionalFields.setText("Optional Fields:");

txtWeight.setBackground(new java.awt.Color(252, 250, 255));
txtWeight.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtWeight.setForeground(new java.awt.Color(8, 8, 8));
txtWeight.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtWeight.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtWeight.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtWeightActionPerformed(evt);
    }
});
txtWeight.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtWeightKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtWeightKeyTyped(evt);
    }
});

lblNotes.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblNotes.setForeground(new java.awt.Color(8, 8, 8));
lblNotes.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblNotes.setText("Notes:");

txtNotes.setBackground(new java.awt.Color(252, 250, 255));
txtNotes.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtNotes.setForeground(new java.awt.Color(8, 8, 8));

```

```

txtNotes.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtNotes.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtNotes.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtNotesActionPerformed(evt);
    }
});
txtNotes.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtNotesKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtNotesKeyTyped(evt);
    }
});

kbtnClearFields.setText("Clear Fields");
kbtnClearFields.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnClearFields.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnClearFields.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnClearFields.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnClearFields.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnClearFields.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnClearFields.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnClearFieldsActionPerformed(evt);
    }
});

kbtnDeselect.setText("Deselect");
kbtnDeselect.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnDeselect.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnDeselect.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnDeselect.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnDeselect.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnDeselect.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnDeselect.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnDeselectActionPerformed(evt);
    }
});

kbtnUpdate.setText("Update");

```



```

kbtnUpdate.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnUpdate.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnUpdate.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnUpdate.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnUpdate.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnUpdate.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnUpdate.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnUpdateActionPerformed(evt);
    }
});

```

```

kbtnAdd.setText("Add");
kbtnAdd.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnAdd.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnAdd.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnAdd.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnAdd.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnAdd.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnAdd.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnAddActionPerformed(evt);
    }
});

```

```

cbHired.setText("Hired / Not");

```

```

lblMiddleName.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblMiddleName.setForeground(new java.awt.Color(8, 8, 8));
lblMiddleName.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblMiddleName.setText("Middle Name:");

```

```

txtMiddleName.setBackground(new java.awt.Color(252, 250, 255));
txtMiddleName.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtMiddleName.setForeground(new java.awt.Color(8, 8, 8));
txtMiddleName.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtMiddleName.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtMiddleName.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtMiddleNameActionPerformed(evt);
    }
});
txtMiddleName.addKeyListener(new java.awt.event.KeyAdapter() {

```

```

        public void keyReleased(java.awt.event.KeyEvent evt) {
            txtMiddleNameKeyReleased(evt);
        }
        public void keyTyped(java.awt.event.KeyEvent evt) {
            txtMiddleNameKeyTyped(evt);
        }
    });

    lblLastName.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
    lblLastName.setForeground(new java.awt.Color(8, 8, 8));
    lblLastName.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
    lblLastName.setText("Last Name:");

    txtLastName.setBackground(new java.awt.Color(252, 250, 255));
    txtLastName.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
    txtLastName.setForeground(new java.awt.Color(8, 8, 8));
    txtLastName.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
    txtLastName.setMargin(new java.awt.Insets(2, 2, 2, 2));
    txtLastName.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            txtLastNameActionPerformed(evt);
        }
    });
    txtLastName.addKeyListener(new java.awt.event.KeyAdapter() {
        public void keyReleased(java.awt.event.KeyEvent evt) {
            txtLastNameKeyReleased(evt);
        }
        public void keyTyped(java.awt.event.KeyEvent evt) {
            txtLastNameKeyTyped(evt);
        }
    });

    lblBirthdate.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
    lblBirthdate.setForeground(new java.awt.Color(8, 8, 8));
    lblBirthdate.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
    lblBirthdate.setText("Birthdate:");

    lblAge.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
    lblAge.setForeground(new java.awt.Color(8, 8, 8));
    lblAge.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
    lblAge.setText("Age (years:)");

    txtAge.setBackground(new java.awt.Color(252, 250, 255));

```

```

txtAge.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtAge.setForeground(new java.awt.Color(8, 8, 8));
txtAge.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtAge.setFocusable(false);
txtAge.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtAge.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtAgeActionPerformed(evt);
    }
});
txtAge.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtAgeKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtAgeKeyTyped(evt);
    }
});

txtAddress.setBackground(new java.awt.Color(252, 250, 255));
txtAddress.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtAddress.setForeground(new java.awt.Color(8, 8, 8));
txtAddress.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtAddress.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtAddress.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtAddressActionPerformed(evt);
    }
});
txtAddress.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtAddressKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtAddressKeyTyped(evt);
    }
});

lblAddress.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblAddress.setForeground(new java.awt.Color(8, 8, 8));
lblAddress.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblAddress.setText("Address:");

```

```

txtPhoneNumber.setBackground(new java.awt.Color(252, 250, 255));
txtPhoneNumber.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtPhoneNumber.setForeground(new java.awt.Color(8, 8, 8));
txtPhoneNumber.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtPhoneNumber.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtPhoneNumber.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtPhoneNumberActionPerformed(evt);
    }
});
txtPhoneNumber.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtPhoneNumberKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtPhoneNumberKeyTyped(evt);
    }
});

```

```

lblPhoneNumber.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblPhoneNumber.setForeground(new java.awt.Color(8, 8, 8));
lblPhoneNumber.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblPhoneNumber.setText("Phone Number:");

```

```

txtEmail.setBackground(new java.awt.Color(252, 250, 255));
txtEmail.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtEmail.setForeground(new java.awt.Color(8, 8, 8));
txtEmail.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtEmail.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtEmail.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtEmailActionPerformed(evt);
    }
});
txtEmail.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtEmailKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtEmailKeyTyped(evt);
    }
});

```

```

});

lblWeight.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblWeight.setForeground(new java.awt.Color(8, 8, 8));
lblWeight.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblWeight.setText("Weight:");

lblHeight.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblHeight.setForeground(new java.awt.Color(8, 8, 8));
lblHeight.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblHeight.setText("Height");

lblEmail.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblEmail.setForeground(new java.awt.Color(8, 8, 8));
lblEmail.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblEmail.setText("Email:");

jCalendarBirthdate.addChangeListener(new javax.swing.event.ChangeListener() {
    public void stateChanged(javax.swing.event.ChangeEvent evt) {
        jCalendarBirthdateStateChanged(evt);
    }
});

javax.swing.GroupLayout pnlDelete1Layout = new javax.swing.GroupLayout(pnlDelete1);
pnlDelete1.setLayout(pnlDelete1Layout);
pnlDelete1Layout.setHorizontalGroup(
    pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlDelete1Layout.createSequentialGroup()
            .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                .addComponent(kbtnClearFields, javax.swing.GroupLayout.PREFERRED_SIZE,
58, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(lblNotes, javax.swing.GroupLayout.PREFERRED_SIZE, 76,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            )
            .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(txtNotes)
                .addGroup(pnlDelete1Layout.createSequentialGroup()
                    .addComponent(kbtnDeselect, javax.swing.GroupLayout.PREFERRED_SIZE,
58, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(18, 18, 18)
                )
            )
        )
);

```

```

        .addComponent(kbtnAdd, javax.swing.GroupLayout.PREFERRED_SIZE, 61,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(kbtnUpdate, javax.swing.GroupLayout.PREFERRED_SIZE, 69,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 10, Short.MAX_VALUE)))
        .addContainerGap()
        .addGroup(pnlDelete1Layout.createSequentialGroup())
        .addContainerGap()

        .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
G)
        .addGroup(pnlDelete1Layout.createSequentialGroup())

        .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
G)
        .addGroup(pnlDelete1Layout.createSequentialGroup())
        .addComponent(lblFirstName,
javax.swing.GroupLayout.PREFERRED_SIZE, 76,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(txtFirstName))
        .addGroup(pnlDelete1Layout.createSequentialGroup())
        .addComponent(lblMiddleName,
javax.swing.GroupLayout.PREFERRED_SIZE, 76,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(txtMiddleName))
        .addGroup(pnlDelete1Layout.createSequentialGroup())
        .addComponent(lblLastName,
javax.swing.GroupLayout.PREFERRED_SIZE, 76,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(txtLastName))
        .addGroup(pnlDelete1Layout.createSequentialGroup())
        .addComponent(lblPhoneNumber)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(txtPhoneNumber))
        .addGroup(pnlDelete1Layout.createSequentialGroup())

```

```

        .addComponent(lblAddress, javax.swing.GroupLayout.PREFERRED_SIZE,
76, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(txtAddress))
        .addGroup(pnlDelete1Layout.createSequentialGroup())

        .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
G)
            .addComponent(lblHeight,
javax.swing.GroupLayout.PREFERRED_SIZE, 76,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(lblEmail, javax.swing.GroupLayout.PREFERRED_SIZE,
76, javax.swing.GroupLayout.PREFERRED_SIZE))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
G)
            .addComponent(txtEmail)
            .addComponent(txtHeight))))
        .addContainerGap())
        .addGroup(pnlDelete1Layout.createSequentialGroup())

        .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
G)
            .addGroup(pnlDelete1Layout.createSequentialGroup())
            .addComponent(lblWeight, javax.swing.GroupLayout.PREFERRED_SIZE,
76, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(txtWeight))
        .addGroup(pnlDelete1Layout.createSequentialGroup())

        .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
G)
            .addGroup(pnlDelete1Layout.createSequentialGroup())
            .addComponent(lblEID)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(txtEID, javax.swing.GroupLayout.PREFERRED_SIZE,
103, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)

```

```

        .addComponent(cbHired,
javax.swing.GroupLayout.PREFERRED_SIZE, 85,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(pnlDelete1Layout.createSequentialGroup())
        .addComponent(lblBirthdate,
javax.swing.GroupLayout.PREFERRED_SIZE, 76,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(jCalendarBirthdate,
javax.swing.GroupLayout.PREFERRED_SIZE, 144,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(pnlDelete1Layout.createSequentialGroup())
        .addGap(6, 6, 6)
        .addComponent(lblAge,
javax.swing.GroupLayout.PREFERRED_SIZE, 76,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(txtAge,
javax.swing.GroupLayout.PREFERRED_SIZE, 79,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addComponent(lblOptionalFields,
javax.swing.GroupLayout.PREFERRED_SIZE, 103,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(0, 0, Short.MAX_VALUE)))
        .addGap(6, 6, 6))))
);
pnlDelete1Layout.setVerticalGroup(
    pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlDelete1Layout.createSequentialGroup())
            .addContainerGap()

        .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(lblEID)
            .addComponent(txtEID, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(cbHired))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

```



```

        .addComponent(txtFirstName, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(lblFirstName))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblMiddleName)
        .addComponent(txtMiddleName, javax.swing.GroupLayout.PREFERRED_SIZE,
20, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblLastName)
        .addComponent(txtLastName, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

    .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    .addComponent(lblBirthdate)
        .addComponent(jCalendarBirthdate, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblAge)
        .addComponent(txtAge, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)

    .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(txtAddress, javax.swing.GroupLayout.PREFERRED_SIZE, 19,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(lblAddress))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(txtPhoneNumber, javax.swing.GroupLayout.PREFERRED_SIZE,
19, javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addComponent(lblPhoneNumber))
    .addGap(18, 18, 18)
    .addComponent(lblOptionalFields)
    .addGap(8, 8, 8)

    .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(txtEmail, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(lblEmail))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlDelete1Layout.createSequentialGroup()
            .addComponent(txtHeight, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED))

    .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlDelete1Layout.createSequentialGroup()
            .addComponent(txtWeight, javax.swing.GroupLayout.PREFERRED_SIZE,
20, javax.swing.GroupLayout.PREFERRED_SIZE)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(lblNotes)
        .addComponent(txtNotes, javax.swing.GroupLayout.PREFERRED_SIZE,
20, javax.swing.GroupLayout.PREFERRED_SIZE))

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

    .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(kbtnClearFields,
javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(kbtnDeselect,
javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addComponent(kbtnUpdate,
javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(kbtnAdd, javax.swing.GroupLayout.PREFERRED_SIZE,
30, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(36, 36, 36))
        .addGroup(pnlDelete1Layout.createSequentialGroup()
        .addComponent(lblWeight)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))))
        .addGroup(pnlDelete1Layout.createSequentialGroup()
        .addComponent(lblHeight)
        .addGap(0, 0, Short.MAX_VALUE))))
    );

    pnlEmail1.setBackground(new java.awt.Color(238, 238, 238));
    pnlEmail1.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
    pnlEmail1.setForeground(new java.awt.Color(36, 48, 94));

    kbtnSalaryGenerator.setText("Salary Generator");
    kbtnSalaryGenerator.setkEndColor(new java.awt.Color(28, 84, 140));
    kbtnSalaryGenerator.setkHoverEndColor(new java.awt.Color(76, 111, 167));
    kbtnSalaryGenerator.setkHoverForeGround(new java.awt.Color(252, 250, 255));
    kbtnSalaryGenerator.setkHoverStartColor(new java.awt.Color(76, 111, 167));
    kbtnSalaryGenerator.setkSelectedColor(new java.awt.Color(252, 250, 255));
    kbtnSalaryGenerator.setkStartColor(new java.awt.Color(28, 84, 140));
    kbtnSalaryGenerator.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            kbtnSalaryGeneratorActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout pnlEmail1Layout = new javax.swing.GroupLayout(pnlEmail1);
    pnlEmail1.setLayout(pnlEmail1Layout);
    pnlEmail1Layout.setHorizontalGroup(
        pnlEmail1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlEmail1Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(kbtnSalaryGenerator, javax.swing.GroupLayout.PREFERRED_SIZE,
0, Short.MAX_VALUE)
        .addContainerGap())
    );

```

```

        pnlEmail1Layout.setVerticalGroup(
            pnlEmail1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(pnlEmail1Layout.createSequentialGroup()
                    .addContainerGap()
                    .addComponent(kbtnSalaryGenerator, javax.swing.GroupLayout.PREFERRED_SIZE,
36, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addContainerGap(9, Short.MAX_VALUE))
                );

        javax.swing.GroupLayout pnlBackgroundLayout = new
        javax.swing.GroupLayout(pnlBackground);
        pnlBackground.setLayout(pnlBackgroundLayout);
        pnlBackgroundLayout.setHorizontalGroup(

pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(pnlBackgroundLayout.createSequentialGroup()
        .addContainerGap()

.addGroup(pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING, false)
    .addGroup(pnlBackgroundLayout.createSequentialGroup()
        .addComponent(pnlEmail, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(pnlEmail1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addComponent(pnlDelete, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(pnlDelete1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
    .addComponent(pnlTable, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(pnlSearch, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    .addContainerGap())
    );
    pnlBackgroundLayout.setVerticalGroup(

pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(pnlBackgroundLayout.createSequentialGroup()

```

```

        .addContainerGap()

    .addGroup(pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
        .addComponent(pnlEmail, javax.swing.GroupLayout.Alignment.TRAILING,
            javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
            Short.MAX_VALUE)
        .addComponent(pnlEmail1, javax.swing.GroupLayout.Alignment.TRAILING,
            javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
            Short.MAX_VALUE)
        .addComponent(pnlSearch, javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
            javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

    .addGroup(pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
            pnlBackgroundLayout.createSequentialGroup()
                .addComponent(pnlDelete1, javax.swing.GroupLayout.DEFAULT_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(pnlDelete, javax.swing.GroupLayout.PREFERRED_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addComponent(pnlTable, javax.swing.GroupLayout.PREFERRED_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap())
    );

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(pnlBackground, javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(pnlBackground, javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );

    pack();
    setLocationRelativeTo(null);

```

```
}// </editor-fold>
```

```
private void kbtnSortActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        String search = txtSearch.getText().trim();

        int column = cmbColumn.getSelectedIndex();

        //Set of column names to know which one to search in the database
        String[] columnnames = {"eid", "hired", "first_name", "middle_name", "last_name",
            "birthdate", "age", "address", "phonenum", "email", "height", "weight", "notes"};

        String strcolumn = columnnames[column];

        //switch case to determine the order in which to sort the table
        int sort = cmbOrder.getSelectedIndex();
        String strsort = "";
        switch (sort){
            case 0:
                strsort = "ASC";
                break;
            case 1:
                strsort = "DESC";
        }

        if (column==0 && sort==0){
            //The table is naturally sorted by eid, so a normal select query will do.
            //EmployeeRecords has its own implementation of refreshable
            refreshable("SELECT * FROM info");
        }
        else{
            //Send the select statement to refreshable for processing
            refreshable("select * from info order by "+strcolumn+" "+strsort);
        }

    }
    catch (Exception e){

    }
}

private void cmbColumnActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}
```

```

private void tblEmployeesMouseClicked(java.awt.event.MouseEvent evt) {
    try{
        //Prerequisite for the update function
        DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
        int selectedrow=tblEmployees.getSelectedRow();
        if (tblEmployees.getRowSorter()!=null) {
            //Gets the correct row for a filtered jtable (RowFilter)
            selectedrow = tblEmployees.getRowSorter().convertRowIndexToModel(selectedrow);
        }
        String eid = "";
        boolean hired = false;
        String firstname = "";
        String middlename = "";
        String lastname = "";

        String strd = "";
        Date d = new Date();

        String age = "";
        String address = "";
        String phonenumber = "";

        String email = "";
        String height = "";
        String weight = "";
        String notes = "";

        eid = model.getValueAt(selectedrow,0).toString();
        hired = Boolean.parseBoolean(model.getValueAt(selectedrow,1).toString());

        firstname = model.getValueAt(selectedrow,2).toString();
        middlename = model.getValueAt(selectedrow,3).toString();
        lastname = model.getValueAt(selectedrow,4).toString();
        strd = model.getValueAt(selectedrow,5).toString();
        System.out.println("Strd: "+strd);

        age =model.getValueAt(selectedrow,6).toString();
        address=model.getValueAt(selectedrow,7).toString();
        phonenumber= model.getValueAt(selectedrow,8).toString();

        System.out.println("Reaches Here 1");
    }
}

```

```

Object objemail = model.getValueAt(selectedrow,9);
System.out.println("Reaches Here 2");
Object objheight = model.getValueAt(selectedrow,10);
System.out.println("Reaches Here 3");
Object objweight = model.getValueAt(selectedrow,11);
Object objnotes = model.getValueAt(selectedrow,12);

if (objemail == null){
    email = "";
}
else{
    email = objemail.toString();
}
if (objheight == null){
    height = "";
}
else{
    height = objheight.toString();
}

if (objweight == null){
    weight = "";
}
else{
    weight = objweight.toString();
}
if (objnotes == null){
    notes = "";
}
else{
    notes = objnotes.toString();
}

this.selectedroweid = eid;
txtEID.setText(eid);
cbHired.setSelected(hired);
txtFirstName.setText(firstname);
txtMiddleName.setText(middlename);
txtLastName.setText(lastname);

SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd");
d = formatter.parse(strd);

```



```

        jCalendarBirthdate.setDate(d);
        txtAge.setText(age);
        txtAddress.setText(address);
        txtPhoneNumber.setText(phonenummer);
        txtEmail.setText(email);
        txtHeight.setText(height);
        txtWeight.setText(weight);
        txtNotes.setText(notes);
    }
    catch (Exception e){
        e.printStackTrace();
    }
}

private void cmbInPropertyChange(java.beans.PropertyChangeEvent evt) {

}

private void cmbInActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void cmbInItemStateChanged(java.awt.event.ItemEvent evt) {
    try{
        search();
    }
    catch (Exception e){

    }
}

private void txtSearchKeyTyped(java.awt.event.KeyEvent evt) {

}

private void txtSearchKeyReleased(java.awt.event.KeyEvent evt) {
    try{
        search();
    }
    catch (Exception e){

    }
}

```

```

private void txtSearchActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void kbtnResetActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        //Resets the search and sort functions; restores table to original
        refreshtable("SELECT * FROM info");
        cmbColumn.setSelectedIndex(0);
        cmbOrder.setSelectedIndex(0);

        txtSearch.setText("");
        cmbIn.setSelectedIndex(0);

        search();
    }
    catch (Exception e){

    }
}

private void kbtnSalaryGeneratorActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        new SalaryGenerator(this.uid,this.email).setVisible(true);
        this.setVisible(false);
    }
    catch (Exception e){

    }
}

private void kbtnDeleteAllActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        int confirm = JOptionPane.showConfirmDialog(null,"Do you want to delete ALL
records?");
        if (confirm==0){
            DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
            int rowcount = model.getRowCount(); //number of rows in model

            if (rowcount==0){
                JOptionPane.showMessageDialog(null, "The table is empty.");
            }
        }
    }
}

```

```

else{
    int eids[] = new int[rowcount];
    for (int i = 0; i<rowcount; i++){
        eids[i] = Integer.parseInt(model.getValueAt(i,0).toString());

    }
    String databaseURL = "jdbc:derby:SALARY_GENERATOR";
    Connection con = DriverManager.getConnection(databaseURL);

    String sql;
    for (int i = 0; i<eids.length; i++){
        //Deletion based on eid
        Statement st = con.createStatement();
        sql = "DELETE FROM info WHERE eid=" + eids[i];
        st.execute(sql);
    }

    maxfinder();

    for (int i = 0; i<eids.length; i++){
        //Deletion based on eid
        Statement st = con.createStatement();
        sql = "DELETE FROM salary_allocations WHERE eid=" + eids[i];
        st.execute(sql);
    }
    con.close();

    model.setRowCount(0);
}
}
}
catch(Exception e){

}
}

private void kbtnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
        int rowcount = model.getRowCount();

        //Allows it to work with the filtered jtable (RowFilter)

```

```

int rows[] = tblEmployees.getSelectedRows();
for (int i = 0; i<rows.length; i++){
    rows[i] = tblEmployees.getRowSorter().convertRowIndexToModel(rows[i]);
}

if ("-1".equals(this.selectedroweid)){
    JOptionPane.showMessageDialog(null, "A row has not been selected.");
}
else{
    int confirm = JOptionPane.showConfirmDialog(null,"Are you sure?");

    if (confirm==0){
        if (rows.length>0){//Delete multiple or single
            int eids[] = new int[rows.length];
            for (int i = 0; i<rows.length; i++){
                eids[i] = Integer.parseInt(model.getValueAt(rows[i],0).toString());
            }

            //Establish connection to the database
            String databaseURL = "jdbc:derby:SALARY_GENERATOR";
            Connection con = DriverManager.getConnection(databaseURL);

            //iterate through the eids generated and delete them one by one
            String sql;
            for (int i = 0; i<eids.length; i++){
                Statement st = con.createStatement();

                sql = "DELETE FROM info WHERE eid=" + eids[i];
                st.execute(sql);
            }

            //Iterate through the employee's salaries in salary_allocations and delete them
too
            for (int i = 0; i<eids.length; i++){
                Statement st = con.createStatement();

                sql = "DELETE FROM salary_allocations WHERE eid=" + eids[i];
                st.execute(sql);
            }
            con.close();

            //        txtFullName.setText("");
            //        txtEmail.setText("");
            //        txtPhoneNumber.setText("");

```

```

        this.selectedroweid = "-1";
        refreshtable("select * from info");

        maxfinder();
    }
}

}
catch (Exception e){

}

}

private void jCalendarBirthdateStateChanged(javax.swing.event.ChangeEvent evt) {
    try{
        Date startDate = new Date(); //the current date
        Date endDate = jCalendarBirthdate.getDate(); //The date inputted by the user.

        // Calculate the difference in milliseconds
        long differenceInMillis = startDate.getTime() - endDate.getTime();

        // Calculate the number of years (approximate)
        long years = differenceInMillis / (1000L * 60 * 60 * 24 * 365);

        if (years<16){
            JOptionPane.showMessageDialog(null, "Age must be at least 16.");
            txtAge.setText("");
        }
        else{
            txtAge.setText(Long.toString(years));
        }
    }
    catch (Exception e){

    }

}

private void txtEmailKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

```

```

private void txtEmailKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtEmailActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtPhoneNumberKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtPhoneNumberKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtPhoneNumberActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtAddressKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtAddressKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtAddressActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtAgeKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtAgeKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtAgeActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

```

```

private void txtLastNameKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtLastNameKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtLastNameActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtMiddleNameKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtMiddleNameKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtMiddleNameActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void kbtnAddActionPerformed(java.awt.event.ActionEvent evt) {
    try{

        boolean hired = false;
        String firstname = "";
        String middlename = "";
        String lastname = "";
        Date d = new Date();
        String age = "";
        String address = "";
        String phonenumber = "";
        String email = "";
        String height = "";
        String weight = "";
        String notes = "";

        //get values from the GUI buttons
        hired = cbHired.isSelected();
        firstname = txtFirstName.getText();
        middlename = txtMiddleName.getText();

```

```

lastname = txtLastName.getText();
d = jCalendarBirthdate.getDate();
age = txtAge.getText();
address = txtAddress.getText();
phonenumber = txtPhoneNumber.getText();
email = txtEmail.getText();
height = txtHeight.getText();
weight = txtWeight.getText();
notes = txtNotes.getText();

if (email.equals("")){
    email = "NULL";
}
if (height.equals("")){
    height = "NULL";
}
if (weight.equals("")){
    weight = "NULL";
}
if (notes.equals("")){
    notes = "NULL";
}

System.out.println("Email: " + email);

//Error trapping
if ("".equals(firstname)){
    JOptionPane.showMessageDialog(null, "First Name field is empty.");
}
else if ("".equals(middlename)){
    JOptionPane.showMessageDialog(null, "Middle Name field is empty.");
}
else if ("".equals(lastname)){
    JOptionPane.showMessageDialog(null, "Last Name field is empty.");
}
else if ("".equals(address)){
    JOptionPane.showMessageDialog(null, "Address field is empty.");
}
else if ("".equals(phonenumber)){
    JOptionPane.showMessageDialog(null, "Phone Number field is empty.");
}
else if ("".equals(age)){
    JOptionPane.showMessageDialog(null, "Please enter a valid birthdate.");
}

```



```

        else if (firstname.length()>50){
            JOptionPane.showMessageDialog(null, "First Name is too long. Maximum is 50
characters.");
        }
        else if (middlename.length()>50){
            JOptionPane.showMessageDialog(null, "Middle Name is too long. Maximum is 50
characters.");
        }
        else if (lastname.length()>50){
            JOptionPane.showMessageDialog(null, "Last Name is too long. Maximum is 50
characters.");
        }
        else if (address.length()>100){
            JOptionPane.showMessageDialog(null, "Address is too long. Maximum is 100
characters.");
        }
        else if (phonenummer.length()>50){
            JOptionPane.showMessageDialog(null, "Phone Number is too long. Maximum is 50
characters.");
        }
        else if (phonenummer.length()>50){
            JOptionPane.showMessageDialog(null, "Email is too long. Maximum is 50
characters.");
        }
        else if (height.length()>10){
            JOptionPane.showMessageDialog(null, "Height is too long. Maximum is 10
characters.");
        }
        else if (weight.length()>10){
            JOptionPane.showMessageDialog(null, "Weight is too long. Maximum is 10
characters.");
        }
        else if (notes.length()>200){
            JOptionPane.showMessageDialog(null, "Notes is too long. Maximum is 200
characters.");
        }
        else{
            maxfinder();
            this.max++;
            String strmax = Integer.toString(this.max);

            System.out.println("strmax: " + strmax);

            //Connect to the database

```

```

String databaseURL = "jdbc:derby:SALARY_GENERATOR";
Connection con = DriverManager.getConnection(databaseURL);
Statement st = con.createStatement();
Statement st2 = con.createStatement();

Date addeddate = new java.sql.Date(d.getTime());

//Insert SQL statement to insert a new row into the info system table
String sql = "INSERT INTO info (eid, userid, hired, first_name, middle_name,
last_name, age, "
    + "birthdate, address, phonenumber, email, height, weight, notes) "
    + "VALUES (" + strmax + ", NULL, " + hired + ", " + firstname + ", " + middlename +
    + " " + lastname + ", "
    + age + ", " + new java.sql.Timestamp(d.getTime()) + ", " + address + ", " +
    + phonenumber + ", " + email
    + ", " + height + ", " + weight + ", " + notes + ")";

maxfinder2();
this.salarymax++;

maxfinder3();
this.historymax++;
//Must insert a salary record into the salary allocations table as the two are linked.
//pay_period determined by the max from salary and loan history
String sql2 = "INSERT INTO salary_allocations (sid, eid, pay_period,
monthly_wage_rate, sss_ee, sss_er, ph_ee_p, "
    + "ph_er_p, hdmf_ee_p, hdmf_er_p, loan, loan_add, loan_sub, deduction,
bonus) "
    + "VALUES (" + this.salarymax + ", " + strmax + ", " + this.historymax + ", " + 0 + ", " +
    + 0 + ", " + 0 + ", " + 0 + ", " + 0 + ", " + 0 //insert default values
    + ", " + 0 + ", " + 0 + ", " + 0 + ", " + 0 + ", " + 0 + ")";
st.execute(sql);
st2.execute(sql2);

con.close();

refreshtable("select * from info");

txtEID.setText(Integer.toString(max));
txtFirstName.setText("");
txtMiddleName.setText("");
txtLastName.setText("");
jCalendarBirthdate.setDate(new Date(70,1,1));

```

```

        txtAge.setText("");
        txtAddress.setText("");
        txtPhoneNumber.setText("");
        txtEmail.setText("");
        txtHeight.setText("");
        txtWeight.setText("");
        txtNotes.setText("");

        JOptionPane.showMessageDialog(null, "Added.");
    }
}
catch (Exception e){
    e.printStackTrace();
}
}

private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        //Initialize variables
        String eid = "";
        boolean hired = false;
        String firstname = "";
        String middlename = "";
        String lastname = "";
        Date d = new Date();
        String age = "";
        String address = "";
        String phonenumber = "";
        String email = "";
        String height = "";
        String weight = "";
        String notes = "";

        //Get values from textboxes and user-inputted data
        eid = txtEID.getText();
        hired = cbHired.isSelected();
        firstname = txtFirstName.getText();
        middlename = txtMiddleName.getText();
        lastname = txtLastName.getText();
        d = jCalendarBirthdate.getDate();
        age = txtAge.getText();
        address = txtAddress.getText();
        phonenumber = txtPhoneNumber.getText();
        email = txtEmail.getText();
    }
}

```

```

height = txtHeight.getText();
weight = txtWeight.getText();
notes = txtNotes.getText();

if (email.equals("")){
    email = "NULL";
}
if (height.equals("")){
    height = "NULL";
}
if (weight.equals("")){
    weight = "NULL";
}
if (notes.equals("")){
    notes = "NULL";
}

System.out.println("Email: " + email);

//Error trapping
if ("".equals(firstname)){
    JOptionPane.showMessageDialog(null, "First Name field is empty.");
}
else if ("".equals(middlename)){
    JOptionPane.showMessageDialog(null, "Middle Name field is empty.");
}
else if ("".equals(lastname)){
    JOptionPane.showMessageDialog(null, "Last Name field is empty.");
}
else if ("".equals(address)){
    JOptionPane.showMessageDialog(null, "Address field is empty.");
}
else if ("".equals(phonenummer)){
    JOptionPane.showMessageDialog(null, "Phone Number field is empty.");
}
else if (firstname.length()>50){
    JOptionPane.showMessageDialog(null, "First Name is too long. Maximum is 50
characters.");
}
else if (middlename.length()>50){
    JOptionPane.showMessageDialog(null, "Middle Name is too long. Maximum is 50
characters.");
}
else if (lastname.length()>50){

```

```

        JOptionPane.showMessageDialog(null, "Last Name is too long. Maximum is 50
characters.");
    }
    else if (address.length()>100){
        JOptionPane.showMessageDialog(null, "Address is too long. Maximum is 100
characters.");
    }
    else if (phonenummer.length()>50){
        JOptionPane.showMessageDialog(null, "Phone Number is too long. Maximum is 50
characters.");
    }
    else if (phonenummer.length()>50){
        JOptionPane.showMessageDialog(null, "Email is too long. Maximum is 50
characters.");
    }
    else if (height.length()>10){
        JOptionPane.showMessageDialog(null, "Height is too long. Maximum is 10
characters.");
    }
    else if (weight.length()>10){
        JOptionPane.showMessageDialog(null, "Weight is too long. Maximum is 10
characters.");
    }
    else if (notes.length()>200){
        JOptionPane.showMessageDialog(null, "Notes is too long. Maximum is 200
characters.");
    }
    else{

        int max = this.max;
        max++;
        this.max = max;
        String strmax = Integer.toString(max);

        //Establish a connection with the database
        String databaseURL = "jdbc:derby:SALARY_GENERATOR";
        Connection con = DriverManager.getConnection(databaseURL);
        Statement st = con.createStatement();

        Date addeddate = new java.sql.Date(d.getTime());

        //Update to the table using SQL statements, inserting error-trapped user-inputted data
        String sql = "UPDATE info SET eid="+eid+", userid=NULL, hired="+hired+",
first_name="+firstname+", middle_name="+middlename+", "

```

```

        + "last_name='"+lastname+"",age = "+age+", birthdate='"+ new
java.sql.Timestamp(d.getTime())+"", address='"+ address + "', "
        + "phonenumber='"+phonenumber+"", email = '"+email+"", height = "+height+", weight
= "+weight+", notes='"+notes+" "
        + "WHERE eid="+eid;
        //Execute the query
        st.execute(sql);
        //Close the connection
        con.close();

        //No row currently selected.
        selectedroweid = "-1";

        JOptionPane.showMessageDialog(null, "Updated.");
        tblEmployees.clearSelection();

        refreshtable("select * from info");
        maxfinder();

        txtEID.setText(Integer.toString(max));
        txtFirstName.setText("");
        txtMiddleName.setText("");
        txtLastName.setText("");
        jCalendarBirthdate.setDate(new Date(70,1,1));
        txtAge.setText("");
        txtAddress.setText("");
        txtPhoneNumber.setText("");
        txtEmail.setText("");
        txtHeight.setText("");
        txtWeight.setText("");
        txtNotes.setText("");
    }
}
catch (Exception e){
    e.printStackTrace();
}
}

private void kbtnDeselectActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        //Deselects all rows
        // DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
        // tblEmployees.removeRowSelectionInterval(0, model.getRowCount()-1);
        tblEmployees.clearSelection();
    }
}

```

```

        //resets selectedroweid
        this.selectedroweid = "-1";
    }
    catch (Exception e){

    }
}

private void kbtnClearFieldsActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        txtEID.setText("");
        txtFirstName.setText("");
        txtMiddleName.setText("");
        txtLastName.setText("");
        Date d = new Date(70,1,1);
        jCalendarBirthdate.setDate(d);
        txtAge.setText("");
        txtAddress.setText("");
        txtPhoneNumber.setText("");
        txtEmail.setText("");
        txtHeight.setText("");
        txtWeight.setText("");
        txtNotes.setText("");
    }
    catch(Exception e){

    }
}

private void txtNotesKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtNotesKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtNotesActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtWeightKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

```

```

private void txtWeightKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtWeightActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtHeightKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtHeightKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtHeightActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtFirstNameKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtFirstNameKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtFirstNameActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtEIDKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtEIDKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtEIDActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

```



```

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Register().setVisible(true);
        }
    });
}

```

```

// Variables declaration - do not modify
private javax.swing.JCheckBox cbHired;
private javax.swing.JComboBox<String> cmbColumn;
private javax.swing.JComboBox<String> cmbIn;
private javax.swing.JComboBox<String> cmbOrder;
private de.wannawork.jcalendar.JCalendarComboBox jCalendarBirthdate;
private javax.swing.JScrollPane jScrollPane1;
private com.k33ptoo.components.KButton kbtnAdd;
private com.k33ptoo.components.KButton kbtnClearFields;
private com.k33ptoo.components.KButton kbtnDelete;
private com.k33ptoo.components.KButton kbtnDeleteAll;
private com.k33ptoo.components.KButton kbtnDeselect;
private com.k33ptoo.components.KButton kbtnReset;
private com.k33ptoo.components.KButton kbtnSalaryGenerator;
private com.k33ptoo.components.KButton kbtnSort;
private com.k33ptoo.components.KButton kbtnUpdate;
private javax.swing.JLabel lblAddress;
private javax.swing.JLabel lblAge;
private javax.swing.JLabel lblBirthdate;
private javax.swing.JLabel lblEID;
private javax.swing.JLabel lblEditPersonalRecords;
private javax.swing.JLabel lblEmail;
private javax.swing.JLabel lblFirstName;
private javax.swing.JLabel lblHeight;
private javax.swing.JLabel lblLastName;
private javax.swing.JLabel lblMiddleName;
private javax.swing.JLabel lblNotes;
private javax.swing.JLabel lblOptionalFields;
private javax.swing.JLabel lblPhoneNumber;
private javax.swing.JLabel lblSearch;
private javax.swing.JLabel lblWeight;
private com.k33ptoo.components.KGradientPanel pnlBackground;
private javax.swing.JPanel pnlDelete;
private javax.swing.JPanel pnlDelete1;
private javax.swing.JPanel pnlEmail;
private javax.swing.JPanel pnlEmail1;
private javax.swing.JPanel pnlSearch;
private javax.swing.JPanel pnlTable;
private javax.swing.JTable tblEmployees;
private javax.swing.JTextField txtAddress;
private javax.swing.JTextField txtAge;
private javax.swing.JTextField txtEID;
private javax.swing.JTextField txtEmail;

```

```
private javax.swing.JTextField txtFirstName;  
private javax.swing.JTextField txtHeight;  
private javax.swing.JLabel txtIn;  
private javax.swing.JTextField txtLastName;  
private javax.swing.JTextField txtMiddleName;  
private javax.swing.JTextField txtNotes;  
private javax.swing.JTextField txtPhoneNumber;  
private javax.swing.JTextField txtSearch;  
private javax.swing.JLabel txtSort;  
private javax.swing.JTextField txtWeight;  
// End of variables declaration  
}
```

Salary History Page:

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
 license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */

/**
 *
 * @author User
 */
import java.awt.Color;
import java.sql.Connection;
import java.sql.DatabaseMetaData;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.MessageFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.print.attribute.HashPrintRequestAttributeSet;
import javax.print.attribute.PrintRequestAttributeSet;
import javax.print.attribute.standard.OrientationRequested;
import javax.swing.JOptionPane;
import javax.swing.JTable;
import javax.swing.RowFilter;
import javax.swing.SwingUtilities;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableModel;
import javax.swing.table.TableRowSorter;

import javax.swing.text.*;

import org.apache.pdfbox.pdmodel.PDDocument;
import org.apache.pdfbox.pdmodel.PDPage;
import org.apache.pdfbox.pdmodel.PDPageContentStream;
import org.apache.pdfbox.pdmodel.font.PDType1Font;
import static org.apache.pdfbox.pdmodel.font.Standard14Fonts.FontName.HELVETICA;
//import org.apache.pdfbox.pdmodel.PDField;
//import org.apache.pdfbox.text.PDFTextStripper;

public class SalaryHistory extends javax.swing.JFrame {
```

```

/**
 * Creates new form Register
 */

//Global Variables
int max = 0;
int payperiod = -1;
String email = "", uid = "";
String selectedroweid = "-1";
//Setting up the regexfilter
TableRowSorter sorter;

private void calcFinalWage(){
    try{
        double wage = Double.parseDouble(txtPayPeriod.getText());
        double sssee = Double.parseDouble(txtSSSEE.getText());
        double phee = Double.parseDouble(txtPHEE.getText());
        double hdmfee = Double.parseDouble(txtHDMFEE.getText());
        double deduction = Double.parseDouble(txtDeduction.getText());
        double bonus = Double.parseDouble(txtBonus.getText());

        double n = wage - sssee - (phee*wage/100) - (hdmfee*wage/100) - deduction + bonus;
        System.out.println("n: "+n);
        String strn = Double.toString(n);
//        String strn = Double.toString((double)Math.round(n*100)/100);
        System.out.println("strn: "+strn);
        txtFinalWage.setText(strn);

    }
    catch(Exception e){
        e.printStackTrace();
    }
}

private void payperiodfinder(){ //figure out the highest pay period
    try{
        String databaseURL = "jdbc:derby:SALARY_GENERATOR";
        Connection con = DriverManager.getConnection(databaseURL);

        String sql;
        Statement st = con.createStatement();

```

```

        sql = "SELECT * FROM salary_history";
        ResultSet rs = st.executeQuery(sql);

        //db stands for database

        int max = 0;
        while (rs.next()==true){
            String dbeid = rs.getString(3);
            int eid = Integer.parseInt(dbeid);
            if (eid>max){
                max = eid;
            }
        }
        this.payperiod = max;
        con.close();
    }
    catch (Exception e){
        e.printStackTrace();
    }
}

private void initComponents2(){
    //Overwrite Java Driver Code

    //Display the email used to login, truncated if too long
    if (email.length()<18){
        lblEmail.setText(email);
    }
    else{
        lblEmail.setText(email.substring(0,15)+"...");
    }

    //Initializing the table with specific settings
    //For the regexfilter later on
    TableRowSorter sorter = new
    TableRowSorter<>((DefaultTableModel)tblEmployees.getModel());
    this.sorter = sorter;
    tblEmployees.setRowSorter(sorter);
    //Disables table sorting
    sorter.setSortable(0, false);
    sorter.setSortable(1, false);
    sorter.setSortable(2, false);
    sorter.setSortable(3, false);

```

```

//Disables rearranging table columns
tblEmployees.getTableHeader().setReorderingAllowed(false);

//Textboxes set to a clear background
/*txtFullName.setBackground(new Color(0, 0, 0, 0));
txtEmail.setBackground(new Color(0, 0, 0, 0));
txtPhoneNumber.setBackground(new Color(0, 0, 0, 0));*/

//Numeric text fields
((AbstractDocument) txtEID.getDocument()).setDocumentFilter(new
IntegerDocumentFilter());
((AbstractDocument) txtPayPeriod.getDocument()).setDocumentFilter(new
IntegerDocumentFilter());

((AbstractDocument) txtWage.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
((AbstractDocument) txtSSSEE.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
((AbstractDocument) txtSSSER.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
((AbstractDocument) txtPHEE.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
((AbstractDocument) txtPHER.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
((AbstractDocument) txtHDMFEE.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
((AbstractDocument) txtHDMFER.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
((AbstractDocument) txtBonus.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
((AbstractDocument) txtDeduction.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
((AbstractDocument) txtFinalWage.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
((AbstractDocument) txtFinalER.getDocument()).setDocumentFilter(new
NumericDocumentFilter());

String maxeid = Integer.toString(this.max);
txtEID.setText(maxeid);

//    cmbSSSBBracket.addItem();

}

```

```

public static boolean doesTableExists(String tableName, Connection conn)
    throws SQLException
{
    DatabaseMetaData meta = conn.getMetaData();
    ResultSet result = meta.getTables(null, null, tableName.toUpperCase(), null);

    return result.next();
}

private void maxfinder(){ //figure out the highest eid
    try{
        String databaseURL = "jdbc:derby:SALARY_GENERATOR";
        Connection con = DriverManager.getConnection(databaseURL);

        String sql;
        Statement st = con.createStatement();

        sql = "SELECT * FROM salary_history";
        ResultSet rs = st.executeQuery(sql);

        //db stands for database

        int max = 0;
        while (rs.next()==true){
            String dbeid = rs.getString(1);
            int eid = Integer.parseInt(dbeid);
            if (eid>max){
                max = eid;
            }
        }
        this.max = max;
        con.close();
    }
    catch (Exception e){

    }
}

private void search(){
    try{
        int in = cmbIn.getSelectedIndex();

        String search = txtSearch.getText().trim();
    }
}

```



```

//Search using RowFilter and regex (regular expression) to find substrings
RowFilter<DefaultTableModel, Object> rf = null;
try {
    if (in == 0){
        rf = RowFilter.regexFilter("(?i)" + search); //("(?i)" for case insensitive filter
    }
    else{
        rf = RowFilter.regexFilter("(?i)" + search, in - 1);
    }
}
catch (java.util.regex.PatternSyntaxException e) {
//    System.out.println("Error Found");
    return;
}
sorter.setRowFilter(rf);
}
catch (Exception e){
    e.printStackTrace();
}
}

private void refreshtable(String sql){//reset table contents, works for SELECT queries only
try{
    DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
    String databaseURL = "jdbc:derby:SALARY_GENERATOR";
    Connection con = DriverManager.getConnection(databaseURL);

    Statement st = con.createStatement();
    ResultSet rs = st.executeQuery(sql);

    model.setRowCount(0); //reset the table
    while (rs.next()==true){

        model.addRow(new
Object[]{rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4),
        rs.getString(5),rs.getString(6),rs.getString(7),rs.getString(8),
        rs.getString(9),rs.getString(10),rs.getString(11),rs.getString(12),
        rs.getString(13),rs.getString(14),rs.getString(15)});
    }
    con.close();

}
}

```

```

        catch (Exception e){

        }

    }

    public SalaryHistory() {
        initComponents();
        try{

            maxfinder();

            refreshtable("select * from salary_history");

            initComponents2();

        }
        catch (Exception e){
            e.printStackTrace();
        }
    }

    public SalaryHistory(String uid, String email){ //CATCH ALL INCOMING DATA
        initComponents();
        try{
            this.uid = uid;
            this.email = email;

            maxfinder();

            refreshtable("select * from salary_history");

            initComponents2();

        }
        catch (Exception e){

        }
    }
}

```

```

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    pnlBackground = new com.k33ptoo.components.KGradientPanel();
    pnlSearch = new javax.swing.JPanel();
    txtSearch = new javax.swing.JTextField();
    lblSearch = new javax.swing.JLabel();
    cmbIn = new javax.swing.JComboBox<>();
    txtIn = new javax.swing.JLabel();
    txtSort = new javax.swing.JLabel();
    cmbColumn = new javax.swing.JComboBox<>();
    cmbOrder = new javax.swing.JComboBox<>();
    kbtnSort = new com.k33ptoo.components.KButton();
    kbtnReset = new com.k33ptoo.components.KButton();
    pnlTable = new javax.swing.JPanel();
    jScrollPane1 = new javax.swing.JScrollPane();
    tblEmployees = new javax.swing.JTable();
    pnlEmail = new javax.swing.JPanel();
    lblEmail = new javax.swing.JLabel();
    pnlDelete = new javax.swing.JPanel();
    kbtnDelete = new com.k33ptoo.components.KButton();
    pnlDelete1 = new javax.swing.JPanel();
    txtEID = new javax.swing.JTextField();
    lblEID = new javax.swing.JLabel();
    txtPayPeriod = new javax.swing.JTextField();
    lblPayPeriod = new javax.swing.JLabel();
    lblSSSEE = new javax.swing.JLabel();
    txtSSSEE = new javax.swing.JTextField();
    lblSSSER = new javax.swing.JLabel();
    txtSSSER = new javax.swing.JTextField();
    lblPHEE = new javax.swing.JLabel();
    txtPHEE = new javax.swing.JTextField();
    lblPHER = new javax.swing.JLabel();
    txtPHER = new javax.swing.JTextField();
    lblHDMFEE = new javax.swing.JLabel();
    txtHDMFEE = new javax.swing.JTextField();
    lblHDMFER = new javax.swing.JLabel();

```

```

txtHDMFER = new javax.swing.JTextField();
lblBonus = new javax.swing.JLabel();
txtBonus = new javax.swing.JTextField();
lblDeduction = new javax.swing.JLabel();
txtDeduction = new javax.swing.JTextField();
lblFinalWage = new javax.swing.JLabel();
txtFinalWage = new javax.swing.JTextField();
kbtnClearFields = new com.k33ptoo.components.KButton();
kbtnDeselect = new com.k33ptoo.components.KButton();
kbtnAdd = new com.k33ptoo.components.KButton();
lblWage = new javax.swing.JLabel();
txtWage = new javax.swing.JTextField();
lblBirthdate = new javax.swing.JLabel();
jCalendarDateSaved = new de.wannawork.jcalendar.JCalendarComboBox();
lblFinalER = new javax.swing.JLabel();
txtFinalER = new javax.swing.JTextField();
pnlEmail1 = new javax.swing.JPanel();
kbtnSalaryGenerator = new com.k33ptoo.components.KButton();
pnlSearch2 = new javax.swing.JPanel();
txtSort2 = new javax.swing.JLabel();
kbtnPrint = new com.k33ptoo.components.KButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setBackground(new java.awt.Color(255, 255, 255));
setResizable(false);

pnlBackground.setkEndColor(new java.awt.Color(66, 133, 244));
pnlBackground.setkGradientFocus(900);
pnlBackground.setkStartColor(new java.awt.Color(66, 133, 244));

pnlSearch.setBackground(new java.awt.Color(238, 238, 238));
pnlSearch.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
pnlSearch.setForeground(new java.awt.Color(36, 48, 94));

txtSearch.setBackground(new java.awt.Color(252, 250, 255));
txtSearch.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtSearch.setForeground(new java.awt.Color(8, 8, 8));
txtSearch.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(38, 117, 247)));
txtSearch.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtSearch.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtSearchActionPerformed(evt);
    }
});

```

```

    }
});
txtSearch.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtSearchKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtSearchKeyTyped(evt);
    }
});

lblSearch.setIcon(new javax.swing.ImageIcon(getClass().getResource("/29x29
Search.png"))); // NOI18N
lblSearch.setText("jLabel2");

cmbIn.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "All", "SHID",
"Pay Period", "EID", "Wage", "Date", "SSS EE", "SSS ER", "PH EE", "PH ER", "HDMF EE",
"HDMF ER", "Bonus", "Deduction", "Final Wage", "Final ER" }));
cmbIn.addItemListener(new java.awt.event.ItemListener() {
    public void itemStateChanged(java.awt.event.ItemEvent evt) {
        cmbInItemStateChanged(evt);
    }
});
cmbIn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cmbInActionPerformed(evt);
    }
});
cmbIn.addPropertyChangeListener(new java.beans.PropertyChangeListener() {
    public void propertyChange(java.beans.PropertyChangeEvent evt) {
        cmbInPropertyChange(evt);
    }
});

txtIn.setFont(new java.awt.Font("Segoe UI Light", 0, 16)); // NOI18N
txtIn.setForeground(new java.awt.Color(8, 8, 8));
txtIn.setText("In:");

txtSort.setFont(new java.awt.Font("Segoe UI Light", 0, 16)); // NOI18N
txtSort.setForeground(new java.awt.Color(8, 8, 8));
txtSort.setText("Sort By:");

```

```

        cmbColumn.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "SHID",
"Pay Period", "EID", "Wage", "Date", "SSS EE", "SSS ER", "PH EE", "PH ER", "HDMF EE",
"HDMF ER", "Bonus", "Deduction", "Final Wage", "Final ER" }));
        cmbColumn.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                cmbColumnActionPerformed(evt);
            }
        });

```

```

        cmbOrder.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {
"Ascending", "Descending" }));

```

```

        kbtnSort.setText("Sort");
        kbtnSort.setkEndColor(new java.awt.Color(28, 84, 140));
        kbtnSort.setkHoverEndColor(new java.awt.Color(76, 111, 167));
        kbtnSort.setkHoverForeGround(new java.awt.Color(252, 250, 255));
        kbtnSort.setkHoverStartColor(new java.awt.Color(76, 111, 167));
        kbtnSort.setkSelectedColor(new java.awt.Color(252, 250, 255));
        kbtnSort.setkStartColor(new java.awt.Color(28, 84, 140));
        kbtnSort.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                kbtnSortActionPerformed(evt);
            }
        });

```

```

        kbtnReset.setText("Reset");
        kbtnReset.setkEndColor(new java.awt.Color(28, 84, 140));
        kbtnReset.setkHoverEndColor(new java.awt.Color(76, 111, 167));
        kbtnReset.setkHoverForeGround(new java.awt.Color(252, 250, 255));
        kbtnReset.setkHoverStartColor(new java.awt.Color(76, 111, 167));
        kbtnReset.setkStartColor(new java.awt.Color(28, 84, 140));
        kbtnReset.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                kbtnResetActionPerformed(evt);
            }
        });

```

```

        javax.swing.GroupLayout pnlSearchLayout = new javax.swing.GroupLayout(pnlSearch);
        pnlSearch.setLayout(pnlSearchLayout);
        pnlSearchLayout.setHorizontalGroup(
            pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(pnlSearchLayout.createSequentialGroup()
                    .addContainerGap()

```

```

        .addComponent(lblSearch, javax.swing.GroupLayout.PREFERRED_SIZE, 29,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(txtSearch, javax.swing.GroupLayout.PREFERRED_SIZE, 156,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(txtIn)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(cmbIn, javax.swing.GroupLayout.PREFERRED_SIZE, 74,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(txtSort)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(cmbColumn, javax.swing.GroupLayout.PREFERRED_SIZE, 133,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(cmbOrder, javax.swing.GroupLayout.PREFERRED_SIZE, 84,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 18,
Short.MAX_VALUE)
        .addComponent(kbtnSort, javax.swing.GroupLayout.PREFERRED_SIZE, 55,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(kbtnReset, javax.swing.GroupLayout.PREFERRED_SIZE, 58,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap())
    );
    pnlSearchLayout.setVerticalGroup(
        pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlSearchLayout.createSequentialGroup())
        .addGap(5, 5, 5)

        .addGroup(pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

        .addGroup(pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

            .addComponent(kbtnSort, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, 0, Short.MAX_VALUE)

            .addGroup(pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

                .addComponent(cmbOrder, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addComponent(cmbColumn, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(kbtnReset, javax.swing.GroupLayout.PREFERRED_SIZE,
27, javax.swing.GroupLayout.PREFERRED_SIZE)))

.addGroup(pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(lblSearch)

.addGroup(pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(txtSearch, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(cmbIn, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(txtIn)
        .addComponent(txtSort))))
.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);

pnlTable.setBackground(new java.awt.Color(55, 71, 133));
pnlTable.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
pnlTable.setForeground(new java.awt.Color(36, 48, 94));

tblEmployees.setAutoCreateRowSorter(true);
tblEmployees.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {

        },
    new String [] {
        "SHID", "Pay Period", "EID", "Wage", "Date", "SSS EE", "SSS ER", "PH EE", "PH ER",
"HDMF EE", "HDMF ER", "Bonus", "Deduction", "Final Wage", "Final ER"
    }
) {
    boolean[] canEdit = new boolean [] {
        false, false, false, false, false, false, false, false, false, false, false, false, false, false, false, false,
false
    };

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});

```



```

tblEmployees.setToolTipText("");
tblEmployees.setAutoResizeMode(javax.swing.JTable.AUTO_RESIZE_LAST_COLUMN);
tblEmployees.getTableHeader().setReorderingAllowed(false);
tblEmployees.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        tblEmployeesMouseClicked(evt);
    }
});
jScrollPane1.setViewportView(tblEmployees);
if (tblEmployees.getColumnModel().getColumnCount() > 0) {
    tblEmployees.getColumnModel().getColumn(0).setResizable(false);
    tblEmployees.getColumnModel().getColumn(1).setResizable(false);
    tblEmployees.getColumnModel().getColumn(3).setResizable(false);
    tblEmployees.getColumnModel().getColumn(4).setResizable(false);
    tblEmployees.getColumnModel().getColumn(5).setResizable(false);
    tblEmployees.getColumnModel().getColumn(6).setResizable(false);
    tblEmployees.getColumnModel().getColumn(7).setResizable(false);
    tblEmployees.getColumnModel().getColumn(8).setResizable(false);
    tblEmployees.getColumnModel().getColumn(9).setResizable(false);
    tblEmployees.getColumnModel().getColumn(10).setResizable(false);
    tblEmployees.getColumnModel().getColumn(11).setResizable(false);
    tblEmployees.getColumnModel().getColumn(12).setResizable(false);
    tblEmployees.getColumnModel().getColumn(13).setResizable(false);
    tblEmployees.getColumnModel().getColumn(14).setResizable(false);
}

javax.swing.GroupLayout pnlTableLayout = new javax.swing.GroupLayout(pnlTable);
pnlTable.setLayout(pnlTableLayout);
pnlTableLayout.setHorizontalGroup(
    pnlTableLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlTableLayout.createSequentialGroup()
            .addComponent(jScrollPane1)
            .addContainerGap())
);
pnlTableLayout.setVerticalGroup(
    pnlTableLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlTableLayout.createSequentialGroup()
            .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 585,
Short.MAX_VALUE)
            .addContainerGap())
);

```

```

    pnlEmail.setBackground(new java.awt.Color(238, 238, 238));
    pnlEmail.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
    pnlEmail.setForeground(new java.awt.Color(36, 48, 94));

    lblEmail.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
    lblEmail.setForeground(new java.awt.Color(8, 8, 8));
    lblEmail.setText("Salary History");

    javax.swing.GroupLayout pnlEmailLayout = new javax.swing.GroupLayout(pnlEmail);
    pnlEmail.setLayout(pnlEmailLayout);
    pnlEmailLayout.setHorizontalGroup(
        pnlEmailLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlEmailLayout.createSequentialGroup()
                .addGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(lblEmail, javax.swing.GroupLayout.PREFERRED_SIZE, 83,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap())
            .addGroup(
pnlEmailLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(pnlEmailLayout.createSequentialGroup()
                    .addGap()
                    .addComponent(lblEmail, javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                .addGap());

    pnlDelete.setBackground(new java.awt.Color(238, 238, 238));
    pnlDelete.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
    pnlDelete.setForeground(new java.awt.Color(36, 48, 94));

    kbtnDelete.setText("Delete");
    kbtnDelete.setKEndColor(new java.awt.Color(28, 84, 140));
    kbtnDelete.setKHoverEndColor(new java.awt.Color(76, 111, 167));
    kbtnDelete.setKHoverForeGround(new java.awt.Color(252, 250, 255));
    kbtnDelete.setKHoverStartColor(new java.awt.Color(76, 111, 167));
    kbtnDelete.setKSelectedColor(new java.awt.Color(252, 250, 255));
    kbtnDelete.setKStartColor(new java.awt.Color(28, 84, 140));
    kbtnDelete.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {

```

```

        kbtnDeleteActionPerformed(evt);
    }
});

javax.swing.GroupLayout pnlDeleteLayout = new javax.swing.GroupLayout(pnlDelete);
pnlDelete.setLayout(pnlDeleteLayout);
pnlDeleteLayout.setHorizontalGroup(
    pnlDeleteLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlDeleteLayout.createSequentialGroup()
            .addGap(134, 134, 134)
            .addComponent(kbtnDelete, javax.swing.GroupLayout.PREFERRED_SIZE, 78,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(134, Short.MAX_VALUE))
        );
pnlDeleteLayout.setVerticalGroup(
    pnlDeleteLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlDeleteLayout.createSequentialGroup()
            .addGroup(pnlDeleteLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                .addGroup(pnlDeleteLayout.createSequentialGroup()
                    .addContainerGap(10, Short.MAX_VALUE)
                    .addComponent(kbtnDelete, javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addContainerGap())
                .addGroup(pnlDeleteLayout.createSequentialGroup()
                    .addContainerGap(134, 134, 134)
                    .addComponent(kbtnDelete, javax.swing.GroupLayout.PREFERRED_SIZE, 78,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addContainerGap(134, Short.MAX_VALUE))
            .addContainerGap(10, Short.MAX_VALUE))
        );

pnlDelete1.setBackground(new java.awt.Color(238, 238, 238));
pnlDelete1.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
pnlDelete1.setForeground(new java.awt.Color(36, 48, 94));

txtEID.setBackground(new java.awt.Color(252, 250, 255));
txtEID.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtEID.setForeground(new java.awt.Color(8, 8, 8));
txtEID.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtEID.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtEID.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtEIDActionPerformed(evt);
    }
});
txtEID.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtEIDKeyReleased(evt);
    }
}

```

```

        public void keyTyped(java.awt.event.KeyEvent evt) {
            txtEIDKeyTyped(evt);
        }
    });

    lblEID.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
    lblEID.setForeground(new java.awt.Color(8, 8, 8));
    lblEID.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
    lblEID.setText("EID:");

    txtPayPeriod.setBackground(new java.awt.Color(252, 250, 255));
    txtPayPeriod.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
    txtPayPeriod.setForeground(new java.awt.Color(8, 8, 8));
    txtPayPeriod.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
    txtPayPeriod.setMargin(new java.awt.Insets(2, 2, 2, 2));
    txtPayPeriod.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            txtPayPeriodActionPerformed(evt);
        }
    });
    txtPayPeriod.addKeyListener(new java.awt.event.KeyAdapter() {
        public void keyReleased(java.awt.event.KeyEvent evt) {
            txtPayPeriodKeyReleased(evt);
        }
        public void keyTyped(java.awt.event.KeyEvent evt) {
            txtPayPeriodKeyTyped(evt);
        }
    });

    lblPayPeriod.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
    lblPayPeriod.setForeground(new java.awt.Color(8, 8, 8));
    lblPayPeriod.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
    lblPayPeriod.setText("Pay Prd.");

    lblSSSEE.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
    lblSSSEE.setForeground(new java.awt.Color(8, 8, 8));
    lblSSSEE.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
    lblSSSEE.setText("SSS EE:");

    txtSSSEE.setBackground(new java.awt.Color(252, 250, 255));
    txtSSSEE.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
    txtSSSEE.setForeground(new java.awt.Color(8, 8, 8));

```

```

txtSSSEE.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtSSSEE.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtSSSEE.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtSSSEEActionPerformed(evt);
    }
});
txtSSSEE.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtSSSEEKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtSSSEEKeyTyped(evt);
    }
});

```

```

lblSSSER.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblSSSER.setForeground(new java.awt.Color(8, 8, 8));
lblSSSER.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblSSSER.setText("SSS ER:");

```

```

txtSSSER.setBackground(new java.awt.Color(252, 250, 255));
txtSSSER.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtSSSER.setForeground(new java.awt.Color(8, 8, 8));
txtSSSER.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtSSSER.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtSSSER.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtSSSERActionPerformed(evt);
    }
});
txtSSSER.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtSSSERKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtSSSERKeyTyped(evt);
    }
});

```

```

lblPHEE.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblPHEE.setForeground(new java.awt.Color(8, 8, 8));

```

```

lblPHEE.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblPHEE.setText("PH %EE:");

txtPHEE.setBackground(new java.awt.Color(252, 250, 255));
txtPHEE.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtPHEE.setForeground(new java.awt.Color(8, 8, 8));
txtPHEE.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtPHEE.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtPHEE.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtPHEEActionPerformed(evt);
    }
});
txtPHEE.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtPHEEKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtPHEEKeyTyped(evt);
    }
});

lblPHER.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblPHER.setForeground(new java.awt.Color(8, 8, 8));
lblPHER.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblPHER.setText("PH %ER:");

txtPHER.setBackground(new java.awt.Color(252, 250, 255));
txtPHER.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtPHER.setForeground(new java.awt.Color(8, 8, 8));
txtPHER.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtPHER.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtPHER.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtPHERActionPerformed(evt);
    }
});
txtPHER.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtPHERKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {

```

```

        txtPHERKeyTyped(evt);
    }
});

lblHDMFEE.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblHDMFEE.setForeground(new java.awt.Color(8, 8, 8));
lblHDMFEE.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblHDMFEE.setText("%EE:");

txtHDMFEE.setBackground(new java.awt.Color(252, 250, 255));
txtHDMFEE.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtHDMFEE.setForeground(new java.awt.Color(8, 8, 8));
txtHDMFEE.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtHDMFEE.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtHDMFEE.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtHDMFEEActionPerformed(evt);
    }
});
txtHDMFEE.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtHDMFEEKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtHDMFEEKeyTyped(evt);
    }
});

lblHDMFER.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblHDMFER.setForeground(new java.awt.Color(8, 8, 8));
lblHDMFER.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblHDMFER.setText("%ER:");

txtHDMFER.setBackground(new java.awt.Color(252, 250, 255));
txtHDMFER.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtHDMFER.setForeground(new java.awt.Color(8, 8, 8));
txtHDMFER.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtHDMFER.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtHDMFER.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtHDMFERActionPerformed(evt);
    }
}

```

```

});
txtHDMFER.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtHDMFERKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtHDMFERKeyTyped(evt);
    }
});

lblBonus.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblBonus.setForeground(new java.awt.Color(8, 8, 8));
lblBonus.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblBonus.setText("Bonus:");

txtBonus.setBackground(new java.awt.Color(252, 250, 255));
txtBonus.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtBonus.setForeground(new java.awt.Color(8, 8, 8));
txtBonus.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtBonus.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtBonus.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtBonusActionPerformed(evt);
    }
});
txtBonus.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtBonusKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtBonusKeyTyped(evt);
    }
});

lblDeduction.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblDeduction.setForeground(new java.awt.Color(8, 8, 8));
lblDeduction.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblDeduction.setText("Deduction:");

txtDeduction.setBackground(new java.awt.Color(252, 250, 255));
txtDeduction.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtDeduction.setForeground(new java.awt.Color(8, 8, 8));

```



```

txtDeduction.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtDeduction.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtDeduction.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtDeductionActionPerformed(evt);
    }
});
txtDeduction.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtDeductionKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtDeductionKeyTyped(evt);
    }
});

```

```

lblFinalWage.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblFinalWage.setForeground(new java.awt.Color(8, 8, 8));
lblFinalWage.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblFinalWage.setText("Final Wage:");

```

```

txtFinalWage.setBackground(new java.awt.Color(252, 250, 255));
txtFinalWage.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtFinalWage.setForeground(new java.awt.Color(8, 8, 8));
txtFinalWage.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtFinalWage.setFocusable(false);
txtFinalWage.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtFinalWage.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtFinalWageActionPerformed(evt);
    }
});
txtFinalWage.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtFinalWageKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtFinalWageKeyTyped(evt);
    }
});

```

```

kbtnClearFields.setText("Clear Fields");

```

```

kbtnClearFields.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnClearFields.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnClearFields.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnClearFields.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnClearFields.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnClearFields.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnClearFields.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnClearFieldsActionPerformed(evt);
    }
});

```

```

kbtnDeselect.setText("Deselect");
kbtnDeselect.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnDeselect.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnDeselect.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnDeselect.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnDeselect.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnDeselect.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnDeselect.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnDeselectActionPerformed(evt);
    }
});

```

```

kbtnAdd.setText("Add");
kbtnAdd.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnAdd.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnAdd.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnAdd.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnAdd.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnAdd.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnAdd.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnAddActionPerformed(evt);
    }
});

```

```

lblWage.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblWage.setForeground(new java.awt.Color(8, 8, 8));
lblWage.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblWage.setText("Wage:");

```

```

txtWage.setBackground(new java.awt.Color(252, 250, 255));

```

```

txtWage.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtWage.setForeground(new java.awt.Color(8, 8, 8));
txtWage.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtWage.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtWage.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtWageActionPerformed(evt);
    }
});
txtWage.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtWageKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtWageKeyTyped(evt);
    }
});

lblBirthdate.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblBirthdate.setForeground(new java.awt.Color(8, 8, 8));
lblBirthdate.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblBirthdate.setText("Date:");

jCalendarDateSaved.setFocusable(false);
jCalendarDateSaved.addChangeListener(new javax.swing.event.ChangeListener() {
    public void stateChanged(javax.swing.event.ChangeEvent evt) {
        jCalendarDateSavedStateChanged(evt);
    }
});

lblFinalER.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblFinalER.setForeground(new java.awt.Color(8, 8, 8));
lblFinalER.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblFinalER.setText("Final ER:");

txtFinalER.setBackground(new java.awt.Color(252, 250, 255));
txtFinalER.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtFinalER.setForeground(new java.awt.Color(8, 8, 8));
txtFinalER.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtFinalER.setFocusable(false);
txtFinalER.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtFinalER.addActionListener(new java.awt.event.ActionListener() {

```

```

        public void actionPerformed(java.awt.event.ActionEvent evt) {
            txtFinalERActionPerformed(evt);
        }
    });
    txtFinalER.addKeyListener(new java.awt.event.KeyAdapter() {
        public void keyReleased(java.awt.event.KeyEvent evt) {
            txtFinalERKeyReleased(evt);
        }
        public void keyPressed(java.awt.event.KeyEvent evt) {
            txtFinalERKeyPressed(evt);
        }
    });

    javax.swing.GroupLayout pnlDelete1Layout = new javax.swing.GroupLayout(pnlDelete1);
    pnlDelete1.setLayout(pnlDelete1Layout);
    pnlDelete1Layout.setHorizontalGroup(
        pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(pnlDelete1Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(txtFinalER, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addContainerGap())
            .addGroup(pnlDelete1Layout.createSequentialGroup()
                .addComponent(lblPHEE, javax.swing.GroupLayout.PREFERRED_SIZE, 90, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(lblSSSEE, javax.swing.GroupLayout.PREFERRED_SIZE, 90, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap())
    );
    pnlDelete1Layout.setVerticalGroup(
        pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(pnlDelete1Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(txtFinalER, javax.swing.GroupLayout.PREFERRED_SIZE, 28, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(lblPHEE)
                    .addComponent(lblSSSEE))
                .addContainerGap(76, Short.MAX_VALUE))
    );

```

```

        .addComponent(txtSSSEE,
javax.swing.GroupLayout.PREFERRED_SIZE, 61,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(txtPHEE,
javax.swing.GroupLayout.PREFERRED_SIZE, 61,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGroup(pnlDelete1Layout.createSequentialGroup()
        .addGap(21, 21, 21)
        .addComponent(lblHDMFEE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(txtHDMFEE,
javax.swing.GroupLayout.PREFERRED_SIZE, 61,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGroup(pnlDelete1Layout.createSequentialGroup()
        .addComponent(lblWage, javax.swing.GroupLayout.PREFERRED_SIZE,
42, javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(txtWage, javax.swing.GroupLayout.PREFERRED_SIZE,
91, javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(lblBirthdate)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(jCalendarDateSaved,
javax.swing.GroupLayout.PREFERRED_SIZE, 122,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(8, 8, 8))
        .addGroup(pnlDelete1Layout.createSequentialGroup()
        .addContainerGap()

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
G)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
G)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlDelete1Layout.createSequentialGroup()
        .addComponent(lblBonus)
        .addGap(224, 224, 224))
        .addGroup(pnlDelete1Layout.createSequentialGroup()
        .addGap(82, 82, 82)

```

```

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                .addGroup(pnlDelete1Layout.createSequentialGroup()

                    .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                        .addGroup(pnlDelete1Layout.createSequentialGroup()

                            .addGap(90, 90, 90)
                            .addComponent(lblSSSER))
                            .addComponent(lblPHER,
javax.swing.GroupLayout.Alignment.TRAILING))

                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

                    .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                        .addComponent(txtPHER,
javax.swing.GroupLayout.PREFERRED_SIZE, 61,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(txtSSSER,
javax.swing.GroupLayout.PREFERRED_SIZE, 61,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlDelete1Layout.createSequentialGroup()
                            .addComponent(lblHDMFER)

                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                            .addComponent(txtHDMFER,
javax.swing.GroupLayout.PREFERRED_SIZE, 61,
javax.swing.GroupLayout.PREFERRED_SIZE))))))
                            .addGroup(pnlDelete1Layout.createSequentialGroup()
                                .addComponent(lblEID)
                                .addGap(18, 18, 18)
                                .addComponent(txtEID, javax.swing.GroupLayout.PREFERRED_SIZE, 93,
javax.swing.GroupLayout.PREFERRED_SIZE))

                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                                .addComponent(lblPayPeriod)

                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                                .addComponent(txtPayPeriod,
javax.swing.GroupLayout.PREFERRED_SIZE, 103,
javax.swing.GroupLayout.PREFERRED_SIZE))

```

```

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
    .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
        .addComponent(txtBonus,
javax.swing.GroupLayout.PREFERRED_SIZE, 199,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
pnlDelete1Layout.createSequentialGroup())

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING,
G)
    .addComponent(lblFinalWage,
javax.swing.GroupLayout.PREFERRED_SIZE, 76,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(lblDeduction))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
G, false)
    .addComponent(txtDeduction,
javax.swing.GroupLayout.DEFAULT_SIZE, 193, Short.MAX_VALUE)
    .addComponent(txtFinalWage))))
    .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
pnlDelete1Layout.createSequentialGroup())
        .addGap(6, 6, 6)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
G)
    .addGroup(pnlDelete1Layout.createSequentialGroup())
        .addComponent(lblFinalER,
javax.swing.GroupLayout.PREFERRED_SIZE, 76,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(txtFinalER))
    .addGroup(pnlDelete1Layout.createSequentialGroup())
        .addComponent(kbtnClearFields,
javax.swing.GroupLayout.PREFERRED_SIZE, 82,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(39, 39, 39)

```

```

        .addComponent(kbtnDeselect,
javax.swing.GroupLayout.PREFERRED_SIZE, 82,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 34,
Short.MAX_VALUE)

        .addComponent(kbtnAdd,
javax.swing.GroupLayout.PREFERRED_SIZE, 74,
javax.swing.GroupLayout.PREFERRED_SIZE))))))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );
    pnlDelete1Layout.setVerticalGroup(
        pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlDelete1Layout.createSequentialGroup()
            .addContainerGap()

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(txtPayPeriod, javax.swing.GroupLayout.PREFERRED_SIZE,
20, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(lblPayPeriod))

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblEID)
    .addComponent(txtEID, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    .addComponent(txtWage, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(lblWage)
    .addComponent(lblBirthdate)
    .addComponent(jCalendarDateSaved,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(14, 14, 14)

```



```

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(pnlDelete1Layout.createSequentialGroup()

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblSSSER)
    .addComponent(txtSSSER, javax.swing.GroupLayout.PREFERRED_SIZE,
20, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(2, 2, 2)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(txtPHER, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(lblPHER))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblHDMFER)
    .addComponent(txtHDMFER, javax.swing.GroupLayout.PREFERRED_SIZE,
20, javax.swing.GroupLayout.PREFERRED_SIZE)))
    .addGroup(pnlDelete1Layout.createSequentialGroup()

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblSSSEE)
    .addComponent(txtSSSEE, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(2, 2, 2)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(txtPHEE, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(lblPHEE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblHDMFEE)

```

```

        .addComponent(txtHDMFEE, javax.swing.GroupLayout.PREFERRED_SIZE,
20, javax.swing.GroupLayout.PREFERRED_SIZE))))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASLI
NE)
        .addComponent(lblBonus)
        .addComponent(txtBonus, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASLI
NE)
        .addComponent(lblDeduction)
        .addComponent(txtDeduction, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASLI
NE)
        .addComponent(lblFinalWage)
        .addComponent(txtFinalWage, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASLI
NE)
        .addComponent(lblFinalER)
        .addComponent(txtFinalER, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 211,
Short.MAX_VALUE)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASLI
NE)
        .addComponent(kbtnClearFields, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(kbtnDeselect, javax.swing.GroupLayout.PREFERRED_SIZE, 45,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(kbtnAdd, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(14, 14, 14))
);

```

```

    pnlEmail1.setBackground(new java.awt.Color(238, 238, 238));
    pnlEmail1.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
    pnlEmail1.setForeground(new java.awt.Color(36, 48, 94));

    kbtnSalaryGenerator.setText("Salary Generator");
    kbtnSalaryGenerator.setkEndColor(new java.awt.Color(28, 84, 140));
    kbtnSalaryGenerator.setkHoverEndColor(new java.awt.Color(76, 111, 167));
    kbtnSalaryGenerator.setkHoverForeGround(new java.awt.Color(252, 250, 255));
    kbtnSalaryGenerator.setkHoverStartColor(new java.awt.Color(76, 111, 167));
    kbtnSalaryGenerator.setkSelectedColor(new java.awt.Color(252, 250, 255));
    kbtnSalaryGenerator.setkStartColor(new java.awt.Color(28, 84, 140));
    kbtnSalaryGenerator.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            kbtnSalaryGeneratorActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout pnlEmail1Layout = new javax.swing.GroupLayout(pnlEmail1);
    pnlEmail1.setLayout(pnlEmail1Layout);
    pnlEmail1Layout.setHorizontalGroup(
        pnlEmail1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(pnlEmail1Layout.createSequentialGroup()
                .add(pnlEmail1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .add(pnlEmail1Layout.createSequentialGroup()
                        .addContainerGap()
                        .addComponent(kbtnSalaryGenerator, javax.swing.GroupLayout.DEFAULT_SIZE,
223, Short.MAX_VALUE)
                        .addContainerGap())
                    .add(pnlEmail1Layout.createSequentialGroup()
                        .addComponent(kbtnSalaryGenerator, javax.swing.GroupLayout.PREFERRED_SIZE,
32, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(18, 18, 18))
                )
            );

    pnlSearch2.setBackground(new java.awt.Color(238, 238, 238));
    pnlSearch2.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
    pnlSearch2.setForeground(new java.awt.Color(36, 48, 94));

    txtSort2.setFont(new java.awt.Font("Segoe UI Light", 0, 16)); // NOI18N

```

```

txtSort2.setForeground(new java.awt.Color(8, 8, 8));

ktbtnPrint.setText("Print");
ktbtnPrint.setkEndColor(new java.awt.Color(28, 84, 140));
ktbtnPrint.setkHoverEndColor(new java.awt.Color(76, 111, 167));
ktbtnPrint.setkHoverForeGround(new java.awt.Color(252, 250, 255));
ktbtnPrint.setkHoverStartColor(new java.awt.Color(76, 111, 167));
ktbtnPrint.setkSelectedColor(new java.awt.Color(252, 250, 255));
ktbtnPrint.setkStartColor(new java.awt.Color(28, 84, 140));
ktbtnPrint.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ktbtnPrintActionPerformed(evt);
    }
});

javax.swing.GroupLayout pnlSearch2Layout = new javax.swing.GroupLayout(pnlSearch2);
pnlSearch2.setLayout(pnlSearch2Layout);
pnlSearch2Layout.setHorizontalGroup(
    pnlSearch2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlSearch2Layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addComponent(ktbtnPrint, javax.swing.GroupLayout.PREFERRED_SIZE, 55,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(233, 233, 233)
            .addComponent(txtSort2))
        );
pnlSearch2Layout.setVerticalGroup(
    pnlSearch2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlSearch2Layout.createSequentialGroup()
            .addGap(5, 5, 5)
            .addGroup(pnlSearch2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(ktbtnPrint, javax.swing.GroupLayout.PREFERRED_SIZE, 29,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(txtSort2))
            .addGap(10, 10, 10))
        );

javax.swing.GroupLayout pnlBackgroundLayout = new
javax.swing.GroupLayout(pnlBackground);
pnlBackground.setLayout(pnlBackgroundLayout);
pnlBackgroundLayout.setHorizontalGroup(

```

```

pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(pnlBackgroundLayout.createSequentialGroup())
    .addContainerGap()

    .addGroup(pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(pnlTable, javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addGroup(pnlBackgroundLayout.createSequentialGroup())
            .addComponent(pnlSearch, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(pnlSearch2, javax.swing.GroupLayout.PREFERRED_SIZE, 76,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(0, 0, Short.MAX_VALUE)))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addGroup(pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlBackgroundLayout.createSequentialGroup())
            .addComponent(pnlEmail1, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(pnlEmail, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

    .addGroup(pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
        .addComponent(pnlDelete1, javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(pnlDelete, javax.swing.GroupLayout.PREFERRED_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGap(5, 5, 5))
    );
pnlBackgroundLayout.setVerticalGroup(

pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(pnlBackgroundLayout.createSequentialGroup())
        .addGap(4, 4, 4)

    .addGroup(pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addComponent(pnlSearch, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(pnlSearch2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGroup(pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addComponent(pnlEmail1, javax.swing.GroupLayout.PREFERRED_SIZE, 52,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(pnlEmail, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(pnlBackgroundLayout.createSequentialGroup()
                .addComponent(pnlDelete1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(pnlDelete, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addComponent(pnlTable, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            .addContainerGap())
    );

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(pnlBackground, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(pnlBackground, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
    );

    pack();
    setLocationRelativeTo(null);

```

```
// </editor-fold>
```

```
private void ktbtnPrintActionPerformed(java.awt.event.ActionEvent evt) {

    try {
        Date d = new Date();
        String sd = d.toString();
        MessageFormat header = new MessageFormat("Salary History");
        MessageFormat footer = new MessageFormat(sd);
        PrintRequestAttributeSet set = new HashPrintRequestAttributeSet();
        set.add(OrientationRequested.PORTRAIT);
        tblEmployees.print(JTable.PrintMode.FIT_WIDTH, header, footer, true, set, true);
        JOptionPane.showMessageDialog(null, "\n Printed Successfully");
    }
    catch (java.awt.print.PrinterException e) {
        JOptionPane.showMessageDialog(null, "\n Printing Failed");
    }

    /*try
    {
        // Step 1: Initialize PDF Document
        PDDocument document = new PDDocument();
        PDPage page = new PDPage();
        document.addPage(page);

        // Step 2: Content Stream for Writing Text
        PDPageContentStream contentStream = new PDPageContentStream(document, page);
        PDType1Font font = new PDType1Font(HELVETICA);
        contentStream.setFont(font, 12);

        // Step 3: Get JTable Data
        TableModel model = tblEmployees.getModel();
        float margin = 50;
        float yPosition = page.getMediaBox().getHeight() - margin; // Start from top
        float tableRowHeight = 20f;
        float tableColumnWidth = (page.getMediaBox().getWidth() - 2 * margin) /
model.getColumnCount();

        // Step 4: Write Column Headers to PDF
        contentStream.beginText();
        contentStream.newLineAtOffset(margin, yPosition);
        for (int col = 0; col < model.getColumnCount(); col++)
        {
```

```

        contentStream.showText(String.format("%-15s", model.getColumnName(col))); //
Adjust width if needed
        contentStream.newLineAtOffset(tableColumnWidth, 0);
    }
    contentStream.endText();

    yPosPosition -= tableRowHeight;

    // Step 5: Write Table Rows to PDF
    for (int row = 0; row < model.getRowCount(); row++)
    {
        contentStream.beginText();
        contentStream.newLineAtOffset(margin, yPosPosition);

        for (int col = 0; col < model.getColumnCount(); col++)
        {
            Object objtemp = model.getValueAt(row, col);

            String str = "";
            if (objtemp==null){
                str = "NULL";
            }
            else{
                str = objtemp.toString();
            }
            contentStream.showText(String.format("%-15s", str)); // Adjust width if needed
            contentStream.newLineAtOffset(tableColumnWidth, 0);
        }
        contentStream.endText();
        yPosPosition -= tableRowHeight;
    }

    // Close the content stream
    contentStream.close();

    // Save the document
    document.save("D:/IA PDFs");
    document.close();

    System.out.println("PDF exported successfully.");
}
catch (Exception e)
{
    e.printStackTrace();
}

```



```

    */
}

private void kbtnSalaryGeneratorActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        new SalaryGenerator(this.uid,this.email).setVisible(true);
        this.setVisible(false);
    }
    catch (Exception e){

    }
}

private void kbtnDeselectActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        //Deselects all rows
        // DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
        // tblEmployees.removeRowSelectionInterval(0, model.getRowCount()-1);
        tblEmployees.clearSelection();
        //resets selectedroweid
        this.selectedroweid = "-1";
    }
    catch (Exception e){

    }
}

private void kbtnClearFieldsActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        txtEID.setText("");
        txtPayPeriod.setText("");
        jCalendarDateSaved.setDate(new Date());
        txtWage.setText("");
        txtSSSEE.setText("");
        txtSSSER.setText("");
        txtPHEE.setText("");
        txtPHER.setText("");
        txtHDMFEE.setText("");
        txtHDMFER.setText("");
        txtDeduction.setText("");
        txtBonus.setText("");
        txtFinalWage.setText("");
        txtFinalER.setText("");
    }
}

```

```

        catch(Exception e){

        }
    }

    private void txtFinalWageKeyTyped(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
    }

    private void txtFinalWageKeyReleased(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
    }

    private void txtFinalWageActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void txtDeductionKeyTyped(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
    }

    private void txtDeductionKeyReleased(java.awt.event.KeyEvent evt) {
        calcFinalWage();
    }

    private void txtDeductionActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void txtBonusKeyTyped(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
    }

    private void txtBonusKeyReleased(java.awt.event.KeyEvent evt) {
        calcFinalWage();
    }

    private void txtBonusActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void txtHDMFERKeyTyped(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
    }

```

```

private void txtHDMFERKeyReleased(java.awt.event.KeyEvent evt) {
    calcFinalWage();
}

private void txtHDMFERActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtHDMFEEKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtHDMFEEKeyReleased(java.awt.event.KeyEvent evt) {
    calcFinalWage();
}

private void txtHDMFEEActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtPHERKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtPHERKeyReleased(java.awt.event.KeyEvent evt) {
    calcFinalWage();
}

private void txtPHERActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtPHEEKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtPHEEKeyReleased(java.awt.event.KeyEvent evt) {
    calcFinalWage();
}

private void txtPHEEActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

```

```

private void txtSSSERKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtSSSERKeyReleased(java.awt.event.KeyEvent evt) {
    calcFinalWage();
}

private void txtSSSERActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtSSSEEKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtSSSEEKeyReleased(java.awt.event.KeyEvent evt) {
    calcFinalWage();
}

private void txtSSSEEActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtPayPeriodKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtPayPeriodKeyReleased(java.awt.event.KeyEvent evt) {

}

private void txtPayPeriodActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtEIDKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtEIDKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

```

```

private void txtEIDActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void tblEmployeesMouseClicked(java.awt.event.MouseEvent evt) {
    try{
        //Prerequisite for the update function
        DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
        int selectedrow=tblEmployees.getSelectedRow();
        if (tblEmployees.getRowSorter()!=null) {
            //Gets the correct row for a filtered jtable (RowFilter)
            selectedrow = tblEmployees.getRowSorter().convertRowIndexToModel(selectedrow);
        }
        String shid = "";
        String eid = "";
        String payperiod = "";
        String strd = "";
        String wage = "";
        String sssee = "";
        String ssser = "";
        String phee = "";
        String pher = "";
        String hdmfee="";
        String hdmfer="";
        String bonus = "";
        String deduction = "";
        String finalwage = "";
        String finaler = "";

        shid = model.getValueAt(selectedrow, 0).toString();
        payperiod = model.getValueAt(selectedrow,1).toString();
        eid = model.getValueAt(selectedrow,2).toString();
        strd = model.getValueAt(selectedrow,3).toString();
        wage = model.getValueAt(selectedrow,4).toString();
        sssee = model.getValueAt(selectedrow,5).toString();
        ssser = model.getValueAt(selectedrow,6).toString();
        phee = model.getValueAt(selectedrow,7).toString();
        pher = model.getValueAt(selectedrow,8).toString();
        hdmfee=model.getValueAt(selectedrow,9).toString();
        hdmfer=model.getValueAt(selectedrow,10).toString();
        bonus = model.getValueAt(selectedrow,11).toString();
        deduction = model.getValueAt(selectedrow,12).toString();
    }
}

```

```

        finalwage = model.getValueAt(selectedrow,13).toString();
        finaler = model.getValueAt(selectedrow,14).toString();

        this.selectedroweid = eid;
        txtEID.setText(eid);
        txtPayPeriod.setText(payperiod);

        SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd");
        Date d = formatter.parse(strd);
        jCalendarDateSaved.setDate(d);

        txtPayPeriod.setText(wage);
        txtSSSEE.setText(sssee);
        txtSSSER.setText(ssser);
        txtPHEE.setText(phee);
        txtPHER.setText(pher);
        txtHDMFEE.setText(hdmfee);
        txtHDMFER.setText(hdmfer);
        txtBonus.setText(bonus);
        txtDeduction.setText(deduction);
        txtFinalWage.setText(finalwage);
        txtFinalER.setText(finaler);

    }
    catch (Exception e){
        e.printStackTrace();
    }
}

private void kbtnResetActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        //Resets the search and sort functions; restores table to original
        refreshtable("SELECT * FROM salary_history");
        cmbColumn.setSelectedIndex(0);
        cmbOrder.setSelectedIndex(0);

        txtSearch.setText("");
        cmbIn.setSelectedIndex(0);

        search();
    }
    catch (Exception e){

    }
}

```

```

}

private void kbtnSortActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        String search = txtSearch.getText().trim();

        int column = cmbColumn.getSelectedIndex();

        String[] columnnames = {"shid","pay_period", "eid", "datesaved", "monthly_wage_rate",
"sss_ee", "sss_er", "ph_ee_p", "ph_er_p", "hdmf_ee_p", "hdmf_er_p", "deduction", "bonus",
"final_wage", "final_er"};
        String strcolumn = columnnames[column];

        int sort = cmbOrder.getSelectedIndex();
        String strsort = "";
        switch (sort){
            case 0:
                strsort = "ASC";
                break;
            case 1:
                strsort = "DESC";
        }

        if (column==0 && sort==0){
            refreshtable("SELECT * FROM salary_history");
        }
        //      else if (column==0 && sort==1){
        //          //Iterate through the resultset backwards
        //          DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
        //          String databaseURL = "jdbc:derby:SALARY_GENERATOR";
        //          Connection con = DriverManager.getConnection(databaseURL);
        //          Statement st =
con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_READ_ONL
Y);
        //          String sql = "SELECT * FROM salary_allocations";
        //          ResultSet rs = st.executeQuery(sql);
        //
        //          //Moves the cursor to the end of the ResultSet object
        //          rs.afterLast();
        //          model.setRowCount(0);
        //          while(rs.previous())
        //          {
        //              model.addRow(new
Object[]{rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4)});

```

```

        //      }
        //      rs.close();
        //      st.close();
        //      con.close();
        //      }
    else{
        refreshtable("select * from salary_history order by "+strcolumn+" "+strsort);
    }

}
catch (Exception e){

}

}

private void cmbColumnActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void cmbInPropertyChange(java.beans.PropertyChangeEvent evt) {

}

private void cmbInActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void cmbInItemStateChanged(java.awt.event.ItemEvent evt) {
    try{
        search();
    }
    catch (Exception e){

    }
}

private void txtSearchKeyTyped(java.awt.event.KeyEvent evt) {

}

private void txtSearchKeyReleased(java.awt.event.KeyEvent evt) {
    try{
        search();
    }
}

```



```

        catch (Exception e){

        }
    }

    private void txtSearchActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void kbtnAddActionPerformed(java.awt.event.ActionEvent evt) {
        try{
            //      String eid = "";

            String eid = "";
            String payperiod = "";
            Date d;
            String wage = "";
            String sssee = "";
            String ssser = "";
            String phee = "";
            String pher = "";
            String hdmfee = "";
            String hdmfer = "";
            String bonus = "";
            String deduction = "";
            String finalwage = "";
            String finaler = "";

            //      eid = txtEID.getText();
            eid = txtEID.getText();
            payperiod = txtPayPeriod.getText();
            wage = txtWage.getText();
            d = jCalendarDateSaved.getDate();
            sssee = txtSSSEE.getText();
            ssser = txtSSSER.getText();
            phee = txtPHEE.getText();
            pher = txtPHER.getText();
            hdmfee = txtHDMFEE.getText();
            hdmfer = txtHDMFER.getText();
            bonus = txtBonus.getText();
            deduction = txtDeduction.getText();
            finalwage = txtFinalWage.getText();
            finaler = txtFinalER.getText();

```

```

System.out.println("Email: " + email);

//Error trapping
if ("".equals(wage)){
    JOptionPane.showMessageDialog(null, "Wage field is empty.");
}
else if ("".equals(sssee)){
    JOptionPane.showMessageDialog(null, "SSSEE field is empty.");
}
else if ("".equals(ssser)){
    JOptionPane.showMessageDialog(null, "SSSER field is empty.");
}
else if ("".equals(phee)){
    JOptionPane.showMessageDialog(null, "PHEE field is empty.");
}
else if ("".equals(pher)){
    JOptionPane.showMessageDialog(null, "PHER field is empty.");
}
else if ("".equals(hdmfee)){
    JOptionPane.showMessageDialog(null, "HDMFEE field is empty.");
}
else if ("".equals(hdmfer)){
    JOptionPane.showMessageDialog(null, "HDMFER field is empty.");
}
else if ("".equals(bonus)){
    JOptionPane.showMessageDialog(null, "Bonus field is empty.");
}
else if ("".equals(deduction)){
    JOptionPane.showMessageDialog(null, "Deduction field is empty.");
}
else if (Double.parseDouble(wage)>99999999 ||
Double.parseDouble(wage)!=((double)Math.round(Double.parseDouble(wage)*100)/100){
    JOptionPane.showMessageDialog(null, "Wage can be a maximum of 99,999,999.99.
Decimal precision is only 2 decimal places.");
}
else if (Double.parseDouble(sssee)>99999999 ||
Double.parseDouble(sssee)!=((double)Math.round(Double.parseDouble(sssee)*100)/100){
    JOptionPane.showMessageDialog(null, "SSSEE be a maximum of 99,999,999.99.
Decimal precision is only 2 decimal places.");
}
else if (Double.parseDouble(ssser)>99999999 ||
Double.parseDouble(ssser)!=((double)Math.round(Double.parseDouble(ssser)*100)/100){
    JOptionPane.showMessageDialog(null, "SSSER can be a maximum of
99,999,999.99. Decimal precision is only 2 decimal places.");
}

```

```

    }
    else if (Double.parseDouble(phee)>99999999 ||
Double.parseDouble(phee)!=((double)Math.round(Double.parseDouble(phee)*100)/100){
        JOptionPane.showMessageDialog(null, "PHEE can be a maximum of 99,999,999.99.
Decimal precision is only 2 decimal places.");
    }
    else if (Double.parseDouble(pher)>99999999 ||
Double.parseDouble(pher)!=((double)Math.round(Double.parseDouble(pher)*100)/100){
        JOptionPane.showMessageDialog(null, "PHER can be a maximum of 99,999,999.99.
Decimal precision is only 2 decimal places.");
    }
    else if (Double.parseDouble(hdmfee)>99999999 ||
Double.parseDouble(hdmfee)!=((double)Math.round(Double.parseDouble(hdmfee)*100)/100){
        JOptionPane.showMessageDialog(null, "HDMFEE can be a maximum of
99,999,999.99. Decimal precision is only 2 decimal places.");
    }
    else if (Double.parseDouble(ssser)>99999999 ||
Double.parseDouble(ssser)!=((double)Math.round(Double.parseDouble(ssser)*100)/100){
        JOptionPane.showMessageDialog(null, "HDMFER can be a maximum of
99,999,999.99. Decimal precision is only 2 decimal places.");
    }
    else if (Double.parseDouble(deduction)>99999999 ||
Double.parseDouble(deduction)!=((double)Math.round(Double.parseDouble(deduction)*100)/100
){
        JOptionPane.showMessageDialog(null, "Deduction can be a maximum of
99,999,999.99. Decimal precision is only 2 decimal places.");
    }
    else if (Double.parseDouble(bonus)>99999999 ||
Double.parseDouble(bonus)!=((double)Math.round(Double.parseDouble(bonus)*100)/100){
        JOptionPane.showMessageDialog(null, "Bonus can be a maximum of 99,999,999.99.
Decimal precision is only 2 decimal places.");
    }
    else{
        maxfinder();
        this.max++;
        String strmax = Integer.toString(this.max);

        System.out.println("strmax: " + strmax);

        double dwage = Double.parseDouble(txtWage.getText());
        double dsssee = Double.parseDouble(txtSSSEE.getText());
        double dphee = Double.parseDouble(txtPHEE.getText());
        double dhdmfee = Double.parseDouble(txtHDMFEE.getText());
        double ddeduction = Double.parseDouble(txtDeduction.getText());

```

```

double dbonus = Double.parseDouble(txtBonus.getText());

double dfinalwage = dwage - dsssee - (dphee*dwage/100) - (dhdmfee*dwage/100) -
ddeduction + dbonus;

double dssser = Double.parseDouble(txtSSSER.getText());
double dpheer = Double.parseDouble(txtPHER.getText());
double dhdmfer = Double.parseDouble(txtHDMFER.getText());

double dfinaler = dssser + dpheer*dwage/100 + dhdmfer*dwage/100;

System.out.println("Dfinalwage: "+dfinalwage);
System.out.println("Dfinaler: "+dfinaler);

String databaseURL = "jdbc:derby:SALARY_GENERATOR";
Connection con = DriverManager.getConnection(databaseURL);
Statement st = con.createStatement();

Date addeddate = new java.sql.Date(d.getTime());

String sql = "INSERT INTO salary_history (shid, pay_period, eid, datesaved,
monthly_wage_rate, sss_ee, sss_er, ph_ee_p, ph_er_p, hdmf_ee_p, hdmf_er_p, deduction,
bonus, final_wage, final_er) "
+ "VALUES (" +strmax+", "+payperiod+", "+eid+", "+new
java.sql.Timestamp(d.getTime())+", "+wage+", "+sssee+", "+ssser+", "+phee+", "+pher+", "
+
hdmfee+", "+hdmfer+", "+deduction+", "+bonus+", "+Double.toString(dfinalwage)+", "+Double.toStri
ng(dfinaler)+")";

st.execute(sql);
con.close();

refreshable("select * from salary_history");

txtEID.setText("");
txtPayPeriod.setText("");
txtWage.setText("");
jCalendarDateSaved.setDate(new Date());
txtSSSEE.setText("");
txtSSSER.setText("");
txtPHEE.setText("");
txtPHER.setText("");
txtHDMFEE.setText("");
txtHDMFER.setText("");

```

```

        txtBonus.setText("");
        txtDeduction.setText("");
        txtFinalWage.setText("");
        txtFinalER.setText("");
        JOptionPane.showMessageDialog(null, "Added.");
    }
}
catch (Exception e){
    e.printStackTrace();
}
}

private void kbtnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
        int rowcount = model.getRowCount();

        //Allows it to work with the filtered jTable (RowFilter)
        int rows[] = tblEmployees.getSelectedRows();

        for(int i = 0; i<rows.length; i++){
            System.out.println(rows[i]);
        }

        for (int i = 0; i<rows.length; i++){
            rows[i] = tblEmployees.getRowSorter().convertRowIndexToModel(rows[i]);
        }

        if ("-1".equals(this.selectedroweid)){
            JOptionPane.showMessageDialog(null, "A row has not been selected.");
        }
        else{
            int confirm = JOptionPane.showConfirmDialog(null,"Are you sure?");

            if (confirm==0){
                if (rows.length>0){//Delete multiple or single
                    int eids[] = new int[rows.length];
                    for (int i = 0; i<rows.length; i++){
                        eids[i] = Integer.parseInt(model.getValueAt(rows[i],0).toString());
                    }
                    String databaseURL = "jdbc:derby:SALARY_GENERATOR";
                    Connection con = DriverManager.getConnection(databaseURL);

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

```

```

        Statement st = con.createStatement();

        String sql = "DELETE FROM salary_history WHERE shid=" + eids[i];
        st.execute(sql);
    }
    con.close();

    //          txtFullName.setText("");
    //          txtEmail.setText("");
    //          txtPhoneNumber.setText("");

    this.selectedroweid = "-1";
    refreshtable("select * from salary_history");

    maxfinder();
    }
    }
    }

    }
    catch (Exception e){

    }
}

private void txtWageActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtWageKeyReleased(java.awt.event.KeyEvent evt) {

}

private void txtWageKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void jCalendarDateSavedStateChanged(javax.swing.event.ChangeEvent evt) {

}

private void txtFinalERActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

```

```

    }

    private void txtFinalERKeyReleased(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
    }

    private void txtFinalERKeyTyped(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
                javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

            java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {

            java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

            java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

```

```

java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Register().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JComboBox<String> cmbColumn;
private javax.swing.JComboBox<String> cmbIn;
private javax.swing.JComboBox<String> cmbOrder;
private de.wannawork.jcalendar.JCalendarComboBox jCalendarDateSaved;
private javax.swing.JScrollPane jScrollPane1;
private com.k33ptoo.components.KButton kbtnAdd;
private com.k33ptoo.components.KButton kbtnClearFields;
private com.k33ptoo.components.KButton kbtnDelete;
private com.k33ptoo.components.KButton kbtnDeselect;
private com.k33ptoo.components.KButton kbtnReset;
private com.k33ptoo.components.KButton kbtnSalaryGenerator;
private com.k33ptoo.components.KButton kbtnSort;
private com.k33ptoo.components.KButton kbtnPrint;
private javax.swing.JLabel lblBirthdate;
private javax.swing.JLabel lblBonus;
private javax.swing.JLabel lblDeduction;
private javax.swing.JLabel lblEID;
private javax.swing.JLabel lblEmail;
private javax.swing.JLabel lblFinalER;
private javax.swing.JLabel lblFinalWage;
private javax.swing.JLabel lblHDMFEE;
private javax.swing.JLabel lblHDMFER;
private javax.swing.JLabel lblPHEE;
private javax.swing.JLabel lblPHER;
private javax.swing.JLabel lblPayPeriod;
private javax.swing.JLabel lblSSSEE;
private javax.swing.JLabel lblSSSER;
private javax.swing.JLabel lblSearch;
private javax.swing.JLabel lblWage;

```



```

private com.k33ptoo.components.KGradientPanel pnlBackground;
private javax.swing.JPanel pnlDelete;
private javax.swing.JPanel pnlDelete1;
private javax.swing.JPanel pnlEmail;
private javax.swing.JPanel pnlEmail1;
private javax.swing.JPanel pnlSearch;
private javax.swing.JPanel pnlSearch2;
private javax.swing.JPanel pnlTable;
private javax.swing.JTable tblEmployees;
private javax.swing.JTextField txtBonus;
private javax.swing.JTextField txtDeduction;
private javax.swing.JTextField txtEID;
private javax.swing.JTextField txtFinalER;
private javax.swing.JTextField txtFinalWage;
private javax.swing.JTextField txtHDMFEE;
private javax.swing.JTextField txtHDMFER;
private javax.swing.JLabel txtIn;
private javax.swing.JTextField txtPHEE;
private javax.swing.JTextField txtPHER;
private javax.swing.JTextField txtPayPeriod;
private javax.swing.JTextField txtSSSEE;
private javax.swing.JTextField txtSSSER;
private javax.swing.JTextField txtSearch;
private javax.swing.JLabel txtSort;
private javax.swing.JLabel txtSort2;
private javax.swing.JTextField txtWage;
// End of variables declaration
}

```

Loan History Page:

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */

/**
 *
 * @author User
 */
import java.awt.Color;
import java.sql.Connection;
import java.sql.DatabaseMetaData;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.MessageFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.print.attribute.HashPrintRequestAttributeSet;
import javax.print.attribute.PrintRequestAttributeSet;
import javax.print.attribute.standard.OrientationRequested;
import javax.swing.JOptionPane;
import javax.swing.JTable;
import javax.swing.RowFilter;
import javax.swing.SwingUtilities;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableModel;
import javax.swing.table.TableRowSorter;

import javax.swing.text.*;

import org.apache.pdfbox.pdmodel.PDDocument;
import org.apache.pdfbox.pdmodel.PDPage;
import org.apache.pdfbox.pdmodel.PDPageContentStream;
import org.apache.pdfbox.pdmodel.font.PDType1Font;
import static org.apache.pdfbox.pdmodel.font.Standard14Fonts.FontName.HELVETICA;
//import org.apache.pdfbox.pdmodel.PDField;
//import org.apache.pdfbox.text.PDFTextStripper;

public class LoanHistory extends javax.swing.JFrame {
```

```

/**
 * Creates new form Register
 */

//Global Variables
int max = 0;
int payperiod = -1;
String email = "", uid = "";
String selectedroweid = "-1";
//Setting up the regexfilter
TableRowSorter sorter;

private void calcFinalWage(){
    try{
        double loan = Double.parseDouble(txtLoan.getText());
        double add = Double.parseDouble(txtAddition.getText());
        double sub = Double.parseDouble(txtSubtraction.getText());

        double final_loan = loan + add - sub;

        String sfinal_loan = Double.toString(final_loan);

        txtFinalLoan.setText(sfinal_loan);
    }
    catch(Exception e){
        e.printStackTrace();
    }
}

private void payperiodfinder(){ //figure out the highest pay period
    try{
        String databaseURL = "jdbc:derby:SALARY_GENERATOR";
        Connection con = DriverManager.getConnection(databaseURL);

        String sql;
        Statement st = con.createStatement();

        sql = "SELECT * FROM loan_history";
        ResultSet rs = st.executeQuery(sql);

        //db stands for database

```

```

        int max = 0;
        while (rs.next() == true) {
            String dbeid = rs.getString(3);
            int eid = Integer.parseInt(dbeid);
            if (eid > max) {
                max = eid;
            }
        }
        this.payperiod = max;
        con.close();
    }
    catch (Exception e) {
        e.printStackTrace();
    }
}

```

```

private void initComponents2() {
    //Overwrite Java Driver Code

```

```

    //Display the email used to login, truncated if too long
    if (email.length() < 18) {
        lblEmail.setText(email);
    }
    else {
        lblEmail.setText(email.substring(0, 15) + "...");
    }

```

```

    //Initializing the table with specific settings
    //For the regexfilter later on
    TableRowSorter sorter = new
TableRowSorter<>((DefaultTableModel)tblEmployees.getModel());
    this.sorter = sorter;
    tblEmployees.setRowSorter(sorter);
    //Disables table sorting
    sorter.setSortable(0, false);
    sorter.setSortable(1, false);
    sorter.setSortable(2, false);
    sorter.setSortable(3, false);
    //Disables rearranging table columns
    tblEmployees.getTableHeader().setReorderingAllowed(false);

```

```

    //Textboxes set to a clear background

```

```

/*txtFullName.setBackground(new Color(0, 0, 0, 0));
txtEmail.setBackground(new Color(0, 0, 0, 0));
txtPhoneNumber.setBackground(new Color(0, 0, 0, 0));*/

//Integer and Numeric text fields
((AbstractDocument) txtEID.getDocument()).setDocumentFilter(new
IntegerDocumentFilter());
((AbstractDocument) txtPayPeriod.getDocument()).setDocumentFilter(new
IntegerDocumentFilter());

((AbstractDocument) txtLoan.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
((AbstractDocument) txtAddition.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
((AbstractDocument) txtSubtraction.getDocument()).setDocumentFilter(new
NumericDocumentFilter());

String maxeid = Integer.toString(this.max);
txtEID.setText(maxeid);

//    cmbSSSBBracket.addItem();

}

public static boolean doesTableExists(String tableName, Connection conn)
throws SQLException
{
    DatabaseMetaData meta = conn.getMetaData();
    ResultSet result = meta.getTables(null, null, tableName.toUpperCase(), null);

    return result.next();
}

private void maxfinder(){ //figure out the highest eid
try{
    String databaseURL = "jdbc:derby:SALARY_GENERATOR";
    Connection con = DriverManager.getConnection(databaseURL);

    String sql;
    Statement st = con.createStatement();

    sql = "SELECT * FROM loan_history";
    ResultSet rs = st.executeQuery(sql);

```

```

//db stands for database

int max = 0;
while (rs.next()==true){
    String dbeid = rs.getString(1);
    int eid = Integer.parseInt(dbeid);
    if (eid>max){
        max = eid;
    }
}
this.max = max;
con.close();
}
catch (Exception e){

}

}

private void search(){
    try{
        int in = cmbIn.getSelectedIndex();

        String search = txtSearch.getText().trim();

        //Search using RowFilter and regex (regular expression) to find substrings
        RowFilter<DefaultTableModel, Object> rf = null;
        try {
            if (in == 0){
                rf = RowFilter.regexFilter("(?i)" + search); //("(?i)" for case insensitive filter
            }
            else{
                rf = RowFilter.regexFilter("(?i)" + search, in - 1);
            }
        }
        catch (java.util.regex.PatternSyntaxException e) {
//            System.out.println("Error Found");
            return;
        }
        sorter.setRowFilter(rf);
    }
    catch (Exception e){
        e.printStackTrace();
    }
}
}

```

```

private void refreshtable(String sql){//reset table contents, works for SELECT queries only
    try{
        DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
        String databaseURL = "jdbc:derby:SALARY_GENERATOR";
        Connection con = DriverManager.getConnection(databaseURL);

        Statement st = con.createStatement();
        ResultSet rs = st.executeQuery(sql);

        model.setRowCount(0); //reset the table
        while (rs.next()==true){

            model.addRow(new
Object[]{rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5),rs.getString(
6),rs.getString(7),rs.getString(8)});
        }
        con.close();

    }
    catch (Exception e){

    }

}

public LoanHistory() {
    initComponents();
    try{

        maxfinder();

        refreshtable("select * from loan_history");

        initComponents2();

    }
    catch (Exception e){
        e.printStackTrace();
    }
}

```

```

    }
}

public LoanHistory(String uid, String email){ //CATCH ALL INCOMING DATA
    initComponents();
    try{
        this.uid = uid;
        this.email = email;

        maxfinder();

        refreshtable("select * from loan_history");

        initComponents2();

    }
    catch (Exception e){

    }
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    pnlBackground = new com.k33ptoo.components.KGradientPanel();
    pnlSearch = new javax.swing.JPanel();
    txtSearch = new javax.swing.JTextField();
    lblSearch = new javax.swing.JLabel();
    cmbIn = new javax.swing.JComboBox<>();
    txtIn = new javax.swing.JLabel();
    txtSort = new javax.swing.JLabel();
    cmbColumn = new javax.swing.JComboBox<>();
    cmbOrder = new javax.swing.JComboBox<>();
    kbtnSort = new com.k33ptoo.components.KButton();
    kbtnReset = new com.k33ptoo.components.KButton();
    pnlTable = new javax.swing.JPanel();
    jScrollPane1 = new javax.swing.JScrollPane();

```



```

tblEmployees = new javax.swing.JTable();
pnlEmail = new javax.swing.JPanel();
lblEmail = new javax.swing.JLabel();
pnlDelete = new javax.swing.JPanel();
kbtnDelete = new com.k33ptoo.components.KButton();
pnlDelete1 = new javax.swing.JPanel();
txtEID = new javax.swing.JTextField();
lblEID = new javax.swing.JLabel();
txtPayPeriod = new javax.swing.JTextField();
lblPayPeriod = new javax.swing.JLabel();
lblLoan = new javax.swing.JLabel();
txtLoan = new javax.swing.JTextField();
lblAddition = new javax.swing.JLabel();
txtAddition = new javax.swing.JTextField();
lblSubtraction = new javax.swing.JLabel();
txtSubtraction = new javax.swing.JTextField();
kbtnClearFields = new com.k33ptoo.components.KButton();
kbtnDeselect = new com.k33ptoo.components.KButton();
kbtnAdd = new com.k33ptoo.components.KButton();
lblBirthdate = new javax.swing.JLabel();
jCalendarDateSaved = new de.wannawork.jcalendar.JCalendarComboBox();
lblFinalLoan = new javax.swing.JLabel();
txtFinalLoan = new javax.swing.JTextField();
pnlEmail1 = new javax.swing.JPanel();
kbtnSalaryGenerator = new com.k33ptoo.components.KButton();
pnlSearch2 = new javax.swing.JPanel();
txtSort2 = new javax.swing.JLabel();
kbtnPrint = new com.k33ptoo.components.KButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setBackground(new java.awt.Color(255, 255, 255));
setResizable(false);

pnlBackground.setkEndColor(new java.awt.Color(66, 133, 244));
pnlBackground.setkGradientFocus(900);
pnlBackground.setkStartColor(new java.awt.Color(66, 133, 244));

pnlSearch.setBackground(new java.awt.Color(238, 238, 238));
pnlSearch.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
pnlSearch.setForeground(new java.awt.Color(36, 48, 94));

txtSearch.setBackground(new java.awt.Color(252, 250, 255));
txtSearch.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N

```

```

txtSearch.setForeground(new java.awt.Color(8, 8, 8));
txtSearch.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(38, 117, 247)));
txtSearch.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtSearch.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtSearchActionPerformed(evt);
    }
});
txtSearch.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtSearchKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtSearchKeyTyped(evt);
    }
});

lblSearch.setIcon(new javax.swing.ImageIcon(getClass().getResource("/29x29
Search.png"))); // NOI18N
lblSearch.setText("jLabel2");

cmbIn.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "All", "LHID",
"EID", "Pay Period", "Date", "Loan", "Add", "Sub", "Final Loan" }));
cmbIn.addItemListener(new java.awt.event.ItemListener() {
    public void itemStateChanged(java.awt.event.ItemEvent evt) {
        cmbInItemStateChanged(evt);
    }
});
cmbIn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cmbInActionPerformed(evt);
    }
});
cmbIn.addPropertyChangeListener(new java.beans.PropertyChangeListener() {
    public void propertyChange(java.beans.PropertyChangeEvent evt) {
        cmbInPropertyChange(evt);
    }
});

txtIn.setFont(new java.awt.Font("Segoe UI Light", 0, 16)); // NOI18N
txtIn.setForeground(new java.awt.Color(8, 8, 8));
txtIn.setText("In:");

```

```

txtSort.setFont(new java.awt.Font("Segoe UI Light", 0, 16)); // NOI18N
txtSort.setForeground(new java.awt.Color(8, 8, 8));
txtSort.setText("Sort By:");

cmbColumn.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "LHID",
"EID", "Pay Period", "Date", "Loan", "Add", "Sub", "Final Loan" }));
cmbColumn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cmbColumnActionPerformed(evt);
    }
});

cmbOrder.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {
"Ascending", "Descending" }));

kbtnSort.setText("Sort");
kbtnSort.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnSort.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnSort.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnSort.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnSort.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnSort.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnSort.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnSortActionPerformed(evt);
    }
});

kbtnReset.setText("Reset");
kbtnReset.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnReset.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnReset.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnReset.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnReset.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnReset.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnResetActionPerformed(evt);
    }
});

javax.swing.GroupLayout pnlSearchLayout = new javax.swing.GroupLayout(pnlSearch);
pnlSearch.setLayout(pnlSearchLayout);
pnlSearchLayout.setHorizontalGroup(
    pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addGroup(pnlSearchLayout.createSequentialGroup())
        .addContainerGap()
        .addComponent(lblSearch, javax.swing.GroupLayout.PREFERRED_SIZE, 29,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(txtSearch, javax.swing.GroupLayout.PREFERRED_SIZE, 156,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(txtIn)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(cmbIn, javax.swing.GroupLayout.PREFERRED_SIZE, 74,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(txtSort)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(cmbColumn, 0, 154, Short.MAX_VALUE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(cmbOrder, javax.swing.GroupLayout.PREFERRED_SIZE, 84,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(kbtnSort, javax.swing.GroupLayout.PREFERRED_SIZE, 55,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(kbtnReset, javax.swing.GroupLayout.PREFERRED_SIZE, 58,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap())
    );
    pnlSearchLayout.setVerticalGroup(
        pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlSearchLayout.createSequentialGroup())
        .addGap(5, 5, 5)

        .addGroup(pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

        .addGroup(pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
            .addComponent(kbtnSort, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, 0, Short.MAX_VALUE)

        .addGroup(pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(cmbOrder, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addComponent(cmbColumn, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(kbtnReset, javax.swing.GroupLayout.PREFERRED_SIZE,
27, javax.swing.GroupLayout.PREFERRED_SIZE)))

.addGroup(pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(lblSearch)

.addGroup(pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(txtSearch, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(cmbIn, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(txtIn)
        .addComponent(txtSort))))
.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);

pnlTable.setBackground(new java.awt.Color(55, 71, 133));
pnlTable.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
pnlTable.setForeground(new java.awt.Color(36, 48, 94));

tblEmployees.setAutoCreateRowSorter(true);
tblEmployees.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {

        },
    new String [] {
        "LHID", "EID", "Pay Period", "Date", "Loan", "Addition", "Subtraction", "Final Loan"
    }
) {
    boolean[] canEdit = new boolean [] {
        false, false, false, false, false, false, false, false
    };

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});
tblEmployees.setToolTipText("");
tblEmployees.setAutoResizeMode(javax.swing.JTable.AUTO_RESIZE_LAST_COLUMN);

```

```

tblEmployees.getTableHeader().setReorderingAllowed(false);
tblEmployees.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        tblEmployeesMouseClicked(evt);
    }
});
jScrollPane1.setViewportView(tblEmployees);
if (tblEmployees.getColumnModel().getColumnCount() > 0) {
    tblEmployees.getColumnModel().getColumn(0).setResizable(false);
    tblEmployees.getColumnModel().getColumn(1).setResizable(false);
    tblEmployees.getColumnModel().getColumn(2).setResizable(false);
    tblEmployees.getColumnModel().getColumn(3).setResizable(false);
    tblEmployees.getColumnModel().getColumn(4).setResizable(false);
    tblEmployees.getColumnModel().getColumn(5).setResizable(false);
    tblEmployees.getColumnModel().getColumn(6).setResizable(false);
}

javax.swing.GroupLayout pnlTableLayout = new javax.swing.GroupLayout(pnlTable);
pnlTable.setLayout(pnlTableLayout);
pnlTableLayout.setHorizontalGroup(
    pnlTableLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlTableLayout.createSequentialGroup()
            .add(pnlTableLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .add(pnlTableLayout.createSequentialGroup()
                    .addContainerGap()
                    .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 814,
Short.MAX_VALUE)
                    .addContainerGap())
                .add(pnlTableLayout.createSequentialGroup()
                    .addContainerGap()
                    .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 583,
Short.MAX_VALUE)
                    .addContainerGap())
            )
        );

pnlEmail.setBackground(new java.awt.Color(238, 238, 238));
pnlEmail.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
pnlEmail.setForeground(new java.awt.Color(36, 48, 94));

lblEmail.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblEmail.setForeground(new java.awt.Color(8, 8, 8));

```

```

lblEmail.setText("Loan History");

javax.swing.GroupLayout pnlEmailLayout = new javax.swing.GroupLayout(pnlEmail);
pnlEmail.setLayout(pnlEmailLayout);
pnlEmailLayout.setHorizontalGroup(
    pnlEmailLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlEmailLayout.createSequentialGroup()
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(lblEmail, javax.swing.GroupLayout.PREFERRED_SIZE, 83,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap())
    );
pnlEmailLayout.setVerticalGroup(
    pnlEmailLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlEmailLayout.createSequentialGroup()
        .addContainerGap()
        .addComponent(lblEmail, javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );

pnlDelete.setBackground(new java.awt.Color(238, 238, 238));
pnlDelete.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
pnlDelete.setForeground(new java.awt.Color(36, 48, 94));

kbtnDelete.setText("Delete");
kbtnDelete.setEndColor(new java.awt.Color(28, 84, 140));
kbtnDelete.setHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnDelete.setHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnDelete.setHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnDelete.setSelectedColor(new java.awt.Color(252, 250, 255));
kbtnDelete.setStartColor(new java.awt.Color(28, 84, 140));
kbtnDelete.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnDeleteActionPerformed(evt);
    }
});

javax.swing.GroupLayout pnlDeleteLayout = new javax.swing.GroupLayout(pnlDelete);
pnlDelete.setLayout(pnlDeleteLayout);
pnlDeleteLayout.setHorizontalGroup(
    pnlDeleteLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlDeleteLayout.createSequentialGroup())
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(kbtnDelete, javax.swing.GroupLayout.PREFERRED_SIZE, 150,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(83, 83, 83))
    );
    pnlDeleteLayout.setVerticalGroup(
        pnlDeleteLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlDeleteLayout.createSequentialGroup())
        .addContainerGap()
        .addComponent(kbtnDelete, javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(10, Short.MAX_VALUE))
    );

    pnlDelete1.setBackground(new java.awt.Color(238, 238, 238));
    pnlDelete1.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
    pnlDelete1.setForeground(new java.awt.Color(36, 48, 94));

    txtEID.setBackground(new java.awt.Color(252, 250, 255));
    txtEID.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
    txtEID.setForeground(new java.awt.Color(8, 8, 8));
    txtEID.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
    txtEID.setMargin(new java.awt.Insets(2, 2, 2, 2));
    txtEID.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            txtEIDActionPerformed(evt);
        }
    });
    txtEID.addKeyListener(new java.awt.event.KeyAdapter() {
        public void keyReleased(java.awt.event.KeyEvent evt) {
            txtEIDKeyReleased(evt);
        }
        public void keyTyped(java.awt.event.KeyEvent evt) {
            txtEIDKeyTyped(evt);
        }
    });

    lblEID.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
    lblEID.setForeground(new java.awt.Color(8, 8, 8));
    lblEID.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);

```



```

lblEID.setText("EID:");

txtPayPeriod.setBackground(new java.awt.Color(252, 250, 255));
txtPayPeriod.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtPayPeriod.setForeground(new java.awt.Color(8, 8, 8));
txtPayPeriod.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtPayPeriod.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtPayPeriod.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtPayPeriodActionPerformed(evt);
    }
});
txtPayPeriod.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtPayPeriodKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtPayPeriodKeyTyped(evt);
    }
});

lblPayPeriod.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblPayPeriod.setForeground(new java.awt.Color(8, 8, 8));
lblPayPeriod.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblPayPeriod.setText("Pay Prd.");

lblLoan.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblLoan.setForeground(new java.awt.Color(8, 8, 8));
lblLoan.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblLoan.setText("Loan:");

txtLoan.setBackground(new java.awt.Color(252, 250, 255));
txtLoan.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtLoan.setForeground(new java.awt.Color(8, 8, 8));
txtLoan.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtLoan.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtLoan.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtLoanActionPerformed(evt);
    }
});
txtLoan.addKeyListener(new java.awt.event.KeyAdapter() {

```

```

        public void keyReleased(java.awt.event.KeyEvent evt) {
            txtLoanKeyReleased(evt);
        }
        public void keyTyped(java.awt.event.KeyEvent evt) {
            txtLoanKeyTyped(evt);
        }
    });

    lblAddition.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
    lblAddition.setForeground(new java.awt.Color(8, 8, 8));
    lblAddition.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
    lblAddition.setText("Addition:");

    txtAddition.setBackground(new java.awt.Color(252, 250, 255));
    txtAddition.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
    txtAddition.setForeground(new java.awt.Color(8, 8, 8));
    txtAddition.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
    txtAddition.setMargin(new java.awt.Insets(2, 2, 2, 2));
    txtAddition.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            txtAdditionActionPerformed(evt);
        }
    });
    txtAddition.addKeyListener(new java.awt.event.KeyAdapter() {
        public void keyReleased(java.awt.event.KeyEvent evt) {
            txtAdditionKeyReleased(evt);
        }
        public void keyTyped(java.awt.event.KeyEvent evt) {
            txtAdditionKeyTyped(evt);
        }
    });

    lblSubtraction.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
    lblSubtraction.setForeground(new java.awt.Color(8, 8, 8));
    lblSubtraction.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
    lblSubtraction.setText("Subtraction:");

    txtSubtraction.setBackground(new java.awt.Color(252, 250, 255));
    txtSubtraction.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
    txtSubtraction.setForeground(new java.awt.Color(8, 8, 8));
    txtSubtraction.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
    txtSubtraction.setMargin(new java.awt.Insets(2, 2, 2, 2));

```

```

txtSubtraction.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtSubtractionActionPerformed(evt);
    }
});
txtSubtraction.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtSubtractionKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtSubtractionKeyTyped(evt);
    }
});

kbtnClearFields.setText("Clear Fields");
kbtnClearFields.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnClearFields.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnClearFields.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnClearFields.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnClearFields.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnClearFields.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnClearFields.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnClearFieldsActionPerformed(evt);
    }
});

kbtnDeselect.setText("Deselect");
kbtnDeselect.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnDeselect.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnDeselect.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnDeselect.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnDeselect.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnDeselect.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnDeselect.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnDeselectActionPerformed(evt);
    }
});

kbtnAdd.setText("Add");
kbtnAdd.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnAdd.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnAdd.setkHoverForeGround(new java.awt.Color(252, 250, 255));

```

```

kbtnAdd.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnAdd.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnAdd.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnAdd.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnAddActionPerformed(evt);
    }
});

lblBirthdate.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblBirthdate.setForeground(new java.awt.Color(8, 8, 8));
lblBirthdate.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblBirthdate.setText("Date:");

jCalendarDateSaved.setFocusable(false);
jCalendarDateSaved.addChangeListener(new javax.swing.event.ChangeListener() {
    public void stateChanged(javax.swing.event.ChangeEvent evt) {
        jCalendarDateSavedStateChanged(evt);
    }
});

lblFinalLoan.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblFinalLoan.setForeground(new java.awt.Color(8, 8, 8));
lblFinalLoan.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblFinalLoan.setText("Final Loan:");

txtFinalLoan.setBackground(new java.awt.Color(252, 250, 255));
txtFinalLoan.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtFinalLoan.setForeground(new java.awt.Color(8, 8, 8));
txtFinalLoan.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtFinalLoan.setFocusable(false);
txtFinalLoan.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtFinalLoan.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtFinalLoanActionPerformed(evt);
    }
});
txtFinalLoan.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtFinalLoanKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtFinalLoanKeyTyped(evt);
    }
});

```

```

    }
    });

    javax.swing.GroupLayout pnlDelete1Layout = new javax.swing.GroupLayout(pnlDelete1);
    pnlDelete1.setLayout(pnlDelete1Layout);
    pnlDelete1Layout.setHorizontalGroup(
        pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(pnlDelete1Layout.createSequentialGroup()
                .addGap(21, 21, 21)
                .addComponent(kbtnClearFields, javax.swing.GroupLayout.PREFERRED_SIZE, 87,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addComponent(kbtnDeselect, javax.swing.GroupLayout.PREFERRED_SIZE, 94,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 14,
Short.MAX_VALUE)
                .addComponent(kbtnAdd, javax.swing.GroupLayout.PREFERRED_SIZE, 84,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(16, 16, 16))
            .addGroup(pnlDelete1Layout.createSequentialGroup()

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    G)
        .addGroup(pnlDelete1Layout.createSequentialGroup()
            .addGap(18, 18, 18)
            .addComponent(txtEID, javax.swing.GroupLayout.PREFERRED_SIZE, 93,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(lblPayPeriod)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(txtPayPeriod, javax.swing.GroupLayout.PREFERRED_SIZE,
103, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(pnlDelete1Layout.createSequentialGroup()
            .addGap(18, 18, 18)
            .addComponent(lblBirthdate)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(jCalendarDateSaved,
javax.swing.GroupLayout.PREFERRED_SIZE, 123,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(pnlDelete1Layout.createSequentialGroup()

```

```

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
        .addComponent(lblSubtraction,
javax.swing.GroupLayout.PREFERRED_SIZE, 76,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(lblAddition))
        .addGroup(pnlDelete1Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(lblFinalLoan,
javax.swing.GroupLayout.PREFERRED_SIZE, 76,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(txtFinalLoan, javax.swing.GroupLayout.PREFERRED_SIZE,
181, javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
        .addComponent(txtAddition)
        .addComponent(txtSubtraction,
javax.swing.GroupLayout.PREFERRED_SIZE, 193,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGroup(pnlDelete1Layout.createSequentialGroup()
        .addGap(43, 43, 43)
        .addComponent(lblLoan)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(txtLoan, javax.swing.GroupLayout.PREFERRED_SIZE, 199,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);
pnlDelete1Layout.setVerticalGroup(
pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlDelete1Layout.createSequentialGroup()
        .addContainerGap()

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(txtPayPeriod, javax.swing.GroupLayout.PREFERRED_SIZE,
20, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(lblPayPeriod))

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblEID)
    .addComponent(txtEID, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    .addComponent(lblBirthdate)
    .addComponent(jCalendarDateSaved,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(18, 18, 18)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblLoan)
    .addComponent(txtLoan, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblAddition)
    .addComponent(txtAddition, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblSubtraction)
    .addComponent(txtSubtraction, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

```

```

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblFinalLoan)
    .addComponent(txtFinalLoan, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(kbtnDeselect, javax.swing.GroupLayout.PREFERRED_SIZE, 70,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(kbtnAdd, javax.swing.GroupLayout.PREFERRED_SIZE, 66,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(kbtnClearFields, javax.swing.GroupLayout.PREFERRED_SIZE,
70, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(8, 8, 8))
);

pnlEmail1.setBackground(new java.awt.Color(238, 238, 238));
pnlEmail1.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
pnlEmail1.setForeground(new java.awt.Color(36, 48, 94));

kbtnSalaryGenerator.setText("Salary Generator");
kbtnSalaryGenerator.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnSalaryGenerator.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnSalaryGenerator.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnSalaryGenerator.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnSalaryGenerator.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnSalaryGenerator.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnSalaryGenerator.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnSalaryGeneratorActionPerformed(evt);
    }
});

javax.swing.GroupLayout pnlEmail1Layout = new javax.swing.GroupLayout(pnlEmail1);
pnlEmail1.setLayout(pnlEmail1Layout);
pnlEmail1Layout.setHorizontalGroup(
    pnlEmail1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlEmail1Layout.createSequentialGroup()
            .addContainerGap()

```



```

        .addComponent(kbtnSalaryGenerator, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addContainerGap()
    );
    pnlEmail1Layout.setVerticalGroup(
        pnlEmail1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlEmail1Layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(kbtnSalaryGenerator, javax.swing.GroupLayout.PREFERRED_SIZE,
32, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );

    pnlSearch2.setBackground(new java.awt.Color(238, 238, 238));
    pnlSearch2.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
    pnlSearch2.setForeground(new java.awt.Color(36, 48, 94));

    txtSort2.setFont(new java.awt.Font("Segoe UI Light", 0, 16)); // NOI18N
    txtSort2.setForeground(new java.awt.Color(8, 8, 8));

    kbtnPrint.setText("Print");
    kbtnPrint.setkEndColor(new java.awt.Color(28, 84, 140));
    kbtnPrint.setkHoverEndColor(new java.awt.Color(76, 111, 167));
    kbtnPrint.setkHoverForeGround(new java.awt.Color(252, 250, 255));
    kbtnPrint.setkHoverStartColor(new java.awt.Color(76, 111, 167));
    kbtnPrint.setkSelectedColor(new java.awt.Color(252, 250, 255));
    kbtnPrint.setkStartColor(new java.awt.Color(28, 84, 140));
    kbtnPrint.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            kbtnPrintActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout pnlSearch2Layout = new javax.swing.GroupLayout(pnlSearch2);
    pnlSearch2.setLayout(pnlSearch2Layout);
    pnlSearch2Layout.setHorizontalGroup(
        pnlSearch2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlSearch2Layout.createSequentialGroup()
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(kbtnPrint, javax.swing.GroupLayout.PREFERRED_SIZE, 55,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(233, 233, 233)
            .addComponent(txtSort2))
    );

```

```

);
pnlSearch2Layout.setVerticalGroup(
    pnlSearch2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlSearch2Layout.createSequentialGroup())
            .addGap(5, 5, 5)

.addGroup(pnlSearch2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(ktbtnPrint, javax.swing.GroupLayout.PREFERRED_SIZE, 29,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(txtSort2))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);

javax.swing.GroupLayout pnlBackgroundLayout = new
javax.swing.GroupLayout(pnlBackground);
pnlBackground.setLayout(pnlBackgroundLayout);
pnlBackgroundLayout.setHorizontalGroup(

pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(pnlBackgroundLayout.createSequentialGroup())
        .addContainerGap()

.addGroup(pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(pnlBackgroundLayout.createSequentialGroup())
        .addComponent(pnlSearch, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(pnlSearch2, javax.swing.GroupLayout.PREFERRED_SIZE, 76,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(pnlEmail1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(pnlEmail, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(pnlBackgroundLayout.createSequentialGroup())
            .addComponent(pnlTable, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(pnlBackgroundLayout.createSequentialGroup())
        .addComponent(pnlSearch, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(pnlSearch2, javax.swing.GroupLayout.PREFERRED_SIZE, 76,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(pnlEmail1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(pnlEmail, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(pnlBackgroundLayout.createSequentialGroup())
            .addComponent(pnlTable, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

```

```

        .addComponent(pnlDelete1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(pnlDelete, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
        .addContainerGap()
    );
    pnlBackgroundLayout.setVerticalGroup(

pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlBackgroundLayout.createSequentialGroup()
            .addContainerGap()

.addGroup(pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addGroup(pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(pnlEmail, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGroup(pnlBackgroundLayout.createSequentialGroup()
            .addGap(2, 2, 2)
            .addComponent(pnlSearch, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addComponent(pnlSearch2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(pnlEmail1, javax.swing.GroupLayout.PREFERRED_SIZE, 52,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlBackgroundLayout.createSequentialGroup()
            .addComponent(pnlDelete1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(pnlDelete, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addComponent(pnlTable, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addContainerGap()
    );

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

```

```

        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
                    layout.createSequentialGroup()
                        .addGap(0, 2, Short.MAX_VALUE)
                        .addComponent(pnlBackground, javax.swing.GroupLayout.PREFERRED_SIZE,
                            javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(pnlBackground, javax.swing.GroupLayout.Alignment.TRAILING,
                    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
                    Short.MAX_VALUE)
                );

        pack();
        setLocationRelativeTo(null);
    }// </editor-fold>

```

```

private void kbtnPrintActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Date d = new Date();
        String sd = d.toString();
        MessageFormat header = new MessageFormat("Loan History");
        MessageFormat footer = new MessageFormat(sd);
        PrintRequestAttributeSet set = new HashPrintRequestAttributeSet();
        set.add(OrientationRequested.PORTRAIT);
        tblEmployees.print(JTable.PrintMode.FIT_WIDTH, header, footer, true, set, true);
        JOptionPane.showMessageDialog(null, "\n Printed Successfully");
    }
    catch (java.awt.print.PrinterException e) {
        JOptionPane.showMessageDialog(null, "\n Printing Failed");
    }
}

```

```

private void kbtnSalaryGeneratorActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        new SalaryGenerator(this.uid,this.email).setVisible(true);
        this.setVisible(false);
    }
    catch (Exception e){

    }
}

```

```

    }

    private void kbtnDeselectActionPerformed(java.awt.event.ActionEvent evt) {
        try{
            //Deselects all rows
            //      DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
            //      tblEmployees.removeRowSelectionInterval(0, model.getRowCount()-1);
            tblEmployees.clearSelection();
            //resets selectedroweid
            this.selectedroweid = "-1";
        }
        catch (Exception e){

        }
    }

    private void kbtnClearFieldsActionPerformed(java.awt.event.ActionEvent evt) {
        try{
            txtEID.setText("");
            txtPayPeriod.setText("");
            jCalendarDateSaved.setDate(new Date());
            txtAddition.setText("");
            txtLoan.setText("");
            txtSubtraction.setText("");
            txtFinalLoan.setText("");
        }
        catch(Exception e){

        }
    }

    private void txtSubtractionKeyTyped(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
    }

    private void txtSubtractionKeyReleased(java.awt.event.KeyEvent evt) {
        calcFinalWage();
    }

    private void txtSubtractionActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void txtAdditionKeyTyped(java.awt.event.KeyEvent evt) {

```

```

    // TODO add your handling code here:
}

private void txtAdditionKeyReleased(java.awt.event.KeyEvent evt) {
    calcFinalWage();
}

private void txtAdditionActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtLoanKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtLoanKeyReleased(java.awt.event.KeyEvent evt) {
    calcFinalWage();
}

private void txtLoanActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtPayPeriodKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtPayPeriodKeyReleased(java.awt.event.KeyEvent evt) {

}

private void txtPayPeriodActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtEIDKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtEIDKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtEIDActionPerformed(java.awt.event.ActionEvent evt) {

```

```

    // TODO add your handling code here:
}

private void tblEmployeesMouseClicked(java.awt.event.MouseEvent evt) {
    try{
        //Prerequisite for the update function
        DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
        int selectedrow=tblEmployees.getSelectedRow();
        if (tblEmployees.getRowSorter()!=null) {
            //Gets the correct row for a filtered jtable (RowFilter)
            selectedrow = tblEmployees.getRowSorter().convertRowIndexToModel(selectedrow);
        }
        String eid = "";
        String payperiod = "";
        String strd = "";
        String loan = "";
        String add = "";
        String sub = "";
        String final_loan = "";

        eid = model.getValueAt(selectedrow,1).toString();
        payperiod = model.getValueAt(selectedrow,2).toString();
        strd = model.getValueAt(selectedrow,3).toString();
        loan = model.getValueAt(selectedrow,4).toString();
        add = model.getValueAt(selectedrow,5).toString();
        sub = model.getValueAt(selectedrow,6).toString();
        final_loan = model.getValueAt(selectedrow,7).toString();

        this.selectedroweid = eid;
        txtEID.setText(eid);

        SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd");
        Date d = formatter.parse(strd);
        jCalendarDateSaved.setDate(d);

        txtPayPeriod.setText(payperiod);
        txtLoan.setText(loan);
        txtAddition.setText(add);
        txtSubtraction.setText(sub);
        txtFinalLoan.setText(final_loan);
    }
    catch (Exception e){
        e.printStackTrace();
    }
}

```

```

    }
}

private void kbtnResetActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        //Resets the search and sort functions; restores table to original
        refreshTable("SELECT * FROM loan_history");
        cmbColumn.setSelectedIndex(0);
        cmbOrder.setSelectedIndex(0);

        txtSearch.setText("");
        cmbIn.setSelectedIndex(0);

        search();
    }
    catch (Exception e){

    }
}

private void kbtnSortActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        String search = txtSearch.getText().trim();

        int column = cmbColumn.getSelectedIndex();

        String[] columnNames =
{"Ihid", "pay_period", "eid", "date_saved", "loan", "loan_add", "loan_sub", "final_loan"};
        String strcolumn = columnNames[column];

        int sort = cmbOrder.getSelectedIndex();
        String strsort = "";
        switch (sort){
            case 0:
                strsort = "ASC";
                break;
            case 1:
                strsort = "DESC";
        }

        if (column==0 && sort==0){
            refreshTable("SELECT * FROM loan_history");
        }
        //      else if (column==0 && sort==1){

```



```

        //          //Iterate through the resultset backwards
        //          DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
        //          String databaseURL = "jdbc:derby:SALARY_GENERATOR";
        //          Connection con = DriverManager.getConnection(databaseURL);
        //          Statement st =
con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_READ_ONL
Y);
        //          String sql = "SELECT * FROM salary_allocations";
        //          ResultSet rs = st.executeQuery(sql);
        //
        //          //Moves the cursor to the end of the ResultSet object
        //          rs.afterLast();
        //          model.setRowCount(0);
        //          while(rs.previous())
        //          {
            //          model.addRow(new
Object[]{rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4)});
            //          }
        //          rs.close();
        //          st.close();
        //          con.close();
        //          }
        else{
            refreshtable("select * from loan_history order by "+strcolumn+" "+strsort);
        }

    }
    catch (Exception e){

    }
}

private void cmbColumnActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void cmbInPropertyChange(java.beans.PropertyChangeEvent evt) {

}

private void cmbInActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

```

```

private void cmbInItemStateChanged(java.awt.event.ItemEvent evt) {
    try{
        search();
    }
    catch (Exception e){

    }
}

private void txtSearchKeyTyped(java.awt.event.KeyEvent evt) {

}

private void txtSearchKeyReleased(java.awt.event.KeyEvent evt) {
    try{
        search();
    }
    catch (Exception e){

    }
}

private void txtSearchActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void kbtnAddActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        //      String eid = "";

        String eid = "";
        String payperiod = "";
        String strd = "";
        String loan = "";
        String add = "";
        String sub = "";
        String final_loan = "";

        //      eid = txtEID.getText();
        eid = txtEID.getText();
        payperiod = txtPayPeriod.getText();

        Date d = jCalendarDateSaved.getDate();

```

```

loan = txtLoan.getText();
add = txtAddition.getText();
sub = txtSubtraction.getText();
final_loan = txtFinalLoan.getText();

System.out.println("Email: " + email);

//Error trapping
if ("".equals(loan)){
    JOptionPane.showMessageDialog(null, "Loan field is empty.");
}
else if ("".equals(add)){
    JOptionPane.showMessageDialog(null, "Loan Add field is empty.");
}
else if ("".equals(sub)){
    JOptionPane.showMessageDialog(null, "Loan Sub field is empty.");
}
else if ("".equals(final_loan)){
    JOptionPane.showMessageDialog(null, "Final Loan field is empty.");
}

else if (Double.parseDouble(loan)>999999999 ||
Double.parseDouble(loan)!=((double)Math.round(Double.parseDouble(loan)*100)/100){
    JOptionPane.showMessageDialog(null, "Loan can be a maximum of 99,999,999.99.
Decimal precision is only 2 decimal places.");
}
else if (Double.parseDouble(add)>999999999 ||
Double.parseDouble(add)!=((double)Math.round(Double.parseDouble(add)*100)/100){
    JOptionPane.showMessageDialog(null, "Loan Add can be a maximum of
99,999,999.99. Decimal precision is only 2 decimal places.");
}
else if (Double.parseDouble(sub)>999999999 ||
Double.parseDouble(sub)!=((double)Math.round(Double.parseDouble(sub)*100)/100){
    JOptionPane.showMessageDialog(null, "Loan Sub can be a maximum of
99,999,999.99. Decimal precision is only 2 decimal places.");
}
else{
    maxfinder();
    this.max++;
    String strmax = Integer.toString(this.max);

    System.out.println("strmax: " + strmax);

    double dloan = Double.parseDouble(txtLoan.getText());

```

```

double dadd = Double.parseDouble(txtAddition.getText());
double dsub = Double.parseDouble(txtSubtraction.getText());

double dfinal_loan = dloan + dadd - dsub;

String databaseURL = "jdbc:derby:SALARY_GENERATOR";
Connection con = DriverManager.getConnection(databaseURL);
Statement st = con.createStatement();

String sql = "INSERT INTO loan_history (lhid, pay_period, eid, datesaved, loan,
loan_add, loan_sub, final_loan) "
+ "VALUES (" + strmax + ", " + payperiod + ", " + eid + ", " + new
java.sql.Timestamp(d.getTime()) + ", " + loan + ", " + add + ", " + sub + ", " + dfinal_loan + ")";

st.execute(sql);
con.close();

refreshTable("select * from loan_history");

txtEID.setText("");
txtPayPeriod.setText("");

jCalendarDateSaved.setDate(new Date());

txtLoan.setText("");
txtAddition.setText("");
txtSubtraction.setText("");
txtFinalLoan.setText("");
JOptionPane.showMessageDialog(null, "Added.");
}
}
catch (Exception e){
    e.printStackTrace();
}
}

private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
        int rowcount = model.getRowCount();

        //Allows it to work with the filtered jTable (RowFilter)
        int rows[] = tblEmployees.getSelectedRows();
        for (int i = 0; i < rows.length; i++){

```

```

        rows[i] = tblEmployees.getRowSorter().convertRowIndexToModel(rows[i]);
    }

    if ("-1".equals(this.selectedroweid)){
        JOptionPane.showMessageDialog(null, "A row has not been selected.");
    }
    else{
        int confirm = JOptionPane.showConfirmDialog(null,"Are you sure?");

        if (confirm==0){
            if (rows.length>0){//Delete multiple or single
                int eids[] = new int[rows.length];
                for (int i = 0; i<rows.length; i++){
                    eids[i] = Integer.parseInt(model.getValueAt(rows[i],0).toString());
                }
                String databaseURL = "jdbc:derby:SALARY_GENERATOR";
                Connection con = DriverManager.getConnection(databaseURL);

                for (int i = 0; i<eids.length; i++){
                    Statement st = con.createStatement();

                    String sql = "DELETE FROM loan_history WHERE lhid=" + eids[i];
                    st.execute(sql);
                }
                con.close();

                //            txtFullName.setText("");
                //            txtEmail.setText("");
                //            txtPhoneNumber.setText("");

                this.selectedroweid = "-1";
                refreshTable("select * from loan_history");

                maxfinder();
            }
        }
    }

}
catch (Exception e){

}
}

```

```

private void jCalendarDateSavedStateChanged(javax.swing.event.ChangeEvent evt) {

}

private void txtFinalLoanActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtFinalLoanKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtFinalLoanKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

```

```

java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

```

```

java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
//</editor-fold>

```

```

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new Register().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JComboBox<String> cmbColumn;
private javax.swing.JComboBox<String> cmbIn;
private javax.swing.JComboBox<String> cmbOrder;
private de.wannawork.jcalendar.JCalendarComboBox jCalendarDateSaved;
private javax.swing.JScrollPane jScrollPane1;
private com.k33ptoo.components.KButton btnAdd;
private com.k33ptoo.components.KButton btnClearFields;
private com.k33ptoo.components.KButton btnDelete;
private com.k33ptoo.components.KButton btnDeselect;
private com.k33ptoo.components.KButton btnReset;
private com.k33ptoo.components.KButton btnSalaryGenerator;
private com.k33ptoo.components.KButton btnSort;
private com.k33ptoo.components.KButton btnPrint;
private javax.swing.JLabel lblAddition;
private javax.swing.JLabel lblBirthdate;
private javax.swing.JLabel lblEID;
private javax.swing.JLabel lblEmail;
private javax.swing.JLabel lblFinalLoan;
private javax.swing.JLabel lblLoan;
private javax.swing.JLabel lblPayPeriod;
private javax.swing.JLabel lblSearch;
private javax.swing.JLabel lblSubtraction;
private com.k33ptoo.components.KGradientPanel pnlBackground;
private javax.swing.JPanel pnlDelete;
private javax.swing.JPanel pnlDelete1;

```

```
private javax.swing.JPanel pnlEmail;  
private javax.swing.JPanel pnlEmail1;  
private javax.swing.JPanel pnlSearch;  
private javax.swing.JPanel pnlSearch2;  
private javax.swing.JPanel pnlTable;  
private javax.swing.JTable tblEmployees;  
private javax.swing.JTextField txtAddition;  
private javax.swing.JTextField txtEID;  
private javax.swing.JTextField txtFinalLoan;  
private javax.swing.JLabel txtIn;  
private javax.swing.JTextField txtLoan;  
private javax.swing.JTextField txtPayPeriod;  
private javax.swing.JTextField txtSearch;  
private javax.swing.JLabel txtSort;  
private javax.swing.JLabel txtSort2;  
private javax.swing.JTextField txtSubtraction;  
// End of variables declaration  
}
```


Manipulate SSS Allocations Page:

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
 license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */
```

```
/**
 *
 * @author User
 */
import java.awt.Color;
import java.sql.Connection;
import java.sql.DatabaseMetaData;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.swing.JOptionPane;
import javax.swing.RowFilter;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableRowSorter;

import javax.swing.text.*;
```

```
public class SSS extends javax.swing.JFrame {
```

```
    /**
     * Creates new form Register
     */
```

```
    //Global Variables
    int max = 0;
    String email = "", uid = "";
    String selectedroweid = "-1";
    //Setting up the regexfilter
    TableRowSorter sorter;
```

```

private void initComponents2(){
    //Overwrite Java Driver Code

    //Initializing the table with specific settings
    //For the regexfilter later on
    TableRowSorter sorter = new
TableRowSorter<>((DefaultTableModel)tblEmployees.getModel());
    this.sorter = sorter;
    tblEmployees.setRowSorter(sorter);
    //Disables table sorting
    sorter.setSortable(0, false);
    sorter.setSortable(1, false);
    sorter.setSortable(2, false);
    sorter.setSortable(3, false);
    //Disables rearranging table columns
    tblEmployees.getTableHeader().setReorderingAllowed(false);
    //Textboxes set to a clear background
    /*txtFullName.setBackground(new Color(0, 0, 0, 0));
    txtEmail.setBackground(new Color(0, 0, 0, 0));
    txtPhoneNumber.setBackground(new Color(0, 0, 0, 0));*/

    //Adding numeric filters to necessary textboxes
    ((AbstractDocument) txtBID.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
    ((AbstractDocument) txtBracketLowerBound.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
    ((AbstractDocument) txtBracketUpperBound.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
    ((AbstractDocument) txtER.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
    ((AbstractDocument) txtEE.getDocument()).setDocumentFilter(new
NumericDocumentFilter());

    String maxeid = Integer.toString(this.max+1);
    txtBID.setText(maxeid);

}

public static boolean doesTableExists(String tableName, Connection conn)
throws SQLException
{
    DatabaseMetaData meta = conn.getMetaData();
    ResultSet result = meta.getTables(null, null, tableName.toUpperCase(), null);

```

```

        return result.next();
    }

private void maxfinder(){ //figure out the highest eid
    try{
        String databaseURL = "jdbc:derby:SALARY_GENERATOR";
        Connection con = DriverManager.getConnection(databaseURL);

        String sql;
        Statement st = con.createStatement();

        sql = "SELECT * FROM sss";
        ResultSet rs = st.executeQuery(sql);

        //db stands for database

        int max = 0;
        while (rs.next()==true){
            String dbeid = rs.getString(1);
            int eid = Integer.parseInt(dbeid);
            if (eid>max){
                max = eid;
            }
        }
        this.max = max;
        con.close();
    }
    catch (Exception e){
        e.printStackTrace();
    }
}

private void search(){
    try{
        int in = cmbIn.getSelectedIndex();

        String search = txtSearch.getText().trim();

        //Search using RowFilter and regex (regular expression) to find substrings
        RowFilter<DefaultTableModel, Object> rf = null;
        try {
            if (in == 0){
                rf = RowFilter.regexFilter("(?i)" + search); //("(?i)" for case insensitive filter
            }
        }
    }
}

```

```

        }
        else{
            rf = RowFilter.regexFilter("(?i)" + search, in - 1);
        }
    }
    catch (java.util.regex.PatternSyntaxException e) {
//        System.out.println("Error Found");
        return;
    }
    sorter.setRowFilter(rf);
}
catch (Exception e){
    e.printStackTrace();
}
}

private void refreshtable(String sql){//reset table contents, works for SELECT queries only
    try{
        DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
        String databaseURL = "jdbc:derby:SALARY_GENERATOR";
        Connection con = DriverManager.getConnection(databaseURL);

        Statement st = con.createStatement();
        ResultSet rs = st.executeQuery(sql);

        model.setRowCount(0); //reset the table
        while (rs.next()==true){

            model.addRow(new Object[]{rs.getString(1), rs.getString(2), rs.getString(3),
rs.getString(5), rs.getString(4)});
        }
        con.close();

    }
    catch (Exception e){
        System.out.println("Refresh table exception");
        e.printStackTrace();
    }
}
}

```

```

public SSS() {
    initComponents();

    try{

        maxfinder();

        refreshtable("select * from sss");

        initComponents2();

    }
    catch (Exception e){
        e.printStackTrace();
    }
}

public SSS(String uid, String email){ //CATCH ALL INCOMING DATA
    initComponents();
    try{

        this.uid = uid;
        this.email = email;

        maxfinder();

        refreshtable("select * from sss");

        initComponents2();

    }
    catch (Exception e){
        e.printStackTrace();
    }
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.

```

```

*/
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    pnlBackground = new com.k33ptoo.components.KGradientPanel();
    pnlSearch = new javax.swing.JPanel();
    txtSearch = new javax.swing.JTextField();
    lblSearch = new javax.swing.JLabel();
    cmbIn = new javax.swing.JComboBox<>();
    txtIn = new javax.swing.JLabel();
    txtSort = new javax.swing.JLabel();
    cmbColumn = new javax.swing.JComboBox<>();
    cmbOrder = new javax.swing.JComboBox<>();
    kbtnSort = new com.k33ptoo.components.KButton();
    kbtnReset = new com.k33ptoo.components.KButton();
    pnlTable = new javax.swing.JPanel();
    jScrollPane1 = new javax.swing.JScrollPane();
    tblEmployees = new javax.swing.JTable();
    pnlEmail = new javax.swing.JPanel();
    lblEditPersonalRecords = new javax.swing.JLabel();
    pnlDelete = new javax.swing.JPanel();
    kbtnDeleteAll = new com.k33ptoo.components.KButton();
    kbtnDelete = new com.k33ptoo.components.KButton();
    pnlDelete1 = new javax.swing.JPanel();
    txtBID = new javax.swing.JTextField();
    lblBID = new javax.swing.JLabel();
    txtBracketLowerBound = new javax.swing.JTextField();
    lblBracketLowerBound = new javax.swing.JLabel();
    kbtnClearFields = new com.k33ptoo.components.KButton();
    kbtnDeselect = new com.k33ptoo.components.KButton();
    kbtnUpdate = new com.k33ptoo.components.KButton();
    kbtnAdd = new com.k33ptoo.components.KButton();
    lblBracketUpperBound = new javax.swing.JLabel();
    txtBracketUpperBound = new javax.swing.JTextField();
    lblER = new javax.swing.JLabel();
    txtER = new javax.swing.JTextField();
    lblEE = new javax.swing.JLabel();
    txtEE = new javax.swing.JTextField();
    pnlEmail1 = new javax.swing.JPanel();
    kbtnSalaryGenerator = new com.k33ptoo.components.KButton();
    pnlLogo = new javax.swing.JPanel();
    lblLogo = new javax.swing.JLabel();

```

```

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setBackground(new java.awt.Color(255, 255, 255));
setResizable(false);

pnlBackground.setkEndColor(new java.awt.Color(66, 133, 244));
pnlBackground.setkGradientFocus(900);
pnlBackground.setkStartColor(new java.awt.Color(66, 133, 244));

pnlSearch.setBackground(new java.awt.Color(238, 238, 238));
pnlSearch.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
pnlSearch.setForeground(new java.awt.Color(36, 48, 94));

txtSearch.setBackground(new java.awt.Color(252, 250, 255));
txtSearch.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtSearch.setForeground(new java.awt.Color(8, 8, 8));
txtSearch.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(38, 117, 247)));
txtSearch.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtSearch.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtSearchActionPerformed(evt);
    }
});
txtSearch.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtSearchKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtSearchKeyTyped(evt);
    }
});

lblSearch.setIcon(new javax.swing.ImageIcon(getClass().getResource("/29x29
Search.png"))); // NOI18N
lblSearch.setText("jLabel2");

cmbIn.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "All", "BID",
"Bracket Lower Bound", "Bracket Upper Bound", "ER", "EE" }));
cmbIn.addItemListener(new java.awt.event.ItemListener() {
    public void itemStateChanged(java.awt.event.ItemEvent evt) {
        cmbInItemStateChanged(evt);
    }
});

```

```

cmbIn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cmbInActionPerformed(evt);
    }
});
cmbIn.addPropertyChangeListener(new java.beans.PropertyChangeListener() {
    public void propertyChange(java.beans.PropertyChangeEvent evt) {
        cmbInPropertyChange(evt);
    }
});

txtIn.setFont(new java.awt.Font("Segoe UI Light", 0, 16)); // NOI18N
txtIn.setForeground(new java.awt.Color(8, 8, 8));
txtIn.setText("In:");

txtSort.setFont(new java.awt.Font("Segoe UI Light", 0, 16)); // NOI18N
txtSort.setForeground(new java.awt.Color(8, 8, 8));
txtSort.setText("Sort By:");

cmbColumn.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "BID",
"Bracket Lower Bound", "Bracket Upper Bound", "ER", "EE" }));
cmbColumn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cmbColumnActionPerformed(evt);
    }
});

cmbOrder.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {
"Ascending", "Descending" }));

kbtnSort.setText("Sort");
kbtnSort.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnSort.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnSort.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnSort.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnSort.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnSort.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnSort.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnSortActionPerformed(evt);
    }
});

kbtnReset.setText("Reset");

```



```

kbtnReset.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnReset.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnReset.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnReset.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnReset.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnReset.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnResetActionPerformed(evt);
    }
});

javax.swing.GroupLayout pnlSearchLayout = new javax.swing.GroupLayout(pnlSearch);
pnlSearch.setLayout(pnlSearchLayout);
pnlSearchLayout.setHorizontalGroup(
    pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlSearchLayout.createSequentialGroup()
            .addGap(18, 18, 18)
            .addComponent(lblSearch, javax.swing.GroupLayout.PREFERRED_SIZE, 29,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(txtSearch, javax.swing.GroupLayout.PREFERRED_SIZE, 156,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(txtIn)
            .addGap(18, 18, 18)
            .addComponent(cmbIn, javax.swing.GroupLayout.PREFERRED_SIZE, 80,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(txtSort)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(cmbColumn, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(cmbOrder, javax.swing.GroupLayout.PREFERRED_SIZE, 84,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(kbtnSort, javax.swing.GroupLayout.PREFERRED_SIZE, 55,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(kbtnReset, javax.swing.GroupLayout.PREFERRED_SIZE, 58,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
        )
);
pnlSearchLayout.setVerticalGroup(

```

```

        pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(pnlSearchLayout.createSequentialGroup()
                .addGap(5, 5, 5)

                .addGroup(pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

                .addGroup(pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                    .addComponent(kbtnSort, javax.swing.GroupLayout.Alignment.TRAILING,
                        javax.swing.GroupLayout.PREFERRED_SIZE, 0, Short.MAX_VALUE)

                    .addGroup(pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                        .addComponent(cmbOrder, javax.swing.GroupLayout.PREFERRED_SIZE,
                            javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(cmbColumn, javax.swing.GroupLayout.PREFERRED_SIZE,
                            javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(kbtnReset, javax.swing.GroupLayout.PREFERRED_SIZE,
                            27, javax.swing.GroupLayout.PREFERRED_SIZE)))

                    .addGroup(pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addComponent(lblSearch)

                        .addGroup(pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                            .addComponent(txtSearch, javax.swing.GroupLayout.PREFERRED_SIZE,
                                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addComponent(cmbIn, javax.swing.GroupLayout.PREFERRED_SIZE,
                                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addComponent(txtIn)
                            .addComponent(txtSort))))
                        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            );

        pnlTable.setBackground(new java.awt.Color(55, 71, 133));
        pnlTable.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
            java.awt.Color(255, 255, 255)));
        pnlTable.setForeground(new java.awt.Color(36, 48, 94));

        tblEmployees.setAutoCreateRowSorter(true);
        tblEmployees.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {

```



```

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlTableLayout.createSequentialGroup())
        .addContainerGap()
        .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 574,
Short.MAX_VALUE)
        .addContainerGap())
    );

    pnlEmail.setBackground(new java.awt.Color(238, 238, 238));
    pnlEmail.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
    pnlEmail.setForeground(new java.awt.Color(36, 48, 94));

    lblEditPersonalRecords.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
    lblEditPersonalRecords.setForeground(new java.awt.Color(8, 8, 8));
    lblEditPersonalRecords.setText("SSS Allocations");

    javax.swing.GroupLayout pnlEmailLayout = new javax.swing.GroupLayout(pnlEmail);
    pnlEmail.setLayout(pnlEmailLayout);
    pnlEmailLayout.setHorizontalGroup(
        pnlEmailLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(pnlEmailLayout.createSequentialGroup()
                .addContainerGap()
                .addComponent(lblEditPersonalRecords)
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );
    pnlEmailLayout.setVerticalGroup(
        pnlEmailLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(pnlEmailLayout.createSequentialGroup()
                .addContainerGap()
                .addComponent(lblEditPersonalRecords,
javax.swing.GroupLayout.PREFERRED_SIZE, 32,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );

    pnlDelete.setBackground(new java.awt.Color(238, 238, 238));
    pnlDelete.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
    pnlDelete.setForeground(new java.awt.Color(36, 48, 94));

    kbtnDeleteAll.setText("Delete All");
    kbtnDeleteAll.setkEndColor(new java.awt.Color(28, 84, 140));
    kbtnDeleteAll.setkHoverEndColor(new java.awt.Color(76, 111, 167));

```

```

kbtnDeleteAll.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnDeleteAll.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnDeleteAll.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnDeleteAll.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnDeleteAll.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnDeleteAllActionPerformed(evt);
    }
});

kbtnDelete.setText("Delete");
kbtnDelete.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnDelete.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnDelete.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnDelete.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnDelete.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnDelete.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnDelete.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnDeleteActionPerformed(evt);
    }
});

javax.swing.GroupLayout pnlDeleteLayout = new javax.swing.GroupLayout(pnlDelete);
pnlDelete.setLayout(pnlDeleteLayout);
pnlDeleteLayout.setHorizontalGroup(
    pnlDeleteLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlDeleteLayout.createSequentialGroup()
            .add(pnlDeleteLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .add(pnlDeleteLayout.createSequentialGroup()
                    .addContainerGap()
                    .add(kbtnDeleteAll, javax.swing.GroupLayout.PREFERRED_SIZE, 98,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .add(kbtnDelete, javax.swing.GroupLayout.PREFERRED_SIZE, 99,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addContainerGap())
                .add(kbtnDelete))
            .addContainerGap())
        .add(kbtnDeleteAll)
    );
pnlDeleteLayout.setVerticalGroup(
    pnlDeleteLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlDeleteLayout.createSequentialGroup()
            .add(pnlDeleteLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .add(pnlDeleteLayout.createSequentialGroup()
                    .addContainerGap()
                    .add(kbtnDeleteAll, javax.swing.GroupLayout.PREFERRED_SIZE, 98,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .add(kbtnDelete, javax.swing.GroupLayout.PREFERRED_SIZE, 99,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addContainerGap())
                .add(kbtnDelete))
            .addContainerGap())
        .add(kbtnDelete)
    );

```

```

        .addComponent(kbtnDelete, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(kbtnDeleteAll, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap(22, Short.MAX_VALUE))
    );

    pnlDelete1.setBackground(new java.awt.Color(238, 238, 238));
    pnlDelete1.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
    pnlDelete1.setForeground(new java.awt.Color(36, 48, 94));

    txtBID.setBackground(new java.awt.Color(252, 250, 255));
    txtBID.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
    txtBID.setForeground(new java.awt.Color(8, 8, 8));
    txtBID.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
    txtBID.setFocusable(false);
    txtBID.setMargin(new java.awt.Insets(2, 2, 2, 2));
    txtBID.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            txtBIDActionPerformed(evt);
        }
    });
    txtBID.addKeyListener(new java.awt.event.KeyAdapter() {
        public void keyReleased(java.awt.event.KeyEvent evt) {
            txtBIDKeyReleased(evt);
        }
        public void keyTyped(java.awt.event.KeyEvent evt) {
            txtBIDKeyTyped(evt);
        }
    });

    lblBID.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
    lblBID.setForeground(new java.awt.Color(8, 8, 8));
    lblBID.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
    lblBID.setText("BID:");

    txtBracketLowerBound.setBackground(new java.awt.Color(252, 250, 255));
    txtBracketLowerBound.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
    txtBracketLowerBound.setForeground(new java.awt.Color(8, 8, 8));
    txtBracketLowerBound.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2,
new java.awt.Color(204, 204, 204)));
    txtBracketLowerBound.setFocusTraversalPolicyProvider(true);

```

```

txtBracketLowerBound.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtBracketLowerBound.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtBracketLowerBoundActionPerformed(evt);
    }
});
txtBracketLowerBound.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtBracketLowerBoundKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtBracketLowerBoundKeyTyped(evt);
    }
});

lblBracketLowerBound.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblBracketLowerBound.setForeground(new java.awt.Color(8, 8, 8));
lblBracketLowerBound.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblBracketLowerBound.setText("Bracket lower bound:");

kbtnClearFields.setText("Clear Fields");
kbtnClearFields.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnClearFields.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnClearFields.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnClearFields.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnClearFields.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnClearFields.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnClearFields.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnClearFieldsActionPerformed(evt);
    }
});

kbtnDeselect.setText("Deselect");
kbtnDeselect.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnDeselect.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnDeselect.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnDeselect.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnDeselect.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnDeselect.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnDeselect.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnDeselectActionPerformed(evt);
    }
}

```

```

});

kbtnUpdate.setText("Update");
kbtnUpdate.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnUpdate.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnUpdate.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnUpdate.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnUpdate.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnUpdate.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnUpdate.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnUpdateActionPerformed(evt);
    }
});

kbtnAdd.setText("Add");
kbtnAdd.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnAdd.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnAdd.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnAdd.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnAdd.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnAdd.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnAdd.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnAddActionPerformed(evt);
    }
});

lblBracketUpperBound.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblBracketUpperBound.setForeground(new java.awt.Color(8, 8, 8));
lblBracketUpperBound.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblBracketUpperBound.setText("Bracket upper bound");

txtBracketUpperBound.setBackground(new java.awt.Color(252, 250, 255));
txtBracketUpperBound.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtBracketUpperBound.setForeground(new java.awt.Color(8, 8, 8));
txtBracketUpperBound.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2,
new java.awt.Color(204, 204, 204)));
txtBracketUpperBound.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtBracketUpperBound.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtBracketUpperBoundActionPerformed(evt);
    }
});

```



```

txtBracketUpperBound.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtBracketUpperBoundKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtBracketUpperBoundKeyTyped(evt);
    }
});

lblER.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblER.setForeground(new java.awt.Color(8, 8, 8));
lblER.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblER.setText("Employer's Share (ER)");

txtER.setBackground(new java.awt.Color(252, 250, 255));
txtER.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtER.setForeground(new java.awt.Color(8, 8, 8));
txtER.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtER.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtER.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtERActionPerformed(evt);
    }
});
txtER.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtERKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtERKeyTyped(evt);
    }
});

lblEE.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblEE.setForeground(new java.awt.Color(8, 8, 8));
lblEE.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblEE.setText("Employee's Share (EE)");

txtEE.setBackground(new java.awt.Color(252, 250, 255));
txtEE.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtEE.setForeground(new java.awt.Color(8, 8, 8));
txtEE.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));

```

```
txtEE.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtEE.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtEEActionPerformed(evt);
    }
});
txtEE.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtEEKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtEEKeyTyped(evt);
    }
});

javax.swing.GroupLayout pnlDelete1Layout = new javax.swing.GroupLayout(pnlDelete1);
pnlDelete1.setLayout(pnlDelete1Layout);
pnlDelete1Layout.setHorizontalGroup(
    pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlDelete1Layout.createSequentialGroup()
            .addContainerGap())
        .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(pnlDelete1Layout.createSequentialGroup()
                .addComponent(lblIER)
                .addPreferredGap(LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(txtER))
            .addGroup(pnlDelete1Layout.createSequentialGroup()
                .addComponent(lblBracketLowerBound)
                .addPreferredGap(LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(lblBracketUpperBound)))
        .addGroup(pnlDelete1Layout.createSequentialGroup()
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED))
        .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(txtBracketUpperBound)
            .addComponent(txtBracketLowerBound)))
        .addGroup(pnlDelete1Layout.createSequentialGroup()
            .addComponent(lblIBID)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED))
```

```

        .addComponent(txtBID, javax.swing.GroupLayout.PREFERRED_SIZE, 103,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, Short.MAX_VALUE))
        .addGroup(pnlDelete1Layout.createSequentialGroup())

        .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
        .addComponent(lblEE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
pnlDelete1Layout.createSequentialGroup()
        .addComponent(kbtnClearFields,
javax.swing.GroupLayout.PREFERRED_SIZE, 67,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(kbtnDeselect,
javax.swing.GroupLayout.PREFERRED_SIZE, 58,
javax.swing.GroupLayout.PREFERRED_SIZE)))

        .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlDelete1Layout.createSequentialGroup())

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(kbtnAdd, javax.swing.GroupLayout.PREFERRED_SIZE,
61, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 128,
Short.MAX_VALUE)
        .addComponent(kbtnUpdate,
javax.swing.GroupLayout.PREFERRED_SIZE, 53,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGroup(pnlDelete1Layout.createSequentialGroup()
        .addGap(1, 1, 1)
        .addComponent(txtEE))))
        .addContainerGap()
    );
    pnlDelete1Layout.setVerticalGroup(
        pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlDelete1Layout.createSequentialGroup()
        .addContainerGap()

```

```

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblBID)
    .addComponent(txtBID, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    .addComponent(lblBracketLowerBound)
    .addComponent(txtBracketLowerBound,
javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblBracketUpperBound)
    .addComponent(txtBracketUpperBound,
javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblER)
    .addComponent(txtER, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblEE)
    .addComponent(txtEE, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(kbtnClearFields, javax.swing.GroupLayout.PREFERRED_SIZE,
30, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(kbtnDeselect, javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addComponent(kbtnUpdate, javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(kbtnAdd, javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );

    pnlEmail1.setBackground(new java.awt.Color(238, 238, 238));
    pnlEmail1.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
    pnlEmail1.setForeground(new java.awt.Color(36, 48, 94));

    kbtnSalaryGenerator.setText("Salary Generator");
    kbtnSalaryGenerator.setkEndColor(new java.awt.Color(28, 84, 140));
    kbtnSalaryGenerator.setkHoverEndColor(new java.awt.Color(76, 111, 167));
    kbtnSalaryGenerator.setkHoverForeGround(new java.awt.Color(252, 250, 255));
    kbtnSalaryGenerator.setkHoverStartColor(new java.awt.Color(76, 111, 167));
    kbtnSalaryGenerator.setkSelectedColor(new java.awt.Color(252, 250, 255));
    kbtnSalaryGenerator.setkStartColor(new java.awt.Color(28, 84, 140));
    kbtnSalaryGenerator.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            kbtnSalaryGeneratorActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout pnlEmail1Layout = new javax.swing.GroupLayout(pnlEmail1);
    pnlEmail1.setLayout(pnlEmail1Layout);
    pnlEmail1Layout.setHorizontalGroup(
        pnlEmail1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlEmail1Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(kbtnSalaryGenerator, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addContainerGap())
    );
    pnlEmail1Layout.setVerticalGroup(
        pnlEmail1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(pnlEmail1Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(kbtnSalaryGenerator, javax.swing.GroupLayout.PREFERRED_SIZE,
36, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(9, Short.MAX_VALUE))
    );

```

```

        pnlLogo.setBackground(new java.awt.Color(36, 48, 94));
        pnlLogo.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
        pnlLogo.setForeground(new java.awt.Color(36, 48, 94));

        lblLogo.setIcon(new javax.swing.ImageIcon(getClass().getResource("/200x150
SSS.png"))); // NOI18N
        lblLogo.setText("jLabel1");

        javax.swing.GroupLayout pnlLogoLayout = new javax.swing.GroupLayout(pnlLogo);
        pnlLogo.setLayout(pnlLogoLayout);
        pnlLogoLayout.setHorizontalGroup(
            pnlLogoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(pnlLogoLayout.createSequentialGroup()
                    .addGap(92, 92, 92)
                    .addComponent(lblLogo, javax.swing.GroupLayout.PREFERRED_SIZE, 200,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(92, 92, 92))
        );
        pnlLogoLayout.setVerticalGroup(
            pnlLogoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(pnlLogoLayout.createSequentialGroup()
                    .addGap(71, 71, 71)
                    .addComponent(lblLogo)
                    .addGap(79, 79, 79))
        );

        javax.swing.GroupLayout pnlBackgroundLayout = new
javax.swing.GroupLayout(pnlBackground);
        pnlBackground.setLayout(pnlBackgroundLayout);
        pnlBackgroundLayout.setHorizontalGroup(
            pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(pnlBackgroundLayout.createSequentialGroup()
                    .addGap(92, 92, 92)
                    .addComponent(pnlDelete, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addGap(92, 92, 92)
                    .addComponent(pnlDelete1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addGap(92, 92, 92))
        );

```

```

        .addGroup(pnlBackgroundLayout.createSequentialGroup())
        .addComponent(pnlEmail, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(pnlEmail1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addComponent(pnlLogo, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addGroup(pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
        .addComponent(pnlTable, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(pnlSearch, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addContainerGap())
    );
    pnlBackgroundLayout.setVerticalGroup(

pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(pnlBackgroundLayout.createSequentialGroup())
        .addContainerGap()

    .addGroup(pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
        .addComponent(pnlEmail, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(pnlEmail1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(pnlSearch, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

    .addGroup(pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlBackgroundLayout.createSequentialGroup())
            .addComponent(pnlDelete1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(pnlDelete, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

```

```

        .addComponent(pnlLogo, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addComponent(pnlTable, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(pnlBackground, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(pnlBackground, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );

    pack();
    setLocationRelativeTo(null);
} // </editor-fold>

```

```

private void kbtnSortActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        String search = txtSearch.getText().trim();

        int column = cmbColumn.getSelectedIndex();

        String[] columnnames =
{"bid","bracketlowerbound","bracketupperbound","sss_ee","sss_er"};

        String strcolumn = columnnames[column];

        int sort = cmbOrder.getSelectedIndex();
        String strsort = "";
        switch (sort){
            case 0:
                strsort = "ASC";
                break;
            case 1:
                strsort = "DESC";
        }
    }
}

```



```

        if (column==0 && sort==0){
            refreshtable("SELECT * FROM sss");
        }
//        else if (column==0 && sort==1){
//            //Iterate through the resultset backwards
//            DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
//            String databaseURL = "jdbc:derby:SALARY_GENERATOR";
//            Connection con = DriverManager.getConnection(databaseURL);
//            Statement st =
con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_READ_ONLY);
//            String sql = "SELECT * FROM info";
//            ResultSet rs = st.executeQuery(sql);
//
//            //Moves the cursor to the end of the ResultSet object
//            rs.afterLast();
//            model.setRowCount(0);
//            while(rs.previous())
//            {
//                model.addRow(new
Object[]{rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4)});
//            }
//            rs.close();
//            st.close();
//            con.close();
//        }
        else{
            refreshtable("select * from sss order by "+strcolumn+" "+strsort);
        }

    }
    catch (Exception e){

    }
}

private void cmbColumnActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void tblEmployeesMouseClicked(java.awt.event.MouseEvent evt) {
    try{
        //Prerequisite for the update function
    }
}

```

```

DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
int selectedrow=tblEmployees.getSelectedRow();
if (tblEmployees.getRowSorter()!=null) {
    //Gets the correct row for a filtered jtable (RowFilter)
    selectedrow = tblEmployees.getRowSorter().convertRowIndexToModel(selectedrow);
}
String eid = "";
String bracketlower = "";
String bracketupper = "";
String er = "";
String ee = "";

//    eid = txtEID.getText();

eid = model.getValueAt(selectedrow,0).toString();
bracketlower = model.getValueAt(selectedrow,1).toString();
bracketupper = model.getValueAt(selectedrow,2).toString();
er = model.getValueAt(selectedrow,3).toString();
ee = model.getValueAt(selectedrow,4).toString();

this.selectedroweid = eid;
txtBID.setText(eid);
txtBracketLowerBound.setText(bracketlower);
txtBracketUpperBound.setText(bracketupper);
txtER.setText(er);
txtEE.setText(ee);
}
catch (Exception e){
    e.printStackTrace();
}
}

private void cmbInPropertyChange(java.beans.PropertyChangeEvent evt) {

}

private void cmbInActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void cmbInItemStateChanged(java.awt.event.ItemEvent evt) {
    try{
        search();
    }
}

```

```

    }
    catch (Exception e){

    }
}

private void txtSearchKeyTyped(java.awt.event.KeyEvent evt) {

}

private void txtSearchKeyReleased(java.awt.event.KeyEvent evt) {
    try{
        search();
    }
    catch (Exception e){

    }
}

private void txtSearchActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void kbtnResetActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        //Resets the search and sort functions; restores table to original
        refreshtable("SELECT * FROM sss");
        cmbColumn.setSelectedIndex(0);
        cmbOrder.setSelectedIndex(0);

        txtSearch.setText("");
        cmbIn.setSelectedIndex(0);

        search();
    }
    catch (Exception e){

    }
}

private void kbtnSalaryGeneratorActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        new SalaryGenerator(this.uid,this.email).setVisible(true);
        this.setVisible(false);
    }
}

```

```

    }
    catch (Exception e){

    }
}

```

```

private void kbtnDeleteAllActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        int confirm = JOptionPane.showConfirmDialog(null,"Are you want to delete ALL
records?");
        if (confirm==0){
            DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
            int rowcount = model.getRowCount(); //number of rows in model

            if (rowcount==0){
                JOptionPane.showMessageDialog(null, "The table is empty.");
            }
            else{
                int eids[] = new int[rowcount];
                for (int i = 0; i<rowcount; i++){
                    eids[i] = Integer.parseInt(model.getValueAt(i,0).toString());

                }
                String databaseURL = "jdbc:derby:SALARY_GENERATOR";
                Connection con = DriverManager.getConnection(databaseURL);

                String sql;
                for (int i = 0; i<eids.length; i++){
                    //Deletion based on eid
                    Statement st = con.createStatement();
                    sql = "DELETE FROM sss WHERE bid=" + eids[i];
                    st.execute(sql);
                }

                maxfinder();

                model.setRowCount(0);
            }
        }
    }
    catch(Exception e){

```

```

    }
}

private void kbtnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
        int rowcount = model.getRowCount();

        //Allows it to work with the filtered jTable (RowFilter)
        int rows[] = tblEmployees.getSelectedRows();
        for (int i = 0; i<rows.length; i++){
            rows[i] = tblEmployees.getRowSorter().convertRowIndexToModel(rows[i]);
        }

        if ("-1".equals(this.selectedroweid)){
            JOptionPane.showMessageDialog(null, "A row has not been selected.");
        }
        else{
            int confirm = JOptionPane.showConfirmDialog(null,"Are you sure?");

            if (confirm==0){
                if (rows.length>0){//Delete multiple or single
                    int eids[] = new int[rows.length];
                    for (int i = 0; i<rows.length; i++){
                        eids[i] = Integer.parseInt(model.getValueAt(rows[i],0).toString());
                    }
                    String databaseURL = "jdbc:derby:SALARY_GENERATOR";
                    Connection con = DriverManager.getConnection(databaseURL);

                    String sql;
                    for (int i = 0; i<eids.length; i++){
                        Statement st = con.createStatement();

                        sql = "DELETE FROM sss WHERE bid=" + eids[i];
                        st.execute(sql);
                    }

                    //            txtFullName.setText("");
                    //            txtEmail.setText("");
                    //            txtPhoneNumber.setText("");

                    this.selectedroweid = "-1";
                    refreshtable("select * from sss");
                }
            }
        }
    }
}

```

```

        maxfinder();
    }
}

}

catch (Exception e){

}

}

private void kbtnAddActionPerformed(java.awt.event.ActionEvent evt) {
    try{
//        String eid = "";
        String bracketlower = "";
        String bracketupper = "";
        String er = "";
        String ee = "";

//        eid = txtEID.getText();

        bracketlower = txtBracketLowerBound.getText();
        bracketupper = txtBracketUpperBound.getText();
        er = txtER.getText();
        ee = txtEE.getText();

        System.out.println("Email: " + email);

        //Error trapping
        if ("".equals(bracketlower)){
            JOptionPane.showMessageDialog(null, "Lower bound cannot be empty.");
        }
        else if ("".equals(bracketupper)){
            JOptionPane.showMessageDialog(null, "Upper bound cannot be empty.");
        }
        else if ("".equals(er)){
            JOptionPane.showMessageDialog(null, "Employer Share cannot be empty.");
        }
        else if ("".equals(ee)){
            JOptionPane.showMessageDialog(null, "Employee Share cannot be empty.");
        }
        else if (Double.parseDouble(bracketlower)>99999999 ||
Double.parseDouble(bracketlower)!=((double)Math.round(Double.parseDouble(bracketlower)*10
0)/100){

```

```

        JOptionPane.showMessageDialog(null, "Bracket Lower Bound can be a maximum of
99,999,999.99. Decimal precision is only 2 decimal places.");
    }
    else if (Double.parseDouble(bracketupper)>999999999 ||
Double.parseDouble(bracketupper)!=((double)Math.round(Double.parseDouble(bracketupper)*1
00)/100)){
        JOptionPane.showMessageDialog(null, "Bracket Upper Bound can be a maximum of
99,999,999.99. Decimal precision is only 2 decimal places.");
    }
    else if (Double.parseDouble(er)>999999999 ||
Double.parseDouble(er)!=((double)Math.round(Double.parseDouble(er)*100)/100){
        JOptionPane.showMessageDialog(null, "Employer Share (ER) can be a maximum of
99,999,999.99. Decimal precision is only 2 decimal places.");
    }
    else if (Double.parseDouble(ee)>999999999 ||
Double.parseDouble(ee)!=((double)Math.round(Double.parseDouble(ee)*100)/100){
        JOptionPane.showMessageDialog(null, "Employee Share (EE) can be a maximum of
99,999,999.99. Decimal precision is only 2 decimal places.");
    }
    else{
        maxfinder();
        this.max++;
        String strmax = Integer.toString(this.max);

        System.out.println("strmax: " + strmax);

        String databaseURL = "jdbc:derby:SALARY_GENERATOR";
        Connection con = DriverManager.getConnection(databaseURL);
        Statement st = con.createStatement();

        String sql = "INSERT INTO sss (bid, bracketlowerbound, bracketupperbound, sss_er,
sss_ee) "
            + "VALUES (" +strmax+", "+bracketlower+", "+bracketupper+", "+er+", "+ee+)";

        //pay_period determined by the max from salary and loan history

        st.execute(sql);
        con.close();

        refreshTable("select * from sss");

        txtBID.setText(Integer.toString(max));
        txtBracketLowerBound.setText("");
        txtBracketUpperBound.setText("");
    }
}

```

```

        txtER.setText("");
        txtEE.setText("");

        JOptionPane.showMessageDialog(null, "Added.");
    }
}
catch (Exception e){
    e.printStackTrace();
}
}

private void kbtnUpdateActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        //Prerequisite for the update function
        String bid = "";
        String bracketlower = "";
        String bracketupper = "";
        String er = "";
        String ee = "";

//        eid = txtEID.getText();

        bid = txtBID.getText();
        bracketlower = txtBracketLowerBound.getText();
        bracketupper = txtBracketUpperBound.getText();
        er = txtER.getText();
        ee = txtEE.getText();

        System.out.println("Email: " + email);

        //Error trapping
        if (Double.parseDouble(bracketlower)>999999999 ||
Double.parseDouble(bracketlower)!=((double)Math.round(Double.parseDouble(bracketlower)*10
0)/100){
            JOptionPane.showMessageDialog(null, "Bracket Lower Bound can be a maximum of
99,999,999.99. Decimal precision is only 2 decimal places.");
        }
        else if (Double.parseDouble(bracketupper)>999999999 ||
Double.parseDouble(bracketupper)!=((double)Math.round(Double.parseDouble(bracketupper)*1
00)/100){
            JOptionPane.showMessageDialog(null, "Bracket Upper Bound can be a maximum of
99,999,999.99. Decimal precision is only 2 decimal places.");
        }
    }
}

```



```

        else if (Double.parseDouble(er)>999999999 ||
Double.parseDouble(er)!=((double)Math.round(Double.parseDouble(er)*100)/100){
            JOptionPane.showMessageDialog(null, "Employer Share (ER) can be a maximum of
99,999,999.99. Decimal precision is only 2 decimal places.");
        }
        else if (Double.parseDouble(ee)>999999999 ||
Double.parseDouble(ee)!=((double)Math.round(Double.parseDouble(ee)*100)/100){
            JOptionPane.showMessageDialog(null, "Employee Share (EE) can be a maximum of
99,999,999.99. Decimal precision is only 2 decimal places.");
        }
        else{
            String strmax = Integer.toString(max);

            String databaseURL = "jdbc:derby:SALARY_GENERATOR";
            Connection con = DriverManager.getConnection(databaseURL);
            Statement st = con.createStatement();

            String sql = "UPDATE sss SET bid="+bid+", bracketlowerbound="+bracketlower+",
bracketupperbound="+bracketupper+", sss_ee="+ee+", sss_er="+er+" WHERE bid="+bid;
            st.execute(sql);
            con.close();

            //No row currently selected.
            selectedroweid = "-1";

            JOptionPane.showMessageDialog(null, "Updated.");
            tblEmployees.clearSelection();

            refreshtable("select * from sss");
            maxfinder();

            int max = this.max;
            max++;

            txtBID.setText(Integer.toString(max));
            txtBracketLowerBound.setText("");
            txtBracketUpperBound.setText("");
            txtER.setText("");
            txtEE.setText("");

        }
    }
    catch (Exception e){
        e.printStackTrace();
    }

```

```

    }
}

private void kbtnDeselectActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        //Deselects all rows
        //      DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
        //      tblEmployees.removeRowSelectionInterval(0, model.getRowCount()-1);
        tblEmployees.clearSelection();
        //resets selectedroweid
        this.selectedroweid = "-1";
    }
    catch (Exception e){

    }
}

private void kbtnClearFieldsActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        txtBID.setText("");
        txtBracketLowerBound.setText("");
        txtBracketUpperBound.setText("");
        txtER.setText("");
        txtEE.setText("");
    }
    catch(Exception e){

    }
}

private void txtBracketLowerBoundKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtBracketLowerBoundKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtBracketLowerBoundActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtBIDKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

```

```

}

private void txtBKIDKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtBKIDActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtBracketUpperBoundActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtBracketUpperBoundKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtBracketUpperBoundKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtERActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtERKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtERKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtEEActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtEEKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtEEKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

```

```

}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Register().setVisible(true);
        }
    });
}

```

```

}

// Variables declaration - do not modify
private javax.swing.JComboBox<String> cmbColumn;
private javax.swing.JComboBox<String> cmbIn;
private javax.swing.JComboBox<String> cmbOrder;
private javax.swing.JScrollPane jScrollPane1;
private com.k33ptoo.components.KButton btnAdd;
private com.k33ptoo.components.KButton btnClearFields;
private com.k33ptoo.components.KButton btnDelete;
private com.k33ptoo.components.KButton btnDeleteAll;
private com.k33ptoo.components.KButton btnDeselect;
private com.k33ptoo.components.KButton btnReset;
private com.k33ptoo.components.KButton btnSalaryGenerator;
private com.k33ptoo.components.KButton btnSort;
private com.k33ptoo.components.KButton btnUpdate;
private javax.swing.JLabel lblBID;
private javax.swing.JLabel lblBracketLowerBound;
private javax.swing.JLabel lblBracketUpperBound;
private javax.swing.JLabel lblEE;
private javax.swing.JLabel lblER;
private javax.swing.JLabel lblEditPersonalRecords;
private javax.swing.JLabel lblLogo;
private javax.swing.JLabel lblSearch;
private com.k33ptoo.components.KGradientPanel pnlBackground;
private javax.swing.JPanel pnlDelete;
private javax.swing.JPanel pnlDelete1;
private javax.swing.JPanel pnlEmail;
private javax.swing.JPanel pnlEmail1;
private javax.swing.JPanel pnlLogo;
private javax.swing.JPanel pnlSearch;
private javax.swing.JPanel pnlTable;
private javax.swing.JTable tblEmployees;
private javax.swing.JTextField txtBID;
private javax.swing.JTextField txtBracketLowerBound;
private javax.swing.JTextField txtBracketUpperBound;
private javax.swing.JTextField txtEE;
private javax.swing.JTextField txtER;
private javax.swing.JLabel txtIn;
private javax.swing.JTextField txtSearch;
private javax.swing.JLabel txtSort;
// End of variables declaration
}

```

Manipulate PhilHealth Allocations:

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */
```

```
/**
 *
 * @author User
 */
import java.awt.Color;
import java.sql.Connection;
import java.sql.DatabaseMetaData;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.swing.JOptionPane;
import javax.swing.RowFilter;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableRowSorter;

import javax.swing.text.*;
```

```
public class PhilHealth extends javax.swing.JFrame {
```

```
    /**
     * Creates new form Register
     */
```

```
    //Global Variables
    int max = 0;
    String email = "", uid = "";
    String selectedroweid = "-1";
    //Setting up the regexfilter
    TableRowSorter sorter;
```

```

private void initComponents2(){
    //Overwrite Java Driver Code

    //Initializing the table with specific settings
    //For the regexfilter later on
    TableRowSorter sorter = new
TableRowSorter<>((DefaultTableModel)tblEmployees.getModel());
    this.sorter = sorter;
    tblEmployees.setRowSorter(sorter);
    //Disables table sorting
    sorter.setSortable(0, false);
    sorter.setSortable(1, false);
    sorter.setSortable(2, false);
    sorter.setSortable(3, false);
    //Disables rearranging table columns
    tblEmployees.getTableHeader().setReorderingAllowed(false);
    //Textboxes set to a clear background
    /*txtFullName.setBackground(new Color(0, 0, 0, 0));
    txtEmail.setBackground(new Color(0, 0, 0, 0));
    txtPhoneNumber.setBackground(new Color(0, 0, 0, 0));*/

    //Adding numeric filters to necessary textboxes
    ((AbstractDocument) txtBID.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
    ((AbstractDocument) txtBracketLowerBound.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
    ((AbstractDocument) txtBracketUpperBound.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
    ((AbstractDocument) txtER.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
    ((AbstractDocument) txtEE.getDocument()).setDocumentFilter(new
NumericDocumentFilter());

    String maxeid = Integer.toString(this.max+1);
    txtBID.setText(maxeid);

}

public static boolean doesTableExists(String tableName, Connection conn)
throws SQLException
{
    DatabaseMetaData meta = conn.getMetaData();
    ResultSet result = meta.getTables(null, null, tableName.toUpperCase(), null);

```

```

        return result.next();
    }

private void maxfinder(){ //figure out the highest eid
    try{
        String databaseURL = "jdbc:derby:SALARY_GENERATOR";
        Connection con = DriverManager.getConnection(databaseURL);

        String sql;
        Statement st = con.createStatement();

        sql = "SELECT * FROM philhealth";
        ResultSet rs = st.executeQuery(sql);

        //db stands for database

        int max = 0;
        while (rs.next()==true){
            String dbeid = rs.getString(1);
            int eid = Integer.parseInt(dbeid);
            if (eid>max){
                max = eid;
            }
        }
        this.max = max;
        con.close();
    }
    catch (Exception e){
        e.printStackTrace();
    }
}

private void search(){
    try{
        int in = cmbIn.getSelectedIndex();

        String search = txtSearch.getText().trim();

        //Search using RowFilter and regex (regular expression) to find substrings
        RowFilter<DefaultTableModel, Object> rf = null;
        try {
            if (in == 0){
                rf = RowFilter.regexFilter("(?i)" + search); //("(?i)" for case insensitive filter
            }
        }
    }
}

```



```

        }
        else{
            rf = RowFilter.regexFilter("(?i)" + search, in - 1);
        }
    }
    catch (java.util.regex.PatternSyntaxException e) {
//        System.out.println("Error Found");
        return;
    }
    sorter.setRowFilter(rf);
}
catch (Exception e){
    e.printStackTrace();
}
}

private void refreshtable(String sql){//reset table contents, works for SELECT queries only
    try{
        DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
        String databaseURL = "jdbc:derby:SALARY_GENERATOR";
        Connection con = DriverManager.getConnection(databaseURL);

        Statement st = con.createStatement();
        ResultSet rs = st.executeQuery(sql);

        model.setRowCount(0); //reset the table
        while (rs.next()==true){

            model.addRow(new Object[]{rs.getString(1), rs.getString(2), rs.getString(3),
rs.getString(5), rs.getString(4)});
        }
        con.close();

    }
    catch (Exception e){
        System.out.println("Refresh table exception");
        e.printStackTrace();
    }
}
}

```

```

public PhilHealth() {
    initComponents();

    try{

        maxfinder();

        refreshtable("SELECT * FROM philhealth");

        initComponents2();

    }
    catch (Exception e){
        e.printStackTrace();
    }
}

public PhilHealth(String uid, String email){ //CATCH ALL INCOMING DATA
    initComponents();
    try{

        this.uid = uid;
        this.email = email;

        maxfinder();

        refreshtable("SELECT * FROM philhealth");

        initComponents2();

    }
    catch (Exception e){
        e.printStackTrace();
    }
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.

```

```

*/
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    pnlBackground = new com.k33ptoo.components.KGradientPanel();
    pnlSearch = new javax.swing.JPanel();
    txtSearch = new javax.swing.JTextField();
    lblSearch = new javax.swing.JLabel();
    cmbIn = new javax.swing.JComboBox<>();
    txtIn = new javax.swing.JLabel();
    txtSort = new javax.swing.JLabel();
    cmbColumn = new javax.swing.JComboBox<>();
    cmbOrder = new javax.swing.JComboBox<>();
    kbtnSort = new com.k33ptoo.components.KButton();
    kbtnReset = new com.k33ptoo.components.KButton();
    pnlTable = new javax.swing.JPanel();
    jScrollPane1 = new javax.swing.JScrollPane();
    tblEmployees = new javax.swing.JTable();
    pnlEmail = new javax.swing.JPanel();
    lblEditPersonalRecords = new javax.swing.JLabel();
    pnlDelete = new javax.swing.JPanel();
    kbtnDeleteAll = new com.k33ptoo.components.KButton();
    kbtnDelete = new com.k33ptoo.components.KButton();
    pnlDelete1 = new javax.swing.JPanel();
    txtBID = new javax.swing.JTextField();
    lblBID = new javax.swing.JLabel();
    txtBracketLowerBound = new javax.swing.JTextField();
    lblBracketLowerBound = new javax.swing.JLabel();
    kbtnClearFields = new com.k33ptoo.components.KButton();
    kbtnDeselect = new com.k33ptoo.components.KButton();
    kbtnUpdate = new com.k33ptoo.components.KButton();
    kbtnAdd = new com.k33ptoo.components.KButton();
    lblBracketUpperBound = new javax.swing.JLabel();
    txtBracketUpperBound = new javax.swing.JTextField();
    lblER = new javax.swing.JLabel();
    txtER = new javax.swing.JTextField();
    lblEE = new javax.swing.JLabel();
    txtEE = new javax.swing.JTextField();
    pnlEmail1 = new javax.swing.JPanel();
    kbtnSalaryGenerator = new com.k33ptoo.components.KButton();
    pnlLogo = new javax.swing.JPanel();
    lblLogo = new javax.swing.JLabel();

```

```

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setBackground(new java.awt.Color(255, 255, 255));
setResizable(false);

pnlBackground.setkEndColor(new java.awt.Color(66, 133, 244));
pnlBackground.setkGradientFocus(900);
pnlBackground.setkStartColor(new java.awt.Color(66, 133, 244));

pnlSearch.setBackground(new java.awt.Color(238, 238, 238));
pnlSearch.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
pnlSearch.setForeground(new java.awt.Color(36, 48, 94));

txtSearch.setBackground(new java.awt.Color(252, 250, 255));
txtSearch.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtSearch.setForeground(new java.awt.Color(8, 8, 8));
txtSearch.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(38, 117, 247)));
txtSearch.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtSearch.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtSearchActionPerformed(evt);
    }
});
txtSearch.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtSearchKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtSearchKeyTyped(evt);
    }
});

lblSearch.setIcon(new javax.swing.ImageIcon(getClass().getResource("/29x29
Search.png"))); // NOI18N
lblSearch.setText("jLabel2");

cmbIn.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "All", "BID",
"Bracket Lower Bound", "Bracket Upper Bound", "ER", "EE" }));
cmbIn.addItemListener(new java.awt.event.ItemListener() {
    public void itemStateChanged(java.awt.event.ItemEvent evt) {
        cmbInItemStateChanged(evt);
    }
});

```

```

cmbIn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cmbInActionPerformed(evt);
    }
});
cmbIn.addPropertyChangeListener(new java.beans.PropertyChangeListener() {
    public void propertyChange(java.beans.PropertyChangeEvent evt) {
        cmbInPropertyChange(evt);
    }
});

txtIn.setFont(new java.awt.Font("Segoe UI Light", 0, 16)); // NOI18N
txtIn.setForeground(new java.awt.Color(8, 8, 8));
txtIn.setText("In:");

txtSort.setFont(new java.awt.Font("Segoe UI Light", 0, 16)); // NOI18N
txtSort.setForeground(new java.awt.Color(8, 8, 8));
txtSort.setText("Sort By:");

cmbColumn.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "BID",
"Bracket Lower Bound", "Bracket Upper Bound", "ER", "EE" }));
cmbColumn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cmbColumnActionPerformed(evt);
    }
});

cmbOrder.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {
"Ascending", "Descending" }));

kbtnSort.setText("Sort");
kbtnSort.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnSort.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnSort.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnSort.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnSort.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnSort.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnSort.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnSortActionPerformed(evt);
    }
});

kbtnReset.setText("Reset");

```

```

kbtnReset.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnReset.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnReset.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnReset.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnReset.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnReset.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnResetActionPerformed(evt);
    }
});

javax.swing.GroupLayout pnlSearchLayout = new javax.swing.GroupLayout(pnlSearch);
pnlSearch.setLayout(pnlSearchLayout);
pnlSearchLayout.setHorizontalGroup(
    pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlSearchLayout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addComponent(lblSearch, javax.swing.GroupLayout.PREFERRED_SIZE, 29,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(txtSearch, javax.swing.GroupLayout.PREFERRED_SIZE, 156,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(txtIn)
            .addGap(18, 18, 18)
            .addComponent(cmbIn, javax.swing.GroupLayout.PREFERRED_SIZE, 80,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(txtSort)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(cmbColumn, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(cmbOrder, javax.swing.GroupLayout.PREFERRED_SIZE, 84,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(kbtnSort, javax.swing.GroupLayout.PREFERRED_SIZE, 55,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(kbtnReset, javax.swing.GroupLayout.PREFERRED_SIZE, 58,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(10, 10, 10)
        )
);
pnlSearchLayout.setVerticalGroup(

```

```

        pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlSearchLayout.createSequentialGroup()
        .addGap(5, 5, 5)

.addGroup(pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addGroup(pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
        .addComponent(kbtnSort, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, 0, Short.MAX_VALUE)

.addGroup(pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(cmbOrder, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(cmbColumn, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(kbtnReset, javax.swing.GroupLayout.PREFERRED_SIZE,
27, javax.swing.GroupLayout.PREFERRED_SIZE)))

.addGroup(pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(lblSearch)

.addGroup(pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(txtSearch, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(cmbIn, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(txtIn)
        .addComponent(txtSort))))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);

pnlTable.setBackground(new java.awt.Color(55, 71, 133));
pnlTable.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
pnlTable.setForeground(new java.awt.Color(36, 48, 94));

tblEmployees.setAutoCreateRowSorter(true);
tblEmployees.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {

```



```

kbtnDeleteAll.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnDeleteAll.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnDeleteAll.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnDeleteAll.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnDeleteAll.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnDeleteAll.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnDeleteAllActionPerformed(evt);
    }
});

kbtnDelete.setText("Delete");
kbtnDelete.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnDelete.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnDelete.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnDelete.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnDelete.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnDelete.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnDelete.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnDeleteActionPerformed(evt);
    }
});

javax.swing.GroupLayout pnlDeleteLayout = new javax.swing.GroupLayout(pnlDelete);
pnlDelete.setLayout(pnlDeleteLayout);
pnlDeleteLayout.setHorizontalGroup(
    pnlDeleteLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlDeleteLayout.createSequentialGroup()
            .addGap(16, 16, 16)
            .addComponent(kbtnDeleteAll, javax.swing.GroupLayout.PREFERRED_SIZE, 78,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(kbtnDelete, javax.swing.GroupLayout.PREFERRED_SIZE, 78,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(16, 16, 16)
        )
);
pnlDeleteLayout.setVerticalGroup(
    pnlDeleteLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlDeleteLayout.createSequentialGroup()
            .addGap(16, 16, 16)
            .addComponent(kbtnDeleteAll, javax.swing.GroupLayout.PREFERRED_SIZE, 30,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(kbtnDelete, javax.swing.GroupLayout.PREFERRED_SIZE, 30,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(16, 16, 16)
        )
);

```

```

.addGroup(pnlDeleteLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(kbtnDeleteAll, javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(kbtnDelete, javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addContainerGap(22, Short.MAX_VALUE))
);

pnlDelete1.setBackground(new java.awt.Color(238, 238, 238));
pnlDelete1.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
pnlDelete1.setForeground(new java.awt.Color(36, 48, 94));

txtBID.setBackground(new java.awt.Color(252, 250, 255));
txtBID.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtBID.setForeground(new java.awt.Color(8, 8, 8));
txtBID.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtBID.setFocusable(false);
txtBID.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtBID.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtBIDActionPerformed(evt);
    }
});
txtBID.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtBIDKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtBIDKeyTyped(evt);
    }
});

lblBID.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblBID.setForeground(new java.awt.Color(8, 8, 8));
lblBID.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblBID.setText("BID:");

txtBracketLowerBound.setBackground(new java.awt.Color(252, 250, 255));
txtBracketLowerBound.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtBracketLowerBound.setForeground(new java.awt.Color(8, 8, 8));

```

```

txtBracketLowerBound.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2,
new java.awt.Color(204, 204, 204)));
txtBracketLowerBound.setFocusTraversalPolicyProvider(true);
txtBracketLowerBound.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtBracketLowerBound.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtBracketLowerBoundActionPerformed(evt);
    }
});
txtBracketLowerBound.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtBracketLowerBoundKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtBracketLowerBoundKeyTyped(evt);
    }
});

```

```

lblBracketLowerBound.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblBracketLowerBound.setForeground(new java.awt.Color(8, 8, 8));
lblBracketLowerBound.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblBracketLowerBound.setText("Bracket lower bound:");

```

```

kbtnClearFields.setText("Clear Fields");
kbtnClearFields.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnClearFields.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnClearFields.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnClearFields.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnClearFields.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnClearFields.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnClearFields.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnClearFieldsActionPerformed(evt);
    }
});

```

```

kbtnDeselect.setText("Deselect");
kbtnDeselect.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnDeselect.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnDeselect.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnDeselect.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnDeselect.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnDeselect.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnDeselect.addActionListener(new java.awt.event.ActionListener() {

```

```

        public void actionPerformed(java.awt.event.ActionEvent evt) {
            kbtnDeselectActionPerformed(evt);
        }
    });

    kbtnUpdate.setText("Update");
    kbtnUpdate.setkEndColor(new java.awt.Color(28, 84, 140));
    kbtnUpdate.setkHoverEndColor(new java.awt.Color(76, 111, 167));
    kbtnUpdate.setkHoverForeground(new java.awt.Color(252, 250, 255));
    kbtnUpdate.setkHoverStartColor(new java.awt.Color(76, 111, 167));
    kbtnUpdate.setkSelectedColor(new java.awt.Color(252, 250, 255));
    kbtnUpdate.setkStartColor(new java.awt.Color(28, 84, 140));
    kbtnUpdate.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            kbtnUpdateActionPerformed(evt);
        }
    });

    kbtnAdd.setText("Add");
    kbtnAdd.setkEndColor(new java.awt.Color(28, 84, 140));
    kbtnAdd.setkHoverEndColor(new java.awt.Color(76, 111, 167));
    kbtnAdd.setkHoverForeground(new java.awt.Color(252, 250, 255));
    kbtnAdd.setkHoverStartColor(new java.awt.Color(76, 111, 167));
    kbtnAdd.setkSelectedColor(new java.awt.Color(252, 250, 255));
    kbtnAdd.setkStartColor(new java.awt.Color(28, 84, 140));
    kbtnAdd.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            kbtnAddActionPerformed(evt);
        }
    });

    lblBracketUpperBound.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
    lblBracketUpperBound.setForeground(new java.awt.Color(8, 8, 8));
    lblBracketUpperBound.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
    lblBracketUpperBound.setText("Bracket upper bound");

    txtBracketUpperBound.setBackground(new java.awt.Color(252, 250, 255));
    txtBracketUpperBound.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
    txtBracketUpperBound.setForeground(new java.awt.Color(8, 8, 8));
    txtBracketUpperBound.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2,
new java.awt.Color(204, 204, 204)));
    txtBracketUpperBound.setMargin(new java.awt.Insets(2, 2, 2, 2));
    txtBracketUpperBound.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {

```

```

        txtBracketUpperBoundActionPerformed(evt);
    }
});
txtBracketUpperBound.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtBracketUpperBoundKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtBracketUpperBoundKeyTyped(evt);
    }
});

lblER.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblER.setForeground(new java.awt.Color(8, 8, 8));
lblER.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblER.setText("Employer's % Share (ER)");

txtER.setBackground(new java.awt.Color(252, 250, 255));
txtER.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtER.setForeground(new java.awt.Color(8, 8, 8));
txtER.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtER.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtER.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtERActionPerformed(evt);
    }
});
txtER.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtERKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtERKeyTyped(evt);
    }
});

lblEE.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblEE.setForeground(new java.awt.Color(8, 8, 8));
lblEE.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblEE.setText("Employee's % Share (EE)");

txtEE.setBackground(new java.awt.Color(252, 250, 255));
txtEE.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N

```



```

        .addComponent(lblEE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
pnlDelete1Layout.createSequentialGroup())
        .addComponent(kbtnClearFields,
javax.swing.GroupLayout.PREFERRED_SIZE, 67,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(kbtnDeselect,
javax.swing.GroupLayout.PREFERRED_SIZE, 58,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlDelete1Layout.createSequentialGroup()
        .addComponent(kbtnAdd, javax.swing.GroupLayout.PREFERRED_SIZE,
61, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 116,
Short.MAX_VALUE)
        .addComponent(kbtnUpdate,
javax.swing.GroupLayout.PREFERRED_SIZE, 53,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(txtEE)))
        .addGroup(pnlDelete1Layout.createSequentialGroup()
        .addComponent(lblBID)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(txtBID, javax.swing.GroupLayout.PREFERRED_SIZE, 103,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, Short.MAX_VALUE))
        .addGroup(pnlDelete1Layout.createSequentialGroup()
        .addComponent(lblER)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(txtER)))
        .addContainerGap()
    );
    pnlDelete1Layout.setVerticalGroup(
        pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlDelete1Layout.createSequentialGroup()
        .addContainerGap()

```



```

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblBID)
    .addComponent(txtBID, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    .addComponent(lblBracketLowerBound)
    .addComponent(txtBracketLowerBound,
javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblBracketUpperBound)
    .addComponent(txtBracketUpperBound,
javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblER)
    .addComponent(txtER, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblEE)
    .addComponent(txtEE, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(kbtnClearFields, javax.swing.GroupLayout.PREFERRED_SIZE,
30, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(kbtnDeselect, javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addComponent(kbtnUpdate, javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(kbtnAdd, javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );

    pnlEmail1.setBackground(new java.awt.Color(238, 238, 238));
    pnlEmail1.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
    pnlEmail1.setForeground(new java.awt.Color(36, 48, 94));

    kbtnSalaryGenerator.setText("Salary Generator");
    kbtnSalaryGenerator.setkEndColor(new java.awt.Color(28, 84, 140));
    kbtnSalaryGenerator.setkHoverEndColor(new java.awt.Color(76, 111, 167));
    kbtnSalaryGenerator.setkHoverForeGround(new java.awt.Color(252, 250, 255));
    kbtnSalaryGenerator.setkHoverStartColor(new java.awt.Color(76, 111, 167));
    kbtnSalaryGenerator.setkSelectedColor(new java.awt.Color(252, 250, 255));
    kbtnSalaryGenerator.setkStartColor(new java.awt.Color(28, 84, 140));
    kbtnSalaryGenerator.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            kbtnSalaryGeneratorActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout pnlEmail1Layout = new javax.swing.GroupLayout(pnlEmail1);
    pnlEmail1.setLayout(pnlEmail1Layout);
    pnlEmail1Layout.setHorizontalGroup(
        pnlEmail1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlEmail1Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(kbtnSalaryGenerator, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addContainerGap())
    );
    pnlEmail1Layout.setVerticalGroup(
        pnlEmail1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(pnlEmail1Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(kbtnSalaryGenerator, javax.swing.GroupLayout.PREFERRED_SIZE,
36, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(9, Short.MAX_VALUE))
    );

```

```

        pnlLogo.setBackground(new java.awt.Color(36, 48, 94));
        pnlLogo.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
        pnlLogo.setForeground(new java.awt.Color(36, 48, 94));

        lblLogo.setIcon(new javax.swing.ImageIcon(getClass().getResource("/250x91 PH.png")));
// NOI18N
        lblLogo.setText("jLabel1");

        javax.swing.GroupLayout pnlLogoLayout = new javax.swing.GroupLayout(pnlLogo);
        pnlLogo.setLayout(pnlLogoLayout);
        pnlLogoLayout.setHorizontalGroup(
            pnlLogoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlLogoLayout.createSequentialGroup()
                    .addGap(71, 71, 71)
                    .addComponent(lblLogo, javax.swing.GroupLayout.PREFERRED_SIZE, 251,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(71, 71, 71))
        );
        pnlLogoLayout.setVerticalGroup(
            pnlLogoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(pnlLogoLayout.createSequentialGroup()
                    .addGap(104, 104, 104)
                    .addComponent(lblLogo)
                    .addGap(105, Short.MAX_VALUE))
        );

        javax.swing.GroupLayout pnlBackgroundLayout = new
javax.swing.GroupLayout(pnlBackground);
        pnlBackground.setLayout(pnlBackgroundLayout);
        pnlBackgroundLayout.setHorizontalGroup(

pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(pnlBackgroundLayout.createSequentialGroup()
                .addGap(71, 71, 71)
                .addComponent(lblDelete, javax.swing.GroupLayout.PREFERRED_SIZE, 251,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(71, 71, 71))
        );
        pnlBackgroundLayout.setVerticalGroup(
            pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(pnlBackgroundLayout.createSequentialGroup()
                    .addGap(104, 104, 104)
                    .addComponent(lblDelete, javax.swing.GroupLayout.PREFERRED_SIZE, 251,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(105, Short.MAX_VALUE))
        );

```

```

        .addGroup(pnlBackgroundLayout.createSequentialGroup())
        .addComponent(pnlEmail, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(pnlEmail1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addComponent(pnlLogo, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addGroup(pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
        .addComponent(pnlTable, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(pnlSearch, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addContainerGap())
    );
    pnlBackgroundLayout.setVerticalGroup(

pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(pnlBackgroundLayout.createSequentialGroup())
    .addContainerGap()

    .addGroup(pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
        .addComponent(pnlEmail, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(pnlSearch, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(pnlEmail1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

    .addGroup(pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlBackgroundLayout.createSequentialGroup())
        .addComponent(pnlDelete1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(pnlDelete, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

```

```

        .addComponent(pnlLogo, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addComponent(pnlTable, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(pnlBackground, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(pnlBackground, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );

    pack();
    setLocationRelativeTo(null);
} // </editor-fold>

```

```

private void kbtnSortActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        String search = txtSearch.getText().trim();

        int column = cmbColumn.getSelectedIndex();

        String[] columnnames =
{"bid","bracketlowerbound","bracketupperbound","ph_er_p","ph_ee_p"};

        String strcolumn = columnnames[column];

        int sort = cmbOrder.getSelectedIndex();
        String strsort = "";
        switch (sort){
            case 0:
                strsort = "ASC";
                break;
            case 1:
                strsort = "DESC";
        }
    }
}

```

```

        if (column==0 && sort==0){
            refreshtable("SELECT * FROM philhealth");
        }
//        else if (column==0 && sort==1){
//            //Iterate through the resultset backwards
//            DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
//            String databaseURL = "jdbc:derby:SALARY_GENERATOR";
//            Connection con = DriverManager.getConnection(databaseURL);
//            Statement st =
con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_READ_ONL
Y);
//            String sql = "SELECT * FROM info";
//            ResultSet rs = st.executeQuery(sql);
//
//            //Moves the cursor to the end of the ResultSet object
//            rs.afterLast();
//            model.setRowCount(0);
//            while(rs.previous())
//            {
//                model.addRow(new
Object[]{rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4)});
//            }
//            rs.close();
//            st.close();
//            con.close();
//        }
        else{
            refreshtable("SELECT * FROM philhealth order by "+strcolumn+" "+strsort);
        }

    }
    catch (Exception e){

    }
}

private void cmbColumnActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void tblEmployeesMouseClicked(java.awt.event.MouseEvent evt) {
    try{
        //Prerequisite for the update function

```

```

DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
int selectedrow=tblEmployees.getSelectedRow();
if (tblEmployees.getRowSorter()!=null) {
    //Gets the correct row for a filtered jtable (RowFilter)
    selectedrow = tblEmployees.getRowSorter().convertRowIndexToModel(selectedrow);
}
String eid = "";
String bracketlower = "";
String bracketupper = "";
String er = "";
String ee = "";

//    eid = txtEID.getText();

eid = model.getValueAt(selectedrow,0).toString();
bracketlower = model.getValueAt(selectedrow,1).toString();
bracketupper = model.getValueAt(selectedrow,2).toString();
er = model.getValueAt(selectedrow,3).toString();
ee = model.getValueAt(selectedrow,4).toString();

this.selectedroweid = eid;
txtBID.setText(eid);
txtBracketLowerBound.setText(bracketlower);
txtBracketUpperBound.setText(bracketupper);
txtER.setText(er);
txtEE.setText(ee);
}
catch (Exception e){
    e.printStackTrace();
}
}

private void cmbInPropertyChange(java.beans.PropertyChangeEvent evt) {

}

private void cmbInActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void cmbInItemStateChanged(java.awt.event.ItemEvent evt) {
    try{
        search();
    }
}

```

```

    }
    catch (Exception e){

    }
}

private void txtSearchKeyTyped(java.awt.event.KeyEvent evt) {

}

private void txtSearchKeyReleased(java.awt.event.KeyEvent evt) {
    try{
        search();
    }
    catch (Exception e){

    }
}

private void txtSearchActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void kbtnResetActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        //Resets the search and sort functions; restores table to original
        refreshtable("SELECT * FROM philhealth");
        cmbColumn.setSelectedIndex(0);
        cmbOrder.setSelectedIndex(0);

        txtSearch.setText("");
        cmbIn.setSelectedIndex(0);

        search();
    }
    catch (Exception e){

    }
}

private void kbtnSalaryGeneratorActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        new SalaryGenerator(this.uid,this.email).setVisible(true);
        this.setVisible(false);
    }
}

```



```

    }
    catch (Exception e){

    }
}

```

```

private void kbtnDeleteAllActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        int confirm = JOptionPane.showConfirmDialog(null,"Are you want to delete ALL
records?");
        if (confirm==0){
            DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
            int rowcount = model.getRowCount(); //number of rows in model

            if (rowcount==0){
                JOptionPane.showMessageDialog(null, "The table is empty.");
            }
            else{
                int eids[] = new int[rowcount];
                for (int i = 0; i<rowcount; i++){
                    eids[i] = Integer.parseInt(model.getValueAt(i,0).toString());

                }
                String databaseURL = "jdbc:derby:SALARY_GENERATOR";
                Connection con = DriverManager.getConnection(databaseURL);

                String sql;
                for (int i = 0; i<eids.length; i++){
                    //Deletion based on eid
                    Statement st = con.createStatement();
                    sql = "DELETE FROM philhealth WHERE bid=" + eids[i];
                    st.execute(sql);
                }

                maxfinder();

                model.setRowCount(0);
            }
        }
    }
    catch(Exception e){

```

```

    }
}

private void kbtnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
        int rowcount = model.getRowCount();

        //Allows it to work with the filtered jTable (RowFilter)
        int rows[] = tblEmployees.getSelectedRows();
        for (int i = 0; i<rows.length; i++){
            rows[i] = tblEmployees.getRowSorter().convertRowIndexToModel(rows[i]);
        }

        if ("-1".equals(this.selectedroweid)){
            JOptionPane.showMessageDialog(null, "A row has not been selected.");
        }
        else{
            int confirm = JOptionPane.showConfirmDialog(null,"Are you sure?");

            if (confirm==0){
                if (rows.length>0){//Delete multiple or single
                    int eids[] = new int[rows.length];
                    for (int i = 0; i<rows.length; i++){
                        eids[i] = Integer.parseInt(model.getValueAt(rows[i],0).toString());
                    }
                    String databaseURL = "jdbc:derby:SALARY_GENERATOR";
                    Connection con = DriverManager.getConnection(databaseURL);

                    String sql;
                    for (int i = 0; i<eids.length; i++){
                        Statement st = con.createStatement();

                        sql = "DELETE FROM philhealth WHERE bid=" + eids[i];
                        st.execute(sql);
                    }

                    //            txtFullName.setText("");
                    //            txtEmail.setText("");
                    //            txtPhoneNumber.setText("");

                    this.selectedroweid = "-1";
                    refreshtable("SELECT * FROM philhealth");
                }
            }
        }
    }
}

```

```

        maxfinder();
    }
}

}

catch (Exception e){

}

}

private void kbtnAddActionPerformed(java.awt.event.ActionEvent evt) {
    try{
//        String eid = "";
        String bracketlower = "";
        String bracketupper = "";
        String er = "";
        String ee = "";

//        eid = txtEID.getText();

        bracketlower = txtBracketLowerBound.getText();
        bracketupper = txtBracketUpperBound.getText();
        er = txtER.getText();
        ee = txtEE.getText();

        System.out.println("Email: " + email);

        //Error trapping
        if ("".equals(bracketlower)){
            JOptionPane.showMessageDialog(null, "Lower bound cannot be empty.");
        }
        else if ("".equals(bracketupper)){
            JOptionPane.showMessageDialog(null, "Upper bound cannot be empty.");
        }
        else if ("".equals(er)){
            JOptionPane.showMessageDialog(null, "Employer Share cannot be empty.");
        }
        else if ("".equals(ee)){
            JOptionPane.showMessageDialog(null, "Employee Share cannot be empty.");
        }
        else if (Double.parseDouble(bracketlower)>99999999 ||
Double.parseDouble(bracketlower)!=((double)Math.round(Double.parseDouble(bracketlower)*10
0)/100){

```

```

        JOptionPane.showMessageDialog(null, "Bracket Lower Bound can be a maximum of
99,999,999.99. Decimal precision is only 2 decimal places.");
    }
    else if (Double.parseDouble(bracketupper)>999999999 ||
Double.parseDouble(bracketupper)!=((double)Math.round(Double.parseDouble(bracketupper)*1
00)/100){
        JOptionPane.showMessageDialog(null, "Bracket Upper Bound can be a maximum of
99,999,999.99. Decimal precision is only 2 decimal places.");
    }
    else if (Double.parseDouble(er)>999999999 ||
Double.parseDouble(er)!=((double)Math.round(Double.parseDouble(er)*100)/100){
        JOptionPane.showMessageDialog(null, "Employer % Share (ER) can be a maximum
of 99,999,999.99. Decimal precision is only 2 decimal places.");
    }
    else if (Double.parseDouble(ee)>999999999 ||
Double.parseDouble(ee)!=((double)Math.round(Double.parseDouble(ee)*100)/100){
        JOptionPane.showMessageDialog(null, "Employee % Share (EE) can be a maximum
of 99,999,999.99. Decimal precision is only 2 decimal places.");
    }
    else{
        maxfinder();
        this.max++;
        String strmax = Integer.toString(this.max);

        System.out.println("strmax: " + strmax);

        String databaseURL = "jdbc:derby:SALARY_GENERATOR";
        Connection con = DriverManager.getConnection(databaseURL);
        Statement st = con.createStatement();

        String sql = "INSERT INTO philhealth (bid, bracketlowerbound, bracketupperbound,
ph_er_p, ph_ee_p) "
            + "VALUES (" + strmax + "," + bracketlower + "," + bracketupper + "," + er + "," + ee + ")";

        //pay_period determined by the max from salary and loan history

        st.execute(sql);
        con.close();

        refreshtable("SELECT * FROM philhealth");

        txtBID.setText(Integer.toString(max));
        txtBracketLowerBound.setText("");
        txtBracketUpperBound.setText("");
    }
}

```

```

        txtER.setText("");
        txtEE.setText("");

        JOptionPane.showMessageDialog(null, "Added.");
    }
}
catch (Exception e){
    e.printStackTrace();
}
}

private void kbtnUpdateActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        //Prerequisite for the update function
        String bid = "";
        String bracketlower = "";
        String bracketupper = "";
        String er = "";
        String ee = "";

//        eid = txtEID.getText();

        bid = txtBID.getText();
        bracketlower = txtBracketLowerBound.getText();
        bracketupper = txtBracketUpperBound.getText();
        er = txtER.getText();
        ee = txtEE.getText();

        System.out.println("Email: " + email);

        //Error trapping
        if (Double.parseDouble(bracketlower)>999999999 ||
Double.parseDouble(bracketlower)!=((double)Math.round(Double.parseDouble(bracketlower)*10
0)/100){
            JOptionPane.showMessageDialog(null, "Bracket Lower Bound can be a maximum of
99,999,999.99. Decimal precision is only 2 decimal places.");
        }
        else if (Double.parseDouble(bracketupper)>999999999 ||
Double.parseDouble(bracketupper)!=((double)Math.round(Double.parseDouble(bracketupper)*1
00)/100){
            JOptionPane.showMessageDialog(null, "Bracket Upper Bound can be a maximum of
99,999,999.99. Decimal precision is only 2 decimal places.");
        }
    }
}

```

```

        else if (Double.parseDouble(er)>999999999 ||
Double.parseDouble(er)!=((double)Math.round(Double.parseDouble(er)*100)/100){
            JOptionPane.showMessageDialog(null, "Employer % Share (ER) can be a maximum
of 99,999,999.99. Decimal precision is only 2 decimal places.");
        }
        else if (Double.parseDouble(ee)>999999999 ||
Double.parseDouble(ee)!=((double)Math.round(Double.parseDouble(ee)*100)/100){
            JOptionPane.showMessageDialog(null, "Employee % Share (EE) can be a maximum
of 99,999,999.99. Decimal precision is only 2 decimal places.");
        }
        else{
            String strmax = Integer.toString(max);

            String databaseURL = "jdbc:derby:SALARY_GENERATOR";
            Connection con = DriverManager.getConnection(databaseURL);
            Statement st = con.createStatement();

            String sql = "UPDATE philhealth SET bid="+bid+",
bracketlowerbound="+bracketlower+", bracketupperbound="+bracketupper+", ph_ee_p="+ee+",
ph_er_p="+er+" WHERE bid="+bid;
            st.execute(sql);
            con.close();

            //No row currently selected.
            selectedroweid = "-1";

            JOptionPane.showMessageDialog(null, "Updated.");
            tblEmployees.clearSelection();

            refreshtable("SELECT * FROM philhealth");
            maxfinder();

            int max = this.max;
            max++;

            txtBID.setText(Integer.toString(max));
            txtBracketLowerBound.setText("");
            txtBracketUpperBound.setText("");
            txtER.setText("");
            txtEE.setText("");
        }
    }
    catch (Exception e){

```

```

        e.printStackTrace();
    }
}

private void kbtnDeselectActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        //Deselects all rows
        //      DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
        //      tblEmployees.removeRowSelectionInterval(0, model.getRowCount()-1);
        tblEmployees.clearSelection();
        //resets selectedroweid
        this.selectedroweid = "-1";
    }
    catch (Exception e){

    }
}

private void kbtnClearFieldsActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        txtBID.setText("");
        txtBracketLowerBound.setText("");
        txtBracketUpperBound.setText("");
        txtER.setText("");
        txtEE.setText("");
    }
    catch(Exception e){

    }
}

private void txtBracketLowerBoundKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtBracketLowerBoundKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtBracketLowerBoundActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtBIDKeyTyped(java.awt.event.KeyEvent evt) {

```

```

    // TODO add your handling code here:
}

private void txtBIDKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtBIDActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtBracketUpperBoundActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtBracketUpperBoundKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtBracketUpperBoundKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtERActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtERKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtERKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtEEActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtEEKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtEEKeyTyped(java.awt.event.KeyEvent evt) {

```



```

    // TODO add your handling code here:
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Register().setVisible(true);
        }
    }

```

```

    });
}

// Variables declaration - do not modify
private javax.swing.JComboBox<String> cmbColumn;
private javax.swing.JComboBox<String> cmbIn;
private javax.swing.JComboBox<String> cmbOrder;
private javax.swing.JScrollPane jScrollPane1;
private com.k33ptoo.components.KButton kbtnAdd;
private com.k33ptoo.components.KButton kbtnClearFields;
private com.k33ptoo.components.KButton kbtnDelete;
private com.k33ptoo.components.KButton kbtnDeleteAll;
private com.k33ptoo.components.KButton kbtnDeselect;
private com.k33ptoo.components.KButton kbtnReset;
private com.k33ptoo.components.KButton kbtnSalaryGenerator;
private com.k33ptoo.components.KButton kbtnSort;
private com.k33ptoo.components.KButton kbtnUpdate;
private javax.swing.JLabel lblBID;
private javax.swing.JLabel lblBracketLowerBound;
private javax.swing.JLabel lblBracketUpperBound;
private javax.swing.JLabel lblEE;
private javax.swing.JLabel lblER;
private javax.swing.JLabel lblEditPersonalRecords;
private javax.swing.JLabel lblLogo;
private javax.swing.JLabel lblSearch;
private com.k33ptoo.components.KGradientPanel pnlBackground;
private javax.swing.JPanel pnlDelete;
private javax.swing.JPanel pnlDelete1;
private javax.swing.JPanel pnlEmail;
private javax.swing.JPanel pnlEmail1;
private javax.swing.JPanel pnlLogo;
private javax.swing.JPanel pnlSearch;
private javax.swing.JPanel pnlTable;
private javax.swing.JTable tblEmployees;
private javax.swing.JTextField txtBID;
private javax.swing.JTextField txtBracketLowerBound;
private javax.swing.JTextField txtBracketUpperBound;
private javax.swing.JTextField txtEE;
private javax.swing.JTextField txtER;
private javax.swing.JLabel txtIn;
private javax.swing.JTextField txtSearch;
private javax.swing.JLabel txtSort;
// End of variables declaration
}

```

Manipulate HDMF Allocations Page:

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
 license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */
```

```
/**
 *
 * @author User
 */
import java.awt.Color;
import java.sql.Connection;
import java.sql.DatabaseMetaData;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.swing.JOptionPane;
import javax.swing.RowFilter;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableRowSorter;

import javax.swing.text.*;
```

```
public class HDMF extends javax.swing.JFrame {
```

```
    /**
     * Creates new form Register
     */
```

```
    //Global Variables
    int max = 0;
    String email = "", uid = "";
    String selectedroweid = "-1";
    //Setting up the regexfilter
    TableRowSorter sorter;
```

```

private void initComponents2(){
    //Overwrite Java Driver Code

    //Initializing the table with specific settings
    //For the regexfilter later on
    TableRowSorter sorter = new
TableRowSorter<>((DefaultTableModel)tblEmployees.getModel());
    this.sorter = sorter;
    tblEmployees.setRowSorter(sorter);
    //Disables table sorting
    sorter.setSortable(0, false);
    sorter.setSortable(1, false);
    sorter.setSortable(2, false);
    sorter.setSortable(3, false);
    //Disables rearranging table columns
    tblEmployees.getTableHeader().setReorderingAllowed(false);
    //Textboxes set to a clear background
    /*txtFullName.setBackground(new Color(0, 0, 0, 0));
    txtEmail.setBackground(new Color(0, 0, 0, 0));
    txtPhoneNumber.setBackground(new Color(0, 0, 0, 0));*/

    //Adding numeric filters to necessary textboxes
    ((AbstractDocument) txtBID.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
    ((AbstractDocument) txtBracketLowerBound.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
    ((AbstractDocument) txtBracketUpperBound.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
    ((AbstractDocument) txtER.getDocument()).setDocumentFilter(new
NumericDocumentFilter());
    ((AbstractDocument) txtEE.getDocument()).setDocumentFilter(new
NumericDocumentFilter());

    String maxeid = Integer.toString(this.max+1);
    txtBID.setText(maxeid);
}

public static boolean doesTableExists(String tableName, Connection conn)
throws SQLException
{
    DatabaseMetaData meta = conn.getMetaData();
    ResultSet result = meta.getTables(null, null, tableName.toUpperCase(), null);

```

```

        return result.next();
    }

private void maxfinder(){ //figure out the highest eid
    try{
        String databaseURL = "jdbc:derby:SALARY_GENERATOR";
        Connection con = DriverManager.getConnection(databaseURL);

        String sql;
        Statement st = con.createStatement();

        sql = "SELECT * FROM hdmf";
        ResultSet rs = st.executeQuery(sql);

        //db stands for database

        int max = 0;
        while (rs.next()==true){
            String dbeid = rs.getString(1);
            int eid = Integer.parseInt(dbeid);
            if (eid>max){
                max = eid;
            }
        }
        this.max = max;
        con.close();
    }
    catch (Exception e){
        e.printStackTrace();
    }
}

private void search(){
    try{
        int in = cmbIn.getSelectedIndex();

        String search = txtSearch.getText().trim();

        //Search using RowFilter and regex (regular expression) to find substrings
        RowFilter<DefaultTableModel, Object> rf = null;
        try {
            if (in == 0){
                rf = RowFilter.regexFilter("(?i)" + search); //("(?i)" for case insensitive filter
            }
        }
    }
}

```

```

        }
        else{
            rf = RowFilter.regexFilter("(?i)" + search, in - 1);
        }
    }
    catch (java.util.regex.PatternSyntaxException e) {
//        System.out.println("Error Found");
        return;
    }
    sorter.setRowFilter(rf);
}
catch (Exception e){
    e.printStackTrace();
}
}

private void refreshtable(String sql){//reset table contents, works for SELECT queries only
    try{
        DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
        String databaseURL = "jdbc:derby:SALARY_GENERATOR";
        Connection con = DriverManager.getConnection(databaseURL);

        Statement st = con.createStatement();
        ResultSet rs = st.executeQuery(sql);

        model.setRowCount(0); //reset the table
        while (rs.next()==true){

            model.addRow(new Object[]{rs.getString(1), rs.getString(2), rs.getString(3),
rs.getString(5), rs.getString(4)});
        }
        con.close();

    }
    catch (Exception e){
        System.out.println("Refresh table exception");
        e.printStackTrace();
    }
}
}

```

```

public HDMF() {
    initComponents();

    try{

        maxfinder();

        refreshtable("SELECT * FROM hdmf");

        initComponents2();

    }
    catch (Exception e){
        e.printStackTrace();
    }
}

public HDMF(String uid, String email){ //CATCH ALL INCOMING DATA
    initComponents();
    try{

        this.uid = uid;
        this.email = email;

        maxfinder();

        refreshtable("SELECT * FROM hdmf");

        initComponents2();

    }
    catch (Exception e){
        e.printStackTrace();
    }
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.

```

```

*/
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    pnlBackground = new com.k33ptoo.components.KGradientPanel();
    pnlSearch = new javax.swing.JPanel();
    txtSearch = new javax.swing.JTextField();
    lblSearch = new javax.swing.JLabel();
    cmbIn = new javax.swing.JComboBox<>();
    txtIn = new javax.swing.JLabel();
    txtSort = new javax.swing.JLabel();
    cmbColumn = new javax.swing.JComboBox<>();
    cmbOrder = new javax.swing.JComboBox<>();
    kbtnSort = new com.k33ptoo.components.KButton();
    kbtnReset = new com.k33ptoo.components.KButton();
    pnlTable = new javax.swing.JPanel();
    jScrollPane1 = new javax.swing.JScrollPane();
    tblEmployees = new javax.swing.JTable();
    pnlEmail = new javax.swing.JPanel();
    lblEditPersonalRecords = new javax.swing.JLabel();
    pnlDelete = new javax.swing.JPanel();
    kbtnDeleteAll = new com.k33ptoo.components.KButton();
    kbtnDelete = new com.k33ptoo.components.KButton();
    pnlDelete1 = new javax.swing.JPanel();
    txtBID = new javax.swing.JTextField();
    lblBID = new javax.swing.JLabel();
    txtBracketLowerBound = new javax.swing.JTextField();
    lblBracketLowerBound = new javax.swing.JLabel();
    kbtnClearFields = new com.k33ptoo.components.KButton();
    kbtnDeselect = new com.k33ptoo.components.KButton();
    kbtnUpdate = new com.k33ptoo.components.KButton();
    kbtnAdd = new com.k33ptoo.components.KButton();
    lblBracketUpperBound = new javax.swing.JLabel();
    txtBracketUpperBound = new javax.swing.JTextField();
    lblER = new javax.swing.JLabel();
    txtER = new javax.swing.JTextField();
    lblEE = new javax.swing.JLabel();
    txtEE = new javax.swing.JTextField();
    pnlEmail1 = new javax.swing.JPanel();
    kbtnSalaryGenerator = new com.k33ptoo.components.KButton();
    pnlLogo = new javax.swing.JPanel();
    lblLogo = new javax.swing.JLabel();

```



```

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setBackground(new java.awt.Color(255, 255, 255));
setResizable(false);

pnlBackground.setkEndColor(new java.awt.Color(66, 133, 244));
pnlBackground.setkGradientFocus(900);
pnlBackground.setkStartColor(new java.awt.Color(66, 133, 244));

pnlSearch.setBackground(new java.awt.Color(238, 238, 238));
pnlSearch.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
pnlSearch.setForeground(new java.awt.Color(36, 48, 94));

txtSearch.setBackground(new java.awt.Color(252, 250, 255));
txtSearch.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtSearch.setForeground(new java.awt.Color(8, 8, 8));
txtSearch.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(38, 117, 247)));
txtSearch.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtSearch.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtSearchActionPerformed(evt);
    }
});
txtSearch.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtSearchKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtSearchKeyTyped(evt);
    }
});

lblSearch.setIcon(new javax.swing.ImageIcon(getClass().getResource("/29x29
Search.png"))); // NOI18N
lblSearch.setText("jLabel2");

cmbIn.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "All", "BID",
"Bracket Lower Bound", "Bracket Upper Bound", "ER", "EE" }));
cmbIn.addItemListener(new java.awt.event.ItemListener() {
    public void itemStateChanged(java.awt.event.ItemEvent evt) {
        cmbInItemStateChanged(evt);
    }
});

```

```

cmbIn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cmbInActionPerformed(evt);
    }
});
cmbIn.addPropertyChangeListener(new java.beans.PropertyChangeListener() {
    public void propertyChange(java.beans.PropertyChangeEvent evt) {
        cmbInPropertyChange(evt);
    }
});

txtIn.setFont(new java.awt.Font("Segoe UI Light", 0, 16)); // NOI18N
txtIn.setForeground(new java.awt.Color(8, 8, 8));
txtIn.setText("In:");

txtSort.setFont(new java.awt.Font("Segoe UI Light", 0, 16)); // NOI18N
txtSort.setForeground(new java.awt.Color(8, 8, 8));
txtSort.setText("Sort By:");

cmbColumn.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "BID",
"Bracket Lower Bound", "Bracket Upper Bound", "ER", "EE" }));
cmbColumn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cmbColumnActionPerformed(evt);
    }
});

cmbOrder.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {
"Ascending", "Descending" }));

kbtnSort.setText("Sort");
kbtnSort.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnSort.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnSort.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnSort.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnSort.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnSort.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnSort.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnSortActionPerformed(evt);
    }
});

kbtnReset.setText("Reset");

```

```

kbtnReset.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnReset.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnReset.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnReset.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnReset.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnReset.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnResetActionPerformed(evt);
    }
});

javax.swing.GroupLayout pnlSearchLayout = new javax.swing.GroupLayout(pnlSearch);
pnlSearch.setLayout(pnlSearchLayout);
pnlSearchLayout.setHorizontalGroup(
    pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlSearchLayout.createSequentialGroup()
            .addGap(18, 18, 18)
            .addComponent(lblSearch, javax.swing.GroupLayout.PREFERRED_SIZE, 29,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(txtSearch, javax.swing.GroupLayout.PREFERRED_SIZE, 156,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(txtIn)
            .addGap(18, 18, 18)
            .addComponent(cmbIn, javax.swing.GroupLayout.PREFERRED_SIZE, 80,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(txtSort)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(cmbColumn, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(cmbOrder, javax.swing.GroupLayout.PREFERRED_SIZE, 84,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(kbtnSort, javax.swing.GroupLayout.PREFERRED_SIZE, 55,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(kbtnReset, javax.swing.GroupLayout.PREFERRED_SIZE, 58,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
        )
        .addGap(18, 18, 18)
);
pnlSearchLayout.setVerticalGroup(

```

```

        pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(pnlSearchLayout.createSequentialGroup()
                .addGap(5, 5, 5)

                .addGroup(pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

                    .addGroup(pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                        .addComponent(kbtnSort, javax.swing.GroupLayout.Alignment.TRAILING,
                            javax.swing.GroupLayout.PREFERRED_SIZE, 0, Short.MAX_VALUE)

                        .addGroup(pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                            .addComponent(cmbOrder, javax.swing.GroupLayout.PREFERRED_SIZE,
                                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addComponent(cmbColumn, javax.swing.GroupLayout.PREFERRED_SIZE,
                                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addComponent(kbtnReset, javax.swing.GroupLayout.PREFERRED_SIZE,
                                27, javax.swing.GroupLayout.PREFERRED_SIZE)))

                        .addGroup(pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addComponent(lblSearch)

                            .addGroup(pnlSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                                .addComponent(txtSearch, javax.swing.GroupLayout.PREFERRED_SIZE,
                                    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                                .addComponent(cmbIn, javax.swing.GroupLayout.PREFERRED_SIZE,
                                    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                                .addComponent(txtIn)
                                .addComponent(txtSort))))
                            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                );

        pnlTable.setBackground(new java.awt.Color(55, 71, 133));
        pnlTable.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
            java.awt.Color(255, 255, 255)));
        pnlTable.setForeground(new java.awt.Color(36, 48, 94));

        tblEmployees.setAutoCreateRowSorter(true);
        tblEmployees.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {

```

```

    },
    new String [] {
        "BID", "Bracket Lower Bound", "Bracket Upper Bound", "ER", "EE"
    }
) {
    boolean[] canEdit = new boolean [] {
        false, false, false, false, false
    };

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});
tblEmployees.setToolTipText("");
tblEmployees.setAutoResizeMode(javax.swing.JTable.AUTO_RESIZE_LAST_COLUMN);
tblEmployees.getTableHeader().setReorderingAllowed(false);
tblEmployees.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        tblEmployeesMouseClicked(evt);
    }
});
jScrollPane1.setViewportView(tblEmployees);
if (tblEmployees.getColumnModel().getColumnCount() > 0) {
    tblEmployees.getColumnModel().getColumn(0).setResizable(false);
    tblEmployees.getColumnModel().getColumn(1).setResizable(false);
    tblEmployees.getColumnModel().getColumn(2).setResizable(false);
    tblEmployees.getColumnModel().getColumn(3).setResizable(false);
    tblEmployees.getColumnModel().getColumn(4).setResizable(false);
}

javax.swing.GroupLayout pnlTableLayout = new javax.swing.GroupLayout(pnlTable);
pnlTable.setLayout(pnlTableLayout);
pnlTableLayout.setHorizontalGroup(
    pnlTableLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlTableLayout.createSequentialGroup()
            .add(pnlTableLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .add(pnlTableLayout.createSequentialGroup()
                    .addContainerGap()
                    .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 757,
Short.MAX_VALUE)
                    .addContainerGap())
            )
        );
pnlTableLayout.setVerticalGroup(
    pnlTableLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlTableLayout.createSequentialGroup())
        .addContainerGap()
        .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 574,
Short.MAX_VALUE)
        .addContainerGap()
    );

    pnlEmail.setBackground(new java.awt.Color(238, 238, 238));
    pnlEmail.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
    pnlEmail.setForeground(new java.awt.Color(36, 48, 94));

    lblEditPersonalRecords.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
    lblEditPersonalRecords.setForeground(new java.awt.Color(8, 8, 8));
    lblEditPersonalRecords.setText("HDMF Allocations");

    javax.swing.GroupLayout pnlEmailLayout = new javax.swing.GroupLayout(pnlEmail);
    pnlEmail.setLayout(pnlEmailLayout);
    pnlEmailLayout.setHorizontalGroup(
        pnlEmailLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(pnlEmailLayout.createSequentialGroup()
                .addGroup(pnlEmailLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(pnlEmailLayout.createSequentialGroup()
                        .addContainerGap()
                        .addComponent(lblEditPersonalRecords)
                        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                    .addGroup(pnlEmailLayout.createSequentialGroup()
                        .addContainerGap()
                        .addComponent(lblEditPersonalRecords,
javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                )
            )
    );

    pnlDelete.setBackground(new java.awt.Color(238, 238, 238));
    pnlDelete.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
    pnlDelete.setForeground(new java.awt.Color(36, 48, 94));

    kbbtnDeleteAll.setText("Delete All");
    kbbtnDeleteAll.setkEndColor(new java.awt.Color(28, 84, 140));
    kbbtnDeleteAll.setkHoverEndColor(new java.awt.Color(76, 111, 167));

```



```

        .addComponent(kbtnDeleteAll, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(kbtnDelete, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addContainerGap(17, Short.MAX_VALUE))
    );

    pnlDelete1.setBackground(new java.awt.Color(238, 238, 238));
    pnlDelete1.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
    pnlDelete1.setForeground(new java.awt.Color(36, 48, 94));

    txtBID.setBackground(new java.awt.Color(252, 250, 255));
    txtBID.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
    txtBID.setForeground(new java.awt.Color(8, 8, 8));
    txtBID.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
    txtBID.setFocusable(false);
    txtBID.setMargin(new java.awt.Insets(2, 2, 2, 2));
    txtBID.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            txtBIDActionPerformed(evt);
        }
    });
    txtBID.addKeyListener(new java.awt.event.KeyAdapter() {
        public void keyReleased(java.awt.event.KeyEvent evt) {
            txtBIDKeyReleased(evt);
        }
        public void keyTyped(java.awt.event.KeyEvent evt) {
            txtBIDKeyTyped(evt);
        }
    });

    lblBID.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
    lblBID.setForeground(new java.awt.Color(8, 8, 8));
    lblBID.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
    lblBID.setText("BID:");

    txtBracketLowerBound.setBackground(new java.awt.Color(252, 250, 255));
    txtBracketLowerBound.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
    txtBracketLowerBound.setForeground(new java.awt.Color(8, 8, 8));
    txtBracketLowerBound.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2,
new java.awt.Color(204, 204, 204)));
    txtBracketLowerBound.setFocusTraversalPolicyProvider(true);

```



```

txtBracketLowerBound.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtBracketLowerBound.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtBracketLowerBoundActionPerformed(evt);
    }
});
txtBracketLowerBound.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtBracketLowerBoundKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtBracketLowerBoundKeyTyped(evt);
    }
});

lblBracketLowerBound.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblBracketLowerBound.setForeground(new java.awt.Color(8, 8, 8));
lblBracketLowerBound.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblBracketLowerBound.setText("Bracket lower bound:");

kbtnClearFields.setText("Clear Fields");
kbtnClearFields.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnClearFields.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnClearFields.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnClearFields.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnClearFields.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnClearFields.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnClearFields.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnClearFieldsActionPerformed(evt);
    }
});

kbtnDeselect.setText("Deselect");
kbtnDeselect.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnDeselect.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnDeselect.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnDeselect.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnDeselect.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnDeselect.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnDeselect.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnDeselectActionPerformed(evt);
    }
}

```

```

});

kbtnUpdate.setText("Update");
kbtnUpdate.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnUpdate.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnUpdate.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnUpdate.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnUpdate.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnUpdate.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnUpdate.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnUpdateActionPerformed(evt);
    }
});

kbtnAdd.setText("Add");
kbtnAdd.setkEndColor(new java.awt.Color(28, 84, 140));
kbtnAdd.setkHoverEndColor(new java.awt.Color(76, 111, 167));
kbtnAdd.setkHoverForeGround(new java.awt.Color(252, 250, 255));
kbtnAdd.setkHoverStartColor(new java.awt.Color(76, 111, 167));
kbtnAdd.setkSelectedColor(new java.awt.Color(252, 250, 255));
kbtnAdd.setkStartColor(new java.awt.Color(28, 84, 140));
kbtnAdd.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kbtnAddActionPerformed(evt);
    }
});

lblBracketUpperBound.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblBracketUpperBound.setForeground(new java.awt.Color(8, 8, 8));
lblBracketUpperBound.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblBracketUpperBound.setText("Bracket upper bound");

txtBracketUpperBound.setBackground(new java.awt.Color(252, 250, 255));
txtBracketUpperBound.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtBracketUpperBound.setForeground(new java.awt.Color(8, 8, 8));
txtBracketUpperBound.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2,
new java.awt.Color(204, 204, 204)));
txtBracketUpperBound.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtBracketUpperBound.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtBracketUpperBoundActionPerformed(evt);
    }
});

```

```

txtBracketUpperBound.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtBracketUpperBoundKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtBracketUpperBoundKeyTyped(evt);
    }
});

lblER.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblER.setForeground(new java.awt.Color(8, 8, 8));
lblER.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblER.setText("Employer's % Share (ER)");

txtER.setBackground(new java.awt.Color(252, 250, 255));
txtER.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtER.setForeground(new java.awt.Color(8, 8, 8));
txtER.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));
txtER.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtER.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtERActionPerformed(evt);
    }
});
txtER.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtERKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtERKeyTyped(evt);
    }
});

lblEE.setFont(new java.awt.Font("Segoe UI Light", 0, 14)); // NOI18N
lblEE.setForeground(new java.awt.Color(8, 8, 8));
lblEE.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
lblEE.setText("Employee's % Share (EE)");

txtEE.setBackground(new java.awt.Color(252, 250, 255));
txtEE.setFont(new java.awt.Font("Segoe UI Light", 0, 18)); // NOI18N
txtEE.setForeground(new java.awt.Color(8, 8, 8));
txtEE.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(204, 204, 204)));

```

```

txtEE.setMargin(new java.awt.Insets(2, 2, 2, 2));
txtEE.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtEEActionPerformed(evt);
    }
});
txtEE.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtEEKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtEEKeyTyped(evt);
    }
});

javax.swing.GroupLayout pnlDelete1Layout = new javax.swing.GroupLayout(pnlDelete1);
pnlDelete1.setLayout(pnlDelete1Layout);
pnlDelete1Layout.setHorizontalGroup(
    pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlDelete1Layout.createSequentialGroup()
            .add(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .add(pnlDelete1Layout.createSequentialGroup()
                    .addContainerGap()
                    .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addGroup(pnlDelete1Layout.createSequentialGroup()
                            .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                .addGroup(pnlDelete1Layout.createSequentialGroup()
                                    .addComponent(lblEE)
                                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                    .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                        .addGroup(pnlDelete1Layout.createSequentialGroup()
                                            .addComponent(kbtnClearFields,
                                                javax.swing.GroupLayout.PREFERRED_SIZE, 67,
                                                javax.swing.GroupLayout.PREFERRED_SIZE)
                                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                            .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                                .addComponent(kbtnDeselect,
                                                    javax.swing.GroupLayout.PREFERRED_SIZE, 64,
                                                    javax.swing.GroupLayout.PREFERRED_SIZE)))
                                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                        .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                            .addGroup(pnlDelete1Layout.createSequentialGroup()
                                                .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                                    .addGroup(pnlDelete1Layout.createSequentialGroup()
                                                        .addComponent(txtEE)

```

```

        .addGroup(pnlDelete1Layout.createSequentialGroup())
        .addComponent(kbtnAdd, javax.swing.GroupLayout.PREFERRED_SIZE,
61, javax.swing.GroupLayout.PREFERRED_SIZE)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(kbtnUpdate,
javax.swing.GroupLayout.PREFERRED_SIZE, 72,
javax.swing.GroupLayout.PREFERRED_SIZE))))
        .addGroup(pnlDelete1Layout.createSequentialGroup())
        .addComponent(lblBID)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(txtBID, javax.swing.GroupLayout.PREFERRED_SIZE, 103,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, Short.MAX_VALUE))
        .addGroup(pnlDelete1Layout.createSequentialGroup())
        .addComponent(lblER)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(txtER))
        .addGroup(pnlDelete1Layout.createSequentialGroup())

    .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
G)
        .addComponent(lblBracketLowerBound)
        .addComponent(lblBracketUpperBound))
        .addGap(23, 23, 23)

    .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
G)
        .addComponent(txtBracketUpperBound)
        .addComponent(txtBracketLowerBound))))
        .addContainerGap()
    );
    pnlDelete1Layout.setVerticalGroup(
        pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlDelete1Layout.createSequentialGroup())
        .addContainerGap()

    .addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
NE)
        .addComponent(lblBID)
        .addComponent(txtBID, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

```

```

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    .addComponent(lblBracketLowerBound)
    .addComponent(txtBracketLowerBound,
javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblBracketUpperBound)
    .addComponent(txtBracketUpperBound,
javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblER)
    .addComponent(txtER, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblEE)
    .addComponent(txtEE, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(pnlDelete1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(kbtnClearFields, javax.swing.GroupLayout.PREFERRED_SIZE,
30, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(kbtnDeselect, javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(kbtnUpdate, javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(kbtnAdd, javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);

```

```

    pnlEmail1.setBackground(new java.awt.Color(238, 238, 238));
    pnlEmail1.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
    pnlEmail1.setForeground(new java.awt.Color(36, 48, 94));

    kbtnSalaryGenerator.setText("Salary Generator");
    kbtnSalaryGenerator.setkEndColor(new java.awt.Color(28, 84, 140));
    kbtnSalaryGenerator.setkHoverEndColor(new java.awt.Color(76, 111, 167));
    kbtnSalaryGenerator.setkHoverForeGround(new java.awt.Color(252, 250, 255));
    kbtnSalaryGenerator.setkHoverStartColor(new java.awt.Color(76, 111, 167));
    kbtnSalaryGenerator.setkSelectedColor(new java.awt.Color(252, 250, 255));
    kbtnSalaryGenerator.setkStartColor(new java.awt.Color(28, 84, 140));
    kbtnSalaryGenerator.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            kbtnSalaryGeneratorActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout pnlEmail1Layout = new javax.swing.GroupLayout(pnlEmail1);
    pnlEmail1.setLayout(pnlEmail1Layout);
    pnlEmail1Layout.setHorizontalGroup(
        pnlEmail1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlEmail1Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(kbtnSalaryGenerator, javax.swing.GroupLayout.DEFAULT_SIZE,
249, Short.MAX_VALUE)
                .addContainerGap())
    );
    pnlEmail1Layout.setVerticalGroup(
        pnlEmail1Layout.createSequentialGroup()
            .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
pnlEmail1Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(kbtnSalaryGenerator, javax.swing.GroupLayout.PREFERRED_SIZE,
36, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(9, Short.MAX_VALUE))
    );

    pnlLogo.setBackground(new java.awt.Color(36, 48, 94));
    pnlLogo.setBorder(javax.swing.BorderFactory.createMatteBorder(5, 5, 5, 5, new
java.awt.Color(255, 255, 255)));
    pnlLogo.setForeground(new java.awt.Color(36, 48, 94));

```

```

lblLogo.setIcon(new javax.swing.ImageIcon(getClass().getResource("/200x216
HDMF.png"))); // NOI18N
lblLogo.setText("jLabel1");

javax.swing.GroupLayout pnlLogoLayout = new javax.swing.GroupLayout(pnlLogo);
pnlLogo.setLayout(pnlLogoLayout);
pnlLogoLayout.setHorizontalGroup(
    pnlLogoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlLogoLayout.createSequentialGroup()
            .addGap(89, 89, 89)
            .addComponent(lblLogo, javax.swing.GroupLayout.PREFERRED_SIZE, 200,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );
pnlLogoLayout.setVerticalGroup(
    pnlLogoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlLogoLayout.createSequentialGroup()
            .addGap(javax.swing.GroupLayout.Alignment.TRAILING,
pnlLogoLayout.createSequentialGroup()
                .addGap(43, Short.MAX_VALUE)
                .addComponent(lblLogo)
                .addGap(41, 41, 41))
            );

javax.swing.GroupLayout pnlBackgroundLayout = new
javax.swing.GroupLayout(pnlBackground);
pnlBackground.setLayout(pnlBackgroundLayout);
pnlBackgroundLayout.setHorizontalGroup(

pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(pnlBackgroundLayout.createSequentialGroup()
        .addGap(pnlBackgroundLayout.createSequentialGroup()
            .addGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(pnlDelete, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(pnlDelete1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addGroup(pnlBackgroundLayout.createSequentialGroup()
                .addComponent(pnlEmail, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(pnlEmail1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            );

```



```

        .addComponent(pnlLogo, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addGroup(pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING, false)
        .addComponent(pnlTable, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(pnlSearch, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addContainerGap()
    );
    pnlBackgroundLayout.setVerticalGroup(

pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(pnlBackgroundLayout.createSequentialGroup()
        .addContainerGap()

    .addGroup(pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING, false)
        .addComponent(pnlEmail, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(pnlEmail1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(pnlSearch, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

    .addGroup(pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
        .addGroup(pnlBackgroundLayout.createSequentialGroup()
            .addComponent(pnlDelete1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(pnlDelete, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(pnlLogo, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addComponent(pnlTable, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );

```

```

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(pnlBackground, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(pnlBackground, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
);

```

```

pack();
setLocationRelativeTo(null);
} // </editor-fold>

```

```

private void kbbtnSortActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        String search = txtSearch.getText().trim();

        int column = cmbColumn.getSelectedIndex();

        String[] columnnames =
{"bid","bracketlowerbound","bracketupperbound","hdmf_er_p","hdmf_ee_p"};

        String strcolumn = columnnames[column];

        int sort = cmbOrder.getSelectedIndex();
        String strsort = "";
        switch (sort){
            case 0:
                strsort = "ASC";
                break;
            case 1:
                strsort = "DESC";
        }

        if (column==0 && sort==0){
            refreshtable("SELECT * FROM hdmf");
        }
        // else if (column==0 && sort==1){
        //     //Iterate through the resultset backwards
    }
}

```

```

//      DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
//      String databaseURL = "jdbc:derby:SALARY_GENERATOR";
//      Connection con = DriverManager.getConnection(databaseURL);
//      Statement st =
con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_READ_ONL
Y);
//      String sql = "SELECT * FROM info";
//      ResultSet rs = st.executeQuery(sql);
//
//      //Moves the cursor to the end of the ResultSet object
//      rs.afterLast();
//      model.setRowCount(0);
//      while(rs.previous())
//      {
//          model.addRow(new
Object[]{rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4)});
//      }
//      rs.close();
//      st.close();
//      con.close();
//  }
    else{
        refreshtable("SELECT * FROM hdmf order by "+strcolumn+" "+strsort);
    }

}
catch (Exception e){

}

}

private void cmbColumnActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void tblEmployeesMouseClicked(java.awt.event.MouseEvent evt) {
    try{
        //Prerequisite for the update function
        DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
        int selectedrow=tblEmployees.getSelectedRow();
        if (tblEmployees.getRowSorter()!=null) {
            //Gets the correct row for a filtered jTable (RowFilter)
            selectedrow = tblEmployees.getRowSorter().convertRowIndexToModel(selectedrow);
        }
    }
}

```

```

String eid = "";
String bracketlower = "";
String bracketupper = "";
String er = "";
String ee = "";

//      eid = txtEID.getText();

eid = model.getValueAt(selectedrow,0).toString();
bracketlower = model.getValueAt(selectedrow,1).toString();
bracketupper = model.getValueAt(selectedrow,2).toString();
er = model.getValueAt(selectedrow,3).toString();
ee = model.getValueAt(selectedrow,4).toString();

this.selectedroweid = eid;
txtBID.setText(eid);
txtBracketLowerBound.setText(bracketlower);
txtBracketUpperBound.setText(bracketupper);
txtER.setText(er);
txtEE.setText(ee);
}
catch (Exception e){
    e.printStackTrace();
}
}

private void cmbInPropertyChange(java.beans.PropertyChangeEvent evt) {

}

private void cmbInActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void cmbInItemStateChanged(java.awt.event.ItemEvent evt) {
    try{
        search();
    }
    catch (Exception e){

    }
}
}

```

```

private void txtSearchKeyTyped(java.awt.event.KeyEvent evt) {

}

private void txtSearchKeyReleased(java.awt.event.KeyEvent evt) {
    try{
        search();
    }
    catch (Exception e){

    }
}

private void txtSearchActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void kbtnResetActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        //Resets the search and sort functions; restores table to original
        refreshtable("SELECT * FROM hdmf");
        cmbColumn.setSelectedIndex(0);
        cmbOrder.setSelectedIndex(0);

        txtSearch.setText("");
        cmbIn.setSelectedIndex(0);

        search();
    }
    catch (Exception e){

    }
}

private void kbtnSalaryGeneratorActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        new SalaryGenerator(this.uid,this.email).setVisible(true);
        this.setVisible(false);
    }
    catch (Exception e){

    }
}

```

```

private void kbtnDeleteAllActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        int confirm = JOptionPane.showConfirmDialog(null,"Are you want to delete ALL
records?");
        if (confirm==0){
            DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
            int rowcount = model.getRowCount(); //number of rows in model

            if (rowcount==0){
                JOptionPane.showMessageDialog(null, "The table is empty.");
            }
            else{
                int eids[] = new int[rowcount];
                for (int i = 0; i<rowcount; i++){
                    eids[i] = Integer.parseInt(model.getValueAt(i,0).toString());
                }
                String databaseURL = "jdbc:derby:SALARY_GENERATOR";
                Connection con = DriverManager.getConnection(databaseURL);

                String sql;
                for (int i = 0; i<eids.length; i++){
                    //Deletion based on eid
                    Statement st = con.createStatement();
                    sql = "DELETE FROM hdmf WHERE bid=" + eids[i];
                    st.execute(sql);
                }

                maxfinder();

                model.setRowCount(0);
            }
        }
    }
    catch(Exception e){

    }
}

private void kbtnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();

```

```

int rowcount = model.getRowCount();

//Allows it to work with the filtered jtable (RowFilter)
int rows[] = tblEmployees.getSelectedRows();
for (int i = 0; i<rows.length; i++){
    rows[i] = tblEmployees.getRowSorter().convertRowIndexToModel(rows[i]);
}

if ("-1".equals(this.selectedroweid)){
    JOptionPane.showMessageDialog(null, "A row has not been selected.");
}
else{
    int confirm = JOptionPane.showConfirmDialog(null,"Are you sure?");

    if (confirm==0){
        if (rows.length>0){//Delete multiple or single
            int eids[] = new int[rows.length];
            for (int i = 0; i<rows.length; i++){
                eids[i] = Integer.parseInt(model.getValueAt(rows[i],0).toString());
            }

            //Establish connection to database
            String databaseURL = "jdbc:derby:SALARY_GENERATOR";
            Connection con = DriverManager.getConnection(databaseURL);

            //Delete statement to remove the specific row that lines up with the selected eid
            String sql;
            for (int i = 0; i<eids.length; i++){
                Statement st = con.createStatement();

                sql = "DELETE FROM hdmf WHERE bid=" + eids[i];
                st.execute(sql);
            }

            //            txtFullName.setText("");
            //            txtEmail.setText("");
            //            txtPhoneNumber.setText("");

            this.selectedroweid = "-1";
            refreshtable("SELECT * FROM hdmf");

            maxfinder();
        }
    }
}

```

```

    }

    }
    catch (Exception e){

    }
}

private void kbtnAddActionPerformed(java.awt.event.ActionEvent evt) {
    try{
//      String eid = "";
      String bracketlower = "";
      String bracketupper = "";
      String er = "";
      String ee = "";

//      eid = txtEID.getText();

      bracketlower = txtBracketLowerBound.getText();
      bracketupper = txtBracketUpperBound.getText();
      er = txtER.getText();
      ee = txtEE.getText();

      System.out.println("Email: " + email);

      //Error trapping
      if ("".equals(bracketlower)){
          JOptionPane.showMessageDialog(null, "Lower bound cannot be empty.");
      }
      else if ("".equals(bracketupper)){
          JOptionPane.showMessageDialog(null, "Upper bound cannot be empty.");
      }
      else if ("".equals(er)){
          JOptionPane.showMessageDialog(null, "Employer Share cannot be empty.");
      }
      else if ("".equals(ee)){
          JOptionPane.showMessageDialog(null, "Employee Share cannot be empty.");
      }
      else if (Double.parseDouble(bracketlower)>999999999 ||
Double.parseDouble(bracketlower)!=(double)Math.round(Double.parseDouble(bracketlower)*10
0)/100){
          JOptionPane.showMessageDialog(null, "Bracket Lower Bound can be a maximum of
99,999,999.99. Decimal precision is only 2 decimal places.");
      }
    }
}

```



```

        else if (Double.parseDouble(bracketupper)>999999999 ||
Double.parseDouble(bracketupper)!=((double)Math.round(Double.parseDouble(bracketupper)*1
00)/100){
            JOptionPane.showMessageDialog(null, "Bracket Upper Bound can be a maximum of
99,999,999.99. Decimal precision is only 2 decimal places.");
        }
        else if (Double.parseDouble(er)>999999999 ||
Double.parseDouble(er)!=((double)Math.round(Double.parseDouble(er)*100)/100){
            JOptionPane.showMessageDialog(null, "Employer % Share (ER) can be a maximum
of 99,999,999.99. Decimal precision is only 2 decimal places.");
        }
        else if (Double.parseDouble(ee)>999999999 ||
Double.parseDouble(ee)!=((double)Math.round(Double.parseDouble(ee)*100)/100){
            JOptionPane.showMessageDialog(null, "Employee % Share (EE) can be a maximum
of 99,999,999.99. Decimal precision is only 2 decimal places.");
        }
        else{
            maxfinder();
            this.max++;
            String strmax = Integer.toString(this.max);

            System.out.println("strmax: " + strmax);

            String databaseURL = "jdbc:derby:SALARY_GENERATOR";
            Connection con = DriverManager.getConnection(databaseURL);
            Statement st = con.createStatement();

            String sql = "INSERT INTO hdmf (bid, bracketlowerbound, bracketupperbound,
hdmf_er_p, hdmf_ee_p) "
                + "VALUES (" +strmax+", "+bracketlower+", "+bracketupper+", "+er+", "+ee+)";

            //pay_period determined by the max from salary and loan history

            st.execute(sql);
            con.close();

            refreshtable("SELECT * FROM hdmf");

            txtBID.setText(Integer.toString(max));
            txtBracketLowerBound.setText("");
            txtBracketUpperBound.setText("");
            txtER.setText("");
            txtEE.setText("");

```

```

        JOptionPane.showMessageDialog(null, "Added.");
    }
}
catch (Exception e){
    e.printStackTrace();
}
}

private void kbtnUpdateActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        //Prerequisite for the update function
        String bid = "";
        String bracketlower = "";
        String bracketupper = "";
        String er = "";
        String ee = "";

//        eid = txtEID.getText();

        bid = txtBID.getText();
        bracketlower = txtBracketLowerBound.getText();
        bracketupper = txtBracketUpperBound.getText();
        er = txtER.getText();
        ee = txtEE.getText();

        System.out.println("Email: " + email);

        //Error trapping
        if (Double.parseDouble(bracketlower)>999999999 ||
Double.parseDouble(bracketlower)!=((double)Math.round(Double.parseDouble(bracketlower)*10
0)/100)){
            JOptionPane.showMessageDialog(null, "Bracket Lower Bound can be a maximum of
99,999,999.99. Decimal precision is only 2 decimal places.");
        }
        else if (Double.parseDouble(bracketupper)>999999999 ||
Double.parseDouble(bracketupper)!=((double)Math.round(Double.parseDouble(bracketupper)*1
00)/100)){
            JOptionPane.showMessageDialog(null, "Bracket Upper Bound can be a maximum of
99,999,999.99. Decimal precision is only 2 decimal places.");
        }
        else if (Double.parseDouble(er)>999999999 ||
Double.parseDouble(er)!=((double)Math.round(Double.parseDouble(er)*100)/100)){
            JOptionPane.showMessageDialog(null, "Employer % Share (ER) can be a maximum
of 99,999,999.99. Decimal precision is only 2 decimal places.");
        }
    }
}

```

```

    }
    else if (Double.parseDouble(ee)>99999999 ||
Double.parseDouble(ee)!=((double)Math.round(Double.parseDouble(ee)*100)/100){
        JOptionPane.showMessageDialog(null, "Employee % Share (EE) can be a maximum
of 99,999,999.99. Decimal precision is only 2 decimal places.");
    }
    else{
        String strmax = Integer.toString(max);

        String databaseURL = "jdbc:derby:SALARY_GENERATOR";
        Connection con = DriverManager.getConnection(databaseURL);
        Statement st = con.createStatement();

        String sql = "UPDATE hdmf SET bid="+bid+", bracketlowerbound="+bracketlower+",
bracketupperbound="+bracketupper+", hdmf_ee_p="+ee+", hdmf_er_p="+er+" WHERE
bid="+bid;
        st.execute(sql);
        con.close();

        //No row currently selected.
        selectedroweid = "-1";

        JOptionPane.showMessageDialog(null, "Updated.");
        tblEmployees.clearSelection();

        refreshtable("SELECT * FROM hdmf");
        maxfinder();

        int max = this.max;
        max++;

        txtBID.setText(Integer.toString(max));
        txtBracketLowerBound.setText("");
        txtBracketUpperBound.setText("");
        txtER.setText("");
        txtEE.setText("");

    }
}
catch (Exception e){
    e.printStackTrace();
}
}

```

```

private void kbtnDeselectActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        //Deselects all rows
        //      DefaultTableModel model = (DefaultTableModel) tblEmployees.getModel();
        //      tblEmployees.removeRowSelectionInterval(0, model.getRowCount()-1);
        tblEmployees.clearSelection();
        //resets selectedroweid
        this.selectedroweid = "-1";
    }
    catch (Exception e){

    }
}

private void kbtnClearFieldsActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        txtBID.setText("");
        txtBracketLowerBound.setText("");
        txtBracketUpperBound.setText("");
        txtER.setText("");
        txtEE.setText("");
    }
    catch(Exception e){

    }
}

private void txtBracketLowerBoundKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtBracketLowerBoundKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtBracketLowerBoundActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtBIDKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtBIDKeyReleased(java.awt.event.KeyEvent evt) {

```

```

    // TODO add your handling code here:
}

private void txtBIDActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtBracketUpperBoundActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtBracketUpperBoundKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtBracketUpperBoundKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtERActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtERKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtERKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtEEActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtEEKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void txtEEKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

/**

```

```

* @param args the command line arguments
*/
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
    * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Register().setVisible(true);
        }
    });
}

// Variables declaration - do not modify

```

```

private javax.swing.JComboBox<String> cmbColumn;
private javax.swing.JComboBox<String> cmbIn;
private javax.swing.JComboBox<String> cmbOrder;
private javax.swing.JScrollPane jScrollPane1;
private com.k33ptoo.components.KButton btnAdd;
private com.k33ptoo.components.KButton btnClearFields;
private com.k33ptoo.components.KButton btnDelete;
private com.k33ptoo.components.KButton btnDeleteAll;
private com.k33ptoo.components.KButton btnDeselect;
private com.k33ptoo.components.KButton btnReset;
private com.k33ptoo.components.KButton btnSalaryGenerator;
private com.k33ptoo.components.KButton btnSort;
private com.k33ptoo.components.KButton btnUpdate;
private javax.swing.JLabel lblID;
private javax.swing.JLabel lblBracketLowerBound;
private javax.swing.JLabel lblBracketUpperBound;
private javax.swing.JLabel lblEE;
private javax.swing.JLabel lblER;
private javax.swing.JLabel lblEditPersonalRecords;
private javax.swing.JLabel lblLogo;
private javax.swing.JLabel lblSearch;
private com.k33ptoo.components.KGradientPanel pnlBackground;
private javax.swing.JPanel pnlDelete;
private javax.swing.JPanel pnlDelete1;
private javax.swing.JPanel pnlEmail;
private javax.swing.JPanel pnlEmail1;
private javax.swing.JPanel pnlLogo;
private javax.swing.JPanel pnlSearch;
private javax.swing.JPanel pnlTable;
private javax.swing.JTable tblEmployees;
private javax.swing.JTextField txtID;
private javax.swing.JTextField txtBracketLowerBound;
private javax.swing.JTextField txtBracketUpperBound;
private javax.swing.JTextField txtEE;
private javax.swing.JTextField txtER;
private javax.swing.JLabel txtIn;
private javax.swing.JTextField txtSearch;
private javax.swing.JLabel txtSort;
// End of variables declaration
}

```

EmailSender.java

```
/**
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
 license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */

/**
 *
 * @author tcoop
 */

import java.util.Properties;
import javax.mail.*;
import javax.mail.internet.*;
import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManager;
import javax.net.ssl.X509TrustManager;
import java.security.cert.X509Certificate;

public class EmailSender {
    // Sender's email credentials
    private final String senderEmail = "ibcsiaemailsender@gmail.com";
    private final String senderPassword = "oxmy zpqr etqw dcpo";

    // Gmail SMTP server details
    private final String emailSMTPserver = "smtp.gmail.com";

    // Constructor for sending an email
    public EmailSender(String receiverEmail, String subject, String body) {
        try {
            // Trust all certificates (for testing purposes only)
            TrustManager[] trustAllCerts = new TrustManager[]{
                new X509TrustManager() {
                    public java.security.cert.X509Certificate[] getAcceptedIssuers() {
                        return null;
                    }
                }
            };
            public void checkClientTrusted(X509Certificate[] certs, String authType) {}
            public void checkServerTrusted(X509Certificate[] certs, String authType) {}
        }
    };
    SSLContext sc = SSLContext.getInstance("TLS");
```



```

sc.init(null, trustAllCerts, new java.security.SecureRandom());
javax.net.ssl.HttpURLConnection.setDefaultSSLSocketFactory(sc.getSocketFactory());

// Define SMTP server properties
Properties props = new Properties();
props.put("mail.smtp.host", emailSMTPserver);
props.put("mail.smtp.port", "465");
props.put("mail.smtp.auth", "true");
props.put("mail.smtp.socketFactory.port", "465");
props.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");
props.put("mail.smtp.socketFactory.fallback", "false");
props.put("mail.smtp.ssl.trust", "smtp.gmail.com");

// Create an Authenticator with the sender's credentials
Authenticator auth = new Authenticator() {
    @Override
    protected PasswordAuthentication getPasswordAuthentication() {
        return new PasswordAuthentication(senderEmail, senderPassword);
    }
};

// Create a new email session with the properties and authenticator
Session session = Session.getInstance(props, auth);
session.setDebug(true); // Enable debug mode to see detailed logs

// Create a new email message
MimeMessage msg = new MimeMessage(session);
msg.setFrom(new InternetAddress(senderEmail));
msg.addRecipient(Message.RecipientType.TO, new InternetAddress(receiverEmail));
msg.setSubject(subject);
msg.setText(body);

// Send the email
Transport.send(msg);
} catch (Exception e) {
    // Handle exceptions and print the stack trace
    e.printStackTrace();
}
}

// Main method to test the email-sending functionality
public static void main(String[] args) {
    // Replace with the receiver's email, subject, and body
    String receiverEmail = "receiver@example.com";

```

```
String subject = "Test Subject (Port 465 with SSL/TLS)";  
String body = "This is a test email sent using JavaMail via port 465 with implicit SSL/TLS.";  
  
// Create a new EmailSender instance to send the email  
new EmailSender(receiverEmail, subject, body);  
}  
}
```

NumericDocumentFilter.java

```
import javax.swing.text.AttributeSet;
import javax.swing.text.BadLocationException;
import javax.swing.text.DocumentFilter;

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */

/**
 * A DocumentFilter that allows only numeric input with an optional decimal point.
 */
public class NumericDocumentFilter extends DocumentFilter {

    // Inserts text only if it results in valid numeric input.
    @Override
    public void insertString(DocumentFilter.FilterBypass fb, int offset, String string, AttributeSet
attr) throws BadLocationException {
        if (isValidInput(fb.getDocument().getText(0, fb.getDocument().getLength()), string)) {
            super.insertString(fb, offset, string, attr);
        }
    }

    // Replaces text only if the new input remains numeric.
    @Override
    public void replace(DocumentFilter.FilterBypass fb, int offset, int length, String text,
AttributeSet attrs) throws BadLocationException {
        if (isValidInput(fb.getDocument().getText(0, fb.getDocument().getLength()), text)) {
            super.replace(fb, offset, length, text, attrs);
        }
    }

    // Allows removal of text without restrictions.
    @Override
    public void remove(DocumentFilter.FilterBypass fb, int offset, int length) throws
BadLocationException {
        super.remove(fb, offset, length);
    }

    // Validates if the new input maintains a proper numeric format.
    private boolean isValidInput(String currentText, String newText) {
```

```
String resultingText = currentText + newText;  
return resultingText.isEmpty() || resultingText.matches("\\d*\\.?\\d*"); // Allows digits and one  
decimal point  
}  
}
```

IntegerDocumentFilter.java

```
import javax.swing.text.AttributeSet;
import javax.swing.text.BadLocationException;
import javax.swing.text.DocumentFilter;

/**
 * A DocumentFilter implementation that allows only integer inputs in a text field.
 */
public class IntegerDocumentFilter extends DocumentFilter {

    @Override
    public void insertString(DocumentFilter.FilterBypass fb, int offset, String string, AttributeSet
attr) throws BadLocationException {
        if (isValidInput(fb.getDocument().getText(0, fb.getDocument().getLength()), string)) {
            super.insertString(fb, offset, string, attr);
        }
    }

    @Override
    public void replace(DocumentFilter.FilterBypass fb, int offset, int length, String text,
AttributeSet attrs) throws BadLocationException {
        if (isValidInput(fb.getDocument().getText(0, fb.getDocument().getLength()), text)) {
            super.replace(fb, offset, length, text, attrs);
        }
    }

    @Override
    public void remove(DocumentFilter.FilterBypass fb, int offset, int length) throws
BadLocationException {
        super.remove(fb, offset, length);
    }

    /**
     * Validates that the resulting text contains only integer values.
     *
     * @param currentText The current text in the document.
     * @param newText The new text being added to the document.
     * @return True if the resulting text is valid; false otherwise.
     */
    private boolean isValidInput(String currentText, String newText) {
        String resultingText = currentText.substring(0, currentText.length()) + newText;
        if (resultingText.isEmpty()) {
            return true; // Allow empty text (useful for clearing the field).
        }
    }
}
```

```
    }  
    return resultingText.matches("\\d*"); // Validates input containing only digits.  
  }  
}
```