

Code

Contents

Code	1
Application GUI:	1
Profile GUI:	108
Start Up GUI:	129
Pupil Object Class:	140
MailMerge Class:	145
SQL Driver Class:	146

Application GUI:

//Imports use for various methods in the program

```
import java.awt.event.KeyAdapter;
```

```
import java.awt.event.KeyEvent;
```

```
import javax.swing.ImageIcon;
```

```
import java.sql.*;
```

```
import java.util.logging.*;
```

```
import java.util.Locale;
```

```
import java.util.Date;
```

```
import javax.swing.table.DefaultTableModel;
```

```
import javax.swing.event.*;
```

```
import java.text.*;
```

```
import java.awt.Desktop;
```

```
import java.io.File;
```

```
import java.io.IOException;
```

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

```
    //Initialise some global variables which will be used throughout the program for various finctions
```

```
    static public Pupil people[] = new Pupil[999];
```

```

static int index = 0;

String search = "";

boolean gen;

boolean hand;

int flag = 0;

int flg = 0;

int vflag = 0;

ImageIcon img = new ImageIcon(getClass().getResource("MCA.PNG"));

int gauge = 0;

```

```

public Application() throws IndexOutOfBoundsException {

    initComponents();

    //The function used to refresh the program

    reboot();

    //The code used to get the selected row when a user clicks on a row using the jTable. This code
    then also opens a JFrame

    //relevant to that selected row

    mainTable.getSelectionModel().addListSelectionListener(new ListSelectionListener() {

        @Override

        public void valueChanged(ListSelectionEvent event) throws IndexOutOfBoundsException {

            try {

                Profile.param = mainTable.getValueAt(mainTable.getSelectedRow(), 0).toString();

                profileStart();

                reboot();

            } catch (IndexOutOfBoundsException e) {

            }

        }

    });

    //The code used to get the selected when a user clicks on a row in the jTable for documents.
    This code then also calls the

    //function that opens that document

```

```

docAccessTable.getSelectionModel().addListSelectionListener(new ListSelectionListener() {
    @Override
    public void valueChanged(ListSelectionEvent event) throws IndexOutOfBoundsException {
        try {
            String file = docAccessTable.getValueAt(docAccessTable.getSelectedRow(), 0).toString();
            openDoc(file);

        } catch (IndexOutOfBoundsException e) {
        }
    }
});

//Setting some fields and buttons to behave in certain ways or adding ToolTipTexts
mainTable.setToolTipText("Click any of the column headers to sort this table");
sessTable.setToolTipText("Click any of the column headers to sort this table");
searchButton.setToolTipText("Search the chosen parameter.");
cancelButton.setToolTipText("Deselect the chosen parameter and erase the field");
mainBoot.setToolTipText("Refresh or boot the system");
addSaveButton.setToolTipText("Create the new pupil");
addResetButton.setToolTipText("Clear all the data above");
removeButton.setToolTipText("Remove the pupils");
sessSureButton.setToolTipText("Send the automated email to the pupil");
sessSaveButton.setToolTipText("Save the above data and generate an automated email");
formSaveButton.setToolTipText("Save the above data in an accessible form");
fileUploadField.setToolTipText("Upload this file to the database");
infoNPassField.setVisible(false);
infoCPassField.setVisible(false);
infoNPassLabel.setVisible(false);
infoCPassLabel.setVisible(false);
infoNPassButton.setVisible(false);
sessSendLabel.setVisible(false);
sessSendField.setVisible(false);

```

```
sessSureButton.setVisible(false);  
sessCancelButton.setVisible(false);  
}
```

```
@SuppressWarnings("unchecked")
```

```
// <editor-fold defaultstate="collapsed" desc="Generated Code">
```

```
private void initComponents() {  
  
    jLabel14 = new javax.swing.JLabel();  
    jTextField10 = new javax.swing.JTextField();  
    jButton5 = new javax.swing.JButton();  
    jScrollPane10 = new javax.swing.JScrollPane();  
    jTextArea4 = new javax.swing.JTextArea();  
    jLabel9 = new javax.swing.JLabel();  
    jTextField8 = new javax.swing.JTextField();  
    tableScrollPane = new javax.swing.JScrollPane();  
    mainTable = new javax.swing.JTable();  
    docsPane = new javax.swing.JTabbedPane();  
    mainTab = new javax.swing.JPanel();  
    searchLabel = new javax.swing.JLabel();  
    searchTextField = new javax.swing.JTextField();  
    searchButton = new javax.swing.JButton();  
    jLabel1 = new javax.swing.JLabel();  
    jScrollPane5 = new javax.swing.JScrollPane();  
    sessTable = new javax.swing.JTable();  
    mainBoot = new javax.swing.JButton();  
    nameToggleSearch = new javax.swing.JToggleButton();  
    ageToggleSearch = new javax.swing.JToggleButton();  
    genToggleSearch = new javax.swing.JToggleButton();  
    emailToggleSearch = new javax.swing.JToggleButton();  
    numToggleSearch = new javax.swing.JToggleButton();  
}
```

```
cancelButton = new javax.swing.JButton();
editorTab = new javax.swing.JPanel();
subEditorTab = new javax.swing.JTabbedPane();
addTab = new javax.swing.JPanel();
starLabel = new javax.swing.JLabel();
addNameLabel = new javax.swing.JLabel();
addNameField = new javax.swing.JTextField();
addDOBLLabel = new javax.swing.JLabel();
addDOBField = new javax.swing.JTextField();
addGenLabel = new javax.swing.JLabel();
maleToggleButton = new javax.swing.JToggleButton();
femaleToggleButton = new javax.swing.JToggleButton();
addEmailLabel = new javax.swing.JLabel();
addEmailField = new javax.swing.JTextField();
addPhoNumLabel = new javax.swing.JLabel();
addEmeNumLabel = new javax.swing.JLabel();
addPhoNumField = new javax.swing.JTextField();
addEmerNumField = new javax.swing.JTextField();
addSessDateLabel = new javax.swing.JLabel();
addSessDateField = new javax.swing.JTextField();
addSessInfoLabel = new javax.swing.JLabel();
jScrollPane2 = new javax.swing.JScrollPane();
addSessInfoField = new javax.swing.JTextArea();
addExpLabel = new javax.swing.JLabel();
jScrollPane3 = new javax.swing.JScrollPane();
addExpField = new javax.swing.JTextArea();
addMedLabel = new javax.swing.JLabel();
jScrollPane4 = new javax.swing.JScrollPane();
addMedField = new javax.swing.JTextArea();
addBatLabel = new javax.swing.JLabel();
addBatLToggle = new javax.swing.JToggleButton();
```

```
addBatRToggle = new javax.swing.JToggleButton();
addSaveButton = new javax.swing.JButton();
addResetButton = new javax.swing.JButton();
addTimeLabel = new javax.swing.JLabel();
jScrollPane6 = new javax.swing.JScrollPane();
addPayField = new javax.swing.JTextArea();
addPayLabel = new javax.swing.JLabel();
addPaySessLabel = new javax.swing.JLabel();
addPaySessField = new javax.swing.JTextField();
addTimeBox = new javax.swing.JComboBox<>();
removeTab = new javax.swing.JPanel();
removeLabel = new javax.swing.JLabel();
removeField = new javax.swing.JTextField();
removeButton = new javax.swing.JButton();
sessionTab = new javax.swing.JPanel();
sessNameLabel = new javax.swing.JLabel();
sessNameField = new javax.swing.JTextField();
sessEnterButton = new javax.swing.JButton();
sessEmailLabel = new javax.swing.JLabel();
sessEmailField = new javax.swing.JTextField();
sessInfoLabel = new javax.swing.JLabel();
jScrollPane12 = new javax.swing.JScrollPane();
sessInfoField = new javax.swing.JTextArea();
sessSaveButton = new javax.swing.JButton();
sessSendLabel = new javax.swing.JLabel();
jScrollPane13 = new javax.swing.JScrollPane();
sessSendField = new javax.swing.JTextArea();
sessSureButton = new javax.swing.JButton();
sessCancelButton = new javax.swing.JButton();
sessDateLabel = new javax.swing.JLabel();
sessDateField = new javax.swing.JTextField();
```

```
sessTimeLabel = new javax.swing.JLabel();
sessPayLabel = new javax.swing.JLabel();
sessPayField = new javax.swing.JTextField();
sessTimeBox = new javax.swing.JComboBox<>();
formTab = new javax.swing.JPanel();
jTabbedPane1 = new javax.swing.JTabbedPane();
jPanel2 = new javax.swing.JPanel();
jLabel3 = new javax.swing.JLabel();
formNameField = new javax.swing.JTextField();
jLabel4 = new javax.swing.JLabel();
formDateField = new javax.swing.JTextField();
jLabel5 = new javax.swing.JLabel();
formNameButton = new javax.swing.JButton();
jLabel6 = new javax.swing.JLabel();
formEmailField = new javax.swing.JTextField();
jLabel7 = new javax.swing.JLabel();
formPhoField = new javax.swing.JTextField();
jLabel8 = new javax.swing.JLabel();
formEmerField = new javax.swing.JTextField();
jLabel10 = new javax.swing.JLabel();
jScrollPane1 = new javax.swing.JScrollPane();
formMedField = new javax.swing.JTextArea();
jLabel11 = new javax.swing.JLabel();
jScrollPane7 = new javax.swing.JScrollPane();
formInjuryField = new javax.swing.JTextArea();
formSaveButton = new javax.swing.JButton();
formCancelButton = new javax.swing.JButton();
formTimeBox = new javax.swing.JComboBox<>();
jPanel1 = new javax.swing.JPanel();
jLabel2 = new javax.swing.JLabel();
accessNameField = new javax.swing.JTextField();
```

```
formsEnterButton = new javax.swing.JButton();
jScrollPane8 = new javax.swing.JScrollPane();
accessFormField = new javax.swing.JTextArea();
accessClearButton = new javax.swing.JButton();
jPanel3 = new javax.swing.JPanel();
jTabbedPane2 = new javax.swing.JTabbedPane();
jPanel4 = new javax.swing.JPanel();
jLabel12 = new javax.swing.JLabel();
docNameField = new javax.swing.JTextField();
jLabel13 = new javax.swing.JLabel();
docFileField = new javax.swing.JTextField();
jLabel15 = new javax.swing.JLabel();
docPathField = new javax.swing.JTextField();
fileUploadField = new javax.swing.JButton();
docCancelButton = new javax.swing.JButton();
jPanel5 = new javax.swing.JPanel();
jLabel16 = new javax.swing.JLabel();
docAccessField = new javax.swing.JTextField();
fileAccessField = new javax.swing.JButton();
jScrollPane9 = new javax.swing.JScrollPane();
docAccessTable = new javax.swing.JTable();
docClearButton = new javax.swing.JButton();
infoTab = new javax.swing.JPanel();
infoNameLabel = new javax.swing.JLabel();
infoDateLabel = new javax.swing.JLabel();
infoIncomeLabel = new javax.swing.JLabel();
infoCEMailLabel = new javax.swing.JLabel();
infoEMailLabel = new javax.swing.JLabel();
infoNumLabel = new javax.swing.JLabel();
infoPassLabel = new javax.swing.JLabel();
infoPassField = new javax.swing.JPasswordField();
```



```
infoPassButton = new javax.swing.JButton();
infoNPassLabel = new javax.swing.JLabel();
infoNPassField = new javax.swing.JPasswordField();
infoCPassLabel = new javax.swing.JLabel();
infoCPassField = new javax.swing.JPasswordField();
infoNPassButton = new javax.swing.JButton();
infoUserLabel = new javax.swing.JLabel();
infoUserField = new javax.swing.JTextField();
infoUserButton = new javax.swing.JButton();
infoSecurityCombo = new javax.swing.JComboBox<>();
infoSecurityLabel = new javax.swing.JLabel();
infoSecurityField = new javax.swing.JTextField();
infoSecurityButton = new javax.swing.JButton();

jLabel14.setText("Enter the name of the pupil you wish to remove:");

jButton5.setText("Remove");

jTextArea4.setColumns(20);
jTextArea4.setRows(5);
jScrollPane10.setViewportView(jTextArea4);

jLabel9.setText("Pupil Emergency Number:");

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle("MasterClass Cricket Academy Pupil Database");
setBackground(new java.awt.Color(255, 255, 204));
setForeground(new java.awt.Color(255, 255, 204));
setIconImage(img.getImage());
setResizable(false);
```

```

mainTable.setAutoCreateRowSorter(true);

mainTable.setBackground(new java.awt.Color(204, 204, 255));

mainTable.setForeground(new java.awt.Color(0, 0, 102));

mainTable.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {

        },
    new String [] {
        "Name ", "Age", "Gender", "Next Session", "Email", "Number"
    }
) {
    Class[] types = new Class [] {
        java.lang.String.class, java.lang.Integer.class, java.lang.String.class, java.lang.String.class,
        java.lang.String.class, java.lang.Object.class
    };
    boolean[] canEdit = new boolean [] {
        false, false, false, false, false, false
    };

    public Class getColumnClass(int columnIndex) {
        return types [columnIndex];
    }

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});

mainTable.setGridColor(new java.awt.Color(204, 204, 255));

mainTable.setSelectionForeground(new java.awt.Color(0, 0, 102));

mainTable.getTableHeader().setReorderingAllowed(false);

tableScrollPane.setViewportView(mainTable);

```

```

if (mainTable.getColumnModel().getColumnCount() > 0) {
    mainTable.getColumnModel().getColumn(0).setResizable(false);
    mainTable.getColumnModel().getColumn(1).setResizable(false);
    mainTable.getColumnModel().getColumn(1).setPreferredWidth(5);
    mainTable.getColumnModel().getColumn(2).setResizable(false);
    mainTable.getColumnModel().getColumn(2).setPreferredWidth(25);
    mainTable.getColumnModel().getColumn(3).setResizable(false);
    mainTable.getColumnModel().getColumn(3).setPreferredWidth(45);
    mainTable.getColumnModel().getColumn(4).setResizable(false);
    mainTable.getColumnModel().getColumn(5).setResizable(false);
    mainTable.getColumnModel().getColumn(5).setPreferredWidth(50);
}

docsPane.setBackground(new java.awt.Color(255, 255, 204));
docsPane.setForeground(new java.awt.Color(0, 0, 102));

mainTab.setBackground(java.awt.SystemColor.info);
mainTab.setForeground(new java.awt.Color(0, 0, 102));

searchLabel.setForeground(new java.awt.Color(0, 0, 102));
searchLabel.setText("Choose the parameter that you would like to search for:");

searchTextField.setForeground(new java.awt.Color(0, 0, 102));

searchButton.setForeground(new java.awt.Color(0, 0, 102));
searchButton.setText("Enter");
searchButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        searchButtonActionPerformed(evt);
    }
});

```

```

jLabel1.setForeground(new java.awt.Color(0, 0, 102));
jLabel1.setText("Upcoming sessions:");

sessTable.setAutoCreateRowSorter(true);
sessTable.setBackground(new java.awt.Color(204, 204, 255));
sessTable.setForeground(new java.awt.Color(0, 0, 102));
sessTable.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {

    },
    new String [] {
        "Name", "Date", "Time", "Pay", "Number"
    }
) {
    boolean[] canEdit = new boolean [] {
        false, false, false, false, false
    };

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});

sessTable.getTableHeader().setReorderingAllowed(false);
jScrollPane5.setViewportViewView(sessTable);
if (sessTable.getColumnModel().getColumnCount() > 0) {
    sessTable.getColumnModel().getColumn(0).setResizable(false);
    sessTable.getColumnModel().getColumn(1).setResizable(false);
    sessTable.getColumnModel().getColumn(2).setResizable(false);
    sessTable.getColumnModel().getColumn(3).setResizable(false);
    sessTable.getColumnModel().getColumn(4).setResizable(false);
}

```

```
}
```

```
mainBoot.setForeground(new java.awt.Color(0, 0, 102));  
mainBoot.setText("Refresh System");  
mainBoot.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        mainBootActionPerformed(evt);  
    }  
});
```

```
nameToggleSearch.setForeground(new java.awt.Color(0, 0, 102));  
nameToggleSearch.setText("Name");  
nameToggleSearch.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        nameToggleSearchActionPerformed(evt);  
    }  
});
```

```
ageToggleSearch.setForeground(new java.awt.Color(0, 0, 102));  
ageToggleSearch.setText("Age");  
ageToggleSearch.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        ageToggleSearchActionPerformed(evt);  
    }  
});
```

```
genToggleSearch.setForeground(new java.awt.Color(0, 0, 102));  
genToggleSearch.setText("Gender");  
genToggleSearch.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        genToggleSearchActionPerformed(evt);  
    }  
});
```

```
    }  
});
```

```
emailToggleSearch.setForeground(new java.awt.Color(0, 0, 102));  
emailToggleSearch.setText("Email");  
emailToggleSearch.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        emailToggleSearchActionPerformed(evt);  
    }  
});
```

```
numToggleSearch.setForeground(new java.awt.Color(0, 0, 102));  
numToggleSearch.setText("Phone Number");  
numToggleSearch.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        numToggleSearchActionPerformed(evt);  
    }  
});
```

```
cancelButton.setForeground(new java.awt.Color(0, 0, 102));  
cancelButton.setText("Cancel");  
cancelButton.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        cancelButtonActionPerformed(evt);  
    }  
});
```

```
javax.swing.GroupLayout mainTabLayout = new javax.swing.GroupLayout(mainTab);  
mainTab.setLayout(mainTabLayout);  
mainTabLayout.setHorizontalGroup(  
    mainTabLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```

        .addGroup(mainTabLayout.createSequentialGroup())
        .addContainerGap()

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

    .addGroup(mainTabLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
mainTabLayout.createSequentialGroup())

        .addComponent(nameToggleSearch)
        .addGap(18, 18, 18)
        .addComponent(ageToggleSearch)
        .addGap(18, 18, 18)
        .addComponent(genToggleSearch)

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

        .addComponent(mainBoot, javax.swing.GroupLayout.PREFERRED_SIZE, 120,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(mainTabLayout.createSequentialGroup())

    .addGroup(mainTabLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jLabel1)

    .addGroup(mainTabLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
        .addComponent(searchTextField,
javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(searchLabel, javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)))
        .addGap(0, 0, Short.MAX_VALUE)))
        .addContainerGap())

    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
mainTabLayout.createSequentialGroup())

        .addGap(0, 267, Short.MAX_VALUE)
        .addComponent(cancelButton)

```

```

        .addGap(18, 18, 18)

        .addComponent(searchButton)

        .addGap(97, 97, 97))

    .addGroup(mainTabLayout.createSequentialGroup())

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

        .addGroup(mainTabLayout.createSequentialGroup()

            .addComponent(emailToggleSearch)

            .addGap(18, 18, 18)

            .addComponent(numToggleSearch))

            .addComponent(jScrollPane5, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

            .addGap(0, 0, Short.MAX_VALUE))))

    );

    mainTabLayout.setVerticalGroup(

        mainTabLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(mainTabLayout.createSequentialGroup()

            .addContainerGap()

            .addComponent(searchLabel)

            .addGap(18, 18, 18)

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

                .addComponent(mainBoot)

                .addComponent(nameToggleSearch)

                .addComponent(ageToggleSearch)

                .addComponent(genToggleSearch))

            .addGap(18, 18, 18)

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

                .addComponent(emailToggleSearch)

                .addComponent(numToggleSearch)

                .addGap(18, 18, 18)

```



```

        .addComponent(searchTextField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(mainTabLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(searchButton)
        .addComponent(cancelButton))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(jLabel1)
        .addGap(18, 18, 18)
        .addComponent(jScrollPane5, javax.swing.GroupLayout.DEFAULT_SIZE, 299,
Short.MAX_VALUE)
        .addContainerGap())
);

```

```
docsPane.addTab("Main", mainTab);
```

```
subEditorTab.setBackground(new java.awt.Color(255, 255, 204));
```

```
subEditorTab.setForeground(new java.awt.Color(0, 0, 102));
```

```
addTab.setBackground(java.awt.SystemColor.info);
```

```
addTab.setForeground(new java.awt.Color(0, 0, 102));
```

```
starLabel.setForeground(new java.awt.Color(0, 0, 102));
```

```
starLabel.setText("**starred fields must be filled");
```

```
addNameLabel.setForeground(new java.awt.Color(0, 0, 102));
```

```
addNameLabel.setText("Name:");
```

```
addNameField.setForeground(new java.awt.Color(0, 0, 102));
```

```
addNameField.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```

        addNameFieldActionPerformed(evt);
    }
});

addDOBLabel.setForeground(new java.awt.Color(0, 0, 102));
addDOBLabel.setText("DOB(dd/mm/yyyy):*");

addDOBField.setForeground(new java.awt.Color(0, 0, 102));

addGenLabel.setForeground(new java.awt.Color(0, 0, 102));
addGenLabel.setText("Gender:*");

maleToggleButton.setForeground(new java.awt.Color(0, 0, 102));
maleToggleButton.setText("M");
maleToggleButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        maleToggleButtonActionPerformed(evt);
    }
});

femaleToggleButton.setForeground(new java.awt.Color(0, 0, 102));
femaleToggleButton.setText("F");
femaleToggleButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        femaleToggleButtonActionPerformed(evt);
    }
});

addEmailLabel.setForeground(new java.awt.Color(0, 0, 102));
addEmailLabel.setText("Email:*");

```

```
addEmailField.setForeground(new java.awt.Color(0, 0, 102));
```

```
addPhoNumLabel.setForeground(new java.awt.Color(0, 0, 102));
```

```
addPhoNumLabel.setText("Phone Number:*");
```

```
addEmeNumLabel.setForeground(new java.awt.Color(0, 0, 102));
```

```
addEmeNumLabel.setText("Emergency Number:");
```

```
addPhoNumField.setForeground(new java.awt.Color(0, 0, 102));
```

```
addEmerNumField.setForeground(new java.awt.Color(0, 0, 102));
```

```
addSessDateLabel.setForeground(new java.awt.Color(0, 0, 102));
```

```
addSessDateLabel.setText("Upcoming Session Date(dd/mm/yyyy):*");
```

```
addSessDateField.setForeground(new java.awt.Color(0, 0, 102));
```

```
addSessInfoLabel.setForeground(new java.awt.Color(0, 0, 102));
```

```
addSessInfoLabel.setText("Upcoming Session Information:");
```

```
addSessInfoField.setColumns(20);
```

```
addSessInfoField.setForeground(new java.awt.Color(0, 0, 102));
```

```
addSessInfoField.setRows(5);
```

```
jScrollPane2.setViewportViewView(addSessInfoField);
```

```
addExpLabel.setForeground(new java.awt.Color(0, 0, 102));
```

```
addExpLabel.setText("Cricketing Experience: ");
```

```
addExpField.setColumns(20);
```

```
addExpField.setForeground(new java.awt.Color(0, 0, 102));
```

```
addExpField.setRows(5);
```

```
jScrollPane3.setViewportViewView(addExpField);
```

```
addMedLabel.setForeground(new java.awt.Color(0, 0, 102));
```

```
addMedLabel.setText("Medical Information: ");
```

```
addMedField.setColumns(20);
```

```
addMedField.setForeground(new java.awt.Color(0, 0, 102));
```

```
addMedField.setRows(5);
```

```
jScrollPane4.setViewportViewView(addMedField);
```

```
addBatLabel.setForeground(new java.awt.Color(0, 0, 102));
```

```
addBatLabel.setText("Batting Hand:*");
```

```
addBatLToggle.setForeground(new java.awt.Color(0, 0, 102));
```

```
addBatLToggle.setText("L");
```

```
addBatLToggle.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        addBatLToggleActionPerformed(evt);
```

```
    }
```

```
});
```

```
addBatRToggle.setForeground(new java.awt.Color(0, 0, 102));
```

```
addBatRToggle.setText("R");
```

```
addBatRToggle.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        addBatRToggleActionPerformed(evt);
```

```
    }
```

```
});
```

```
addSaveButton.setForeground(new java.awt.Color(0, 0, 102));
```

```
addSaveButton.setText("Save");
```

```
addSaveButton.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        addSaveButtonActionPerformed(evt);  
    }  
});
```

```
addResetButton.setForeground(new java.awt.Color(0, 0, 102));  
addResetButton.setText("Reset");  
addResetButton.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        addResetButtonActionPerformed(evt);  
    }  
});
```

```
addTimeLabel.setForeground(new java.awt.Color(0, 0, 102));  
addTimeLabel.setText("Time:");
```

```
addPayField.setColumns(20);  
addPayField.setForeground(new java.awt.Color(0, 0, 102));  
addPayField.setRows(5);  
jScrollPane6.setViewportView(addPayField);
```

```
addPayLabel.setForeground(new java.awt.Color(0, 0, 102));  
addPayLabel.setText("Payment Information");
```

```
addPaySessLabel.setForeground(new java.awt.Color(0, 0, 102));  
addPaySessLabel.setText("Payment Per Session");
```

```
addPaySessField.setForeground(new java.awt.Color(0, 0, 102));
```

```

        addTimeBox.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Select Time",
"00:00", "00:15", "00:30", "00:45", "01:00", "01:15", "01:30", "01:45", "02:00", "02:15", "02:30",
"02:45", "03:00", "03:15", "03:30", "03:45", "04:00", "04:15", "04:30", "04:45", "05:00", "05:15",
"05:30", "05:45", "06:00", "06:15", "06:30", "06:45", "07:00", "07:15", "07:30", "07:45", "08:00",
"08:15", "08:30", "08:45", "09:00", "09:15", "09:30", "09:45", "10:00", "10:15", "10:30", "10:45",
"11:00", "11:15", "11:30", "11:45", "12:00", "12:15", "12:30", "12:45", "13:00", "13:15", "13:30",
"13:45", "14:00", "14:15", "14:30", "14:45", "15:00", "15:15", "15:30", "15:45", "16:00", "16:15",
"16:30", "16:45", "17:00", "17:15", "17:30", "17:45", "18:00", "18:15", "18:30", "18:45", "19:00",
"19:15", "19:30", "19:45", "20:00", "20:15", "20:30", "20:45", "21:00", "21:15", "21:30", "21:45",
"22:00", "22:15", "22:30", "22:45", "23:00", "23:15", "23:30", "23:45" }));

```

```

        javax.swing.GroupLayout addTabLayout = new javax.swing.GroupLayout(addTab);
        addTab.setLayout(addTabLayout);
        addTabLayout.setHorizontalGroup(
            addTabLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
                    addTabLayout.createSequentialGroup()
                        .addComponent(addSaveButton)
                        .addGap(18, 18, 18)
                        .addComponent(addResetButton)
                        .addGap(51, 51, 51)
                        .addGroup(addTabLayout.createSequentialGroup()
                            .addComponent(addTabLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                .addGroup(addTabLayout.createSequentialGroup()
                                    .addComponent(addNameLabel)
                                    .addGap(18, 18, 18)
                                    .addComponent(addNameField, javax.swing.GroupLayout.PREFERRED_SIZE, 133,
                                        javax.swing.GroupLayout.PREFERRED_SIZE)
                                    .addGap(18, 18, 18)
                                    .addComponent(starLabel))
                                .addGroup(addTabLayout.createSequentialGroup()
                                    .addComponent(addTimeBox)
                                    .addGap(18, 18, 18)
                                    .addComponent(addTimeBox)))
                            .addGap(18, 18, 18))
                )
        );

```

```

.addGroup(addTabLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(addTabLayout.createSequentialGroup()
        .addComponent(addGenLabel)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(maleToggleButton)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(femaleToggleButton)
        .addGap(18, 18, 18)
        .addComponent(addBatLabel)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(addBatLToggle)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(addBatRToggle))
    .addGroup(addTabLayout.createSequentialGroup()
        .addComponent(addDOBLLabel)
        .addGap(18, 18, 18)
        .addComponent(addDOBField, javax.swing.GroupLayout.PREFERRED_SIZE, 101,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
        .addComponent(addEmailLabel)
        .addGap(18, 18, 18)
        .addComponent(addEmailField, javax.swing.GroupLayout.PREFERRED_SIZE, 142,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGroup(addTabLayout.createSequentialGroup()
        .addComponent(addPhoNumLabel)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(addPhoNumField, javax.swing.GroupLayout.PREFERRED_SIZE, 99,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(addEmeNumLabel)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

```

```

        .addComponent(addEmerNumField, javax.swing.GroupLayout.PREFERRED_SIZE,
101, javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGroup(addTabLayout.createSequentialGroup())

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

        .addComponent(addSessInfoLabel)

        .addComponent(addExpLabel)

        .addComponent(addMedLabel)

        .addComponent(addPayLabel)

        .addComponent(addPaySessLabel))

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

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

        .addComponent(jScrollPane6, javax.swing.GroupLayout.DEFAULT_SIZE, 286,
Short.MAX_VALUE)

        .addComponent(jScrollPane2)

        .addComponent(jScrollPane3)

        .addComponent(jScrollPane4)

        .addComponent(addPaySessField, javax.swing.GroupLayout.PREFERRED_SIZE,
80, javax.swing.GroupLayout.PREFERRED_SIZE)))

        .addGroup(addTabLayout.createSequentialGroup())

        .addComponent(addSessDateLabel)

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

        .addComponent(addSessDateField, javax.swing.GroupLayout.PREFERRED_SIZE,
101, javax.swing.GroupLayout.PREFERRED_SIZE)

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

        .addComponent(addTimeLabel)

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

        .addComponent(addTimeBox, javax.swing.GroupLayout.PREFERRED_SIZE, 128,
javax.swing.GroupLayout.PREFERRED_SIZE)))

        .addGap(0, 23, Short.MAX_VALUE)))

.addContainerGap()

);

```



```

addTabLayout.setVerticalGroup(

    addTabLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(addTabLayout.createSequentialGroup())

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

                .addComponent(starLabel)

                .addGroup(addTabLayout.createSequentialGroup())

                    .addContainerGap()

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

                    .addComponent(addNameField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

                    .addComponent(addNameLabel))))

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

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

                .addComponent(addDOBField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

                .addComponent(addDOBLLabel)

                .addComponent(addEmailField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

                .addComponent(addEmailLabel))

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

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

                .addComponent(addGenLabel)

                .addComponent(maleToggleButton)

                .addComponent(femaleToggleButton)

                .addComponent(addBatLabel)

                .addComponent(addBatLToggle)

                .addComponent(addBatRToggle))

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

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

```

```

        .addComponent(addPhoNumLabel)

        .addComponent(addEmeNumLabel)

        .addComponent(addPhoNumField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(addEmerNumField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

        .addComponent(addSessDateLabel)

        .addComponent(addSessDateField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(addTimeLabel)

        .addComponent(addTimeBox, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(52, 52, 52)

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

        .addComponent(addSessInfoLabel)

        .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 51,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

        .addComponent(addExpLabel)

        .addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED_SIZE, 51,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

        .addComponent(addMedLabel)

        .addComponent(jScrollPane4, javax.swing.GroupLayout.PREFERRED_SIZE, 51,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

```

```

.addGroup(addTabLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(addPayLabel)
    .addComponent(jScrollPane6, javax.swing.GroupLayout.PREFERRED_SIZE, 52,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 13,
Short.MAX_VALUE)

```

```

.addGroup(addTabLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(addPaySessLabel)
    .addComponent(addPaySessField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(15, 15, 15)

```

```

.addGroup(addTabLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(addSaveButton)
    .addComponent(addResetButton)))
);

```

```

addPhoNumField.addKeyListener(new KeyAdapter() {
    public void keyTyped(KeyEvent ev) {
        char c = ev.getKeyChar();
        if ( ((c < '0') || (c > '9')) && (c != KeyEvent.VK_BACK_SPACE)) {
            ev.consume(); // ignore event
        }
    }
});

```

```

addEmerNumField.addKeyListener(new KeyAdapter() {
    public void keyTyped(KeyEvent ev) {
        char c = ev.getKeyChar();
        if ( ((c < '0') || (c > '9')) && (c != KeyEvent.VK_BACK_SPACE)) {
            ev.consume(); // ignore event
        }
    }
});

```

```

    }
});
addPaySessField.addKeyListener(new KeyAdapter() {
    public void keyTyped(KeyEvent ev) {
        char c = ev.getKeyChar();
        if ( ((c < '0') || (c > '9')) && (c != KeyEvent.VK_BACK_SPACE)) {
            ev.consume(); // ignore event
        }
    }
});

subEditorTab.addTab("Add Pupil", addTab);

removeTab.setBackground(java.awt.SystemColor.info);

removeLabel.setForeground(new java.awt.Color(0, 0, 102));
removeLabel.setText("Enter the name of the pupil you would like to remove:");

removeField.setForeground(new java.awt.Color(0, 0, 102));

removeButton.setForeground(new java.awt.Color(0, 0, 102));
removeButton.setText("Submit");
removeButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        removeButtonActionPerformed(evt);
    }
});

javax.swing.GroupLayout removeTabLayout = new javax.swing.GroupLayout(removeTab);
removeTab.setLayout(removeTabLayout);
removeTabLayout.setHorizontalGroup(

```

```

        removeTabLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
removeTabLayout.createSequentialGroup())
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(removeButton)
                .addGap(128, 128, 128))
            .addGroup(removeTabLayout.createSequentialGroup())
                .addContainerGap()

        .addGroup(removeTabLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(removeLabel)
            .addComponent(removeField, javax.swing.GroupLayout.PREFERRED_SIZE, 130,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addContainerGap(244, Short.MAX_VALUE))
    );

    removeTabLayout.setVerticalGroup(
        removeTabLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(removeTabLayout.createSequentialGroup())
                .addContainerGap()
                .addComponent(removeLabel)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(removeField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(removeButton)
                .addContainerGap(414, Short.MAX_VALUE))
    );

    subEditorTab.addTab("Remove Pupil", removeTab);

    javax.swing.GroupLayout editorTabLayout = new javax.swing.GroupLayout(editorTab);
    editorTab.setLayout(editorTabLayout);

```

```

editorTabLayout.setHorizontalGroup(
    editorTabLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(subEditorTab)
);
editorTabLayout.setVerticalGroup(
    editorTabLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(subEditorTab)
);

docsPane.addTab("Pupils", editorTab);

sessionTab.setBackground(java.awt.SystemColor.info);
sessionTab.setForeground(new java.awt.Color(0, 0, 102));

sessNameLabel.setForeground(new java.awt.Color(0, 0, 102));
sessNameLabel.setText("Enter pupil name:");

sessNameField.setForeground(new java.awt.Color(0, 0, 102));

sessEnterButton.setForeground(new java.awt.Color(0, 0, 102));
sessEnterButton.setText("Enter");
sessEnterButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        sessEnterButtonActionPerformed(evt);
    }
});

sessEmailLabel.setForeground(new java.awt.Color(0, 0, 102));
sessEmailLabel.setText("Pupil Email:");

sessEmailField.setForeground(new java.awt.Color(0, 0, 102));

```

```
sessInfoLabel.setForeground(new java.awt.Color(0, 0, 102));  
sessInfoLabel.setText("Pupil current upcoming session details:");
```

```
sessInfoField.setColumns(20);  
sessInfoField.setForeground(new java.awt.Color(0, 0, 102));  
sessInfoField.setRows(5);  
jScrollPane12.setViewportViewView(sessInfoField);
```

```
sessSaveButton.setForeground(new java.awt.Color(0, 0, 102));  
sessSaveButton.setText("Save");  
sessSaveButton.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        sessSaveButtonActionPerformed(evt);  
    }  
});
```

```
sessSendLabel.setForeground(new java.awt.Color(0, 0, 102));  
sessSendLabel.setText("Would you like to send this automated email:");
```

```
sessSendField.setColumns(20);  
sessSendField.setForeground(new java.awt.Color(0, 0, 102));  
sessSendField.setRows(5);  
jScrollPane13.setViewportViewView(sessSendField);
```

```
sessSureButton.setForeground(new java.awt.Color(0, 0, 102));  
sessSureButton.setText("Sure");  
sessSureButton.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        sessSureButtonActionPerformed(evt);  
    }  
});
```

```
});
```

```
sessCancelButton.setForeground(new java.awt.Color(0, 0, 102));
```

```
sessCancelButton.setText("Cancel");
```

```
sessCancelButton.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        sessCancelButtonActionPerformed(evt);
```

```
    }
```

```
});
```

```
sessDateLabel.setForeground(new java.awt.Color(0, 0, 102));
```

```
sessDateLabel.setText("Upcoming Session Date(dd/mm/yyyy):*");
```

```
sessDateField.setForeground(new java.awt.Color(0, 0, 102));
```

```
sessTimeLabel.setForeground(new java.awt.Color(0, 0, 102));
```

```
sessTimeLabel.setText("Upcoming Session Time:");
```

```
sessPayLabel.setForeground(new java.awt.Color(0, 0, 102));
```

```
sessPayLabel.setText("Enter new value of payment per session (if needed):");
```

```
sessPayField.setForeground(new java.awt.Color(0, 0, 102));
```

```
sessTimeBox.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Select  
Time", "00:00", "00:15", "00:30", "00:45", "01:00", "01:15", "01:30", "01:45", "02:00", "02:15",  
"02:30", "02:45", "03:00", "03:15", "03:30", "03:45", "04:00", "04:15", "04:30", "04:45", "05:00",  
"05:15", "05:30", "05:45", "06:00", "06:15", "06:30", "06:45", "07:00", "07:15", "07:30", "07:45",  
"08:00", "08:15", "08:30", "08:45", "09:00", "09:15", "09:30", "09:45", "10:00", "10:15", "10:30",  
"10:45", "11:00", "11:15", "11:30", "11:45", "12:00", "12:15", "12:30", "12:45", "13:00", "13:15",  
"13:30", "13:45", "14:00", "14:15", "14:30", "14:45", "15:00", "15:15", "15:30", "15:45", "16:00",  
"16:15", "16:30", "16:45", "17:00", "17:15", "17:30", "17:45", "18:00", "18:15", "18:30", "18:45",  
"19:00", "19:15", "19:30", "19:45", "20:00", "20:15", "20:30", "20:45", "21:00", "21:15", "21:30",  
"21:45", "22:00", "22:15", "22:30", "22:45", "23:00", "23:15", "23:30", "23:45" }));
```



```

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

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

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

        .addGroup(sessionTabLayout.createSequentialGroup()
            .addComponent(sessNameLabel)
            .addGap(18, 18, 18)
            .addComponent(sessNameField, javax.swing.GroupLayout.PREFERRED_SIZE,
120, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(sessionTabLayout.createSequentialGroup()
            .addComponent(sessDateLabel)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(sessDateField, javax.swing.GroupLayout.PREFERRED_SIZE, 100,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(sessionTabLayout.createSequentialGroup()
            .addComponent(sessTimeLabel)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(sessTimeBox, javax.swing.GroupLayout.PREFERRED_SIZE, 128,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addComponent(sessInfoLabel)
            .addComponent(jScrollPane12, javax.swing.GroupLayout.PREFERRED_SIZE, 348,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGroup(sessionTabLayout.createSequentialGroup()
            .addComponent(sessPayLabel)

```

```

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(sessPayField))
    .addGroup(sessionTabLayout.createSequentialGroup())
        .addComponent(sessEmailLabel)
        .addGap(18, 18, 18)
        .addComponent(sessEmailField, javax.swing.GroupLayout.PREFERRED_SIZE, 132,
javax.swing.GroupLayout.PREFERRED_SIZE)))
    .addComponent(sessSendLabel)
    .addComponent(jScrollPane13, javax.swing.GroupLayout.PREFERRED_SIZE, 300,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(0, 148, Short.MAX_VALUE))
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
sessionTabLayout.createSequentialGroup())
    .addGap(0, 0, Short.MAX_VALUE)

    .addGroup(sessionTabLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(sessEnterButton, javax.swing.GroupLayout.Alignment.TRAILING)
        .addComponent(sessSaveButton, javax.swing.GroupLayout.Alignment.TRAILING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
sessionTabLayout.createSequentialGroup())
            .addComponent(sessSureButton)
            .addGap(18, 18, 18)
            .addComponent(sessCancelButton))))))
    .addContainerGap())
);
sessionTabLayout.setVerticalGroup(
    sessionTabLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(sessionTabLayout.createSequentialGroup())
            .addContainerGap()

    .addGroup(sessionTabLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(sessNameLabel)

```

```

        .addComponent(sessNameField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(sessEnterButton)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(sessionTabLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(sessEmailLabel)

        .addComponent(sessEmailField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(sessionTabLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(sessDateLabel)

        .addComponent(sessDateField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(sessionTabLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(sessTimeLabel)

        .addComponent(sessTimeBox, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(sessInfoLabel)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jScrollPane12, javax.swing.GroupLayout.PREFERRED_SIZE, 50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(sessionTabLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(sessPayLabel)

        .addComponent(sessPayField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

```

```

        .addComponent(sessSaveButton)

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

        .addComponent(sessSendLabel)

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

        .addComponent(jScrollPane13, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

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

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

        .addComponent(sessCancelButton)

        .addComponent(sessSureButton))

.addContainerGap(82, Short.MAX_VALUE))

);

```

```

sessPayField.addKeyListener(new KeyAdapter() {

    public void keyTyped(KeyEvent ev) {

        char c = ev.getKeyChar();

        if ( ((c < '0') || (c > '9')) && (c != KeyEvent.VK_BACK_SPACE)) {

            ev.consume(); // ignore event

        }

    }

});

```

```

docsPane.addTab("Sessions", sessionTab);

```

```

jTabbedPane1.setBackground(new java.awt.Color(255, 255, 204));

jTabbedPane1.setForeground(new java.awt.Color(0, 0, 102));

```

```

jPanel2.setBackground(java.awt.SystemColor.info);

jPanel2.setForeground(new java.awt.Color(0, 0, 102));

```

```

jLabel3.setForeground(new java.awt.Color(0, 0, 102));

```

```
jLabel3.setText("Enter the name of the pupil:");
```

```
formNameField.setForeground(new java.awt.Color(0, 0, 102));
```

```
jLabel4.setForeground(new java.awt.Color(0, 0, 102));
```

```
jLabel4.setText("Enter the date of injury:");
```

```
formDateField.setForeground(new java.awt.Color(0, 0, 102));
```

```
jLabel5.setForeground(new java.awt.Color(0, 0, 102));
```

```
jLabel5.setText("Enter the time of injury:");
```

```
formNameButton.setForeground(new java.awt.Color(0, 0, 102));
```

```
formNameButton.setText("Enter");
```

```
formNameButton.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        formNameButtonActionPerformed(evt);  
    }  
});
```

```
jLabel6.setForeground(new java.awt.Color(0, 0, 102));
```

```
jLabel6.setText("Pupil Email:");
```

```
formEmailField.setForeground(new java.awt.Color(0, 0, 102));
```

```
jLabel7.setForeground(new java.awt.Color(0, 0, 102));
```

```
jLabel7.setText("Pupil Phone Number:");
```

```
formPhoField.setForeground(new java.awt.Color(0, 0, 102));
```

```
jLabel8.setForeground(new java.awt.Color(0, 0, 102));
```

```
jLabel8.setText("Pupil Emergency Number:");
```

```
formEmerField.setForeground(new java.awt.Color(0, 0, 102));
```

```
jLabel10.setForeground(new java.awt.Color(0, 0, 102));
```

```
jLabel10.setText("Pupil Medical Information:");
```

```
formMedField.setColumns(20);
```

```
formMedField.setForeground(new java.awt.Color(0, 0, 102));
```

```
formMedField.setRows(5);
```

```
jScrollPane1.setViewportViewView(formMedField);
```

```
jLabel11.setForeground(new java.awt.Color(0, 0, 102));
```

```
jLabel11.setText("Pupil Nature of Injury:");
```

```
formInjuryField.setColumns(20);
```

```
formInjuryField.setForeground(new java.awt.Color(0, 0, 102));
```

```
formInjuryField.setRows(5);
```

```
jScrollPane7.setViewportViewView(formInjuryField);
```

```
formSaveButton.setForeground(new java.awt.Color(0, 0, 102));
```

```
formSaveButton.setText("Save as Form");
```

```
formSaveButton.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        formSaveButtonActionPerformed(evt);
```

```
    }
```

```
});
```

```
formCancelButton.setForeground(new java.awt.Color(0, 0, 102));
```

```
formCancelButton.setText("Cancel");
```

```
formCancelButton.addActionListener(new java.awt.event.ActionListener() {
```



```

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(formPhoField, javax.swing.GroupLayout.PREFERRED_SIZE, 109,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addComponent(jLabel10)
        .addComponent(jScrollPane7)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup())
        .addComponent(jLabel4)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(formDateField, javax.swing.GroupLayout.DEFAULT_SIZE, 105,
Short.MAX_VALUE)
        .addGap(77, 77, 77))
        .addComponent(jLabel11)
        .addComponent(jScrollPane1)
        .addGroup(jPanel2Layout.createSequentialGroup())
        .addComponent(jLabel5)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(formTimeBox, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        .addGroup(jPanel2Layout.createSequentialGroup())
        .addComponent(jLabel3)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(formNameField, javax.swing.GroupLayout.PREFERRED_SIZE,
106, javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGroup(jPanel2Layout.createSequentialGroup())
        .addComponent(jLabel8)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(formEmerField, javax.swing.GroupLayout.PREFERRED_SIZE, 109,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGap(0, 182, Short.MAX_VALUE))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup())
        .addGap(0, 0, Short.MAX_VALUE)

```



```

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(formNameButton, javax.swing.GroupLayout.Alignment.TRAILING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()
    .addComponent(formSaveButton)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(formCancelButton))))))
.addContainerGap()
);
jPanel2Layout.setVerticalGroup(
jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel2Layout.createSequentialGroup()
.addContainerGap()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel3)
    .addComponent(formNameField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel4)
    .addComponent(formDateField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel5)
    .addComponent(formTimeBox, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(formNameButton)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

```

```

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel6)
    .addComponent(formEmailField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel7)
    .addComponent(formPhoField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel8)
    .addComponent(formEmerField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(jLabel10)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(jLabel11)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(jScrollPane7, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(formSaveButton)
    .addComponent(formCancelButton))
    .addContainerGap(26, Short.MAX_VALUE))
);

```

```
formPhoField.addKeyListener(new KeyAdapter() {  
    public void keyTyped(KeyEvent ev) {  
        char c = ev.getKeyChar();  
        if ( ((c < '0') || (c > '9')) && (c != KeyEvent.VK_BACK_SPACE)) {  
            ev.consume(); // ignore event  
        }  
    }  
});
```

```
formEmerField.addKeyListener(new KeyAdapter() {  
    public void keyTyped(KeyEvent ev) {  
        char c = ev.getKeyChar();  
        if ( ((c < '0') || (c > '9')) && (c != KeyEvent.VK_BACK_SPACE)) {  
            ev.consume(); // ignore event  
        }  
    }  
});
```

```
jTabbedPane1.addTab("Create New Form", jPanel2);
```

```
jPanel1.setBackground(java.awt.SystemColor.info);  
jPanel1.setForeground(new java.awt.Color(0, 0, 102));
```

```
jLabel2.setBackground(new java.awt.Color(255, 255, 204));  
jLabel2.setForeground(new java.awt.Color(0, 0, 102));  
jLabel2.setText("Enter the name of the pupil:");
```

```
accessNameField.setForeground(new java.awt.Color(0, 0, 102));
```

```
formsEnterButton.setForeground(new java.awt.Color(0, 0, 102));  
formsEnterButton.setText("Enter");
```



```

        .addGap(18, 18, 18)

        .addComponent(accessNameField, javax.swing.GroupLayout.PREFERRED_SIZE,
110, javax.swing.GroupLayout.PREFERRED_SIZE))

        .addComponent(jScrollPane8, javax.swing.GroupLayout.PREFERRED_SIZE, 349,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(0, 142, Short.MAX_VALUE))

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup())

        .addGap(0, 0, Short.MAX_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(formsEnterButton, javax.swing.GroupLayout.Alignment.TRAILING)

        .addComponent(accessClearButton,
javax.swing.GroupLayout.Alignment.TRAILING))))

        .addContainerGap())

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup())

.addContainerGap())

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(jLabel2)

        .addComponent(accessNameField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

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

        .addComponent(formsEnterButton)

        .addGap(18, 18, 18)

        .addComponent(jScrollPane8, javax.swing.GroupLayout.PREFERRED_SIZE, 300,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(18, 18, 18)

        .addComponent(accessClearButton)

        .addContainerGap(75, Short.MAX_VALUE))

);

```

```
jTabbedPane1.addTab("Access Forms", jPanel1);
```

```
javax.swing.GroupLayout formTabLayout = new javax.swing.GroupLayout(formTab);  
formTab.setLayout(formTabLayout);  
formTabLayout.setHorizontalGroup(  
    formTabLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
        .addComponent(jTabbedPane1)  
);  
formTabLayout.setVerticalGroup(  
    formTabLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
        .addComponent(jTabbedPane1)  
);
```

```
docsPane.addTab("Emergency Form", formTab);
```

```
jTabbedPane2.setBackground(new java.awt.Color(255, 255, 204));  
jTabbedPane2.setForeground(new java.awt.Color(0, 0, 102));
```

```
jPanel4.setBackground(java.awt.SystemColor.info);
```

```
jLabel12.setForeground(new java.awt.Color(0, 0, 102));  
jLabel12.setText("Enter the name of the pupil:");
```

```
docNameField.setForeground(new java.awt.Color(0, 0, 102));
```

```
jLabel13.setForeground(new java.awt.Color(0, 0, 102));  
jLabel13.setText("Enter the name of the file:");
```

```
docFileField.setForeground(new java.awt.Color(0, 0, 102));
```

```
jLabel15.setForeground(new java.awt.Color(0, 0, 102));
```

```
jLabel15.setText("Enter the file path:");
```

```
docPathField.setForeground(new java.awt.Color(0, 0, 102));
```

```
fileUploadField.setForeground(new java.awt.Color(0, 0, 102));
```

```
fileUploadField.setText("Enter");
```

```
fileUploadField.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        fileUploadFieldActionPerformed(evt);
```

```
    }
```

```
});
```

```
docCancelButton.setForeground(new java.awt.Color(0, 0, 102));
```

```
docCancelButton.setText("Cancel");
```

```
docCancelButton.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        docCancelButtonActionPerformed(evt);
```

```
    }
```

```
});
```

```
javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);
```

```
jPanel4.setLayout(jPanel4Layout);
```

```
jPanel4Layout.setHorizontalGroup(
```

```
    jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
        .addGroup(jPanel4Layout.createSequentialGroup()
```

```
            .addGap(10, 10, 10)
```

```
        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
            .addGroup(jPanel4Layout.createSequentialGroup()
```

```
        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```

        .addGroup(jPanel4Layout.createSequentialGroup())
        .addComponent(jLabel12)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(docNameField, javax.swing.GroupLayout.PREFERRED_SIZE, 106,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(jPanel4Layout.createSequentialGroup())
        .addComponent(jLabel13)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(docFileField, javax.swing.GroupLayout.PREFERRED_SIZE, 110,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(jPanel4Layout.createSequentialGroup())
        .addComponent(jLabel15)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(docPathField, javax.swing.GroupLayout.PREFERRED_SIZE, 299,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addContainerGap(101, Short.MAX_VALUE))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel4Layout.createSequentialGroup())
        .addGap(0, 0, Short.MAX_VALUE)
        .addComponent(fileUploadField)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(docCancelButton)
        .addGap(18, 18, 18)))
);

jPanel4Layout.setVerticalGroup(
    jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel4Layout.createSequentialGroup())
            .addContainerGap()

        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel12)

            .addComponent(docNameField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

```



```

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

        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel13)
            .addComponent(docFileField, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel15)
            .addComponent(docPathField, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(fileUploadField)
            .addComponent(docCancelButton))
        .addContainerGap(387, Short.MAX_VALUE))
    );

    formTimeField.addKeyListener(new KeyAdapter() {
        public void keyTyped(KeyEvent ev) {
            char c = ev.getKeyChar();
            if ( ((c < '0') || (c > '9')) && (c != KeyEvent.VK_BACK_SPACE)) {
                ev.consume(); // ignore event
            }
        }
    });

    jTabbedPane2.addTab("Upload Document", jPanel4);

    jPanel5.setBackground(java.awt.SystemColor.info);

```

```

jLabel16.setForeground(new java.awt.Color(0, 0, 102));
jLabel16.setText("Enter the name of the pupil:");

docAccessField.setForeground(new java.awt.Color(0, 0, 102));

fileAccessField.setForeground(new java.awt.Color(0, 0, 102));
fileAccessField.setText("Enter");
fileAccessField.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        fileAccessFieldActionPerformed(evt);
    }
});

docAccessTable.setBackground(new java.awt.Color(204, 204, 255));
docAccessTable.setForeground(new java.awt.Color(0, 0, 102));
docAccessTable.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {

        },
    new String [] {
        "File Name"
    }
) {
    boolean[] canEdit = new boolean [] {
        false
    };

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});

```

```

jScrollPane9.setViewportViewView(docAccessTable);

if (docAccessTable.getColumnModel().getColumnCount() > 0) {
    docAccessTable.getColumnModel().getColumn(0).setResizable(false);
}

docClearButton.setForeground(new java.awt.Color(0, 0, 102));
docClearButton.setText("Clear");
docClearButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        docClearButtonActionPerformed(evt);
    }
});

javax.swing.GroupLayout jPanel5Layout = new javax.swing.GroupLayout(jPanel5);
jPanel5.setLayout(jPanel5Layout);
jPanel5Layout.setHorizontalGroup(
    jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel5Layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel5Layout.createSequentialGroup()
                    .addComponent(docAccessField, javax.swing.GroupLayout.PREFERRED_SIZE, 106,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jScrollPane9, javax.swing.GroupLayout.PREFERRED_SIZE, 301,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(10, 10, 10))
                .addGroup(jPanel5Layout.createSequentialGroup()
                    .addComponent(docClearButton, javax.swing.GroupLayout.PREFERRED_SIZE, 100,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(10, 10, 10))
            )
        )
        .addGap(10, 10, 10)
);

```

```

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel5Layout.createSequentialGroup())

        .addGap(0, 0, Short.MAX_VALUE)

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(fileAccessField, javax.swing.GroupLayout.Alignment.TRAILING)

        .addComponent(docClearButton, javax.swing.GroupLayout.Alignment.TRAILING))))

.addContainerGap())

);

jPanel5Layout.setVerticalGroup(

jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel5Layout.createSequentialGroup())

.addContainerGap()

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(jLabel16)

        .addComponent(docAccessField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

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

.addComponent(fileAccessField)

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

.addComponent(jScrollPane9, javax.swing.GroupLayout.PREFERRED_SIZE, 109,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

.addComponent(docClearButton)

.addContainerGap(290, Short.MAX_VALUE))

);

jTabbedPane2.addTab("Access Documents", jPanel5);

javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);
jPanel3.setLayout(jPanel3Layout);
jPanel3Layout.setHorizontalGroup(

```

```
jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jTabbedPane2)
);
jPanel3Layout.setVerticalGroup(
    jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jTabbedPane2)
    );

docsPane.addTab("Documents", jPanel3);

infoTab.setBackground(java.awt.SystemColor.info);
infoTab.setForeground(new java.awt.Color(0, 0, 102));

infoNameLabel.setBackground(new java.awt.Color(255, 255, 204));
infoNameLabel.setForeground(new java.awt.Color(0, 0, 102));
infoNameLabel.setText("Company Name:   Masterclass Cricket Academy");

infoDateLabel.setBackground(new java.awt.Color(255, 255, 204));
infoDateLabel.setForeground(new java.awt.Color(0, 0, 102));
infoDateLabel.setText("Company Start Date:   02/04/2016");

infoIncomeLabel.setBackground(new java.awt.Color(255, 255, 204));
infoIncomeLabel.setForeground(new java.awt.Color(0, 0, 102));
infoIncomeLabel.setText("Company Estimated Income:");

infoCEMailLabel.setBackground(new java.awt.Color(255, 255, 204));
infoCEMailLabel.setForeground(new java.awt.Color(0, 0, 102));
infoCEMailLabel.setText("Company Email:   Uzi@masterclassca.com");

infoMEmailLabel.setBackground(new java.awt.Color(255, 255, 204));
infoMEmailLabel.setForeground(new java.awt.Color(0, 0, 102));
```

```
infoMEmailLabel.setText("My Email: Usman@gmail.com");
```

```
infoNumLabel.setBackground(new java.awt.Color(255, 255, 204));
```

```
infoNumLabel.setForeground(new java.awt.Color(0, 0, 102));
```

```
infoNumLabel.setText("Work Number: 03492032");
```

```
infoPassLabel.setBackground(new java.awt.Color(255, 255, 204));
```

```
infoPassLabel.setForeground(new java.awt.Color(0, 0, 102));
```

```
infoPassLabel.setText("To change password, enter current password below:");
```

```
infoPassField.setForeground(new java.awt.Color(0, 0, 102));
```

```
infoPassButton.setForeground(new java.awt.Color(0, 0, 102));
```

```
infoPassButton.setText("Enter");
```

```
infoPassButton.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        infoPassButtonActionPerformed(evt);
```

```
    }
```

```
});
```

```
infoNPassLabel.setBackground(new java.awt.Color(255, 255, 204));
```

```
infoNPassLabel.setForeground(new java.awt.Color(0, 0, 102));
```

```
infoNPassLabel.setText("New Password:");
```

```
infoNPassField.setForeground(new java.awt.Color(0, 0, 102));
```

```
infoCPassLabel.setBackground(new java.awt.Color(255, 255, 204));
```

```
infoCPassLabel.setForeground(new java.awt.Color(0, 0, 102));
```

```
infoCPassLabel.setText("Confirm Password:");
```

```
infoCPassField.setForeground(new java.awt.Color(0, 0, 102));
```

```
infoNPassButton.setForeground(new java.awt.Color(0, 0, 102));  
infoNPassButton.setText("Enter");  
infoNPassButton.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        infoNPassButtonActionPerformed(evt);  
    }  
});
```

```
infoUserLabel.setBackground(new java.awt.Color(255, 255, 204));  
infoUserLabel.setForeground(new java.awt.Color(0, 0, 102));  
infoUserLabel.setText("To change your username, enter the new username below:");
```

```
infoUserField.setForeground(new java.awt.Color(0, 0, 102));
```

```
infoUserButton.setForeground(new java.awt.Color(0, 0, 102));  
infoUserButton.setText("Enter");  
infoUserButton.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        infoUserButtonActionPerformed(evt);  
    }  
});
```

```
infoSecurityCombo.setForeground(new java.awt.Color(0, 0, 102));  
infoSecurityCombo.setModel(new javax.swing.DefaultComboBoxModel<>(new String[]  
{"Choose a security question", "What is your mothers maiden name?", "What was your first pets  
name?", "Where was your first ever vacation?", "Who is your favourite nephew?"}));
```

```
infoSecurityLabel.setBackground(new java.awt.Color(255, 255, 204));  
infoSecurityLabel.setForeground(new java.awt.Color(0, 0, 102));  
infoSecurityLabel.setText("Answer the question here:");
```

```
infoSecurityField.setForeground(new java.awt.Color(0, 0, 102));
```

```
infoSecurityButton.setForeground(new java.awt.Color(0, 0, 102));
```

```
infoSecurityButton.setText("Save");
```

```
infoSecurityButton.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        infoSecurityButtonActionPerformed(evt);
```

```
    }
```

```
});
```

```
javax.swing.GroupLayout infoTabLayout = new javax.swing.GroupLayout(infoTab);
```

```
infoTab.setLayout(infoTabLayout);
```

```
infoTabLayout.setHorizontalGroup(
```

```
    infoTabLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
        .addGroup(infoTabLayout.createSequentialGroup()
```

```
            .addGap()
```

```
        .addGroup(infoTabLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
        .addGroup(infoTabLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
```

```
            .addComponent(infoUserButton)
```

```
        .addGroup(infoTabLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
            .addGroup(infoTabLayout.createSequentialGroup()
```

```
                .addGap(163, 163, 163)
```

```
                .addComponent(infoPassButton))
```

```
        .addGroup(infoTabLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
```

```
            .addComponent(infoNPassButton)
```

```
        .addGroup(infoTabLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
            .addComponent(infoNameLabel)
```

```
            .addComponent(infoDateLabel)
```



```

        .addComponent(infoIncomeLabel)
        .addComponent(infoCEMailLabel)
        .addComponent(infoMEmailLabel)
        .addComponent(infoNumLabel)
        .addComponent(infoPassLabel)
        .addGroup(infoTabLayout.createSequentialGroup())
        .addComponent(infoNPassLabel)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(infoNPassField, javax.swing.GroupLayout.PREFERRED_SIZE,
105, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(infoTabLayout.createSequentialGroup())
        .addComponent(infoCPassLabel)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(infoCPassField, javax.swing.GroupLayout.PREFERRED_SIZE,
108, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addComponent(infoPassField, javax.swing.GroupLayout.PREFERRED_SIZE, 108,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addComponent(infoUserLabel)
        .addComponent(infoUserField, javax.swing.GroupLayout.PREFERRED_SIZE, 110,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addComponent(infoSecurityButton, javax.swing.GroupLayout.PREFERRED_SIZE, 69,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addComponent(infoSecurityCombo, javax.swing.GroupLayout.PREFERRED_SIZE, 216,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(infoSecurityLabel)
        .addComponent(infoSecurityField, javax.swing.GroupLayout.PREFERRED_SIZE, 140,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap(221, Short.MAX_VALUE))
    );
    infoTabLayout.setVerticalGroup(
        infoTabLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(infoTabLayout.createSequentialGroup())

```

```
.addContainerGap()

.addComponent(infoNameLabel)

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

.addComponent(infoDateLabel)

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

.addComponent(infoIncomeLabel)

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

.addComponent(infoCEMailLabel)

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

.addComponent(infoMEmailLabel)

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

.addComponent(infoNumLabel)

.addGap(29, 29, 29)

.addComponent(infoPassLabel)

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

.addComponent(infoPassField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(10, 10, 10)

.addComponent(infoPassButton)

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

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

    .addComponent(infoNPassLabel)

    .addComponent(infoNPassField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

    .addComponent(infoCPassLabel)

    .addComponent(infoCPassField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

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

.addComponent(infoNPassButton)
```

```

        .addGap(18, 18, 18)

        .addComponent(infoUserLabel)

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

        .addComponent(infoUserField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

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

        .addComponent(infoUserButton)

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

        .addComponent(infoSecurityCombo, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

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

        .addComponent(infoSecurityLabel)

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

        .addComponent(infoSecurityField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

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

        .addComponent(infoSecurityButton)

        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );

```

```
docsPane.addTab("Personal Information", infoTab);
```

```

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

layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(docsPane, javax.swing.GroupLayout.PREFERRED_SIZE, 521,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(tableScrollPane, javax.swing.GroupLayout.DEFAULT_SIZE, 619,
Short.MAX_VALUE))
        );

```

```

layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(docsPane)
        .addGroup(layout.createSequentialGroup()
            .addComponent(tableScrollPane)
            .addContainerGap())
        );

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

```

```

private void mainBootActionPerformed(java.awt.event.ActionEvent evt) {
    // Calls reboot method to load or refresh data in the database
    reboot();
}

```

```

private void searchButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // Performs search for a specific chosen parameter while also doing consistent error checking
    Connection con = null;
    Statement st = null;
    ResultSet rs = null;
    ResultSet rt = null;
    String url = "jdbc:mysql://localhost:3306/CricketDatabase";
    String user = "root";
    String password = "";
    int rowCount = 0;
    Object[] row = new String[6];
    String temp[] = new String[15];
    DefaultTableModel model = (DefaultTableModel) mainTable.getModel();
    if (search.equals("")) {
        output("Please choose one of the above fields");
    }
}

```

```

        nameToggleSearch.setVisible(true);
        ageToggleSearch.setVisible(true);
        genToggleSearch.setVisible(true);
        emailToggleSearch.setVisible(true);
        numToggleSearch.setVisible(true);
        nameToggleSearch.setSelected(false);
        ageToggleSearch.setSelected(false);
        genToggleSearch.setSelected(false);
        emailToggleSearch.setSelected(false);
        numToggleSearch.setSelected(false);
        searchTextField.setText("");
    }

    if (searchTextField.getText().equals("")) {
        output("You must enter a valid input");
        nameToggleSearch.setVisible(true);
        ageToggleSearch.setVisible(true);
        genToggleSearch.setVisible(true);
        emailToggleSearch.setVisible(true);
        numToggleSearch.setVisible(true);
        nameToggleSearch.setSelected(false);
        ageToggleSearch.setSelected(false);
        genToggleSearch.setSelected(false);
        emailToggleSearch.setSelected(false);
        numToggleSearch.setSelected(false);
        searchTextField.setText("");
    } else {
        try {
            con = DriverManager.getConnection(url, user, password);
            st = con.createStatement();

            rt = st.executeQuery("select count(*) from Pupils where " + search + "=" +
searchTextField.getText() + "");

```

```

if (rt.next()) {
    rowCount = (rt.getInt(1));
}
if (rowCount < 1) {
    output("No entries match your search");
    nameToggleSearch.setVisible(true);
    ageToggleSearch.setVisible(true);
    genToggleSearch.setVisible(true);
    emailToggleSearch.setVisible(true);
    numToggleSearch.setVisible(true);
    nameToggleSearch.setSelected(false);
    ageToggleSearch.setSelected(false);
    genToggleSearch.setSelected(false);
    emailToggleSearch.setSelected(false);
    numToggleSearch.setSelected(false);
    searchTextField.setText("");
} else {
    //the actual code to search the SQL database and to load the successful queries in the
table
    model.setNumRows(0);

    rs = st.executeQuery("select * from Pupils where " + search + "=" +
searchTextField.getText() + "");

    for (int count = 1; count <= rowCount; count++) {
        if (rs.absolute(count)) {
            for (int i = 2; i <= 16; i++) {
                temp[i - 2] = rs.getString(i);
            }
        }

        row[0] = temp[0];
        row[1] = temp[1];
        row[2] = temp[3];
        row[3] = temp[8];
    }
}

```

```

        row[4] = temp[4];
        row[5] = temp[5];
        mainTable.removeAll();
        model.addRow(row);
    }

    output("Search successful. Click on any entries to view and edit the full pupil profile. To
revert back to the "

        + "full table, refresh the system");
    nameToggleSearch.setVisible(true);
    ageToggleSearch.setVisible(true);
    genToggleSearch.setVisible(true);
    emailToggleSearch.setVisible(true);
    numToggleSearch.setVisible(true);
    nameToggleSearch.setSelected(false);
    ageToggleSearch.setSelected(false);
    genToggleSearch.setSelected(false);
    emailToggleSearch.setSelected(false);
    numToggleSearch.setSelected(false);
    searchTextField.setText("");
}

} catch (SQLException ex) {
    Logger lgr = Logger.getLogger(tester.class.getName());
    lgr.log(Level.SEVERE, ex.getMessage(), ex);
} finally {
    try {
        if (rs != null) {
            rs.close();
        }
        if (st != null) {
            st.close();
        }
    }
}

```

```

        if (con != null) {
            con.close();
        }
    } catch (SQLException ex) {
        Logger lgr = Logger.getLogger(tester.class.getName());
        lgr.log(Level.WARNING, ex.getMessage(), ex);
    }
}
}
}
}

```

```

private void nameToggleSearchActionPerformed(java.awt.event.ActionEvent evt) {
    // Toggle button used for search feature
    ageToggleSearch.setVisible(false);
    genToggleSearch.setVisible(false);
    emailToggleSearch.setVisible(false);
    numToggleSearch.setVisible(false);
    search = "Name";
}

```

```

private void ageToggleSearchActionPerformed(java.awt.event.ActionEvent evt) {
    // Toggle button used for search feature
    nameToggleSearch.setVisible(false);
    genToggleSearch.setVisible(false);
    emailToggleSearch.setVisible(false);
    numToggleSearch.setVisible(false);
    search = "Age";
}

```

```

private void genToggleSearchActionPerformed(java.awt.event.ActionEvent evt) {
    // Toggle button used for search feature

```



```
ageToggleSearch.setVisible(false);
nameToggleSearch.setVisible(false);
emailToggleSearch.setVisible(false);
numToggleSearch.setVisible(false);
search = "Gender";
}
```

```
private void emailToggleSearchActionPerformed(java.awt.event.ActionEvent evt) {
    // Toggle button used for search feature
    ageToggleSearch.setVisible(false);
    genToggleSearch.setVisible(false);
    nameToggleSearch.setVisible(false);
    numToggleSearch.setVisible(false);
    search = "Email";
}
```

```
private void numToggleSearchActionPerformed(java.awt.event.ActionEvent evt) {
    // Toggle button used for search feature
    ageToggleSearch.setVisible(false);
    genToggleSearch.setVisible(false);
    emailToggleSearch.setVisible(false);
    nameToggleSearch.setVisible(false);
    search = "Phone";
}
```

```
private void cancelButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // Button used to cancel search
    nameToggleSearch.setVisible(true);
    ageToggleSearch.setVisible(true);
    genToggleSearch.setVisible(true);
    emailToggleSearch.setVisible(true);
}
```

```
numToggleSearch.setVisible(true);  
searchTextField.setText("");  
}
```

```
private void maleToggleButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    // Toggle button used to select gender in adding pupil feature  
    gen = true;  
    flag = 1;  
}
```

```
private void femaleToggleButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    // Toggle button used to select gender in adding pupil feature  
    gen = false;  
    flag = 1;  
}
```

```
private void addBatLToggleActionPerformed(java.awt.event.ActionEvent evt) {  
    // Toggle button used to select batting hand in adding pupil feature  
    hand = false;  
    flg = 1;  
}
```

```
private void addBatRToggleActionPerformed(java.awt.event.ActionEvent evt) {  
    // Toggle button used to select batting hand in adding pupil feature  
    hand = true;  
    flg = 1;  
}
```

```
private void addResetButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    // Button used to reset the add feature  
    addNameField.setText("");  
}
```

```

addDOBField.setText("");
maleToggleButton.setSelected(false);
femaleToggleButton.setSelected(false);
addEmailField.setText("");
addPhoNumField.setText("");
addEmerNumField.setText("");
addBatLToggle.setSelected(false);
addBatRToggle.setSelected(false);
addSessDateField.setText("");
addTimeBox.setSelectedIndex(0);
addSessInfoField.setText("");
addExpField.setText("");
addMedField.setText("");
addPayField.setText("");
addPaySessField.setText("");
}

```

```

private void addSaveButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // Button used to save the added pupil
    String sure = "";
    int minFlg = 0, dflag = 0, finFlag = 0;
    Date date = new Date();
    DateFormat fDate = new SimpleDateFormat("dd/MM/yyyy", Locale.ENGLISH);
    //Check placed to make sure al starred fields have been filled
    if (addNameField.getText().equals("") || addDOBField.getText().equals("") || flag == 0 || flg == 0
        || addEmailField.getText().equals("") || addPhoNumField.getText().equals("") ||
addSessDateField.getText().equals("")) {
        output("All starred fields must be filled");
        minFlg = 2;
    }

    //Check used to make sure details have not been repeated

```

```

for (int count = 0; count < index; count++) {

    if (people[count].getName().equals(addNameField.getText()) ||
people[count].getEmail().equals(addEmailField.getText()) ||
people[count].getPhoNum().equals(addPhoNumField.getText())) {

        minFlg = 1;

    }

}

if (minFlg == 1) {

    output("This pupil already exists. Please enter the correct values");

} else if (minFlg == 0) {

    try {

        //Check used to verify date is coherent, done by calling a method to check the date
        verifyDate(fDate.parse(addDOBField.getText()));

        long diff = Math.abs(date.getTime() - fDate.parse(addDOBField.getText()).getTime());

        long age = (diff / (86400000)) / 365;

        String agee = age + "";

        //Check used that date is both coherent and sensible

        if (age > 0 && vflag == 0) {

            people[index] = new Pupil(addNameField.getText(), Integer.parseInt(agee),
fDate.parse(addDOBField.getText()), gen,

                addEmailField.getText(), addPhoNumField.getText());

        } else {

            output("The date of birth value entered is invalid.");

            vflag = 0;

            finFlag = 1;

        }

    } catch (ParseException ex) {

        output("Please enter correct values for the date");

        finFlag = 1;

    }

}

if (!addEmerNumField.getText().equals("")) {

    people[index].setEmerNum(addEmerNumField.getText());

```

```

    }

    if (flg == 0) {
        people[index].setHand(hand);
    }

    fDate = new SimpleDateFormat("dd/MM/yyyy", Locale.ENGLISH);
    if (!addSessDateField.getText().equals("")) {
        try {
            verifyDate(fDate.parse(addSessDateField.getText()));

            //Check used to make sure date is coherent and that the session does not clash with
another. Client is

            //informed in all cases and advised to correct this using editor
            for (int count = 0; count < index; count++) {
                if (people[count].getSessDate().equals(fDate.parse(addSessDateField.getText()))) {
                    dflag = 1;
                }
            }

            if (dflag == 1) {
                sure = input("The session date you are saving clashes with another. Are you sure you
want to proceed(\"Yes\"\\\"No\")?");
            }

            if (vflag == 0 && (dflag == 0 || sure.equalsIgnoreCase("Yes"))) {
                people[index].setSessDate(fDate.parse(addSessDateField.getText()));
            } else {
                output("Please fix the date value");

                vflag = 0;

                finFlag = 1;
            }
        } catch (ParseException ex) {
            output("Please enter correct values for the date");
        }
    }

    if (!addTimeBox.getSelectedItem().equals("Select Time")) {

```

```

        people[index].setSessTime(addTimeBox.getSelectedItem() + "");
    }
    if (!addSessInfoField.getText().equals("")) {
        people[index].setSessInfo(addSessInfoField.getText());
    }
    if (!addExpField.getText().equals("")) {
        people[index].setExp(addExpField.getText());
    }
    if (!addMedField.getText().equals("")) {
        people[index].setMedical(addMedField.getText());
    }
    if (!addPayField.getText().equals("")) {
        people[index].setPay(addPayField.getText());
    }
    if (!addPaySessField.getText().equals("")) {
        people[index].setPaySess(Integer.parseInt(addPaySessField.getText()));
    }
    index++;
    if (finFlag == 0) {
        //reWrite method is called to save the added pupil to the database
        try {
            reWrite();
        } catch (ParseException ex) {
            Logger.getLogger(Application.class.getName()).log(Level.SEVERE, null, ex);
        }
        reboot();
        output("The Pupil has been added");
        addNameField.setText("");
        addDOBFIELD.setText("");
        maleToggleButton.setSelected(false);
        femaleToggleButton.setSelected(false);
    }
}

```

```

        addEmailField.setText("");
        addPhoNumField.setText("");
        addEmerNumField.setText("");
        addBatLToggle.setSelected(false);
        addBatRToggle.setSelected(false);
        addSessDateField.setText("");
        addTimeBox.setSelectedIndex(0);
        addSessInfoField.setText("");
        addExpField.setText("");
        addMedField.setText("");
        addPayField.setText("");
        addPaySessField.setText("");
    }
    else{
        index--;
    }
}
}

```

```

private void removeButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // Button used to remove a pupil from the database

    int flag = 0;

    if (removeField.getText().equals("")) {
        output("Please enter a name");
    } else {
        if (input("Are you sure you want to delete this pupil from the
database(\"Yes\\\"\\\"No\\\")").equalsIgnoreCase("Yes")) {
            Connection con = null;
            Statement st = null;
            ResultSet rs = null;

            String url = "jdbc:mysql://localhost:3306/CricketDatabase";

```

```

String user = "root";
String password = "";
for (int count = 0; count < index; count++) {
    if (people[count].getName().equals(removeField.getText())) {
        flag = 1;
        try {
            con = DriverManager.getConnection(url, user, password);
            st = con.createStatement();
            st.executeUpdate("delete from Pupils where Name='" + removeField.getText() + "'");
            //The relevant pupil is also checked for through other tables, if exists then those
records are deleted too
            st.executeUpdate("delete from Forms where Name =" + removeField.getText() +
""");
            st.executeUpdate("delete from Documents where Name =" + removeField.getText()
+ """);

            output("The pupil has been deleted");
            removeField.setText("");
            reboot();
        } catch (SQLException ex) {
            Logger lgr = Logger.getLogger(tester.class.getName());
            lgr.log(Level.SEVERE, ex.getMessage(), ex);
        } finally {
            //Close used to clean and close SQL resources. Catches exceptions and makes sure
no data is lost
            //This part of code is used in every SQL method for the stated reasons
            try {
                if (rs != null) {
                    rs.close();
                }
                if (st != null) {
                    st.close();
                }
            }

```



```

        if (con != null) {
            con.close();
        }
    } catch (SQLException ex) {
        Logger lgr = Logger.getLogger(tester.class.getName());
        lgr.log(Level.WARNING, ex.getMessage(), ex);
    }
}

}

}

if (flag == 0) {
    output("This person does not exist in the database. Please enter another name");
    reboot();
    removeField.setText("");
}

} else {
    removeField.setText("");
}

}

}

```

```

private void sessEnterButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // Button used to find the name in the database and then automatically load relvant data in
    fields

```

```

    int flag = 0;

```

```

    for (int count = 0; count < index; count++) {

```

```

        if (people[count].getName().equals(sessNameField.getText())) {

```

```

            sessEmailField.setText(people[count].getEmail());

```

```

            SimpleDateFormat fDate = new SimpleDateFormat("dd/MM/yyyy");

```

```

            sessDateField.setText(fDate.format(people[count].getSessDate()));

```

```

            sessTimeBox.setSelectedItem(people[count].getSessTime());

```

```

        sessInfoField.setText(people[count].getSessInfo());
        sessPayField.setText(people[count].getPaySess() + "");
        flag = 1;
        break;
    }
}
if (flag == 0) {
    output("This pupil does not exist. Please enter another name.");
    sessNameField.setText("");
    sessEmailField.setText("");
    sessDateField.setText("");
    sessTimeBox.setSelectedIndex(0);
    sessInfoField.setText("");
    sessPayField.setText("");
    sessSendField.setText("");
}
}

```

```

private void sessSaveButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // Button used to update data regarding relevant session and also automatically generate an
    email to send to relevant pupil

    String sure = "";
    int flag = 0, dflag = 0;
    DateFormat fDate = new SimpleDateFormat("dd/MM/yyyy", Locale.ENGLISH);
    if (sessDateField.getText().equals("")) {
        output("All starred field must be filled.");
    }
    for (int count = 0; count < index; count++) {
        if (people[count].getName().equals(sessNameField.getText())) {
            try {

```

```

        verifyDate(fDate.parse(sessDateField.getText()));

        //Checks for coherent date values and session clashes are maintained
        for (int counter = 0; counter < index; counter++) {
            if (people[counter].getSessDate().equals(fDate.parse(sessDateField.getText()))) {
                dflag = 1;
            }
        }

        if (dflag == 1) {
            sure = input("The session date you are saving clashes with another. Are you sure you
want to proceed(\"Yes\"\\\"No\")?");
        }

        if (vflag == 0 && (dflag == 0 || sure.equalsIgnoreCase("Yes"))) {
            people[count].setSessDate(fDate.parse(sessDateField.getText()));
        } else {
            output("The date you have entered is incorrect");
            vflag = 0;
        }
    } catch (ParseException ex) {
        Logger.getLogger(Application.class.getName()).log(Level.SEVERE, null, ex);
        output("Please enter correct values");
    }

    people[count].setSessTime(sessTimeBox.getSelectedItem() + "");
    people[count].setSessInfo(sessInfoField.getText());
    people[count].setPaySess(Integer.parseInt(sessPayField.getText()));

    flag = 1;
    break;
}

}

if (flag == 0) {
    output("This pupil does not exist. Please enter another name.");
}

```

```

//Email fields and labels are added. The email is auto-generated.

sessSendLabel.setVisible(true);

sessSendField.setVisible(true);

sessSureButton.setVisible(true);

sessCancelButton.setVisible(true);

sessSendField.setText(sessNameField.getText() + ", your next cricketing session with Mr. Arif has
been set for " + sessDateField.getText()

    + " at " + sessTimeBox.getSelectedItem() + ".\nOther information pertaining to your session
is as follows:\n" + sessInfoField.getText()

    + "\nIf you have any questions regarding this information, please contact Mr. Arif through
his private email. Thank you.");

try {

    reWrite();

} catch (ParseException ex) {

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

}

reboot();

}

```

```

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

    // Sends email using the mailMerge class

    String email = sessEmailField.getText();

    String sub = "MasterClass Cricket Academy Session Update";

    String body = sessSendField.getText();

    mailMerge.sendMail(email, sub, body);

    output("The email has been sent");

    sessNameField.setText("");

    sessEmailField.setText("");

    sessDateField.setText("");

    sessTimeBox.setSelectedIndex(0);

    sessInfoField.setText("");

    sessPayField.setText("");

```

```

sessSendField.setText("");
sessSendLabel.setVisible(false);
sessSendField.setVisible(false);
sessSureButton.setVisible(false);
sessCancelButton.setVisible(false);
}

```

```

private void sessCancelButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // Cancels email option
    sessNameField.setText("");
    sessEmailField.setText("");
    sessDateField.setText("");
    sessTimeBox.setSelectedIndex(0);
    sessInfoField.setText("");
    sessPayField.setText("");
    sessSendField.setText("");
    sessSendLabel.setVisible(false);
    sessSendField.setVisible(false);
    sessSureButton.setVisible(false);
    sessCancelButton.setVisible(false);
}

```

```

private void infoPassButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // Checks to see if password entered matches current password. This allows you to proceed to
    change password
    Connection con = null;
    Statement st = null;
    ResultSet rs = null;
    String url = "jdbc:mysql://localhost:3306/CricketDatabase";
    String user = "root";
    String password = "";

```

```

try {
    con = DriverManager.getConnection(url, user, password);
    st = con.createStatement();
    rs = st.executeQuery("select * from Login");
    rs.absolute(1);
    if (rs.getString(2).equals(infoPassField.getText())) {
        infoNPassField.setVisible(true);
        infoCPassField.setVisible(true);
        infoNPassLabel.setVisible(true);
        infoCPassLabel.setVisible(true);
        infoNPassButton.setVisible(true);
        infoPassField.setText("");
    } else {
        output("This is not the current password");
        infoPassField.setText("");
    }
} catch (SQLException ex) {
    Logger lgr = Logger.getLogger(tester.class.getName());
    lgr.log(Level.SEVERE, ex.getMessage(), ex);
} finally {
    try {
        if (rs != null) {
            rs.close();
        }
        if (st != null) {
            st.close();
        }
        if (con != null) {
            con.close();
        }
    } catch (SQLException ex) {

```

```

        Logger lgr = Logger.getLogger(tester.class.getName());

        lgr.log(Level.WARNING, ex.getMessage(), ex);
    }
}
}

```

```

private void infoNPassButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // Checks whether the two new passwords match, if they do then the new password is added to
    the database

    Connection con = null;
    Statement st = null;
    ResultSet rs = null;
    String url = "jdbc:mysql://localhost:3306/CricketDatabase";
    String user = "root";
    String password = "";
    try {
        con = DriverManager.getConnection(url, user, password);
        st = con.createStatement();
        rs = st.executeQuery("select Username from Login");
        if (rs.absolute(1)) {
            String userN = rs.getString(1);
            if (infoNPassField.getText().equals(infoCPassField.getText())) {
                st.executeUpdate("truncate Login");
                st.executeUpdate("insert into Login values('" + userN + "','" + infoCPassField.getText() +
                "')");
                output("Your password has been saved");
                infoNPassField.setText("");
                infoCPassField.setText("");
                infoNPassField.setVisible(false);
                infoCPassField.setVisible(false);
                infoNPassLabel.setVisible(false);
                infoCPassLabel.setVisible(false);
            }
        }
    } catch (SQLException ex) {
        Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

```

        infoNPassButton.setVisible(false);
    } else {
        output("The two passwords you have entered do not match. Please confirm the same
password.");
        infoNPassField.setText("");
        infoCPassField.setText("");
    }
}
} catch (SQLException ex) {
    Logger lgr = Logger.getLogger(tester.class.getName());
    lgr.log(Level.SEVERE, ex.getMessage(), ex);
} finally {
    try {
        if (rs != null) {
            rs.close();
        }
        if (st != null) {
            st.close();
        }
        if (con != null) {
            con.close();
        }
    } catch (SQLException ex) {
        Logger lgr = Logger.getLogger(tester.class.getName());
        lgr.log(Level.WARNING, ex.getMessage(), ex);
    }
}
}
}

```

```

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

    // Button used to check whether or not name exists. This is taken alongside other data. If name
is found relevant data is automatically loaded into the fields

```



```

int flag = 0;
for (int count = 0; count < index; count++) {
    if (people[count].getName().equals(formNameField.getText())) {
        formEmailField.setText(people[count].getEmail());
        formPhoField.setText(people[count].getPhoNum());
        formEmerField.setText(people[count].getEmerNum());
        formMedField.setText(people[count].getMedical());
        flag = 1;
        break;
    }
}
if (flag == 0) {
    output("This pupil does not exist. Please enter another name");
}
}

```

```

private void formCancelButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // Button used to cancel form creation feature
    formNameField.setText("");
    formDateField.setText("");
    formTimeBox.setSelectedIndex(0);
    formEmailField.setText("");
    formPhoField.setText("");
    formEmerField.setText("");
    formMedField.setText("");
    formInjuryField.setText("");
}

```

```

private void formSaveButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // Button used to save the emergency form for the relevant pupil
    DateFormat fDate = new SimpleDateFormat("dd/MM/yyyy", Locale.ENGLISH);
}

```

```

try {
    verifyDate(fDate.parse(formDateField.getText()));
} catch (ParseException ex) {
    Logger.getLogger(Application.class.getName()).log(Level.SEVERE, null, ex);
}

if (vflag == 1) {
    output("Please enter a proper date");
    vflag = 0;
} else {
    String form = "Date: " + formDateField.getText() + "\n" + formNameField.getText() + "
recieved an injury at " + formTimeBox.getSelectedItem() + ".\n"
        + "Contact Information:\nPupil Email: " + formEmailField.getText() + "\nPupil Phone
Number: " + formPhoField.getText()
        + "\nPupil Emergency Number: " + formEmerField.getText() + "\nMedical Information:\n"
+ rep(formMedField.getText()) + "\nInjury Details:"
        + rep(formInjuryField.getText()) + "\n";

    Connection con = null;
    Statement st = null;
    ResultSet rs = null;

    String url = "jdbc:mysql://localhost:3306/CricketDatabase";
    String user = "root";
    String password = "";

    try {
        con = DriverManager.getConnection(url, user, password);
        st = con.createStatement();
        rs = st.executeQuery("select * from forms where name='" + formNameField.getText() + "'");
        if (rs.absolute(1)) {
            form = form + rs.getString(2);
        }

        st.executeUpdate("delete from forms where Name='" + formNameField.getText() + "'");
        //All forms are then saved as one entry
    }
}

```

```

        st.executeUpdate("insert into forms values('" + formNameField.getText() + "','" + form +
        ""));

        formNameField.setText("");
        formDateField.setText("");
        formTimeBox.setSelectedIndex(0);
        formEmailField.setText("");
        formPhoField.setText("");
        formEmerField.setText("");
        formMedField.setText("");
        formInjuryField.setText("");

        output("The Emergency Injury Form has been added to the database");
    } catch (SQLException ex) {

        Logger lgr = Logger.getLogger(tester.class.getName());

        lgr.log(Level.SEVERE, ex.getMessage(), ex);
    } finally {
        try {
            if (rs != null) {
                rs.close();
            }

            if (st != null) {
                st.close();
            }

            if (con != null) {
                con.close();
            }
        } catch (SQLException ex) {

            Logger lgr = Logger.getLogger(tester.class.getName());

            lgr.log(Level.WARNING, ex.getMessage(), ex);
        }
    }
}

```

```
}
```

```
private void formsEnterButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    // Used to access forms created for pupils  
    int flag = 0;  
    if (accessNameField.getText().equals("")) {  
        output("Please enter a name in the field");  
    }  
    for (int count = 0; count < index; count++) {  
        if (people[count].getName().equals(accessNameField.getText())) {  
            flag = 1;  
        }  
    }  
    if (flag == 1) {  
        Connection con = null;  
        Statement st = null;  
        ResultSet rs = null;  
        String url = "jdbc:mysql://localhost:3306/CricketDatabase";  
        String user = "root";  
        String password = "";  
        try {  
            con = DriverManager.getConnection(url, user, password);  
            st = con.createStatement();  
            rs = st.executeQuery("select * from forms where name='" + accessNameField.getText() +  
""");  
            if (rs.absolute(1)) {  
                accessFormField.setText(rs.getString(2));  
            }  
            if (accessFormField.getText().equals("")) {  
                output("This pupil does not have any previous forms");  
            }  
        }  
    }  
}
```

```

    } catch (SQLException ex) {

        Logger lgr = Logger.getLogger(tester.class.getName());

        lgr.log(Level.SEVERE, ex.getMessage(), ex);

    } finally {

        try {

            if (rs != null) {

                rs.close();

            }

            if (st != null) {

                st.close();

            }

            if (con != null) {

                con.close();

            }

        } catch (SQLException ex) {

            Logger lgr = Logger.getLogger(tester.class.getName());

            lgr.log(Level.WARNING, ex.getMessage(), ex);

        }

    }

} else {

    output("This name was not found in the database. Please enter a valid name");

    accessNameField.setText("");

}

}

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

    // Button used to clear search for forms for any one pupil
    h query

    accessNameField.setText("");

    accessFormField.setText("");

}

```

```

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

    // Button used for client to update their username for the database

    Connection con = null;

    Statement st = null;

    ResultSet rs = null;

    String url = "jdbc:mysql://localhost:3306/CricketDatabase";

    String user = "root";

    String password = "";

    boolean check = false;

    try {

        con = DriverManager.getConnection(url, user, password);

        st = con.createStatement();

        rs = st.executeQuery("select Passcode from Login");

        if (rs.absolute(1)) {

            String passN = rs.getString(1);

            st.executeUpdate("truncate Login");

            st.executeUpdate("insert into Login values('" + infoUserField.getText() + "','" + passN + "')");

            output("The username has been changed");

            infoUserField.setText("");

        }

    } catch (SQLException ex) {

        Logger lgr = Logger.getLogger(tester.class.getName());

        lgr.log(Level.SEVERE, ex.getMessage(), ex);

    } finally {

        try {

            if (rs != null) {

                rs.close();

            }

            if (st != null) {

                st.close();

            }

        }

    }

}

```

```

    }

    if (con != null) {
        con.close();
    }
} catch (SQLException ex) {
    Logger lgr = Logger.getLogger(tester.class.getName());
    lgr.log(Level.WARNING, ex.getMessage(), ex);
}
}
}

```

```

private void infoSecurityButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // Button used to set up a security option in case login credentials are forgotten. In which case
    information is emailed

    if (infoSecurityCombo.getSelectedIndex() == 0) {
        output("Please select a question for security");
    } else if (infoSecurityField.getText().equals("")) {
        output("Please enter an answer to the chosen question");
    } else {
        output("Thank you, your response has been saved");

        Connection con = null;
        Statement st = null;
        ResultSet rs = null;

        String url = "jdbc:mysql://localhost:3306/CricketDatabase";
        String user = "root";
        String password = "";

        try {
            con = DriverManager.getConnection(url, user, password);
            st = con.createStatement();
            st.executeUpdate("truncate Forgot");
            st.executeUpdate("insert into Forgot values('true', '" +
infoSecurityCombo.getSelectedItem() + "', '"

```

```

        + infoSecurityField.getText() + """);
    } catch (SQLException ex) {

        Logger lgr = Logger.getLogger(tester.class.getName());

        lgr.log(Level.SEVERE, ex.getMessage(), ex);
    } finally {

        try {

            if (rs != null) {

                rs.close();

            }

            if (st != null) {

                st.close();

            }

            if (con != null) {

                con.close();

            }

        } catch (SQLException ex) {

            Logger lgr = Logger.getLogger(tester.class.getName());

            lgr.log(Level.WARNING, ex.getMessage(), ex);

        }

    }

    infoSecurityCombo.setSelectedIndex(0);

    infoSecurityField.setText("");

}
}

```

```

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

    // button used to upload a file into the database and save it

    int flag = 0;

    if (docNameField.getText().equals("") || docFileField.getText().equals("") ||
    docPathField.getText().equals("")) {

        output("You must enter all the information");
    }
}

```



```

} else {

    File doc = new File(docPathField.getText());

    if (!doc.exists()) {

        output("FilePath is invalid. Please enter a correct file path.");

    } else {

        for (int count = 0; count < index; count++) {

            if (people[count].getName().equals(docNameField.getText())) {

                Connection con = null;

                Statement st = null;

                ResultSet rs = null;

                String url = "jdbc:mysql://localhost:3306/CricketDatabase";

                String user = "root";

                String password = "";

                try {

                    con = DriverManager.getConnection(url, user, password);

                    st = con.createStatement();

                    String temp = docPathField.getText().replace("\\", "##");

                    st.executeUpdate("insert into Documents values('" + docNameField.getText() +

                        "','" + docFileField.getText() + "','" + temp + "')");

                    flag = 1;

                    output("The file has been added to the database");

                    docNameField.setText("");

                    docFileField.setText("");

                    docPathField.setText("");

                } catch (SQLException ex) {

                    Logger lgr = Logger.getLogger(tester.class.getName());

                    lgr.log(Level.SEVERE, ex.getMessage(), ex);

                } finally {

                    try {

                        if (rs != null) {

                            rs.close();

                        }

                    }

                }

            }

        }

    }

}

```



```

Connection con = null;

Statement st = null;

ResultSet rs = null;

String url = "jdbc:mysql://localhost:3306/CricketDatabase";

String user = "root";

String password = "";

try {

    con = DriverManager.getConnection(url, user, password);

    st = con.createStatement();

    rs = st.executeQuery("select * from Documents where Name ='" +
docAccessField.getText() + "'");

    rs.last();

    if(rs.last()==false){

        output("There are no documents saved for this pupil");

    }

    for (int i = 1; i < rs.getRow() + 1; i++) {

        rs.absolute(i);

        row[0] = rs.getString(2);

        model.addRow(row);

    }

    flag = 1;

} catch (SQLException ex) {

    Logger lgr = Logger.getLogger(tester.class.getName());

    lgr.log(Level.SEVERE, ex.getMessage(), ex);

} finally {

    try {

        if (rs != null) {

            rs.close();

        }

        if (st != null) {

            st.close();

        }

    }

}

```

```

    }
    if (con != null) {
        con.close();
    }
} catch (SQLException ex) {
    Logger lgr = Logger.getLogger(tester.class.getName());
    lgr.log(Level.WARNING, ex.getMessage(), ex);
}
}
}
}
if (flag == 0) {
    output("The name was not found");
}
}
}

```

```

private void docClearButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // Button used to clear search for documents
    DefaultTableModel model = (DefaultTableModel) docAccessTable.getModel();
    model.setNumRows(0);
    docAccessField.setText("");
    gauge = 0;
}

```

```

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

```

```

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

```

```

        docNameField.setText("");
        docFileField.setText("");
        docPathField.setText("");
    }

    public static void main(String args[]) {
        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(Application.class.getName()).log(java.util.logging.Level.SEVERE,
                null, ex);

        } catch (InstantiationException ex) {

            java.util.logging.Logger.getLogger(Application.class.getName()).log(java.util.logging.Level.SEVERE,
                null, ex);

        } catch (IllegalAccessException ex) {

            java.util.logging.Logger.getLogger(Application.class.getName()).log(java.util.logging.Level.SEVERE,
                null, ex);

        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

            java.util.logging.Logger.getLogger(Application.class.getName()).log(java.util.logging.Level.SEVERE,
                null, ex);

        }

        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Application().setVisible(true);
            }
        }
    }

```

```
});  
}
```

//Method used to connect and begin loading data from database into objects

```
public void access() throws ParseException {  
    index = 0;  
    String temp[] = new String[15];  
    boolean tempe;  
    Connection con = null;  
    Statement st = null;  
    ResultSet rs = null;  
    ResultSet rt = null;  
    String url = "jdbc:mysql://localhost:3306/CricketDatabase";  
    String user = "root";  
    String password = "";  
    int rowCount = 0;  
    try {  
        con = sqlDriver.connectDatabase();  
        con = DriverManager.getConnection(url, user, password);  
        st = con.createStatement();  
        rt = st.executeQuery("select count(*) from Pupils");  
        if (rt.next()) {  
            rowCount = (rt.getInt(1));  
        }  
        rs = st.executeQuery("select * from Pupils");  
        for (int count = 1; count <= rowCount; count++) {  
            if (rs.absolute(count)) {  
                for (int i = 2; i <= 16; i++) {  
                    if (!rs.getString(i).equals("null")) {  
                        temp[i - 2] = rs.getString(i);  
                    } else {
```

```

        temp[i - 2] = "";
    }
}
}
if (temp[3].equals("Male")) {
    tempe = true;
} else {
    tempe = false;
}
DateFormat format = new SimpleDateFormat("yyyy-MM-dd", Locale.ENGLISH);
String datt = temp[2];
java.util.Date dat = format.parse(datt);
people[count - 1] = new Pupil(temp[0], Integer.parseInt(temp[1]), dat, tempe, temp[4],
temp[5]);
people[count - 1].setEmerNum(temp[6]);
if (temp[7].equals("Right")) {
    tempe = true;
} else {
    tempe = false;
}
people[count - 1].setHand(tempe);
datt = temp[8];
dat = format.parse(datt);
people[count - 1].setSessDate(dat);
people[count - 1].setSessTime(temp[9]);
people[count - 1].setSessInfo(temp[10]);
people[count - 1].setExp(temp[11]);
people[count - 1].setMedical(temp[12]);
people[count - 1].setPay(temp[13]);
people[count - 1].setPaySess(Integer.parseInt(temp[14]));
index++;

```

```

    }
} catch (SQLException ex) {
    Logger lgr = Logger.getLogger(tester.class.getName());
    lgr.log(Level.SEVERE, ex.getMessage(), ex);
} finally {
    try {
        if (rs != null) {
            rs.close();
        }
        if (st != null) {
            st.close();
        }
        if (con != null) {
            con.close();
        }
    } catch (SQLException ex) {
        Logger lgr = Logger.getLogger(tester.class.getName());
        lgr.log(Level.WARNING, ex.getMessage(), ex);
    }
}
}

```

//Method used to complete loading data into the program, using table method to load the data

```

public void reboot() {
    try {
        access();
    } catch (ParseException ex) {
        Logger.getLogger(Application.class.getName()).log(Level.SEVERE, null, ex);
    }
    DefaultTableModel model = (DefaultTableModel) mainTable.getModel();
    DefaultTableModel mode = (DefaultTableModel) sessTable.getModel();
}

```



```

Object[] row = new String[6];
Object[] ro = new String[5];
String temp;
model.setNumRows(0);
mode.setNumRows(0);
int income = 0;
for (int count = 0; count < index; count++) {
    row[0] = people[count].getName();
    row[1] = people[count].getAge() + "";
    if (people[count].getGender() == true) {
        temp = "Male";
    } else {
        temp = "Female";
    }
    row[2] = temp;
    SimpleDateFormat fDate = new SimpleDateFormat("dd/MM/yyyy");
    row[3] = (fDate.format(people[count].getSessDate()));
    row[4] = people[count].getEmail();
    row[5] = people[count].getPhoNum();
    model.addRow(row);
    ro[0] = people[count].getName();
    ro[1] = (fDate.format(people[count].getSessDate()));
    ro[2] = people[count].getSessTime() + "";
    ro[3] = people[count].getPaySess() + "";
    income = income + people[count].getPaySess();
    ro[4] = people[count].getPhoNum();
    mode.addRow(ro);
}
searchTextField.setText("");
infoIncomeLabel.setText("Company Esitimated Income:  " + income);
}

```

```

//Method used to make changes into the database and update it accordingly
public void reWrite() throws ParseException {
    Connection con = null;
    Statement st = null;
    ResultSet rs = null;
    String url = "jdbc:mysql://localhost:3306/CricketDatabase";
    String user = "root";
    String password = "";
    try {
        con = DriverManager.getConnection(url, user, password);
        st = con.createStatement();
        st.executeUpdate("truncate Pupils");
        for (int count = 0; count < index; count++) {
            String gen = "", hand = "";
            java.util.Date utilDate = new SimpleDateFormat("EEE MMM d HH:mm:ss zzz
yyyy").parse(people[count].getDOB() + "");
            java.sql.Date dateB = new java.sql.Date(utilDate.getTime());
            java.sql.Date dateS = null;
            if (people[count].getSessDate() != null) {
                java.util.Date utilDatea = new SimpleDateFormat("EEE MMM d HH:mm:ss zzz
yyyy").parse(people[count].getSessDate() + "");
                dateS = new java.sql.Date(utilDatea.getTime());
            }
            if (people[count].getGender() == true) {
                gen = "Male";
            } else {
                gen = "Female";
            }
            if (people[count].getHand() == true) {
                hand = "Right";
            } else if (people[count].getHand() == false) {

```

```

        hand = "Left";
    }

    st.executeUpdate("insert into Pupils values(" + (count + 1) + "," +
rep(people[count].getName()) + "," + people[count].getAge() + "," + dateB + "," +
        + gen + "," + rep(people[count].getEmail()) + "," + people[count].getPhoNum() + "," +
+ people[count].getEmerNum() + "," +
        + hand + "," + dateS + "," + people[count].getSessTime() + "," +
rep(people[count].getSessInfo()) + "," +
        + rep(people[count].getExp()) + "," + rep(people[count].getMedical()) + "," +
rep(people[count].getPay()) + "," + people[count].getPaySess()
        + ")");
    }

} catch (SQLException ex) {

    Logger lgr = Logger.getLogger(tester.class.getName());

    lgr.log(Level.SEVERE, ex.getMessage(), ex);

} finally {

    try {

        if (rs != null) {

            rs.close();

        }

        if (st != null) {

            st.close();

        }

        if (con != null) {

            con.close();

        }

    } catch (SQLException ex) {

        Logger lgr = Logger.getLogger(tester.class.getName());

        lgr.log(Level.WARNING, ex.getMessage(), ex);

    }

}

}

```

```

//Method used to open seperate frame for editing one pupil
public void profileStart() {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Profile().setVisible(true);
        }
    });
}

//Method used to make sure dates are coherent
public void verifyDate(Date dat) {
    DateFormat fDate = new SimpleDateFormat("dd/MM/yyyy", Locale.ENGLISH);
    String tempDate = fDate.format(dat);
    if (Integer.parseInt(tempDate.substring(0, 2)) < 1 || Integer.parseInt(tempDate.substring(0, 2))
> 31
        || Integer.parseInt(tempDate.substring(3, 5)) < 1 || Integer.parseInt(tempDate.substring(3,
5)) > 12
        || Integer.parseInt(tempDate.substring(6)) < 1900) {
        vflag = 1;
    }
}

//Method used to actually open selected documents from the program
public void openDoc(String file) {
    Connection con = null;
    Statement st = null;
    ResultSet rs = null;
    String url = "jdbc:mysql://localhost:3306/CricketDatabase";
    String user = "root";
    String password = "";
    try {
        con = DriverManager.getConnection(url, user, password);
        st = con.createStatement();
    }
}

```

```

        rs = st.executeQuery("select * from Documents where Name ='" + docAccessField.getText() +
        "' AND FileName = '" + file + "'");

        rs.absolute(1);

        String temp = rs.getString(3);

        temp = temp.replace("##", "\\");

        File docs = new File(temp);

        if (gauge == 0) {

            try {

                Desktop.getDesktop().open(docs);

                gauge = 1;

            } catch (IOException ex) {

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

            }

        }

    } catch (SQLException ex) {

        Logger lgr = Logger.getLogger(tester.class.getName());

        lgr.log(Level.SEVERE, ex.getMessage(), ex);

    } finally {

        try {

            if (rs != null) {

                rs.close();

            }

            if (st != null) {

                st.close();

            }

            if (con != null) {

                con.close();

            }

        }

    } catch (SQLException ex) {

        Logger lgr = Logger.getLogger(tester.class.getName());

        lgr.log(Level.WARNING, ex.getMessage(), ex);

```

```

    }
}

public String rep(String rep) {
    try{
        if(rep.contains(""))
            return rep.replaceAll("", "");
        else
            return rep;
    } catch(NullPointerException e){}
    return rep;
}

// Variables declaration - do not modify
private javax.swing.JButton accessClearButton;
private javax.swing.JTextArea accessFormField;
private javax.swing.JTextField accessNameField;
private javax.swing.JToggleButton addBatLToggle;
private javax.swing.JLabel addBatLabel;
private javax.swing.JToggleButton addBatRToggle;
private javax.swing.JTextField addDOBField;
private javax.swing.JLabel addDOBLLabel;
private javax.swing.JTextField addEmailField;
private javax.swing.JLabel addEmailLabel;
private javax.swing.JLabel addEmeNumLabel;
private javax.swing.JTextField addEmerNumField;
private javax.swing.JTextArea addExpField;
private javax.swing.JLabel addExpLabel;
private javax.swing.JLabel addGenLabel;
private javax.swing.JTextArea addMedField;
private javax.swing.JLabel addMedLabel;
private javax.swing.JTextField addNameField;

```

```
private javax.swing.JLabel addNameLabel;
private javax.swing.JTextArea addPayField;
private javax.swing.JLabel addPayLabel;
private javax.swing.JTextField addPaySessField;
private javax.swing.JLabel addPaySessLabel;
private javax.swing.JTextField addPhoNumField;
private javax.swing.JLabel addPhoNumLabel;
private javax.swing.JButton addResetButton;
private javax.swing.JButton addSaveButton;
private javax.swing.JTextField addSessDateField;
private javax.swing.JLabel addSessDateLabel;
private javax.swing.JTextArea addSessInfoField;
private javax.swing.JLabel addSessInfoLabel;
private javax.swing.JPanel addTab;
private javax.swing.JComboBox<String> addTimeBox;
private javax.swing.JLabel addTimeLabel;
private javax.swing.JToggleButton ageToggleSearch;
private javax.swing.JButton cancelButton;
private javax.swing.JTextField docAccessField;
private javax.swing.JTable docAccessTable;
private javax.swing.JButton docCancelButton;
private javax.swing.JButton docClearButton;
private javax.swing.JTextField docFileField;
private javax.swing.JTextField docNameField;
private javax.swing.JTextField docPathField;
private javax.swing.JTabbedPane docsPane;
private javax.swing.JPanel editorTab;
private javax.swing.JToggleButton emailToggleSearch;
private javax.swing.JToggleButton femaleToggleButton;
private javax.swing.JButton fileAccessField;
private javax.swing.JButton fileUploadField;
```

```
private javax.swing.JButton formCancelButton;
private javax.swing.JTextField formDateField;
private javax.swing.JTextField formEmailField;
private javax.swing.JTextField formEmerField;
private javax.swing.JTextArea formInjuryField;
private javax.swing.JTextArea formMedField;
private javax.swing.JButton formNameButton;
private javax.swing.JTextField formNameField;
private javax.swing.JTextField formPhoField;
private javax.swing.JButton formSaveButton;
private javax.swing.JPanel formTab;
private javax.swing.JComboBox<String> formTimeBox;
private javax.swing.JButton formsEnterButton;
private javax.swing.JToggleButton genToggleSearch;
private javax.swing.JLabel infoCEMailLabel;
private javax.swing.JPasswordField infoCPassField;
private javax.swing.JLabel infoCPassLabel;
private javax.swing.JLabel infoDateLabel;
private javax.swing.JLabel infoIncomeLabel;
private javax.swing.JLabel infoMEmailLabel;
private javax.swing.JButton infoNPassButton;
private javax.swing.JPasswordField infoNPassField;
private javax.swing.JLabel infoNPassLabel;
private javax.swing.JLabel infoNameLabel;
private javax.swing.JLabel infoNumLabel;
private javax.swing.JButton infoPassButton;
private javax.swing.JPasswordField infoPassField;
private javax.swing.JLabel infoPassLabel;
private javax.swing.JButton infoSecurityButton;
private javax.swing.JComboBox<String> infoSecurityCombo;
private javax.swing.JTextField infoSecurityField;
```



```
private javax.swing.JLabel infoSecurityLabel;
private javax.swing.JPanel infoTab;
private javax.swing.JButton infoUserButton;
private javax.swing.JTextField infoUserField;
private javax.swing.JLabel infoUserLabel;
private javax.swing.JButton jButton5;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel14;
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel16;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JPanel jPanel4;
private javax.swing.JPanel jPanel5;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane10;
private javax.swing.JScrollPane jScrollPane12;
private javax.swing.JScrollPane jScrollPane13;
```

```
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JScrollPane jScrollPane3;
private javax.swing.JScrollPane jScrollPane4;
private javax.swing.JScrollPane jScrollPane5;
private javax.swing.JScrollPane jScrollPane6;
private javax.swing.JScrollPane jScrollPane7;
private javax.swing.JScrollPane jScrollPane8;
private javax.swing.JScrollPane jScrollPane9;
private javax.swing.JTabbedPane jTabbedPane1;
private javax.swing.JTabbedPane jTabbedPane2;
private javax.swing.JTextArea jTextArea4;
private javax.swing.JTextField jTextField10;
private javax.swing.JTextField jTextField8;
private javax.swing.JButton mainBoot;
private javax.swing.JPanel mainTab;
private javax.swing.JTable mainTable;
private javax.swing.JToggleButton maleToggleButton;
private javax.swing.JToggleButton nameToggleSearch;
private javax.swing.JToggleButton numToggleSearch;
private javax.swing.JButton removeButton;
private javax.swing.JTextField removeField;
private javax.swing.JLabel removeLabel;
private javax.swing.JPanel removeTab;
private javax.swing.JButton searchButton;
private javax.swing.JLabel searchLabel;
private javax.swing.JTextField searchTextField;
private javax.swing.JButton sessCancelButton;
private javax.swing.JTextField sessDateField;
private javax.swing.JLabel sessDateLabel;
private javax.swing.JTextField sessEmailField;
private javax.swing.JLabel sessEmailLabel;
```

```

private javax.swing.JButton sessEnterButton;
private javax.swing.JTextArea sessInfoField;
private javax.swing.JLabel sessInfoLabel;
private javax.swing.JTextField sessNameField;
private javax.swing.JLabel sessNameLabel;
private javax.swing.JTextField sessPayField;
private javax.swing.JLabel sessPayLabel;
private javax.swing.JButton sessSaveButton;
private javax.swing.JTextArea sessSendField;
private javax.swing.JLabel sessSendLabel;
private javax.swing.JButton sessSureButton;
private javax.swing.JTable sessTable;
private javax.swing.JComboBox<String> sessTimeBox;
private javax.swing.JLabel sessTimeLabel;
private javax.swing.JPanel sessionTab;
private javax.swing.JLabel starLabel;
private javax.swing.JTabbedPane subEditorTab;
private javax.swing.JScrollPane tableScrollPane;

// End of variables declaration

//Methods used to show dialog boxes and messages, for either taking input or simply outputting a
message

static String input(String prompt) {
    return javax.swing.JOptionPane.showInputDialog(null, prompt);
}

static void output(String message) {
    javax.swing.JOptionPane.showMessageDialog(null, message);
}

private static class formTimeField {
    public formTimeField() {

```

```

    }

    private static void addKeyListener(KeyAdapter keyAdapter) {

    }

}
}

```

Profile GUI:

```

//Imports use for various methods in the program

import static csia.Application.people;

import java.sql.*;

import java.text.*;

import java.util.Locale;

import java.util.logging.*;

import javax.swing.ImageIcon;


public class Profile extends javax.swing.JFrame {


    //Initialise some global variables which will be used throughout the program for various finctions

    public static String param;

    public boolean gen;

    public boolean hand;

    ImageIcon img = new ImageIcon(getClass().getResource("MCA.PNG"));


    public Profile() {

        initComponents();

        //Method used to load data about relevant object or pupil into all the fields

        values();

    }


    @SuppressWarnings("unchecked")

    // <editor-fold defaultstate="collapsed" desc="Generated Code">

```

```
private void initComponents() {

    phoLabel = new javax.swing.JLabel();
    nameLabel = new javax.swing.JLabel();
    emailField = new javax.swing.JTextField();
    nameField = new javax.swing.JTextField();
    emerLabel = new javax.swing.JLabel();
    emailLabel = new javax.swing.JLabel();
    sessInfoLabel = new javax.swing.JLabel();
    ageField = new javax.swing.JTextField();
    ageLabel = new javax.swing.JLabel();
    jScrollPane7 = new javax.swing.JScrollPane();
    sessInfoField = new javax.swing.JTextArea();
    expLabel = new javax.swing.JLabel();
    jScrollPane8 = new javax.swing.JScrollPane();
    expField = new javax.swing.JTextArea();
    medLabel = new javax.swing.JLabel();
    jScrollPane9 = new javax.swing.JScrollPane();
    medField = new javax.swing.JTextArea();
    dobLabel = new javax.swing.JLabel();
    dobField = new javax.swing.JTextField();
    genLabel = new javax.swing.JLabel();
    payInfoLabel = new javax.swing.JLabel();
    jScrollPane1 = new javax.swing.JScrollPane();
    payInfoField = new javax.swing.JTextArea();
    handLabel = new javax.swing.JLabel();
    sessTimeLabel = new javax.swing.JLabel();
    emerField = new javax.swing.JTextField();
    maleToggleButton = new javax.swing.JToggleButton();
    femToggleButton = new javax.swing.JToggleButton();
    rightToggleButton = new javax.swing.JToggleButton();
}
```

```
leftTogButton = new javax.swing.JToggleButton();
phoField = new javax.swing.JTextField();
sessDateLabel = new javax.swing.JLabel();
sessDateField = new javax.swing.JTextField();
paySessLabel = new javax.swing.JLabel();
paySessField = new javax.swing.JTextField();
exitButton = new javax.swing.JButton();
saveButton = new javax.swing.JButton();
resetButton = new javax.swing.JButton();
timeBox = new javax.swing.JComboBox<>();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
setTitle("MCA Pupil Profile");
setBackground(java.awt.SystemColor.info);
setForeground(new java.awt.Color(255, 255, 204));
setIconImage(img.getImage());
setPreferredSize(new java.awt.Dimension(730, 650));

phoLabel.setForeground(new java.awt.Color(0, 0, 102));
phoLabel.setText("Phone number:");

nameLabel.setForeground(new java.awt.Color(0, 0, 102));
nameLabel.setText("Name:");

emailField.setForeground(new java.awt.Color(0, 0, 102));

nameField.setForeground(new java.awt.Color(0, 0, 102));

emerLabel.setForeground(new java.awt.Color(0, 0, 102));
emerLabel.setText("Emergency number:");
```

```
emailLabel.setForeground(new java.awt.Color(0, 0, 102));  
emailLabel.setText("Email:");
```

```
sessInfoLabel.setForeground(new java.awt.Color(0, 0, 102));  
sessInfoLabel.setText("Next Session Information:");
```

```
ageField.setForeground(new java.awt.Color(0, 0, 102));
```

```
ageLabel.setForeground(new java.awt.Color(0, 0, 102));  
ageLabel.setText("Age:");
```

```
jScrollPane7.setForeground(new java.awt.Color(0, 0, 102));
```

```
sessInfoField.setColumns(20);  
sessInfoField.setFont(new java.awt.Font("Monospaced", 0, 11)); // NOI18N  
sessInfoField.setRows(5);  
jScrollPane7.setViewportViewView(sessInfoField);
```

```
expLabel.setForeground(new java.awt.Color(0, 0, 102));  
expLabel.setText("Cricketing Experience:");
```

```
jScrollPane8.setForeground(new java.awt.Color(0, 0, 102));
```

```
expField.setColumns(20);  
expField.setFont(new java.awt.Font("Monospaced", 0, 11)); // NOI18N  
expField.setRows(5);  
jScrollPane8.setViewportViewView(expField);
```

```
medLabel.setForeground(new java.awt.Color(0, 0, 102));  
medLabel.setText("Medical Information:");
```

```
jScrollPane9.setForeground(new java.awt.Color(0, 0, 102));
```

```
medField.setColumns(20);
```

```
medField.setFont(new java.awt.Font("Monospaced", 0, 11)); // NOI18N
```

```
medField.setRows(5);
```

```
jScrollPane9.setViewportViewView(medField);
```

```
dobLabel.setForeground(new java.awt.Color(0, 0, 102));
```

```
dobLabel.setText("DOB(dd/mm/yyyy):");
```

```
dobField.setForeground(new java.awt.Color(0, 0, 102));
```

```
genLabel.setForeground(new java.awt.Color(0, 0, 102));
```

```
genLabel.setText("Gender:");
```

```
payInfoLabel.setForeground(new java.awt.Color(0, 0, 102));
```

```
payInfoLabel.setText("Payment Information");
```

```
jScrollPane1.setForeground(new java.awt.Color(0, 0, 102));
```

```
payInfoField.setColumns(20);
```

```
payInfoField.setFont(new java.awt.Font("Monospaced", 0, 11)); // NOI18N
```

```
payInfoField.setRows(5);
```

```
jScrollPane1.setViewportViewView(payInfoField);
```

```
handLabel.setForeground(new java.awt.Color(0, 0, 102));
```

```
handLabel.setText("Batting Hand:");
```

```
sessTimeLabel.setForeground(new java.awt.Color(0, 0, 102));
```

```
sessTimeLabel.setText("Next Session Time:");
```



```
emerField.setForeground(new java.awt.Color(0, 0, 102));
```

```
maleTogButton.setForeground(new java.awt.Color(0, 0, 102));
```

```
maleTogButton.setText("Male");
```

```
maleTogButton.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        maleTogButtonActionPerformed(evt);
```

```
    }
```

```
});
```

```
femTogButton.setForeground(new java.awt.Color(0, 0, 102));
```

```
femTogButton.setText("Female");
```

```
femTogButton.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        femTogButtonActionPerformed(evt);
```

```
    }
```

```
});
```

```
rightTogButton.setForeground(new java.awt.Color(0, 0, 102));
```

```
rightTogButton.setText("Right");
```

```
rightTogButton.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        rightTogButtonActionPerformed(evt);
```

```
    }
```

```
});
```

```
leftTogButton.setForeground(new java.awt.Color(0, 0, 102));
```

```
leftTogButton.setText("Left");
```

```
leftTogButton.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        leftTogButtonActionPerformed(evt);
```

```

    }
});

phoField.setForeground(new java.awt.Color(0, 0, 102));

sessDateLabel.setForeground(new java.awt.Color(0, 0, 102));
sessDateLabel.setText("Next Session Date:");

sessDateField.setForeground(new java.awt.Color(0, 0, 102));

paySessLabel.setForeground(new java.awt.Color(0, 0, 102));
paySessLabel.setText("Pay per Session:");

paySessField.setForeground(new java.awt.Color(0, 0, 102));

exitButton.setForeground(new java.awt.Color(0, 0, 102));
exitButton.setText("Exit");
exitButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        exitButtonActionPerformed(evt);
    }
});

saveButton.setForeground(new java.awt.Color(0, 0, 102));
saveButton.setText("Save & Exit");
saveButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        saveButtonActionPerformed(evt);
    }
});

```

```

resetButton.setForeground(new java.awt.Color(0, 0, 102));

resetButton.setText("Reset");

resetButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        resetButtonActionPerformed(evt);
    }
});

```

```

timeBox.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Select Time",
"00:00", "00:15", "00:30", "00:45", "01:00", "01:15", "01:30", "01:45", "02:00", "02:15", "02:30",
"02:45", "03:00", "03:15", "03:30", "03:45", "04:00", "04:15", "04:30", "04:45", "05:00", "05:15",
"05:30", "05:45", "06:00", "06:15", "06:30", "06:45", "07:00", "07:15", "07:30", "07:45", "08:00",
"08:15", "08:30", "08:45", "09:00", "09:15", "09:30", "09:45", "10:00", "10:15", "10:30", "10:45",
"11:00", "11:15", "11:30", "11:45", "12:00", "12:15", "12:30", "12:45", "13:00", "13:15", "13:30",
"13:45", "14:00", "14:15", "14:30", "14:45", "15:00", "15:15", "15:30", "15:45", "16:00", "16:15",
"16:30", "16:45", "17:00", "17:15", "17:30", "17:45", "18:00", "18:15", "18:30", "18:45", "19:00",
"19:15", "19:30", "19:45", "20:00", "20:15", "20:30", "20:45", "21:00", "21:15", "21:30", "21:45",
"22:00", "22:15", "22:30", "22:45", "23:00", "23:15", "23:30", "23:45" }));

```

```

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

layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(15, 15, 15)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(handLabel)
                .addGroup(layout.createSequentialGroup()
                    .addGap(15, 15, 15)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addComponent(sessDateLabel)
                        .addComponent(sessTimeLabel)
                        .addGap(28, 28, 28)
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)

```

```

        .addComponent(paySessField)

        .addComponent(sessDateField, javax.swing.GroupLayout.DEFAULT_SIZE, 134,
Short.MAX_VALUE)

        .addComponent(timeBox, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)))

    .addComponent(paySessLabel)

    .addGroup(layout.createSequentialGroup())

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

        .addComponent(phoLabel)

        .addComponent(emerLabel)

        .addComponent(nameLabel)

        .addComponent(ageLabel)

        .addComponent(dobLabel)

        .addComponent(emailLabel)

        .addComponent(genLabel))

    .addGap(22, 22, 22)

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

        .addGroup(layout.createSequentialGroup()

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

                .addGroup(layout.createSequentialGroup()

                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING,
false)

                        .addComponent(dobField)

                        .addComponent(ageField)

                        .addComponent(nameField, javax.swing.GroupLayout.PREFERRED_SIZE, 130,
javax.swing.GroupLayout.PREFERRED_SIZE))

                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING,
false)

                        .addComponent(phoField)

                        .addComponent(emerField, javax.swing.GroupLayout.DEFAULT_SIZE, 130,
Short.MAX_VALUE))

                    .addComponent(emailField, javax.swing.GroupLayout.PREFERRED_SIZE, 130,
javax.swing.GroupLayout.PREFERRED_SIZE))

                .addGap(28, 28, 28))

```

```

        .addGroup(layout.createSequentialGroup())
            .addComponent(maleToggleButton, javax.swing.GroupLayout.PREFERRED_SIZE, 65,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(femToggleButton))
        .addGroup(layout.createSequentialGroup())
            .addComponent(leftToggleButton, javax.swing.GroupLayout.PREFERRED_SIZE, 57,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(rightToggleButton))))
    .addGap(18, 18, Short.MAX_VALUE)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(resetButton)
            .addGap(18, 18, 18)
            .addComponent(saveButton)
            .addGap(18, 18, 18)
            .addComponent(exitButton))
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING,
false)
            .addComponent(jScrollPane9, javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jScrollPane8, javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(sessInfoLabel, javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jScrollPane7, javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(medLabel, javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(payInfoLabel, javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(expLabel, javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 305,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGap(28, 28, 28))
    );
    layout.setVerticalGroup(

```

```

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

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(nameLabel)

.addComponent(nameField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(sessInfoLabel))

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

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

.addGroup(layout.createSequentialGroup())

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

.addComponent(ageLabel)

.addComponent(ageField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

.addComponent(dobLabel)

.addComponent(dobField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

.addComponent(genLabel)

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

.addComponent(maleToggleButton, javax.swing.GroupLayout.PREFERRED_SIZE, 23,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(femToggleButton))))

.addComponent(jScrollPane7, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

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

.addComponent(expLabel)

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

```

```
.addGroup(layout.createSequentialGroup())

    .addGap(6, 6, 6)

    .addComponent(jScrollPane8, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

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

    .addComponent(medLabel)

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

    .addComponent(jScrollPane9, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

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

    .addComponent(payInfoLabel)

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

    .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

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

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

        .addComponent(resetButton)

        .addComponent(saveButton)

        .addComponent(exitButton))

    .addContainerGap(38, Short.MAX_VALUE))

    .addGroup(layout.createSequentialGroup())

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

            .addComponent(emailLabel)

            .addComponent(emailField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

                .addComponent(phoLabel)

                .addComponent(phoField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

                .addComponent(emerLabel)
```

```

        .addComponent(emerField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(18, 18, 18)

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

            .addComponent(handLabel)

            .addComponent(rightToggleButton)

            .addComponent(leftToggleButton))

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

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

            .addComponent(sessDateLabel)

            .addComponent(sessDateField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

            .addComponent(sessTimeLabel)

            .addComponent(timeBox, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

            .addComponent(paySessField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(paySessLabel))

        .addGap(0, 0, Short.MAX_VALUE))))

    );

```

```

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

```

```

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

    //Toggle button used to select pupil gender

    gen = true;

}

```



```
private void femTogButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    //Toggle button used to select pupil gender  
    gen = false;  
}
```

```
private void leftTogButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    //Toggle button used to select pupil batting hand  
    hand = false;  
}
```

```
private void rightTogButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    //Toggle button used to select pupil batting hand  
    hand = true;  
}
```

```
private void resetButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    //Button used to return fields to their original state  
    values();  
}
```

```
private void saveButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    // Button used to update object after changes have been made  
    Connection con = null;  
    Statement st = null;  
    ResultSet rs = null;  
    String url = "jdbc:mysql://localhost:3306/CricketDatabase";  
    String user = "root";  
    String password = "";  
    int rowNum = 0;  
    try {
```

```

con = DriverManager.getConnection(url, user, password);

st = con.createStatement();

rs = st.executeQuery("select distinct ID from Pupils where Name='" + param + "'");

if (rs.next()) {
    rowNum = (rs.getInt(1));
}

for (int count = 0; count < Application.index; count++) {
    if (Application.people[count].getName().equals(param)) {
        Application.people[count].setName(nameField.getText());
        Application.people[count].setAge(Integer.parseInt(ageField.getText()));
        DateFormat fDate = new SimpleDateFormat("dd/MM/yyyy", Locale.ENGLISH);
        try {
            Application.people[count].setDOB(fDate.parse(dobField.getText()));
        } catch (ParseException ex) {
            Logger.getLogger(Profile.class.getName()).log(Level.SEVERE, null, ex);
        }
        Application.people[count].setGender(gen);
        Application.people[count].setEmail(emailField.getText());
        Application.people[count].setPhoNum(phoField.getText());
        Application.people[count].setEmerNum(emerField.getText());
        Application.people[count].setHand(hand);
        try {
            Application.people[count].setSessDate(fDate.parse(sessDateField.getText()));
        } catch (ParseException ex) {
            Logger.getLogger(Profile.class.getName()).log(Level.SEVERE, null, ex);
        }
        Application.people[count].setSessTime(timeBox.getSelectedItem() + "");
        Application.people[count].setPaySess(Integer.parseInt(paySessField.getText()));
        Application.people[count].setSessInfo(sessInfoField.getText());
        Application.people[count].setExp(expField.getText());
        Application.people[count].setMedical(medField.getText());
    }
}

```

```

        Application.people[count].setPay(payInfoField.getText());

        try {
            reWrite();
        } catch (ParseException ex) {
            Logger.getLogger(Profile.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}

} catch (SQLException ex) {
    Logger lgr = Logger.getLogger(tester.class.getName());
    lgr.log(Level.SEVERE, ex.getMessage(), ex);
} finally {
    try {
        if (rs != null) {
            rs.close();
        }
        if (st != null) {
            st.close();
        }
        if (con != null) {
            con.close();
        }
    } catch (SQLException ex) {
        Logger lgr = Logger.getLogger(tester.class.getName());
        lgr.log(Level.WARNING, ex.getMessage(), ex);
    }
}

this.dispose();
}

```

```

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

```

```

//Exits this frame and goes back to the main one
this.dispose();
}

public static void main(String args[]) {
    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(Profile.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(Profile.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(Profile.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(Profile.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    }
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Profile().setVisible(true);
        }
    });
}

//Method used to rewrite changes and updates back to the database
public void reWrite() throws ParseException {

```

```

Connection con = null;

Statement st = null;

ResultSet rs = null;

String url = "jdbc:mysql://localhost:3306/CricketDatabase";

String user = "root";

String password = "";

try {

    con = DriverManager.getConnection(url, user, password);

    st = con.createStatement();

    st.executeUpdate("truncate Pupils");

    for (int count = 0; count < Application.index; count++) {

        String gen, hand;

        java.util.Date utilDate = new SimpleDateFormat("EEE MMM d HH:mm:ss zzz
yyyy").parse(Application.people[count].getDOB() + "");

        java.sql.Date dateB = new java.sql.Date(utilDate.getTime());

        java.util.Date utilDatea = new SimpleDateFormat("EEE MMM d HH:mm:ss zzz
yyyy").parse(Application.people[count].getSessDate() + "");

        java.sql.Date dateS = new java.sql.Date(utilDatea.getTime());

        if (Application.people[count].getGender() == true) {

            gen = "Male";

        } else {

            gen = "Female";

        }

        if (Application.people[count].getHand() == true) {

            hand = "Right";

        } else {

            hand = "Left";

        }

        st.executeUpdate("insert into Pupils values(" + (count + 1) + ", " + "" +
rep(Application.people[count].getName()) + ", " + "" + Application.people[count].getAge() + ", " + "" + dateB + ", " + ""
+ gen + ", " + "" + rep(Application.people[count].getEmail()) + ", " + "" + Application.people[count].getPhoNum() + ", " + ""
+ Application.people[count].getEmerNum() + ", " + ""

```

```

        + hand + "," + dateS + "," + people[count].getSessTime() + "," +
rep(people[count].getSessInfo()) + ","
        + rep(people[count].getExp()) + "," + rep(people[count].getMedical()) + "," +
rep(people[count].getPay()) + "," + people[count].getPaySess()
        + ")");
    }
} catch (SQLException ex) {
    Logger lgr = Logger.getLogger(tester.class.getName());
    lgr.log(Level.SEVERE, ex.getMessage(), ex);
} finally {
    try {
        if (rs != null) {
            rs.close();
        }
        if (st != null) {
            st.close();
        }
        if (con != null) {
            con.close();
        }
    } catch (SQLException ex) {
        Logger lgr = Logger.getLogger(tester.class.getName());
        lgr.log(Level.WARNING, ex.getMessage(), ex);
    }
}
}

public String rep(String rep) {
    try{
        if(rep.contains(""))
            return rep.replaceAll("", "");
        else
            return rep;
    }
}

```

```
    } catch(NullPointerException e){}

    return rep;
}

// Variables declaration - do not modify
private javax.swing.JTextField ageField;
private javax.swing.JLabel ageLabel;
private javax.swing.JTextField dobField;
private javax.swing.JLabel dobLabel;
private javax.swing.JTextField emailField;
private javax.swing.JLabel emailLabel;
private javax.swing.JTextField emerField;
private javax.swing.JLabel emerLabel;
private javax.swing.JButton exitButton;
private javax.swing.JTextArea expField;
private javax.swing.JLabel expLabel;
private javax.swing.JToggleButton femTogButton;
private javax.swing.JLabel genLabel;
private javax.swing.JLabel handLabel;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane7;
private javax.swing.JScrollPane jScrollPane8;
private javax.swing.JScrollPane jScrollPane9;
private javax.swing.JToggleButton leftTogButton;
private javax.swing.JToggleButton maleTogButton;
private javax.swing.JTextArea medField;
private javax.swing.JLabel medLabel;
private javax.swing.JTextField nameField;
private javax.swing.JLabel nameLabel;
private javax.swing.JTextArea payInfoField;
private javax.swing.JLabel payInfoLabel;
private javax.swing.JTextField paySessField;
```

```

private javax.swing.JLabel paySessLabel;
private javax.swing.JTextField phoField;
private javax.swing.JLabel phoLabel;
private javax.swing.JButton resetButton;
private javax.swing.JToggleButton rightTogButton;
private javax.swing.JButton saveButton;
private javax.swing.JTextField sessDateField;
private javax.swing.JLabel sessDateLabel;
private javax.swing.JTextArea sessInfoField;
private javax.swing.JLabel sessInfoLabel;
private javax.swing.JLabel sessTimeLabel;
private javax.swing.JComboBox<String> timeBox;
// End of variables declaration
//Method used to load values into every field from current pupil or object values
private void values() {
    for (int count = 0; count < Application.index; count++) {
        if (Application.people[count].getName().equals(param)) {
            nameField.setText(Application.people[count].getName());
            ageField.setText(Application.people[count].getAge() + "");
            SimpleDateFormat fDate = new SimpleDateFormat("dd/MM/yyyy");
            dobField.setText(fDate.format(Application.people[count].getDOB()));
            if (Application.people[count].getGender() == true) {
                genLabel.setText("Gender(Currently: Male)");
            } else {
                genLabel.setText("Gender(Currently: Female)");
            }
            emailField.setText(Application.people[count].getEmail());
            phoField.setText(Application.people[count].getPhoNum());
            emerField.setText(Application.people[count].getEmerNum());
            if (Application.people[count].getHand() == true) {
                handLabel.setText("Batting Hand(Currently: Right)");
            }
        }
    }
}

```



```

    } else {
        handLabel.setText("Batting Hand(Currently: Left)");
    }

    sessDateField.setText(fDate.format(Application.people[count].getSessDate()));
    timeBox.setSelectedItem(Application.people[count].getSessTime());
    paySessField.setText(Application.people[count].getPaySess() + "");
    sessInfoField.setText(Application.people[count].getSessInfo());
    expField.setText(Application.people[count].getExp());
    medField.setText(Application.people[count].getMedical());
    payInfoField.setText(Application.people[count].getPay());
}
}
}
}

```

Start Up GUI:

//Imports use for various methods in the program

```
import java.sql.*;
```

```
import java.util.logging.*;
```

```
import javax.swing.ImageIcon;
```

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

```
    //Initialise some global variables which will be used throughout the program for various finctions
```

```
    static boolean safeCondition;
```

```
    static String safeQuestion, safeAnswer;
```

```
    static boolean defCondition = false;
```

```
    ImageIcon img = new ImageIcon(getClass().getResource("MCA.PNG"));
```

```
    public StartUp() {
```

```
        initComponents();
```

```
        //Method used to check if it is first time using database. Default login credentials are set and
        steps to update
    }
}

```

```

//the data are outlined
defaultCheck();

//Method used to check if the security option has been set up, as makes sure that even if login
details are forgotten

//they can be retrieved through validation checks and email
safeCheck();

forgotButton.setVisible(safeCondition);

forgotButton.setEnabled(safeCondition);
}

```

```

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

```

```

    jButton1 = new javax.swing.JButton();
    userLabel = new javax.swing.JLabel();
    userField = new javax.swing.JTextField();
    passLabel = new javax.swing.JLabel();
    passField = new javax.swing.JPasswordField();
    enterButton = new javax.swing.JButton();
    forgotButton = new javax.swing.JButton();
    mainLabel = new javax.swing.JLabel();

```

```

    jButton1.setText("jButton1");

```

```

    setTitle("MCA Login");
    setIconImage(img.getImage());
    getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

```

```

    userLabel.setFont(new java.awt.Font("Tahoma", 3, 14)); // NOI18N
    userLabel.setForeground(new java.awt.Color(251, 5, 5));

```

```

userLabel.setText("Username:");

getContentPane().add(userLabel, new org.netbeans.lib.awtextra.AbsoluteConstraints(110, 80,
80, -1));

userField.setForeground(new java.awt.Color(0, 0, 102));

getContentPane().add(userField, new org.netbeans.lib.awtextra.AbsoluteConstraints(190, 80,
80, 30));

passLabel.setFont(new java.awt.Font("Tahoma", 3, 14)); // NOI18N
passLabel.setForeground(new java.awt.Color(255, 6, 6));
passLabel.setText("Password:");

getContentPane().add(passLabel, new org.netbeans.lib.awtextra.AbsoluteConstraints(110, 120,
80, -1));

passField.setForeground(new java.awt.Color(0, 0, 102));

getContentPane().add(passField, new org.netbeans.lib.awtextra.AbsoluteConstraints(190, 120,
80, 30));

enterButton.setForeground(new java.awt.Color(0, 0, 102));
enterButton.setText("Submit");
enterButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        enterButtonActionPerformed(evt);
    }
});

getContentPane().add(enterButton, new org.netbeans.lib.awtextra.AbsoluteConstraints(180,
160, 90, -1));

forgotButton.setForeground(new java.awt.Color(0, 0, 102));
forgotButton.setText("Forgot username or password?");
forgotButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        forgotButtonActionPerformed(evt);
    }
});

```

```

    }

});

getContentPane().add(forgotButton, new org.netbeans.lib.awtextra.AbsoluteConstraints(180,
210, 210, -1));

mainLabel.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/csia/Cricket_Kids.jpg"))); // NOI18N

getContentPane().add(mainLabel, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, -
1));

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

```

```

private void enterButtonActionPerformed(java.awt.event.ActionEvent evt) {
    //Button used to enter the username and password which are btoh then checked
    if (userField.getText().equals("")) {
        output("You must enter a username");
    } else if (passField.getText().equals("")) {
        output("You must enter a password");
    }

    //Method databaseCheck is used to make sure that entered login details match those in
    database

    } else if (databaseCheck()) {
        appStart();

        if (safeCondition == true) {
            output("Welcome");
        } else {
            output("Welcome. Please set up a security option in the Personal Information tab as this
            will better "
                + "secure your account");
        }

        this.dispose();
    } else {
        output("Invalid username or password");
    }
}

```

```
}  
}
```

```
private void forgotButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    //Button used to retrieve forgotten login credentials  
    String answer = "", email = "", sub = "", body = "";  
    Connection con = null;  
    Statement st = null;  
    ResultSet rs = null;  
    String url = "jdbc:mysql://localhost:3306/CricketDatabase";  
    String user = "root";  
    String password = "";  
    try {  
        con = DriverManager.getConnection(url, user, password);  
        st = con.createStatement();  
        rs = st.executeQuery("select * from Forgot");  
        rs.absolute(1);  
        safeQuestion = rs.getString(2);  
        safeAnswer = rs.getString(3);  
        answer = input("We'll get you your username and password immediately. Firstly, please  
answer your security question\n"  
            + safeQuestion);  
        if (answer.equalsIgnoreCase(safeAnswer)) {  
            email = input("Correct. Now can you enter the email to which you wish to send your login  
details. Be careful to enter your own email.");  
            sub = "Masterclass Database Login Details";  
            body = "Your login credentials are as follows:\n";  
            rs = st.executeQuery("select * from Login");  
            rs.absolute(1);  
            body = body + "Username: " + rs.getString(1) + "\nPassword: " + rs.getString(2) + "\nThank  
you.";  
            //mailMerge class is used to send email to client with credentials
```

```

        mailMerge.sendMail(email, sub, body);
    } else {
        output("This isn't the right answer. Please try again.");
    }
} catch (SQLException ex) {
    Logger lgr = Logger.getLogger(tester.class.getName());
    lgr.log(Level.SEVERE, ex.getMessage(), ex);
} finally {
    try {
        if (rs != null) {
            rs.close();
        }
        if (st != null) {
            st.close();
        }
        if (con != null) {
            con.close();
        }
    } catch (SQLException ex) {
        Logger lgr = Logger.getLogger(tester.class.getName());
        lgr.log(Level.WARNING, ex.getMessage(), ex);
    }
}

}

public static void main(String args[]) {
    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(StartUp.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
} catch (InstantiationException ex) {
    java.util.logging.Logger.getLogger(StartUp.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
} catch (IllegalAccessException ex) {
    java.util.logging.Logger.getLogger(StartUp.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
} catch (javax.swing.UnsupportedLookAndFeelException ex) {
    java.util.logging.Logger.getLogger(StartUp.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
}
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new StartUp().setVisible(true);
    }
});
}

//Methods used to show dialog boxes and messages, for either taking input or simply outputting a
message
public String input(String prompt) {
    return javax.swing.JOptionPane.showInputDialog(null, prompt);
}

public void output(String message) {
    javax.swing.JOptionPane.showMessageDialog(null, message);
}

//Method used to start the actual database if login credentials are correct
public void appStart() {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {

```

```

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

//Method databaseCheck is used to make sure that entered login details match those in database
public boolean databaseCheck() {
    Connection con = null;
    Statement st = null;
    ResultSet rs = null;
    String url = "jdbc:mysql://localhost:3306/CricketDatabase";
    String user = "root";
    String password = "";
    boolean check = false;
    try {
        con = sqlDriver.connectDatabase();
        con = DriverManager.getConnection(url, user, password);
        st = con.createStatement();
        rs = st.executeQuery("select * from Login");
        rs.absolute(1);
        if (rs.getString(1).equals(userField.getText()) && rs.getString(2).equals(passField.getText())) {
            check = true;
        }
        rs.absolute(1);
    } catch (SQLException ex) {
        Logger lgr = Logger.getLogger(tester.class.getName());
        lgr.log(Level.SEVERE, ex.getMessage(), ex);
    } finally {
        try {
            if (rs != null) {
                rs.close();
            }
        }
    }
}

```



```

        if (st != null) {
            st.close();
        }
        if (con != null) {
            con.close();
        }
    } catch (SQLException ex) {
        Logger lgr = Logger.getLogger(tester.class.getName());
        lgr.log(Level.WARNING, ex.getMessage(), ex);
    }
}
return check;
}

```

//Method used to check if the security option has been set up, as makes sure that even if login details are forgotten

//they can be retrieved through validation checks and email

```

public void safeCheck() {
    Connection con = null;
    Statement st = null;
    ResultSet rs = null;
    String url = "jdbc:mysql://localhost:3306/CricketDatabase";
    String user = "root";
    String password = "";
    try {
        con = DriverManager.getConnection(url, user, password);
        st = con.createStatement();
        rs = st.executeQuery("select * from Forgot");
        rs.absolute(1);
        try {
            if (rs.getString(1).equals("true")) {
                safeCondition = true;
            }
        }
    }
}

```

```

        } else {
            safeCondition = false;
        }
    } catch (SQLException e) {
    }
} catch (SQLException ex) {
    Logger lgr = Logger.getLogger(tester.class.getName());
    lgr.log(Level.SEVERE, ex.getMessage(), ex);
} finally {
    try {
        if (rs != null) {
            rs.close();
        }
        if (st != null) {
            st.close();
        }
        if (con != null) {
            con.close();
        }
    } catch (SQLException ex) {
        Logger lgr = Logger.getLogger(tester.class.getName());
        lgr.log(Level.WARNING, ex.getMessage(), ex);
    }
}
}

```

//Method used to check if it is first time using database. Default login credentials are set and steps to update

//the data are outlined

```

public void defaultCheck() {
    Connection con = null;
    Statement st = null;

```

```

ResultSet rs = null;

String url = "jdbc:mysql://localhost:3306/CricketDatabase";

String user = "root";

String password = "";

try {

    con = sqlDriver.connectDatabase();

    con = DriverManager.getConnection(url, user, password);

    st = con.createStatement();

    rs = st.executeQuery("select * from Login");

    if (rs.last() == false) {

        defCondition = true;

        st.executeUpdate("insert into login values('Admin','Admin')");

        output("This is your first login. The username and password are set to: \"Admin\". Please "

            + "change this in the Personal Information tab in the database immediately. Please also

set up a security "

            + "option to secure your account");

    }

} catch (SQLException ex) {

    Logger lgr = Logger.getLogger(tester.class.getName());

    lgr.log(Level.SEVERE, ex.getMessage(), ex);

} finally {

    try {

        if (rs != null) {

            rs.close();

        }

        if (st != null) {

            st.close();

        }

        if (con != null) {

            con.close();

        }

    }

```

```

    } catch (SQLException ex) {

        Logger lgr = Logger.getLogger(tester.class.getName());

        lgr.log(Level.WARNING, ex.getMessage(), ex);

    }

}

// Variables declaration - do not modify
private javax.swing.JButton enterButton;
private javax.swing.JButton forgotButton;
private javax.swing.JButton jButton1;
private javax.swing.JLabel mainLabel;
private javax.swing.JPasswordField passField;
private javax.swing.JLabel passLabel;
private javax.swing.JTextField userField;
private javax.swing.JLabel userLabel;

// End of variables declaration
}

```

Pupil Object Class:

//Import use for methods in the program

```
import java.util.Date;
```

```

public class Pupil {

    //Initialise some global variables which will be used throughout the program for various finctions

    private String name;

    private int age;

    private Date dob;

    private boolean gender;

    private String email;

    private String phoNum;

    private String emerNum;

    private boolean hand;
}

```

```
private Date sessDate;  
private String sessTime;  
private String sessInfo;  
private String exp;  
private String medical;  
private String pay;  
private int paySess;
```

//Constructor method used to create an instance of the object and store data on it

```
Pupil(String nm, int ag, Date db, boolean gen, String em, String pho) {  
    name = nm;  
    age = ag;  
    dob = db;  
    gender = gen;  
    email = em;  
    phoNum = pho;  
}
```

//The accessor and mutator method for each parameter

```
String getName() {  
    return name;  
}
```

```
int getAge() {  
    return age;  
}
```

```
Date getDOB() {  
    return dob;  
}
```

```
boolean getGender() {
```

```
        return gender;
    }
}
```

```
String getEmail() {
    return email;
}
```

```
String getPhoNum() {
    return phoNum;
}
```

```
String getEmerNum() {
    return emerNum;
}
```

```
boolean getHand() {
    return hand;
}
```

```
Date getSessDate() {
    return sessDate;
}
```

```
String getSessTime() {
    return sessTime;
}
```

```
String getSessInfo() {
    return sessInfo;
}
```

```
String getExp() {  
    return exp;  
}
```

```
String getMedical() {  
    return medical;  
}
```

```
String getPay() {  
    return pay;  
}
```

```
int getPaySess() {  
    return paySess;  
}
```

```
void setName(String nm) {  
    name = nm;  
}
```

```
void setAge(int ag) {  
    age = ag;  
}
```

```
void setDOB(Date db) {  
    dob = db;  
}
```

```
void setGender(boolean gen) {  
    gender = gen;  
}
```

```
void setEmail(String em) {  
    email = em;  
}
```

```
void setPhoNum(String pho) {  
    phoNum = pho;  
}
```

```
void setEmerNum(String emer) {  
    emerNum = emer;  
}
```

```
void setHand(boolean hd) {  
    hand = hd;  
}
```

```
void setSessDate(Date sd) {  
    sessDate = sd;  
}
```

```
void setSessTime(String st) {  
    sessTime = st;  
}
```

```
void setSessInfo(String si) {  
    sessInfo = si;  
}
```

```
void setExp(String ex) {  
    exp = ex;  
}
```



```
}
```

```
void setMedical(String med) {  
    medical = med;  
}
```

```
void setPay(String py) {  
    pay = py;  
}
```

```
void setPaySess(int ps) {  
    paySess = ps;  
}  
}
```

MailMerge Class:

//Imports use for various methods in the program

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

```
public class mailMerge {  
    //Initialise some global variables which will be used throughout the program for finctions  
    private static String email_my = "mcadatabase.noreply@gmail.com",  
        password_my = "MasterClass",  
        host_at = "smtp.gmail.com",  
        port_num = "465";  
    //Actual method used to send emails given the necessary parameters  
    public static boolean sendMail(String strTo, String strSubject, String strBody) {  
        Properties property = new Properties();  
        property.put("mail.smtp.user", email_my);  
        property.put("mail.smtp.host", host_at);
```

```

property.put("mail.smtp.port", port_num);
property.put("mail.smtp.starttls.enable", "true");
property.put("mail.smtp.auth", "true");
property.put("mail.smtp.socketFactory.port", port_num);
property.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");
property.put("mail.smtp.socketFactory.fallback", "false");
try {
    Session sess = Session.getDefaultInstance(property, null);
    try (Transport trans = sess.getTransport("smtp")) {
        MimeMessage message = new MimeMessage(sess);
        message.setText(strBody);
        message.setSubject(strSubject);
        message.setFrom(new InternetAddress(email_my));
        message.addRecipient(Message.RecipientType.TO, new InternetAddress(strTo));
        message.saveChanges();
        trans.connect(host_at, email_my, password_my);
        trans.sendMessage(message, message.getAllRecipients());
    }
    return true;
} catch (Exception ex) {
    ex.printStackTrace();
    return false;
}
}
}

```

SQL Driver Class:

//Imports use for various methods in the program

```

import java.sql.*;
import java.util.logging.*;
import javax.swing.table.DefaultTableModel;
import java.text.*;

```

```

import java.util.Locale;

import javax.swing.event.*;

public class sqlDriver {

    //This method is called multiple times throughout the program. It ensures that the database and
    the table exists.

    //If they do not exist then this code creates them with no rows

    public static Connection connectDatabase() throws SQLException {

        Connection con;

        Statement st = null;

        String url = "jdbc:mysql://localhost:3306";

        String user = "root";

        String password = "";

        con = DriverManager.getConnection(url, user, password);

        st = con.createStatement();

        st.executeUpdate("CREATE DATABASE IF NOT EXISTS CricketDatabase");

        st.executeUpdate("CREATE TABLE IF NOT EXISTS CricketDatabase.pupils("

            + "ID int(11) NOT NULL AUTO_INCREMENT,"

            + "Name text,"

            + "Age int(11) DEFAULT NULL,"

            + "DOB date DEFAULT NULL,"

            + "Gender text,"

            + "Email text,"

            + "Phone text,"

            + "Emergency text,"

            + "Hand text,"

            + "SessDate date DEFAULT NULL,"

            + "SessTime text DEFAULT NULL,"

            + "SessInfo text,"

            + "Exp text,"

            + "Medical text,"

```

```

        + "Pay text,"
        + "PaySess int(11) DEFAULT NULL,"
        + "PRIMARY KEY (ID)"
        + "));

st.executeUpdate("CREATE TABLE IF NOT EXISTS CricketDatabase.login("
        + " Username text,"
        + " Passcode text"
        + "));

st.executeUpdate("CREATE TABLE IF NOT EXISTS CricketDatabase.forms("
        + " Name text,"
        + " Form text"
        + "));

st.executeUpdate("CREATE TABLE IF NOT EXISTS CricketDatabase.forgot("
        + " State text,"
        + " Question text,"
        + " Answer text"
        + "));

st.executeUpdate("CREATE TABLE IF NOT EXISTS CricketDatabase.documents("
        + " Name text,"
        + " FileName text,"
        + " FilePath text"
        + "));

return con;
}
}

```