**FEE REPORT JAVA CODE**

package feereport;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.Map;

class MyFrame extends JFrame{

JLabel la;

JPanel p;

JButton a,v,e,d,l;

JPanel buttonPanel;

JPanel textPanel;

JTextArea displayArea;

ArrayList<String> studentList;

Map<String, Double> dueFeeMap; // Store due fee amounts for each student

MyFrame(){

super("fee report");

setLayout(new FlowLayout());

// Create space at the top

JPanel topSpace = new JPanel();

topSpace.setPreferredSize(new Dimension(400, 20)); // Adjust the height as needed

add(topSpace, BorderLayout.NORTH);

// Create a panel for the main content

JPanel contentPanel = new JPanel(); // Create a new panel for content

contentPanel.setLayout(new BorderLayout());

// Create and configure the label for the account section

la = new JLabel("Account Section");

la.setHorizontalAlignment(JLabel.CENTER); // Center align the label

contentPanel.add(la, BorderLayout.NORTH);

a=new JButton("Add Student");

v=new JButton("view Student");

e=new JButton("Edit Student");

d=new JButton("Due Fee");

l=new JButton("Logout");

studentList = new ArrayList<>();

dueFeeMap = new HashMap<>();

// Create a panel for buttons and add buttons to it

buttonPanel = new JPanel();

GridLayout gridLayout = new GridLayout(0, 2);

gridLayout.setVgap(10);

buttonPanel.setLayout(gridLayout);

a.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

v.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

e.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

d.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

l.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

JPanel aPanel = new JPanel();

aPanel.add(a);

JPanel vPanel = new JPanel();

vPanel.add(v);

JPanel ePanel = new JPanel();

ePanel.add(e);

JPanel dPanel = new JPanel();

dPanel.add(d);

JPanel lPanel = new JPanel();

lPanel.add(l);

buttonPanel.add(aPanel);

buttonPanel.add(vPanel);

buttonPanel.add(ePanel);

buttonPanel.add(dPanel);

buttonPanel.add(lPanel);

// Add action listeners to buttons

a.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// Handle the Add Student button click event

String studentName = JOptionPane.showInputDialog("Enter student name:");

if (studentName != null && !studentName.isEmpty()) {

studentList.add(studentName);

JOptionPane.showMessageDialog(MyFrame.this, "Student added: " + studentName);

}

}

});

v.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if (!studentList.isEmpty()) {

StringBuilder studentNames = new StringBuilder();

for (String name : studentList) {

studentNames.append(name).append("\n");

}

JOptionPane.showMessageDialog(MyFrame.this, "Student List:\n" + studentNames.toString());

} else {

JOptionPane.showMessageDialog(MyFrame.this, "No students added yet.");

}

}

});

e.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// Handle the Edit Student button click event

if (studentList.isEmpty()) {

JOptionPane.showMessageDialog(MyFrame.this, "No students added yet.");

return;

}

// Create a dialog to select a student

String[] studentArray = studentList.toArray(new String[0]);

String selectedStudent = (String) JOptionPane.showInputDialog(

MyFrame.this,

"Select a student to edit:",

"Edit Student",

JOptionPane.PLAIN\_MESSAGE,

null,

studentArray,

studentArray[0]

);

if (selectedStudent != null) {

// Get the index of the selected student

int selectedIndex = studentList.indexOf(selectedStudent);

// Create a dialog for editing student information

String updatedName = JOptionPane.showInputDialog("Edit student name:", selectedStudent);

if (updatedName != null && !updatedName.isEmpty()) {

// Update the student's name in the ArrayList

studentList.set(selectedIndex, updatedName);

JOptionPane.showMessageDialog(MyFrame.this, "Student updated: " + updatedName);

}

}

}

});

d.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// Handle the Due Fee button click event

if (studentList.isEmpty()) {

JOptionPane.showMessageDialog(MyFrame.this, "No students added yet.");

return;

}

// Create a dialog to select a student

String[] studentArray = studentList.toArray(new String[0]);

String selectedStudent = (String) JOptionPane.showInputDialog(

MyFrame.this,

"Select a student to set due fee:",

"Due Fee",

JOptionPane.PLAIN\_MESSAGE,

null,

studentArray,

studentArray[0]

);

if (selectedStudent != null) {

// Get the index of the selected student

int selectedIndex = studentList.indexOf(selectedStudent);

// Prompt the user to enter the due fee amount

String dueFeeAmount = JOptionPane.showInputDialog("Enter due fee amount for " + selectedStudent + ":");

if (dueFeeAmount != null && !dueFeeAmount.isEmpty()) {

// Display the due fee amount in a dialog or store it in a data structure

dueFeeMap.put(selectedStudent, Double.parseDouble(dueFeeAmount));

JOptionPane.showMessageDialog(MyFrame.this, "Due fee set for " + selectedStudent + ": " + dueFeeAmount);

}

}

}

});

l.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

System.exit(0);

JOptionPane.showMessageDialog(MyFrame.this, "Logout button clicked.");

}

});

contentPanel.add(buttonPanel, BorderLayout.CENTER);

add(contentPanel, BorderLayout.CENTER);

}

}

public class FeeReport {

public static void main(String[] args) {

SwingUtilities.invokeLater(new Runnable() {

public void run() {

MyFrame f = new MyFrame();

f.setSize(400, 400);

f.setVisible(true);

}

});

}

}