import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class StockStoreManagementSwing {

private DefaultTableModel tableModel;

public StockStoreManagementSwing() {

// Main frame

JFrame frame = new JFrame("Stock Store Management System");

frame.setSize(600, 400);

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

frame.setLayout(new BorderLayout());

// Table for displaying products

String[] columns = {"Product Name", "Quantity", "Price"};

tableModel = new DefaultTableModel(columns, 0);

JTable table = new JTable(tableModel);

JScrollPane tableScrollPane = new JScrollPane(table);

frame.add(tableScrollPane, BorderLayout.CENTER);

// Input panel

JPanel inputPanel = new JPanel(new GridLayout(2, 4, 5, 5));

JTextField nameField = new JTextField();

JTextField quantityField = new JTextField();

JTextField priceField = new JTextField();

JButton addButton = new JButton("Add Product");

JButton updateButton = new JButton("Update Quantity");

inputPanel.add(new JLabel("Name:"));

inputPanel.add(nameField);

inputPanel.add(new JLabel("Quantity:"));

inputPanel.add(quantityField);

inputPanel.add(new JLabel("Price:"));

inputPanel.add(priceField);

inputPanel.add(addButton);

inputPanel.add(updateButton);

frame.add(inputPanel, BorderLayout.SOUTH);

// Add product button action

addButton.addActionListener(e -> {

String name = nameField.getText();

String quantityText = quantityField.getText();

String priceText = priceField.getText();

if (name.isEmpty() || quantityText.isEmpty() || priceText.isEmpty()) {

JOptionPane.showMessageDialog(frame, "All fields are required!", "Error", JOptionPane.ERROR\_MESSAGE);

return;

}

try {

int quantity = Integer.parseInt(quantityText);

double price = Double.parseDouble(priceText);

tableModel.addRow(new Object[]{name, quantity, price});

nameField.setText("");

quantityField.setText("");

priceField.setText("");

JOptionPane.showMessageDialog(frame, "Product added successfully!");

} catch (NumberFormatException ex) {

JOptionPane.showMessageDialog(frame, "Invalid quantity or price format!", "Error", JOptionPane.ERROR\_MESSAGE);

}

});

// Update product quantity button action

updateButton.addActionListener(e -> {

int selectedRow = table.getSelectedRow();

if (selectedRow == -1) {

JOptionPane.showMessageDialog(frame, "Please select a product to update!", "Error", JOptionPane.ERROR\_MESSAGE);

return;

}

String newQuantityText = quantityField.getText();

if (newQuantityText.isEmpty()) {

JOptionPane.showMessageDialog(frame, "Please enter a new quantity!", "Error", JOptionPane.ERROR\_MESSAGE);

return;

}

try {

int newQuantity = Integer.parseInt(newQuantityText);

tableModel.setValueAt(newQuantity, selectedRow, 1);

quantityField.setText("");

JOptionPane.showMessageDialog(frame, "Quantity updated successfully!");

} catch (NumberFormatException ex) {

JOptionPane.showMessageDialog(frame, "Invalid quantity format!", "Error", JOptionPane.ERROR\_MESSAGE);

}

});

// Display the frame

frame.setVisible(true);

}

public static void main(String[] args) {

SwingUtilities.invokeLater(StockStoreManagementSwing::new);

}

}