**SOURCE CODE**

import java.awt.\*;

import java.awt.event.\*;

import java.util.\*;

// Product class to represent individual product items

class Product {

private int id;

private String name;

private int quantity;

private double price;

public Product(int id, String name, int quantity, double price) {

this.id = id;

this.name = name;

this.quantity = quantity;

this.price = price;

}

public int getId() {

return id;

}

public String getName() {

return name;

}

public int getQuantity() {

return quantity;

}

public void setQuantity(int quantity) {

this.quantity = quantity;

}

public double getPrice() {

return price;

}

@Override

public String toString() {

return "ID: " + id + ", Name: " + name + ", Quantity: " + quantity + ", Price: " + price;

}

}

// Inventory class to manage products

class Inventory {

private Map<Integer, Product> products = new HashMap<>();

public void addProduct(Product product) {

products.put(product.getId(), product);

}

public void updateProductQuantity(int productId, int quantity) {

if (products.containsKey(productId)) {

Product product = products.get(productId);

product.setQuantity(quantity);

}

}

public String displayProducts() {

if (products.isEmpty()) {

return "No products in inventory.";

} else {

StringBuilder sb = new StringBuilder();

for (Product product : products.values()) {

sb.append(product).append("\n");

}

return sb.toString();

}

}

public void removeProduct(int productId) {

products.remove(productId);

}

}

// Main class with AWT GUI

public class InventoryManagementSystemAWT extends Frame {

private Inventory inventory = new Inventory();

public InventoryManagementSystemAWT() {

// Frame setup

setTitle("Inventory Management System");

setSize(500, 400);

setLayout(new FlowLayout());

setResizable(false);

// Components

Label label = new Label("Inventory Management System", Label.CENTER);

label.setFont(new Font("Arial", Font.BOLD, 20));

add(label);

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

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

Button displayButton = new Button("Display Products");

Button removeButton = new Button("Remove Product");

Button exitButton = new Button("Exit");

add(addButton);

add(updateButton);

add(displayButton);

add(removeButton);

add(exitButton);

TextArea outputArea = new TextArea(15, 40);

outputArea.setEditable(false);

add(outputArea);

// Event listeners

addButton.addActionListener(e -> {

Dialog dialog = new Dialog(this, "Add Product", true);

dialog.setLayout(new FlowLayout());

dialog.setSize(300, 200);

Label idLabel = new Label("ID:");

TextField idField = new TextField(10);

Label nameLabel = new Label("Name:");

TextField nameField = new TextField(20);

Label quantityLabel = new Label("Quantity:");

TextField quantityField = new TextField(10);

Label priceLabel = new Label("Price:");

TextField priceField = new TextField(10);

Button submitButton = new Button("Submit");

submitButton.addActionListener(ae -> {

try {

int id = Integer.parseInt(idField.getText());

String name = nameField.getText();

int quantity = Integer.parseInt(quantityField.getText());

double price = Double.parseDouble(priceField.getText());

inventory.addProduct(new Product(id, name, quantity, price));

outputArea.setText("Product added successfully!\n" + inventory.displayProducts());

dialog.dispose();

} catch (NumberFormatException ex) {

outputArea.setText("Invalid input. Please try again.");

}

});

dialog.add(idLabel);

dialog.add(idField);

dialog.add(nameLabel);

dialog.add(nameField);

dialog.add(quantityLabel);

dialog.add(quantityField);

dialog.add(priceLabel);

dialog.add(priceField);

dialog.add(submitButton);

dialog.setVisible(true);

});

updateButton.addActionListener(e -> {

Dialog dialog = new Dialog(this, "Update Quantity", true);

dialog.setLayout(new FlowLayout());

dialog.setSize(300, 150);

Label idLabel = new Label("Product ID:");

TextField idField = new TextField(10);

Label quantityLabel = new Label("New Quantity:");

TextField quantityField = new TextField(10);

Button submitButton = new Button("Submit");

submitButton.addActionListener(ae -> {

try {

int id = Integer.parseInt(idField.getText());

int quantity = Integer.parseInt(quantityField.getText());

inventory.updateProductQuantity(id, quantity);

outputArea.setText("Quantity updated successfully!\n" + inventory.displayProducts());

dialog.dispose();

} catch (NumberFormatException ex) {

outputArea.setText("Invalid input. Please try again.");

}

});

dialog.add(idLabel);

dialog.add(idField);

dialog.add(quantityLabel);

dialog.add(quantityField);

dialog.add(submitButton);

dialog.setVisible(true);

});

displayButton.addActionListener(e -> outputArea.setText(inventory.displayProducts()));

removeButton.addActionListener(e -> {

Dialog dialog = new Dialog(this, "Remove Product", true);

dialog.setLayout(new FlowLayout());

dialog.setSize(300, 150);

Label idLabel = new Label("Product ID:");

TextField idField = new TextField(10);

Button submitButton = new Button("Submit");

submitButton.addActionListener(ae -> {

try {

int id = Integer.parseInt(idField.getText());

inventory.removeProduct(id);

outputArea.setText("Product removed successfully!\n" + inventory.displayProducts());

dialog.dispose();

} catch (NumberFormatException ex) {

outputArea.setText("Invalid input. Please try again.");

}

});

dialog.add(idLabel);

dialog.add(idField);

dialog.add(submitButton);

dialog.setVisible(true);

});

exitButton.addActionListener(e -> System.exit(0));

// Window close handler

addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

System.exit(0);

}

});

}

public static void main(String[] args) {

EventQueue.invokeLater(() -> {

InventoryManagementSystemAWT ims = new InventoryManagementSystemAWT();

ims.setVisible(true);

});

}

}