```
import java.awt.*;
import java.awt.event.*;
import java.util.ArrayList;
class Item {
int id;
String name;
int quantity;
double price;
public Item(int id, String name, int quantity, double price) {
this.id = id;
this.name = name;
this.quantity = quantity;
this.price = price;
}
@Override
public String toString() {
return "ID: " + id + ", Name: " + name + ", Quantity: " + quantity + ",
Price: " + price;
}
}
public class InventoryManagementSystemAWT extends Frame {
ArrayList<Item> inventory = new ArrayList<>();
TextArea displayArea;
TextField idField, nameField, quantityField, priceField;
Label messageLabel;
public InventoryManagementSystemAWT() {
// Layout setup
setLayout(new FlowLayout());
// Title
Label title = new Label("Inventory Management System");
title.setFont(new Font("Arial", Font.BOLD, 16));
add(title);
// Input fields
add(new Label("ID:"));
idField = new TextField(10);
add(idField);
add(new Label("Name:"));
nameField = new TextField(15);
add(nameField);
add(new Label("Quantity:"));
quantityField = new TextField(5);
add(quantityField);
add(new Label("Price:"));
priceField = new TextField(7);
add(priceField);
// Buttons
Button addButton = new Button("Add Item");
Button updateButton = new Button("Update Item");
```

```
Button deleteButton = new Button("Delete Item");
Button displayButton = new Button("Display Items");
Button exitButton = new Button("Exit");
add(addButton);
add(updateButton);
add(deleteButton);
add(displayButton);
add(exitButton);
// Display area
displayArea = new TextArea(10, 50);
displayArea.setEditable(false);
add(displayArea);
// Message label
messageLabel = new Label(" ");
messageLabel.setForeground(Color.RED);
add(messageLabel);
// Button actions
addButton.addActionListener(e -> addItem());
updateButton.addActionListener(e -> updateItem());
deleteButton.addActionListener(e -> deleteItem());
displayButton.addActionListener(e -> displayItems());
exitButton.addActionListener(e -> System.exit(0));
// Frame settings
setSize(600, 400);
setTitle("Inventory Management System");
setVisible(true);
// Close window action
addWindowListener(new WindowAdapter() {
public void windowClosing(WindowEvent e) {
System.exit(0);
}
});
}
private void addItem() {
try {
int id = Integer.parseInt(idField.getText());
String name = nameField.getText();
int quantity = Integer.parseInt(quantityField.getText());
double price = Double.parseDouble(priceField.getText());
inventory.add(new Item(id, name, quantity, price));
messageLabel.setText("Item added successfully!");
clearFields();
} catch (Exception ex) {
messageLabel.setText("Error: Invalid input!");
}
}
private void updateItem() {
try {
```

```
int id = Integer.parseInt(idField.getText());
boolean found = false;
for (Item item : inventory) {
if (item.id == id) {
int quantity = Integer.parseInt(quantityField.getText());
double price = Double.parseDouble(priceField.getText());
item.quantity = quantity;
item.price = price;
messageLabel.setText("Item updated successfully!");
found = true;
break;
}
}
if (!found) {
messageLabel.setText("Item not found!");
}
clearFields();
} catch (Exception ex) {
messageLabel.setText("Error: Invalid input!");
}
}
private void deleteItem() {
try {
int id = Integer.parseInt(idField.getText());
boolean found = false;
for (int i = 0; i < inventory.size(); i++) {
if (inventory.get(i).id == id) {
inventory.remove(i);
messageLabel.setText("Item deleted successfully!");
found = true;
break;
}
}
if (!found) {
messageLabel.setText("Item not found!");
}
clearFields();
} catch (Exception ex) {
messageLabel.setText("Error: Invalid input!");
}
}
private void displayItems() {
if (inventory.isEmpty()) {
displayArea.setText("No items in the inventory.");
} else {
StringBuilder builder = new StringBuilder();
for (Item item : inventory) {
builder.append(item).append("\n");
```

```
}
displayArea.setText(builder.toString());
}
}
private void clearFields() {
idField.setText("");
nameField.setText("");
quantityField.setText("");
priceField.setText("");
}
public static void main(String[] args) {
new InventoryManagementSystemAWT();
}
}
```