Main Class: Main.java

**import** javax.swing.\*;

**import** java.awt.\*;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.sql.\*;

**import** java.util.ArrayList;

**import** java.util.List;

**class** ProductFrame **extends** JFrame {

**private** ProductDAO productDAO; // Instance of ProductDAO for database interaction

**private** JTextField idField;

**private** JTextField nameField;

**private** JTextField priceField;

**private** JTextField quantityField;

**private** JTextArea productListArea;

**public** ProductFrame() {

**super**("Product Management");

productDAO = ProductDAOImp.*getInstance*(); // Initialize ProductDAO

// Create UI components and layout

idField = **new** JTextField(20);

nameField = **new** JTextField(20);

priceField = **new** JTextField(20);

quantityField = **new** JTextField(20);

productListArea = **new** JTextArea(10, 30);

productListArea.setEditable(**false**);

// Create labels for the text fields

JLabel idLabel = **new** JLabel("ID:");

JLabel nameLabel = **new** JLabel("Name:");

JLabel priceLabel = **new** JLabel("Price:");

JLabel quantityLabel = **new** JLabel("Quantity:");

// Create buttons for CRUD operations

JButton createButton = **new** JButton("Create Product");

JButton showButton = **new** JButton("Show Products");

JButton updateButton = **new** JButton("Update Product");

JButton deleteButton = **new** JButton("Delete Product");

// Add action listeners for buttons

createButton.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

// Get user input from text fields

String name = nameField.getText();

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

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

Product prod = **new** Product(0, name, price, quantity);

productDAO.createProduct(prod);

// Update UI or display success message

JOptionPane.*showMessageDialog*(ProductFrame.**this**, "Product created successfully!");

}

});

showButton.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

// Fetch all products and display them

List<Product> products = productDAO.getAllProducts();

productListArea.setText("");

**for** (Product product : products) {

//productListArea.append(product.toString() + "\n");

System.***out***.println(product.toString() + "\n");

}

}

});

updateButton.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

// Get user input from text fields

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

String name = nameField.getText();

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

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

// Create a new product object

Product product = **new** Product(id, name, price, quantity);

// Call updateProduct method from ProductDAO

productDAO.updateProduct(product);

// Update UI or display success message

JOptionPane.*showMessageDialog*(ProductFrame.**this**, "Product updated successfully!");

}

});

deleteButton.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

// Get product ID to delete

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

// Call deleteProduct method from ProductDAO

productDAO.deleteProduct(id);

// Update UI or display success message

JOptionPane.*showMessageDialog*(ProductFrame.**this**, "Product deleted successfully!");

}

});

// Set layout and add components to the frame using GridBagLayout

setLayout(**new** GridBagLayout());

GridBagConstraints gbc = **new** GridBagConstraints();

gbc.insets = **new** Insets(5, 5, 5, 5);

gbc.gridx = 0;

gbc.gridy = 0;

add(idLabel, gbc);

gbc.gridx = 1;

gbc.gridy = 0;

add(idField, gbc);

gbc.gridx = 0;

gbc.gridy = 1;

add(nameLabel, gbc);

gbc.gridx = 1;

gbc.gridy = 1;

add(nameField, gbc);

gbc.gridx = 0;

gbc.gridy = 2;

add(priceLabel, gbc);

gbc.gridx = 1;

gbc.gridy = 2;

add(priceField, gbc);

gbc.gridx = 0;

gbc.gridy = 3;

add(quantityLabel, gbc);

gbc.gridx = 1;

gbc.gridy = 3;

add(quantityField, gbc);

gbc.gridx = 0;

gbc.gridy = 4;

add(createButton, gbc);

gbc.gridx = 1;

gbc.gridy = 4;

add(showButton, gbc);

gbc.gridx = 0;

gbc.gridy = 5;

add(updateButton, gbc);

gbc.gridx = 1;

gbc.gridy = 5;

add(deleteButton, gbc);

gbc.gridx = 0;

gbc.gridy = 6;

gbc.gridwidth = 2;

add(**new** JScrollPane(productListArea), gbc);

pack(); // Adjust frame size based on components

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***); // Close the application when the frame is closed

setVisible(**true**);

}

}

**public** **class** Main {

**public** **static** **void** main(String[] args) {

ProductFrame frame = **new** ProductFrame();

}

}

ProductDAOImpClass: ProductDAOImp.java

**import** java.sql.\*;

**import** java.util.ArrayList;

**import** java.util.List;

**interface** ProductDAO {

Product createProduct(Product product);

Product getProductById(**int** id);

List<Product> getAllProducts();

**void** updateProduct(Product product);

**void** deleteProduct(**int** id);

}

**public** **class** ProductDAOImp **implements** ProductDAO {

**private** **static** ProductDAOImp *instance*;

**private** **static** **final** String ***URL*** = "jdbc:mysql://localhost:3306/pos\_database";

**private** **static** **final** String ***USER*** = "root";

**private** **static** **final** String ***PASSWORD*** = "";

**private** ProductDAOImp() {}

**public** **static** ProductDAOImp getInstance() {

**if** (*instance* == **null**) {

*instance* = **new** ProductDAOImp();

}

**return** *instance*;

}

**private** Connection getConnection() **throws** SQLException {

**return** DriverManager.*getConnection*(***URL***, ***USER***, ***PASSWORD***);

}

@Override

**public** Product createProduct(Product product) {

String sql = "INSERT INTO products (name, price, quantity) VALUES (?, ?, ?)";

**try** (Connection conn = getConnection();

PreparedStatement ps = conn.prepareStatement(sql, Statement.***RETURN\_GENERATED\_KEYS***)) {

ps.setString(1, product.getName());

ps.setDouble(2, product.getPrice());

ps.setInt(3, product.getQuantity());

ps.executeUpdate();

**try** (ResultSet rs = ps.getGeneratedKeys()) {

**if** (rs.next()) {

product.setId(rs.getInt(1));

}

}

} **catch** (SQLException e) {

e.printStackTrace();

}

**return** product;

}

@Override

**public** Product getProductById(**int** id) {

Product product = **null**;

String sql = "SELECT \* FROM products WHERE id = ?";

**try** (Connection conn = getConnection();

PreparedStatement ps = conn.prepareStatement(sql)) {

ps.setInt(1, id);

**try** (ResultSet rs = ps.executeQuery()) {

**if** (rs.next()) {

product = **new** Product(rs.getInt("id"), rs.getString("name"), rs.getDouble("price"), rs.getInt("quantity"));

}

}

} **catch** (SQLException e) {

e.printStackTrace();

}

**return** product;

}

@Override

**public** List<Product> getAllProducts() {

List<Product> products = **new** ArrayList<>();

String sql = "SELECT \* FROM products";

**try** (Connection conn = getConnection();

PreparedStatement ps = conn.prepareStatement(sql);

ResultSet rs = ps.executeQuery()) {

**while** (rs.next()) {

products.add(**new** Product(rs.getInt("id"), rs.getString("name"), rs.getDouble("price"), rs.getInt("quantity")));

}

} **catch** (SQLException e) {

e.printStackTrace();

}

**return** products;

}

@Override

**public** **void** updateProduct(Product product) {

String sql = "UPDATE products SET name = ?, price = ?, quantity = ? WHERE id = ?";

**try** (Connection conn = getConnection();

PreparedStatement ps = conn.prepareStatement(sql)) {

ps.setString(1, product.getName());

ps.setDouble(2, product.getPrice());

ps.setInt(3, product.getQuantity());

ps.setInt(4, product.getId());

ps.executeUpdate();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

@Override

**public** **void** deleteProduct(**int** id) {

String sql = "DELETE FROM products WHERE id = ?";

**try** (Connection conn = getConnection();

PreparedStatement ps = conn.prepareStatement(sql)) {

//ps.setInt(1, id);

ps.executeUpdate();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

}

Product Class: Product .java

**public** **class** Product {

**private** **int** id;

**private** String name;

**private** **double** price;

**private** **int** quantity;

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

**this**.id = id;

**this**.name = name;

**this**.price = price;

**this**.quantity = quantity;

}

**public** **int** getId() {

**return** id;

}

**public** String getName() {

**return** name;

}

**public** **double** getPrice() {

**return** price;

}

**public** **int** getQuantity() {

**return** quantity;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** **void** setPrice(**double** price) {

**this**.price = price;

}

**public** String toString() {

**return** "Product [id=" + id + ", name=" + name + ", price=" + price + ", quantity=" + quantity + "]";

}

}