**Inventory Management using JavaFX and MySQL**

1. **Main Java**
2. **Data base**

* Db Connection.java

1. **DAO**

* Matches.java

1. **DTO**

* CustomerDTO.java
* ProductDTO.java
* SupplierDTO.java
* UserDTO.java

1. **UI**

* CurreStockPage.java
* CustomerPage.java
* Dashboard.java
* HomePage.java
* LoginPage.java
* ProductPage.java
* PurchasePage.java
* SalesPage.java
* SupplierPage.java
* UserLogsPage.java
* UsersPage.java

1. **View**

* CurreStockPage.form
* CustomerPage.form
* Dashboard.form
* HomePage.form
* LoginPage.form
* ProductPage.form
* PurchasePage.form
* SalesPage.form
* SupplierPage.form
* UserLogsPage.form
* UsersPage.form

**Application:**

**DataBase:**

**DbConnection.java**

package com.inventory.Database;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.FileWriter;

import java.io.IOException;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

import java.util.Properties;

//Class to retrieve connection for database and login verfication.

public class ConnectionFactory {

static final String driver = "com.mysql.cj.jdbc.Driver";

static final String url = "jdbc:mysql://localhost:3306/inventory";

static String username;

static String password;

Properties prop;

Connection conn = null;

Statement statement = null;

ResultSet resultSet = null;

public ConnectionFactory(){

try {

//Username and Password saved as configurable properties to allow changes without recompilation.

prop = new Properties();

prop.loadFromXML(new FileInputStream("lib/DBCredentials.xml"));

} catch (IOException e) {

e.printStackTrace();

}

username = prop.getProperty("username");

password = prop.getProperty("password");

try {

Class.forName(driver);

conn = DriverManager.getConnection(url, username, password);

statement = conn.createStatement();

} catch (Exception e) {

e.printStackTrace();

}

}

public Connection getConn() {

try {

Class.forName(driver);

conn = DriverManager.getConnection(url, username, password);

System.out.println("Connected successfully.");

} catch (Exception e) {

e.printStackTrace();

}

return conn;

}

//Login verification method

public boolean checkLogin(String username, String password, String userType){

String query = "SELECT \* FROM users WHERE username='"

+ username

+ "' AND password='"

+ password

+ "' AND usertype='"

+ userType

+ "' LIMIT 1";

try {

resultSet = statement.executeQuery(query);

if(resultSet.next()) return true;

} catch (Exception ex) {

ex.printStackTrace();

}

return false;

}

**DTO:**

**CustomerDTO.java**

package com.inventory.DAO;

import com.inventory.DTO.CustomerDTO;

import com.inventory.Database.ConnectionFactory;

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import java.sql.\*;

import java.util.Locale;

import java.util.Vector;

// Data Access Object for Customers

public class CustomerDAO {

Connection conn = null;

PreparedStatement prepStatement= null;

Statement statement = null;

ResultSet resultSet = null;

public CustomerDAO() {

try {

conn = new ConnectionFactory().getConn();

statement = conn.createStatement();

} catch (SQLException e) {

e.printStackTrace();

}

}

// Methods to add new custoemr

public void addCustomerDAO(CustomerDTO customerDTO) {

try {

String query = "SELECT \* FROM customers WHERE fullname='"

+customerDTO.getFullName()

+ "' AND location='"

+customerDTO.getLocation()

+ "' AND phone='"

+customerDTO.getPhone()

+ "'";

resultSet = statement.executeQuery(query);

if (resultSet.next())

JOptionPane.showMessageDialog(null, "Customer already exists.");

else

addFunction(customerDTO);

} catch (SQLException e) {

e.printStackTrace();

}

}

public void addFunction(CustomerDTO customerDTO) {

try {

String query = "INSERT INTO customers VALUES(null,?,?,?,?)";

prepStatement = conn.prepareStatement(query);

prepStatement.setString(1, customerDTO.getCustCode());

prepStatement.setString(2, customerDTO.getFullName());

prepStatement.setString(3, customerDTO.getLocation());

prepStatement.setString(4, customerDTO.getPhone());

prepStatement.executeUpdate();

JOptionPane.showMessageDialog(null, "New customer has been added.");

} catch (SQLException e) {

e.printStackTrace();

}

}

// Method to edit existing customer details

public void editCustomerDAO(CustomerDTO customerDTO) {

try {

String query = "UPDATE customers SET fullname=?,location=?,phone=? WHERE customercode=?";

prepStatement = conn.prepareStatement(query);

prepStatement.setString(1, customerDTO.getFullName());

prepStatement.setString(2, customerDTO.getLocation());

prepStatement.setString(3, customerDTO.getPhone());

prepStatement.setString(4, customerDTO.getCustCode());

prepStatement.executeUpdate();

JOptionPane.showMessageDialog(null, "Customer details have been updated.");

} catch (SQLException e) {

e.printStackTrace();

}

}

// Method to delete existing customer

public void deleteCustomerDAO(String custCode) {

try {

String query = "DELETE FROM customers WHERE customercode='" +custCode+ "'";

statement.executeUpdate(query);

JOptionPane.showMessageDialog(null, "Customer removed.");

} catch (SQLException e) {

e.printStackTrace();

}

}

// Method to retrieve data set to be displayed

public ResultSet getQueryResult() {

try {

String query = "SELECT customercode,fullname,location,phone FROM customers";

resultSet = statement.executeQuery(query);

} catch (SQLException e) {

e.printStackTrace();

}

return resultSet;

}

// Method to retrieve search data

public ResultSet getCustomerSearch(String text) {

try {

String query = "SELECT customercode,fullname,location,phone FROM customers " +

"WHERE customercode LIKE '%"+text+"%' OR fullname LIKE '%"+text+"%' OR " +

"location LIKE '%"+text+"%' OR phone LIKE '%"+text+"%'";

resultSet = statement.executeQuery(query);

} catch (SQLException e) {

e.printStackTrace();

}

return resultSet;

}

public ResultSet getCustName(String custCode) {

try {

String query = "SELECT \* FROM customers WHERE customercode='" +custCode+ "'";

resultSet = statement.executeQuery(query);

} catch (SQLException e) {

e.printStackTrace();

}

return resultSet;

}

public ResultSet getProdName(String prodCode) {

try {

String query = "SELECT productname,currentstock.quantity FROM products " +

"INNER JOIN currentstock ON products.productcode=currentstock.productcode " +

"WHERE currentstock.productcode='" +prodCode+ "'";

resultSet = statement.executeQuery(query);

} catch (SQLException e) {

e.printStackTrace();

}

return resultSet;

}

// Method to display data set in tabular form

public DefaultTableModel buildTableModel(ResultSet resultSet) throws SQLException {

ResultSetMetaData metaData = resultSet.getMetaData();

Vector<String> columnNames = new Vector<String>();

int colCount = metaData.getColumnCount();

for (int col=1; col <= colCount; col++){

columnNames.add(metaData.getColumnName(col).toUpperCase(Locale.ROOT));

}

Vector<Vector<Object>> data = new Vector<Vector<Object>>();

while (resultSet.next()) {

Vector<Object> vector = new Vector<Object>();

for (int col=1; col<=colCount; col++) {

vector.add(resultSet.getObject(col));

}

data.add(vector);

}

return new DefaultTableModel(data, columnNames);

}

}

**ProductDTO.java**

package com.inventory.DAO;

import com.inventory.DTO.ProductDTO;

import com.inventory.Database.ConnectionFactory;

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import java.sql.\*;

import java.util.Locale;

import java.util.Vector;

// Data Access Object for Products, Purchase, Stock and Sales

public class ProductDAO {

Connection conn = null;

PreparedStatement prepStatement = null;

PreparedStatement prepStatement2 = null;

Statement statement = null;

Statement statement2 = null;

ResultSet resultSet = null;

public ProductDAO() {

try {

conn = new ConnectionFactory().getConn();

statement = conn.createStatement();

statement2 = conn.createStatement();

} catch (Exception ex) {

ex.printStackTrace();

}

}

public ResultSet getSuppInfo() {

try {

String query = "SELECT \* FROM suppliers";

resultSet = statement.executeQuery(query);

} catch (Exception e) {

e.printStackTrace();

}

return resultSet;

}

public ResultSet getCustInfo() {

try {

String query = "SELECT \* FROM customers";

resultSet = statement.executeQuery(query);

} catch (Exception e) {

e.printStackTrace();

}

return resultSet;

}

public ResultSet getProdStock() {

try {

String query = "SELECT \* FROM currentstock";

resultSet = statement.executeQuery(query);

} catch (Exception e) {

e.printStackTrace();

}

return resultSet;

}

public ResultSet getProdInfo() {

try {

String query = "SELECT \* FROM products";

resultSet = statement.executeQuery(query);

} catch (Exception e) {

e.printStackTrace();

}

return resultSet;

}

public Double getProdCost(String prodCode) {

Double costPrice = null;

try {

String query = "SELECT costprice FROM products WHERE productcode='" +prodCode+ "'";

resultSet = statement.executeQuery(query);

if (resultSet.next())

costPrice = resultSet.getDouble("costprice");

} catch (Exception e) {

e.printStackTrace();

}

return costPrice;

}

public Double getProdSell(String prodCode) {

Double sellPrice = null;

try {

String query = "SELECT sellprice FROM products WHERE productcode='" +prodCode+ "'";

resultSet = statement.executeQuery(query);

if (resultSet.next())

sellPrice = resultSet.getDouble("sellprice");

} catch (Exception e) {

e.printStackTrace();

}

return sellPrice;

}

String suppCode;

public String getSuppCode(String suppName) {

try {

String query = "SELECT suppliercode FROM suppliers WHERE fullname='" +suppName+ "'";

resultSet = statement.executeQuery(query);

while (resultSet.next()) {

suppCode = resultSet.getString("suppliercode");

}

} catch (SQLException e) {

e.printStackTrace();

}

return suppCode;

}

String prodCode;

public String getProdCode(String prodName) {

try {

String query = "SELECT productcode FROM products WHERE productname='" +prodName+ "'";

resultSet = statement.executeQuery(query);

while (resultSet.next()) {

suppCode = resultSet.getString("productcode");

}

} catch (SQLException e) {

e.printStackTrace();

}

return prodCode;

}

String custCode;

public String getCustCode(String custName) {

try {

String query = "SELECT customercode FROM suppliers WHERE fullname='" +custName+ "'";

resultSet = statement.executeQuery(query);

while (resultSet.next()) {

suppCode = resultSet.getString("customercode");

}

} catch (SQLException e) {

e.printStackTrace();

}

return custCode;

}

// Method to check for availability of stock in Inventory

boolean flag = false;

public boolean checkStock(String prodCode) {

try {

String query = "SELECT \* FROM currentstock WHERE productcode='" +prodCode+ "'";

resultSet = statement.executeQuery(query);

while (resultSet.next()) {

flag = true;

}

} catch (SQLException e) {

e.printStackTrace();

}

return flag;

}

// Methods to add a new product

public void addProductDAO(ProductDTO productDTO) {

try {

String query = "SELECT \* FROM products WHERE productname='"

+ productDTO.getProdName()

+ "' AND costprice='"

+ productDTO.getCostPrice()

+ "' AND sellprice='"

+ productDTO.getSellPrice()

+ "' AND brand='"

+ productDTO.getBrand()

+ "'";

resultSet = statement.executeQuery(query);

if (resultSet.next())

JOptionPane.showMessageDialog(null, "Product has already been added.");

else

addFunction(productDTO);

} catch (SQLException e) {

e.printStackTrace();

}

}

public void addFunction(ProductDTO productDTO) {

try {

String query = "INSERT INTO products VALUES(null,?,?,?,?,?)";

prepStatement = (PreparedStatement) conn.prepareStatement(query);

prepStatement.setString(1, productDTO.getProdCode());

prepStatement.setString(2, productDTO.getProdName());

prepStatement.setDouble(3, productDTO.getCostPrice());

prepStatement.setDouble(4, productDTO.getSellPrice());

prepStatement.setString(5, productDTO.getBrand());

String query2 = "INSERT INTO currentstock VALUES(?,?)";

prepStatement2 = conn.prepareStatement(query2);

prepStatement2.setString(1, productDTO.getProdCode());

prepStatement2.setInt(2, productDTO.getQuantity());

prepStatement.executeUpdate();

prepStatement2.executeUpdate();

JOptionPane.showMessageDialog(null, "Product added and ready for sale.");

} catch (SQLException throwables) {

throwables.printStackTrace();

}

}

// Method to add a new purchase transaction

public void addPurchaseDAO(ProductDTO productDTO) {

try {

String query = "INSERT INTO purchaseinfo VALUES(null,?,?,?,?,?)";

prepStatement = conn.prepareStatement(query);

prepStatement.setString(1, productDTO.getSuppCode());

prepStatement.setString(2, productDTO.getProdCode());

prepStatement.setString(3, productDTO.getDate());

prepStatement.setInt(4, productDTO.getQuantity());

prepStatement.setDouble(5, productDTO.getTotalCost());

prepStatement.executeUpdate();

JOptionPane.showMessageDialog(null, "Purchase log added.");

} catch (SQLException throwables) {

throwables.printStackTrace();

}

String prodCode = productDTO.getProdCode();

if(checkStock(prodCode)) {

try {

String query = "UPDATE currentstock SET quantity=quantity+? WHERE productcode=?";

prepStatement = conn.prepareStatement(query);

prepStatement.setInt(1, productDTO.getQuantity());

prepStatement.setString(2, prodCode);

prepStatement.executeUpdate();

} catch (SQLException throwables) {

throwables.printStackTrace();

}

}

else if (!checkStock(prodCode)) {

try {

String query = "INSERT INTO currentstock VALUES(?,?)";

prepStatement = (PreparedStatement) conn.prepareStatement(query);

prepStatement.setString(1, productDTO.getProdCode());

prepStatement.setInt(2, productDTO.getQuantity());

prepStatement.executeUpdate();

} catch (SQLException throwables) {

throwables.printStackTrace();

}

}

deleteStock();

}

// Method to update existing product details

public void editProdDAO(ProductDTO productDTO) {

try {

String query = "UPDATE products SET productname=?,costprice=?,sellprice=?,brand=? WHERE productcode=?";

prepStatement = (PreparedStatement) conn.prepareStatement(query);

prepStatement.setString(1, productDTO.getProdName());

prepStatement.setDouble(2, productDTO.getCostPrice());

prepStatement.setDouble(3, productDTO.getSellPrice());

prepStatement.setString(4, productDTO.getBrand());

prepStatement.setString(5, productDTO.getProdCode());

String query2 = "UPDATE currentstock SET quantity=? WHERE productcode=?";

prepStatement2 = conn.prepareStatement(query2);

prepStatement2.setInt(1, productDTO.getQuantity());

prepStatement2.setString(2, productDTO.getProdCode());

prepStatement.executeUpdate();

prepStatement2.executeUpdate();

JOptionPane.showMessageDialog(null, "Product details updated.");

} catch (SQLException throwables) {

throwables.printStackTrace();

}

}

// Methods to handle updating of stocks in Inventory upon any transaction made

public void editPurchaseStock(String code, int quantity) {

try {

String query = "SELECT \* FROM currentstock WHERE productcode='" +code+ "'";

resultSet = statement.executeQuery(query);

if(resultSet.next()) {

String query2 = "UPDATE currentstock SET quantity=quantity-? WHERE productcode=?";

prepStatement = conn.prepareStatement(query2);

prepStatement.setInt(1, quantity);

prepStatement.setString(2, code);

prepStatement.executeUpdate();

}

} catch (SQLException throwables) {

throwables.printStackTrace();

}

}

public void editSoldStock(String code, int quantity) {

try {

String query = "SELECT \* FROM currentstock WHERE productcode='" +code+ "'";

resultSet = statement.executeQuery(query);

if(resultSet.next()) {

String query2 = "UPDATE currentstock SET quantity=quantity+? WHERE productcode=?";

prepStatement = conn.prepareStatement(query2);

prepStatement.setInt(1, quantity);

prepStatement.setString(2, code);

prepStatement.executeUpdate();

}

} catch (SQLException throwables) {

throwables.printStackTrace();

}

}

public void deleteStock() {

try {

String query = "DELETE FROM currentstock WHERE productcode NOT IN(SELECT productcode FROM purchaseinfo)";

String query2 = "DELETE FROM salesinfo WHERE productcode NOT IN(SELECT productcode FROM products)";

statement.executeUpdate(query);

statement.executeUpdate(query2);

} catch (SQLException throwables) {

throwables.printStackTrace();

}

}

// Method to permanently delete a product from inventory

public void deleteProductDAO(String code) {

try {

String query = "DELETE FROM products WHERE productcode=?";

prepStatement = conn.prepareStatement(query);

prepStatement.setString(1, code);

String query2 = "DELETE FROM currentstock WHERE productcode=?";

prepStatement2 = conn.prepareStatement(query2);

prepStatement2.setString(1, code);

prepStatement.executeUpdate();

prepStatement2.executeUpdate();

JOptionPane.showMessageDialog(null, "Product has been removed.");

} catch (SQLException e){

e.printStackTrace();

}

deleteStock();

}

public void deletePurchaseDAO(int ID){

try {

String query = "DELETE FROM purchaseinfo WHERE purchaseID=?";

prepStatement = conn.prepareStatement(query);

prepStatement.setInt(1, ID);

prepStatement.executeUpdate();

JOptionPane.showMessageDialog(null, "Transaction has been removed.");

} catch (SQLException e){

e.printStackTrace();

}

deleteStock();

}

public void deleteSaleDAO(int ID) {

try {

String query = "DELETE FROM salesinfo WHERE salesID=?";

prepStatement = conn.prepareStatement(query);

prepStatement.setInt(1, ID);

prepStatement.executeUpdate();

JOptionPane.showMessageDialog(null, "Transaction has been removed.");

} catch (SQLException e){

e.printStackTrace();

}

deleteStock();

}

// Sales transaction handling

public void sellProductDAO(ProductDTO productDTO, String username) {

int quantity = 0;

String prodCode = null;

try {

String query = "SELECT \* FROM currentstock WHERE productcode='" +productDTO.getProdCode()+ "'";

resultSet = statement.executeQuery(query);

while (resultSet.next()) {

prodCode = resultSet.getString("productcode");

quantity = resultSet.getInt("quantity");

}

if (productDTO.getQuantity()>quantity)

JOptionPane.showMessageDialog(null, "Insufficient stock for this product.");

else if (productDTO.getQuantity()<=0)

JOptionPane.showMessageDialog(null, "Please enter a valid quantity");

else {

String stockQuery = "UPDATE currentstock SET quantity=quantity-'"

+productDTO.getQuantity()

+"' WHERE productcode='"

+productDTO.getProdCode()

+"'";

String salesQuery = "INSERT INTO salesinfo(date,productcode,customercode,quantity,revenue,soldby)" +

"VALUES('"+productDTO.getDate()+"','"+productDTO.getProdCode()+"','"+productDTO.getCustCode()+

"','"+productDTO.getQuantity()+"','"+productDTO.getTotalRevenue()+"','"+username+"')";

statement.executeUpdate(stockQuery);

statement.executeUpdate(salesQuery);

JOptionPane.showMessageDialog(null, "Product sold.");

}

} catch (SQLException throwables) {

throwables.printStackTrace();

}

}

// Products data set retrieval for display

public ResultSet getQueryResult() {

try {

String query = "SELECT productcode,productname,costprice,sellprice,brand FROM products ORDER BY pid";

resultSet = statement.executeQuery(query);

} catch (SQLException throwables) {

throwables.printStackTrace();

}

return resultSet;

}

// Purchase table data set retrieval

public ResultSet getPurchaseInfo() {

try {

String query = "SELECT PurchaseID,purchaseinfo.ProductCode,ProductName,Quantity,Totalcost " +

"FROM purchaseinfo INNER JOIN products " +

"ON products.productcode=purchaseinfo.productcode ORDER BY purchaseid;";

resultSet = statement.executeQuery(query);

} catch (SQLException throwables) {

throwables.printStackTrace();

}

return resultSet;

}

// Stock table data set retrieval

public ResultSet getCurrentStockInfo() {

try {

String query = """

SELECT currentstock.ProductCode,products.ProductName,

currentstock.Quantity,products.CostPrice,products.SellPrice

FROM currentstock INNER JOIN products

ON currentstock.productcode=products.productcode;

""";

resultSet = statement.executeQuery(query);

} catch (SQLException throwables) {

throwables.printStackTrace();

}

return resultSet;

}

// Sales table data set retrieval

public ResultSet getSalesInfo() {

try {

String query = """

SELECT salesid,salesinfo.productcode,productname,

salesinfo.quantity,revenue,users.name AS Sold\_by

FROM salesinfo INNER JOIN products

ON salesinfo.productcode=products.productcode

INNER JOIN users

ON salesinfo.soldby=users.username;

""";

resultSet = statement.executeQuery(query);

} catch (SQLException throwables) {

throwables.printStackTrace();

}

return resultSet;

}

// Search method for products

public ResultSet getProductSearch(String text) {

try {

String query = "SELECT productcode,productname,costprice,sellprice,brand FROM products " +

"WHERE productcode LIKE '%"+text+"%' OR productname LIKE '%"+text+"%' OR brand LIKE '%"+text+"%'";

resultSet = statement.executeQuery(query);

} catch (SQLException e) {

e.printStackTrace();

}

return resultSet;

}

public ResultSet getProdFromCode(String text) {

try {

String query = "SELECT productcode,productname,costprice,sellprice,brand FROM products " +

"WHERE productcode='" +text+ "' LIMIT 1";

resultSet = statement.executeQuery(query);

} catch (SQLException e) {

e.printStackTrace();

}

return resultSet;

}

// Search method for sales

public ResultSet getSalesSearch(String text) {

try {

String query = "SELECT salesid,salesinfo.productcode,productname,\n" +

" salesinfo.quantity,revenue,users.name AS Sold\_by\n" +

" FROM salesinfo INNER JOIN products\n" +

" ON salesinfo.productcode=products.productcode\n" +

" INNER JOIN users\n" +

" ON salesinfo.soldby=users.username\n" +

" INNER JOIN customers\n" +

" ON customers.customercode=salesinfo.customercode\n" +

"WHERE salesinfo.productcode LIKE '%"+text+"%' OR productname LIKE '%"+text+"%' " +

"OR users.name LIKE '%"+text+"%' OR customers.fullname LIKE '%"+text+"%' ORDER BY salesid;";

resultSet = statement.executeQuery(query);

} catch (SQLException e) {

e.printStackTrace();

}

return resultSet;

}

// Search method for purchase logs

public ResultSet getPurchaseSearch(String text) {

try {

String query = "SELECT PurchaseID,purchaseinfo.productcode,products.productname,quantity,totalcost " +

"FROM purchaseinfo INNER JOIN products ON purchaseinfo.productcode=products.productcode " +

"INNER JOIN suppliers ON purchaseinfo.suppliercode=suppliers.suppliercode" +

"WHERE PurchaseID LIKE '%"+text+"%' OR productcode LIKE '%"+text+"%' OR productname LIKE '%"+text+"%' " +

"OR suppliers.fullname LIKE '%"+text+"%' OR purchaseinfo.suppliercode LIKE '%"+text+"%' " +

"OR date LIKE '%"+text+"%' ORDER BY purchaseid";

resultSet = statement.executeQuery(query);

} catch (SQLException e) {

e.printStackTrace();

}

return resultSet;

}

public ResultSet getProdName(String code) {

try {

String query = "SELECT productname FROM products WHERE productcode='" +code+ "'";

resultSet = statement.executeQuery(query);

} catch (SQLException throwables) {

throwables.printStackTrace();

}

return resultSet;

}

public String getSuppName(int ID) {

String name = null;

try {

String query = "SELECT fullname FROM suppliers " +

"INNER JOIN purchaseinfo ON suppliers.suppliercode=purchaseinfo.suppliercode " +

"WHERE purchaseid='" +ID+ "'";

resultSet = statement.executeQuery(query);

if (resultSet.next())

name = resultSet.getString("fullname");

} catch (SQLException throwables) {

throwables.printStackTrace();

}

return name;

}

public String getCustName(int ID) {

String name = null;

try {

String query = "SELECT fullname FROM customers " +

"INNER JOIN salesinfo ON customers.customercode=salesinfo.customercode " +

"WHERE salesid='" +ID+ "'";

resultSet = statement.executeQuery(query);

if (resultSet.next())

name = resultSet.getString("fullname");

} catch (SQLException throwables) {

throwables.printStackTrace();

}

return name;

}

public String getPurchaseDate(int ID) {

String date = null;

try {

String query = "SELECT date FROM purchaseinfo WHERE purchaseid='" +ID+ "'";

resultSet = statement.executeQuery(query);

if (resultSet.next())

date = resultSet.getString("date");

} catch (SQLException throwables) {

throwables.printStackTrace();

}

return date;

}

public String getSaleDate(int ID) {

String date = null;

try {

String query = "SELECT date FROM salesinfo WHERE salesid='" +ID+ "'";

resultSet = statement.executeQuery(query);

if (resultSet.next())

date = resultSet.getString("date");

} catch (SQLException throwables) {

throwables.printStackTrace();

}

return date;

}

// Method to display product-related data set in tabular form

public DefaultTableModel buildTableModel(ResultSet resultSet) throws SQLException {

ResultSetMetaData metaData = resultSet.getMetaData();

Vector<String> columnNames = new Vector<String>();

int colCount = metaData.getColumnCount();

for (int col=1; col <= colCount; col++){

columnNames.add(metaData.getColumnName(col).toUpperCase(Locale.ROOT));

}

Vector<Vector<Object>> data = new Vector<Vector<Object>>();

while (resultSet.next()) {

Vector<Object> vector = new Vector<Object>();

for (int col=1; col<=colCount; col++) {

vector.add(resultSet.getObject(col));

}

data.add(vector);

}

return new DefaultTableModel(data, columnNames);

}

}

**SupplierDTO.java:**

package com.inventory.DAO;

import com.inventory.DTO.SupplierDTO;

import com.inventory.Database.ConnectionFactory;

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import java.sql.\*;

import java.util.Locale;

import java.util.Vector;

// Data Access Object for Suppliers

public class SupplierDAO {

Connection conn = null;

Statement statement = null;

PreparedStatement prepStatement = null;

ResultSet resultSet = null;

public SupplierDAO() {

try {

conn = new ConnectionFactory().getConn();

statement = conn.createStatement();

} catch (SQLException e) {

e.printStackTrace();

}

}

// Methods to add new supplier

public void addSupplierDAO(SupplierDTO supplierDTO) {

try {

String query = "SELECT \* FROM suppliers WHERE fullname='"

+supplierDTO.getFullName()

+ "' AND location='"

+supplierDTO.getLocation()

+ "' AND mobile='"

+supplierDTO.getPhone()

+ "'";

resultSet = statement.executeQuery(query);

if (resultSet.next())

JOptionPane.showMessageDialog(null, "This supplier already exists.");

else

addFunction(supplierDTO);

} catch (SQLException e) {

e.printStackTrace();

}

}

public void addFunction(SupplierDTO supplierDTO) {

try {

String query = "INSERT INTO suppliers VALUES(null,?,?,?,?)";

prepStatement = conn.prepareStatement(query);

prepStatement.setString(1, supplierDTO.getSuppCode());

prepStatement.setString(2, supplierDTO.getFullName());

prepStatement.setString(3, supplierDTO.getLocation());

prepStatement.setString(4, supplierDTO.getPhone());

prepStatement.executeUpdate();

JOptionPane.showMessageDialog(null, "New supplier has been added successfully.");

} catch (SQLException e) {

e.printStackTrace();

}

}

// Method to edit existing suppleir details

public void editSupplierDAO(SupplierDTO supplierDTO) {

try {

String query = "UPDATE suppliers SET fullname=?,location=?,mobile=? WHERE suppliercode=?";

prepStatement = conn.prepareStatement(query);

prepStatement.setString(1, supplierDTO.getFullName());

prepStatement.setString(2, supplierDTO.getLocation());

prepStatement.setString(3, supplierDTO.getPhone());

prepStatement.setString(4, supplierDTO.getSuppCode());

prepStatement.executeUpdate();

JOptionPane.showMessageDialog(null, "Supplier details have been updated.");

} catch (SQLException e) {

e.printStackTrace();

}

}

// Method to delete existing supplier

public void deleteSupplierDAO(String suppCode) {

try {

String query = "DELETE FROM suppliers WHERE suppliercode='" +suppCode+ "'";

statement.executeUpdate(query);

JOptionPane.showMessageDialog(null, "Supplier has been removed.");

} catch (SQLException e) {

e.printStackTrace();

}

}

// Supplier data set retrieval method

public ResultSet getQueryResult() {

try {

String query = "SELECT suppliercode, fullname, location, mobile FROM suppliers";

resultSet = statement.executeQuery(query);

} catch (Exception e) {

e.printStackTrace();

}

return resultSet;

}

// Search method

public ResultSet getSearchResult(String searchText) {

try {

String query = "SELECT suppliercode, fullname, location, mobile FROM suppliers " +

"WHERE suppliercode LIKE '%"+searchText+"%' OR location LIKE '%"+searchText+"%' " +

"OR fullname LIKE '%"+searchText+"%' OR mobile LIKE '%"+searchText+"%'";

resultSet = statement.executeQuery(query);

} catch (SQLException e) {

e.printStackTrace();

}

return resultSet;

}

// Method to set/update supplier combo box

public DefaultComboBoxModel<String> setComboItems(ResultSet resultSet) throws SQLException {

Vector<String> suppNames = new Vector<>();

while (resultSet.next()){

suppNames.add(resultSet.getString("fullname"));

}

return new DefaultComboBoxModel<>(suppNames);

}

// Method to display retrieved data set in tabular form

public DefaultTableModel buildTableModel(ResultSet resultSet) throws SQLException {

ResultSetMetaData metaData = resultSet.getMetaData();

Vector<String> columnNames = new Vector<String>();

int colCount = metaData.getColumnCount();

for (int col=1; col <= colCount; col++){

columnNames.add(metaData.getColumnName(col).toUpperCase(Locale.ROOT));

}

Vector<Vector<Object>> data = new Vector<Vector<Object>>();

while (resultSet.next()) {

Vector<Object> vector = new Vector<Object>();

for (int col=1; col<=colCount; col++) {

vector.add(resultSet.getObject(col));

}

data.add(vector);

}

return new DefaultTableModel(data, columnNames);

}

}

**UserDTO.java:**

package com.inventory.DTO;

/\*\*

\*

\* @author asjad

\*/

// Data Transfer Object (DTO) class for Users

public class UserDTO {

private int ID;

private String fullName, location, phone, username, password, userType;

public String getInTime() {

return inTime;

}

public void setInTime(String inTime) {

this.inTime = inTime;

}

public String getOutTime() {

return outTime;

}

public void setOutTime(String outTime) {

this.outTime = outTime;

}

private String inTime, outTime;

public int getID() {

return ID;

}

public void setID(int ID) {

this.ID = ID;

}

public String getFullName() {

return fullName;

}

public void setFullName(String fullName) {

this.fullName = fullName;

}

public String getLocation() {

return location;

}

public void setLocation(String location) {

this.location = location;

}

public String getPhone() {

return phone;

}

public void setPhone(String phone) {

this.phone = phone;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public String getUserType() {

return userType;

}

public void setUserType(String userType) {

this.userType = userType;

}

}

**UI**

**UsersPage.java**

package com.inventory.UI;

import com.inventory.DAO.UserDAO;

import com.inventory.DTO.UserDTO;

import javax.swing.\*;

import java.sql.\*;

public class UsersPage extends javax.swing.JPanel {

/\*\*

\* Creates new form UsersPage

\*/

public UsersPage() {

initComponents();

loadDataSet();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents

private void initComponents() {

jLabel1 = new javax.swing.JLabel();

jSeparator1 = new javax.swing.JSeparator();

entryPanel = new javax.swing.JPanel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

jLabel5 = new javax.swing.JLabel();

jLabel6 = new javax.swing.JLabel();

nameText = new javax.swing.JTextField();

locationText = new javax.swing.JTextField();

phoneText = new javax.swing.JTextField();

usernameText = new javax.swing.JTextField();

passText = new javax.swing.JPasswordField();

userTypeCombo = new javax.swing.JComboBox<>();

addButton = new javax.swing.JButton();

deleteButton = new javax.swing.JButton();

clearButton = new javax.swing.JButton();

jScrollPane1 = new javax.swing.JScrollPane();

userTable = new javax.swing.JTable();

jLabel1.setFont(new java.awt.Font("Impact", 0, 24)); // NOI18N

jLabel1.setText("USERS");

entryPanel.setBorder(javax.swing.BorderFactory.createTitledBorder("Enter User Details"));

jLabel2.setText("Full Name:");

jLabel3.setText("Location:");

jLabel4.setText("Contact:");

jLabel5.setText("Username:");

jLabel6.setText("Password:");

userTypeCombo.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "ADMINISTRATOR", "EMPLOYEE" }));

addButton.setText("Add");

addButton.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND\_CURSOR));

addButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

addButtonActionPerformed(evt);

}

});

deleteButton.setText("Delete");

deleteButton.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND\_CURSOR));

deleteButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

deleteButtonActionPerformed(evt);

}

});

clearButton.setText("Clear");

clearButton.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND\_CURSOR));

clearButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

clearButtonActionPerformed(evt);

}

});

javax.swing.GroupLayout entryPanelLayout = new javax.swing.GroupLayout(entryPanel);

entryPanel.setLayout(entryPanelLayout);

entryPanelLayout.setHorizontalGroup(

entryPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(entryPanelLayout.createSequentialGroup()

.addContainerGap()

.addGroup(entryPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(userTypeCombo, 0, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(entryPanelLayout.createSequentialGroup()

.addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED\_SIZE, 68, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(nameText))

.addGroup(entryPanelLayout.createSequentialGroup()

.addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED\_SIZE, 68, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(locationText))

.addGroup(entryPanelLayout.createSequentialGroup()

.addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED\_SIZE, 68, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(phoneText))

.addGroup(entryPanelLayout.createSequentialGroup()

.addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED\_SIZE, 68, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(passText))

.addGroup(entryPanelLayout.createSequentialGroup()

.addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED\_SIZE, 68, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(usernameText)))

.addContainerGap())

.addGroup(entryPanelLayout.createSequentialGroup()

.addComponent(addButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 78, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(deleteButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 75, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(clearButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 75, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 0, Short.MAX\_VALUE))

);

entryPanelLayout.setVerticalGroup(

entryPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(entryPanelLayout.createSequentialGroup()

.addContainerGap()

.addGroup(entryPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(nameText, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(entryPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(locationText, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(entryPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(phoneText, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(entryPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(usernameText, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(entryPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(passText, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(userTypeCombo, javax.swing.GroupLayout.PREFERRED\_SIZE, 33, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addGroup(entryPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(addButton)

.addComponent(deleteButton)

.addComponent(clearButton))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

userTable.setBorder(javax.swing.BorderFactory.createEtchedBorder());

userTable.setForeground(new java.awt.Color(102, 102, 102));

userTable.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

{null, null, null, null},

{null, null, null, null},

{null, null, null, null},

{null, null, null, null}

},

new String [] {

"Title 1", "Title 2", "Title 3", "Title 4"

}

) {

boolean[] canEdit = new boolean [] {

false, false, false, false

};

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

userTable.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND\_CURSOR));

userTable.setName("Users"); // NOI18N

userTable.setShowGrid(true);

userTable.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

userTableMouseClicked(evt);

}

});

jScrollPane1.setViewportView(userTable);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(this);

this.setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jSeparator1)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT\_SIZE, 432, Short.MAX\_VALUE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(entryPanel, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 62, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 0, Short.MAX\_VALUE)))

.addContainerGap())

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 36, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED\_SIZE, 10, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(entryPanel, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 0, Short.MAX\_VALUE))

.addGap(95, 95, 95))

);

}// </editor-fold>//GEN-END:initComponents

private void clearButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_clearButtonActionPerformed

nameText.setText("");

locationText.setText("");

phoneText.setText("");

usernameText.setText("");

passText.setText("");

}//GEN-LAST:event\_clearButtonActionPerformed

String userType;

private void addButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_addButtonActionPerformed

UserDTO userDTO = new UserDTO();

if (nameText.getText().equals("") || locationText.getText().equals("") || phoneText.getText().equals(""))

JOptionPane.showMessageDialog(null, "Please fill all the required fields.");

else {

userType = (String) userTypeCombo.getSelectedItem();

userDTO.setFullName(nameText.getText());

userDTO.setLocation(locationText.getText());

userDTO.setPhone(phoneText.getText());

userDTO.setUsername(usernameText.getText());

userDTO.setPassword(passText.getText());

userDTO.setUserType(userType);

new UserDAO().addUserDAO(userDTO, userType);

loadDataSet();

}

}//GEN-LAST:event\_addButtonActionPerformed

private void deleteButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_deleteButtonActionPerformed

if (userTable.getSelectedRow()<0)

JOptionPane.showMessageDialog(null, "Please select an entry from the table");

else{

int opt = JOptionPane.showConfirmDialog(

null,

"Are you sure you want to delete this user?",

"Confirmation",

JOptionPane.YES\_NO\_OPTION);

if(opt==JOptionPane.YES\_OPTION) {

new UserDAO().deleteUserDAO(

String.valueOf(

userTable.getValueAt

(userTable.getSelectedRow(), 4)));

loadDataSet();

}

}

}//GEN-LAST:event\_deleteButtonActionPerformed

String username;

private void userTableMouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_userTableMouseClicked

int row = userTable.getSelectedRow();

int col = userTable.getColumnCount();

Object[] val = new Object[col];

for(int i=0; i<col; i++) {

val[i] = userTable.getValueAt(row, i);

}

nameText.setText(val[1].toString());

locationText.setText(val[2].toString());

phoneText.setText(val[3].toString());

usernameText.setText(val[4].toString());

userTypeCombo.setSelectedItem(val[6].toString());

}//GEN-LAST:event\_userTableMouseClicked

public void loadDataSet() {

try {

UserDAO userDAO = new UserDAO();

userTable.setModel(userDAO.buildTableModel(userDAO.getQueryResult()));

} catch (SQLException ex) {

ex.printStackTrace();

}

}

// Variables declaration - do not modify//GEN-BEGIN:variables

private javax.swing.JButton addButton;

private javax.swing.JButton clearButton;

private javax.swing.JButton deleteButton;

private javax.swing.JPanel entryPanel;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JSeparator jSeparator1;

private javax.swing.JTextField locationText;

private javax.swing.JTextField nameText;

private javax.swing.JPasswordField passText;

private javax.swing.JTextField phoneText;

private javax.swing.JTable userTable;

private javax.swing.JComboBox<String> userTypeCombo;

private javax.swing.JTextField usernameText;

// End of variables declaration//GEN-END:variables

}

**CurreStockPage.java**

<?xml version="1.0" encoding="UTF-8" ?>

<Form version="1.3" maxVersion="1.9" type="org.netbeans.modules.form.forminfo.JPanelFormInfo">

<AuxValues>

<AuxValue name="FormSettings\_autoResourcing" type="java.lang.Integer" value="0"/>

<AuxValue name="FormSettings\_autoSetComponentName" type="java.lang.Boolean" value="false"/>

<AuxValue name="FormSettings\_generateFQN" type="java.lang.Boolean" value="true"/>

<AuxValue name="FormSettings\_generateMnemonicsCode" type="java.lang.Boolean" value="false"/>

<AuxValue name="FormSettings\_i18nAutoMode" type="java.lang.Boolean" value="false"/>

<AuxValue name="FormSettings\_layoutCodeTarget" type="java.lang.Integer" value="1"/>

<AuxValue name="FormSettings\_listenerGenerationStyle" type="java.lang.Integer" value="0"/>

<AuxValue name="FormSettings\_variablesLocal" type="java.lang.Boolean" value="false"/>

<AuxValue name="FormSettings\_variablesModifier" type="java.lang.Integer" value="2"/>

</AuxValues>

<Layout>

<DimensionLayout dim="0">

<Group type="103" groupAlignment="0" attributes="0">

<Group type="102" alignment="1" attributes="0">

<EmptySpace max="-2" attributes="0"/>

<Group type="103" groupAlignment="1" attributes="0">

<Component id="jSeparator1" max="32767" attributes="0"/>

<Component id="jScrollPane1" pref="701" max="32767" attributes="0"/>

<Group type="102" alignment="0" attributes="0">

<Component id="jLabel1" min="-2" pref="165" max="-2" attributes="0"/>

<EmptySpace max="32767" attributes="0"/>

<Component id="refreshButton" min="-2" max="-2" attributes="0"/>

</Group>

</Group>

<EmptySpace max="-2" attributes="0"/>

</Group>

</Group>

</DimensionLayout>

<DimensionLayout dim="1">

<Group type="103" groupAlignment="0" attributes="0">

<Group type="102" alignment="0" attributes="0">

<EmptySpace max="-2" attributes="0"/>

<Group type="103" groupAlignment="1" attributes="0">

<Component id="refreshButton" min="-2" pref="31" max="-2" attributes="0"/>

<Component id="jLabel1" min="-2" pref="44" max="-2" attributes="0"/>

</Group>

<EmptySpace max="32767" attributes="0"/>

<Component id="jSeparator1" min="-2" pref="10" max="-2" attributes="0"/>

<EmptySpace max="-2" attributes="0"/>

<Component id="jScrollPane1" min="-2" pref="330" max="-2" attributes="0"/>

<EmptySpace pref="88" max="32767" attributes="0"/>

</Group>

</Group>

</DimensionLayout>

</Layout>

<SubComponents>

<Component class="javax.swing.JLabel" name="jLabel1">

<Properties>

<Property name="font" type="java.awt.Font" editor="org.netbeans.beaninfo.editors.FontEditor">

<Font name="Impact" size="24" style="0"/>

</Property>

<Property name="text" type="java.lang.String" value="CURRENT STOCK"/>

<Property name="toolTipText" type="java.lang.String" value=""/>

</Properties>

</Component>

<Component class="javax.swing.JSeparator" name="jSeparator1">

</Component>

<Container class="javax.swing.JScrollPane" name="jScrollPane1">

<Layout class="org.netbeans.modules.form.compat2.layouts.support.JScrollPaneSupportLayout"/>

<SubComponents>

<Component class="javax.swing.JTable" name="stockTable">

<Properties>

<Property name="model" type="javax.swing.table.TableModel" editor="org.netbeans.modules.form.editors2.TableModelEditor">

<Table columnCount="4" rowCount="4">

<Column editable="true" title="Title 1" type="java.lang.Object"/>

<Column editable="true" title="Title 2" type="java.lang.Object"/>

<Column editable="true" title="Title 3" type="java.lang.Object"/>

<Column editable="true" title="Title 4" type="java.lang.Object"/>

</Table>

</Property>

</Properties>

</Component>

</SubComponents>

</Container>

<Component class="javax.swing.JButton" name="refreshButton">

<Properties>

<Property name="font" type="java.awt.Font" editor="org.netbeans.beaninfo.editors.FontEditor">

<Font name="Segoe UI" size="12" style="1"/>

</Property>

<Property name="text" type="java.lang.String" value="REFRESH"/>

</Properties>

<Events>

<EventHandler event="actionPerformed" listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent" handler="refreshButtonActionPerformed"/>

</Events>

</Component>

</SubComponents>

</Form>

**CustomerPage.java:**

<?xml version="1.0" encoding="UTF-8" ?>

<Form version="1.3" maxVersion="1.9" type="org.netbeans.modules.form.forminfo.JPanelFormInfo">

<AuxValues>

<AuxValue name="FormSettings\_autoResourcing" type="java.lang.Integer" value="0"/>

<AuxValue name="FormSettings\_autoSetComponentName" type="java.lang.Boolean" value="false"/>

<AuxValue name="FormSettings\_generateFQN" type="java.lang.Boolean" value="true"/>

<AuxValue name="FormSettings\_generateMnemonicsCode" type="java.lang.Boolean" value="false"/>

<AuxValue name="FormSettings\_i18nAutoMode" type="java.lang.Boolean" value="false"/>

<AuxValue name="FormSettings\_layoutCodeTarget" type="java.lang.Integer" value="1"/>

<AuxValue name="FormSettings\_listenerGenerationStyle" type="java.lang.Integer" value="0"/>

<AuxValue name="FormSettings\_variablesLocal" type="java.lang.Boolean" value="false"/>

<AuxValue name="FormSettings\_variablesModifier" type="java.lang.Integer" value="2"/>

</AuxValues>

<Layout>

<DimensionLayout dim="0">

<Group type="103" groupAlignment="0" attributes="0">

<Group type="102" attributes="0">

<EmptySpace max="-2" attributes="0"/>

<Group type="103" groupAlignment="0" attributes="0">

<Group type="102" attributes="0">

<Component id="jLabel1" min="-2" pref="122" max="-2" attributes="0"/>

<EmptySpace max="32767" attributes="0"/>

<Component id="jLabel8" min="-2" max="-2" attributes="0"/>

<EmptySpace max="-2" attributes="0"/>

<Component id="searchText" min="-2" pref="181" max="-2" attributes="0"/>

</Group>

<Component id="jSeparator1" max="32767" attributes="0"/>

<Group type="102" alignment="1" attributes="0">

<Component id="jScrollPane1" pref="445" max="32767" attributes="0"/>

<EmptySpace max="-2" attributes="0"/>

<Component id="entryPanel" min="-2" max="-2" attributes="0"/>

</Group>

</Group>

<EmptySpace max="-2" attributes="0"/>

</Group>

</Group>

</DimensionLayout>

<DimensionLayout dim="1">

<Group type="103" groupAlignment="0" attributes="0">

<Group type="102" alignment="0" attributes="0">

<EmptySpace max="-2" attributes="0"/>

<Group type="103" groupAlignment="3" attributes="0">

<Component id="jLabel1" alignment="3" min="-2" pref="40" max="-2" attributes="0"/>

<Component id="searchText" alignment="3" min="-2" max="-2" attributes="0"/>

<Component id="jLabel8" alignment="3" min="-2" max="-2" attributes="0"/>

</Group>

<EmptySpace max="-2" attributes="0"/>

<Component id="jSeparator1" min="-2" pref="10" max="-2" attributes="0"/>

<EmptySpace max="-2" attributes="0"/>

<Group type="103" groupAlignment="0" max="-2" attributes="0">

<Component id="entryPanel" max="32767" attributes="0"/>

<Component id="jScrollPane1" pref="0" max="32767" attributes="0"/>

</Group>

<EmptySpace pref="67" max="32767" attributes="0"/>

</Group>

</Group>

</DimensionLayout>

</Layout>

<SubComponents>

<Component class="javax.swing.JLabel" name="jLabel1">

<Properties>

<Property name="font" type="java.awt.Font" editor="org.netbeans.beaninfo.editors.FontEditor">

<Font name="Impact" size="24" style="0"/>

</Property>

<Property name="text" type="java.lang.String" value="CUSTOMERS"/>

</Properties>

</Component>

<Component class="javax.swing.JSeparator" name="jSeparator1">

</Component>

<Container class="javax.swing.JPanel" name="entryPanel">

<Properties>

<Property name="border" type="javax.swing.border.Border" editor="org.netbeans.modules.form.editors2.BorderEditor">

<Border info="org.netbeans.modules.form.compat2.border.TitledBorderInfo">

<TitledBorder title="Enter Customer Details"/>

</Border>

</Property>

</Properties>

<Layout>

<DimensionLayout dim="0">

<Group type="103" groupAlignment="0" attributes="0">

<Group type="102" alignment="0" attributes="0">

<EmptySpace max="-2" attributes="0"/>

<Group type="103" groupAlignment="0" attributes="0">

<Component id="clearButton" max="32767" attributes="0"/>

<Group type="102" alignment="0" attributes="0">

<Group type="103" groupAlignment="0" max="-2" attributes="0">

<Component id="jLabel2" alignment="0" max="32767" attributes="0"/>

<Component id="jLabel4" max="32767" attributes="0"/>

<Component id="jLabel5" max="32767" attributes="0"/>

<Component id="jLabel6" max="32767" attributes="0"/>

<Component id="jLabel7" max="32767" attributes="0"/>

<Component id="jLabel3" alignment="0" max="32767" attributes="0"/>

</Group>

<EmptySpace max="-2" attributes="0"/>

<Group type="103" groupAlignment="0" attributes="0">

<Component id="phoneText" max="32767" attributes="0"/>

<Component id="creditText" max="32767" attributes="0"/>

<Component id="debitText" max="32767" attributes="0"/>

<Component id="locationText" alignment="0" max="32767" attributes="0"/>

<Component id="codeText" alignment="0" max="32767" attributes="0"/>

<Component id="nameText" alignment="0" max="32767" attributes="0"/>

</Group>

</Group>

<Group type="102" attributes="0">

<Component id="addButton" min="-2" max="-2" attributes="0"/>

<EmptySpace max="-2" attributes="0"/>

<Component id="editButton" min="-2" max="-2" attributes="0"/>

<EmptySpace max="-2" attributes="0"/>

<Component id="deleteButton" min="-2" max="-2" attributes="0"/>

<EmptySpace min="0" pref="0" max="32767" attributes="0"/>

</Group>

</Group>

<EmptySpace max="-2" attributes="0"/>

</Group>

</Group>

</DimensionLayout>

<DimensionLayout dim="1">

<Group type="103" groupAlignment="0" attributes="0">

<Group type="102" alignment="0" attributes="0">

<EmptySpace max="-2" attributes="0"/>

<Group type="103" groupAlignment="3" attributes="0">

<Component id="jLabel2" alignment="3" min="-2" pref="29" max="-2" attributes="0"/>

<Component id="codeText" alignment="3" min="-2" pref="29" max="-2" attributes="0"/>

</Group>

<EmptySpace max="-2" attributes="0"/>

<Group type="103" groupAlignment="3" attributes="0">

<Component id="jLabel3" alignment="3" min="-2" pref="29" max="-2" attributes="0"/>

<Component id="nameText" alignment="3" min="-2" pref="29" max="-2" attributes="0"/>

</Group>

<EmptySpace max="-2" attributes="0"/>

<Group type="103" groupAlignment="3" attributes="0">

<Component id="jLabel4" alignment="3" min="-2" pref="29" max="-2" attributes="0"/>

<Component id="locationText" alignment="3" min="-2" pref="29" max="-2" attributes="0"/>

</Group>

<EmptySpace max="-2" attributes="0"/>

<Group type="103" groupAlignment="3" attributes="0">

<Component id="jLabel5" alignment="3" min="-2" pref="29" max="-2" attributes="0"/>

<Component id="phoneText" alignment="3" min="-2" pref="29" max="-2" attributes="0"/>

</Group>

<EmptySpace max="-2" attributes="0"/>

<Group type="103" groupAlignment="3" attributes="0">

<Component id="jLabel6" alignment="3" min="-2" pref="29" max="-2" attributes="0"/>

<Component id="debitText" alignment="3" min="-2" pref="29" max="-2" attributes="0"/>

</Group>

<EmptySpace max="-2" attributes="0"/>

<Group type="103" groupAlignment="3" attributes="0">

<Component id="jLabel7" alignment="3" min="-2" pref="29" max="-2" attributes="0"/>

<Component id="creditText" alignment="3" min="-2" pref="29" max="-2" attributes="0"/>

</Group>

<EmptySpace type="unrelated" max="-2" attributes="0"/>

<Group type="103" groupAlignment="3" attributes="0">

<Component id="addButton" alignment="3" min="-2" max="-2" attributes="0"/>

<Component id="editButton" alignment="3" min="-2" max="-2" attributes="0"/>

<Component id="deleteButton" alignment="3" min="-2" max="-2" attributes="0"/>

</Group>

<EmptySpace max="-2" attributes="0"/>

<Component id="clearButton" min="-2" pref="30" max="-2" attributes="0"/>

<EmptySpace pref="18" max="32767" attributes="0"/>

</Group>

</Group>

</DimensionLayout>

</Layout>

<SubComponents>

<Component class="javax.swing.JLabel" name="jLabel2">

<Properties>

<Property name="text" type="java.lang.String" value="Customer Code:"/>

</Properties>

</Component>

<Component class="javax.swing.JLabel" name="jLabel3">

<Properties>

<Property name="text" type="java.lang.String" value="Full Name:"/>

</Properties>

</Component>

<Component class="javax.swing.JLabel" name="jLabel4">

<Properties>

<Property name="text" type="java.lang.String" value="Location:"/>

</Properties>

</Component>

<Component class="javax.swing.JLabel" name="jLabel5">

<Properties>

<Property name="text" type="java.lang.String" value="Contact:"/>

</Properties>

</Component>

<Component class="javax.swing.JLabel" name="jLabel6">

<Properties>

<Property name="text" type="java.lang.String" value="Debit Amount:"/>

</Properties>

</Component>

<Component class="javax.swing.JLabel" name="jLabel7">

<Properties>

<Property name="text" type="java.lang.String" value="Credit Amount:"/>

</Properties>

</Component>

<Component class="javax.swing.JTextField" name="phoneText">

</Component>

<Component class="javax.swing.JTextField" name="locationText">

</Component>

<Component class="javax.swing.JTextField" name="codeText">

</Component>

<Component class="javax.swing.JTextField" name="nameText">

</Component>

<Component class="javax.swing.JTextField" name="creditText">

</Component>

<Component class="javax.swing.JTextField" name="debitText">

</Component>

<Component class="javax.swing.JButton" name="addButton">

<Properties>

<Property name="text" type="java.lang.String" value="Add"/>

<Property name="cursor" type="java.awt.Cursor" editor="org.netbeans.modules.form.editors2.CursorEditor">

<Color id="Hand Cursor"/>

</Property>

</Properties>

<Events>

<EventHandler event="actionPerformed" listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent" handler="addButtonActionPerformed"/>

</Events>

</Component>

<Component class="javax.swing.JButton" name="editButton">

<Properties>

<Property name="text" type="java.lang.String" value="Edit"/>

<Property name="cursor" type="java.awt.Cursor" editor="org.netbeans.modules.form.editors2.CursorEditor">

<Color id="Hand Cursor"/>

</Property>

</Properties>

<Events>

<EventHandler event="actionPerformed" listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent" handler="editButtonActionPerformed"/>

</Events>

</Component>

<Component class="javax.swing.JButton" name="deleteButton">

<Properties>

<Property name="text" type="java.lang.String" value="Delete"/>

<Property name="cursor" type="java.awt.Cursor" editor="org.netbeans.modules.form.editors2.CursorEditor">

<Color id="Hand Cursor"/>

</Property>

</Properties>

<Events>

<EventHandler event="actionPerformed" listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent" handler="deleteButtonActionPerformed"/>

</Events>

</Component>

<Component class="javax.swing.JButton" name="clearButton">

<Properties>

<Property name="font" type="java.awt.Font" editor="org.netbeans.beaninfo.editors.FontEditor">

<Font name="Segoe UI" size="14" style="0"/>

</Property>

<Property name="text" type="java.lang.String" value="CLEAR"/>

<Property name="cursor" type="java.awt.Cursor" editor="org.netbeans.modules.form.editors2.CursorEditor">

<Color id="Hand Cursor"/>

</Property>

</Properties>

<Events>

<EventHandler event="actionPerformed" listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent" handler="clearButtonActionPerformed"/>

</Events>

</Component>

</SubComponents>

</Container>

<Container class="javax.swing.JScrollPane" name="jScrollPane1">

<Layout class="org.netbeans.modules.form.compat2.layouts.support.JScrollPaneSupportLayout"/>

<SubComponents>

<Component class="javax.swing.JTable" name="custTable">

<Properties>

<Property name="model" type="javax.swing.table.TableModel" editor="org.netbeans.modules.form.editors2.TableModelEditor">

<Table columnCount="4" rowCount="4">

<Column editable="true" title="Title 1" type="java.lang.Object"/>

<Column editable="true" title="Title 2" type="java.lang.Object"/>

<Column editable="true" title="Title 3" type="java.lang.Object"/>

<Column editable="true" title="Title 4" type="java.lang.Object"/>

</Table>

</Property>

</Properties>

<Events>

<EventHandler event="mouseClicked" listener="java.awt.event.MouseListener" parameters="java.awt.event.MouseEvent" handler="custTableMouseClicked"/>

</Events>

</Component>

</SubComponents>

</Container>

<Component class="javax.swing.JTextField" name="searchText">

<Events>

<EventHandler event="keyReleased" listener="java.awt.event.KeyListener" parameters="java.awt.event.KeyEvent" handler="searchTextKeyReleased"/>

</Events>

</Component>

<Component class="javax.swing.JLabel" name="jLabel8">

<Properties>

<Property name="text" type="java.lang.String" value="Search:"/>

</Properties>

</Component>

</SubComponents>

</Form>

**Dashboard.java:**

<?xml version="1.0" encoding="UTF-8" ?>

<Form version="1.3" maxVersion="1.9" type="org.netbeans.modules.form.forminfo.JFrameFormInfo">

<NonVisualComponents>

<Menu class="javax.swing.JMenuBar" name="jMenuBar1">

<SubComponents>

<Menu class="javax.swing.JMenu" name="jMenu1">

<Properties>

<Property name="text" type="java.lang.String" value="File"/>

</Properties>

</Menu>

<Menu class="javax.swing.JMenu" name="jMenu2">

<Properties>

<Property name="text" type="java.lang.String" value="Edit"/>

</Properties>

</Menu>

</SubComponents>

</Menu>

</NonVisualComponents>

<Properties>

<Property name="defaultCloseOperation" type="int" value="3"/>

<Property name="title" type="java.lang.String" value="Inventory Manager"/>

<Property name="bounds" type="java.awt.Rectangle" editor="org.netbeans.beaninfo.editors.RectangleEditor">

<Rectangle value="[400, 100, 0, 0]"/>

</Property>

</Properties>

<SyntheticProperties>

<SyntheticProperty name="menuBar" type="java.lang.String" value="jMenuBar1"/>

<SyntheticProperty name="formSizePolicy" type="int" value="1"/>

<SyntheticProperty name="generateCenter" type="boolean" value="false"/>

</SyntheticProperties>

<AuxValues>

<AuxValue name="FormSettings\_autoResourcing" type="java.lang.Integer" value="0"/>

<AuxValue name="FormSettings\_autoSetComponentName" type="java.lang.Boolean" value="false"/>

<AuxValue name="FormSettings\_generateFQN" type="java.lang.Boolean" value="true"/>

<AuxValue name="FormSettings\_generateMnemonicsCode" type="java.lang.Boolean" value="false"/>

<AuxValue name="FormSettings\_i18nAutoMode" type="java.lang.Boolean" value="false"/>

<AuxValue name="FormSettings\_layoutCodeTarget" type="java.lang.Integer" value="1"/>

<AuxValue name="FormSettings\_listenerGenerationStyle" type="java.lang.Integer" value="0"/>

<AuxValue name="FormSettings\_variablesLocal" type="java.lang.Boolean" value="false"/>

<AuxValue name="FormSettings\_variablesModifier" type="java.lang.Integer" value="2"/>

</AuxValues>

<Layout>

<DimensionLayout dim="0">

<Group type="103" groupAlignment="0" attributes="0">

<Component id="mainPanel" alignment="0" max="32767" attributes="0"/>

</Group>

</DimensionLayout>

<DimensionLayout dim="1">

<Group type="103" groupAlignment="0" attributes="0">

<Component id="mainPanel" alignment="0" max="32767" attributes="0"/>

</Group>

</DimensionLayout>

</Layout>

<SubComponents>

<Container class="javax.swing.JPanel" name="mainPanel">

<Layout>

<DimensionLayout dim="0">

<Group type="103" groupAlignment="0" attributes="0">

<Group type="102" attributes="0">

<Group type="103" groupAlignment="0" max="-2" attributes="0">

<Component id="navPanel" max="32767" attributes="0"/>

<Component id="menuPanel" pref="125" max="32767" attributes="0"/>

</Group>

<EmptySpace max="-2" attributes="0"/>

<Group type="103" groupAlignment="0" attributes="0">

<Component id="displayPanel" max="32767" attributes="0"/>

<Component id="userPanel" alignment="0" max="32767" attributes="0"/>

</Group>

</Group>

</Group>

</DimensionLayout>

<DimensionLayout dim="1">

<Group type="103" groupAlignment="0" attributes="0">

<Group type="102" attributes="0">

<Group type="103" groupAlignment="0" attributes="0">

<Component id="userPanel" alignment="0" max="-2" attributes="0"/>

<Component id="menuPanel" alignment="0" min="-2" pref="50" max="-2" attributes="0"/>

</Group>

<EmptySpace max="-2" attributes="0"/>

<Group type="103" groupAlignment="0" max="-2" attributes="0">

<Component id="displayPanel" max="32767" attributes="0"/>

<Component id="navPanel" max="32767" attributes="0"/>

</Group>

<EmptySpace max="32767" attributes="0"/>

</Group>

</Group>

</DimensionLayout>

</Layout>

<SubComponents>

<Container class="javax.swing.JPanel" name="menuPanel">

<Properties>

<Property name="preferredSize" type="java.awt.Dimension" editor="org.netbeans.beaninfo.editors.DimensionEditor">

<Dimension value="[120, 26]"/>

</Property>

</Properties>

<Layout>

<DimensionLayout dim="0">

<Group type="103" groupAlignment="0" attributes="0">

<Component id="menuButton" alignment="0" max="32767" attributes="0"/>

</Group>

</DimensionLayout>

<DimensionLayout dim="1">

<Group type="103" groupAlignment="0" attributes="0">

<Group type="102" alignment="0" attributes="0">

<Component id="menuButton" min="-2" pref="52" max="-2" attributes="0"/>

<EmptySpace min="-2" max="-2" attributes="0"/>

</Group>

</Group>

</DimensionLayout>

</Layout>

<SubComponents>

<Component class="javax.swing.JButton" name="menuButton">

<Properties>

<Property name="font" type="java.awt.Font" editor="org.netbeans.beaninfo.editors.FontEditor">

<Font name="Segoe UI" size="14" style="1"/>

</Property>

<Property name="icon" type="javax.swing.Icon" editor="org.netbeans.modules.form.editors2.IconEditor">

<Image iconType="3" name="/com/inventory/UI/Icons/menu\_icon\_150667.png"/>

</Property>

<Property name="text" type="java.lang.String" value="MENU"/>

<Property name="cursor" type="java.awt.Cursor" editor="org.netbeans.modules.form.editors2.CursorEditor">

<Color id="Hand Cursor"/>

</Property>

</Properties>

<Events>

<EventHandler event="actionPerformed" listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent" handler="menuButtonActionPerformed"/>

</Events>

</Component>

</SubComponents>

</Container>

<Container class="javax.swing.JPanel" name="navPanel">

<Properties>

<Property name="border" type="javax.swing.border.Border" editor="org.netbeans.modules.form.editors2.BorderEditor">

<Border info="org.netbeans.modules.form.compat2.border.SoftBevelBorderInfo">

<BevelBorder/>

</Border>

</Property>

</Properties>

<Layout>

<DimensionLayout dim="0">

<Group type="103" groupAlignment="0" attributes="0">

<Group type="102" attributes="0">

<EmptySpace max="-2" attributes="0"/>

<Group type="103" groupAlignment="0" attributes="0">

<Component id="homeButton" alignment="0" max="32767" attributes="0"/>

<Component id="prodButton" max="32767" attributes="0"/>

<Component id="stockButton" pref="107" max="32767" attributes="0"/>

<Component id="custButton" max="32767" attributes="0"/>

<Component id="suppButton" max="32767" attributes="0"/>

<Component id="salesButton" max="32767" attributes="0"/>

<Component id="usersButton" max="32767" attributes="0"/>

<Component id="purchaseButton" max="32767" attributes="0"/>

<Component id="logsButton" max="32767" attributes="0"/>

</Group>

<EmptySpace max="-2" attributes="0"/>

</Group>

</Group>

</DimensionLayout>

<DimensionLayout dim="1">

<Group type="103" groupAlignment="0" attributes="0">

<Group type="102" alignment="0" attributes="0">

<EmptySpace max="-2" attributes="0"/>

<Component id="homeButton" min="-2" pref="35" max="-2" attributes="0"/>

<EmptySpace type="separate" max="-2" attributes="0"/>

<Component id="prodButton" min="-2" pref="35" max="-2" attributes="0"/>

<EmptySpace type="separate" max="-2" attributes="0"/>

<Component id="stockButton" min="-2" pref="35" max="-2" attributes="0"/>

<EmptySpace type="separate" max="-2" attributes="0"/>

<Component id="custButton" min="-2" pref="35" max="-2" attributes="0"/>

<EmptySpace type="separate" max="-2" attributes="0"/>

<Component id="suppButton" min="-2" pref="35" max="-2" attributes="0"/>

<EmptySpace type="separate" max="-2" attributes="0"/>

<Component id="salesButton" min="-2" pref="35" max="-2" attributes="0"/>

<EmptySpace type="separate" max="-2" attributes="0"/>

<Component id="purchaseButton" min="-2" pref="35" max="-2" attributes="0"/>

<EmptySpace type="separate" max="-2" attributes="0"/>

<Component id="usersButton" min="-2" pref="35" max="-2" attributes="0"/>

<EmptySpace type="separate" max="-2" attributes="0"/>

<Component id="logsButton" min="-2" pref="35" max="-2" attributes="0"/>

<EmptySpace pref="24" max="32767" attributes="0"/>

</Group>

</Group>

</DimensionLayout>

</Layout>

<SubComponents>

<Component class="javax.swing.JButton" name="homeButton">

<Properties>

<Property name="icon" type="javax.swing.Icon" editor="org.netbeans.modules.form.editors2.IconEditor">

<Image iconType="3" name="/com/inventory/UI/Icons/homepages\_home\_house\_icon\_150665.png"/>

</Property>

<Property name="cursor" type="java.awt.Cursor" editor="org.netbeans.modules.form.editors2.CursorEditor">

<Color id="Hand Cursor"/>

</Property>

</Properties>

<Events>

<EventHandler event="actionPerformed" listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent" handler="homeButtonActionPerformed"/>

</Events>

</Component>

<Component class="javax.swing.JButton" name="prodButton">

<Properties>

<Property name="text" type="java.lang.String" value="Products"/>

<Property name="cursor" type="java.awt.Cursor" editor="org.netbeans.modules.form.editors2.CursorEditor">

<Color id="Hand Cursor"/>

</Property>

</Properties>

<Events>

<EventHandler event="actionPerformed" listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent" handler="prodButtonActionPerformed"/>

</Events>

</Component>

<Component class="javax.swing.JButton" name="stockButton">

<Properties>

<Property name="text" type="java.lang.String" value="Current Stock"/>

<Property name="cursor" type="java.awt.Cursor" editor="org.netbeans.modules.form.editors2.CursorEditor">

<Color id="Hand Cursor"/>

</Property>

</Properties>

<Events>

<EventHandler event="actionPerformed" listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent" handler="stockButtonActionPerformed"/>

</Events>

</Component>

<Component class="javax.swing.JButton" name="custButton">

<Properties>

<Property name="text" type="java.lang.String" value="Customers"/>

<Property name="cursor" type="java.awt.Cursor" editor="org.netbeans.modules.form.editors2.CursorEditor">

<Color id="Hand Cursor"/>

</Property>

</Properties>

<Events>

<EventHandler event="actionPerformed" listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent" handler="custButtonActionPerformed"/>

</Events>

</Component>

<Component class="javax.swing.JButton" name="suppButton">

<Properties>

<Property name="text" type="java.lang.String" value="Suppliers"/>

<Property name="cursor" type="java.awt.Cursor" editor="org.netbeans.modules.form.editors2.CursorEditor">

<Color id="Hand Cursor"/>

</Property>

</Properties>

<Events>

<EventHandler event="actionPerformed" listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent" handler="suppButtonActionPerformed"/>

</Events>

</Component>

<Component class="javax.swing.JButton" name="salesButton">

<Properties>

<Property name="text" type="java.lang.String" value="Sales"/>

<Property name="cursor" type="java.awt.Cursor" editor="org.netbeans.modules.form.editors2.CursorEditor">

<Color id="Hand Cursor"/>

</Property>

</Properties>

<Events>

<EventHandler event="actionPerformed" listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent" handler="salesButtonActionPerformed"/>

</Events>

</Component>

<Component class="javax.swing.JButton" name="usersButton">

<Properties>

<Property name="text" type="java.lang.String" value="Users"/>

<Property name="cursor" type="java.awt.Cursor" editor="org.netbeans.modules.form.editors2.CursorEditor">

<Color id="Hand Cursor"/>

</Property>

</Properties>

<Events>

<EventHandler event="actionPerformed" listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent" handler="usersButtonActionPerformed"/>

</Events>

</Component>

<Component class="javax.swing.JButton" name="purchaseButton">

<Properties>

<Property name="text" type="java.lang.String" value="Purchase"/>

<Property name="cursor" type="java.awt.Cursor" editor="org.netbeans.modules.form.editors2.CursorEditor">

<Color id="Hand Cursor"/>

</Property>

</Properties>

<Events>

<EventHandler event="actionPerformed" listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent" handler="purchaseButtonActionPerformed"/>

</Events>

</Component>

<Component class="javax.swing.JButton" name="logsButton">

<Properties>

<Property name="text" type="java.lang.String" value="User Logs"/>

<Property name="cursor" type="java.awt.Cursor" editor="org.netbeans.modules.form.editors2.CursorEditor">

<Color id="Hand Cursor"/>

</Property>

</Properties>

<Events>

<EventHandler event="actionPerformed" listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent" handler="logsButtonActionPerformed"/>

</Events>

</Component>

</SubComponents>

</Container>

<Container class="javax.swing.JPanel" name="displayPanel">

<Layout class="org.netbeans.modules.form.compat2.layouts.DesignCardLayout"/>

</Container>

<Container class="javax.swing.JPanel" name="userPanel">

<Layout>

<DimensionLayout dim="0">

<Group type="103" groupAlignment="0" attributes="0">

<Group type="102" alignment="0" attributes="0">

<EmptySpace pref="401" max="32767" attributes="0"/>

<Component id="nameLabel" min="-2" pref="262" max="-2" attributes="0"/>

<EmptySpace max="-2" attributes="0"/>

<Component id="logoutButton" min="-2" max="-2" attributes="0"/>

</Group>

</Group>

</DimensionLayout>

<DimensionLayout dim="1">

<Group type="103" groupAlignment="0" attributes="0">

<Group type="102" alignment="0" attributes="0">

<Group type="103" groupAlignment="1" max="-2" attributes="0">

<Group type="102" alignment="0" attributes="0">

<EmptySpace max="-2" attributes="0"/>

<Component id="nameLabel" max="32767" attributes="0"/>

</Group>

<Component id="logoutButton" min="-2" pref="36" max="-2" attributes="0"/>

</Group>

<EmptySpace min="0" pref="0" max="32767" attributes="0"/>

</Group>

</Group>

</DimensionLayout>

</Layout>

<SubComponents>

<Component class="javax.swing.JLabel" name="nameLabel">

<Properties>

<Property name="font" type="java.awt.Font" editor="org.netbeans.beaninfo.editors.FontEditor">

<Font name="Segoe UI Black" size="12" style="0"/>

</Property>

<Property name="icon" type="javax.swing.Icon" editor="org.netbeans.modules.form.editors2.IconEditor">

<Image iconType="3" name="/com/inventory/UI/Icons/user\_icon\_150670.png"/>

</Property>

<Property name="text" type="java.lang.String" value="User: "/>

<Property name="verticalAlignment" type="int" value="1"/>

</Properties>

</Component>

<Component class="javax.swing.JButton" name="logoutButton">

<Properties>

<Property name="icon" type="javax.swing.Icon" editor="org.netbeans.modules.form.editors2.IconEditor">

<Image iconType="3" name="/com/inventory/UI/Icons/log-out\_icon-icons.com\_50106.png"/>

</Property>

<Property name="text" type="java.lang.String" value="Sign out"/>

<Property name="cursor" type="java.awt.Cursor" editor="org.netbeans.modules.form.editors2.CursorEditor">

<Color id="Hand Cursor"/>

</Property>

</Properties>

<Events>

<EventHandler event="actionPerformed" listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent" handler="logoutButtonActionPerformed"/>

</Events>

</Component>

</SubComponents>

</Container>

</SubComponents>

</Container>

</SubComponents>

</Form>

**HomePage.java:**

<?xml version="1.0" encoding="UTF-8" ?>

<Form version="1.3" maxVersion="1.9" type="org.netbeans.modules.form.forminfo.JPanelFormInfo">

<AuxValues>

<AuxValue name="FormSettings\_autoResourcing" type="java.lang.Integer" value="0"/>

<AuxValue name="FormSettings\_autoSetComponentName" type="java.lang.Boolean" value="false"/>

<AuxValue name="FormSettings\_generateFQN" type="java.lang.Boolean" value="true"/>

<AuxValue name="FormSettings\_generateMnemonicsCode" type="java.lang.Boolean" value="false"/>

<AuxValue name="FormSettings\_i18nAutoMode" type="java.lang.Boolean" value="false"/>

<AuxValue name="FormSettings\_layoutCodeTarget" type="java.lang.Integer" value="1"/>

<AuxValue name="FormSettings\_listenerGenerationStyle" type="java.lang.Integer" value="0"/>

<AuxValue name="FormSettings\_variablesLocal" type="java.lang.Boolean" value="false"/>

<AuxValue name="FormSettings\_variablesModifier" type="java.lang.Integer" value="2"/>

</AuxValues>

<Layout>

<DimensionLayout dim="0">

<Group type="103" groupAlignment="0" attributes="0">

<Group type="102" attributes="0">

<Group type="103" groupAlignment="0" attributes="0">

<Group type="102" attributes="0">

<EmptySpace max="-2" attributes="0"/>

<Component id="welcomeLabel" min="-2" pref="355" max="-2" attributes="0"/>

</Group>

<Group type="102" alignment="0" attributes="0">

<EmptySpace min="-2" pref="54" max="-2" attributes="0"/>

<Component id="jLabel1" pref="355" max="32767" attributes="0"/>

</Group>

</Group>

<EmptySpace pref="84" max="32767" attributes="0"/>

</Group>

</Group>

</DimensionLayout>

<DimensionLayout dim="1">

<Group type="103" groupAlignment="0" attributes="0">

<Group type="102" alignment="0" attributes="0">

<EmptySpace min="-2" pref="12" max="-2" attributes="0"/>

<Component id="welcomeLabel" min="-2" pref="48" max="-2" attributes="0"/>

<EmptySpace max="-2" attributes="0"/>

<Component id="jLabel1" min="-2" pref="133" max="-2" attributes="0"/>

<EmptySpace pref="174" max="32767" attributes="0"/>

</Group>

</Group>

</DimensionLayout>

</Layout>

<SubComponents>

<Component class="javax.swing.JLabel" name="welcomeLabel">

<Properties>

<Property name="font" type="java.awt.Font" editor="org.netbeans.beaninfo.editors.FontEditor">

<Font name="Impact" size="36" style="0"/>

</Property>

<Property name="text" type="java.lang.String" value="Welcome"/>

</Properties>

</Component>

<Component class="javax.swing.JLabel" name="jLabel1">

<Properties>

<Property name="font" type="java.awt.Font" editor="org.netbeans.beaninfo.editors.FontEditor">

<Font name="Impact" size="18" style="0"/>

</Property>

<Property name="horizontalAlignment" type="int" value="0"/>

<Property name="text" type="java.lang.String" value="&lt;html&gt;Manage your inventory, transactions and personnel, all in one place.&lt;br&gt;&lt;br&gt;Click on the Menu button to start.&lt;html&gt;"/>

</Properties>

</Component>

</SubComponents>

</Form>

**LoginPage.java**

<?xml version="1.0" encoding="UTF-8" ?>

<Form version="1.3" maxVersion="1.9" type="org.netbeans.modules.form.forminfo.JFrameFormInfo">

<Properties>

<Property name="defaultCloseOperation" type="int" value="2"/>

<Property name="title" type="java.lang.String" value="Login"/>

<Property name="background" type="java.awt.Color" editor="org.netbeans.beaninfo.editors.ColorEditor">

<Color blue="66" green="66" red="66" type="rgb"/>

</Property>

<Property name="bounds" type="java.awt.Rectangle" editor="org.netbeans.beaninfo.editors.RectangleEditor">

<Rectangle value="[500, 100, 0, 0]"/>

</Property>

<Property name="name" type="java.lang.String" value="loginFrame" noResource="true"/>

</Properties>

<SyntheticProperties>

<SyntheticProperty name="formSizePolicy" type="int" value="1"/>

<SyntheticProperty name="generateCenter" type="boolean" value="false"/>

</SyntheticProperties>

<AuxValues>

<AuxValue name="FormSettings\_autoResourcing" type="java.lang.Integer" value="0"/>

<AuxValue name="FormSettings\_autoSetComponentName" type="java.lang.Boolean" value="false"/>

<AuxValue name="FormSettings\_generateFQN" type="java.lang.Boolean" value="true"/>

<AuxValue name="FormSettings\_generateMnemonicsCode" type="java.lang.Boolean" value="false"/>

<AuxValue name="FormSettings\_i18nAutoMode" type="java.lang.Boolean" value="false"/>

<AuxValue name="FormSettings\_layoutCodeTarget" type="java.lang.Integer" value="1"/>

<AuxValue name="FormSettings\_listenerGenerationStyle" type="java.lang.Integer" value="0"/>

<AuxValue name="FormSettings\_variablesLocal" type="java.lang.Boolean" value="false"/>

<AuxValue name="FormSettings\_variablesModifier" type="java.lang.Integer" value="2"/>

</AuxValues>

<Layout>

<DimensionLayout dim="0">

<Group type="103" groupAlignment="0" attributes="0">

<Group type="102" alignment="0" attributes="0">

<EmptySpace min="-2" pref="47" max="-2" attributes="0"/>

<Group type="103" groupAlignment="0" max="-2" attributes="0">

<Group type="102" alignment="0" attributes="0">

<EmptySpace min="-2" pref="42" max="-2" attributes="0"/>

<Component id="loginButton" min="-2" pref="100" max="-2" attributes="0"/>

<EmptySpace type="separate" max="-2" attributes="0"/>

<Component id="clearButton" min="-2" pref="100" max="-2" attributes="0"/>

</Group>

<Group type="102" attributes="0">

<Component id="jLabel1" min="-2" pref="74" max="-2" attributes="0"/>

<EmptySpace max="-2" attributes="0"/>

<Component id="userText" max="32767" attributes="0"/>

</Group>

<Group type="102" alignment="0" attributes="0">

<Component id="jLabel2" min="-2" pref="74" max="-2" attributes="0"/>

<EmptySpace max="-2" attributes="0"/>

<Component id="passText" max="32767" attributes="0"/>

</Group>

<Component id="jComboBox1" max="32767" attributes="0"/>

<Component id="jLabel3" pref="284" max="32767" attributes="0"/>

</Group>

<EmptySpace pref="52" max="32767" attributes="0"/>

</Group>

</Group>

</DimensionLayout>

<DimensionLayout dim="1">

<Group type="103" groupAlignment="0" attributes="0">

<Group type="102" alignment="0" attributes="0">

<EmptySpace min="-2" pref="44" max="-2" attributes="0"/>

<Component id="jLabel3" min="-2" pref="44" max="-2" attributes="0"/>

<EmptySpace min="-2" pref="57" max="-2" attributes="0"/>

<Group type="103" groupAlignment="0" max="-2" attributes="0">

<Component id="userText" pref="31" max="32767" attributes="0"/>

<Component id="jLabel1" max="32767" attributes="0"/>

</Group>

<EmptySpace type="unrelated" max="-2" attributes="0"/>

<Group type="103" groupAlignment="1" attributes="0">

<Component id="passText" min="-2" pref="32" max="-2" attributes="0"/>

<Group type="102" attributes="0">

<Component id="jLabel2" min="-2" pref="33" max="-2" attributes="0"/>

<EmptySpace min="1" pref="1" max="-2" attributes="0"/>

</Group>

</Group>

<EmptySpace type="separate" max="-2" attributes="0"/>

<Component id="jComboBox1" pref="28" max="32767" attributes="0"/>

<EmptySpace min="-2" pref="46" max="-2" attributes="0"/>

<Group type="103" groupAlignment="3" attributes="0">

<Component id="loginButton" alignment="3" min="-2" pref="37" max="-2" attributes="0"/>

<Component id="clearButton" alignment="3" min="-2" pref="37" max="-2" attributes="0"/>

</Group>

<EmptySpace min="-2" pref="80" max="-2" attributes="0"/>

</Group>

</Group>

</DimensionLayout>

</Layout>

<SubComponents>

<Component class="javax.swing.JLabel" name="jLabel1">

<Properties>

<Property name="font" type="java.awt.Font" editor="org.netbeans.beaninfo.editors.FontEditor">

<Font name="Segoe UI" size="14" style="0"/>

</Property>

<Property name="text" type="java.lang.String" value="Username:"/>

</Properties>

</Component>

<Component class="javax.swing.JLabel" name="jLabel2">

<Properties>

<Property name="font" type="java.awt.Font" editor="org.netbeans.beaninfo.editors.FontEditor">

<Font name="Segoe UI" size="14" style="0"/>

</Property>

<Property name="text" type="java.lang.String" value="Password:"/>

</Properties>

</Component>

<Component class="javax.swing.JTextField" name="userText">

</Component>

<Component class="javax.swing.JPasswordField" name="passText">

</Component>

<Component class="javax.swing.JLabel" name="jLabel3">

<Properties>

<Property name="font" type="java.awt.Font" editor="org.netbeans.beaninfo.editors.FontEditor">

<Font name="Poor Richard" size="24" style="1"/>

</Property>

<Property name="horizontalAlignment" type="int" value="0"/>

<Property name="text" type="java.lang.String" value="STORE INVENTORY"/>

</Properties>

</Component>

<Component class="javax.swing.JComboBox" name="jComboBox1">

<Properties>

<Property name="model" type="javax.swing.ComboBoxModel" editor="org.netbeans.modules.form.editors2.ComboBoxModelEditor">

<StringArray count="2">

<StringItem index="0" value="ADMINISTRATOR"/>

<StringItem index="1" value="EMPLOYEE"/>

</StringArray>

</Property>

</Properties>

<AuxValues>

<AuxValue name="JavaCodeGenerator\_TypeParameters" type="java.lang.String" value="&lt;String&gt;"/>

</AuxValues>

</Component>

<Component class="javax.swing.JButton" name="loginButton">

<Properties>

<Property name="text" type="java.lang.String" value="LOGIN"/>

<Property name="cursor" type="java.awt.Cursor" editor="org.netbeans.modules.form.editors2.CursorEditor">

<Color id="Hand Cursor"/>

</Property>

</Properties>

<Events>

<EventHandler event="actionPerformed" listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent" handler="loginButtonActionPerformed"/>

</Events>

</Component>

<Component class="javax.swing.JButton" name="clearButton">

<Properties>

<Property name="text" type="java.lang.String" value="CLEAR"/>

<Property name="cursor" type="java.awt.Cursor" editor="org.netbeans.modules.form.editors2.CursorEditor">

<Color id="Hand Cursor"/>

</Property>

</Properties>

<Events>

<EventHandler event="actionPerformed" listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent" handler="clearButtonActionPerformed"/>

</Events>

</Component>

</SubComponents>

</Form>