RAJALAKSHMI
ENGINEERING COLLEGE
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

INVENTORY MANAGEMENT
A MINI PROJECT REPORT

Submitted by

ANIRUDH C **230701028**

ABISHEK S **230701006**

In partial fulfilment for the award of the degree of

BACHELOR OF

ENGINEERING IN

COMPUTER SCIENCE AND ENGINEEING

RAJALAKSHMI ENGINEERING COLLEGE (AUTONOMOUS)

THANDALAM

CHENNAI-602105

2024-2025

BONAFIDE CERTIFICATE

Certified that this project report "**INVENTORY MANAGEMENT SYSTEM**" is the Bonafide work of **"ANIRUDH C (230701028),"ABISHEK S (230701006)"** who carried out the project work under my supervision.

Submitted for the Practical Examination held on

SIGNATURE

Mrs. Deepa B,

Assistant Professor(SS)CSE,

RajalakshmiEngineeringCollege

(Thandalam, Chennai - 602 105

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

The IMS is a comprehensive solution designed to streamline electronic device inventory for retailers. The system was built using Java for the front-end interface, JDBC for database connectivity, and MySQL for storing product and user data. It enables the retailers to perform the basic operations such as adding, updating, and deleting products in the inventory. The primary function of the IMS is maintaining and preventing a stock overflow, which it can do by setting and enforcing predefined stock limits.

The system is also integrated with a login sign-in mechanism, which ensures that only the authorized user (retailer) is able to access and modify the inventory. New users can create a username and password to get signed on, while return users need to authenticate themselves with a valid password. This system is therefore able to design itself in a friendly way to enable fast upgrading and to track the accurate stock levels and, thereby improve the efficiency of operations and reduce human error.

With robust functionality, the IMS offers retailers a better way of managing product stock accurately, keeping the on-hand records true to prevent a situation like a stockout or overstocking.

Language Used – JAVA

IDE used – VS CODE

Database Used – MY SQL

INTRODUCTION

- The Inventory Management System (IMS) is a software solution designed for retailers to efficiently manage their electronic device inventory.

- It allows users (retailers) to add, update, and delete products, track stock levels, and ensure data integrity with stock limit enforcement.

- Built using Java for the front-end, JDBC for database connectivity, and MySQL as the back-end database, the system provides a secure and scalable solution.

- The system features a sign-in and login mechanism, enabling authentication for both new and existing users.

- Designed to minimize human errors in inventory management, improve operational efficiency, and prevent issues such as stockouts or overstocking.

Entity Relationship Diagram (ERD)

The ERD represents the structure of the system's database and the relationships between different entities.

- Entities:

  - User: Stores details of the retailer (e.g., username, password).

  - Product: Contains details about each product (e.g., product ID, name, category, price).

  - Inventory: Tracks the stock levels for each product (e.g., product ID, quantity available).

- Relationships:

  - User-Product: A retailer (User) can manage many products, but each product can only be managed by one retailer.

  - Product-Inventory: Each product is associated with a specific stock level in the inventory.

Table: `products`

Column Name	Data Type	Constraints
sku	VARCHAR(50)	PRIMARY KEY
name	VARCHAR(100)	NOT NULL
quantity	INT	NOT NULL
price	DECIMAL(10, 2)	NOT NULL
supplier	VARCHAR(100)	NOT NULL

Table: `user_inventory`

Column Name	Data Type	Constraints
user_inventory_id	INT	PRIMARY KEY, AUTO_INCREMENT
user_id	INT	FOREIGN KEY REFERENCES `users(user_id)`
product_id	INT	FOREIGN KEY REFERENCES `products(product_id)`
quantity	INT	
min_stock	INT	DEFAULT 5
max_stock	INT	DEFAULT 100
UNIQUE(user_id, product_id)	N/A	Ensures unique combination per user

Table: `users`

Column Name	Data Type	Constraints
user_id	INT	PRIMARY KEY, AUTO_INCREMENT
username	VARCHAR(255)	UNIQUE, NOT NULL
password	VARCHAR(255)	NOT NULL

-

Software Requirements

Software Requirements:

- Java Development Kit (JDK) 8 or higher for developing the front-end interface.
- MySQL Database to store product and user data.
- JDBC Driver for seamless communication between Java and MySQL.
- IDE: Eclipse or any other Java-compatible IDE for development.
- MySQL Workbench (optional) for managing and interacting with the database.

System Requirements

- A stable internet connection for downloading and installing the software packages.
- Configured MySQL server to interact with the application.

Functional Requirements

- User Authentication:
  - Sign In and Login system to authenticate retailers.
  - New users can create an account with a username and password.
  - Returning users must enter a valid password for access.

- Inventory Management:

  - Add Products: Retailers can add new products to the inventory.

  - Update Stock Levels: Users can increase or decrease the stock of products.

  - Delete Products: Retailers can remove products from the system, ensuring no active stock links are present.

  - Stock Limit Check: Prevents adding stock beyond the predefined limit.

- Data Validation:

  - Ensures that stock levels do not exceed the limit.

  - Error messages are displayed when users try to perform invalid actions.

Non-Functional Requirements

Performance:

- The system should handle a large number of products and users without significant performance degradation.

- Operations such as adding, updating, and deleting products should be executed in a timely manner.

Security:

- User passwords must be securely stored and encrypted.

- The login process should be protected to prevent unauthorized access.

Scalability:

- The system should be able to handle future enhancements, such as adding more features or scaling up to accommodate more products and users.

Usaility: -

- The user interface should be intuitive and easy to navigate for retail users with minimal technical knowledge.

SOURCE CODE

SETTINGS.JSON

```
{
    "java.project.sourcePaths": ["src"],
    "java.project.outputPath": "bin",
    "java.project.referencedLibraries": [
        "lib/**/*.jar",
        "c:\\ani\\javafx\\javafx-sdk-23.0.1\\lib\\javafx.base.jar",
        "c:\\ani\\javafx\\javafx-sdk-23.0.1\\lib\\javafx.controls.jar",
        "c:\\ani\\javafx\\javafx-sdk-23.0.1\\lib\\javafx.fxml.jar",
        "c:\\ani\\javafx\\javafx-sdk-23.0.1\\lib\\javafx.graphics.jar",
        "c:\\ani\\javafx\\javafx-sdk-23.0.1\\lib\\javafx.media.jar",
        "c:\\ani\\javafx\\javafx-sdk-23.0.1\\lib\\javafx.swing.jar",
        "c:\\ani\\javafx\\javafx-sdk-23.0.1\\lib\\javafx.web.jar",
        "c:\\ani\\javafx\\javafx-sdk-23.0.1\\lib\\javafx-swt.jar",
        "lib/mysql-connector-j-9.1.0.jar"
    ]

}
```

LAUNCH.JSON

```jsonc
{
    // Use IntelliSense to learn about possible attributes.
    // Hover to view descriptions of existing attributes.
    // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
    "version": "0.2.0",
    "configurations": [
        {
            "type": "java",
            "name": "ProductManagementPage",
            "request": "launch",
            "mainClass": "ProductManagementPage",
            "projectName": "javaprj_6a54bf33"
        },
        {
            "type": "java",
            "name": "Current File",
            "request": "launch",
            "vmArgs": "--module-path \"C:/ani/javafx/javafx-sdk-23.0.1/lib\" --add-modules javafx.controls,javafx.controls,javafx.fxml,javafx.web",
            "mainClass": "${file}"
        },
        {
            "type": "java",
            "name": "App",
            "request": "launch",
            "vmArgs": "--module-path \"C:/ani/javafx/javafx-sdk-23.0.1/lib\" --add-modules javafx.controls,javafx.controls,javafx.fxml,javafx.web",
            "mainClass": "App",
            "projectName": "javaprj_6a54bf33"
        }
    ]
}
```

USERPROFILEPAGE.JAVA

```
// import javafx.application.Application;
// import javafx.geometry.Pos;
// import javafx.scene.Scene;
// import javafx.scene.control.*;
// import javafx.scene.layout.*;
// import javafx.stage.Stage;

// import java.sql.*;

// public class UserProfilePage extends Application {

//     private Connection connection;
//     private int userId;
//     private TextField usernameField;
//     private PasswordField passwordField;

//     // Constructor to pass userId
//     public UserProfilePage(int userId) {
//         this.userId = userId;
//     }

//     @Override
//     public void start(Stage primaryStage) {
//         // Connect to the database
//         connectToDatabase();

//         // Create UI elements
//         Label titleLabel = new Label("User Profile");
//         titleLabel.setStyle("-fx-font-size: 20px; -fx-font-weight: bold;");

//         // Create input fields for user data
//         usernameField = new TextField();
//         passwordField = new PasswordField();
```

```java
public class UserProfilePage extends Application {
    private void updateUserInfo() {
        String query = "UPDATE users SET username = ?, password = ? WHERE user_id = ?";

        try (PreparedStatement preparedStatement = connection.prepareStatement(query))
            preparedStatement.setString(1, username);
            preparedStatement.setString(2, password);
            preparedStatement.setInt(3, userId);

            int rowsUpdated = preparedStatement.executeUpdate();

            if (rowsUpdated > 0) {
                Alert alert = new Alert(Alert.AlertType.INFORMATION);
                alert.setTitle("Success");
                alert.setHeaderText(null);
                alert.setContentText("Your profile has been updated.");
                alert.showAndWait();
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    @Override
    public void stop() throws Exception {
        if (connection != null && !connection.isClosed()) {
            connection.close();  // Close connection when the app stops
        }
        super.stop();
    }

    public static void main(String[] args) {
        launch(args);
    }
}
```

PRODUCTMANAGEMENT.JAVA

```
// import javafx.application.Application;
// import javafx.geometry.Pos;
// import javafx.scene.Scene;
// import javafx.scene.control.*;
// import javafx.scene.layout.*;
// import javafx.stage.Stage;
// import javafx.collections.FXCollections;
// import javafx.collections.ObservableList;
// import javafx.scene.input.KeyEvent;
// import javafx.beans.property.*;

// import java.sql.*;

// public class ProductManagementPage extends Application {

//     private Connection connection;
//     private TableView<Product> productTable;
//     private ObservableList<Product> productList;
//     private TextField searchField;

//     public ProductManagementPage() {
//         connectToDatabase();
//     }

//     @Override
//     public void start(Stage primaryStage) {
//         // Create UI components
//         Label titleLabel = new Label("Product Management");
//         titleLabel.setStyle("-fx-font-size: 20px; -fx-font-weight: bold;");
```

```
    public static class Product {
        private final StringProperty sku;
        private final StringProperty name;
        private final IntegerProperty quantity;
        private final DoubleProperty price;
        private final StringProperty supplier;

        public Product(String sku, String name, int quantity, double price, String supplier) {
            this.sku = new SimpleStringProperty(sku);
            this.name = new SimpleStringProperty(name);
            this.quantity = new SimpleIntegerProperty(quantity);
            this.price = new SimpleDoubleProperty(price);
            this.supplier = new SimpleStringProperty(supplier);
        }

        public String getSku() { return sku.get(); }
        public String getName() { return name.get(); }
        public int getQuantity() { return quantity.get(); }
        public double getPrice() { return price.get(); }
        public String getSupplier() { return supplier.get(); }

        public void setName(String name) { this.name.set(name); }
        public void setQuantity(int quantity) { this.quantity.set(quantity); }
        public void setPrice(double price) { this.price.set(price); }
        public void setSupplier(String supplier) { this.supplier.set(supplier); }

        public StringProperty skuProperty() { return sku; }
        public StringProperty nameProperty() { return name; }
        public IntegerProperty quantityProperty() { return quantity; }
        public DoubleProperty priceProperty() { return price; }
        public StringProperty supplierProperty() { return supplier; }
    }
}
```

DASHBOARDPAGE.JAVA

```
// public class DashboardPage extends Application {

//     private Connection connection;
//     private int userId;

//     // Constructor to pass userId
//     public DashboardPage(int userId) {
//         this.userId = userId;
//     }

//     @Override
//     public void start(Stage primaryStage) {
//         // Connect to the database
//         connectToDatabase();

//         // Create UI elements
//         Label titleLabel = new Label("Dashboard - Inventory Overview");
//         titleLabel.setStyle("-fx-font-size: 20px; -fx-font-weight: bold;");

//         TextArea alertsArea = new TextArea();
//         alertsArea.setEditable(false);
//         alertsArea.setStyle("-fx-font-size: 14px; -fx-text-fill: red;");

//         // Fetch inventory data
//         List<String> alerts = getStockAlerts(userId);

//         // Display the alerts in the text area
//         alertsArea.setText(String.join("\n", alerts));
```

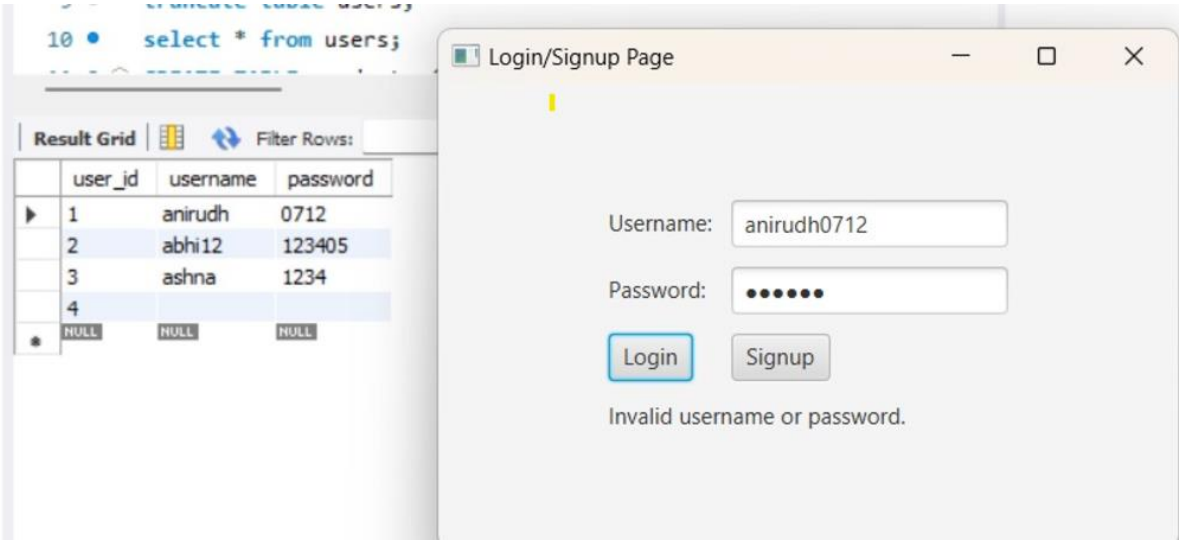
```
        while (resultSet.next()) {
            String productName = resultSet.getString("product_name");
            int quantity = resultSet.getInt("quantity");
            int minStock = resultSet.getInt("min_stock");
            int maxStock = resultSet.getInt("max_stock");

            // Low stock alert
            if (quantity < minStock) {
                alerts.add("Low stock alert for " + productName + ": Only " + quantity + " left (Min stock: " + minStock + ")");
            }
            // Max stock alert
            if (quantity > maxStock) {
                alerts.add("Max stock alert for " + productName + ": " + quantity + " exceeds max limit of " + maxStock);
            }
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }

    if (alerts.isEmpty()) {
        alerts.add("All inventory levels are within normal limits.");
    }

    return alerts;
}
```

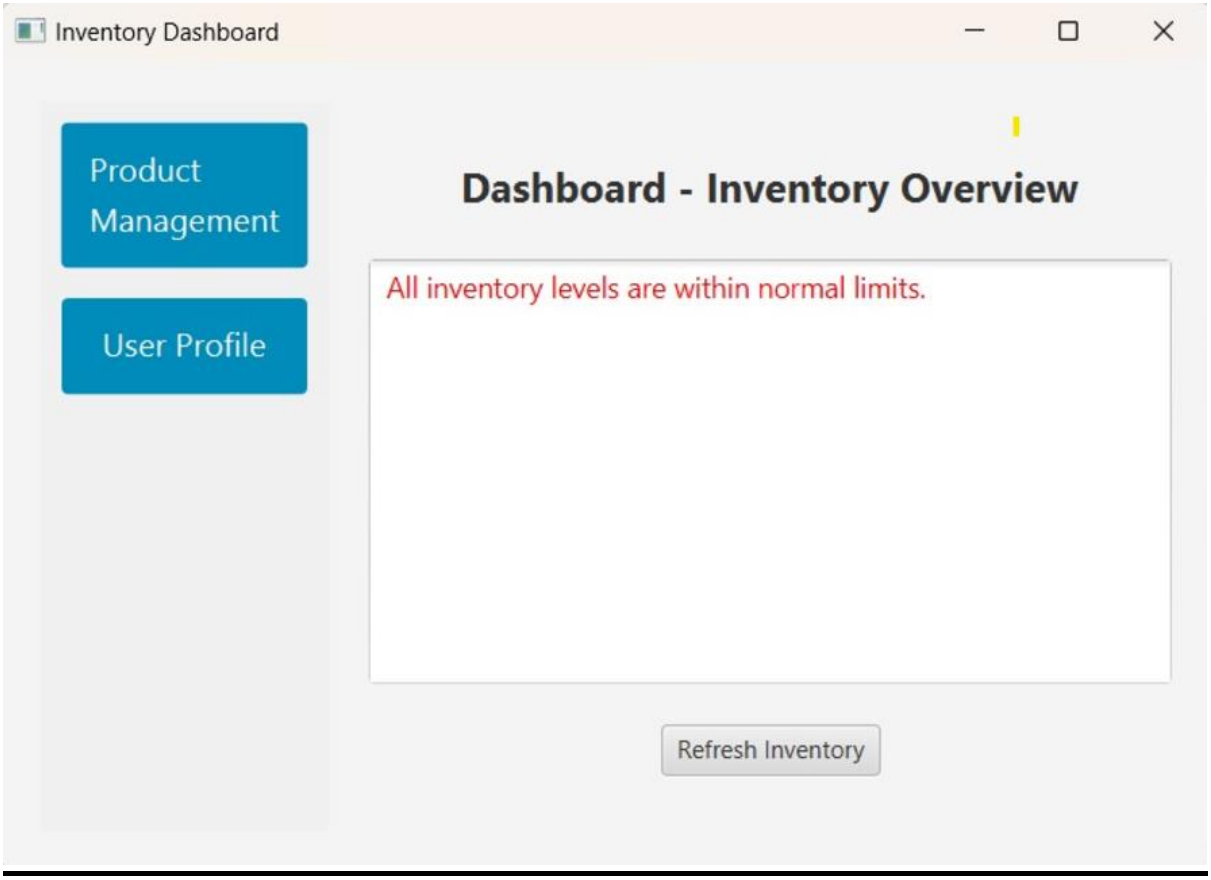
APP.JAVA

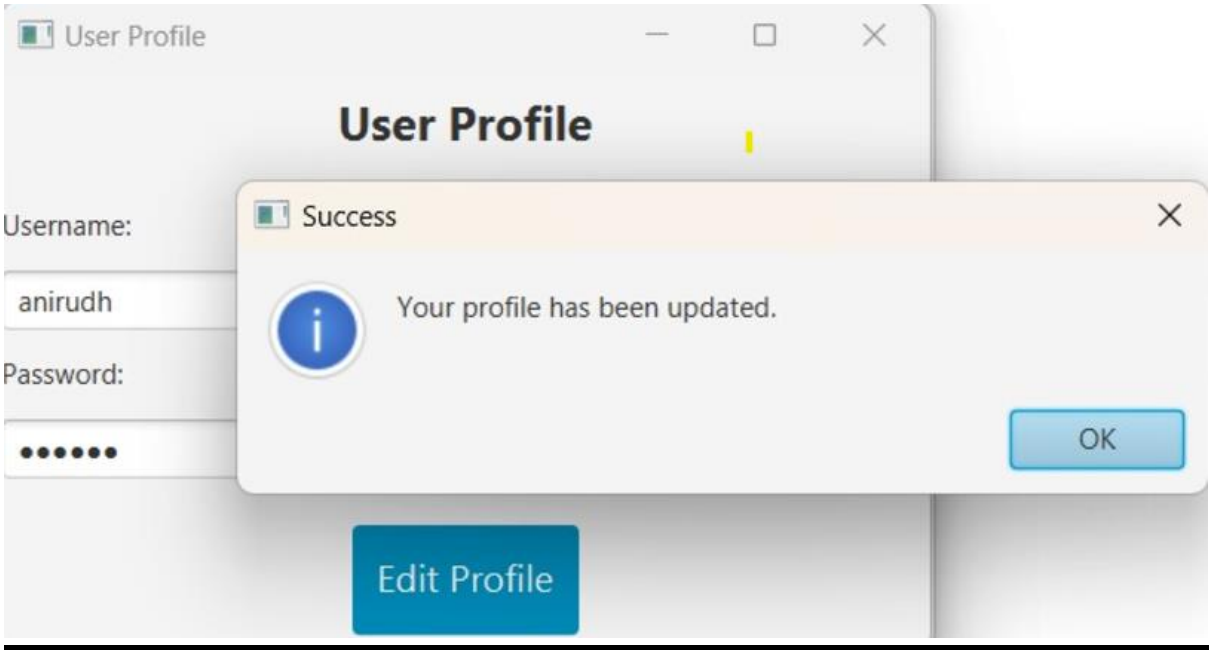
```java
import javafx.application.Application;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.GridPane;
import javafx.stage.Stage;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class App extends Application {

    private Connection connection;

    @Override
    public void start(Stage primaryStage) {
        // Connect to MySQL
        connectToDatabase();

        // Create UI elements
        Label usernameLabel = new Label("Username:");
        TextField usernameField = new TextField();

        Label passwordLabel = new Label("Password:");
        PasswordField passwordField = new PasswordField();

        Button loginButton = new Button("Login");
        Button signupButton = new Button("Signup");

        Label messageLabel = new Label();
```

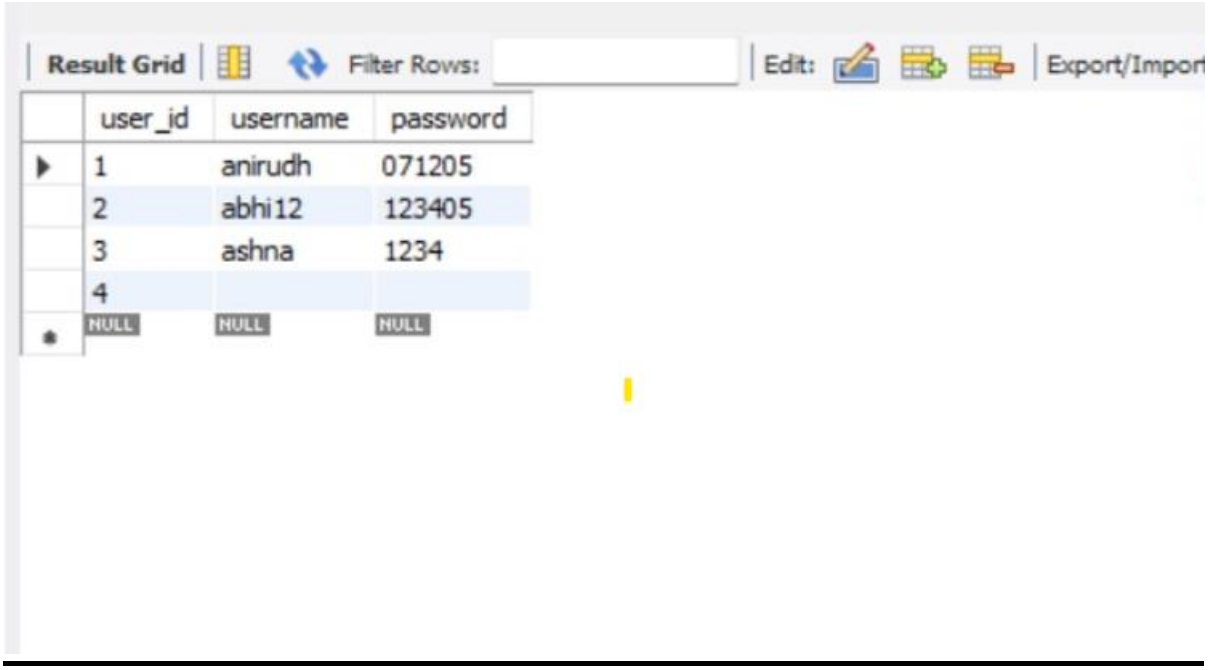
```
// Set up layout
GridPane gridPane = new GridPane();
gridPane.setAlignment(Pos.CENTER);
gridPane.setVgap(10);
gridPane.setHgap(10);

gridPane.add(usernameLabel, 0, 0);
gridPane.add(usernameField, 1, 0);
gridPane.add(passwordLabel, 0, 1);
gridPane.add(passwordField, 1, 1);
gridPane.add(loginButton, 0, 2);
gridPane.add(signupButton, 1, 2);
gridPane.add(messageLabel, 0, 3, 2, 1);

// Set button actions
loginButton.setOnAction(e -> {
    int userId = loginUser(usernameField.getText(), passwordField.getText());
    if (userId != -1) {
        messageLabel.setText("Login successful!");
        // Navigate to DashboardPage with the userId
        openDashboardPage(userId);
    } else {
        messageLabel.setText("Invalid username or password.");
    }
});

signupButton.setOnAction(e -> {
    if (signupUser(usernameField.getText(), passwordField.getText())) {
        messageLabel.setText("Signup successful!");
        // After successful signup, get the userId and go to DashboardPage
        int userId = getUserId(usernameField.getText());
```

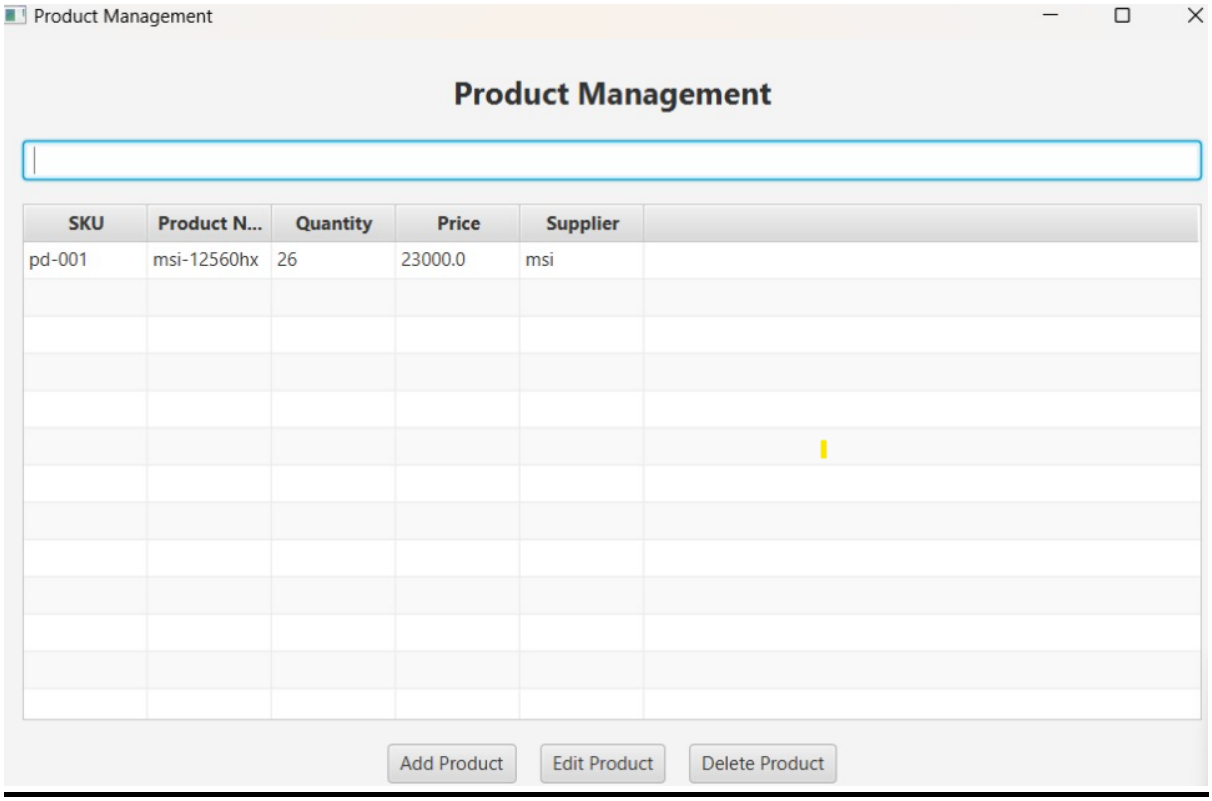
OUTPUT

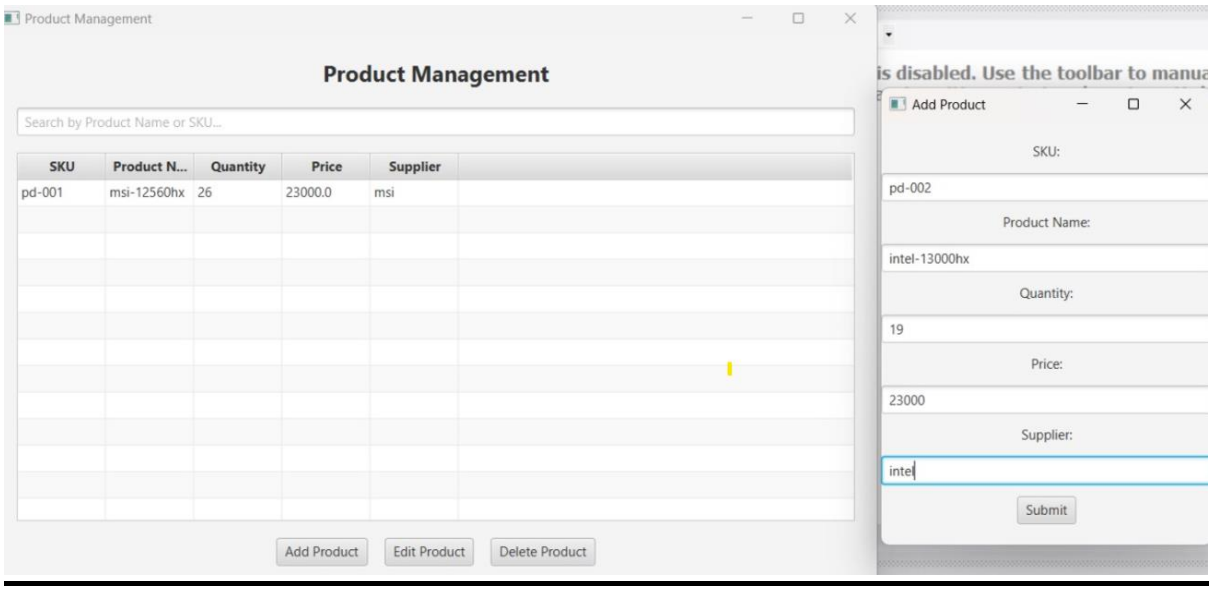

10 select * from users;
Result Grid
Filter Rows:
user_id username password
1 anirudh 0712
2 abhi12 123405
3 ashna 1234
4
NULL NULL NULL
Login/Signup Page
Username: anirudh0712
Password: ••••••
Login
Signup
Invalid username or password.

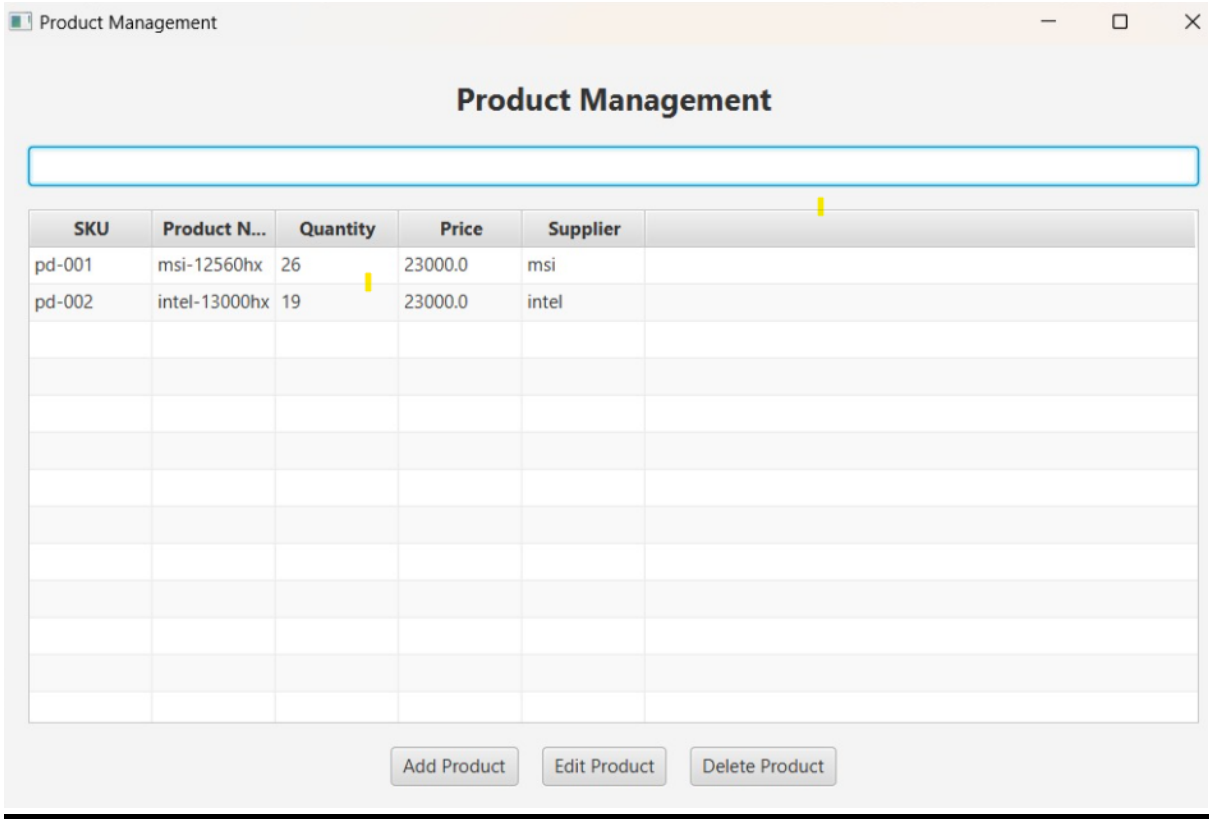

Inventory Dashboard
Product Management
User Profile
Dashboard - Inventory Overview
All inventory levels are within normal limits.
Refresh Inventory

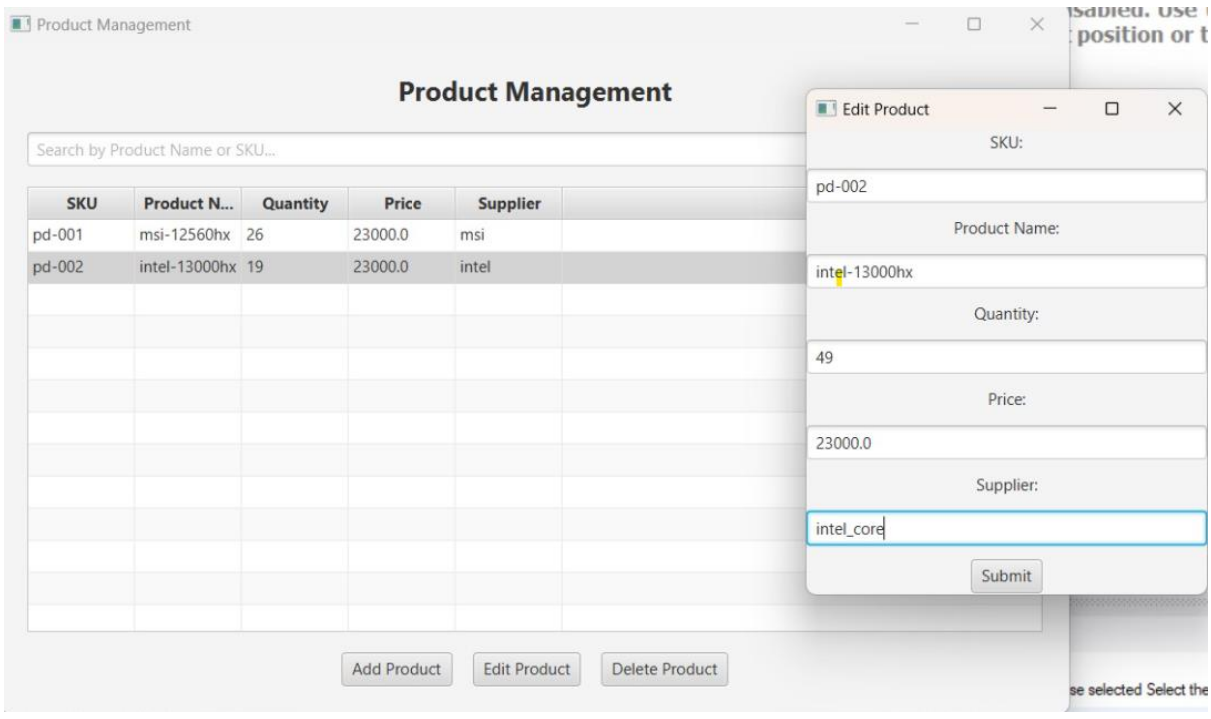

User Profile
User Profile
Username:
anirudh
Password:
Success
Your profile has been updated.
OK
Edit Profile

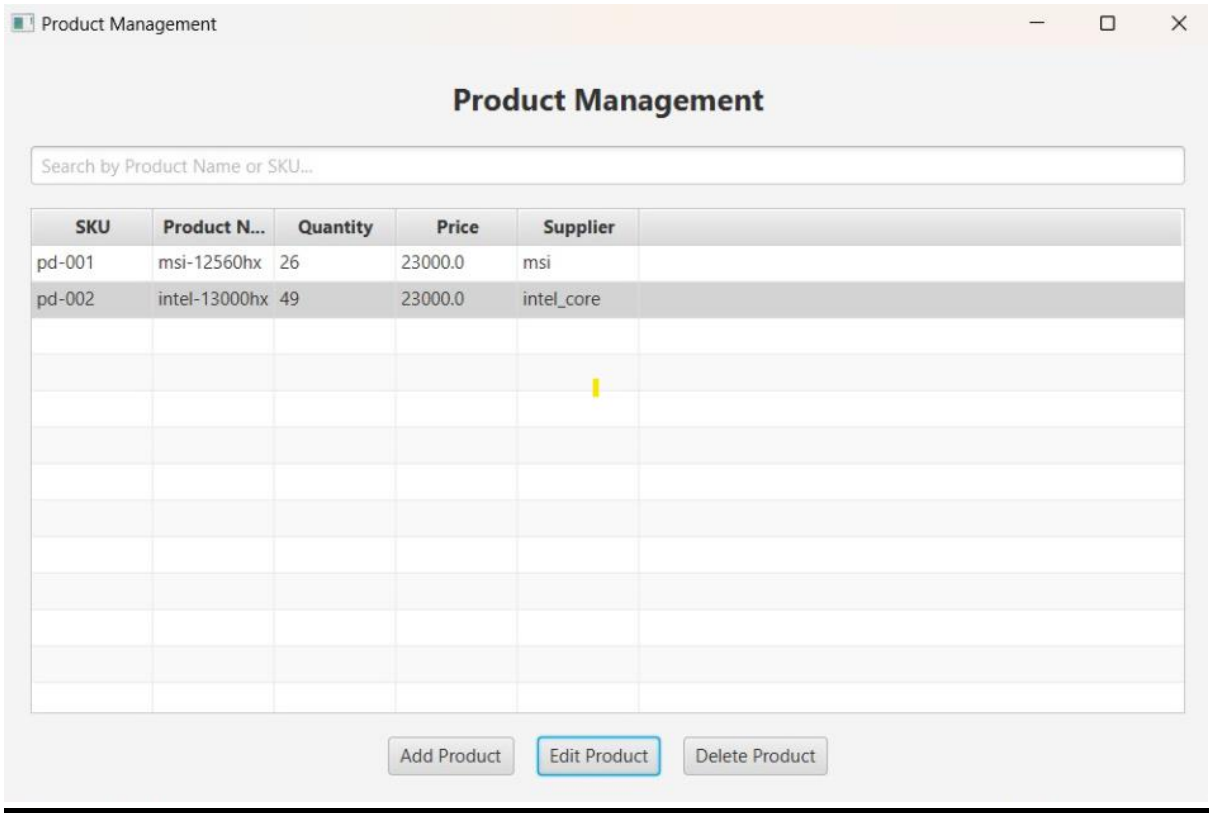

Result Grid
Filter Rows:
Edit:
Export/Import
user_id
username
password
1
anirudh
071205
2
abhi12
123405
3
ashna
1234
4
NULL
NULL
NULL


Product Management
Product Management
SKU
Product N...
Quantity
Price
Supplier
pd-001
msi-12560hx
26
23000.0
msi
Add Product
Edit Product
Delete Product


Product Management
Product Management
Search by Product Name or SKU...
SKU
Product N...
Quantity
Price
Supplier
pd-001
msi-12560hx
26
23000.0
msi
Add Product
Edit Product
Delete Product
is disabled. Use the toolbar to manua
Add Product
SKU:
pd-002
Product Name:
intel-13000hx
Quantity:
19
Price:
23000
Supplier:
intel
Submit


Product Management
Product Management
SKU
Product N...
Quantity
Price
Supplier
pd-001
msi-12560hx
26
23000.0
msi
pd-002
intel-13000hx
19
23000.0
intel
Add Product
Edit Product
Delete Product


Product Management
Product Management
Search by Product Name or SKU...
SKU
Product N...
Quantity
Price
Supplier
pd-001
msi-12560hx
26
23000.0
msi
pd-002
intel-13000hx
19
23000.0
intel
Add Product
Edit Product
Delete Product
Edit Product
SKU:
pd-002
Product Name:
intel-13000hx
Quantity:
49
Price:
23000.0
Supplier:
intel_core
Submit


Product Management
Product Management
Search by Product Name or SKU...
SKU
Product N...
Quantity
Price
Supplier
pd-001
msi-12560hx
26
23000.0
msi
pd-002
intel-13000hx
49
23000.0
intel_core
Add Product
Edit Product
Delete Product