Adv Java Mini Project: Inventory Management System (IMS)

Prerequisites (To Complete Before Day 1)

Technical Setup

- Java JDK (17 or later recommended)
- MySQL installed and running
- MySQL Workbench / DBeaver (optional GUI)
- MySQL JDBC Driver (Connector/J)
- IntelliJ IDEA / Eclipse set up
- Swing knowledge (JFrame, JPanel, JTable, etc.)
- JDBC basics (establishing DB connections)

Project Setup

- Create MySQL database (`inventory_db`)
- Tables: `users` , `products` , `stock_transactions`
- Set up Git repository (optional but recommended)
- Organize project folder:

```
/src
  /gui
  /db
  /models
  /utils
```

Must-Have Features

1. User Login/Authentication
2. Product Management
   - Add Product
   - View Product
   - Edit Product
   - Delete Product
3. Stock Management
   - Record Stock-In
   - Record Stock-Out/Sale
   - Stock Adjustments
4. View Current Inventory Levels
5. Low Stock Alerts

6. Search and Filter Products

Good-to-Have Features (Optional)

- Export reports to PDF or CSV
- Date range filters for stock transactions
- Role-based access control (Admin, Manager)
- View Sales history
- UI improvements (e.g., dark mode)
- Backup and restore database functionality

5-Day Timeline

Day 1: Setup and Login System

- Set up MySQL database and required tables
- Configure JDBC connection
- Build Login GUI with backend validation
- Store user credentials securely (hash passwords if possible)

Goal: Working login screen connected to the database

Day 2: Product Management Module

- Design and implement GUI for:
  - Adding new products
  - Viewing all products using JTable
  - Editing and deleting products
- Implement database integration for all operations

Goal: Complete CRUD operations for product management

Day 3: Stock Management Module

- GUI and backend for:
  - Recording Stock-In transactions
  - Recording Stock-Out/Sales
  - Adjusting stock levels (admin-only)
- Update product quantities in real-time

Goal: Accurate stock tracking based on transactions

Day 4: Reports and Filtering

- Create an inventory overview screen
- Implement low stock alerts (e.g., highlight or notify below threshold)
- Enable search and filtering by name, category, or quantity

Goal: User-friendly reporting and search system

Day 5: Finalization and Testing

- Add validations, error handling, and feedback messages
- Polish UI layout and structure
- Test all features and fix bugs
- (Optional) Add export or backup features
- Prepare short demo and documentation

Goal: Stable, complete project ready for submission

Sample Table Structures

`users` Table

Field	Type
id	INT, PK, AI
username	VARCHAR(50)
password	VARCHAR(255)
role	VARCHAR(20)

`products` Table

Field	Type
id	INT, PK, AI
name	VARCHAR(100)
category	VARCHAR(50)
price	DECIMAL(10,2)
quantity	INT

`stock_transactions` Table

Field	Type
id	INT, PK, AI
product_id	INT, FK
type	VARCHAR(10)
quantity	INT
date	DATETIME

Let me know if you need SQL scripts, UI mockups, or Swing boilerplate code to get started.

```
create table users(id int primary key auto_increment, username varchar(50) , password varchar(255) , ro
```

```
create table products (id int primary key auto_increment, name varchar(100) , category varchar(50) , pric
```

```
create table stock_transaction(id int primary key auto_increment, product_id int not null ,type varchar(10)
```