



**INVENTRA - INTELLIGENT INVENTORY MANAGEMENT SYSTEM**

**MILESTONE-3**

**Rajeswari R**

# What is inventor?

Inventra is an Inventory Management System that automates stock tracking, forecasting, and order management.



## INTRODUCTION

1. Modern applications are divided into independent modules
2. Each module performs a specific responsibility
3. Modular design improves security, maintainability, and scalability
4. All modules work together to form a complete intelligent system



# MODULE OVERVIEW

- 1. Module 1: Controls system access using authentication and roles**
- 2. Module 2: Manages products, inventory, and stock calculations**
- 3. Module 3: Monitors system conditions and generates alerts**



**Objective**

Accurate stock tracking

Secure access

Easy billing

# Users

- Admin
- Inventory Manager
- Billing Staff
- Ware house Staff





# MODULE 1 - What is Authentication?

- Authentication means verifying the identity of the user
- It checks who is trying to login

User need to enter:

- Username / Email
- Password

Controls login and role-based access.

Login → Role Check → Access Granted/Denied

# Database Schema

Table :Role

| Column Name  | Type         | Description             |
|--------------|--------------|-------------------------|
| user_id (PK) | INT          | Unique user identifier  |
| name         | VARCHAR(100) | User full name          |
| email        | VARCHAR(150) | Unique email for login  |
| password     | VARCHAR(255) | Encrypted password      |
| role         | VARCHAR(50)  | Admin / Manager / Staff |
| created_at   | DATETIME     | Account creation time   |
| updated_at   | DATETIME     | Last update timestamp   |

## Table:roles

| Column Name  | Type        | Description    |
|--------------|-------------|----------------|
| role_id (PK) | INT         | Unique role ID |
| role_name    | VARCHAR(50) | Role type      |

## Table: User Roles

| Column Name  | Type        | Description    |
|--------------|-------------|----------------|
| role_id (PK) | INT         | Unique role ID |
| role_name    | VARCHAR(50) | Role type      |

# Functional Flow

AuthController -> AuthService -> UserRepository-> Database (User table)

**The AuthController receives the login request and passes it to the AuthService for authentication processing. The AuthService verifies user credentials and generates tokens by fetching user data from the Database (User table) through the UserRepository.**

# PSEUDOCODE-AUTHENTICATION MODULE

## Classes Involved

START

AuthController receives email **and** password

AuthService validates input

UserRepository fetches **User using** email

IF **User not found**

**RETURN** error

IF password does **not match**

**RETURN** error

JWTUtility generates token **for User**

**RETURN** login success **with** token

(EmailService **is** used separately **for** forgot password)

END

# LOGIN PAGE

```
public String validateLogin(String username, String password) {  
    if(username == null || username.isEmpty() || password == null || password.isEmpty()) {  
        return "Invalid Input";  
    }  
    User user = userRepo.findByUsername(username);  
    if(user == null) {  
        return "User Not Found";  
    }  
    if(!user.getPassword().equals(password)) {  
        return "Invalid Credentials";  
    }  
    switch(user.getRole()) {  
        case "Admin";  
            return "Welcome Admin!";  
            break;  
        case "Manager";  
            return "Welcome Manager";  
            break;  
        case "Staff";  
        default "Access Denied";  
    }  
}
```

Login

Username

JohnDoe

Password

.....

Login

Forgot Password?

Login using user names

Forgot Password?

# **Module 2: Product and Inventory Management System**

The purpose of this module

- this module is to allow the system to store complete product details such as

**1. Product name**

**2. Category**

**3. Quantity**

**4. Price**

**5. Suppliers**

**6. Reorder Level**

**7. Stock Transactions**

**Main Classes in this Module is:**

**1. Product**

**2. Product Controller**

**3. Product Service**

**4. Product Repository**

---

---

# • FLOW

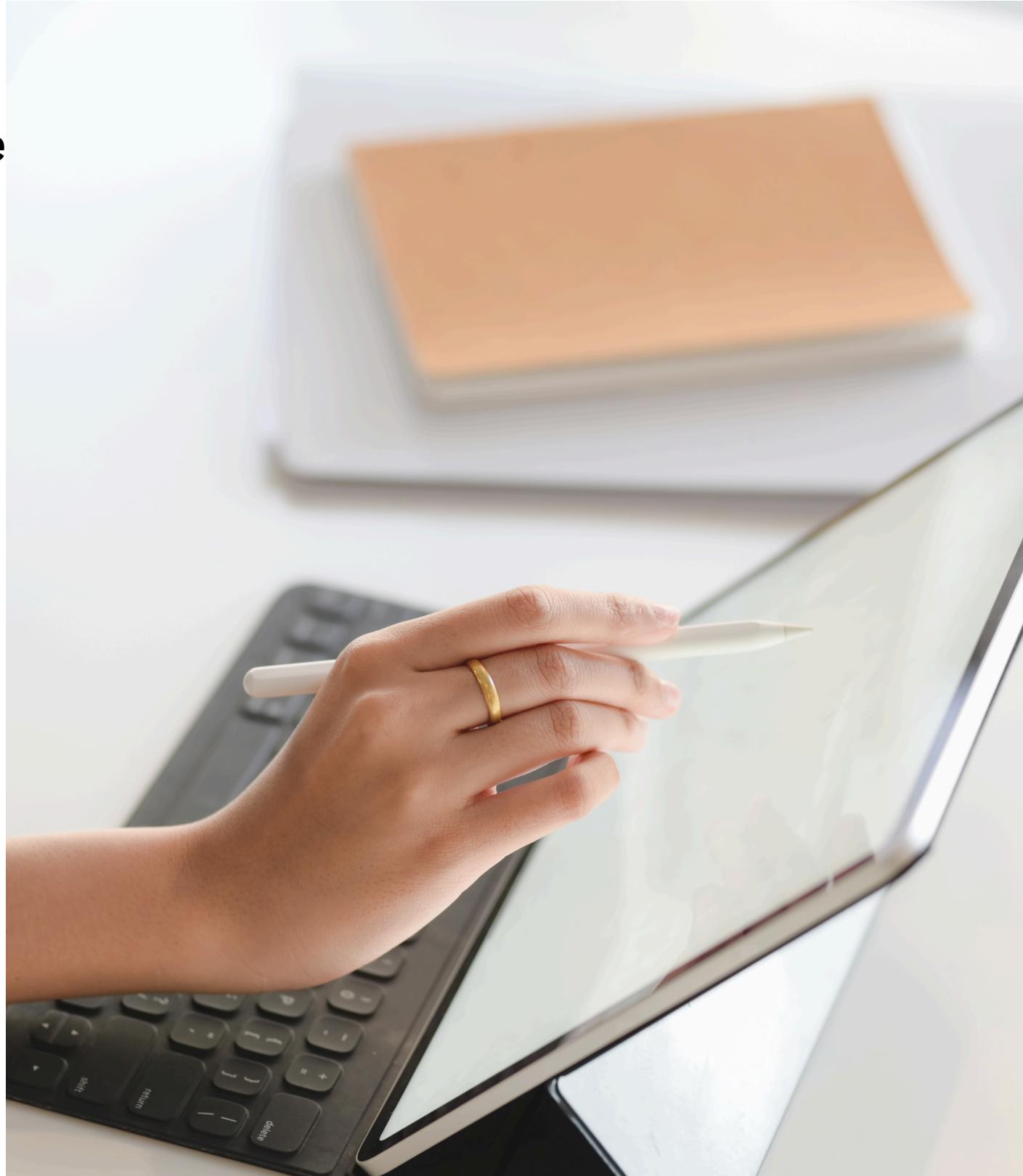
## **Flow: AuthController-AuthService-UserRepository--Database**

### **Purpose**

Manage products

Track stock levels

Prevent duplicate items



# Pseudocode : Manages products, inventory, and stock calculations

```
START Module2_ProductManagement

INPUT productId, quantity

Controller receives request

Service validates product and stock
IF product exists THEN
    IF quantity is valid THEN
        Repository updates stock in
database
        Service records transaction
    ELSE
        RETURN "Invalid Quantity"
    END IF
ELSE
    RETURN "Product Not Found"
END IF

RETURN "Stock Updated Successfully"

END Module2_ProductManagement
```

# Database Schema:

| <b>Class</b>    | <b>Need DB table?</b> | <b>Reason</b>   |
|-----------------|-----------------------|---|
| User (model)    | Yes                   | Stores user details and authentication info                   |
| User Repository | No                    | Only performs database queries for user data                  |
| User Service    | No                    | Business logic layer for user operations (e.g., registration) |
| User Controller | No                    | Request handling layer for user interface interactions        |

MileStone -3

Inventra Intelligent Inventory Management System

Module 3: Inventory/Alerts

---

# INTRODUCTION

- Core Function: It is responsible for monitoring inventory levels and automatically generating alerts.
- Mechanism: It works continuously in the background, checking stock data received from a separate inventory module.
- Benefit: Its primary purpose is to help users take timely action before stock runs out. By reducing the need for manual monitoring, it adds intelligence and efficiency to the inventory system.

Handles billing and alerts.

# PURPOSE:

- An Alert Module helps manage inventory by warning you about potential problems like running out of stock or having too much stock.
- It compares current stock levels to set limits and flags issues (risk management) so managers can take timely action, making operations more efficient.

Select Product → Quantity → Total

# Key Classes:

- **Product:** A data object that represents an inventory item, including its stock level and alert threshold.
- **Alert:** A data record created when an inventory threshold is breached.
- **AlertController:** The interface handling user requests related to alerts.
- **AlertService:** The core business logic layer responsible for monitoring stock and managing alerts.
- **AlertRepository:** The data access layer that manages product and alert records in the database.

# Functional Flow

**ALERT SERVICE---PRODUCT REPOSITORY----CHECKS QUANTITY < REORDER LEVEL**

**ALERTSERVICE ----ALERT REPOSITORY-CREATES ALERT**

**ALERT CONTROLLER----SHOWS ALERT TO UI**

- **Check:** The AlertService asks the ProductRepository for stock levels.
- **Compare:** The Service checks if the stock is too low.
- **Save:** If stock is low, the AlertRepository saves a permanent record of the alert
- **Show:** The AlertController grabs that record and shows a warning on the UI.

# INVENTORY ALERTS MODULE PSEUDOCODE

**AlertService Class**

**Pseudocode**

**CLASS AlertService**

**FUNCTION checkLowStock(product)**

**IF product.stock < product.minStockLevel THEN**

**PURPOSE**

**NotificationService.sendAlert(product)**

**1.TO AVOID STOCK-OUT**

**END IF**

**2.TO INFORM RESPONSIBLE STAFF**

**END FUNCTION**

**END CLASS**

# NOTIFICATIONSERVICE CLASS

Pseudocode

**CLASS Notification Service**

**FUNCTION sendAlert(product)**

**show dashboard alert**

**send email or SMS**

**Explanation:**

**END FUNCTION**

- **Uses communication channels**
- **Improves operational efficiency**

**END CLASS**

# KEY BENEFITS

- **Prevents Stockouts:** Ensures timely reorders when stock reaches a critical point.
- **Reduces Errors:** Eliminates manual, error-prone tracking.
- **Enables Timely Action:** Allows users to react before stock runs out, maximizing sales and improving customer satisfaction.
- **Adds Intelligence:** Focuses management effort on items needing attention, boosting efficiency.

The module's main advantage is shifting the process from manual, reactive tracking to automated, proactive management.

# Conclusion:

This system provides a complete solution for inventory management.

- **Module 1** ensures secure login using role-based access.
- **Module 2** manages products and maintains accurate stock levels.
- **Module 3** handles billing and generates alerts for low stock.

Together, these modules reduce manual work, prevent errors, and improve system efficiency.



**Thank  
You**

