# **Inventory Control System**

# **Product Documentation (MVP)**

1. Product Overview

The Inventory Control System (ICS) is a lightweight web-based application designed for small businesses, retail shops, warehouses, and pharmacies to manage their stock efficiently. The system allows users to track inventory items, record stock in/out transactions, and generate simple reports.

The goal of the MVP is to provide **a working, usable system within 8 weeks** that covers the most essential inventory operations without unnecessary complexity.

1. Core MVP Features
   1. **User Authentication**

* Secure login/logout
* Role-based access (Admin, Staff)
  1. **Item Management**
* Add new item (name, code, category, unit, price)
* Update or delete items
* Search and list all items
  1. **Stock transactions**
* Record stock-in (purchases, new arrivals)
* Record stock-out (sales, usage, waste)
* Maintain running balance for each item
  1. **Reports**
* Current stock levels
* Low-stock alerts
* Transaction history by item/date range

## 3. Non-Goals (Not in MVP)

* Barcode generation/scanning.
* Invoice generation in PDF.
* Multi-location inventory tracking.
* Mobile app.
* Supplier and purchase order management.

## 4. User Roles

## **Admin**

## Full access to all features.

## Can add/edit/delete users.

## **Staff**

## Limited access to stock in/out and viewing reports.

## Cannot delete items or manage users.

## 5. User Flows

### **Flow 1: Login**

User navigates to login page.

Enters credentials → API validates → JWT token returned.

Redirect to dashboard.

### **Flow 2: Item Management**

Admin navigates to **Items** page.

Clicks **Add Item**, fills form, saves.

Item appears in inventory list.

### **Flow 3: Stock In/Out**

User selects **Stock Transaction**.

Enters item, quantity, type (IN/OUT).

API updates stock levels.

Inventory list reflects new balance.

### **Flow 4: Reports**

User navigates to **Reports**.

Selects report type (Current Stock / Low Stock / Transaction History).

System fetches and displays results in a table.

## **6. Technology Stack**

**Frontend:** Angular 20 + Angular Material

**Backend:** .NET Core 9 Web API (C#)

**Database:** PostgreSQL 17

**Authentication:** JWT-based auth

**Deployment:**

Backend → Azure App Service / Docker container

Frontend → Azure Static Web App / Nginx

## **7. High-Level Architecture**

[ Browser (Angular UI) ]

|

v

[ .NET Core Web API ]

|

v

[ PostgreSQL Database ]

## **8. Development Timeline (8 Weeks)**

**Week 1:** Setup projects, DB schema, layouts.

**Week 2:** User authentication (login/register).

**Week 3:** Item management CRUD.

**Week 4:** Stock in/out transactions.

**Week 5:** Reports (current stock, low stock, history).

**Week 6:** Role-based access (Admin/Staff).

**Week 7:** Polishing (error handling, navigation, seed data).

**Week 8:** Deployment (Azure, demo ready).

## **9. Future Enhancements (V1.1+)**

Barcode printing and scanning.

Invoice/PDF generation.

Multi-location warehouse support.

Supplier and purchase order management.

Dashboard with charts/analytics.

Mobile-friendly responsive design.