



Smart Order & Inventory System

A secure, role-driven platform for efficient order management, inventory control, and business analytics.

SANJANA BISWAS

06/01/2025

Project Overview

Title: Smart Order & Inventory Management System

Objective:

A full-stack enterprise web application designed to streamline order processing, inventory tracking, and business reporting for a multi-warehouse retail organization.

Technology Stack:

ASP.NET Core Web API • Angular • SQL Server

Project Type:

Full-Stack Capstone Project

Problem Statement

Description of problem the project solves :

Business Challenges:

- Manual and delayed order processing
- No centralized inventory visibility
- Poor coordination between sales, warehouse, and finance teams
- Lack of analytical insights for decision-making

Need:

A secure, scalable, role-based system that integrates **orders, inventory, users, and reports** into a single platform

Proposes Solution

A modern, API-driven web application with role-based access and real-time data visibility.

Key Highlights:

- JWT-based authentication & authorization
- Centralized inventory and order lifecycle management
- Role-specific dashboards
- Analytical reports for business insights

Solution & Key Features

Order Management

- Create, track, cancel orders
- Complete lifecycle: Created → Approved → Packed → Shipped → Delivered

Inventory Management

- Stock tracking per warehouse
- Automatic stock reservation & deduction
- Low-stock alerts

Security & Roles

- Admin, Sales Executive, Warehouse Manager,
- Finance Officer, Customer
- Role-based access control

Reports & Analytics

- Sales by date & product
- Inventory status
- Top-selling products

Technology Stack

Backend

- ASP.NET Core Web API (.NET 8)
- Entity Framework Core + LINQ
- JWT Authentication
- Swagger for API documentation
- Layered architecture (Controllers, Services, DTOs)

Frontend

- Angular
- Angular Material
- Reactive Forms
- HTTP Interceptors & Route Guards

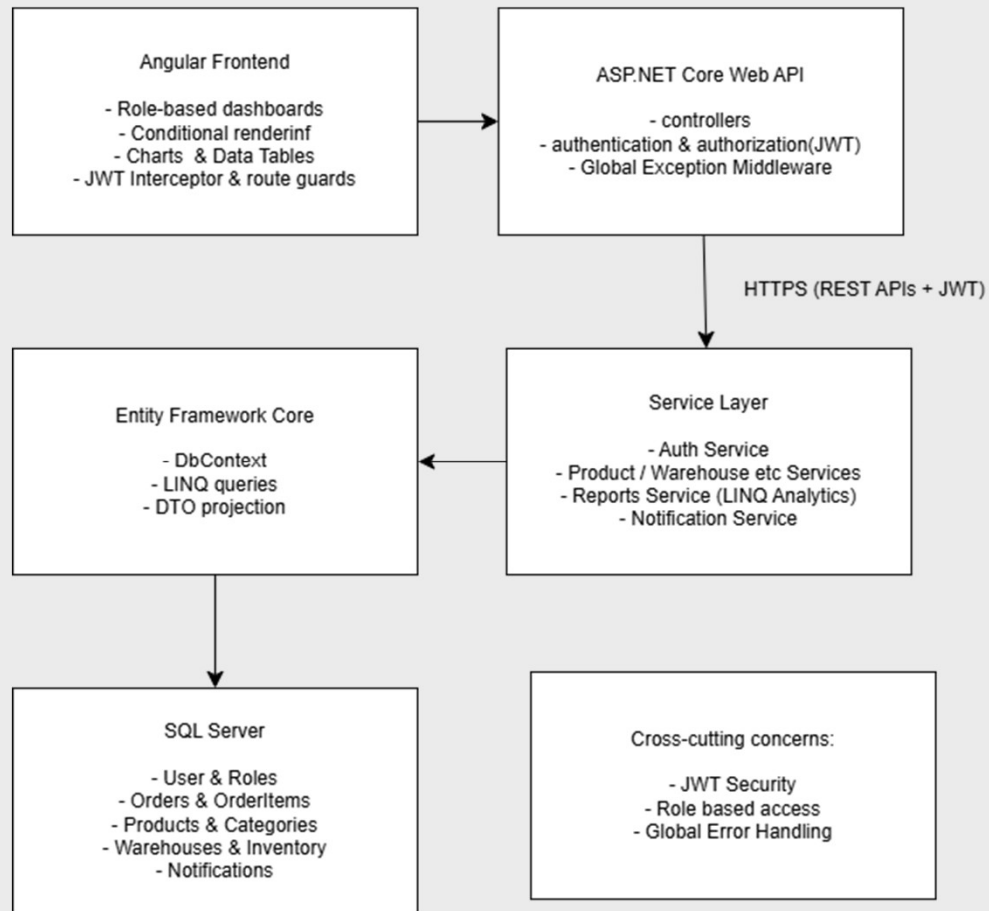
Roles

- Admin, Sales Executive, Warehouse Manager, Finance Officer, Customer
- Role-based access control

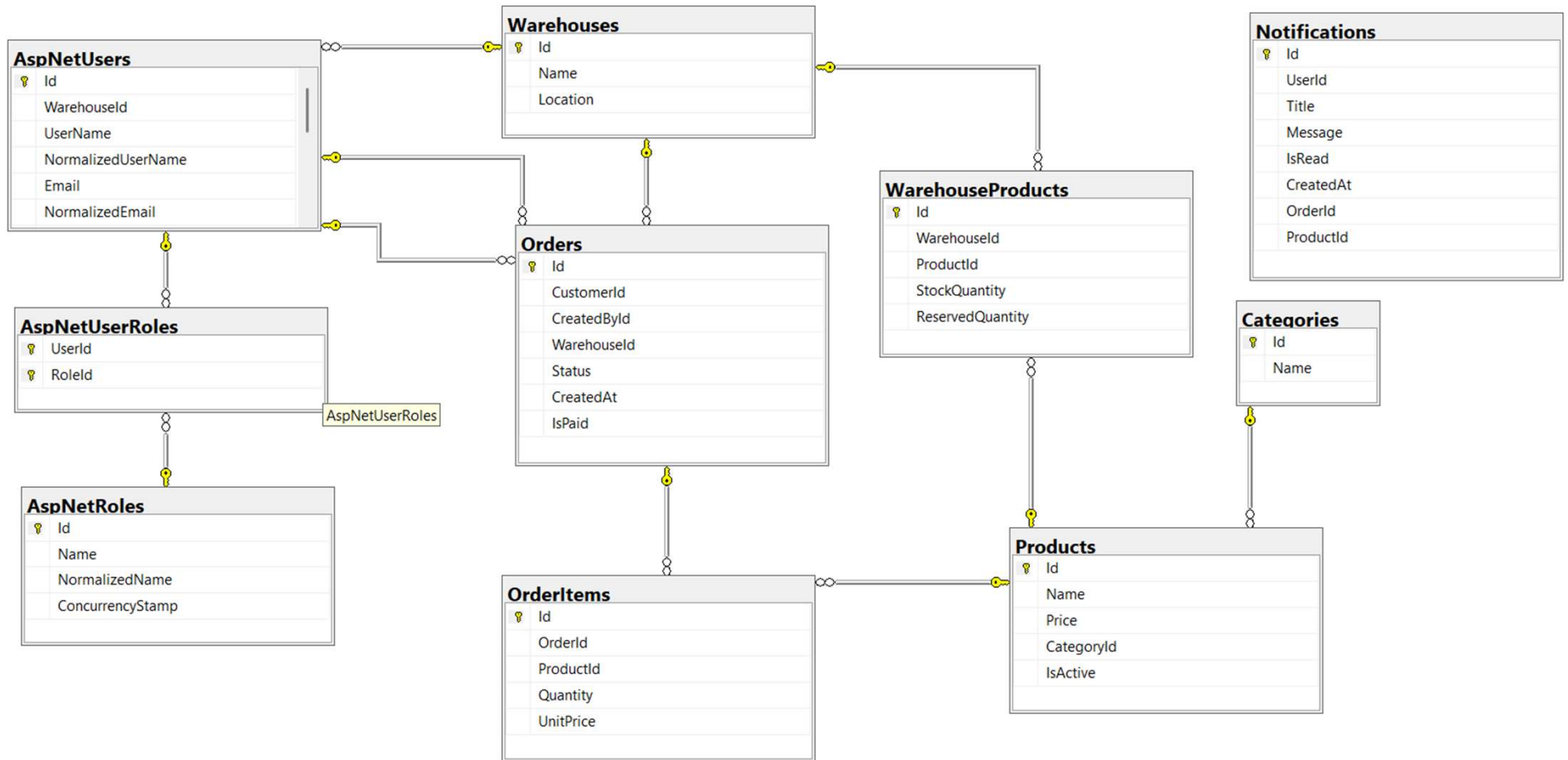
Database

- SQL Server
- Relational schema with constraints

System Architecture





Database Design



Admin

StockSphere

DashboardUsersCategoriesProductsWarehouses



Welcome Admin !

Overview of users, inventory, and warehouse operations

Total Users

13

Admins

1

Warehouse Managers

3

Finance Officers

1

Sales Executive

2

Customers

6

Warehouses

9

Products

39

Categories

18

User Management Screen

StockSphere

Dashboard

Users

Categories

Products

Warehouses

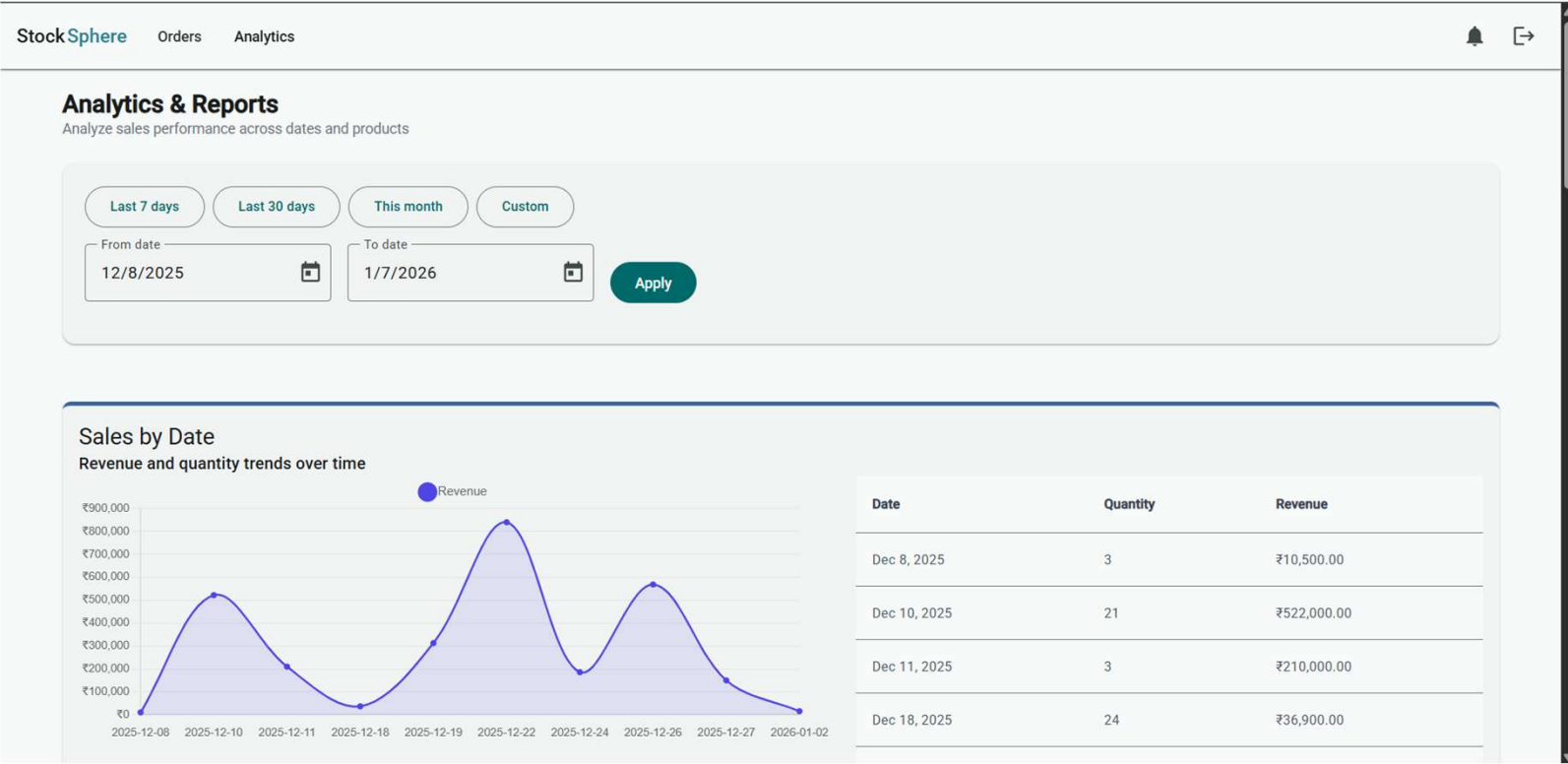
Users

Manage user roles

Search products

Email ↑	Role	Warehouse	
aarav.kumar@demo.com	Customer	N/A	
admin@sys.com	Admin	N/A	
ananya.singh@demo.com	Customer	N/A	
finance@sys.com	Financeofficer	N/A	
kavya.nair@demo.com	Customer	N/A	
neha.sharma@demo.com	Customer	N/A	
rahul.mehta@demo.com	Customer	N/A	

Reports & Analytics



Results & Achievements

- Designed a **normalized relational database** with well-defined entity relationships
- Implemented a **clean backend architecture** using controllers, services, DTOs, and middleware
- Used **LINQ with Entity Framework Core** for efficient server-side analytics and reporting
- Built a complete **order lifecycle and inventory management system** with warehouse-based stock handling
- Implemented a reusable **notification service** for order and inventory events
- Developed **role-based frontend dashboards** with component reusability and conditional rendering
- Visualized business data using **charts and tables** for analytics and KPIs

Live Demonstration

Live Demonstration
See the application in action

