ARTISELITE FULLSTACK ENGINEER - TECHNICAL CHALLENGE

Problem statement: Design and implement a **production-grade**, full-stack **Warehouse Management System** (WMS) that enables efficient digital operations across inventory, inbound, outbound, and user workflows. The system must be **containerized with Docker**, use **Tailwind CSS** for modern UI, and be **deployed to AWS EC2** with real-world deployment architecture with the following features:

COMPULSORY FEATURE

1. Inventory Management

- Add, update, archive, and delete products with fields: name, SKU, tags, description, category, quantity.
- Real-time inventory tracking with audit logs.
- Search/filter by keyword, tag, category, or SKU.
- Low stock alerts with thresholds per item.
- Bulk inventory upload via CSV/XLSX.

2. Inbound Management

- Log incoming stock: product, supplier, quantity, invoice/ref document, received date.
- Bulk inbound uploads via CSV/XLSX.
- File attachments (e.g., invoices, delivery orders).
- Automatically updates inventory.
- Associate to supplier records.

3. Outbound Management

- Record outbound transactions: product, quantity, customer, SO ref, date.
- Prevent negative stock dispatches.
- Bulk outbound uploads via CSV/XLSX.
- File attachment support (e.g., signed DOs).
- Real-time deduction from inventory.

4. User & Role Management

- Authentication & session/JWT-based authorization.
- Roles: Admin, Manager, Operator with clearly defined access levels.
- Granular permission matrix support (read/write/delete per module).
- Activity logs: who did what, when, and on what record.

5. Dashboard & Insights

- Stats: total inventory items, inbound/outbound today, low stock alerts.
- Recent activity stream.
- Daily transaction volume chart.

ADDITIONAL FEATURE (Implement at least 3)

1. Cycle Count & Reconciliation

- Allow periodic manual stock count input.
- Compare with system quantity.
- Log discrepancy and adjustment reason.

2. Barcode/QR Code Integration

- Generate barcode per SKU.
- Enable barcode scanning during inbound/outbound workflows.

3. Stock Forecasting

- Predict when an item will run out using historical outbound data.
- Forecast model can use basic ARIMA or rolling average.

4. Multi-Warehouse Support

- Each product has per-warehouse quantity.
- Allow internal stock transfer between warehouses.

5. Batch & Expiry Tracking

- Allow batch ID and expiry per stock entry.
- Support FIFO outbound fulfillment by batch expiry.

6. Webhook & Integration Hooks

- Emit real-time webhooks on:
    - Inventory threshold breach
    - Successful bulk upload
    - Inbound/outbound creation

7. **Audit Dashboard**

- Visual log of user activity.
- Track most edited items, frequent stock adjustments, etc.

8. **Real-Time Notifications**

- Optional: Use WebSocket or polling to update front-end when:
    - New stock is received
    - A dispatch is initiated
    - Low stock alert is triggered

9. **Inventory Valuation**

- Track product cost (by batch or average).
- Show total stock valuation per warehouse or product group.

Tech Stack

Backend: Django, Flask, Express, or Spring Boot

Frontend: Tailwind CSS + your choice of React, Vue, Svelte, or vanilla JS

Database: PostgreSQL or MySQL

Containerization: Docker + Docker Compose

Deployment: AWS EC2 instance with:

- Reverse proxy (Nginx or Apache)
- HTTPS (Let's Encrypt)
- Gunicorn/PM2/WSGI for serving

You will be assessed on these criteria:

Category	Details
System Design	Modular architecture, scalable DB design, separation of concerns
Feature Completeness	All compulsory features + at least 3 advanced features
UI/UX	Clean, responsive Tailwind UI, intuitive flow
Code Quality	Readable, well-structured, documented
Security	Auth, file validation, environment variable handling
Deployment	Properly hosted on AWS with Docker, HTTPS, reverse proxy
Documentation	Setup, usage guide, system overview
Bonus	PWA, CI/CD pipeline, mobile-first UI, additional integrations

END