Netwest

-------------   
  
Pharmacy Inventory Management System

# Features

## 1. Basic Inventory Tracking

• Store basic information for each drug: drug name, quantity in stock, and price.  
• View a simple list of all medications currently in stock, with quantities displayed.  
• Search for specific drugs by name or category (e.g., antibiotics, pain relievers).

## 2. Stock Updates

• Manually update stock levels when drugs are received from suppliers or dispensed to customers.  
• Display an option to adjust stock (increase or decrease) when new deliveries arrive or sales happen.

## 3. Low Stock Alerts

• Automatically notify staff when a drug's quantity falls below a predefined threshold (e.g., when stock reaches 10 units).  
• Allow users to manually set a 'low stock' threshold for each drug.

## 4. Expiration Date Tracking

• Track expiration dates for each medication.  
• Display warnings when drugs are near expiration or expired (e.g., within 30 days of expiry).

## 5. Basic Reporting

• Generate a simple report showing all drugs that are low in stock.  
• View a basic report of total drugs dispensed within a specific time frame.

# Ideal Users for This Platform

• Pharmacy Staff: To manage inventory, add new stock, and ensure that medications are available for customers.  
• Pharmacy Owner/Manager: To monitor overall inventory, see low-stock warnings, and manage simple reporting tasks.

# Simplified Workflow Example

## Adding New Drug to Inventory

A pharmacy staff member can enter a new drug into the system, specifying the drug name, quantity, price, and expiration date.

## Updating Stock Levels

When new stock is delivered, the staff updates the quantity in the system. If drugs are dispensed to a customer, the stock quantity decreases automatically.

## Low Stock Alert

The system automatically flags any drugs that have low stock based on the set threshold, ensuring the pharmacy is aware of what needs to be reordered.

## Expiration Date Notification

The system alerts staff members when drugs are close to expiration, so they can remove or mark expired stock.

# Implementation Approach

• Frontend: Angular 19 with Signal Reactive Forms.  
• UI/Design: Angular Material for creating responsive and modern interfaces.  
• Backend: JSON Server for simulating REST API calls.  
• Architecture: Simplified client-server setup suitable for s

Key API Endpoints

1. **Authentication**
   * POST /api/auth/login - User login
   * POST /api/auth/register - User registration
2. **Drugs**
   * GET /api/drugs - Get all drugs (with filtering)
   * POST /api/drugs - Create new drug
   * GET /api/drugs/:id - Get single drug
   * PUT /api/drugs/:id - Update drug
   * DELETE /api/drugs/:id - Delete drug
   * PATCH /api/drugs/:id/stock - Adjust stock
3. **Suppliers**
   * GET /api/suppliers - Get all suppliers
   * POST /api/suppliers - Create new supplier
   * CRUD operations similar to drugs
4. **Transactions**
   * GET /api/transactions - Get all transactions
   * POST /api/transactions - Create new transaction
5. **Alerts**
   * GET /api/alerts - Get all alerts
   * PATCH /api/alerts/:id/read - Mark alert as read
6. **Reports**
   * GET /api/reports/low-stock - Get low stock report
   * GET /api/reports/expiration - Get expiration report
   * GET /api/reports/transactions - Get transaction report

Database (MongoDB with Mongoose)

Collections (As Previously Designed)

1. **users** - Pharmacy staff accounts
2. **drugs** - Medication inventory
3. **suppliers** - Medication suppliers
4. **drug\_suppliers** - Many-to-many relationship
5. **transactions** - Inventory movement records
6. **alerts** - System notifications

Key Features Implementation

1. **Low Stock Alerts**
   * Middleware that checks stock levels after transactions
   * Background job that runs daily to check all drugs
2. **Expiration Tracking**
   * Scheduled task to check expiring drugs daily
   * API endpoint to get drugs expiring soon
3. **Transaction History**
   * Automatic transaction recording on stock changes
   * Complete audit trail
4. **User Authentication**
   * JWT-based authentication
   * Role-based access control

Development Approach

Phase 1: Setup and Basic Structure

1. Initialize Angular and Node.js projects
2. Set up MongoDB connection
3. Create basic project structure
4. Implement authentication system

Phase 2: Core Inventory Management

1. Drug CRUD operations
2. Supplier management
3. Stock adjustment functionality
4. Basic dashboard

Phase 3: Advanced Features

1. Alert system
2. Reporting functionality
3. Transaction history
4. Advanced filtering and search

Phase 4: Polish and Testing

1. UI refinement with Angular Material
2. Form validation
3. Error handling
4. Testing and bug fixes

Key Technical Considerations

1. **Authentication & Authorization**
   * JWT tokens for API authentication
   * Route guards in Angular for frontend protection
   * Role-based access control
2. **Data Validation**
   * Mongoose schema validation
   * Express validator middleware
   * Angular form validation
3. **Error Handling**
   * Consistent error responses from API
   * Global error handler in Angular
   * User-friendly error messages
4. **Performance Optimization**
   * Pagination for large datasets
   * Selective field population
   * Proper indexing in MongoDB
5. **Real-time Features** (Optional)
   * WebSocket connections for real-time alerts
   * Live dashboard updates

maller pharmacies.