

Phase 1 – 6-Hour MVP PRD

SaaS Inventory Management System (Scope-Reduced)

0. Document

- Name: StockFlow MVP (6-hr build)
- Version: v0.1
- Goal: Ship a working, minimal SaaS inventory app in ~6 hours for demo/internal use.

1. Objective

Deliver a simplest-possible multi-tenant inventory app where a user can:

- Sign up / log in.
- Create and manage products (name, SKU, quantity, prices).
- See a basic dashboard with inventory summary and low-stock items.

No integrations, no multi-warehouse, no purchase orders, no advanced reporting.

2. Users & Personas (MVP)

- Single-role User (Owner/Admin for their own org)
 - Small seller or internal tester.
 - Needs a quick way to keep a list of products and how many units they have in stock.
 - Uses app from a laptop/desktop browser.

Role management, invites, and staff users are out of scope.

3. In-Scope Features (MVP)

3.1 Authentication & Tenant Basics

FR-1: Signup & Login

- User can sign up with:
 - Email
 - Password
 - Organization name (e.g., “My Test Store”)
- Email & password login with basic validation.
- Password reset is NOT required in MVP (can be done via DB/manual reset initially).

FR-2: Organization Context

- On signup, an Organization record is created and linked to the user.
- All data (products, inventory) is scoped by organization.
- No multi-user support in MVP; one user per org assumed.

3.2 Product & Inventory Management

FR-3: Product Model (Simple)

Each product has:

- ID
- Organization ID
- Name (required)
- SKU (required, unique per org)
- Description (optional)
- Quantity on Hand (integer)
- Cost Price (optional, numeric)
- Selling Price (optional, numeric)
- Low Stock Threshold (optional, default global value if empty)
- Created At / Updated At

No variants, no categories, no images in MVP.

FR-4: Product CRUD

- Create Product:
 - Form with Name, SKU, Quantity on Hand, Cost Price, Selling Price, Low Stock Threshold.
- Read:
 - Products list page with table:
 - Columns: Name, SKU, Quantity, Low Stock Indicator, Selling Price.
 - Product detail page (optional; or reuse edit form as detail view).
- Update:
 - Edit product form for any field including quantity.
- Delete:
 - Soft delete OR hard delete allowed in MVP (hard delete is acceptable for now).
 - Confirm dialog before delete.

FR-5: Simple Stock Updates

- Quantity on Hand can be changed via:
 - Direct edit in the product edit form.
 - (Optional) “Adjust stock” small inline control on the list or detail page:
 - Enter “+/- N units” and a note.
- No full movement history is required for MVP, but storing a simple “last_updated_by” and timestamp is recommended.

3.3 Dashboard

FR-6: Dashboard Overview

- After login, user lands on Dashboard.
- Show:
 - Total number of products.
 - Sum of Quantity on Hand across all products.
 - A small table/card section: “Low stock items”.
 - A product is “low stock” if Quantity on Hand \leq Low Stock Threshold.
 - If Low Stock Threshold is empty, ignore or use a global default (e.g., 5).
- Low stock table columns:
 - Name

- SKU
- Quantity on Hand
- Low Stock Threshold

No charts or complex analytics in MVP.

3.4 Settings (Minimal)

FR-7: Global Settings (Optional, Simple)

- Page “Settings” with just one field for now:
 - Default Low Stock Threshold (integer, e.g., 5).
- If a product’s Low Stock Threshold is empty, this default is used for the low-stock logic.

3.5 Non-Functional (MVP)

- Single-region deployment.
- Simple email/password auth with server-side sessions or JWT.
- Basic input validation on forms.
- Responsive-ish UI (works on desktop first; mobile responsiveness is optional).

4. Out-of-Scope for 6-Hour MVP

Explicitly excluded from this phase:

- Multi-warehouse support.
- Variants (size/color, etc.).
- Channel integrations (Shopify, WooCommerce, Amazon, etc.).
- Order management and reservations.
- Purchase orders and supplier management.
- Email notifications (low stock, failures, etc.).
- File imports (CSV upload).
- API access for external systems.
- Role-based access control beyond single owner user.
- Audit logs and detailed stock movement history.
- Billing / subscription management and payment integration.

5. Suggested Tech Stack (Optional Guidance)

(You can change based on team preference; this is just to make 6-hour execution realistic.)

Backend:

- Node.js + Express or NestJS
- PostgreSQL or MySQL (or even SQLite for fast prototype)
- Simple ORM (Prisma/TypeORM/Sequelize) for quick schema setup

Frontend:

- Next.js or React SPA
- Simple component library (e.g., Tailwind + basic components / any UI kit)

Auth:

- Simple session-based or JWT-based auth.
- Minimal hashing (bcrypt) for passwords.

6. Rough Screen List

1) Auth

- Login page
- Signup page (Email, Password, Confirm Password, Organization Name)

2) Dashboard

- Summary cards
- Low stock list

3) Products

- Product List:
 - Table with search by SKU or Name (simple client-side filter or basic backend query).
 - "Add Product" button.
- Create/Edit Product Form

4) Settings

- Global Default Low Stock Threshold

7. Success Criteria for 6-Hour MVP

- A new user can:
 - Sign up, log in, and see a dashboard.
 - Create at least one product with SKU and quantity.
 - See that product in the list and on the dashboard summary.
 - Mark some products as low stock (based on threshold) and see them in the low-stock section.
- All actions are scoped to their organization (no cross-tenant data leaks).
- App can be demoed end-to-end in a browser without manual DB edits.