HSUS Project Context v2: Technical Specs + CSV Schema

This document consolidates `System_Design_Spec_v2.1` and the actual schemas derived from the legacy CSV files (`customers` , `products` , `serial_numbers`).

1. Directory Structure & Tech Stack

Tech Stack:

- Runtime: Node.js 20+ (TypeScript)
- Framework: Express.js
- Database: Supabase (PostgreSQL) - Use `pg` library.
- Deployment: Google Cloud Run.

Directory Structure:

```
hsus-cloud-run/
├── src/
│   ├── config/           # database.ts (Pool setup)
│   ├── controllers/      # Route handlers
│   ├── services/         # Business logic
│   ├── repositories/     # SQL queries
│   ├── types/            # Interface definitions
│   ├── middlewares/      # errorHandler.ts
│   ├── utils/            # apiResponse.ts
│   └── app.ts            # Entry point
├── db/
│   └── migrations/       # SQL scripts
├── scripts/              # ETL scripts (for CSV import)
├── tests/
├── package.json
└── tsconfig.json
```

2. Database Schema (Source of Truth)

Create ENUMs FIRST:

1. `order_source_enum` : 'DEALER', 'QUOTE'
2. `order_status_enum` : 'DRAFT', 'CONFIRMED', 'ALLOCATING', 'PARTIALLY_SHIPPED', 'SHIPPED', 'COMPLETED', 'CANCELLED'
3. `invoice_status_enum` : 'OPEN', 'OVERDUE', 'PAID'

2.1 Physical Asset Tables (Matched to CSV Headers)

Table: `customers` (Source: `customers_rows.csv`)

- `id` (UUID, PK): `uuid_generate_v4()`
- `qbo_id` (TEXT, Unique): Corresponds to CSV column `qbo_id` .

- `company_name` (TEXT): CSV column `company_name` .
- `contact_email` (TEXT): CSV column `contact_email` .
- `payment_terms` (TEXT): CSV column `payment_terms` .
- `rsm_name` (TEXT): CSV column `rsm_name` .

Table: `products` (Source: `products_rows.csv`)

- `sku` (TEXT, PK): CSV column `sku` . (Primary Key).
- `model_name` (TEXT): CSV column `model_name` .
- `description` (TEXT): CSV column `description` .
- `unit_price` (DECIMAL): CSV column `unit_price` .
- `cost` (DECIMAL): CSV column `cost` .
- `category` (TEXT): CSV column `category` .
- `item_type` (TEXT): CSV column `item_type` .

Table: `serial_numbers` (Source: `serial_numbers_rows.csv`)

- `serial_no` (TEXT, PK): CSV column `serial_no` .
- `sku` (TEXT, FK): CSV column `sku` . References `products(sku)` .
- `warehouse_location` (TEXT): CSV column `warehouse_location` .
- `inbound_date` (TIMESTAMPTZ): CSV column `inbound_date` .
- `status` (TEXT): CSV column `status` .
- `bol_number` (TEXT): CSV column `bol_number` . (Link to shipments later).
- `po_sku_key` (TEXT): CSV column `po_sku_key` . (Legacy key for tracing).
- `created_at` (TIMESTAMPTZ): CSV column `created_at` .

2.2 Core Business Tables (Spec v2.1)

Table: `orders`

- `id` (UUID, PK)
- `order_number` (TEXT, Unique)
- `source` (ENUM)
- `status` (ENUM)
- `customer_info` (JSONB)
- `created_at` , `updated_at`

Table: `shipments`

- `id` (UUID, PK)
- `order_id` (UUID, FK) -> `orders(id)`
- `tracking_number` (TEXT)

- `shipped_at` (TIMESTAMPTZ)
- `items` (JSONB) `[{ "sku": "...", "qty": 1 }]`

3. API Contract (Simplified)

Global Response: `{ success: boolean, message: string, data: any, error: any }`

POST /api/v1/shipments

- Payload: `{ order_number, tracking_number, shipped_at, items: [{sku, qty}] }`
- Logic: Validate Order -> Insert Shipment -> Update Order Status.