Backend Engineering Intern Case Study – StockFlow

Part 1: Code Review & Debugging

Original Issues Identified:

1. Product should not include warehouse_id directly.
2. SKU uniqueness not enforced.
3. Missing field validation and error handling.
4. Price should be Decimal, not float.
5. No transaction block – risk of partial commit.
6. No handling of optional fields.

Impacts:

- Incorrect product-warehouse modeling.
- Duplicate SKUs can break stock tracking.
- Poor user experience and API crashes.
- Financial data inaccuracies.
- Data inconsistency in case of partial commit.

Fix:

A corrected version of the endpoint was implemented using proper validation, error handling, SKU uniqueness checks, and transaction-safe database operations.

Part 2: Database Design

Schema Highlights:

- Companies, Warehouses, Products, Inventory tables with relationships.
- Inventory logs to track changes.
- Supplier and ProductBundle relationships.
- Indexes and constraints for integrity and performance.

Key Tables :

- companies(id, name)
- warehouses(id, company_id, name)
- products(id, name, sku, price, threshold)
- inventory(product_id, warehouse_id, quantity)
- inventory_logs(product_id, warehouse_id, change, reason, created_at)
- suppliers(id, name, contact_email)
- supplier_products(supplier_id, product_id)
- product_bundles(bundle_id, product_id, quantity)

Part 3: API Implementation

GET /api/companies/{company_id}/alerts/low-stock

Returns a list of low-stock products across warehouses for a given company, considering recent sales activity and supplier info.

Key Features:

- Filters based on sales in the past 30 days.
- Joins inventory, product, warehouse, supplier info.
- Includes current stock, thresholds, and supplier contact.
- Handles missing suppliers gracefully.
- Assumes basic sales and product models.

Sample Output:


```
{
  "alerts": [
    {
      "product_id": 123,
      "product_name": "Widget A",
      "sku": "WID-001",
      "warehouse_id": 456,
      "warehouse_name": "Main Warehouse",
      "current_stock": 5,
      "threshold": 20,
      "days_until_stockout": 12,
```

```
      "supplier": {
        "id": 789,
        "name": "Supplier Corp",
        "contact_email": "orders@supplier.com"
      }
    }
  ],
  "total_alerts": 1
}
```


Assumptions:

- Sales table exists with created_at timestamps.
- Thresholds are defined globally per product.
- Randomized days_until_stockout logic placeholder.