

## **Requirements Document – E-Commerce Mini Backend**

By Jared Krekeler

**Purpose:** Build a complete e-commerce backend to support product browsing, customer accounts, credit cards on file, and purchasing via orders with multiple line items. Staff can add and manage products.

### **Scope & Actors**

- Customers: create an account, store multiple credit cards, search/browse products, place orders.
- Staff: create/edit products (name, category, price, inventory), view sales queries.
- System: persist data, enforce constraints (FKs, unique keys), support simple analytics queries.

### **Key Use Cases**

- Customer registration & login (CLI demo simulates this).
- Staff adds products / updates inventory.
- Customer saves one or more credit cards.
- Customer places an order with multiple products.
- Queries: high-value purchases, top customers, revenue by product, order details by date window.

### **Data Requirements**

- Customer(id, name, email, created\_at). Email being unique.
- CreditCard(id, customer\_id, brand, last4, exp\_month, exp\_year, nickname).
- Staff(id, name, email).
- Product(id, name, category, price, inventory\_qty, is\_active).
- Order(id, customer\_id, created\_at, total\_amount, status, payment\_method).
- OrderItem(id, order\_id, product\_id, quantity, unit\_price, line\_total).

### **Non-Functional Requirements**

- Relational integrity enforced via foreign keys.
- Indexes on common lookup fields (email, product name).
- Simple CLI for business logic demo.
- SQLite used for ease of grading; SQL written to be SQLite-compatible ideally...

### **Assumptions & Constraints**

- No payments processed; record a selected credit card nickname only.
- Inventory decremented at checkout; orders fail if insufficient inventory.
- Tax/shipping omitted for simplicity.
  - User login is existing and hashed on the front end