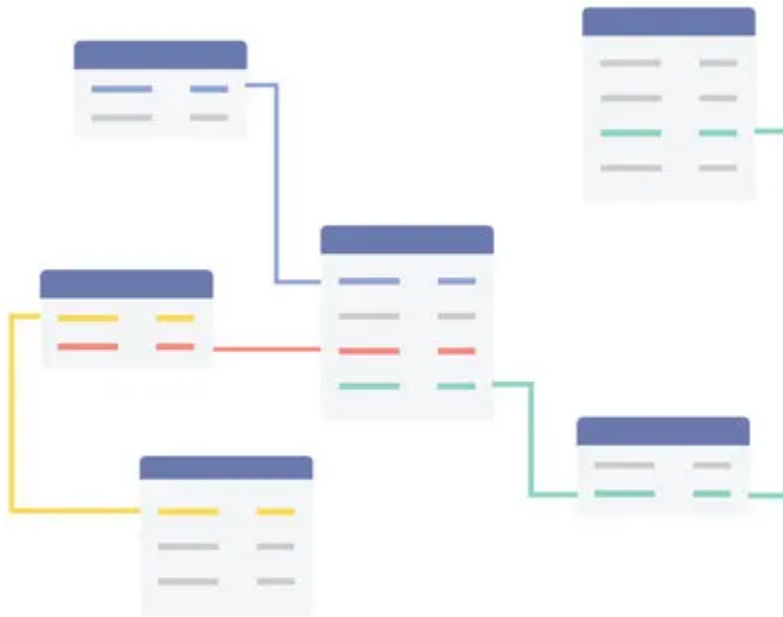# CSE311.10 PROJECT REPORT

A web-based Organic Farm Products Sales Management System for an SME that produces or sources their items locally

Md. Ashik

223 2660 042

ECE Dept.

1. Introduction

1.1 Project Overview

A web-based Organic Farm Products Sales Management System for an SME that produces or sources their items locally, featuring customer browsing/ordering and manager inventory/order management interfaces.

1.2 Technologies Used

- **Frontend:** HTML5, CSS3, JavaScript ES6
- **Backend:** PHP 8+
- **Database:** MySQL
- **Environment:** XAMPP

2. System Design

2.1 Architecture Overview

Three-tier architecture:

Presentation Layer (HTML/CSS/JS) → Application Layer (PHP) → Data Layer (MySQL)

2.2 Database Design

Entity Relationship Diagram

Products (1) ←→ (Many) Order_Items (Many) ←→ (1) Orders

Table Structures

Products Table:

```sql
CREATE TABLE products (
  product_id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(50) NOT NULL,
  category VARCHAR(30) NOT NULL,
  price_per_unit INT NOT NULL,
  stock_quantity INT NOT NULL
);
```

Orders Table:

```sql
CREATE TABLE orders (
  order_id INT AUTO_INCREMENT PRIMARY KEY,
  customer_name VARCHAR(50) NOT NULL,
  customer_phone VARCHAR(20),
  customer_address TEXT,
  customer_email VARCHAR(50),
  total_amount INT NOT NULL,
  status VARCHAR(20) DEFAULT 'Pending',
  order_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Order Items Table:

```sql
CREATE TABLE order_items (
  item_id INT AUTO_INCREMENT PRIMARY KEY,
  order_id INT,
  product_id INT,
  quantity INT NOT NULL,
  price INT NOT NULL,
  FOREIGN KEY (order_id) REFERENCES orders(order_id) ON DELETE CASCADE,
  FOREIGN KEY (product_id) REFERENCES products(product_id) ON DELETE CASCADE
);
```

2.3 Data Flow

Frontend ↔ AJAX ↔ PHP Backend ↔ MySQL Database

3. Implementation Details

3.1 Key Features

- **Customer Module:** Product browsing, cart management, order placement
- **Manager Module:** Product CRUD operations, order status management
- **Visual Design:** Emoji-based product identification, glassmorphism UI

3.2 Technical Implementation

Frontend Components

- Tab-based navigation between customer/manager views
- Responsive product grid with category organization

- Shopping cart with real-time total calculation
- Custom notification system

Backend Architecture

Single db.php file handling all operations:
- Action routing via URL parameters
- Prepared statements for security
- JSON responses for frontend communication
- Transaction-safe order processing

Core Functions

1. get_products: Category-sorted product retrieval
2. place_order: Multi-table transaction with stock updates
3. update_product: Product information modification
4. delete_product: Dependency-checked removal

4. Technical Challenges & Solutions

4.1 Product Visualization

Challenge: No image management system
Solution: Category-based emoji mapping (🥛 Dairy, 🥚 Eggs, etc.)

4.2 Data Integrity

Challenge: Preventing overselling and orphaned records
Solution: Real-time stock validation and foreign key constraints

5. Testing & Results

5.1 Testing Approach

Manual functional testing of all CRUD operations and user workflows

5.2 Key Test Cases

- Product browsing and cart operations
- Order placement with validation
- Manager product management
- Order status updates

5.3 Results

- Successful implementation of all core features
- Responsive interface across device sizes
- Consistent performance under normal usage

6. Conclusion

6.1 Achievements

- Complete CRUD web application using core technologies
- Modern UI with glassmorphism design
- Efficient database relationships and transactions

6.2 Limitations

- Single-user system (no authentication)
- No advanced reporting features
- Limited to core functionality

Screenshots

Customer View, the default view when the page is loaded:
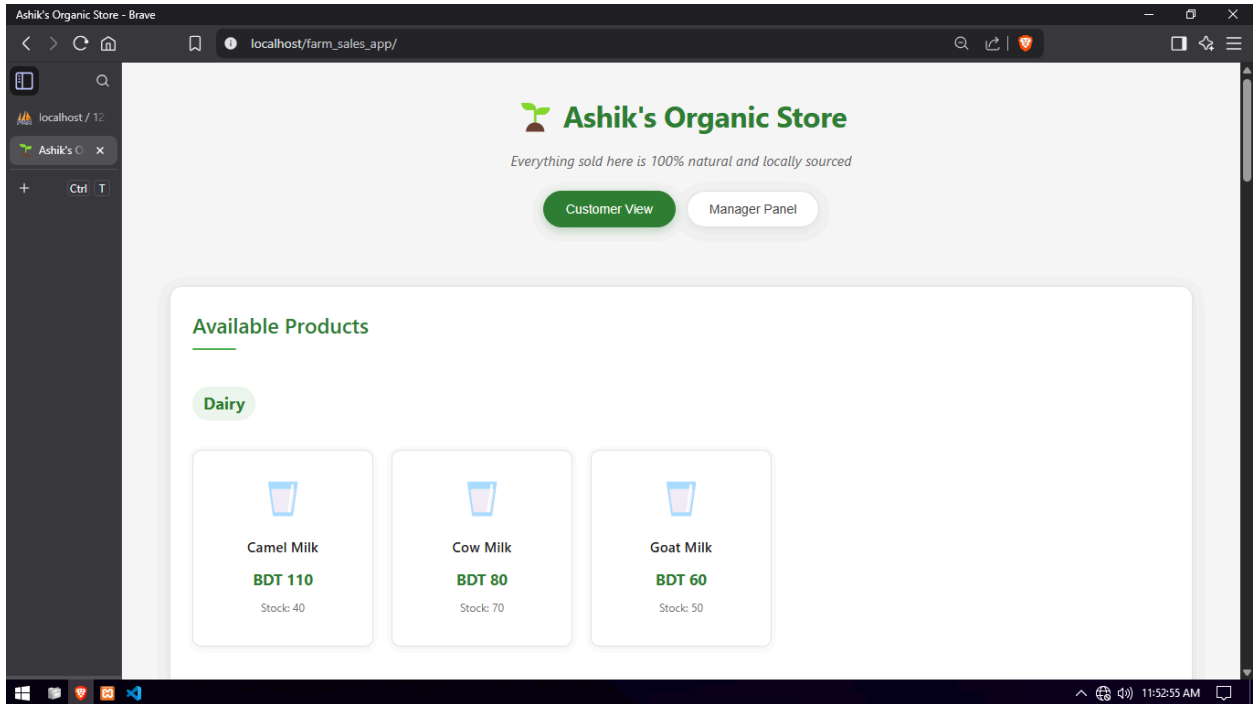
Different Products to choose from:
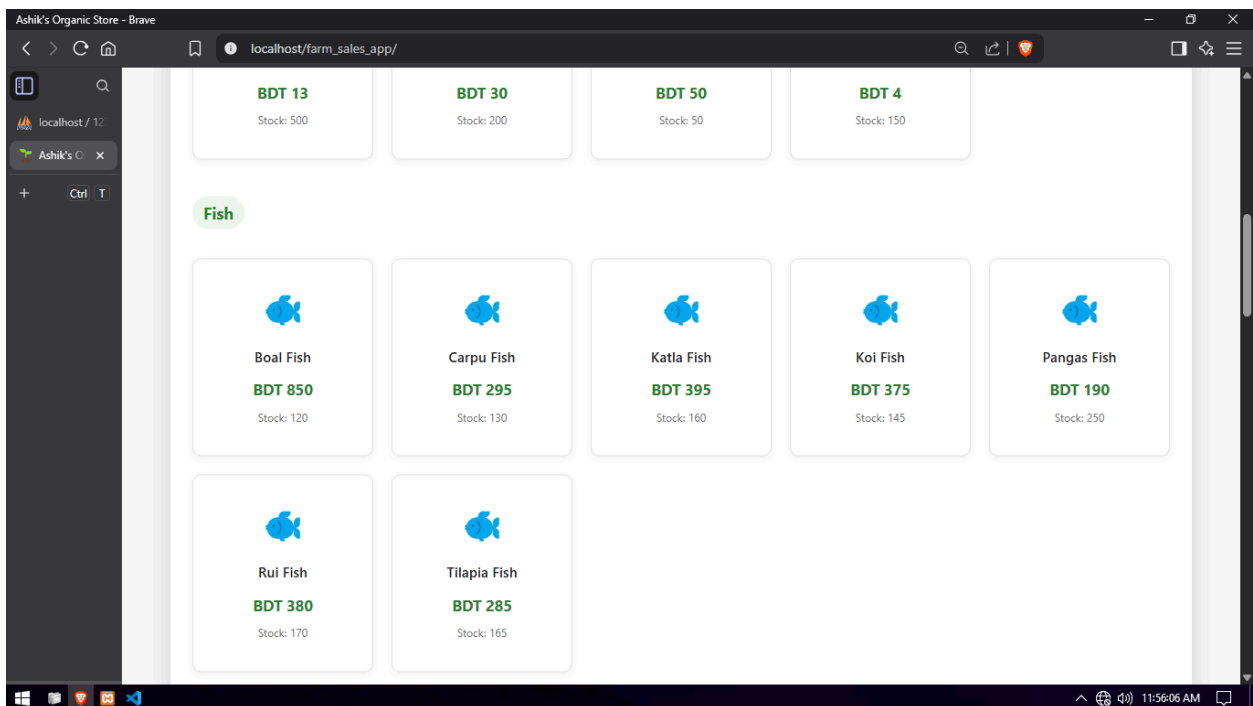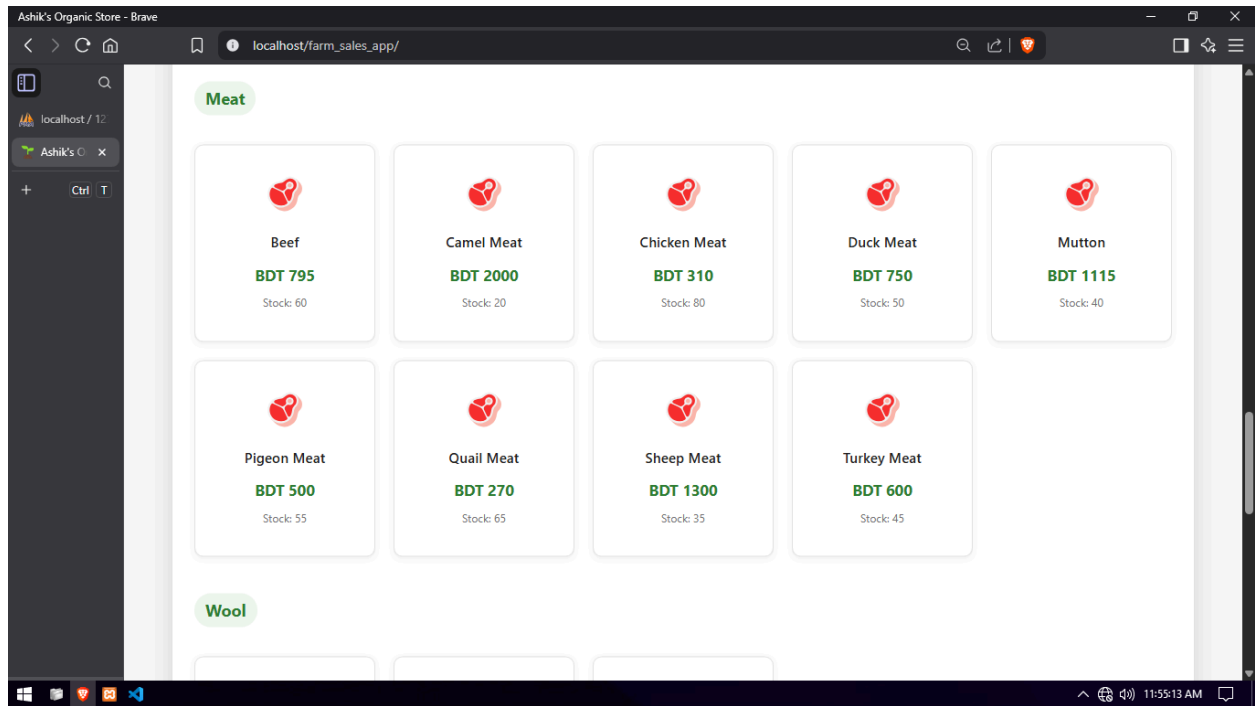
Order placement section:

Orders being placed:

The manager's panel allows the manager to add new items:

New product being added:
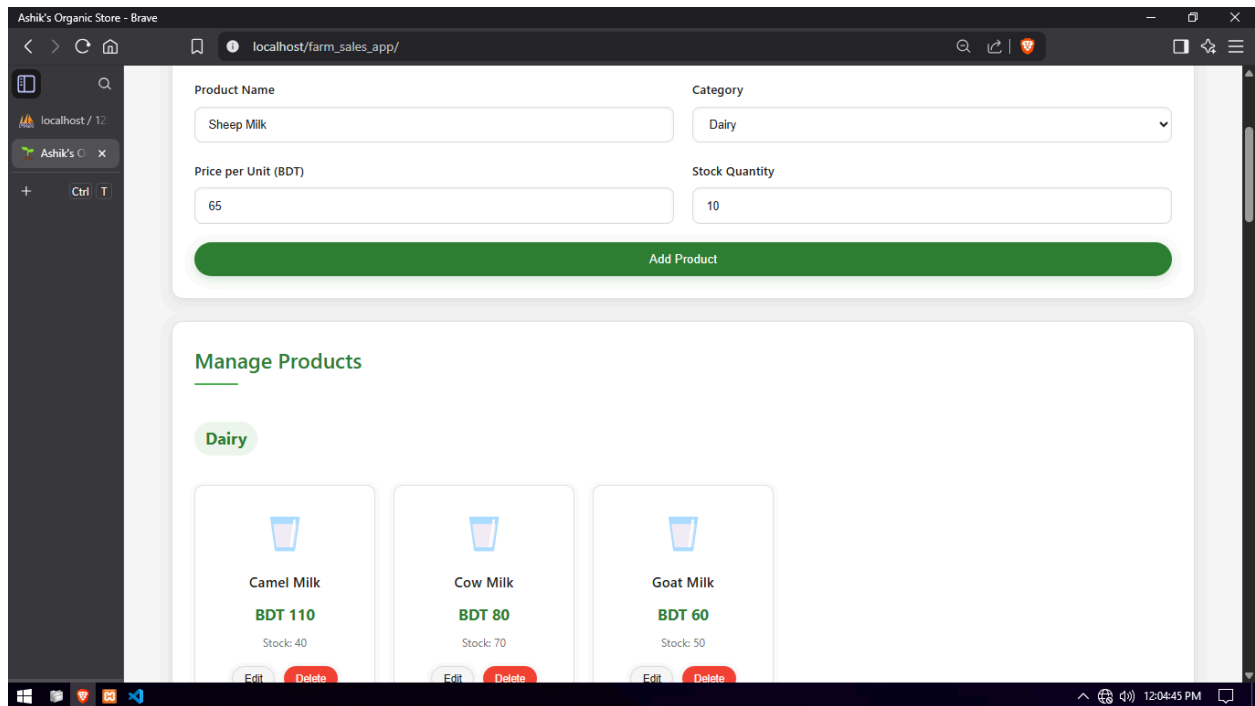
Added:
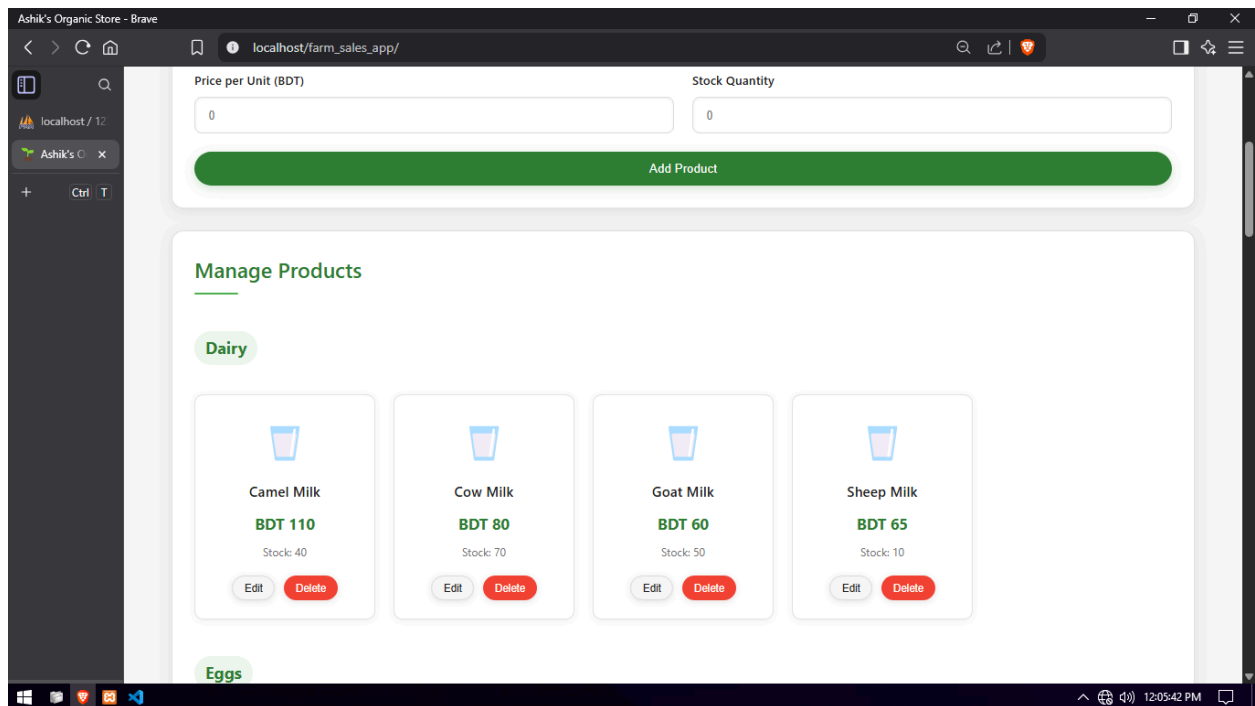
Product modifying feature:
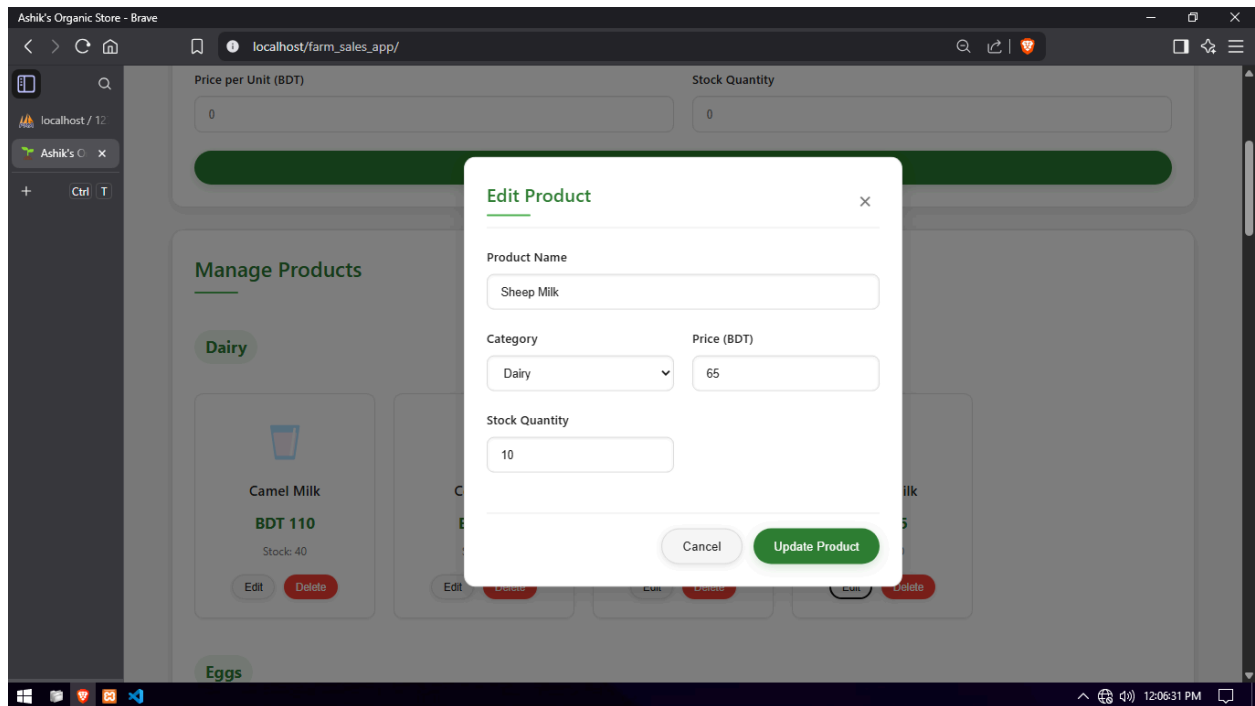
Modified:
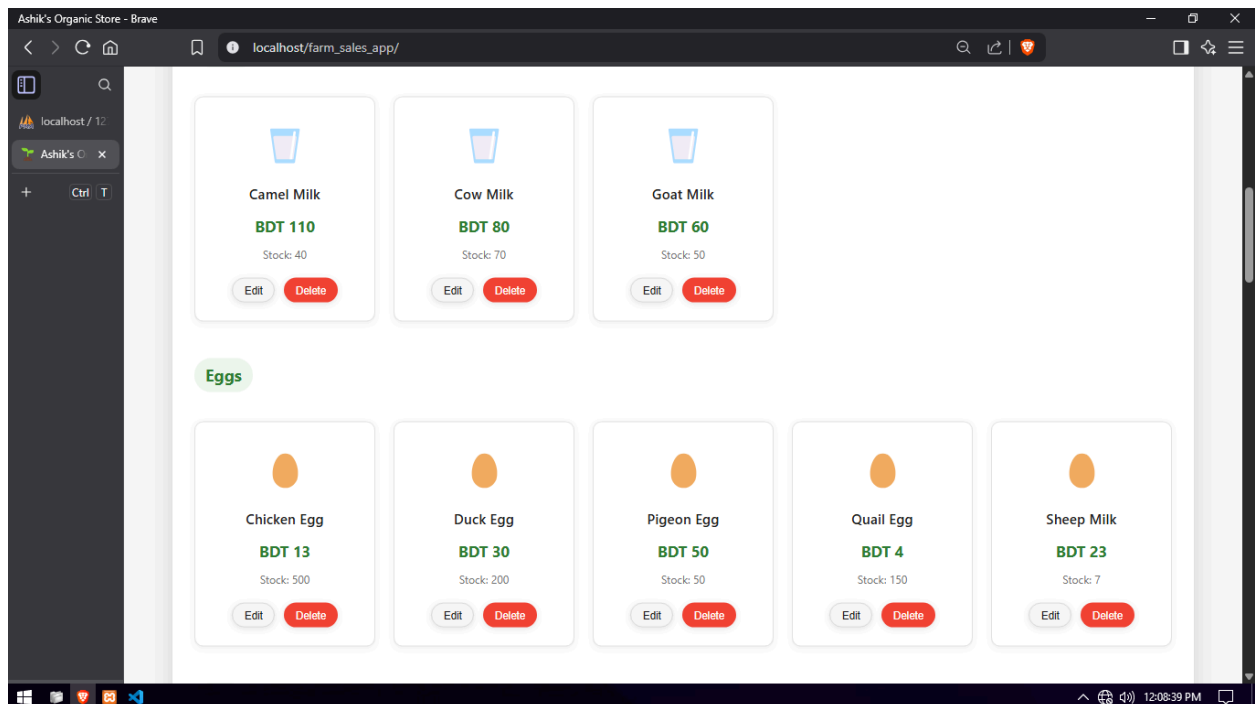
Deleting a product:
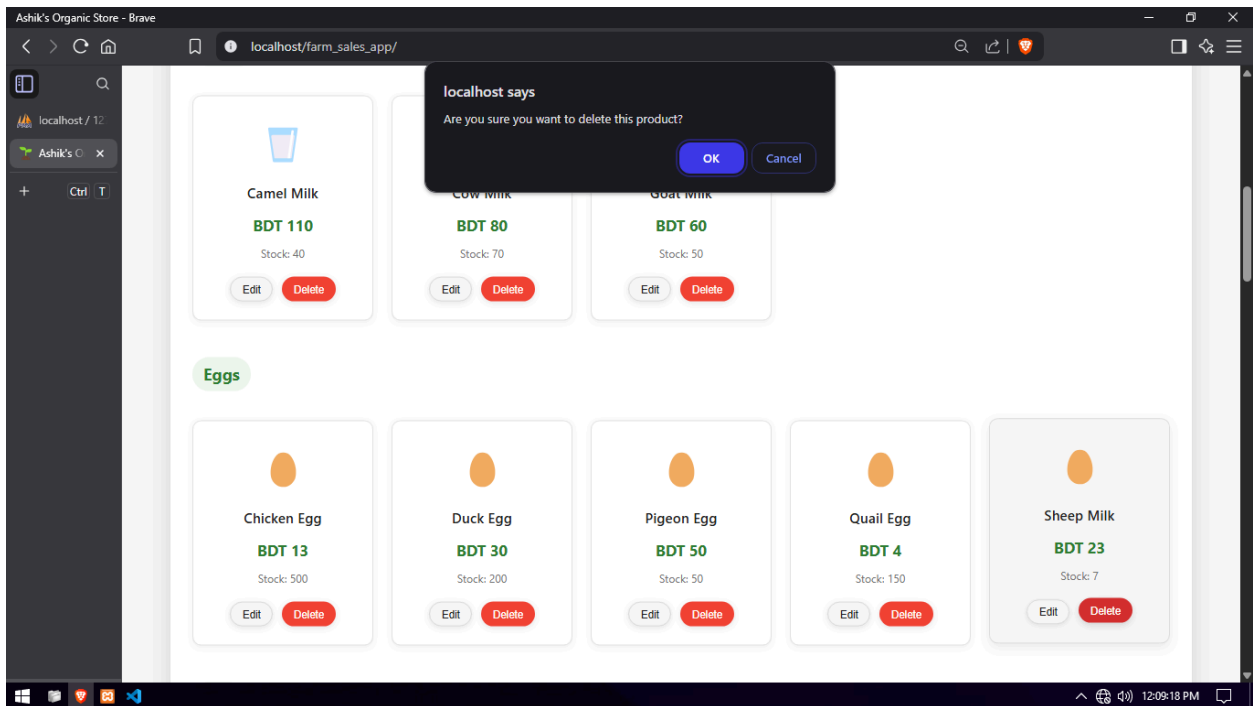
Deleted:
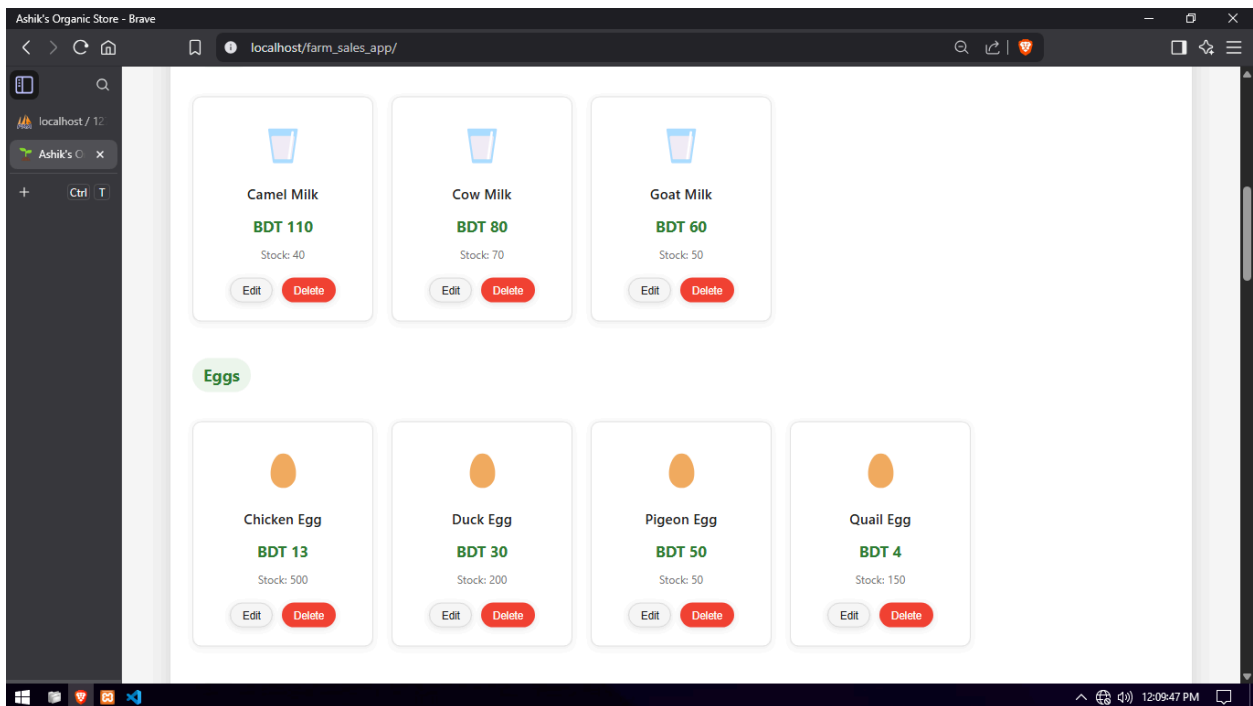
Delivery tracking segment:
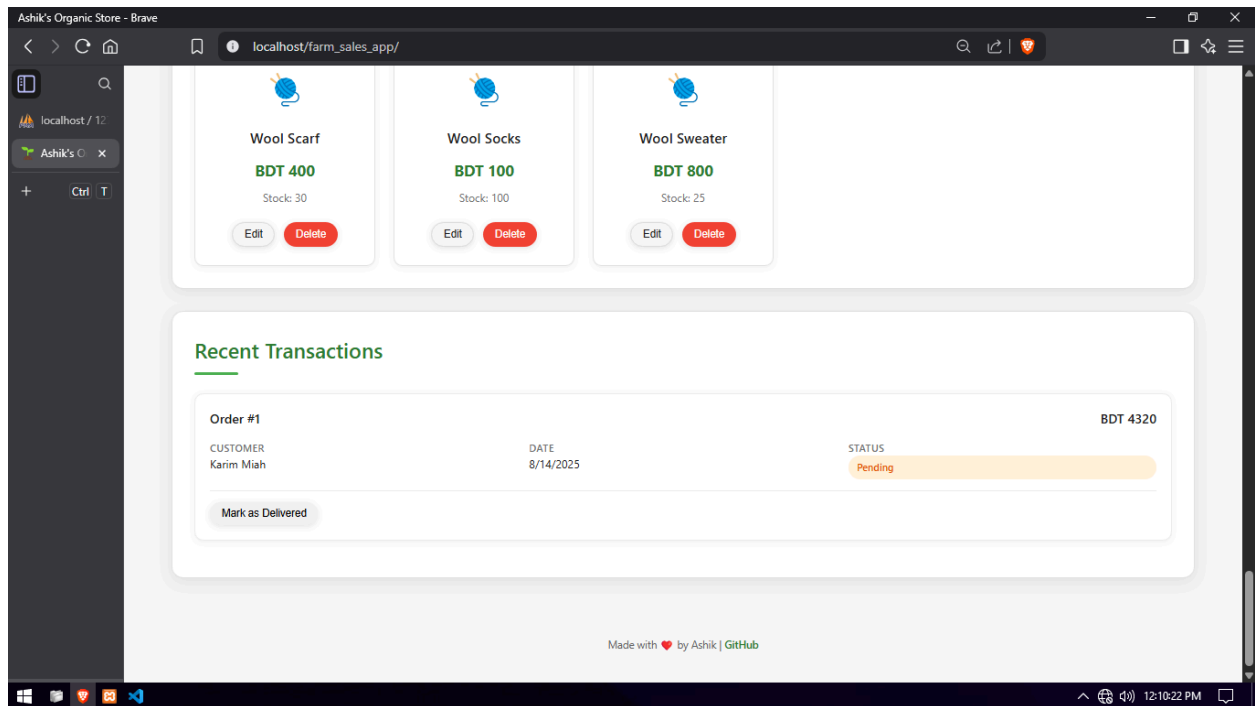
Order delivered:
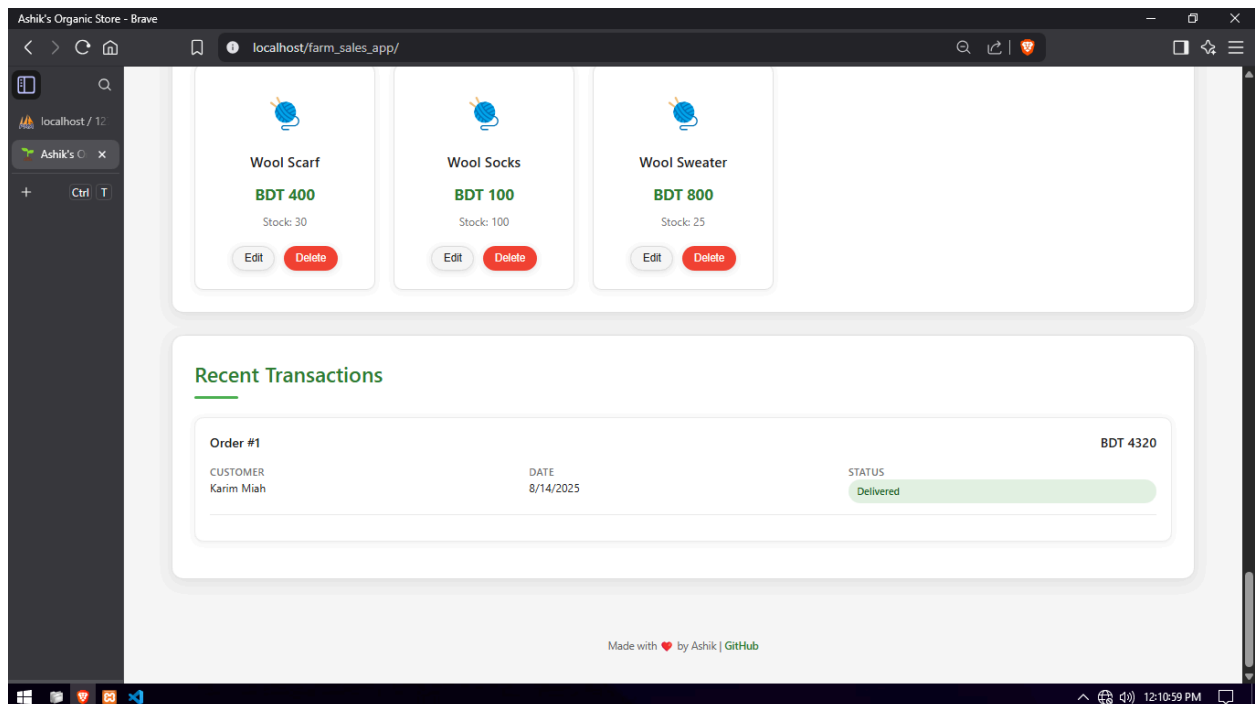
What the database looks like;

Orders table:

Order_items table:

Products table: