Requirements Document for Agricultural Products Selling Store

Project Title: **SPRING FOODS**

Technology Stack: Java (Backend), Angular (Frontend), MySQL (Database)

---

1. Project Overview

SPRING FOOD is an online platform that connects agricultural merchants with consumers. Merchants can list and manage their products, while users can browse, search, and place orders for the products.

---

2. Objectives

- Provide a seamless experience for merchants to manage their products.
- Allow users to search and order agricultural products.
- Ensure efficient communication between frontend and backend services.
- Store and manage all data in a secure MySQL database.

---

3. System Users

The system has two types of users:

1. **Merchant**
    - Manages products (add, update, delete, view).
    - Views and manages orders placed by users.
    - Updates order statuses (pending, shipped, completed).
2. **User (Customer)**
    - Browses and searches for products.
    - Places orders for selected products.
    - Views the status of their placed orders.

---

4. Functional Requirements

4.1 User Registration and Authentication

- **Merchant:**
  - Registration with required information (name, email, password, contact number, etc.).
  - Login and logout functionalities.
  - Password reset functionality.
- **User:**
  - Registration with necessary details (name, email, password, contact information).
  - Login and logout functionalities.
  - Password reset functionality.

4.2 Merchant Dashboard

- **Product Management:**
  - **Add Products**: Merchants can add new products with details such as name, description, price, stock, and category (e.g., grains, fruits, vegetables, etc.).
  - **Update Products**: Merchants can edit product details.
  - **Delete Products**: Merchants can remove products from the store.
  - **View Products**: Merchants can view all products they have listed.
- **Order Management:**
  - View all orders placed by users.
  - Update the order status (pending, shipped, completed).
  - View order details (e.g., product details, quantity, customer name, shipping information).

4.3 User Dashboard

- **Browse Products**: Users can browse available products by category or search for specific products.
- **View Product Details**: Users can click on a product to view its details (name, description, price, availability).
- **Place Orders**: Users can add products to their cart and place an order.
- **Order Management**: Users can view their order history and check the status of their orders.

4.4 Shopping Cart Functionality

- Users can add products to their shopping cart.
- Users can view and edit their cart before placing the order.
- Users can remove products from the cart.
- Cart total calculation based on the selected products.

4.5 Order Confirmation and Notifications

- Upon placing an order, users receive a confirmation notification.
- Merchants receive notifications for new orders.
- Users receive updates on order status changes (e.g., shipped, completed)..

4.6 Payment Integration

- Placeholder for future payment integration (e.g., mobile money, credit card).
- Orders will initially be marked as cash-on-delivery until payment integration is implemented.

---

5. Non-Functional Requirements

5.1 Performance

- The system should handle concurrent users efficiently.
- Response times for user actions should be under 5 seconds.

5.2 Security

- User data (passwords, sensitive details) should be securely stored using encryption.
- Authentication should be handled securely using tokens (e.g., JWT).
- Prevent unauthorized access to merchant functions.

5.3 Scalability

- The system should be scalable to accommodate increasing numbers of merchants and users.

5.4 Usability

- The platform should be user-friendly, with an intuitive UI for both merchants and customers.

5.5 Database Integrity

- Ensure data consistency and integrity in MySQL.
- Implement backup and recovery mechanisms.

---

6. Frontend Requirements (Angular)

6.1 Merchant Interface

- Dashboard for managing products and viewing orders.
- Forms for adding, updating, and deleting products.
- Order list and status update functionality.

6.2 User Interface

- Product catalog with search and filter capabilities.
- Shopping cart and checkout process.
- Order history and status tracking.

---

7. Deployment and Testing

- The system will be deployed on a web server and should be accessible via browsers.
- Unit tests, integration tests, and end-to-end tests should be written to ensure the reliability of the system.

---

8. Milestones

- Week 1: Requirement gathering and database design.
- Week 2: Setup backend with Java and MySQL, and implement basic authentication.
- Week 3: Implement product management for merchants.
- Week 4: Implement order management for users.
- Week 5: Integrate frontend with Angular and finalize user interfaces.
- Week 6: Testing, bug fixes, and final deployment.

---

9. Conclusion

This document outlines the structure and functionality required for building an agricultural product selling store. The development team should ensure all functional and non-functional requirements are met to deliver a seamless experience for both merchants and customers.