Project Documentation: Agricart

➢ Table of Contents

1. Introduction

Agricart is a platform designed to allow customers to buy fresh, seasonal fruits and vegetables directly from farmers. The goal is to provide a convenient, healthy, and cost-effective solution while helping farmers get better prices for their produce. The platform avoids preservatives to maintain the taste and health value of the products.

2. Project Overview

Agricart is a web-based application developed to facilitate direct transactions between farmers and consumers. Users can register, log in, browse through various seasonal produce, and place orders. Farmers benefit from fair pricing, and customers receive fresh produce.

Key Features:

- User Registration and Login: Users can create an account and log in to make purchases.

- Product Listings: Display of seasonal fruits and vegetables with detailed information.
- Order Management: Users can place, view, and track their orders.
- Admin Management: Admins can manage users and products.

3. Features

- User Registration: Customers can sign up by entering their details (name, email, phone, etc.).
- User Login: Registered users can log into the system to place orders and view order history.
- Admin Panel: Admins can add, edit, and delete users, as well as manage the products available on the platform.
- Product Search: Users can browse available seasonal products and place orders.
- Order History: Users can track their previous orders and view their status.

4. Technologies Used

✓ **Frontend:**
- HTML, CSS, JavaScript
- Bootstrap (for responsive UI)

✓ **Backend:**
- Java (JSP)
- Spring Boot
- JDBC for database interaction

✓ **Database:**
- MySQL

✓ **Testing:**
- JUnit 5 (for unit testing)

✓ **Other:**
- Maven (for project management)

5. System Design

Architecture Overview:

Agricart follows a traditional 3-layer architecture: Presentation Layer, Business Logic Layer, and Data Access Layer.

- Frontend (Presentation Layer): Implements the user interface using JSP, HTML, CSS, and JavaScript.
- Backend (Business Logic Layer): Spring Boot is used to handle the backend services, processing requests from the front end and interacting with the database.
- Database (Data Access Layer): MySQL is used to store the user and product data. It handles user registration, login, and order management.

User Flow:

- User registers an account via the registration form.
- After registration, the user can log in and browse available products.
- Once products are selected, users can place an order.
- Admins manage users and products via an admin panel.

6. Database Design

The database consists of the following tables:

Tables:

➢ Users

- user_id (Primary Key)
- name
- email
- password
- phone_no
- address
- created_at
- updated_at

➢ Products

- product_id (Primary Key)
- name

- description
- price
- quantity
- created_at
- updated_at

➢ Orders

- order_id (Primary Key)
- user_id (Foreign Key)
- product_id (Foreign Key)
- quantity
- order_date
- status

7. Installation and Setup

Pre-requisites:

- Java Development Kit (JDK) 8 or above
- MySQL Database installed and running
- Maven for project management

Installation:

1. Clone the repository:

   git clone https://github.com/Prashanteeeee/agricart.git

2. Set up the MySQL Database:
- Create a database named Agricart and configure it in the UserDAO.java class with your database credentials.
3. Install project dependencies:
- mvn install
4. Run the project:
- Run the application using your IDE or by executing the following Maven command:

  mvn spring-boot:run

5. Access the Application:

- Open a browser and go to http://localhost:8080 to access the platform.

8. Usage

User Registration:

- Navigate to the registration page, fill in the required fields, and click "Register".

Login:

- After registration, log in using your credentials to access the product catalog and place orders.

Admin Panel:

- Admin users can log in and manage users and products from the admin dashboard.

9. Testing

The project includes unit tests using JUnit 5 to ensure the correctness of methods in the UserDAO class. You can run these tests using Maven:

mvn test

Test cases include validation for user creation, user retrieval, product management, and order handling.

10. Conclusion

Agricart provides a seamless and efficient way for customers to purchase fresh, seasonal produce directly from farmers, while helping farmers earn fair prices for their produce. With features like user registration, product search, and order management, the platform aims to revolutionize the way consumers access healthy food.