Krishipath - Online Farmers Platform

Table of Contents

1. Introduction

Krishipath is a web-based application designed to connect farmers, buyers, and suppliers directly.
The platform ensures fair trade practices, better pricing for farmers, and high-quality agricultural products for consumers.
It aims to revolutionize the agricultural ecosystem with technology-driven solutions.

2. Project Overview

Krishipath enables users to register, browse products, and place orders while offering farmers an intuitive way to list and sell their goods.
Administrators can manage the platform efficiently to ensure smooth operations.

Key Features:

- Direct Farmer-to-Buyer Interaction
- Real-Time Product Listings
- Order and Inventory Management
- Secure Payments
- Logistics and Tracking

3. Features

- **User Registration:** Create accounts with user-friendly forms.
- **Login System:** Secure authentication for farmers, buyers, and admins.
- **Product Listings:** Detailed catalog of agricultural products.
- **Order Management:** Place, view, and track orders easily.
- **Admin Dashboard:** Manage users, products, and transactions.

4. Technologies Used

- **Frontend:** HTML, CSS, JavaScript, Bootstrap
- **Backend:** JSP, Servlets
- **Build Tool:** Maven
- **Server:** Apache Tomcat
- **Database:** MySQL
- **Testing:** JUnit 5

5. System Design

Architecture Overview:

Krishipath follows a 3-layer architecture:

1. **Presentation Layer:** JSP, HTML, CSS, and JavaScript.

2. **Business Logic Layer:** Java Servlets for processing requests.

3. **Data Access Layer:** MySQL for data storage and retrieval.

User Flow:

- Users register an account and log in.

- Buyers browse products and place orders.

- Farmers list products with pricing and inventory details.

- Admins manage platform operations.

6. Database Design

The database comprises the following tables:

Users Table:

- `user_id` (Primary Key)

- `name`, `email`, `password`, `phone_no`, `address`

Products Table:

- `product_id` (Primary Key)

- `name`, `description`, `price`, `quantity`

Orders Table:

- `order_id` (Primary Key)

- `user_id` (Foreign Key), `product_id` (Foreign Key)

- `quantity`, `order_date`, `status`

7. Installation and Setup

Prerequisites:

- Java JDK 8+
- MySQL Server
- Maven
- Apache Tomcat

Steps:

1. **Clone the Repository:**

   ```bash
   git clone https://github.com/yourusername/krishipath.git
   cd krishipath
   ```

2. **Database Setup:**

   - Create a database named `krishipath_db`.
   - Import the SQL scripts from `/sql` folder.

3. **Configure Database:**

   Update database credentials in `db-config.properties`:

   ```properties
   db.url=jdbc:mysql://localhost:3306/krishipath_db
   db.user=root
   db.password=yourpassword
   ```

4. **Build the Project:**

   ```bash

   mvn clean install

   ```

5. **Deploy to Tomcat:**

   Copy the `.war` file from `target/` to the `webapps/` folder of Apache Tomcat.

6. **Run the Application:**

   Access the platform at `http://localhost:8080/krishipath`.

8. Usage

- **User Registration:** Register with personal details to access the platform.

- **Browse Products:** Search for agricultural goods and place orders.

- **Admin Panel:** Admins manage platform data and operations.

9. Testing

Unit testing is performed using JUnit 5 to validate functionality for:

- User registration and login.

- Product management.

- Order processing.

Run tests with Maven:

```bash

mvn test

```

10. Conclusion

Krishipath creates a seamless connection between farmers and buyers, ensuring fair trade, quality products, and a streamlined agricultural marketplace.

The platform's user-centric design and robust functionality aim to empower farmers and benefit consumers.