INSTITUTE FOR ADVANCED COMPUTING AND SOFTWARE DEVELOPMENT, AKURDI, PUNE

"Fields and Farmers"

PG-DAC August 2024

Submitted By:

Group No: 22

Roll No.	**Name of Student**
248035	Ishika Sahu
248051	Mohite Janhavi Jaydeep

Mrs. Geeta Darunte
Project Guide

Mr. Rohit Puranik
Centre Coordinator

ABSTRACT

The Business to Consumer Model has advanced significantly since its inception. Despite its expansion into a variety of product categories, the fresh foods sector is yet unexplored. Given the current organic food trends and the growing health consciousness of the consumer generation, fresh foods have the potential to become the next big thing in e-commerce.

The goal of this project is to create an online store for the sale of fresh food products. It offers a list of farmers who sell fresh produce as well as a page with the things that each farmer sells. Additionally, it offers a cart to make it easier to recall the options the user has chosen. In order to return to the farmer from whom they bought the most recent batch of goods, the user can also see their order history.

Java and React were the two primary technologies utilized in this project. The backend was written in Java. React is utilized for client-side page rendering, which relieves the client of the burden of rendering views and offers a seamless single-page experience. A database called MySQL has been utilized to store user, farmer, and product lists.

The multilayer architecture used in the design and implementation of this project aims for maximum cohesiveness and minimal coupling.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to **IACSD Pune** for providing us with the opportunity to work on this project, *Fields and Farmers*. This project has been an incredible learning experience, allowing us to apply our knowledge and skills in a practical setting.

We extend our heartfelt thanks to **Mrs. Kishori Khadilkar** for her invaluable guidance and encouragement throughout the project. Her insights and expertise were instrumental in shaping our work.

A special note of thanks to **Mrs. Madhura Anturkar**, our faculty for J2SE and J2EE, whose mentorship and continuous encouragement helped us overcome various technical challenges. Her guidance was invaluable in refining our approach and enhancing the quality of our work.

We would also like to extend our deepest gratitude to our Center Coordinator, **Mr. Rohit Puranik** and Project Guide, **Mrs. Geeta Darunte**, for her continuous support, constructive feedback, and expert guidance throughout the development of *Fields and Farmers*. Her valuable suggestions and mentorship helped us refine our project and address critical aspects with clarity and efficiency.

Ishika Sahu (248035)
Mohite Janhavi Jaydeep (248051)

Table of Contents

1. INTRODUCTION

Fresh produce industries across the world are facing a roller-coaster ride of new developments and trends. Although there might be a few tight turns and steep slopes, the latest trends paint an inspirational picture of what lies ahead in the next five to 10 years.

In the fresh produce sector, technology and retail innovations abound. From futuristic hi-tech grocery stores, the rise of e-commerce opportunities, culinary innovation centers and revolutionary robotics technology to vertical farming and plant-based food innovations like cauliflower pizza and vegetable steaks.

Online Shopping of Fresh Food opens up a new world of options. Users won't have to go from store to store to hunt for fresh food. They won't have to worry about wondering whether their food is organic or inorganic. They will be able to refill their fridges in just one click, all while sitting at home.

Our system offers one stop solution to all fresh food needs. Users can log into their accounts and then they will be taken to produces offered by the farmer.

Customer can pick what foods they want to order and add to the cart. Once they are done selecting what they require, after reviewing cart summary they can simply click on check out button to pay bill and they will get an order details pdf on their registered email for the same. Their cart will be delivered to their houses.

This can be done from any place, at any time all from the internet, thus making it easy to get your daily need of fresh foods.

1. Admin Role

The Admin plays a crucial role in managing and overseeing the entire platform to ensure smooth operations. Their responsibilities span across user management, inventory control, order processing, payments, system configuration, reporting, and security.

Responsibilities:

- **User Management:**

  Manage customer and farmer accounts (registration, approval, and deactivation).

  Verify and authenticate farmers selling fresh produce.

  Handle user queries, complaints, and account-related issues.

- **Inventory Management:**

  Oversee the availability of fresh produce listed by farmers.

  Ensure product quality compliance with platform standards.

  Update and remove products that are out of stock or unavailable.

- **Order Management:**

  Monitor customer orders and ensure smooth processing.

  Coordinate with farmers for order fulfillment and delivery.

  Track order statuses and resolve delivery-related issues.

- **Payment Management:**

  Handle online payment processing and transaction verification.

  Manage refunds, cancellations, and disputes related to payments.

  Ensure secure financial transactions between customers and farmers.

- **System Configuration:**

  Manage website settings, including UI/UX updates and feature enhancements.

  Configure tax rates, pricing rules, and discount policies.

  Set up notification and email configurations for order updates.

- **Reporting & Analytics:**

  Generate reports on sales, revenue, and customer trends.

  Track farmer performance and product demand analytics.

  Monitor website traffic, user engagement, and financial transactions.

- **Security Management:**

  Enforce data privacy and platform security measures.

  Implement authentication and role-based access control.

  Monitor for fraud detection and unauthorized access.

Access Level:

- Full access to all platform features, including user, inventory, order, and payment management.
- Authority to add, update, or remove products, users, and configurations.
- View and manage financial reports, analytics, and security logs.
- Control over system settings, including promotional offers, discounts, and policies.

2. User Role

The User Role in the system is divided into two main categories:

1. Employees (Farmers & Inventory Managers) – Manage product listings and stock levels.
2. Customers – Browse, order, and pay for fresh produce.

Responsibilities:

- **Inventory Access (For Employees - Farmers & Inventory Managers):**

  Add, update, and manage fresh produce listings.

  Maintain accurate stock levels and update availability in real-time.

  Ensure product quality and compliance with platform standards.

- **Order Placement (for customers):**

  Browse the available fresh produce.

  Select and add items to the cart.

  Proceed to checkout and complete the order.

  Track order status and delivery updates.

- **Profile Management:**

  Update personal details such as name, contact information, and address.

  View order history and saved payment methods.

- **Payments:**

  Make secure online payments using multiple payment

  options. View payment receipts and transaction details.

  Manage saved payment methods for faster checkout.

- **Support Requests:**

  Raise support tickets for order issues, product inquiries, or account-related problems.

  Contact customer support for assistance with transactions and deliveries.

  Provide feedback or complaints regarding the system or products.

Access Level:

- Limited access based on assigned permissions.

  Can view and interact with specific features (e.g., order placement, product browsing) but cannot alter system settings.

No access to sensitive system configurations, financial data, or user management functionalities.

Role-Based Access Control (RBAC)

- The system follows **RBAC**, where users can only access the functionalities assigned to their roles:
- **Customers**: Can browse, purchase, and track orders but cannot access inventory management.
- **Employees (Farmers & Inventory Managers)**: Can manage product listings and stock but cannot place customer orders.
- **Admins**: Have full access to all system functionalities.

Security Considerations

To ensure data integrity and secure access, the system implements **Spring Security** with the following features:

- **Authentication:** Users must log in with valid credentials before accessing the system.
- **Authorization:** Role-based permissions ensure users access only the resources allowed by their role.
- **Session Management:** Secure session handling prevents unauthorized access and session hijacking.

The Role-Based Structure ensures tasks are appropriately delegated while maintaining security.

- Admins handle strategic management and decision-making.
- Employees manage inventory and stock.
- Customers interact with the system for purchases and support.

This ensures a secure, efficient, and user-friendly platform for fresh produce shopping.

1.1 Purpose

As the name implies, The Fields and Farmers is about farmers and the goods they display. Its goal is to eliminate the middleman by bringing farmers and consumers together directly. This guarantees that consumers will receive fresh food at a very low cost. Additionally, it guarantees that every farmer has an equal opportunity to attract clients, so eliminating the need for intermediaries.

1.2 Scope

"Fields and Farmers" aims to deliver a web-based application that hosts a wide collection of the food-items that users can browse through. Users can place orders and make payment. They can update their profile, add delivery address. They can view their order history as well.

Admins can manage various product details like stock, price, adding new products, and categories etc. Only admin can add farmers. Admins can even delete users and/or farmers, if the need arises.

The actual payment logic and food item logistics and delivery are not supported by this project. We are assuming that the organization that implements it will be using third-party payment API which can easily be integrated in our application if needed. Fields and Farmers is only an interface for both customers (for browsing and shopping for food items) and admins (for managing products, farmers, users listing).

1.3 Objective of Inventory

The primary objective of this project is to develop an online fresh produce marketplace that connects farmers directly with consumers, providing a seamless and efficient platform for purchasing fresh food. The system aims to:

1. **Provide a One-Stop Solution** – Offer a convenient platform where users can browse, select, and order fresh produce directly from farmers.

2. **Enhance Accessibility** – Enable customers to purchase fresh food from any location and at any time via an online platform.

3. **Promote Transparency** – Ensure that customers have clear information about the source, quality, and type (organic/inorganic) of their produce.

4. **Streamline Ordering & Delivery** – Facilitate easy selection, checkout, payment, and home delivery of fresh food items.

5. **Support Farmers** – Provide farmers with a direct-to-consumer sales channel, eliminating middlemen and increasing their profits.

6. **Encourage Technological Adoption** – Leverage digital solutions such as e-commerce, automated order processing, and electronic invoicing for an improved shopping experience.

7. **Improve Customer Convenience** – Save customers time by eliminating the need to visit multiple stores while ensuring the availability of high-quality fresh food.

This project aims to revolutionize the fresh food supply chain by integrating technology, e-commerce, and direct farmer-to-consumer interactions for a more sustainable and efficient food shopping experience.

1.4 Functionalities Provided by Inventory Sell-Mart

1. User Management

- User Registration and Login:
- User can register, login, and update Profile Management: Customers can update their profile and delivery details. their profile.
- Role-Based Access Control:
  Admins manage farmers, products, and orders.

2. Inventory Management

- Product Catalog Management:
  Admins can add, update, and delete product categories and individual products.
- Stock Level Monitoring:
  Admins can manage product stock and update inventory.
- Inventory Adjustment:
  Admins can modify product details, including stock updates.

- Automated Reordering:

  Not explicitly mentioned, but it can be implemented in the future.

3. Sales Management

- Order Processing:

  Customers can add products to the cart, place orders, and view order history.

- Payment Integration:

  Payment handling is assumed to be done via third-party integration.

- Order Tracking:

  Customers can view their order history.

- Invoicing and Billing:

  Order details are sent via email.

4. Customer Management

- Customer Profiles:

  Customers can register, update profiles, and manage addresses.

- Loyalty Programs and Discounts:

  Future scope includes discounts based on purchase history.

- Customer Support:

  Not explicitly mentioned, but feedback features could be implemented.

- Feedback and Reviews:

  Future scope includes product and farmer ratings.

5. Reporting and Analytics

- Sales Reporting:

  Admins can view order details for all users.

- Inventory Reporting:

  Admins manage product stock and availability.

- Customer Analytics:

  Purchase history is available for customers.

- Financial Reporting:

  Not explicitly mentioned but could be extended.

6. Security and Compliance

- Data Encryption:

  Not explicitly mentioned, but the system uses Spring Boot and Hibernate for security.

- Audit Trails:

  Admins can manage users and track orders.

- Compliance with Legal Requirements:

  Not explicitly mentioned but could be adapted.

7. Customer Experience Enhancement

- Search and Filtering:

  Customers can browse and sort products.

- Wishlist and Shopping Cart:

  A cart feature is implemented.

- Responsive Design:

  The UI is built using React for a seamless user experience.

8. Integration and Extensibility

- API Integration:

  The project is built on Java Spring Boot, which supports API integration.

- Third-Party Integrations:

  Payment processing can be integrated.

2. SOFTWARE REQUIREMENT SPECIFICATION

2.1 Functional Requirements for Fields and Farmers

1. User Management

- **User Registration:**

  Users (farmers, customers, and admins) can register by providing necessary details such as name, email, phone number, and password.

- **User Authentication:**

  Secure login functionality using email/password authentication with optional multi-factor authentication.

- **Role-Based Access Control:**

  Different user roles (Admin, Farmer, Customer) with specific permissions to access and manage system functionalities.

- **Profile Management:**

  Users can update their profile information, including contact details, delivery addresses, and preferences.

2. Inventory Management

- **Product Management:**

  Farmers and admins can add, update, and delete products with descriptions, images, categories, and pricing.

- **Stock Level Monitoring:**

  The system tracks available inventory in real-time and updates stock status dynamically.

- **Inventory Adjustment:**

  Manual stock updates for returns, damages, or corrections.

- **Reordering:**

  Automatic alerts for low stock levels, enabling timely restocking of products.

3. Sales Management

- **Order Placement:**

  Customers can browse products, add them to the cart, and place orders.

- **Order Processing:**

  Orders are processed based on product availability, payment confirmation, and delivery details.

- **Payment Processing:**

  Secure integration with third-party payment gateways for online transactions.

- **Order Tracking:**

  Customers can track their orders with real-time updates on shipment and delivery status.

- **Invoicing:**

  Automatically generated invoices are sent via email or downloadable from the user's account.

4. Customer Management

- **Customer Profiles:**

  Customers can manage their personal details, shipping addresses, and payment preferences.

- **Order History:**

  Customers can view their past purchases, invoices, and transaction details.

- **Loyalty Programs:**

  Discounts, reward points, and membership benefits for frequent buyers.

- **Feedback and Reviews:**

  Customers can leave reviews and ratings for purchased products and farmers.

5. Reporting and Analytics

- **Sales Reports:**

  Detailed sales reports for admins and farmers, including revenue, order volume, and trends.

- **Inventory Reports:**

  Stock status, product performance, and restocking alerts.

- **Customer Analytics:**

  Insights into customer purchasing behavior, preferences, and engagement levels.

6. Security

- **Data Encryption:**

  Secure storage and transmission of sensitive data using encryption techniques.

- **Authentication and Authorization:**

  Role-based authentication with secure login mechanisms and user authorization protocols.

2.2 Non-Functional Requirements for Fields and Farmers

1. Performance

- Response Time:

  The system should respond to user actions (e.g., browsing products, placing orders) within 2-3 seconds under normal load.

- Scalability:

  The system should support increased traffic without performance degradation by allowing horizontal and vertical scaling.

  It should be capable of handling thousands of concurrent users without failure.

- Throughput:

  The system should efficiently process at least 1000 transactions per second under peak conditions.

2. Reliability

- Availability:

  The system should have an uptime of at least 99.9%, ensuring continuous access.

- Fault Tolerance:

  The system should be able to recover from failures automatically without data loss.

  It should implement redundancy in key components (e.g., databases, application servers).

- Error Handling:

  The system should provide clear error messages and logs to assist users and developers in debugging issues.

3. Usability

- User Interface:

  The system should have an intuitive and responsive UI, accessible on both desktop and mobile devices.

  It should support multiple languages if required.

  A consistent theme and well-known UI components should be used for a seamless user experience.

4. Maintainability

- Modularity:

  The system should be designed using a modular architecture, allowing independent updates and improvements.

- Code Quality:

  The codebase should follow industry best practices, ensuring readability, reusability, and maintainability.

  Version control (e.g., Git) should be used to track changes and manage releases.

- Testing:

  The system should undergo unit testing, integration testing, and load testing to ensure stability.

  Automated testing should be implemented to identify issues early in the development cycle.

- **Other Requirements:**

Hardware Requirements

- **Server Requirements:**

  Processor: Intel Xeon or AMD Ryzen (Quad-core or higher)

  RAM: Minimum 8GB (16GB recommended for scalability)

  Storage: Minimum 100GB SSD (Expandable as needed)

  Network: High-speed internet connection with static IP

- **Client Requirements:**

  Processor: Intel Core i5 or AMD Ryzen

  5

  RAM: 4GB or higher

  Storage: Minimum 10GB free space

  Internet: Stable broadband connection

Software Requirements

- **Operating System:** Windows 10/11, macOS, or Linux
- **Backend Framework:** Spring Boot
- **Frontend Framework:** React.js
- **Database:** MySQL
- **Development Tools:** VS Code, STS 4, Postman, Git

3. DIAGRAMS

3.1 Entity Relationship Diagram:

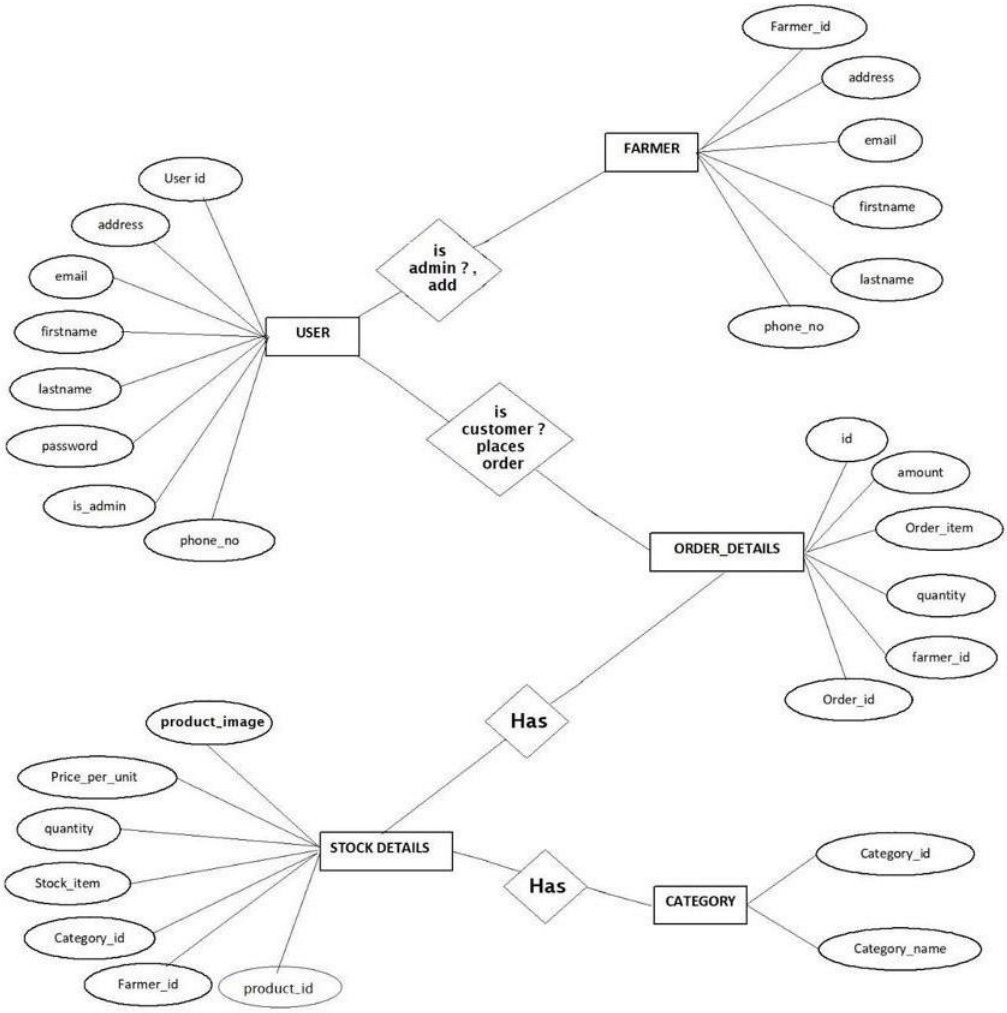


Fig. ER Diagram for Fields and Farmers

3.2 Use Case Diagram:

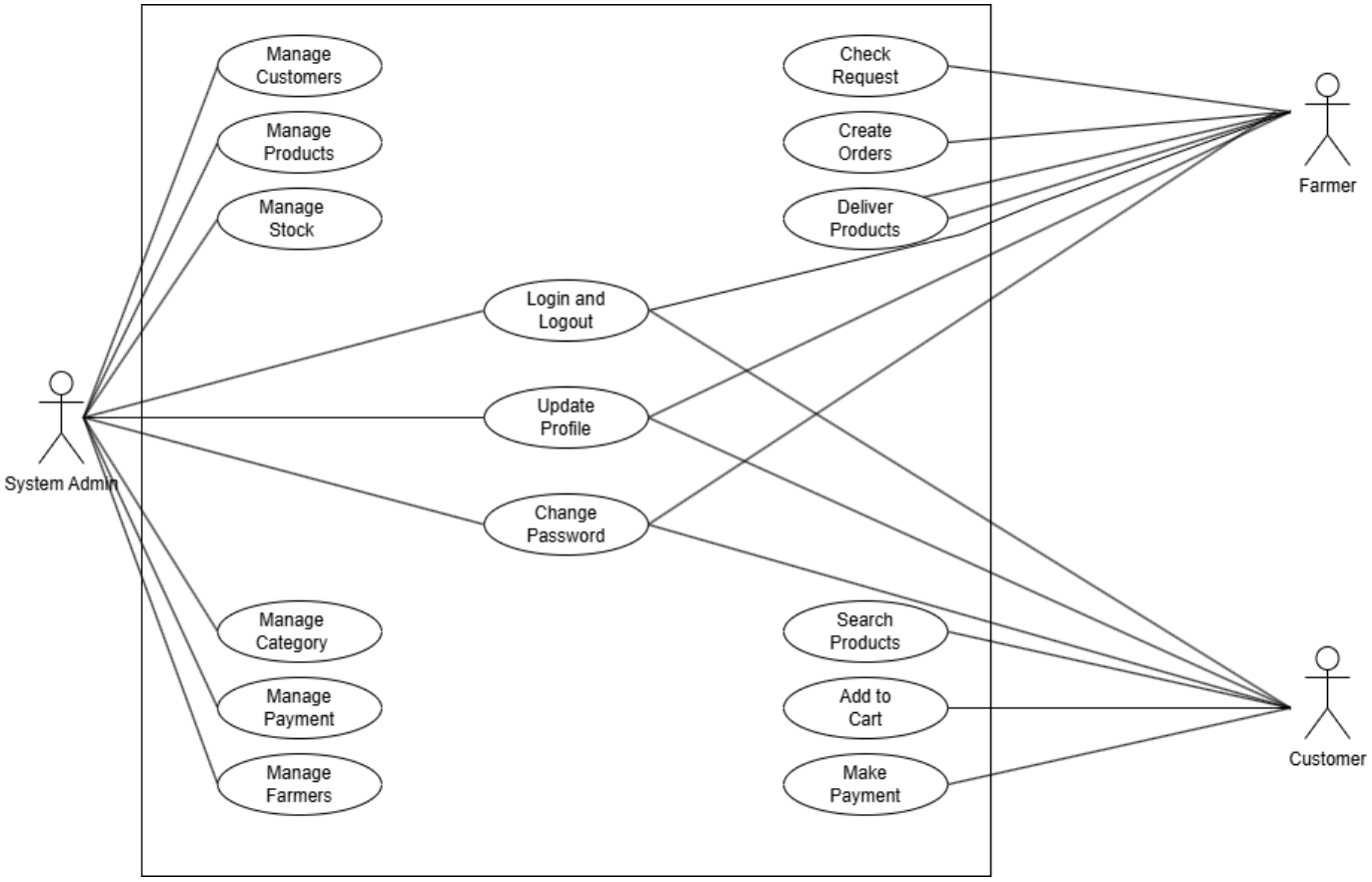


Fig. Use Case Diagram for Fields and Farmers

3.3 Data Flow Diagram:

DFD Level 0 :

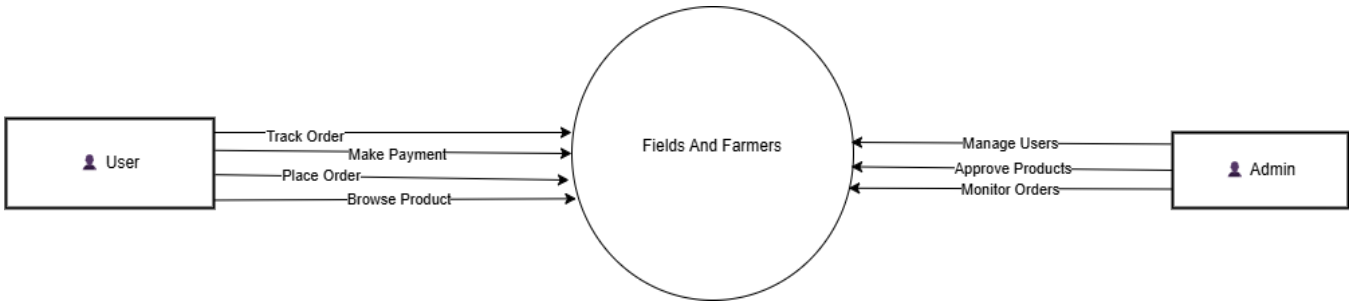
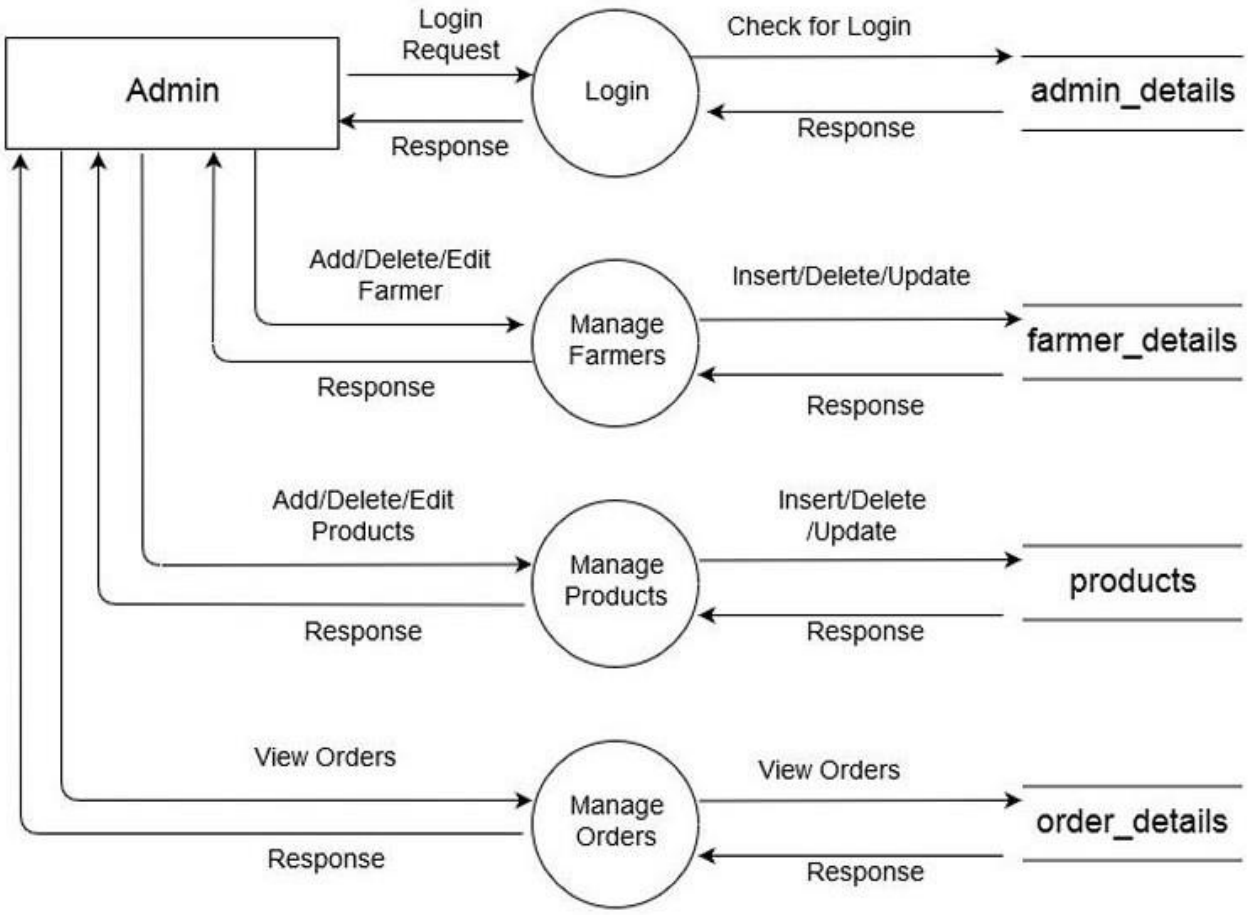


Fig. DFD Level 0

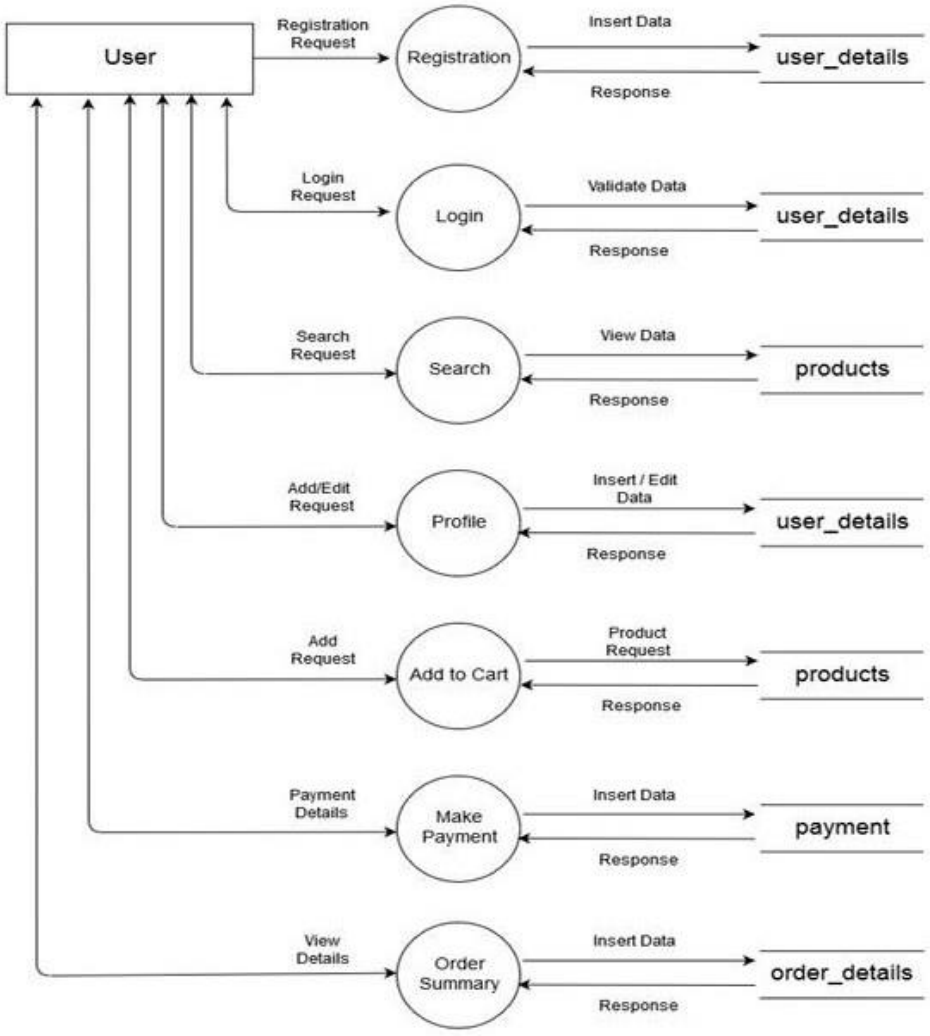
Level 1 : Admin Side DFD:



Admin Side DFD

Fig. Admin DFD Level for Fields and Farmers

Level 1 : User Side DFD:



User Side DFD

Fig. User DFD for Fields and Farmers

3.4 Activity Diagram :

User Activity Diagram:

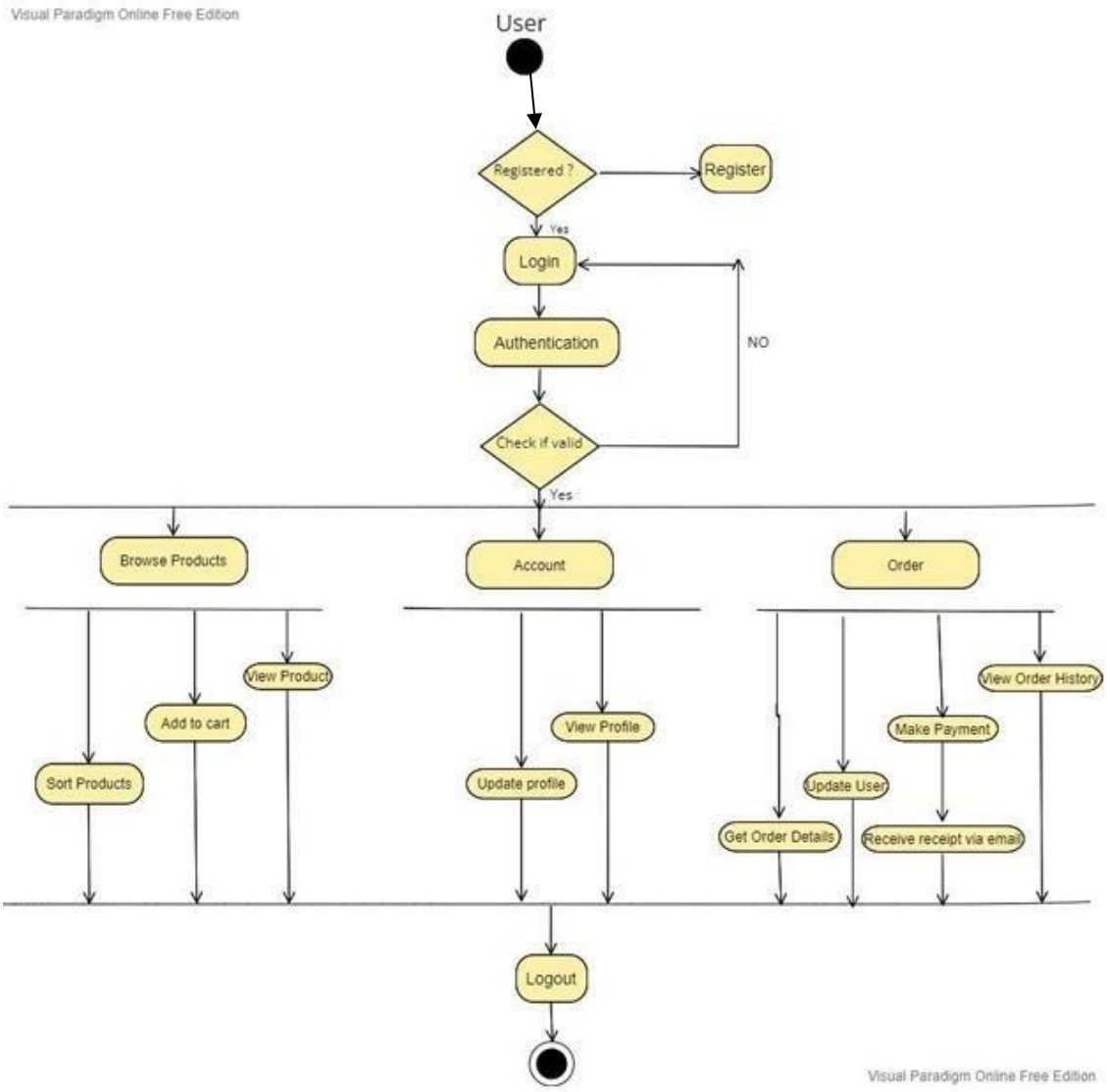


Fig. User Activity Diagram for Fields and Farmers

Admin Activity Diagram:

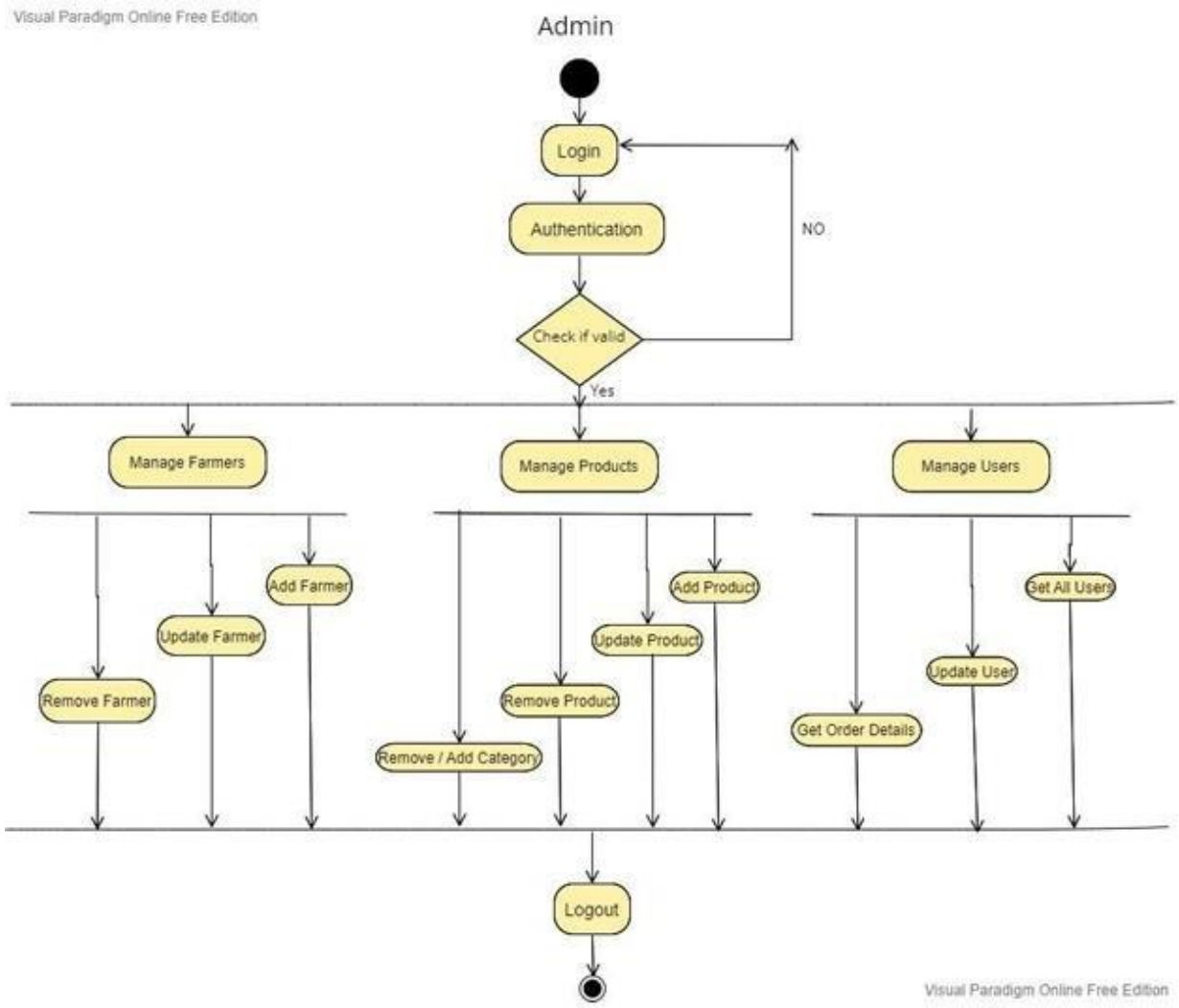


Fig. Admin Activity Diagram for Fields and Farmers

3.5 Class Diagram:

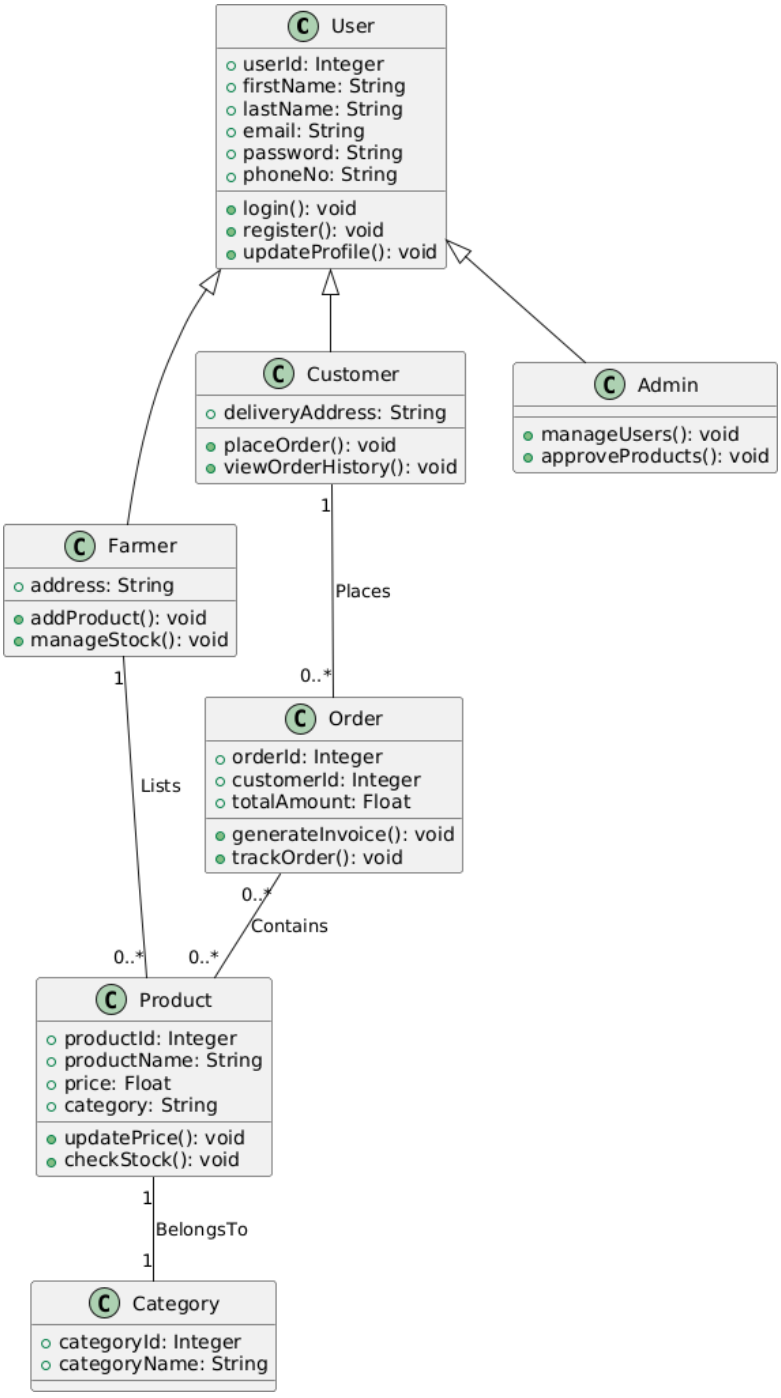
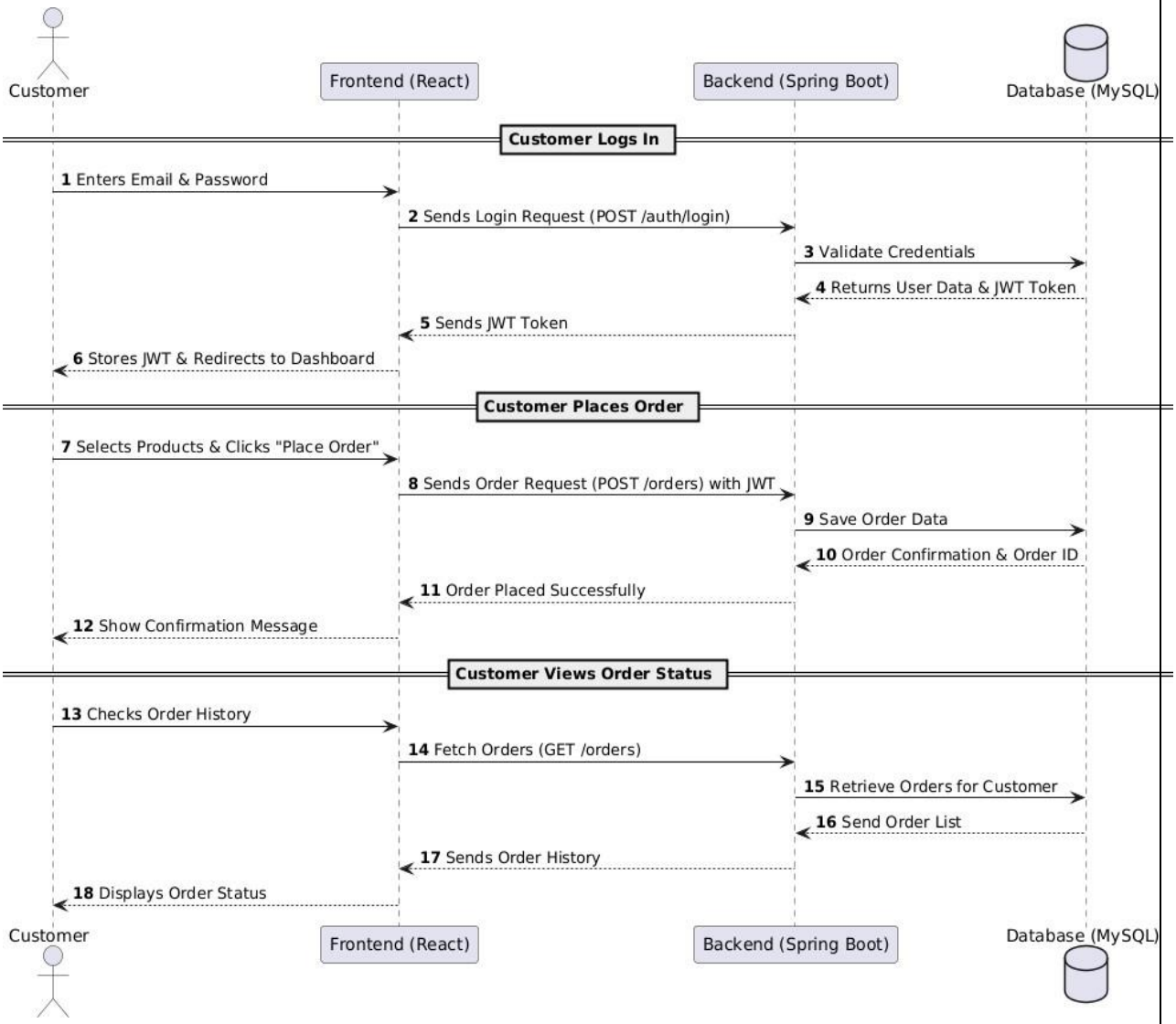


Fig. Class Diagram for Fields and Farmers

3.6 Sequence Diagram



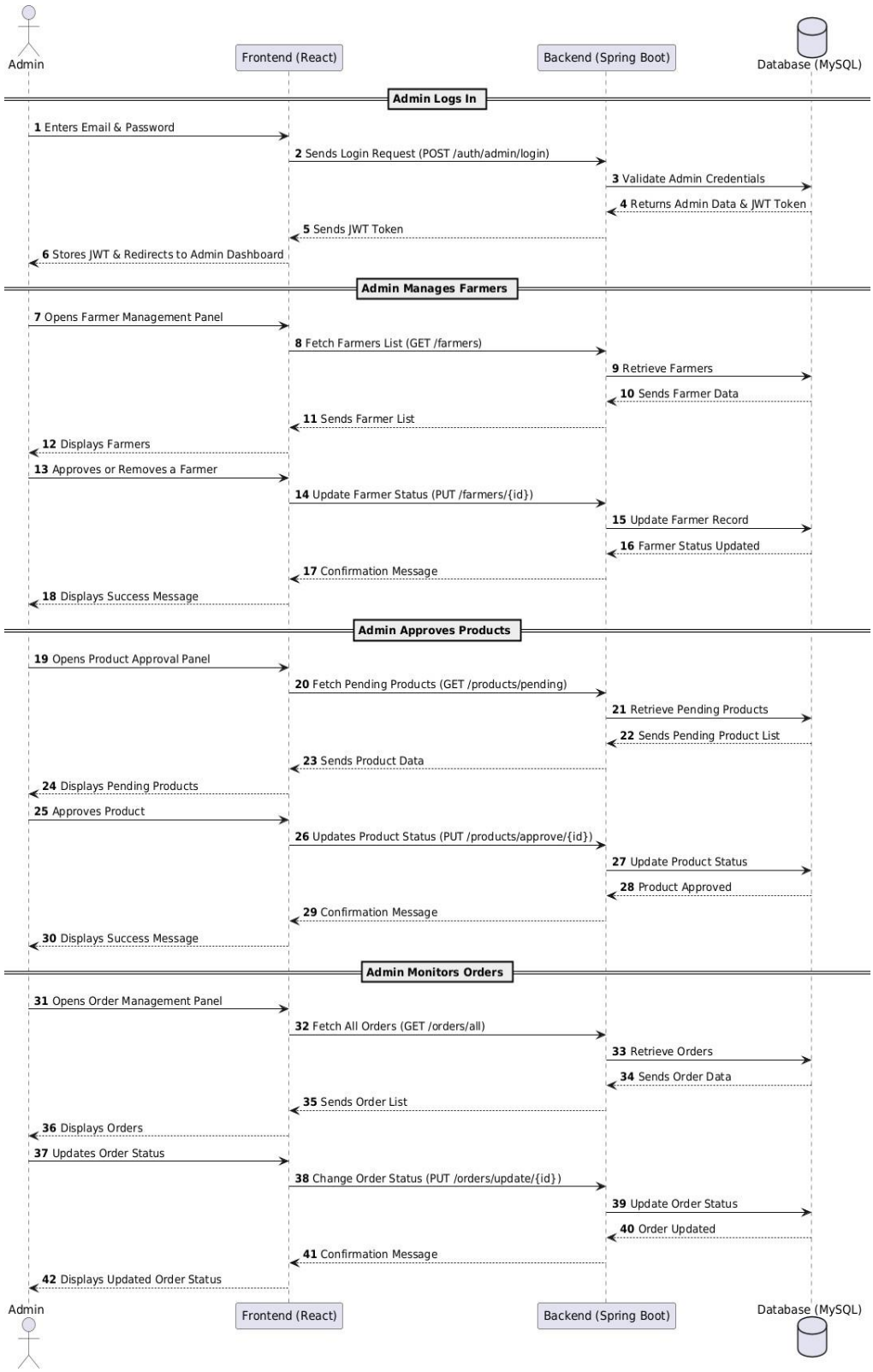


Fig. Sequence Diagram for Fields and Farmers

3.7 Tables:

The following table structures depict the database design.

a. Users Table

```
mysql> desc user;
+-----------+--------------+------+-----+---------+----------------+
| Field     | Type         | Null | Key | Default | Extra          |
+-----------+--------------+------+-----+---------+----------------+
| user_id   | int          | NO   | PRI | NULL    | auto_increment |
| address   | varchar(200) | YES  |     | NULL    |                |
| email     | varchar(50)  | YES  | UNI | NULL    |                |
| firstname | varchar(20)  | YES  |     | NULL    |                |
| is_admin  | bit(1)       | YES  |     | NULL    |                |
| lastname  | varchar(20)  | YES  |     | NULL    |                |
| password  | varchar(30)  | YES  |     | NULL    |                |
| phone_no  | varchar(15)  | YES  |     | NULL    |                |
+-----------+--------------+------+-----+---------+----------------+
8 rows in set (0.00 sec)
```

b. Farmers Table

```
mysql> desc farmer;
+-----------+--------------+------+-----+---------+----------------+
| Field     | Type         | Null | Key | Default | Extra          |
+-----------+--------------+------+-----+---------+----------------+
| farmer_id | int          | NO   | PRI | NULL    | auto_increment |
| address   | varchar(200) | YES  |     | NULL    |                |
| email     | varchar(50)  | YES  |     | NULL    |                |
| firstname | varchar(20)  | YES  |     | NULL    |                |
| lastname  | varchar(20)  | YES  |     | NULL    |                |
| phone_no  | varchar(15)  | YES  | UNI | NULL    |                |
+-----------+--------------+------+-----+---------+----------------+
6 rows in set (0.00 sec)
```

c. Stock Details Table

```
mysql> desc stock_details;
+----------------+--------------+------+-----+---------+----------------+
| Field          | Type         | Null | Key | Default | Extra          |
+----------------+--------------+------+-----+---------+----------------+
| product_id     | int          | NO   | PRI | NULL    | auto_increment |
| product_image  | varchar(400) | YES  |     | NULL    |                |
| price_per_unit | float        | YES  |     | NULL    |                |
| quantity       | int          | NO   |     | NULL    |                |
| stock_item     | varchar(50)  | NO   | UNI | NULL    |                |
| category_id    | int          | YES  | MUL | NULL    |                |
| farmer_id      | int          | YES  | MUL | NULL    |                |
+----------------+--------------+------+-----+---------+----------------+
7 rows in set (0.01 sec)
```

d. Orders Table

```
mysql> desc orders;
+------------------+---------+------+-----+---------+----------------+
| Field            | Type    | Null | Key | Default | Extra          |
+------------------+---------+------+-----+---------+----------------+
| order_id         | int     | NO   | PRI | NULL    | auto_increment |
| delivery_date    | date    | YES  |     | NULL    |                |
| delivery_status  | bit(1)  | YES  |     | NULL    |                |
| payment_status   | bit(1)  | YES  |     | NULL    |                |
| place_order_date | date    | YES  |     | NULL    |                |
| user_id          | int     | NO   | MUL | NULL    |                |
+------------------+---------+------+-----+---------+----------------+
6 rows in set (0.00 sec)
```

e. Order Details Table

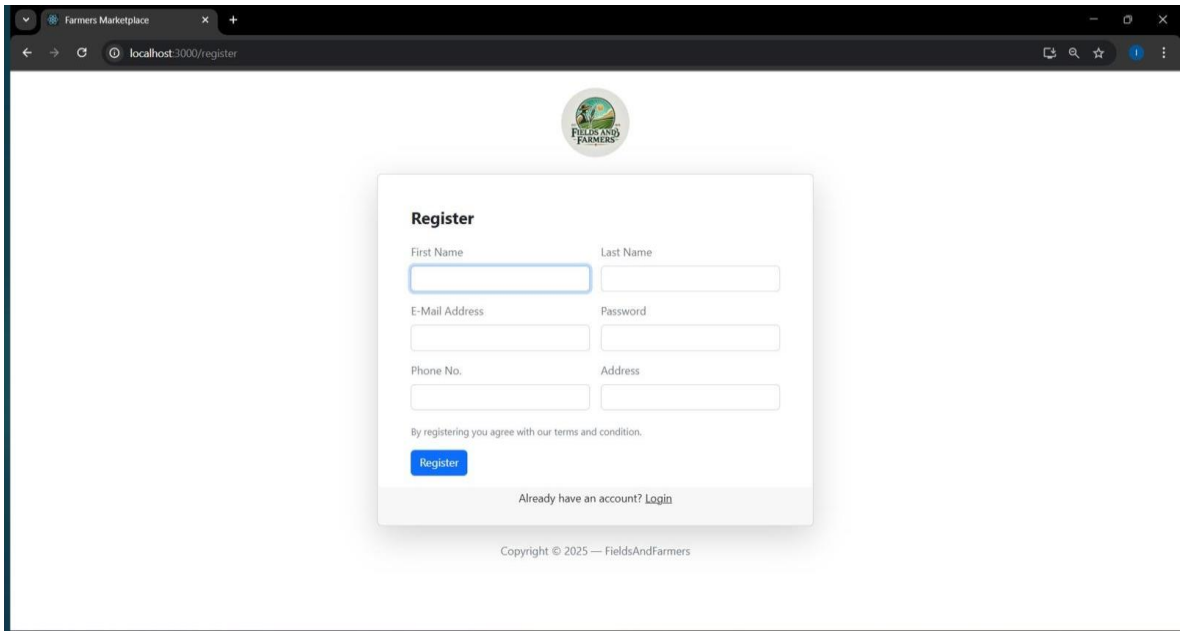
```
mysql> desc order_details;
+------------+-------------+------+-----+---------+----------------+
| Field      | Type        | Null | Key | Default | Extra          |
+------------+-------------+------+-----+---------+----------------+
| id         | int         | NO   | PRI | NULL    | auto_increment |
| amount     | double      | NO   |     | NULL    |                |
| order_item | varchar(20) | NO   |     | NULL    |                |
| quantity   | int         | NO   |     | NULL    |                |
| farmer_id  | int         | YES  | MUL | NULL    |                |
| order_id   | int         | NO   | MUL | NULL    |                |
+------------+-------------+------+-----+---------+----------------+
6 rows in set (0.00 sec)
```

f. Category Table

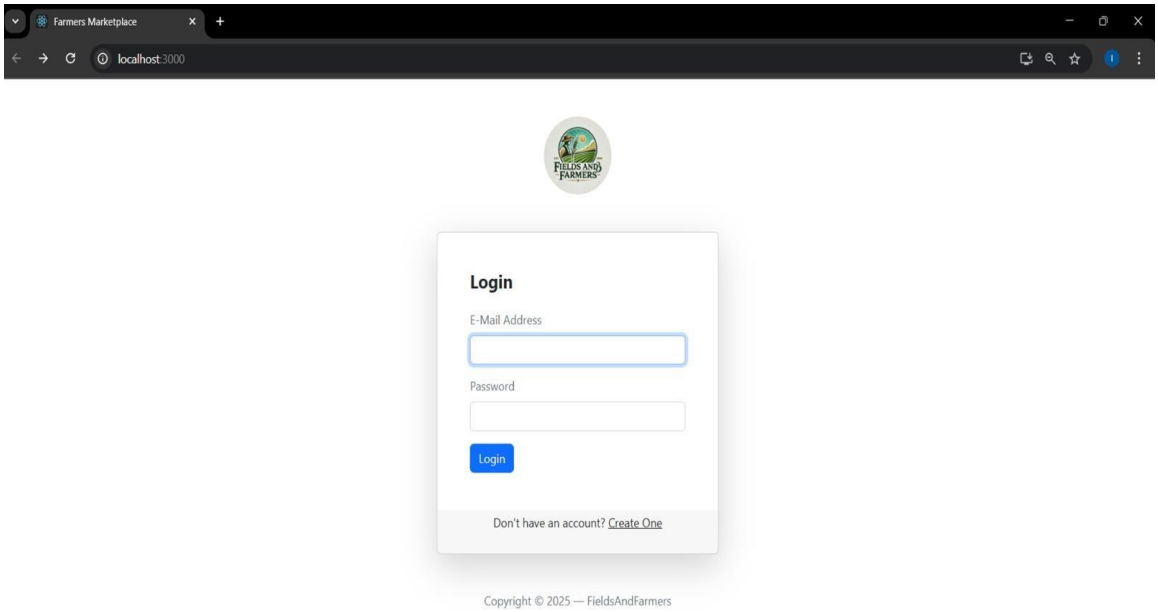
```
mysql> desc category;
+---------------+--------------+------+-----+---------+----------------+
| Field         | Type         | Null | Key | Default | Extra          |
+---------------+--------------+------+-----+---------+----------------+
| category_id   | int          | NO   | PRI | NULL    | auto_increment |
| category_name | varchar(255) | YES  | UNI | NULL    |                |
+---------------+--------------+------+-----+---------+----------------+
2 rows in set (0.00 sec)
```

4. SNAPSHOTS

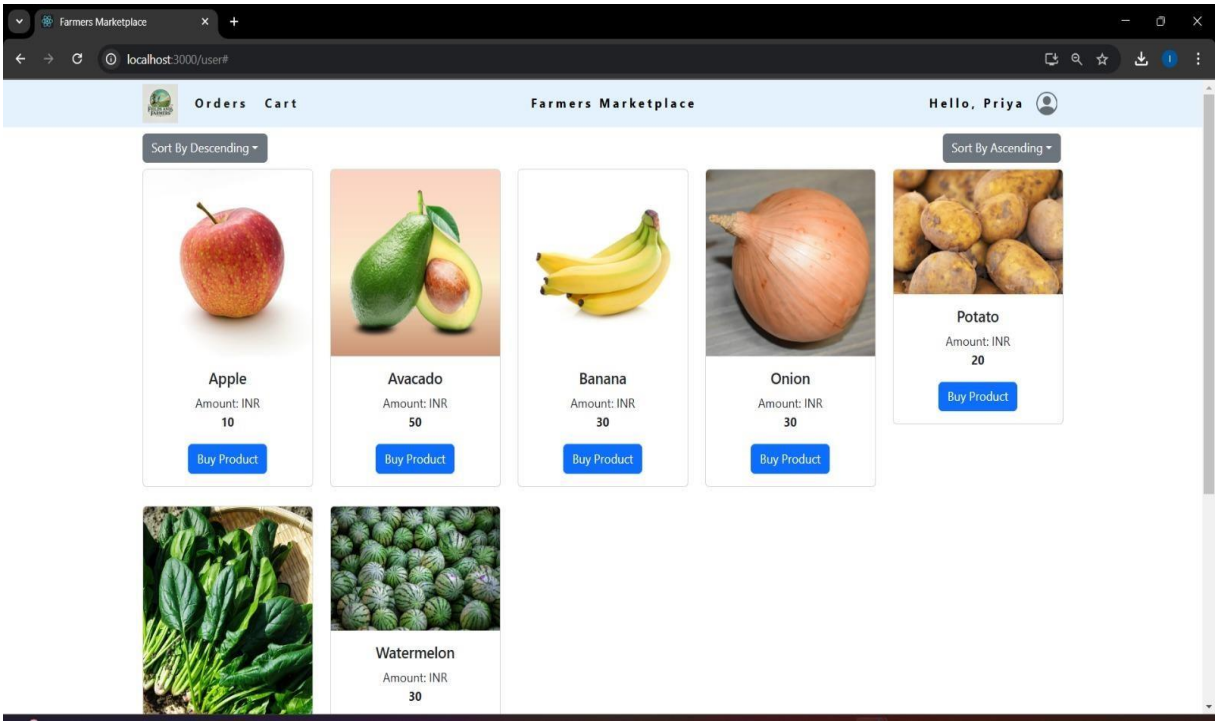
1. Register Page



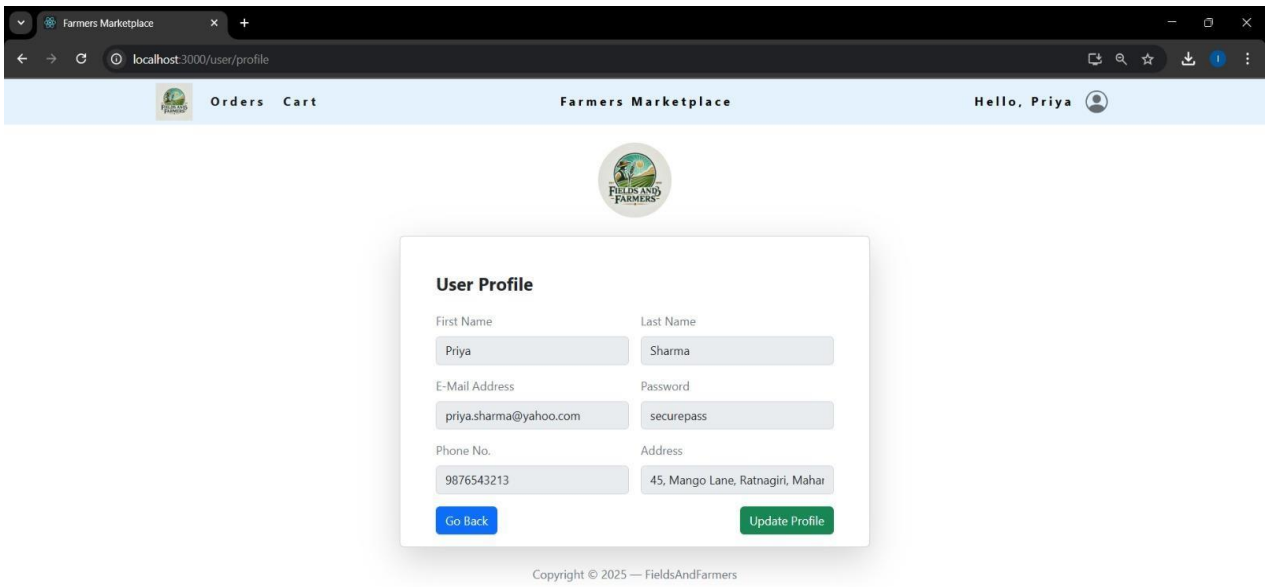
2. Login Page



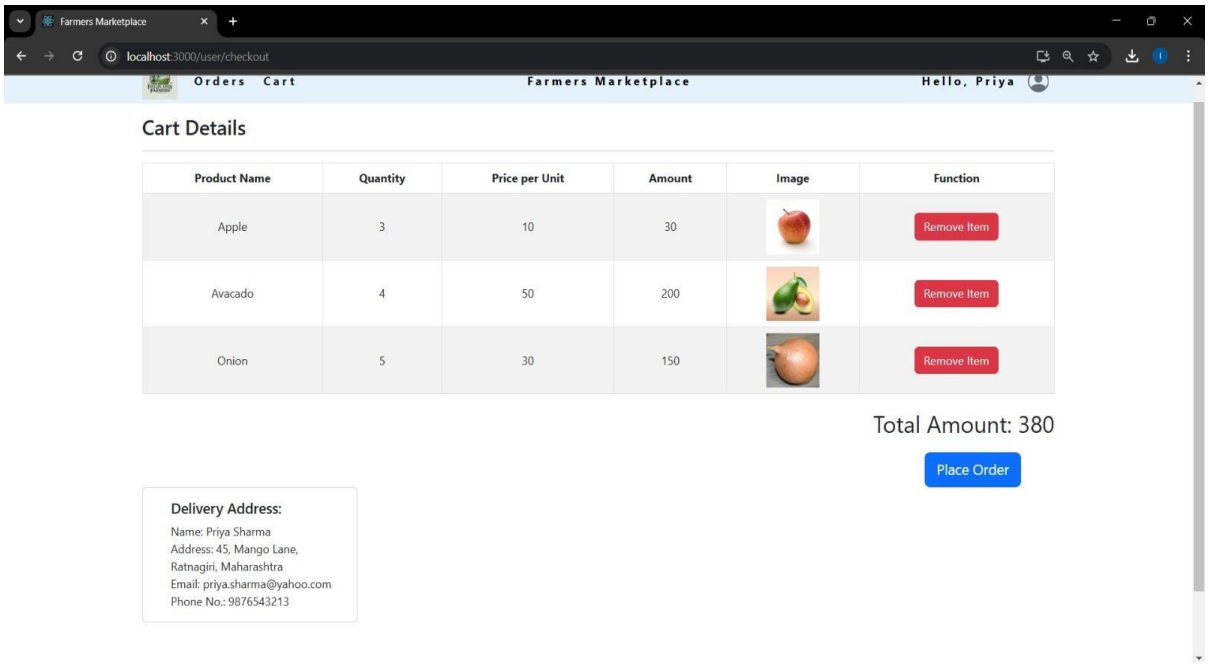
3. Home Page



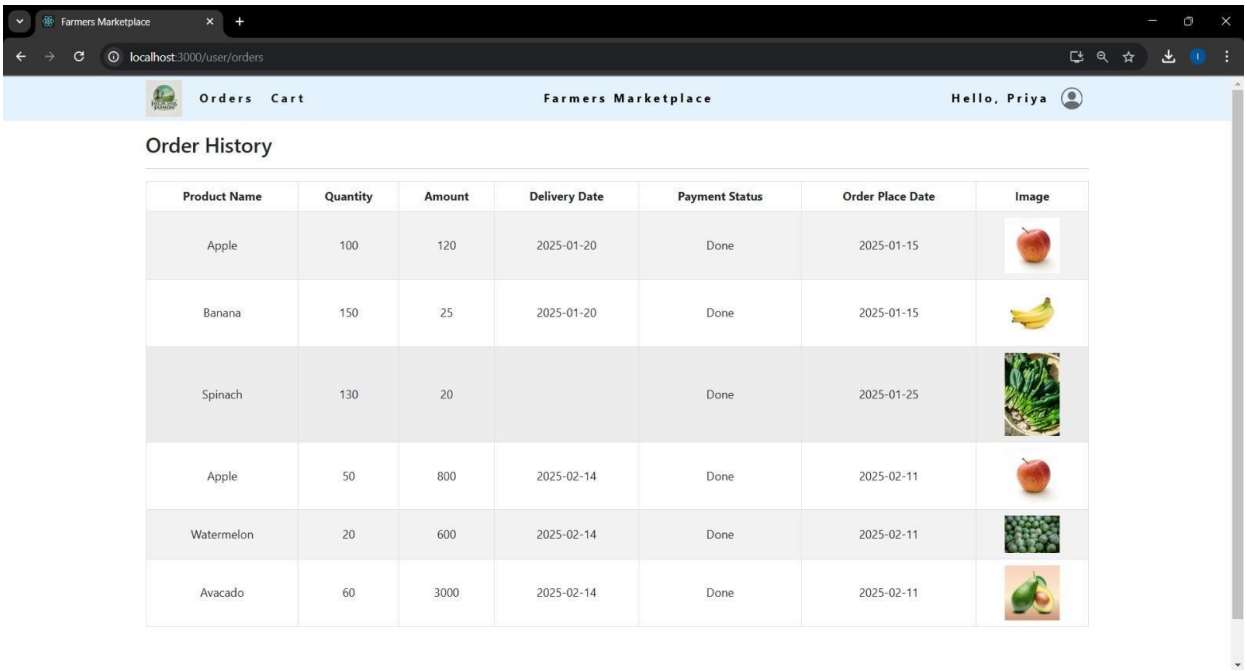
4. User Profile



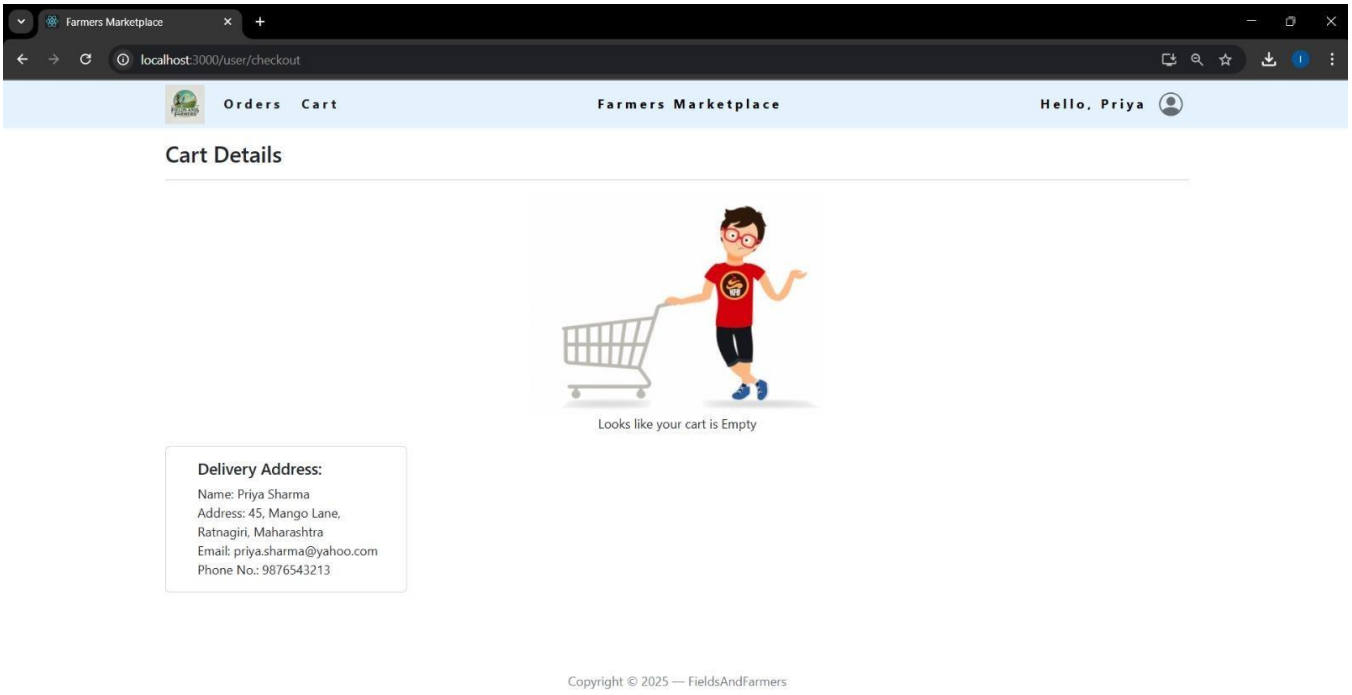
5. User Cart



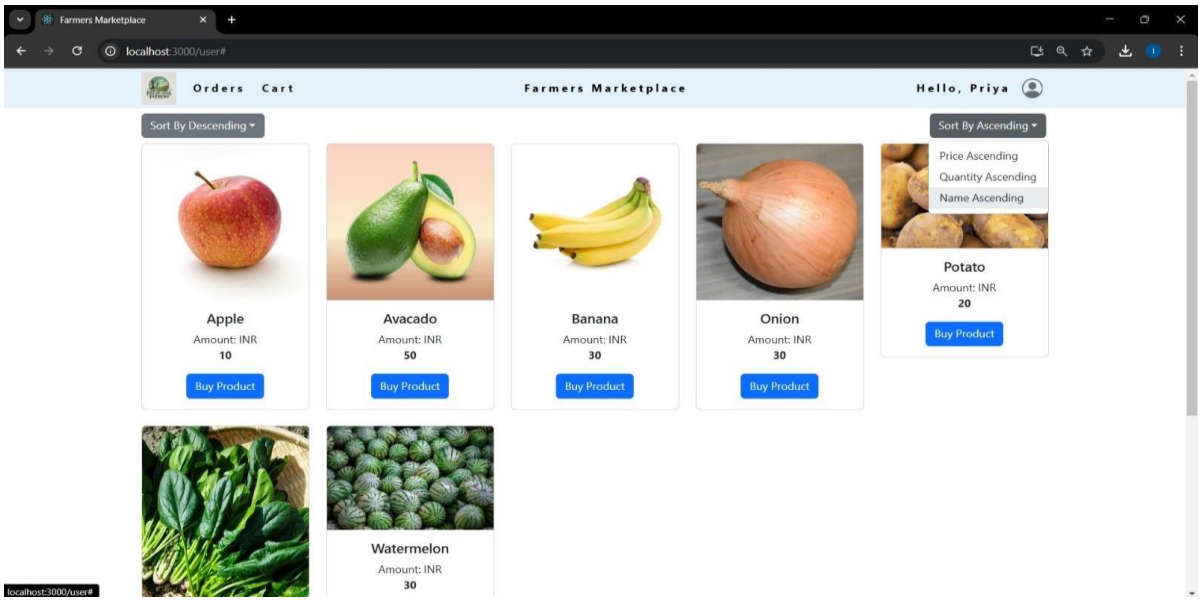
6. Order History



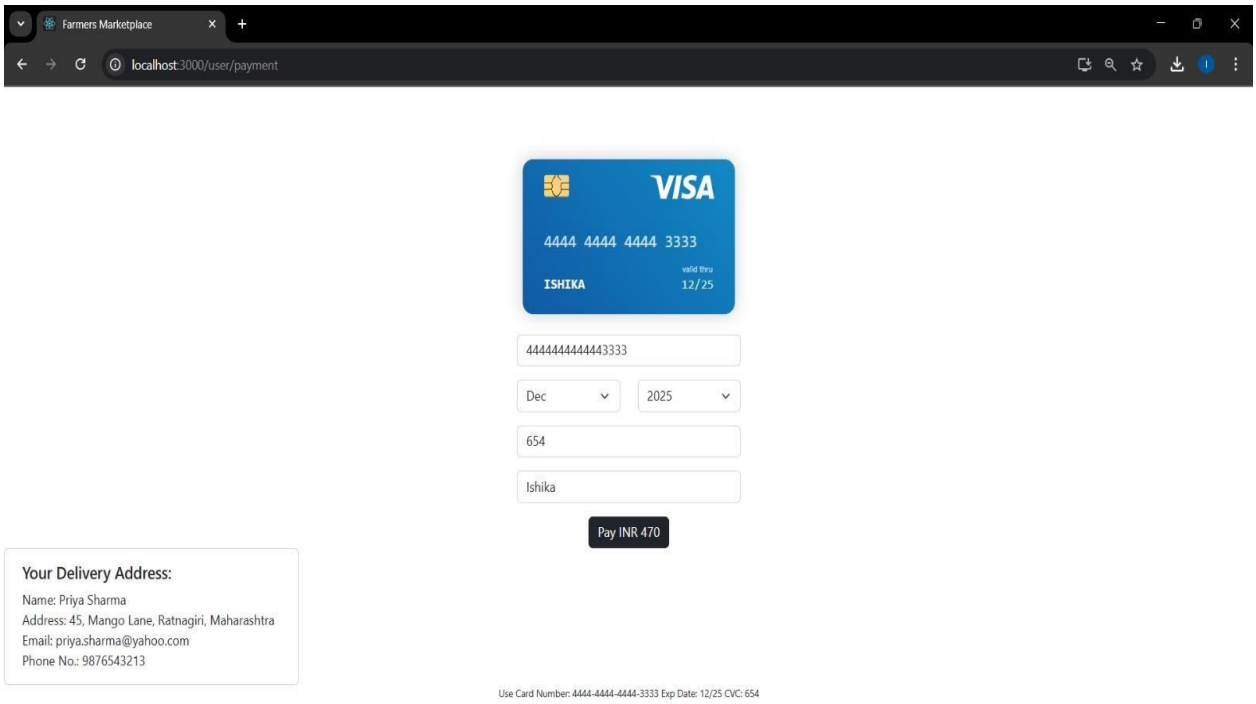
7. Empty Cart



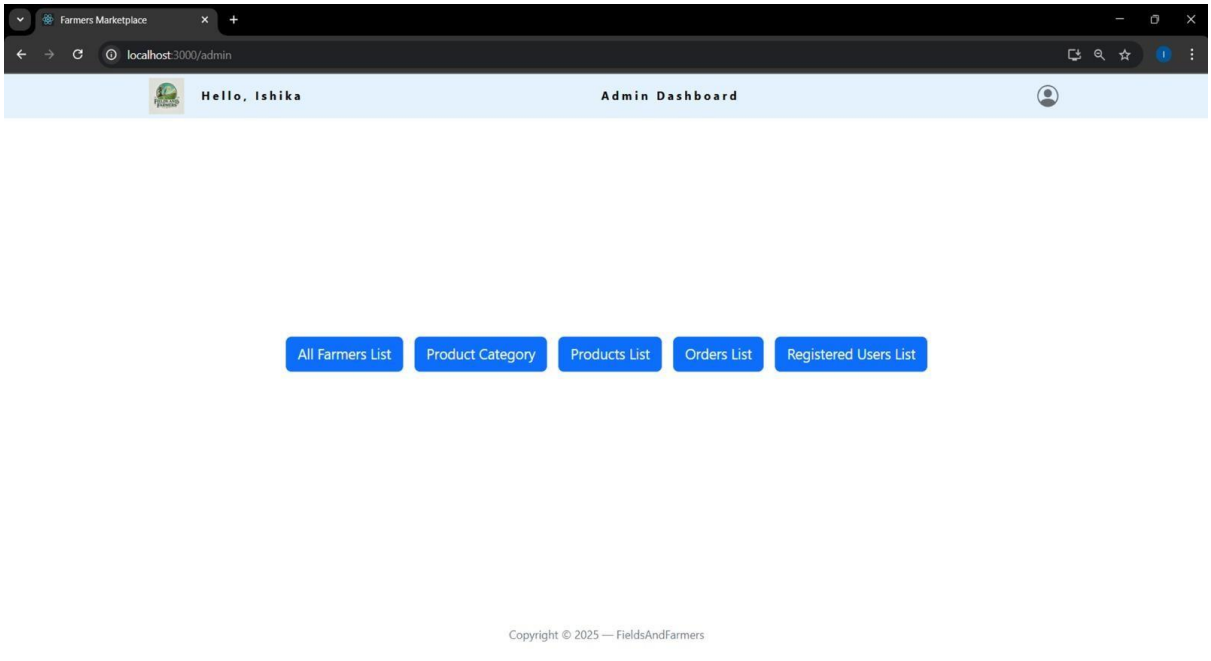
8. Sort Products



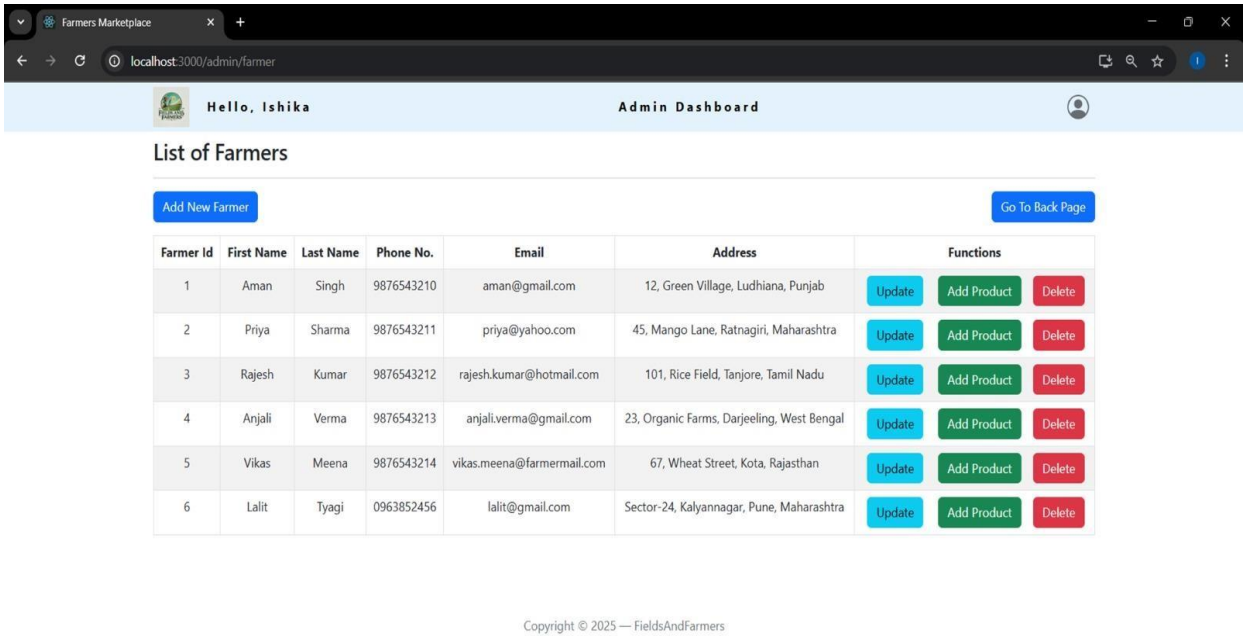
9. Payment Gateway



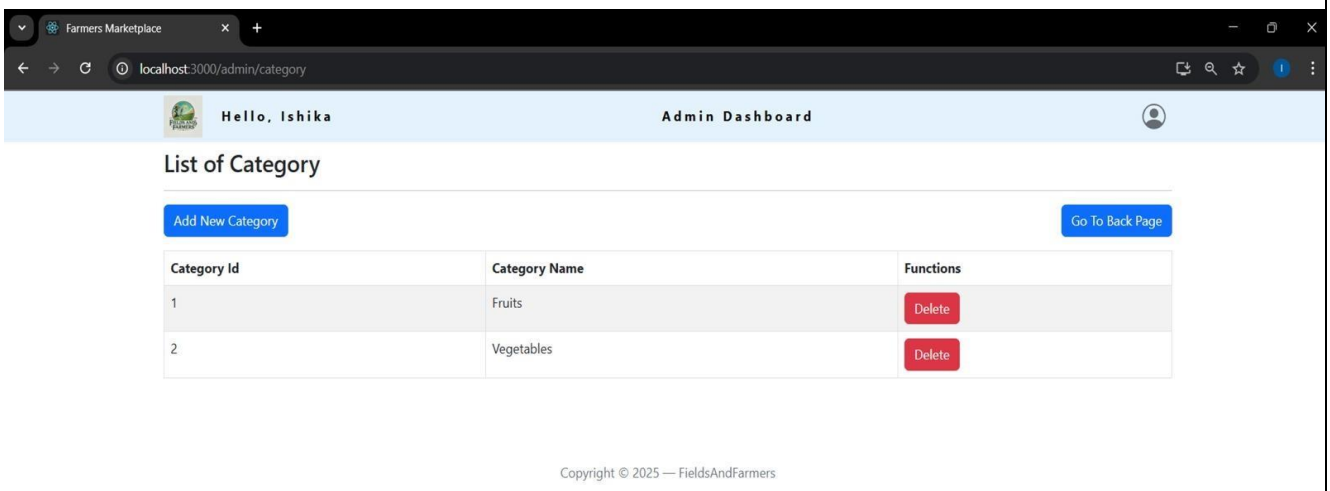
10. Admin Dashboard



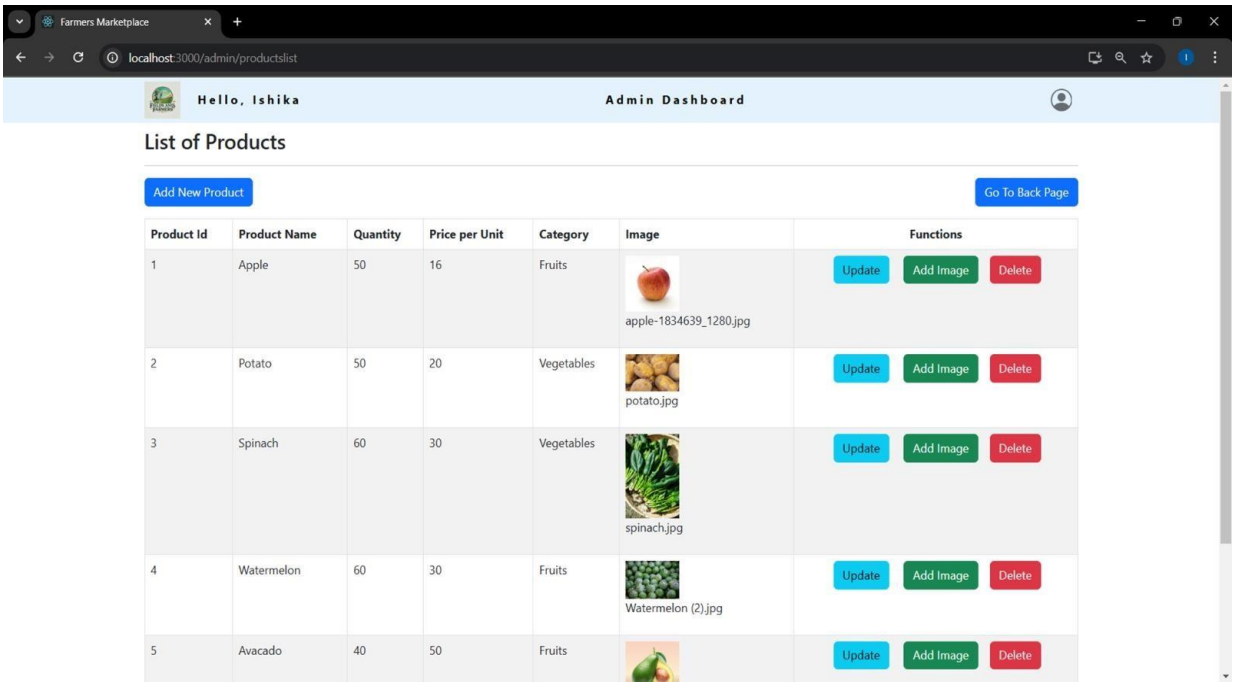
11. List of Farmers



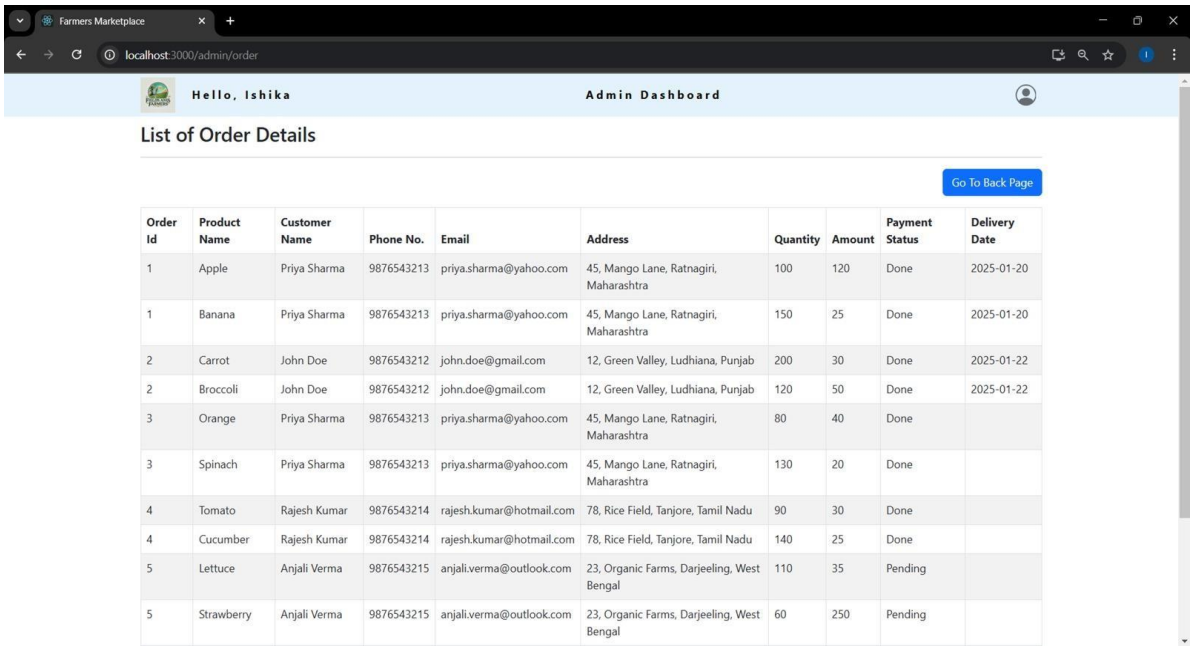
12. List of Categories



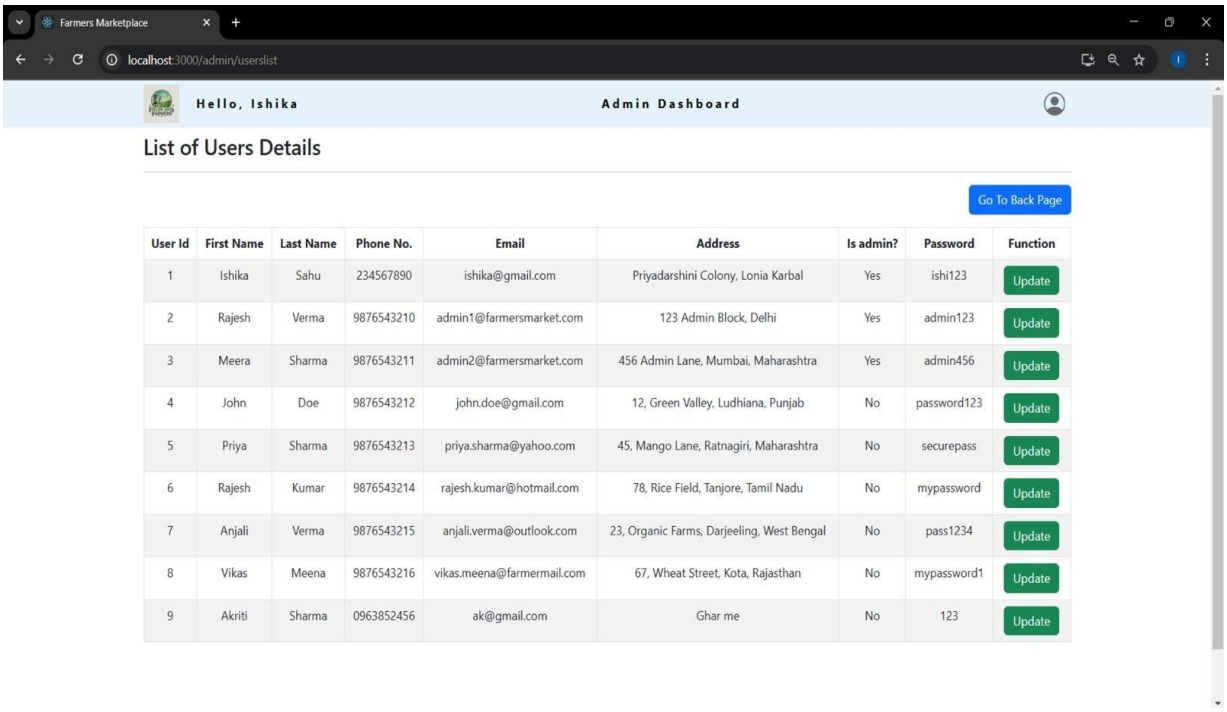
13. List of Products



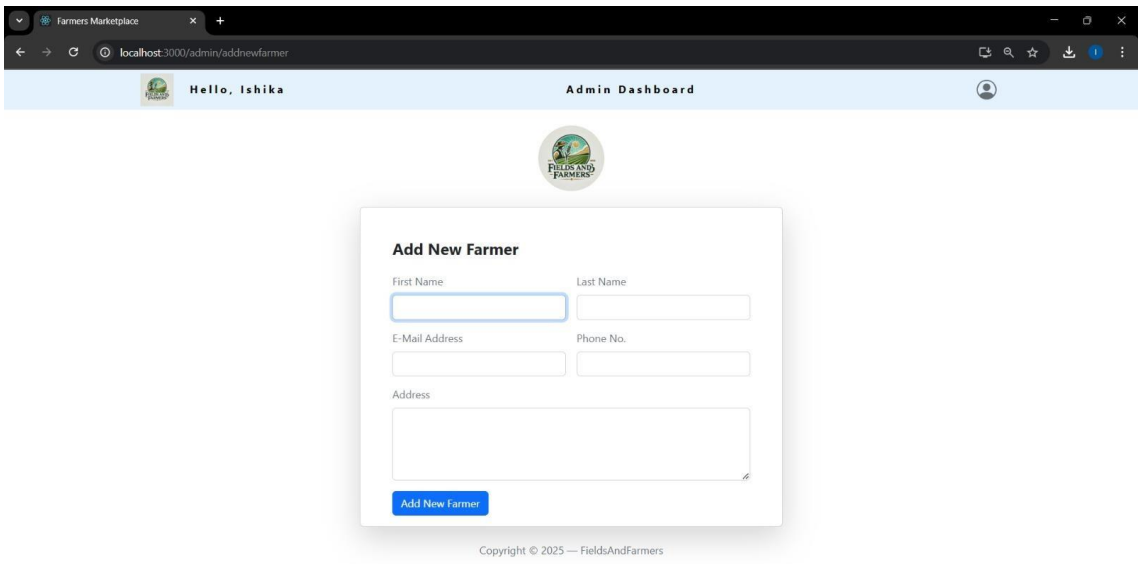
14. List of Order Details



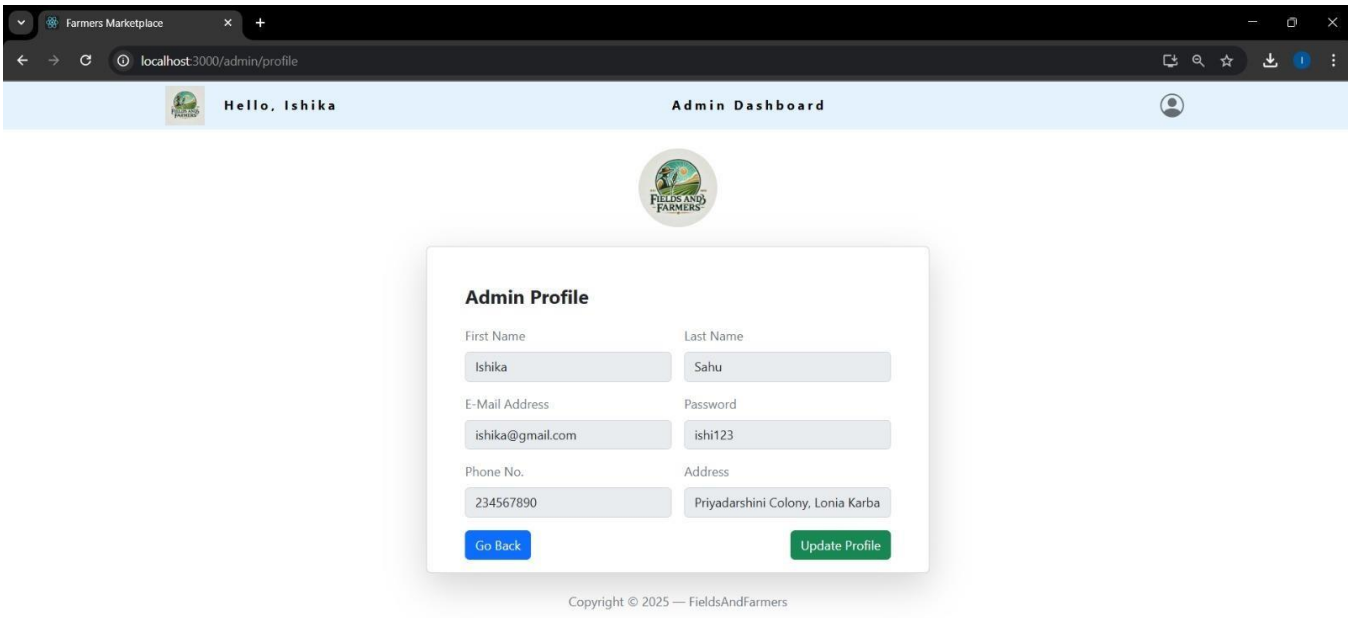
15. List of User Details



16. Add a New Farmer

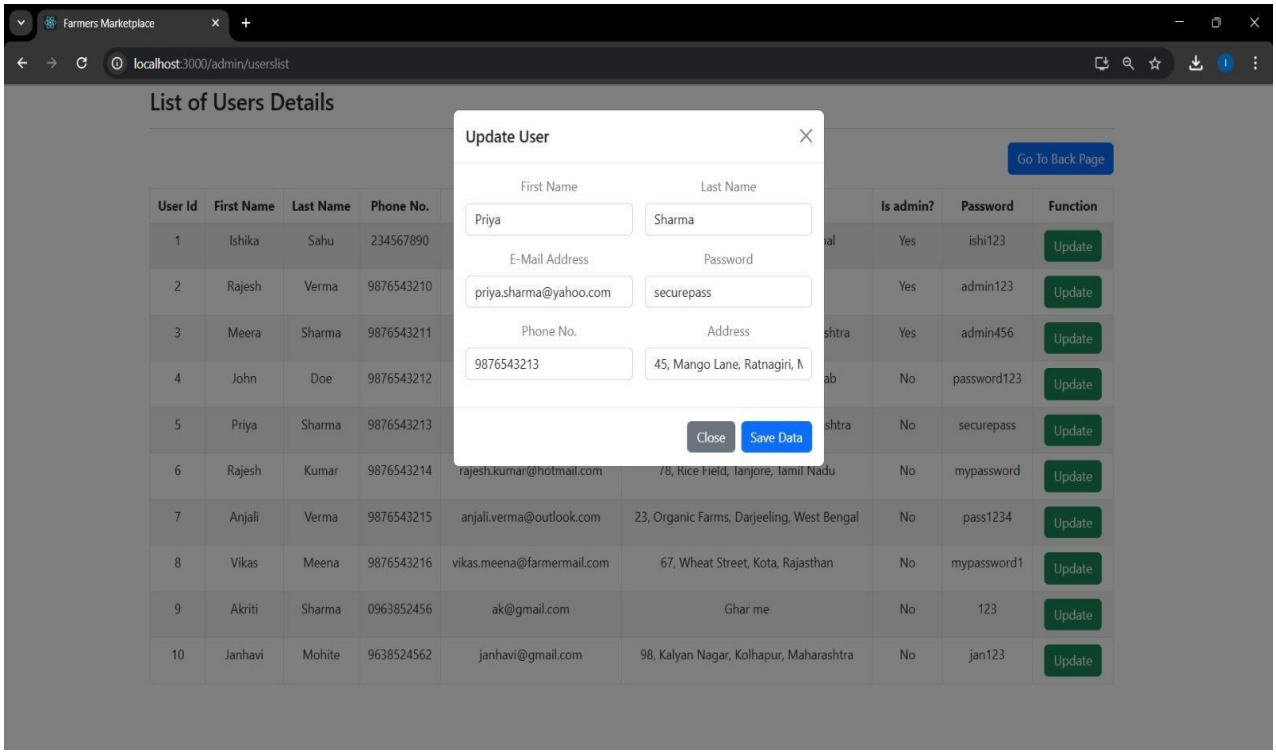


17. Admin Profile

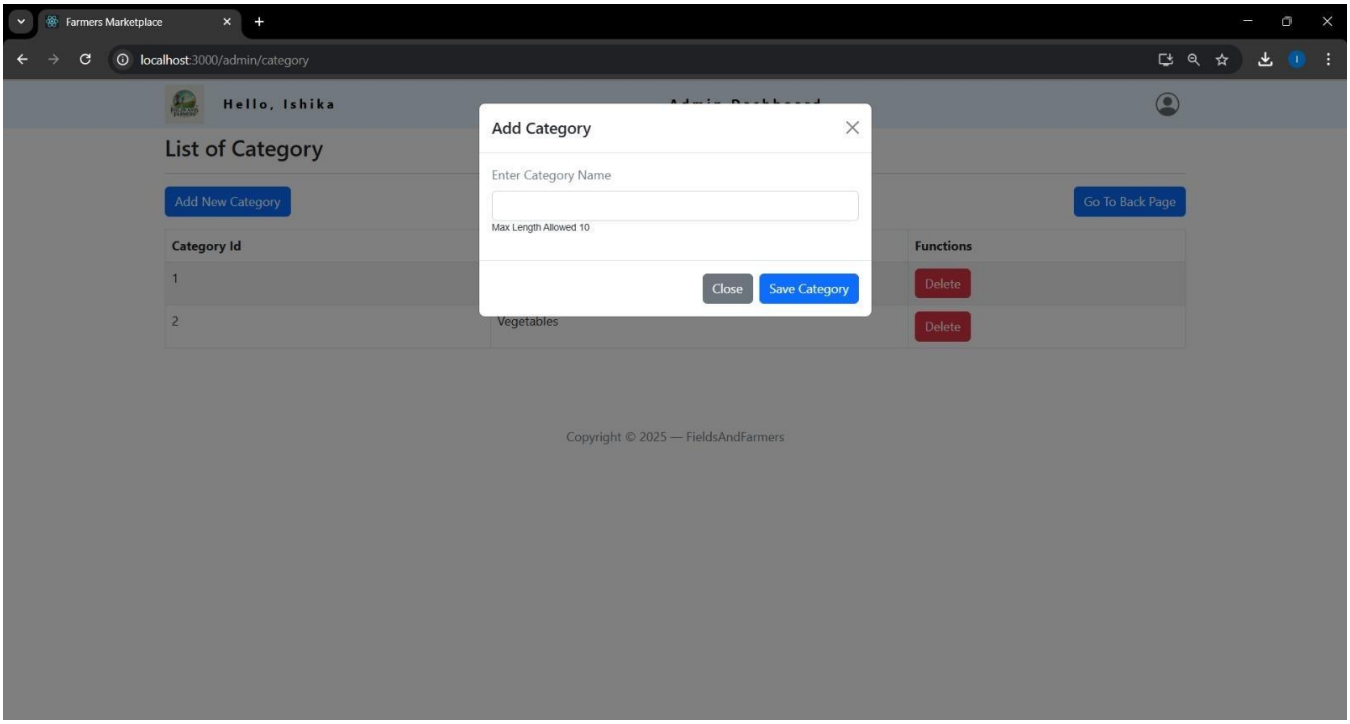


---

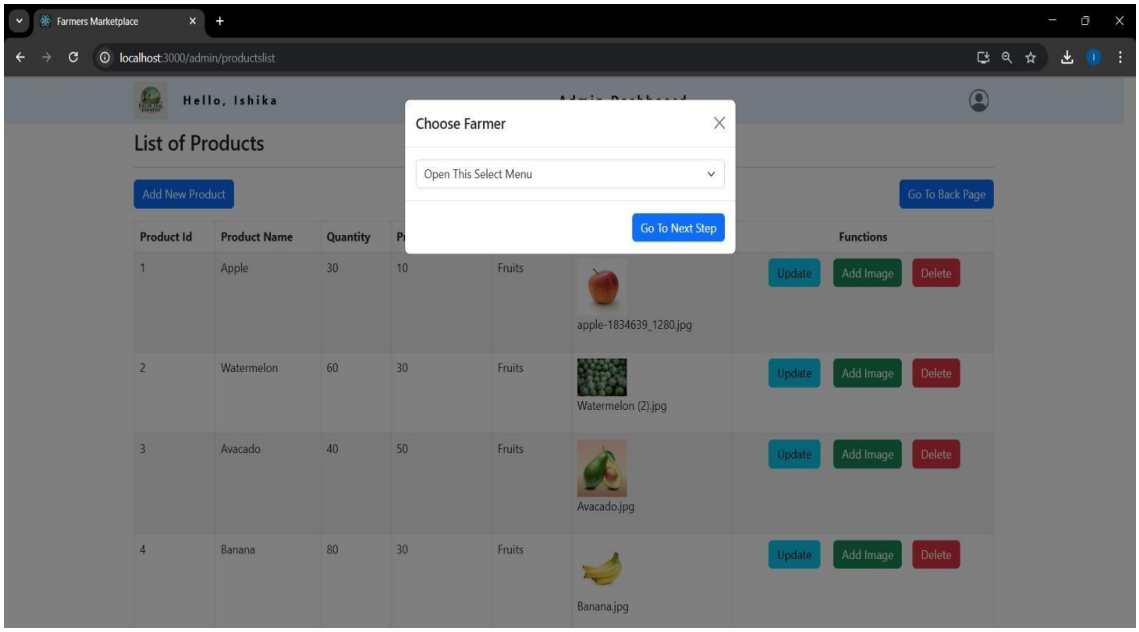
18. Update User



19. Add Category



20. Add New Product



5. CONCLUSION

Our project team created an online grocery store application called "Fields and Farmers" to make it easier to buy and sell fresh, organic products online.

We experimented with the newest, most reliable, cross-platform technology. Because all of the software we utilised was open-source, manufacturing costs were kept to a minimal. To ensure that using our website is simple and seamless, we also paid close attention to the user experience component of our application.

To sum up, "Fields and Farmers" is an app that would undoubtedly be an excellent option for any farmers that trade fresh food products and want to get into the online market. In addition, it offers a one-stop shop where customers may buy the goods they need every day straight from verified farmers.

We are sure that the application's many features and eye-catching appearance will significantly benefit farmers.

6. FUTURE SCOPE

We attempted to make our project as user-friendly and feature-rich as we could in the short amount of time allocated for the project work, using what we have acquired throughout this course.

Having said that, our application may undoubtedly benefit from further functionalities. Listed below are a few of those:

1. To further market merchandise, the most popular products that are sponsored or purchased might be promoted as consumer favourites.

2. A Farmers and Products rating chart.

3. Product Display with Farmers and corresponding Ratings per Category.

4. Depending on the customer's past purchases and the quantity of items they purchase at once, discounts may be offered on a per-user basis.

5. Customers have the ability to report feedback and upvote or downvote.

6. In addition to cards, other payment methods can be added.

7. A "reset password" feature can be included in case the user forgets their password.

8. The login page can have CAPTCHA added.

7. REFERENCES

Following is the list of websites we referred during the course of our project:

1. https://getbootstrap.com/docs/5.1/getting-started/introduction/
2. https://reactjs.org/docs/getting-started.html
3. https://www.baeldung.com/
4. https://www.w3schools.com/
5. https://docs.spring.io/spring- data/jpa/docs/current/reference/html/#reference
6. https://javaee.github.io/javaee-spec/javadocs/
7. https://javadoc.io/doc/org.springframework.data/spring-data- jpa/latest/index.html
8. https://github.com/amaroteam/react-credit-cards
9. https://github.com/Janhavi-Mohite/CDAC---PG-DAC-Project---Fields-and-Farmers