GreenSource: Direct Farmer-to-Consumer Web Application

Team Information

- **Team Name:** GreenSource
- **Team Members:** Aryan Sajan Kulathinal, Bharath Lakkoju

---

1. Problem Statement

Small farmers often face challenges such as limited market access, lack of pricing transparency, and inefficient sales processes. The existing market system, dominated by intermediaries, results in reduced profits for farmers and higher costs for consumers. The need exists for a streamlined solution enabling farmers to sell directly to consumers, fostering a fair and efficient marketplace.

Understanding of the Problem Statement

a. Problem Context: The platform is designed to empower small farmers by providing them direct access to consumers. It serves two primary groups: farmers who wish to increase profitability and transparency, and consumers looking for fresh produce at fair prices. This project addresses a common issue for farmers who struggle to get fair prices for their products due to multiple layers of intermediaries. Middlemen often take a significant cut, resulting in reduced profits for farmers and inflated prices for consumers.

b. Key Requirements Identified:

- Create a Minimum Viable Product (MVP) that connects farmers with customers.
- Ensure an intuitive user experience that enables users to easily browse, order, and communicate with farmers.
- Integrate secure payment options for transactions.
- Implement profiles for farmers and customers.
- Ensure scalability to accommodate growth in both farmers and customer base.
- Direct farmer-consumer communication, bypassing intermediaries.
- Transparent pricing aligned with market standards via a government API.
- Efficient product listing and order management functionalities.
- Delivery tracking and order management
- Support through an admin dashboard.

---

2. Solution Overview

a. Solution Summary:

The proposed solution is a comprehensive web application enabling farmers to list and manage products, view real-time market prices, and communicate directly with consumers. The admin dashboard centralizes order and user management, while a delivery module streamlines product delivery. In short, the GreenSource platform allows farmers to list their products and customers to browse and purchase directly. It removes middlemen, ensuring that farmers receive fair compensation while customers get quality, farm-fresh products. The solution also includes a user-friendly interface, profiles for farmers and consumers, order tracking, and a secure payment gateway.

b. Objectives:

- Create a transparent platform for direct farmer-to-consumer transactions.
- Increase profitability for farmers by removing intermediaries.
- Provide fresh produce to customers at competitive prices.
- Build a community around sustainable agriculture.
- Facilitate a seamless, user-friendly connection between farmers and consumers.
- Ensure that farmers can manage inventory and pricing while tracking market rates.
- Provide consumers with easy access to fresh produce and efficient order tracking.
- Use TypeScript across the application to enhance code quality, readability, and error handling.

---

3. Features and Functionalities

a. Core Features:

- **User Registration and Authentication**: Secure login for farmers and consumers.
- **Product Listing**: Farmers can add and manage their products, including descriptions, prices, and availability.
- **Product Browsing and Search**: Customers can browse products by categories or search directly.
- **Order Placement and Tracking**: Customers can place orders and track delivery status.
- **Product Management**: Farmers can add, edit, and remove product listings, complete with descriptions and images.
- **Market Price Viewing**: Farmers can view real-time market prices for competitive pricing.
- **Order Management**: Consumers can place orders, and farmers can manage and track them.
- **Delivery Service**: Supports tracking and updating of delivery status for both farmers and consumers.
- **Typescript Integration**: Used extensively in both frontend and backend to improve maintainability and type safety.
- **Admin Dashboard**: Provides centralized control over product listings, orders, user accounts, and analytics.

b. Additional Features:

- **Rating and Reviews**: Customers can review products, allowing farmers to build trust and credibility.
- **Notification System**: Real-time notifications for order updates and other related services.

c. User Flows:

• **Registration and Profile Setup**: Farmers, customers, and admins can sign up and complete their profiles. Farmers set up essential details for product listings, while customers and delivery agents manage personal details and contact information.

• **Product Management and Browsing**: Farmers add, edit, and remove product listings, including descriptions, prices, images, and inventory management. Customers browse products, filter by category, location, and price, and add items to their cart.

• **Order Placement, Confirmation, and Tracking**: Customers complete purchases at checkout, while farmers receive and confirm incoming orders. Orders are updated in real-time, allowing customers to track status and view order history. Admins oversee order processing to ensure smooth operations.

• **Market Price Access and Analytics**: Farmers view real-time market prices from a government API to set competitive pricing. Additionally, they access sales analytics, view customer feedback, and review product popularity trends. Admins monitor platform analytics, including engagement and order volume.

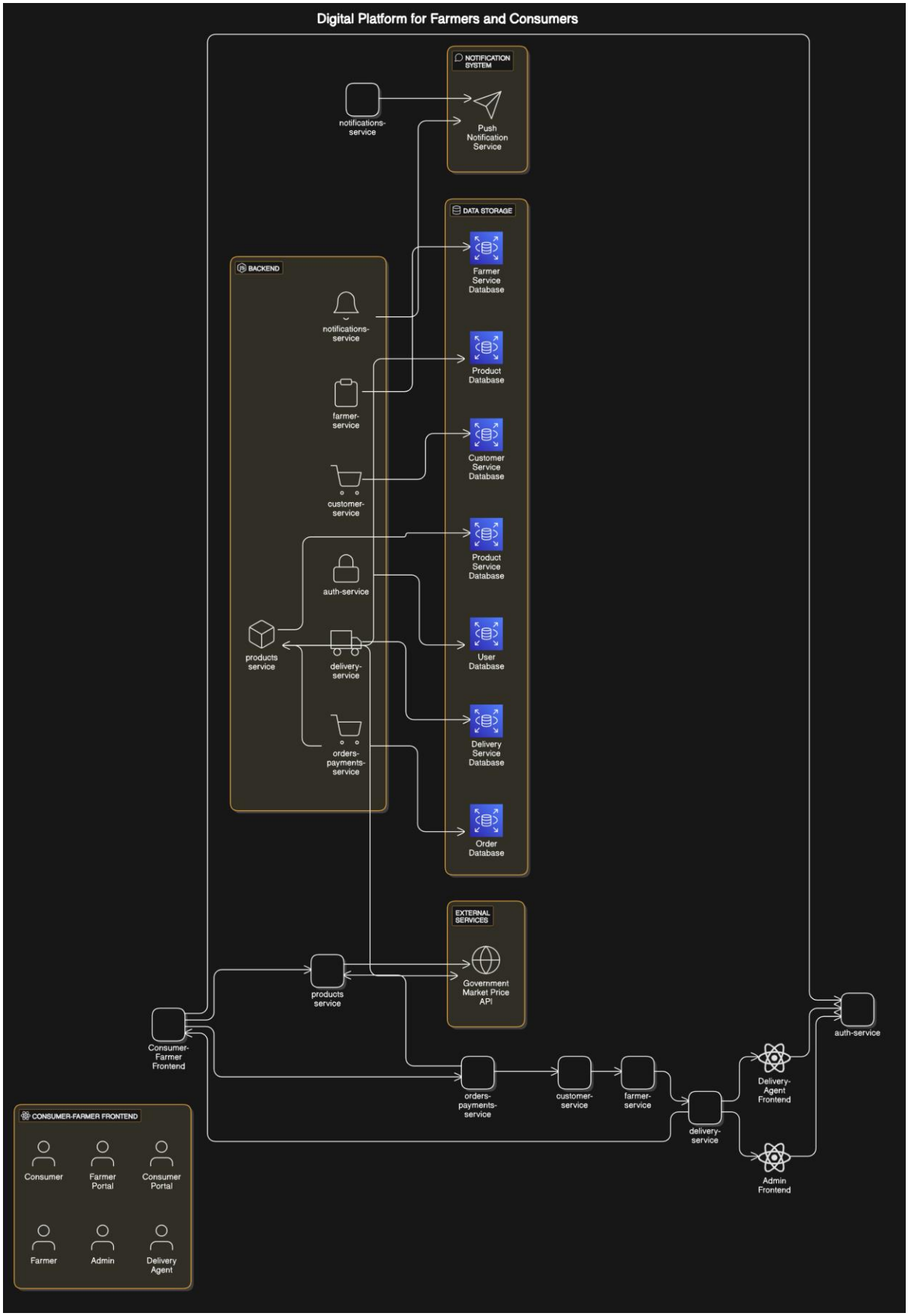
• **Order Delivery and Notifications**: Delivery agents receive assigned orders, manage delivery routes, and update order statuses for customers and farmers. Upon completion, the order status is updated, notifying all parties involved. Admins oversee order assignments and monitor delivery performance.

• **Feedback, Ratings, and Performance Metrics**: Customers leave ratings and reviews on products, providing valuable feedback for farmers. Farmers and delivery agents access personal dashboards showing ratings, performance insights, and customer preferences, while admins can manage content, user accounts, and platform configurations.

---

4. Architecture Diagram

a. System Architecture: The architecture includes a frontend interface (React with TypeScript), a Node.js backend, and MongoDB database. Components include the admin dashboard, farmer and consumer interfaces, database, APIs, and notification system.


Digital Platform for Farmers and Consumers
NOTIFICATION SYSTEM
notifications-service
Push Notification Service
DATA STORAGE
BACKEND
Farmer Service Database
notifications-service
Product Database
farmer-service
Customer Service Database
customer-service
Product Service Database
auth-service
products service
delivery-service
User Database
orders-payments-service
Delivery Service Database
Order Database
EXTERNAL SERVICES
products service
Government Market Price API
auth-service
Consumer-Farmer Frontend
orders-payments-service
customer-service
farmer-service
Delivery-Agent Frontend
delivery-service
Admin Frontend
CONSUMER-FARMER FRONTEND
Consumer
Farmer Portal
Consumer Portal
Farmer
Admin
Delivery Agent

b. Key Components:

• **Frontend**: Built with React and styled using Tailwind CSS, enabling interactive user experiences.
• **Backend**: Node.js and Express handle user authentication, order management, and integration with government APIs.
• **Database**: MongoDB stores the following:

- **Farmers**: Profiles and product listings.
- **Consumers**: User information, order history, and preferences.
- **Users**: Login credentials and authentication roles for admins, delivery agents, farmers, and consumers.
- **Products**: Detailed product information, inventory, and price.
- **Orders and Payments**: Order details, payment status, and delivery tracking.
- **Details**: Market price data retrieved from external API sources.

---

5. Technical Stack

a. Frontend:

- **React** (for component-based user interfaces)
- **Redux** (for centralized state management)
- **Tailwind CSS** (for efficient and responsive styling)
- **Axios** (for handling API requests)
- **TypeScript** (for enhanced code structure and error handling)

b. Backend:

- **Node.js** (for server-side logic)
- **Express.js** (for routing and middleware)
- **TypeScript** (used to improve type safety and reduce runtime errors)
- **Axios** (for external API requests, including the government market price API)

c. Database:

- **MongoDB**: Stores user data, product information, and orders, with schema tailored for high-performance retrieval.

---

6. Prerequisites and Requirements

a. Technical Requirements:

- Development environment supporting React, Node.js, MongoDB, and TypeScript.
- Access to APIs for retrieving real-time market price data.

b. Data Requirements:

- **Government Market Price API**: Necessary for displaying up-to-date market rates.
- **User Data**: For creating user profiles and managing product listings and orders.

c. Access Permissions:

- User roles and permissions, with access to the admin dashboard limited to authorized users.

---

7. Future Improvements [Optional]

a. Planned Enhancements:

- **Advanced Analytics**: Provide farmers with insights on sales and customer preferences.
- **Payment Gateway Integration**: To streamline consumer payments.
- **Enhanced Analytics**: Adding more advanced insights for farmers and administrators.
- **Improved Admin Dashboard**: Additional features for data visualization and insights.

Conclusion

a. Summary of Achievements:
GreenSource successfully provides a direct link between farmers and consumers, fostering a fair marketplace. The solution addresses the problem by removing intermediaries, ensuring transparency, and providing fresh products to consumers. In short, it successfully bridges the gap between farmers and consumers, offering a streamlined platform for fresh produce sales. By integrating real-time market insights, delivery tracking, and robust order management, it addresses the core challenges identified in the problem statement.

b. Value Provided:
This platform provides significant value to both farmers and consumers. Farmers gain access to a broader market and pricing insights and empowers them by giving them control over pricing and product distribution., while consumers benefit from direct access to fresh, local produce.