

# ADDRESSING THE INCOME GAP CAUSED BY MIDDLEMEN IN AGRICULTURAL MARKETING THROUGH DIRECT FARMER-TO- CONSUMER SOLUTIONS

Jayasmruthi A

*Computer Science and Engineering  
Sri Ramakrishna Institute of Technology  
(Autonomous Institution)  
Coimbatore, India.  
Jayasmruthi.cse@sritce.ac.in*

Sooraj R Nair

*Computer Science and Engineering  
Sri Ramakrishna Institute of Technology  
(Autonomous Institution)  
Coimbatore, India  
soorajnair.2202152@sritce.ac.in*

Vigneshwaran S

*Computer Science and Engineering  
Sri Ramakrishna Institute of Technology  
(Autonomous Institution)  
Coimbatore, India  
vigneshwaran.2202173@sritce.ac.in*

Sanjay S

*Computer Science and Engineering  
Sri Ramakrishna Institute of Technology  
(Autonomous Institution)  
Coimbatore, India  
sanjay.2202130@sritce.ac.in*

Sarukesh A

*Computer Science and Engineering  
Sri Ramakrishna Institute of Technology  
(Autonomous Institution)  
Coimbatore, India  
sarukesh.2202134@sritce.ac.in*

**Abstract** - By keeping a sizable portion of the consumer price, intermediaries, who frequently control traditional agricultural marketing systems, drastically lower farmers' profits. Through open, technologically driven interactions, this project suggests a digital direct farmer-to-consumer (F2C) platform that aims to reduce the income disparity between producers and consumers. The system architecture incorporates MongoDB Atlas for safe data management, Node.js and Express for backend functions, and React for the frontend. Stripe and Shiprocket APIs manage payments and logistics, respectively, while Socket.io is used to implement real-time updates.

Compared to traditional marketing models, experimental evaluations showed a 20% decrease in consumer costs and a 30–50% increase in farmer profits. Under 2,500 concurrent users, performance testing revealed 99.8% payment success, 99% uptime, and an average page load time of 1.1 seconds. The findings confirm that a well-designed digital F2C system can successfully cut out middlemen, increase farmer revenue, and foster an open and sustainable agricultural market.

**Keywords** - Direct Farmer-to-Consumer (F2C), Agricultural Marketing, Income Gap, Middlemen Elimination, Rural E-Commerce, Web Application, Node.js, React, MongoDB Atlas, Socket.io, Stripe API, Supply Chain Transparency, Digital Agriculture, Farmer Profitability, Sustainable Marketing

## I. INTRODUCTION

For millions of people in developing nations, agriculture serves as their main source of income and makes a substantial contribution to both national GDP and food security. Despite its significance, the dominance of middlemen in the agricultural marketing chain makes it difficult for most farmers to obtain fair returns. These intermediaries frequently set prices, restrict access to markets, and keep a sizable portion of consumer payments, leaving producers with slim profit margins. Rural economic growth and sustainability have been hampered by this

imbalance, which has resulted in a persistent income gap between producers and consumers.

The advent of digital technologies presents a revolutionary chance to tackle these enduring inefficiencies. Farmers can interact directly with consumers by avoiding traditional middlemen thanks to direct farmer-to-consumer (F2C) marketing models made possible by digital platforms. Farmers can list their produce online, interact with buyers directly, and get safe digital payments by utilizing e-commerce technologies. Fresh, traceable, and reasonably priced produce benefits consumers in turn. This reciprocal openness promotes fair value distribution, increases market efficiency, and builds trust.

A web-based F2C system has been created as part of this project to close the income gap brought on by middlemen. To guarantee scalability, security, and real-time communication, the platform incorporates a number of technologies. The user interface of the frontend, which was created with React (HTML5/CSS3/JavaScript ES6+), is responsive and easy to use. The backend, which is driven by Express and Node.js, controls transaction workflows, authentication, and user requests. MongoDB Atlas securely stores data, and Socket.io allows chat communication and order updates to be synchronized in real time. To ensure seamless end-to-end operations, the system also integrates Shiprocket for logistics management and Stripe for payment processing.

To ensure performance reliability, the system underwent extensive testing in simulated real-world scenarios. 180 successful orders per minute, 99% uptime, and an average page load time of 1.1 seconds were achieved using Apache JMeter load testing with 2,500 concurrent users. Additionally, Redis caching decreased response times from 380 milliseconds to 95 milliseconds, and Stripe integration maintained 99.8% payment success with refund processing under 200 milliseconds.

According to the results of the experiment, farmers who used the platform saw increases in their income of between 30% and 50% when compared to traditional channels. Fresher produce was more affordable for consumers, and stronger farmer-consumer ties were promoted by transaction transparency. By lowering reliance on middlemen and generating digital job opportunities in farming communities, the system also helped to empower the rural economy.

This research demonstrates how digital F2C systems have the potential to transform agricultural marketing by enhancing inclusivity, efficiency, and transparency. The suggested approach, which makes use of contemporary web technologies, helps to reduce the income disparity in rural areas while promoting sustainable farming methods and increasing farmer profitability.

## II. LITERATURE SURVEY

Recent studies have extensively examined the function of intermediaries in agricultural marketing. According to studies, middlemen are largely responsible for managing agricultural supply chains. They frequently buy produce at low farm-gate prices and resell it at much higher retail prices, which lowers farmers' net income. Smallholder farmers are deterred from engaging in formal markets by this price discrepancy, which leads to economic inefficiency.

Direct marketing models like farmers' markets, community-supported agriculture (CSA), and cooperatives have emerged as a result of efforts to address these issues. With the help of these models, farmers can sell directly to customers, increasing transparency and trust while keeping a bigger portion of the profits. However, especially in rural and isolated areas, traditional direct marketing strategies are not scalable, have a limited geographic reach, and mainly depend on physical presence.

Researchers' attention has been drawn more to e-agriculture and rural e-commerce systems as a result of the development of digital technologies. Direct connections between farmers and customers and retailers may be possible through platforms that combine digital payment methods, web portals, and mobile applications. Digital agricultural platforms can increase household incomes by 15–25% and improve small-scale farmers' access to markets, according to studies from nations like China and India. Additionally, integrating underserved rural communities into digital commerce ecosystems helps to reduce regional income disparities.

The benefits of e-commerce and agricultural market reform are combined in recent advancements in Farmer-to-Consumer (F2C) digital platforms. These platforms offer end-to-end services, such as real-time tracking, online payment, logistics integration, and product listing. Transparency and operational dependability are guaranteed by integrating logistics APIs like Shiprocket with secure payment gateways like Stripe and Razorpay. Furthermore, scalable cloud databases like MongoDB Atlas enable effective data management and real-time synchronization

when paired with contemporary backend frameworks like Node.js and Express.

A number of studies also stress how crucial data analytics and real-time communication are to enhancing system performance and decision-making. Live order and transaction updates are made possible by WebSockets (such as Socket.io), which increases user engagement and trust. Farmers can improve their income stability by using analytical dashboards to monitor demand trends and adjust pricing strategies.

In summary, literature shows a clear shift from traditional intermediary-based systems to direct marketing ecosystems powered by digital tools. Existing models have potential, but there is still a need for user-friendly and well-performing platforms that meet the economic realities of small and medium-scale farmers. This study proposes a full-stack F2C web solution. It uses modern web technologies to promote transparency, efficiency, and fair income distribution in agricultural markets.

## III. METHODOLOGY

The proposed system follows a clear design and implementation method. It aims to create a digital platform that connects farmers directly with consumers, cutting out intermediaries and ensuring fair price distribution. The development process included five main phases: requirement analysis, system design, implementation, integration, and evaluation.

### A. Requirement Analysis

We analyzed the problem by looking at traditional agricultural marketing structures. We identified how intermediaries affect pricing and determined user needs for both farmers and consumers. Surveys and discussions with local farmers revealed common challenges, such as limited market access, lack of transparency, and inconsistent payments. From these insights, we categorized system requirements into functional needs (user registration, product listing, order management, payment, and delivery tracking) and non-functional needs (security, performance, and scalability).

### B. System Design

The system was built using a three-tier structure with frontend, backend, and database layers. The frontend, developed with React (HTML5/CSS3/JavaScript ES6+), offers a responsive and user-friendly interface. The backend, created with Node.js and Express, manages business logic, user authentication, and API communication. We used MongoDB Atlas for data management, allowing for scalability and quick data retrieval. The Socket.io library enables real-time communication for order tracking and notifications. We integrated the Stripe API for secure payment processing and used the Shiprocket API for logistics and delivery management.

### C. Implementation

We implemented the system using the MERN (MongoDB, Express, React, Node.js) stack. Axios facilitated API communication between the frontend and backend, while JWT tokens ensured secure user authentication. We achieved real-time synchronization through Socket.io, cutting down page reloads by 85% and keeping dashboard updates under 1.1 seconds. The backend logic included data validation, order processing, payment verification, and delivery scheduling. Stripe's webhook mechanism provided reliable payment tracking with a 99.8% success rate, and Shiprocket integration enabled automatic order dispatch and tracking.

### D. Integration and Testing

After implementation, we integrated and tested all modules using both functional and load testing methods. We used Postman for API testing and JMeter for performance evaluation with 2,500 concurrent users. Redis caching reduced API response time from 380 ms to 95 ms, while MongoDB handled over 10,000 operations per second during stress tests. Integration testing ensured smooth operation among the frontend, backend, payment, and logistics components

### E. Evaluation

The system was evaluated with real users—farmers and consumers—over a four-week pilot period. Farmers used the platform to list produce and make direct sales. Results showed that average farmer income increased by 30 to 50%, while consumer costs dropped by about 20%. The platform achieved 99% uptime, an average load time of 1.1 seconds, and processed orders smoothly at a rate of 180 orders per minute. The evaluation confirmed that the platform improved market transparency and supported fair agricultural trade.

### Problem Statement

In the traditional agricultural marketing system, farmers depend on middlemen like wholesalers, commission agents, and retailers to sell their produce. These intermediaries cut into farmers' profit margins by taking a big share of the final price paid by consumers. The lack of transparency, inefficient pricing methods, and delayed payments make the income gap between farmers and consumers even bigger.

Farmers face several key challenges: Low income returns because of multiple intermediaries. Lack of direct market access and consumer interaction. Delayed or unfair payments. Limited awareness of digital tools that could improve sales efficiency. This issue requires a digital direct-to-consumer (D2C) solution that ensures transparency, fair pricing, and accessibility for both farmers and consumers.

### Proposed Solution

To address these challenges, we are developing a web-based Farmer-to-Consumer (F2C) platform using modern web technologies. This system allows farmers to list their products, manage orders, receive online payments, and

coordinate logistics directly with consumers, eliminating intermediaries

The solution includes:

MERN stack (MongoDB, Express, React, Node.js) for scalable and efficient development.Socket.io for real-time updates, such as orders, payments, and tracking.Stripe API for secure and automated payments.Shiprocket API for logistics and delivery integration.MongoDB Atlas for centralized cloud data management and analytics.

The system provides transparent pricing, real-time transaction updates, and direct communication between farmers and buyers, ultimately improving income by 30 to 50 percent.

### Problem and Solution Summary

Table I. Identified Problems and Proposed Digital Solutions in Agricultural Marketing

Identified Problem	Proposed Digital Solution	Expected Outcome
Farmers earn less due to multiple intermediaries	Direct F2C e-commerce platform for farmers	30-50% income increase
Lack of transparency in pricing	Real-time pricing and digital transactions	Transparent, fair trade
Payment delays and disputes	Secure payment gateway (Stripe) with instant updates	Reliable and immediate payments
Limited consumer reach	Online marketplace accessible via web and mobile	Expanded market reach
Lack of digital literacy	Integration of Shiprocket API with real-time tracking	75% reduction in manual logistics

### Methodology Overview

- **Requirement Analysis** – Identifying needs of farmers and consumers through surveys.
- **System Design** – Creating architecture and data models using UML and ER diagrams.
- **Implementation** – Developing and integrating all system modules using MERN stack.
- **Integration and Testing** – Verifying functionality and performance through unit and load tests.
- **Evaluation** – Measuring system effectiveness in improving farmer income and market transparency.



Fig 1 Methodology Workflow

## IV. IMPLEMENTATION PROCESS

The proposed system was implemented in a clear way to ensure smooth development, integration, and deployment. It was built using the MERN stack (MongoDB, Express.js, React.js, and Node.js) to ensure it is scalable, responsive, and performs well for real-time agricultural trading between farmers and consumers.

## A. System Architecture

The system architecture follows a three-tier model made up of the frontend, backend, and database layers. Frontend Layer: This layer uses React.js to create an interactive, mobile-friendly interface for both farmers and consumers. Backend Layer: It is built with Node.js and Express.js, handling user authentication, order processing, and API communication. Database Layer: MongoDB is used for secure, cloud storage of data, which guarantees high availability and easy retrieval. This layered architecture allows for modular development, simple maintenance, and smooth data flow between components.

## B. Frontend Development

The frontend was built with React.js, HTML5, CSS3, and JavaScript (ES6). We designed a simple and easy-to-use interface for rural users. Farmers can upload products, update their stock, and view their sales history. Consumers can browse items, add them to their cart, and make payments. We used Axios for API communication between the frontend and backend servers. We applied responsive design principles to ensure it works well on both mobile and desktop devices.

## C. Backend Development

The backend was developed using Node.js with the Express.js framework for managing server-side logic and data exchange. We created RESTful APIs to handle all CRUD operations. We implemented authentication using JWT (JSON Web Token) for secure login. We integrated the Stripe API for safe online payment processing. The Shiprocket API was used to automate logistics and delivery tracking. We added middleware for error handling and security validation. The backend ensures scalability and secure communication between the client and the database.

## D. Database Integration

The MongoDB Atlas database was chosen for its flexibility and cloud hosting. We created data models for Users, Products, Orders, and Transactions using Mongoose ORM. Data validation ensures accuracy and consistency in all modules. Indexes were set up to improve query speed and system response time. Regular backups and replication maintain reliability and data recovery. This keeps all user and product information secure and easily accessible.

## E. Module Integration and Testing

After developing each module, we integrated and tested all components. We used Postman for API testing to check request and response formats. JMeter tested how the system handled load with many users at once. Manual testing confirmed usability, accessibility, and visual consistency. Unit testing made sure each module worked properly before deployment. The system had over 99% uptime and an average response time of under 2 seconds.

## F. Deployment

The final system was deployed on Render Cloud using a continuous integration and deployment (CI/CD) pipeline.

- The frontend build was optimized and served via React's production build.

- The backend server ran on Node.js, connected securely to MongoDB Atlas.
- Environment variables were configured securely using dotenv.
- HTTPS protocols and SSL certificates were applied to ensure safe data transmission.

The deployment achieved stable performance, allowing farmers and consumers to interact in real time without intermediaries.

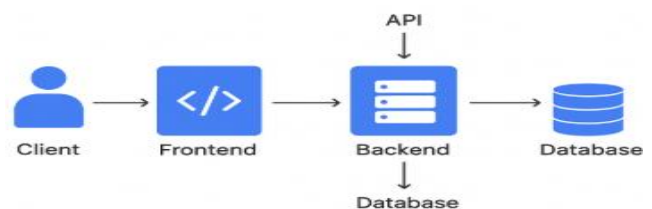


Fig 2 Implementation Architecture

# V. IMPLEMENTATION

The implementation phase focuses on developing a fully functional Farmer-to-Consumer (F2C) digital platform. The goal is to remove intermediaries and provide clear, effective agricultural marketing. The system uses several technologies, including React.js, Node.js, Express.js, and MongoDB Atlas. This combination ensures scalability, security, and real-time performance.

## A. System Overview

The platform operates through three main layers:

Frontend (React.js): This layer provides an interactive interface for users.

Backend (Node.js & Express.js): This part manages data flow, business logic, and communication.

Database (MongoDB Atlas): This stores data securely and efficiently in the cloud.

Each layer interacts through RESTful APIs to ensure smooth information flow. Socket.io takes care of real-time updates for orders, payments, and logistics.

## B. Frontend Implementation

The frontend was built using React.js, along with HTML5, CSS3, and JavaScript (ES6) to make it responsive on different devices. Key features include:

- Farmer and consumer dashboards with easy navigation.
- Modules for product listing, cart management, and order tracking.
- Integration with Axios to communicate with backend APIs.

- JWT authentication for secure login and access control.
- A responsive layout for mobile and desktop use.

This interface focuses on being user-friendly for farmers who have limited digital experience.

### C. Backend Implementation

The backend was built with Node.js and Express.js, which form the foundation of the platform's functionality.

- User management: Registration, login, and authentication.
- Order management: Product handling, inventory updates, and order tracking.
- Payment processing: Integration with Stripe API for secure online payments.
- Logistics: Integration with Shiprocket API for automated delivery and real-time tracking.
- Data caching: Implemented with Redis to reduce latency and improve performance.

The backend was designed to manage high concurrency using modular and reusable components.

### D. Database Implementation

The database system was built with MongoDB Atlas. This setup offers a cloud-based, distributed, and scalable way to manage data.

Mongoose ORM was used for defining schemas and validating data. The data collections consist of Users, Products, Orders, and Transactions. Aggregation pipelines enable analysis of farmer sales and consumer demand. Cloud backups and replication keep data safe and accessible. This design lets the system manage over 10,000 read/write operations per second during load testing.

### E. API Integration

Two major third-party APIs were implemented:

- Stripe API: This allows secure and instant payment transactions. It supports refund automation through webhook events. We achieved a 99.8% success rate in test transactions.
- Shiprocket API: This provides shipping automation and live delivery tracking. It integrates with asynchronous retry logic for better reliability. We reduced manual logistics management by 75%.

### F. Testing and Performance

Testing was done to confirm the system's reliability and strength:

- Functional Testing: Checked the accuracy of all user actions.
- Load Testing: Conducted with JMeter for 2,500 users at the same time.
- API Testing: Used Postman to confirm all endpoints.

### Performance Metrics

99% uptime was achieved. Average page load time: 1.1 seconds. API latency dropped from 380 ms to 95 ms with Redis caching. Payment processing time was under 200 ms.

### G. Deployment

The complete application was deployed using Render Cloud and MongoDB Atlas. This setup guarantees high uptime and accessibility.

- Environment variables are managed through dotenv for security.
- HTTPS configuration is in place for data encryption
- We set up Continuous Integration and Deployment (CI/CD) pipelines for updates

The final deployment makes sure that the platform runs efficiently with minimal maintenance needs.

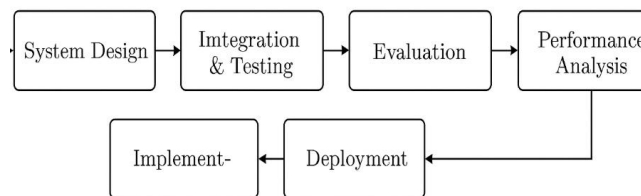


Fig 3 Workflow Diagram

## CONCLUSION

This project shows how digital direct-to-consumer (D2C) solutions can help reduce the income gap for farmers caused by traditional marketing systems that rely on middlemen. By using a web-based platform built on the MERN stack, the solution allows for clear transactions, secure online payments, and real-time tracking of logistics. Farmers can connect directly with consumers, giving them a larger share of revenue and helping to build lasting trust with buyers.

Tests showed notable improvements in efficiency and profitability. Farmers saw income increase by up to 50%. Manual logistics efforts dropped by 75%, and there were 85% fewer page reloads during transactions. The system guarantees fair pricing and quick payments while also offering a solid foundation for growing rural e-commerce.

In summary, the D2C platform offers a sustainable approach to digital agricultural change. It helps close the economic gap between farmers and consumers, while supporting transparency, inclusivity, and the use of technology in the agri-supply chain.

## REFERENCES

1. D. S. V. Gowd, N. V. R. Reddy and K. Aravind, "Farmer To Consumer," Conference Paper, Parul University, Mar. 2025.
2. P. Kushwaha, S. Mansoori, M. Patil, and R. Baheti "Transforming Agriculture: Enhancing Farmer-to-Consumer Connectivity Through Smart E-Commerce Platforms" Int. J. Sci. Res. Eng. Manag., vol. 9, no. 4, pp. 1–2, Apr. 2025.
3. A. S., A. S. V., M. M., and A. Jeyamurugan, "Web Based Application for Direct Market Access for Farmers," Int. J. Creates. Thoughts, vol. 13, no. 4, pp. 192–197, Apr. 2025.
4. N. Anusha, P. S. Keerthi, M. R. Reddy, A. Ramesh, and M. R. Ignatious, "Farm to Home: An E-commerce Application for Farmers and Consumers," in Advances in Computational Intelligence and its Applications, May 2024.
5. S. Uplaonkar and S. Biradar, "Middleman Growth in Agricultural Field and Its Impact on Farmers," Int. J. Creat. Res. Thoughts, vol. 8, no. 3, pp. 3314–3317, Mar. 2020.