

---

**EFFICIENT FARMER TO CONSUMER WEBAPP**

**Aryan S. Nipane<sup>\*1</sup>, Darshan B. Kadam<sup>\*2</sup>, Herschel P. Pawar<sup>\*3</sup>,  
Manish S. Thorat<sup>\*4</sup>, Madhuri Patil<sup>\*5</sup>**

<sup>\*1,2,3,4</sup>Student, Information Technology, MGM's College Of Engineering And Technology, Kamothe, Maharashtra, India.

<sup>\*5</sup>Professor, Information Technology, MGM's College Of Engineering And Technology, Kamothe, Maharashtra, India.

DOI : <https://www.doi.org/10.56726/IRJMETS51491>

---

**ABSTRACT**

This paper introduces a MERN stack-based web platform designed to connect farmers, consumers, and dealers, with the primary aim of facilitating direct exchanges of agricultural products. The platform enables farmers to list and promote their produce, consumers to browse and purchase locally sourced items, and dealers to market agricultural supplies. The website's core objective is to eliminate intermediaries in the agricultural supply chain, fostering more efficient and transparent transactions. By promoting direct sales and reducing overhead, this platform empowers farmers, benefits consumers with access to fresh local products, and offers dealers a dedicated agricultural clientele. The potential impact includes a more sustainable, equitable food system and improved agricultural sustainability. The website uses MERN stack to enable scalability, and responsiveness, while offering a user-friendly interface for all participants.

**Keywords:** MERN Stack, Serverless, Web Platform, Agriculture Marketplace, Local Sourcing, Sustainability

---

**I. INTRODUCTION**

The agricultural sector, a cornerstone of human civilization, is currently at the cusp of significant transformation, driven by the influence of technology and evolving consumer preferences. In recent years, there has been a discernible shift in how agricultural products reach consumers. The traditional model, characterized by a convoluted and often inefficient supply chain with multiple intermediaries, is making way for a more direct, digitally infused approach. This shift arises from the increasing demand for fresh, locally sourced products and a growing awareness of the environmental and economic benefits associated with reduced supply chain complexity.

In response to these evolving demands, a novel web platform has to emerge as a key player, leveraging the capabilities of modern web-stack. This platform has to address the critical challenges of bridging the gap between three integral players in the agricultural ecosystem: farmers, consumers, and dealers. By fostering direct interactions and transactions among these stakeholders, it aims to optimize the exchange of agricultural products while promoting transparency among these stakeholders. It has to optimize the exchange of agricultural products while promoting transparency, trust, and sustainability within the sector.

This paper aims to outline an online marketplace where farmers can showcase their produce and connect them directly with consumers. By doing so, it removes the aforementioned problem. This approach streamlines the agricultural supply chain and also gives the consumers access to fresh, locally sourced agricultural products. Furthermore, the platform extends its reach to dealers, who offer an assortment of essential agricultural supplies, including seeds, equipments, livestock, and dairy products.

The outline provides functionality using MERN stack since these components are reliable and battle-tested [1] [2]. They should all be hosted using serverless functions since it reduces the latency suffered by traditional server hosted applications [3][4].

**II. LITERATURE REVIEW**

Here is the summary of the various scholarly articles which were read.

**1. Digital Market: E-Commerce Application For Farmers (2018)** The paper proposes a digital market platform to integrate farmers, merchants/markets, government, and end-users in the agricultural sector. It addresses challenges faced by Indian farmers in obtaining fair profits and proposes solutions including

mobile-based Android and web-based Java applications for market access, analysis using KNN and Haversine algorithms, and government regulation enforcement. Challenges such as dynamic transportation tracking are identified.

2. **Research on Influencing Factors of Retail Sales in E-Commerce Market (2021)** The paper investigates factors influencing retail sales in China's e-commerce market, focusing on urban and rural residents' consumption expenditure. Through multiple linear regression analysis, it finds that urban and rural consumption expenditures significantly impact e-commerce retail sales, with rural spending exhibiting a positive influence.
3. **Research on E-commerce Development of Small and Medium-sized Enterprises Under the Background of Internet + (2021)** The research explores the development of e-commerce among small and medium-sized enterprises (SMEs) within the context of the "Internet +" environment. It highlights how SMEs can leverage the internet to access global markets, reduce transaction costs, and maintain competitive pricing.
4. **Node.js: Using JavaScript to Build High-Performance Network Programs (2010)** Node.js is presented as a framework for developing high-performance, concurrent programs in the server-side JavaScript space. It utilizes asynchronous I/O with an event-driven programming model, rather than traditional multithreading, to build efficient network programs.
5. **MongoDB scheme analysis (2017)** The paper aims to create a web administration interface for analyzing data stored in MongoDB databases. It discusses the design of a tool comprising a console application for analysis and a web application for visualization, highlighting its speed, features, and testing.
6. **Webapp Service for Booking Handyman Using MongoDB, Express JS, React JS, Node JS (2021)** The paper describes a web application facilitating booking of handyman services using MongoDB, Express JS, React JS, and Node JS. It enables users to book workers conveniently, showcasing their skills and availability. The use of React JS ensures speed, productivity, and SEO friendliness, while MongoDB's scalability simplifies data management.
7. **Improving Web Development Process of MERN Stack (2021)** The thesis discusses enhancements to the web development process of the MERN stack. It covers the frontend built with React and Next.js, data management with Apollo and GraphQL, and the backend utilizing KeystoneJS and MongoDB. The aim is to customize development for specialized applications.
8. **Full-Stack vs MEAN Stack vs MERN Stack: The Right Technology Stack for You in 2024 (2023)** An online article compares full-stack, MEAN stack, and MERN stack development approaches, considering their suitability for different projects in 2024. It provides insights into choosing the appropriate technology stack for web development needs.
9. **The rise of serverless computing (2019)** The article discusses the emergence of serverless computing as a paradigm shift in application development. It explores the benefits and implications of serverless architecture, highlighting its potential to revolutionize traditional server-based models.
10. **How can serverless computing improve performance? | Lambda performance (2024)** Cloudflare presents an article discussing how serverless computing, particularly Lambda performance, can enhance application performance. It explores the advantages of serverless architectures and their impact on performance optimization.

### III. METHODOLOGY

#### Architectural Design

The architectural design of the Web Platform is structured around seamless user experience, robust backend functionality, and efficient media management.

The core components include:

- **Client-side Application:** Represented by the React Single Page Application (SPA), this component provides users with an intuitive interface accessible through web browsers.
- **Cloudflare Functions:** Serving as the backend, these functions, developed in Node.js, handle requests from the client, interact with databases, and manage external services.

- MongoDB Database: Hosted on MongoDB Atlas, this cloud-based database stores and retrieves various types of data, including user profiles, product listings, and transaction records.
- Cloudinary: This service facilitates media storage and delivery, enabling seamless uploading, fetching, and optimization of images, videos, and other media files.
- Twilio: This service allows us to send OTP to the user for secure passwordless login.

### **Development**

The web platform involves implementing using MERN stack.

Using React.js we are able to create a dynamic and responsive user interface which caters to the diverse needs of our stakeholders. On the backend, Express.js and Node.js enable us to build an efficient server-side infrastructure, while MongoDB serves as the database, offering a flexible and scalable solution.

Furthermore, we embrace serverless architecture by using Cloudflare Serverless Functions for hosting our codebase and serving it on the edge. This reduces the latency, leading to a better user experience and performative application.

### **Features and Functionality**

The Farmer's Dashboard serves as a comprehensive platform for farmers, allowing easy listing and promotion of agricultural produce while facilitating direct communication with consumers for transparent transactions. The Consumer's Experience prioritizes seamless browsing and purchasing of locally sourced agricultural products, supplemented by valuable information on the products' local sources to promote sustainability. In the Seller's Marketplace, a dedicated space empowers dealers to market and sell various agricultural supplies, from seeds and equipment to livestock and dairy products. This integrated approach aims to connect farmers, consumers, and dealers efficiently, fostering a transparent and sustainable agricultural ecosystem.

### **Sustainability and Local Sourcing**

In the pursuit of sustainability and local sourcing, our platform plays a pivotal role in revolutionizing agricultural practices. By eliminating intermediaries and reducing supply chain complexity, the platform actively contributes to a more environmentally friendly and sustainable food system. Emphasizing the importance of local sourcing, the platform provides consumers with valuable information about the origin of products, fostering a deeper connection between consumers and local farmers.

## **IV. MODELING AND ANALYSIS**

### **System Architecture Modeling**

In our project, we used system architecture modeling techniques using PlantUML and diagrams.net (formerly draw.io) to create various UML diagrams. These diagrams include architecture, data flow, use case, and entity-relationship (ER) diagrams, enabling us to visualize and communicate the structure and behavior of our system effectively.

Our approach to system architecture modeling prioritized considerations such as scalability, security, and robustness. By incorporating these design principles into our diagrams, we ensured that our system is capable of accommodating growth and maintaining robust security measures in the cloud environment.

Through the use of tools like PlantUML and diagrams.net, we were able to create detailed and insightful representations of our system's architecture and functionality, laying a solid foundation for development and implementation. This modeling process facilitated clear communication and provided valuable insights into the our system's design.

### **Security Analysis**

In our system architecture, security is paramount to safeguard user data and ensure the integrity of system operations. We leverage MongoDB as our database solution, known for its robust security features such as authentication, access control, and encryption at rest. Additionally, Cloudflare's serverless functions power our server-side logic, providing scalability and agility without compromising security. To enhance security further, we implement encryption for data transmission between the user's PC and the server, utilizing HTTPS protocols. We enforce strict authentication mechanisms to verify user identities and limit access rights based on role-based permissions.

By combining these measures, we maintain a robust security posture, ensuring the confidentiality, integrity, and availability of our system and user data.

### Performance Evaluation:

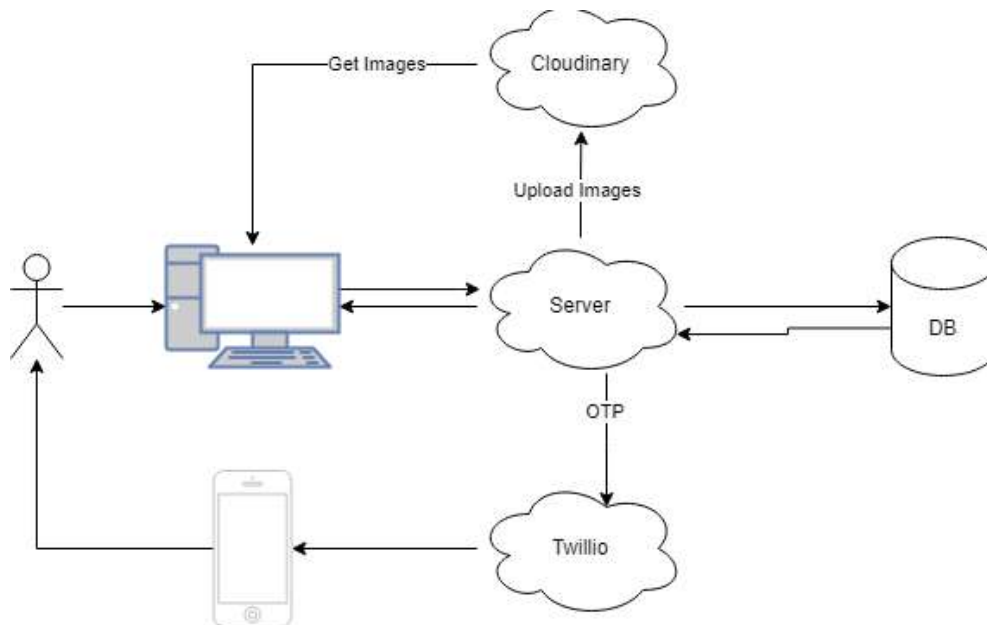
Our system architecture undergoes regular performance evaluations to ensure optimal functionality and responsiveness. We closely monitor key metrics such as response time, throughput, scalability, and resource utilization. By analyzing these metrics, we can identify areas for improvement and optimize the system's performance to deliver a seamless user experience. Through continuous testing and optimization, we strive to maintain high levels of efficiency and scalability, ensuring that our system can effectively handle varying levels of demand while maximizing resource utilization.

### Hosting Technology

Choosing the correct technology for hosting our application was a paramount decision since we need to ensure the scalability, reliability, security, and availability of our system.

After careful consideration of various hosting options, including on-premise servers, renting virtual private clouds (VPCs), and major cloud providers like AWS and GCP, we ultimately opted for Cloudflare Workers. Cloudflare's serverless platform offers a multitude of benefits, including automatic scaling, a high-performance global network, support for multiple programming languages, exceptionally low costs, and a serverless architecture that eliminates the need for managing servers.

### System Architecture



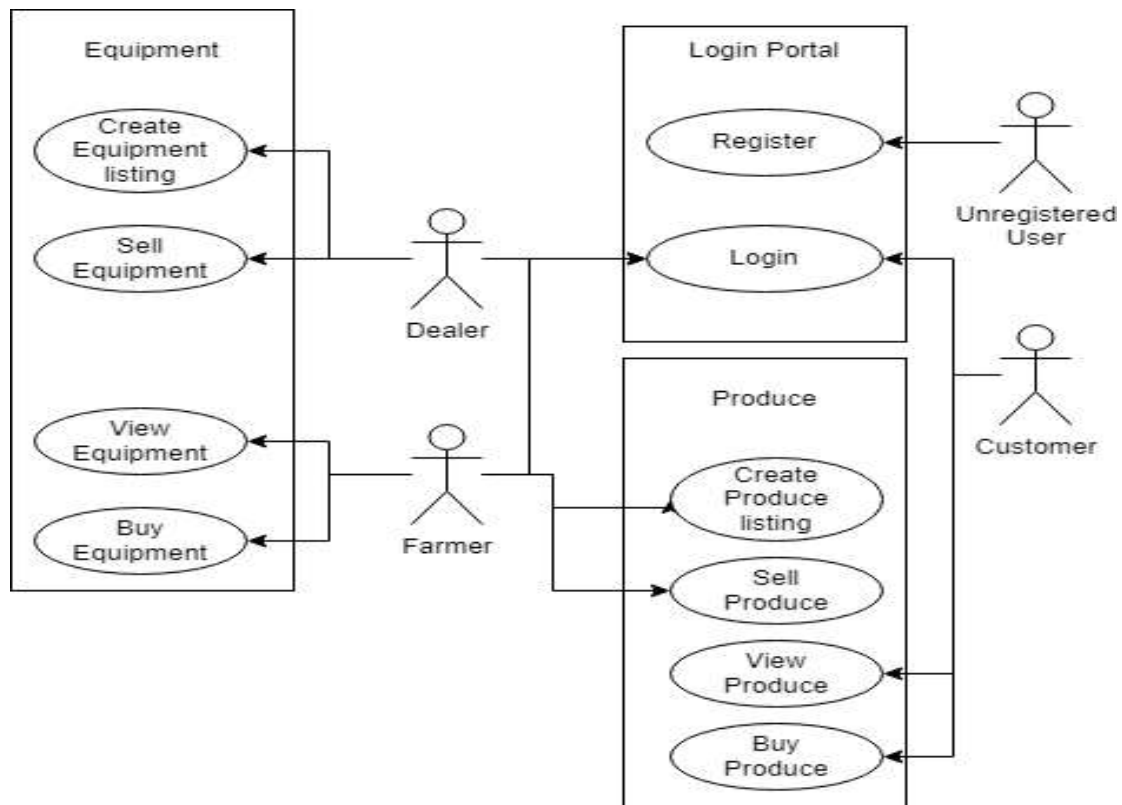
**Figure 1:** Architecture for the system

The system architecture depicted in the diagram involves multiple actors collaborating to facilitate user interactions and data management processes:

- **User:** The primary actor initiating interactions with the system via a personal computer (PC). Users perform actions such as accessing services and submitting requests.
- **PC:** Serves as the interface through which users interact with the system. Users initiate actions on their PCs, which are then processed by the server.
- **Phone:** Used for logging into the system via one time password (OTP)
- **Server:** Acts as an intermediary between the user and various backend services. It receives requests from the PC, processes them, and interacts with other components to fulfill the requested actions.
- **Database (DB):** Stores and manages structured data related to products, user registrations, and equipment. The server communicates with the database to perform operations such as adding and retrieving data.

- Twilio: A third-party service utilized for communication purposes. The server interacts with Twilio's API to send one-time passwords (OTPs) to users' phones for authentication or verification purposes.
- Cloudinary: A cloud-based media management platform utilized for handling images. The server interacts with Cloudinary to upload and manage visual content such as product images.

#### Use case



**Figure 2:** Use case for the system

The use case diagram depicts the interactions between various actors and the system for a farming market webapp. The actors involved in the system are:

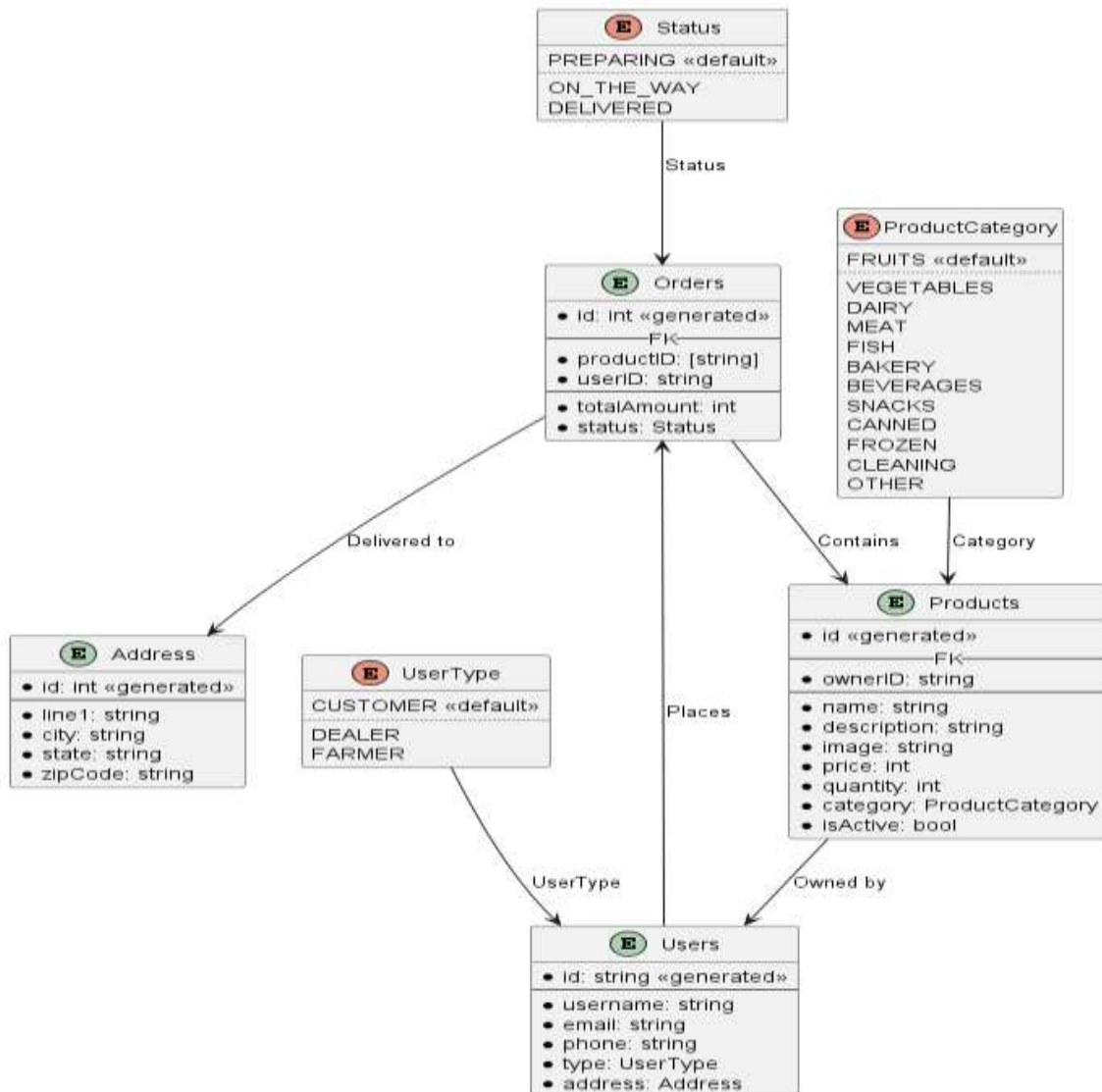
- Farmer: A user who produces crops or other agricultural products.
- Dealer: A user who sells farming equipment, seeds, fertilizer, pesticides, etc to the farmer.
- Customer: A user who buys produce directly from farmers.
- Unregistered User: A potential user who has not yet registered with the platform.

The main functionalities of the system include:

- Register: Unregistered users can register to become members of the platform.
- Login: Farmers, dealers, and customers need to log in to access the platform.
- View and Buy Produce: Customers can view and purchase produce from farmers.
- View and Buy Equipment: Farmers can view and purchase available equipment.
- Create and Sell Produce/Equipment: Farmers can list their produce for sale, and dealers can list their equipment for sale.



## ER Diagram



**Figure 3:** ER Diagram for the system

### 1) Entities:

#### a) Orders:

i) Represents orders placed by users. Each order has a unique identifier (id) and contains information about the products included, total amount, and status.

#### b) Address:

i) Represents delivery addresses associated with orders. Each address has a unique identifier (id) and includes details like line 1, city, state, and ZIP code.

#### c) Users:

i) Represents users of the system. Each user has a unique identifier (id) and includes attributes such as username, email, phone, and type (e.g., customer, dealer). Users are associated with addresses.

#### d) Products:

i) Represents products available for purchase. Each product has a unique identifier (id) and includes details like name, description, price, quantity, category, and ownership information.

### 2) Relationships:

#### a) Places:

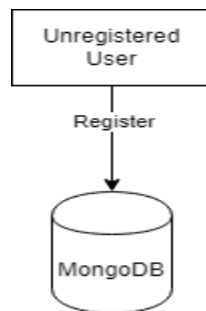
i) Indicates that users place orders. Each order is associated with a user.

#### b) Contains:

- i) Indicates that orders contain products. Each order can include multiple products.
- c) Owned by:
  - i) Indicates that products are owned by users. Each product is associated with a user.
- d) Delivered to:
  - i) Indicates the delivery address for orders. Each order is associated with an address.
- 3) Enumerations:
  - a) Status:
    - i) Enumerates the various statuses an order can have, such as "Preparing," "On the Way," and "Delivered."
  - b) UserType:
    - i) Enumerates the different types of users, such as "Customer," "Dealer," and "Farmer."
  - c) ProductCategory:
    - i) Enumerates the categories products can belong to, such as "Fruits," "Vegetables," etc.

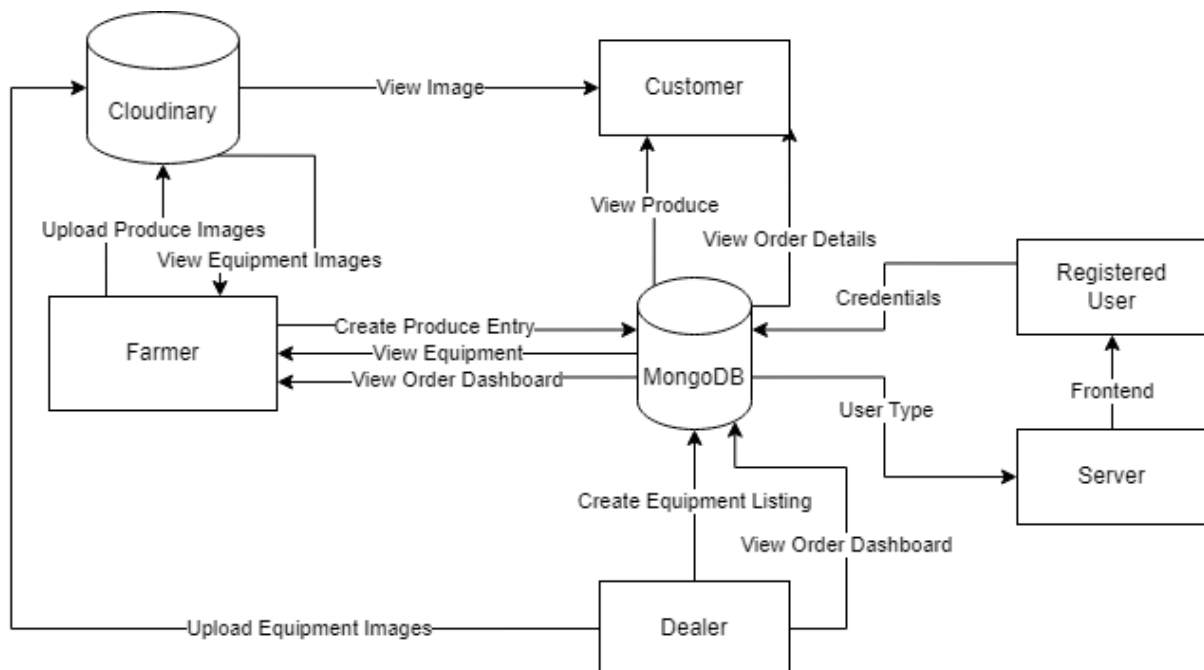
This ER diagram represents the relationships between users, orders, addresses, and products in a system for managing orders and deliveries. It shows how users place orders, the products included in orders, the addresses for delivery, and the types of users and products available.

#### Data Flow Diagram



**Figure 4:** Data Flow for an unregistered user

An unregistered user can only register to the system and thus the data only flows from them to the MongoDB instance.



**Figure 5:** Data Flow for the system

- 1) MongoDB
- a) Represents the database system.

- b) Plays a central role in the system since majority of the data is stored in the database
- 2) Cloudinary (Cloud Storage)
  - a) Stores the images uploaded by farmers (for produce) and dealers (for equipment)
  - b) The images are then viewed by farmers (for equipment) and customers (for produce)
- 3) Twillio (SMS Service)
  - a) Sends the OTP to the user for logging into the system

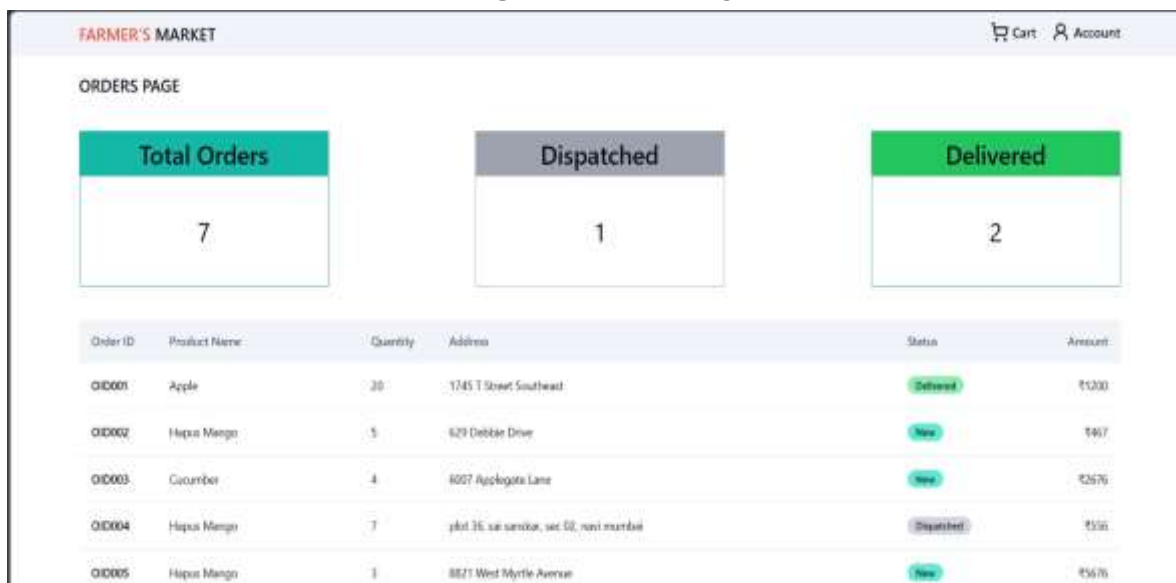
A registered user will enter their credentials and the server will check the credentials in the database, the database responds if the credentials are valid and what their role is. The server then proceeds to show the appropriate pages to the user.

## V. RESULTS AND DISCUSSION

The MERN stack-based web platform successfully addressed the challenges of connecting farmers, consumers, and dealers, facilitating direct exchanges of agricultural products. Farmers were able to efficiently list and promote their produce, while consumers enjoyed a seamless experience browsing and purchasing locally sourced items. dealers found an effective avenue to market their agricultural supplies. The platform's core objective of eliminating intermediaries in the agricultural supply chain was achieved, leading to more efficient and transparent transactions. The use of the MERN stack and serverless architecture contributed to the platform's scalability and responsiveness, enhancing the overall user experience.



**Image 1: Order Cart Page**



**Image 2: Farmer Orders Dashboard**



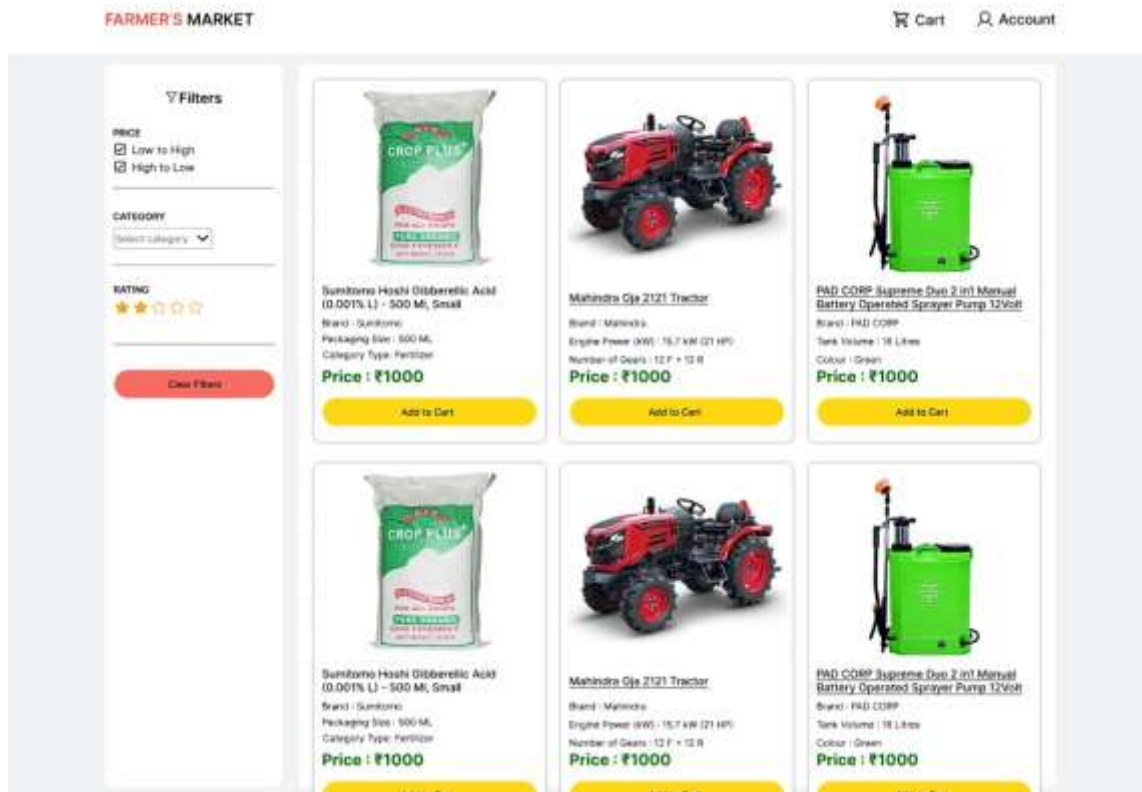


Image 3: Browsing Products from Dealer

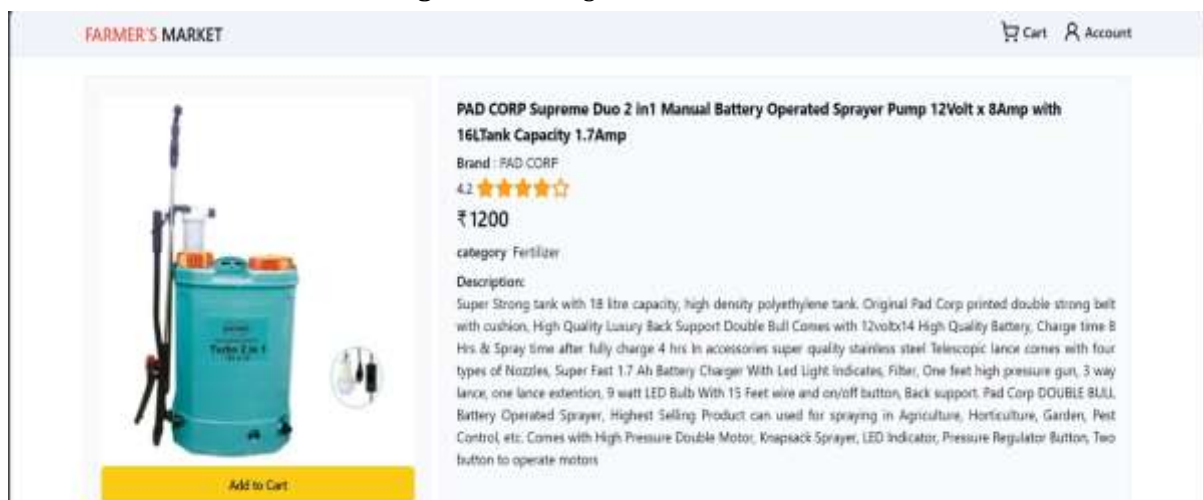


Image 4: Product Page

## VI. CONCLUSION

In conclusion, this conceptual exploration delves into the potential of a MERN stack-based agricultural platform, emphasizing its transformative role in addressing challenges within the traditional supply chain. The platform envisions connecting farmers, consumers, and dealers directly, streamlining the exchange of agricultural products. By empowering farmers to showcase their produce and providing consumers with easy access to locally sourced items, the platform eliminates intermediaries, fostering efficiency and transparency. The adoption of the MERN stack ensures scalability and responsiveness, enhancing the user experience. This conceptual study illuminates the promising possibilities of modern web technologies in reshaping the dynamics of the agricultural sector, offering a vision of a more direct, user-centric, and sustainable approach to agricultural transactions.

## VII. REFERENCES

- [1] S. Tilkov and S. Vinoski, "Node.js: Using JavaScript to Build High-Performance Network Programs", IEEE Internet Computing, vol. 14, no. 6, pp. 80–83, Nov. 2010, doi: 10.1109/MIC.2010.145.
- [2] K. Saundariya, M. Abirami, K. R. Senthil, D. Prabakaran, B. Srimathi, and G. Nagarajan, "Webapp Service for Booking Handyman Using Mongodb, Express JS, React JS, Node JS", in 2021 3rd International Conference on Signal Processing and Communication (ICPSC), May 2021, pp. 180–183. doi: 10.1109/ICSPC51351.2021.9451783.
- [3] Cloudflare, Inc., "How can serverless computing improve performance? | Lambda performance", 2024, [Online]. Available: <https://www.cloudflare.com/learning/serverless/serverless-performance/>
- [4] P. Castro, V. Ishakian, V. Muthusamy, and A. Slominski, "The rise of serverless computing", Commun. ACM, vol. 62, no. 12, p. 44, Nov. 2019, doi: 10.1145/3368454.
- [5] Bhende, Manisha and Avatade, Mohini S. and Patil, Suvarna and Mishra, Pooja and Prasad, Pooja and Shewalkar, Shubham, "Digital Market: E-Commerce Application For Farmers", 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), pp. 1-7, Aug. 2018, doi: 10.1109/ICCUBEA.2018.8697615.
- [6] Y. Pei, W. Xue, Y. Su, and D. Li, "Discussion on influence factors and evaluation of customer experience for B2C E-commerce enterprises," in 2015 International Conference on Logistics, Informatics and Service Sciences (LISS), July 2015, pp. 1–5. doi: 10.1109/LISS.2015.7369651.
- [7] J. Qidi, "Research on Influencing Factors of Retail Sales in E-Commerce Market," in 2021 2nd International Conference on E-Commerce and Internet Technology (ECIT), March 2021, pp. 16–19. doi: 10.1109/ECIT52743.2021.00011.
- [8] W. Ma and D. Liu, "Research on E-commerce Development of Small and Medium-sized Enterprises Under the Background of Internet +," in 2021 2nd International Conference on E-Commerce and Internet Technology (ECIT), March 2021, pp. 181–184. doi: 10.1109/ECIT52743.2021.00048.
- [9] J. Huijing, Y. Xuefeng, P. Wenxue, G. Longwei, F. Linfeng, S. Yongbo, and W. ping, "Practical Exploration of Rural E-commerce Boosting Rural Revitalization Based on 4C Model," in 2021 2nd International Conference on E-Commerce and Internet Technology (ECIT), March 2021, pp. 74–78. doi: 10.1109/ECIT52743.2021.00023.
- [10] M. Rao, A. Singh, N. V. Subba Reddy, and D. Acharya, "Crop prediction using machine learning," Journal of Physics: Conference Series, vol. 2161, Jan. 2022, p. 012033. doi: 10.1088/1742-6596/2161/1/012033.
- [11] A. Motwani, P. Patil, V. Nagaria, S. Verma, and S. Ghane, "Soil Analysis and Crop Recommendation using Machine Learning," in 2022 International Conference for Advancement in Technology (ICONAT), Jan. 2022, pp. 1–7. doi: 10.1109/ICONAT53423.2022.9725901.
- [12] L. Vokorokos, M. Uchnár, and A. Baláž, "MongoDB scheme analysis," in 2017 IEEE 21st International Conference on Intelligent Engineering Systems (INES), Oct. 2017, pp. 67–70. doi: 10.1109/INES.2017.8118530.
- [13] B. Nguyen, "Improving Web Development Process of MERN Stack," Bachelor's Thesis, Metropolia University of Applied Sciences, May 2021, pp. 52. Available: <https://www.theseus.fi/handle/10024/498420>.
- [14] M. Gadhavi, "Full-Stack vs MEAN Stack vs MERN Stack: The Right Technology Stack for You in 2024," Radixweb, Dec. 18, 2023. [Online]. Available: <https://radixweb.com/blog/full-stack-vs-mean-stack-vs-mern-stack-development>.