
TRIBHUVAN UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY
CENTRAL CAMPUS OF TECHNOLOGY (CCT), HATTISAR, DHARAN

*A Project Proposal On*

# DOKO: A Smart Agro-based Marketplace

Submitted To:
DEPARTMENT OF CSIT/IT
CENTRAL CAMPUS OF TECHNOLOGY (CCT), HATTISAR, DHARAN, SUNSARI, KOSHI, NEPAL

*In partial fulfilment of the requirements for the Bachelor's of Science in Computer Science and Information Technology (B.Sc. CSIT)*

Submitted By:
**Chhyosang Lepcha** (29591)
**Anjil Khamlung Rai** (29586)
**Sanshika Lawati** (29610)

June, 2025

# TABLE OF CONTENTS

# List of Figures

# List of Tables

# 1. Introduction

Agriculture remains the backbone of Nepal's economy, employing over 60 percentage of the population and contributing significantly to the national GDP. Despite its vital role, the sector continues to face long-standing challenges such as limited market access, poor price transparency, reliance on intermediaries, and a lack of digital integration. These constraints have suppressed farmers' income potential and stunted the development of a sustainable agricultural ecosystem [1].

To tackle these challenges, DOKO: A Smart Agro-Based Marketplace is proposed as an innovative digital solution tailored to the Nepali context. DOKO aims to create an inclusive, user-friendly platform that directly connects farmers, agro-vendors, and buyers—eliminating unnecessary middlemen and fostering transparent, fair, and efficient trade.

The platform will include features customized for local needs, such as mobile accessibility, and integration with popular digital payment systems like eSewa and Khalti. By lowering the technical barriers to entry, DOKO enables farmers from diverse backgrounds to participate actively in the digital economy.

This initiative supports Nepal's broader agenda of agricultural digitization, rural livelihood improvement, and food security enhancement. By streamlining the supply chain and empowering stakeholders at every level, DOKO aims to make agriculture a more rewarding, resilient, and future-ready sector in Nepal. [2]

# 2. Problem Statement

i. Limited Market Access for Farmers

Most farmers in Nepal sell their produce through local traders or middlemen, often at low prices. They have few options to access larger or better-paying markets, reducing their potential income.

ii. Lack of Real-Time Information

· Farmers often do not have access to updated information on:

· Market prices

· Crop demand

· Weather conditions

Best farming practices This leads to poor planning, overproduction or underproduction, and financial losses.

iii. Overdependence on Middlemen

Middlemen often control pricing and market access. This reduces transparency and takes a large share of the profit, leaving farmers with unfair compensation for their hard work.

iv. Lack of Transparency in Transactions

There is little trust between buyers and sellers due to the absence of digital records or verified sources. This affects fair trade, product traceability, and accountability.

v. Limited Use of Technology in Agriculture

Digital tools and platforms are rarely used in farming communities, especially in remote areas. This creates a gap in innovation, planning, and smart market connections. [3]

## 3. Objectives

i. To Connect Farmers Directly with Buyers

DOKO aims to build a digital platform where farmers can sell their products directly to buyers such as wholesalers, retailers, and consumers. This helps eliminate middlemen and ensures farmers receive fair prices.

ii. To Provide Real-Time Market Information

The platform will offer live updates on crop prices, demand, weather forecasts, and farming tips. This will help farmers make better decisions about what to grow, when to harvest, and where to sell.

iii. To Support Smallholder Farmers

The project will empower small and rural farmers by giving them access to larger markets, digital payments, and agricultural services. This can improve their income and strengthen

their role in the economy.

iv. To Ensure Transparency and Trust

By using digital records and product tracking, DOKO will promote fair trade and honest transactions. Buyers and sellers will be able to view each other's ratings and transaction history.

# 4. Methodology

Software Development Approach: Iterative Waterfall Model For the development of the DOKO agro-based marketplace, we will follow the Iterative Waterfall Software Development Life Cycle (SDLC). This model is a structured and phased approach that allows for systematic progress through well-defined stages: requirements, design, development, testing, deployment, and maintenance. Unlike the traditional waterfall model, the iterative version enables feedback and refinements at the end of each stage, allowing for improvements without starting from scratch.

## 4.1. Requirement identification

**Iterative Waterfall SDLC for DOKO Project**

**Overview:**

The Iterative Waterfall model follows a sequential flow (like the traditional waterfall), but each phase is revisited iteratively to refine and improve the system based on feedback or testing at every stage. [4]

**i. Requirement Analysis**

Goal: Understand what users (farmers, buyers, transporters, and admin) need.

Activities:

Identify functional and non-functional requirements.

Define user roles and their actions.

Conduct stakeholder meetings or surveys.

Output: Software Requirement Specification (SRS) document.

**ii. System Design**

Goal: Plan how the system will work.

Activities:

Design architecture using the MERN stack.

Create flowcharts, data models, and wireframes.

Choose database structure (MongoDB collections for users, products, orders, etc.).

Output: High-level system architecture and database design.

**iii. Implementation (Coding)**

Goal: Develop the actual system.

Activities:

·Build front-end with React.js.

·Develop backend APIs with Node.js and Express.js.

·Connect to MongoDB for data storage.

·Integrate payment APIs (eSewa, Khalti).

·Output: Working modules (e.g., registration, product listing, payment, delivery tracking).

**iv. Testing**

·Goal: Ensure each part of the system works properly.

Activities:

·Unit testing of frontend and backend.

·Integration testing (e.g., buyer placing order and farmer being notified).

·Fix bugs and improve system based on feedback.

·Output: Bug-free and stable version of the platform.

**v. Deployment**

·Goal: Launch the platform for use.

Activities:

Deploy the web app to a server (e.g., Heroku, Vercel, AWS).

Set up domain and SSL for secure access. Make it live for end-users.

Output: Live, accessible web-based marketplace.

vi. **Maintenance and Iteration**

Goal: Fix issues, upgrade features, and enhance performance.

Activities:

Monitor platform health (admin dashboard).

Gather user feedback for future iterations.

Push regular updates with improvements or new features.

Output: Continuously improved platform with better performance and user experience.

**Why Iterative Waterfall Fits DOKO:**

Allows feedback and improvements at each stage.

Helps adapt to changes in agricultural market needs.

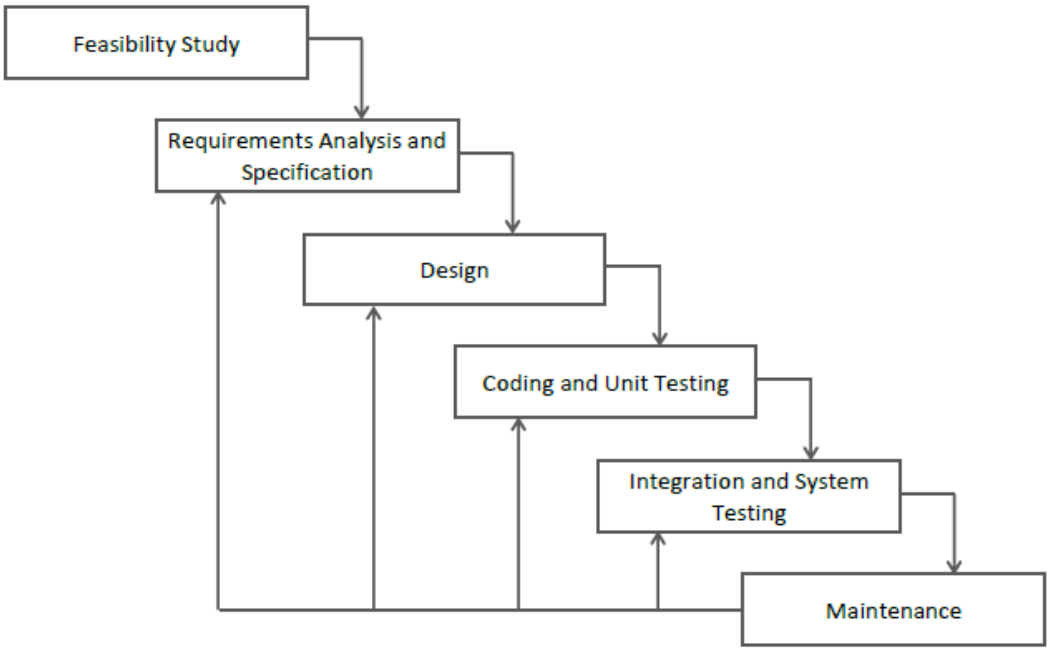


Figure 4.1: Phases of Iterative Waterfall Model

### 4.1.1. Study of Existing System / Literature Review

Nepal's agricultural sector is progressively integrating digital platforms to enhance market access, transparency, and efficiency. Several initiatives have been launched to connect farmers, buyers, and other stakeholders through online marketplaces and information systems but most popular are listed below:-

· **Smart Krishi** is Nepal's first digital agriculture platform designed to empower farmers with accessible, practical, and real-time agricultural information. Launched in 2015, the platform leverages mobile and internet technology to provide guidance on crop cultivation, livestock management, weather updates, market prices, and expert consultations. Through its mobile app and online presence, Smart Krishi aims to bridge the information gap in rural farming communities, promote sustainable agricultural practices, and enhance farmers' income by connecting them directly to markets and resources. [5]

· **AgroMart Nepal** is a leading digital agricultural marketplace founded in 2013 in Kathmandu. It connects farmers, gardeners, and agribusinesses with quality agricultural products such as seeds, fertilizers, and grafted fruit plants. The platform supports sustainable farming by providing technical training, expert advice, and easy online access to farming inputs. AgroMart promotes commercial farming, improves market access, and empowers Nepalese farmers with modern tools and knowledge, helping to boost productivity and income across the country.

· **Agrobase Nepal** is Nepal's first dedicated digital agricultural marketplace and e-commerce platform designed to connect farmers with a broad network of suppliers, including agrovets, seed companies, fertilizer distributors, and equipment providers. The platform enables seamless buying and selling of agricultural inputs such as seeds, fertilizers, pesticides, tools, and machinery from multiple vendors across Nepal. [6]

### 4.1.2. Requirement Analysis

#### Functional Requirements

These describe what the system should do — the core functions and features of the DOKO platform.

#### i. User Registration and Authentication

· Farmers, buyers, and admins must be able to create accounts and securely log in (via OTP,

password, or biometrics).

**ii. Product Listing and Management**

· Farmers can upload details of their produce including quantity, quality, price, and expected delivery date.

· Buyers can browse, search, and filter produce listings.

**iii. Real-Time Market Information**

· The system must provide live updates on crop prices, demand trends, and weather forecasts relevant to users' locations.

· Order Placement and Tracking

· Buyers can place orders directly on the platform.

· Both parties can track order status and delivery schedules.

**iv. Payment Processing**

· The platform supports secure digital payments (mobile wallets, bank transfers) and manages transaction records.

· Option for cash-on-delivery where digital payments are not feasible.

**v. Ratings and Feedback System**

· Buyers and sellers can rate each other and provide feedback to improve transparency and trust.

**vi. Logistics Coordination**

· Integration with transport partners to schedule pickups and deliveries.

· Notifications to users about logistics status.

**vii. Admin Dashboard and Reporting**

· Administrators can monitor user activity, manage listings, handle disputes, and generate reports.

**Non-Functional Requirements**

These describe how the system performs and the qualities it must have.

**i. Usability**

· The platform should have an intuitive, easy-to-use interface accessible even to users with limited digital literacy.

**ii. Performance and Scalability**

· The system should handle large numbers of simultaneous users without slowing down.

· It must scale as more farmers and buyers join over time.

**iii. Reliability and Availability**

· The platform should be available 24/7 with minimal downtime.

· Offline or SMS-based fallback options should be considered for areas with poor connectivity.

**iv. Security**

· User data and transactions must be protected with strong encryption and secure authentication methods.

· The system should comply with data privacy laws.

**v. Maintainability and Extensibility**

· The software architecture should allow for easy updates, bug fixes, and addition of new features.

**vi. Accessibility**

· The platform should be accessible on low-end smartphones and support low bandwidth conditions.

**Hardware Requirements**

i. For Developers Laptop: i5/Ryzen 5, 8GB+ RAM

Phones: Android 10+ and iOS 14+ (for testing)

Internet: Stable 10+ Mbps

ii. For Servers Cloud: AWS/Google Cloud (2CPU, 4GB RAM)

Database: MongoDB Atlas (10GB+)

iii. For Farmers/Buyers Phone: Basic Android 8+/iOS 12+

Internet: 3G/4G or WiFi

## 4.2. Feasibility Study

Feasibility study is research/study which is to see if the system on completion will serve the purpose of the organization for work, effort and the time that it is spent on it. Feasibility study lets developers see the future of the system and the usefulness. A feasibility study of the system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources.

### 4.2.1. Technical Feasibility

The DOKO platform will be built using the MERN stack, a modern and popular JavaScript-based technology stack that is efficient for building full-stack web applications.

**Technology Stack: MERN Stack**

MongoDB: NoSQL database used for flexible and scalable data storage (e.g., products, users, transactions).

Express.js: Backend framework for building RESTful APIs and handling server-side logic.

React.js: Frontend library for building dynamic and responsive user interfaces.

Node.js: JavaScript runtime for executing backend code and enabling full-stack JavaScript development. [7]

**Frontend Design**

Built using React.js with Tailwind CSS for responsive, mobile-first UI.

Supports single-page application (SPA) structure for fast performance.

Optimized for low-bandwidth rural environments.

**Backend Functionality**

Node.js with Express.js handles user management, API routing, product listings, and transaction logic.

RESTful API design ensures modularity and scalability of system components.

**Database System**

MongoDB Atlas provides cloud-hosted, schema-less database for storing unstructured and semi-structured data.

Enables scalability, data indexing, and real-time updates.

**Security Measures**

bcrypt: Used for secure password hashing with salt to prevent credential theft.

JWT (JSON Web Tokens): For stateless, token-based user authentication and secure session management.

Additional Security: Input validation, API rate limiting, CSRF protection, and XSS prevention.

**Payment Gateway Integration**

Integration with Nepali digital wallets: eSewa and Khalti.

Supports secure transactions and includes cash-on-delivery as a fallback option.

Uses encrypted payment APIs and real-time transaction confirmation.

**Deployment and Hosting**

Frontend Hosting: Deployed on Vercel with CI/CD support.

Backend Hosting: Hosted on Render or AWS, depending on cost and scaling requirements.

Database Hosting: Handled by MongoDB Atlas, offering performance monitoring and backups.

**Performance and Accessibility**

Built as a Progressive Web App (PWA) to enable offline access and push notifications.

Optimized for 2G/3G network environments common in rural Nepal.

Accessible from low-end smartphones and basic browsers.

**Scalability and Maintainability**

Modular architecture allows easy addition of features (e.g., mobile app, analytics).

Use of reusable React components and standard coding practices improves maintainability.

Designed to support growth in user base and data volume without major redesign.

**Monitoring and Logging**

Integration of monitoring tools (e.g., Sentry, LogRocket) for real-time error tracking and system diagnostics.

Admin dashboard includes basic analytics, uptime monitoring, and activity logs.

### 4.2.2. Operational Feasibility

This study helps to provide enough support for users. The system is developed using simple and widely available technologies, which makes it easy to maintain and run. It is working well during development and is expected to perform reliably after deployment. The proposed DOKO system is operationally feasible for the following reasons:

· Users benefit more as most of their time is saved by avoiding physical market visits and middlemen.

· The system serves users at their own place—farmers, buyers, and sellers can use it from home or farm using a mobile phone or computer.

· The cost of the system is very low when compared to the value and benefits it provides in terms of market access, transparency, and convenience.

· Cooperatives and local organizations can easily support users with training and guidance.

· No advanced technical skills are needed to use the platform, as it is designed to be simple and user-friendly. [3]

### 4.2.3. Economic

Conduct a cost-benefit analysis to determine the economic feasibility of your project. Consider factors such as development costs, maintenance costs, and potential benefits or savings.

Table 4.1: sample Cost-Benefit Analysis of the Proposed Project

| **Item** | **Description** | **Cost ($)** | **Benefit ($)** |
|---|---|---|---|
| Development Costs | Software Development | 50,000 | - |
| Hardware Costs | Servers and Equipment | 70,000 | - |
| Training Costs | User Training Sessions | 10,000 | - |
| Maintenance Costs | Annual Maintenance | 5,000 | - |
| **Total Costs** | | **1,35,000** | - |
| Increased Efficiency | Time Savings | - | 50,000 |
| Improved User Satisfaction | User Feedback | - | 50,000 |
| Revenue Increase | New Customers | - | 1,00,000 |
| **Total Benefits** | | - | **2,00,000** |
| **Net Benefit** | | **65,000** | **65,000** |

Analyze the cost-effectiveness of the project. Consider the budget, expected benefits, and potential return on investment. Provide a cost-benefit analysis to justify the project's financial viability.

### 4.2.4. Schedule Feasibility(Gantt chart showing the project timeline)

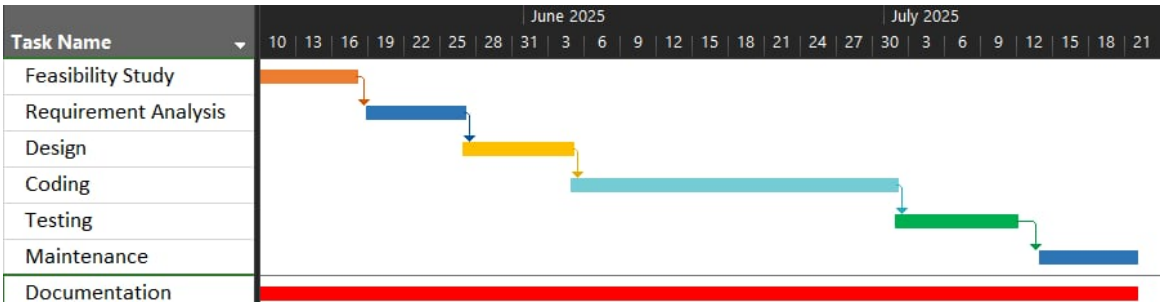


Figure 4.2: Gantt Chart Demonstrating Schedule Feasibility

## 4.3. High-Level Design of System

The proposed DOKO platform is a web-based smart marketplace designed to connect farmers, buyers, and transporters through a streamlined and interactive interface. The high-level

design includes the system architecture, user roles, data flow, and technologies used.

The system will follow a modular, component-based architecture that separates the frontend, backend, and database layers, enabling maintainability, scalability, and future upgrades.

### 4.3.1. Methodology of the proposed system

- Development Approach: Object-Oriented Methodology The development of DOKO will follow the Object-Oriented Methodology (OOM) because it aligns well with JavaScript-based frameworks like React and Node.js used in the MERN stack. This approach enables:
  - Reusability through components and modules
  - Better data encapsulation
  - Easier maintenance and scaling
  - A clear structure for managing users, products, orders, and transactions
- Techniques and Tools Phase Approach/Technique Requirement Analysis Interviews, surveys, use-case analysis with stakeholders Design UML diagrams (Use Case, Class Diagram), wireframes, ERD Development MERN stack (MongoDB, Express.js, React.js, Node.js) Testing Unit testing, integration testing, and user acceptance testing Deployment Hosting on cloud (e.g., Vercel for frontend, Render/AWS for backend) Maintenance Regular updates, bug fixing, performance optimization
- Design Highlights
  - Frontend (React.js): Developed using a component-based structure. Each feature like login, product listing, or checkout will be a separate component.
  - Backend (Node.js + Express): RESTful APIs will be used to handle data exchange between the frontend and database.
  - Database (MongoDB): Schema-less, document-based data storage that allows flexibility in storing user profiles, product details, and transaction logs.
  - Responsiveness: The system will be fully responsive, ensuring smooth use across desktops and mobile browsers.

### 4.3.2. Flow Charts/Working Mechanism of Proposed System

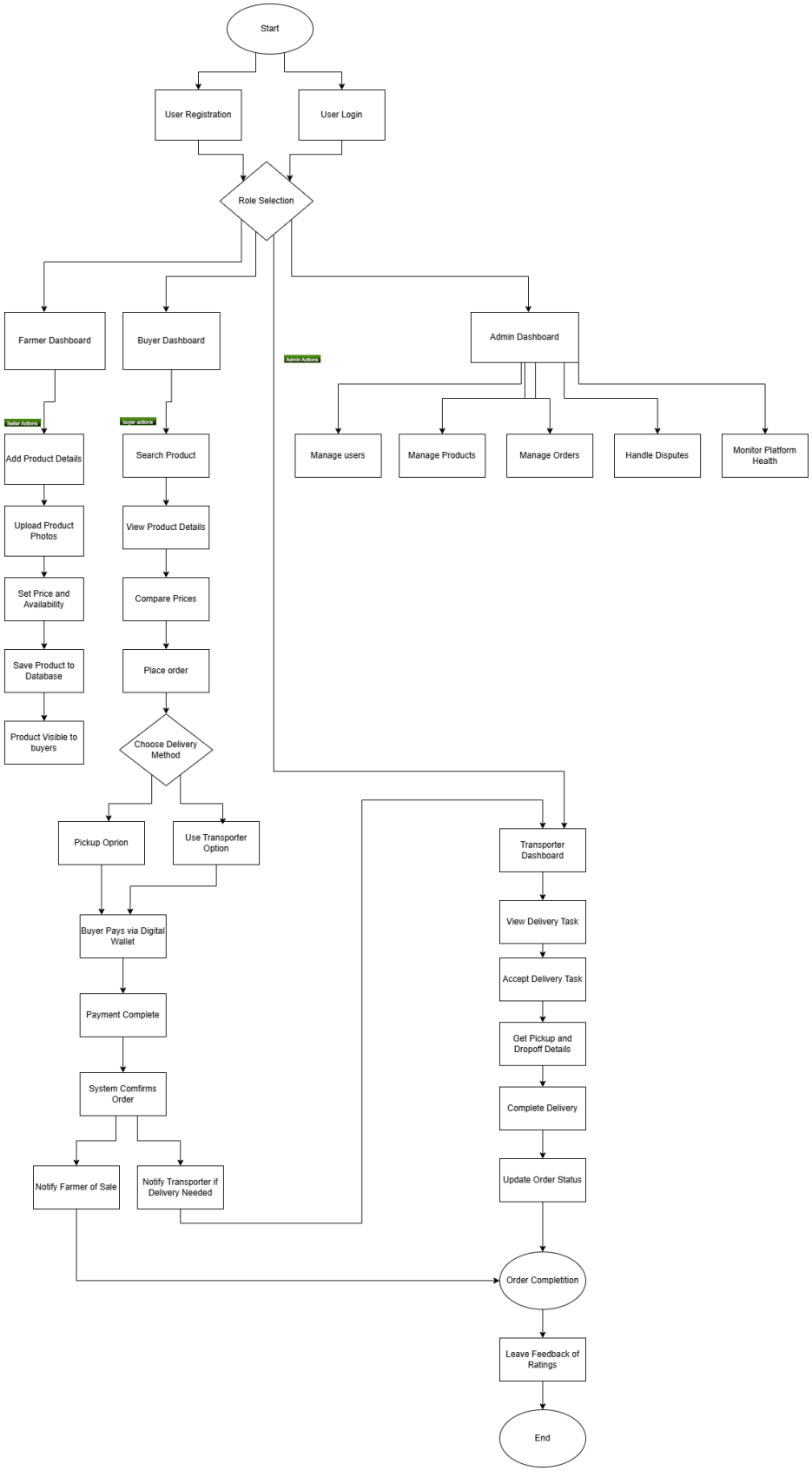


Figure 4.3: Flowchart of the proposed system


Figure 4.4: Use-Case Diagram of the proposed system

### 4.3.3. Description of Algorithms

Description of Algorithms The DOKO project will utilize several essential algorithms to ensure secure user authentication, efficient data management, and improved user experience. Each algorithm serves a specific role in achieving the platform's functionality and reliability.

1. Secure Password Hashing (bcrypt)
    - Purpose: Protect user credentials from breaches.
    - Implementation:
        - Uses the bcrypt algorithm with a salt round of 12 to hash passwords before storing them in MongoDB.
        - Automatically salts each password to prevent rainbow table attacks.
    - Relevance: Critical for farmers/buyers with limited cybersecurity awareness.
2. Token-Based Authentication (JWT with RS256)
    - Purpose: Manage user sessions securely.
    - Implementation:
        - Generates JSON Web Tokens (JWT) signed with RS256 asymmetric encryption for login verification.
        - Tokens include user roles (farmer/buyer/transporter) to authorize actions (e.g., listing products, placing orders).
    - Relevance: Prevents impersonation and restricts unauthorized access to sensitive features.
3. Dynamic Pricing Algorithm
    - Purpose: Adjust crop prices based on real-time demand and supply.
    - Implementation:
        - Linear regression analyzes historical price trends, weather data, and market demand (scraped from APIs like Smart Krishi).
        - Farmers receive price suggestions when listing products (e.g., "Tomatoes: Current avg. price = NPR 50/kg").
    - Relevance: Addresses the problem of price exploitation by middlemen.
4. Geospatial Matching for Transporters

- Purpose: Optimize delivery routes between farmers and buyers.
- Implementation:
  - Uses the Haversine formula to calculate distances between farmer locations (GPS coordinates) and urban buyers.
  - Prioritizes transporters within a 20km radius to minimize fuel costs and delays.
- Relevance: Reduces logistics overhead for perishable goods.

5. Search Ranking Algorithm

   - Purpose: Improve product discoverability.
   - Implementation:
     - TF-IDF (Term Frequency-Inverse Document Frequency) ranks search results by relevance (e.g., ”organic tomatoes” prioritizes listings with these keywords).
     - Filters by location, price, and farmer ratings (stored in MongoDB as indexed fields).
   - Relevance: Helps buyers find high-quality produce quickly.

6. Offline Data Sync (Conflict Resolution)

   - Purpose: Handle poor connectivity in rural areas.
   - Implementation:
     - Operational Transform (OT) algorithm syncs offline actions (e.g., order updates) when users reconnect.
     - Resolves conflicts (e.g., if a buyer purchases the last stock while offline) via timestamp-based priority.
   - Relevance: Ensures uninterrupted service in low-bandwidth regions.

## 5. Expected Output

The successful implementation of the DOKO platform is expected to deliver a fully functional web-based agro-marketplace tailored for farmers, buyers, and transporters in Nepal. This system will simplify the buying and selling of agricultural products using modern digital tools, supporting local economies and addressing long-standing problems in the agricultural supply chain.

Anticipated Results and Deliverables:

**i. User-Friendly Web Platform** A responsive and easy-to-use web application where:

- Farmers can list and manage their agricultural products.
- Buyers can browse, search, and purchase items easily.
- Transporters can receive and manage delivery requests.

**ii.Real-Time Product and Order Management**

- Dynamic stock updates based on purchases.
- Farmers and buyers receive instant order confirmation.
- Order status tracking for both buyers and transporters.

**iii.Secure Digital Payment Integration**

- ·Integration with local payment systems like eSewa and Khalti.
- ·Ensures faster, cashless transactions with confirmation.

**iv.Transporter Matching Feature**

- ·Delivery coordination system for assigning nearby transporters to orders.
- ·Helps bridge the gap between rural producers and urban buyers.

**v.Admin Dashboard (Optional)**

- ·For monitoring users, products, transactions, and system performance.

**How This Solves the Problem**

This project directly addresses the current challenges in Nepal's agro sector:

- It reduces the reliance on middlemen.
- It saves time and effort for both farmers and buyers.
- It brings transparency and organization to a largely informal market.

**Benefits and Impact**

- For Farmers: More control over pricing, direct market access, and better income opportunities.
- For Buyers: Easy access to fresh products at fair prices.
- For Transporters: New job opportunities and structured delivery workflow.
- For the Agricultural Ecosystem: Encourages digital transformation and boosts the rural economy.

**Contribution to Knowledge and Practice**

- Promotes digital literacy and adoption of e-commerce in rural Nepal.
- Demonstrates how MERN stack technologies can be applied in agriculture.
- Can be scaled or adapted for use in other sectors or regions.

## References


[1] M. of Agriculture and L. Development, *Statistical Information on Nepalese Agriculture*. Government of Nepal, 2022. [Online]. Available: https://moald.gov.np

[2] . K. R. Sharma, J. R., *Digital Agriculture in Nepal: Opportunities and Challenges*. Nepal Agricultural Research Council (NARC), 2021.

[3] FAO, *E-commerce in agriculture: Case studies from developing countries.* Food and Agriculture Organization., 2019.

[4] R. Pressman, *Software Engineering: A Practitioner's Approach.* McGraw-Hill, 2019.

[5] U. N. D. P. (UNDP), *Digital Economy in Nepal: Recommendations for a Digital Future*. UNDP Nepal, 2020.

[6] . S. R. K. Gautam, S., "Digital platforms for smallholder farmers in nepal: A review of opportunities and barriers," *Journal of Agriculture and Food Systems, 10(2), 45-60.*, 2021.

[7] M. Inc., *MERN Stack Guide*. MongoDB Inc., 2023.