RAILWAY TRAINING INSTITUTE

DEPARTMENT OF INFORMATION COMMUNICATION TECHNOLOGY AND LIBRARY STUDIES

DIPLOMA IN INFORMATION COMMUNICATION TECHNOLOGY

**SOKO LANGU MARKET TRADE AND MARKETING HUB SYSTEM**

PRESENTED BY:

NANA JUMA CHAJE

4010270126

A TRADE PROJECT REPORT SUBMITTED TO THE KENYA NATIONAL EXAMINATION COUNCIL IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE AWARD OF DIPLOMA IN INFORMATION COMMUNICATION TECHNOLOGY

SUPERVISED BY:

MR. JOHN NJIRU

MARCH 2025

# DECLARATION

I declare that this project report titled **Soko Langu Market Trade and Marketing Hub System** is my original work and has not been presented to any institution for academic credit.

Any references from other works have been duly acknowledged.

**CANDIDATE NAME: NANA JUMA CHAJE**

**SIGNATURE: ..............................................................**

**DATE: ...........................................................................**

This research project has been submitted for defense with my approval as the candidate’s appointed supervisor by Railway Training Institute.

**SUPERVISOR NAME: ..............................................**

**SIGNATURE: .............................................................**

**DATE:..........................................................................**

# ABSTRACT

The **Soko Langu Market Trade and Marketing Hub System** was developed to bridge the gap between farmers and consumers by providing an efficient and user-friendly platform for direct market interaction.

The system allows farmers to list their products, and consumers can browse, search, and order agricultural produce based on their region.

The system was designed using **React.js for the frontend** and **PHP with MySQL for the backend**, ensuring a seamless and dynamic user experience. It follows the **Waterfall Model** to ensure a structured development process.

The project focuses on improving market accessibility, reducing post-harvest losses, and enabling fair pricing for agricultural produce.

This report outlines the system's development, methodology, implementation, limitations, and recommendations for future improvements.

# ACKNOWLEDGEMENT

I wish to express my sincere gratitude to first and foremost the **Almighty Allah** for enabling me start and finish up this project in good and in best health.

Also to Mr. John Njiru my supervisor, whose guidance and support were invaluable during this project.

Special thanks to my family and friends for their encouragement and motivation throughout this journey.

I also appreciate my classmates and instructors at Railway Training Institute for their insights and constructive feedback.

# DEDICATION

This project is dedicated to local farmers specifically at Kariobangi Soko Market, to enable them reach more consumers without interference of middlemen.

To Railway Training Institute students to guide them accordingly.

To future researchers who might be willing to dive in more on a project like same.

# ****LIST OF ABBREVIATIONS****

**IDE:** Integrated Development Environment  
**SQL:** Structured Query Language  
**DFD:** Data Flow Diagram  
**ICT:** Information and Communication Technology  
**UI:** User Interface  
**UX:** User Experience  
**DBMS:** Database Management System  
**API:** Application Programming Interface  
**CRUD:** Create, Read, Update, Delete  
**KYC:** Know Your Customer  
**RBAC:** Role-Based Access Control

# ****SYNOPSIS****

**Soko Langu Market Trade and Marketing Hub System** was developed to address the challenges faced by farmers and consumers in agricultural trade. The system provides a **digital platform that connects farmers directly to consumers**, eliminating middlemen, ensuring fair pricing, and enhancing market efficiency. It enables **farmers to list products, manage orders, and communicate with consumers**, while **consumers can browse available products, place orders, and interact with farmers** in real time.

**Chapter One: Introduction** – Introduces the research area, including the **title, background of the study, problem statement, objectives of the study, and scope.** This chapter highlights the need for a **digital agricultural trade system** to enhance accessibility and transparency in the market.

**Chapter Two: Literature Review** – Analyzes previous research on **digital marketplaces in agriculture**, highlighting existing solutions and identifying gaps that **Soko Langu aims to fill**. The review provides insights into **e-commerce trends in farming, trade challenges, and technological solutions** that have been proposed in previous studies.

**Chapter Three: Research Methodology** – Explains th**e methodology used in carrying out the research**, including **data collection techniques, system development approach, and feasibility analysis**. The **Waterfall Model** was adopted for system development, ensuring a structured and sequential approach.

**Chapter Four: System Analysis** – Focuses on the **system study, understanding current trade challenges, and defining user requirements**. It includes **data gathering techniques, use case analysis, and system modeling through diagrams such as Data Flow Diagrams (DFDs) and Entity-Relationship Diagrams (ERDs).**

**Chapter Five: System Design** – Details the **technical architecture of the system**, including **frontend, backend, and database structures**. It also covers **user interface (UI) design, security considerations, and system flow** to ensure a **scalable and secure platform.**

**Chapter Six: System Implementation** – Describes the **coding standards, technologies used (React.js, PHP, MySQL), integration with the database, testing procedures, and deployment strategies**. The implementation phase ensured **system functionality, usability, and security.**

**Chapter Seven: Limitations, Conclusion, and Recommendations** – Identifies **system limitations, challenges encountered, and recommendations for future enhancements**. It concludes with the overall impact of the system and its potential for **expansion, including mobile app development and integration with digital payment solutions.**

TABLE OF CONTENTS

[DECLARATION i](#_Toc190119813)

[ABSTRACT ii](#_Toc190119814)

[ACKNOWLEDGEMENT iii](#_Toc190119815)

[DEDICATION iv](#_Toc190119816)

[LIST OF ABBREVIATIONS v](#_Toc190119817)

[SYNOPSIS vi](#_Toc190119818)

[TABLE OF FIGURES AND TABLES xi](#_Toc190119819)

[CHAPTER ONE: 1](#_Toc190119820)

[INTRODUCTION 1](#_Toc190119821)

[1.1 Introduction 1](#_Toc190119822)

[1.2 Background of the Problem 1](#_Toc190119823)

[1.3 Statement of the Problem 2](#_Toc190119824)

[1.4 Objectives of the Study 3](#_Toc190119825)

[1.4.1 General Objective 3](#_Toc190119826)

[1.5 Research Questions 3](#_Toc190119827)

[1.6 Significance of the Study 4](#_Toc190119828)

[1.7 Scope of the Study 4](#_Toc190119829)

[1.8 Limitations of the Study 4](#_Toc190119830)

[1.9 Basic Assumptions 5](#_Toc190119831)

[1.10 Operational Definition of Terms 5](#_Toc190119832)

[CHAPTER TWO: 6](#_Toc190119833)

[LITERATURE REVIEW 6](#_Toc190119834)

[2.1 Introduction 6](#_Toc190119835)

[2.2 Discussion of Existing Design Literature 6](#_Toc190119836)

[2.2.1 Agricultural Digital Marketplaces 6](#_Toc190119837)

[2.2.2 E-commerce in Agricultural Trade 7](#_Toc190119838)

[2.3 Evaluation of Proposed Literature 7](#_Toc190119839)

[Key Findings from Literature 7](#_Toc190119840)

[2.4 Summary of Gaps Identified 8](#_Toc190119841)

[2.5 Conclusion 8](#_Toc190119842)

[CHAPTER THREE: 9](#_Toc190119843)

[SYSTEM METHODOLOGY 9](#_Toc190119844)

[3.1 Introduction 9](#_Toc190119845)

[3.2 Software Development Model 9](#_Toc190119846)

[3.2.1 Overview of Software Development Models 9](#_Toc190119847)

[3.2.2 Justification for Using the Waterfall Model 10](#_Toc190119848)

[3.3 Phases of System Development (Waterfall Model) 10](#_Toc190119849)

[3.3.1 Requirement Analysis 10](#_Toc190119850)

[3.3.2 System Design 10](#_Toc190119851)

[3.3.3 Implementation 11](#_Toc190119852)

[3.3.4 Testing 11](#_Toc190119853)

[3.3.5 Deployment & Maintenance 11](#_Toc190119854)

[3.4 Tools and Technologies Used 11](#_Toc190119855)

[3.5 System Requirements 12](#_Toc190119856)

[3.5.1 Functional Requirements 12](#_Toc190119857)

[3.5.2 Non-Functional Requirements 12](#_Toc190119858)

[3.6 Summary 12](#_Toc190119859)

[CHAPTER FOUR: 13](#_Toc190119860)

[SYSTEM ANALYSIS 13](#_Toc190119861)

[4.1 Introduction 13](#_Toc190119862)

[4.2 Understanding Software Limitations & Problems in Existing Systems 13](#_Toc190119863)

[4.3 Impact on Organization & Users 14](#_Toc190119864)

[4.4 Data Gathering Techniques 14](#_Toc190119865)

[4.4.1 Interviews 14](#_Toc190119866)

[4.4.2 Observations 14](#_Toc190119867)

[4.4.3 Literature Review 14](#_Toc190119868)

[4.5 Requirements Determination 15](#_Toc190119869)

[4.5.1 Functional Requirements 15](#_Toc190119870)

[4.5.2 Non-Functional Requirements 15](#_Toc190119871)

[4.6 System Models & Diagrams 15](#_Toc190119872)

[4.6.1 Use Case Diagram 15](#_Toc190119873)

[4.6.2 Data Flow Diagrams (DFDs) 16](#_Toc190119874)

[4.6.3 Flowcharts 18](#_Toc190119875)

[4.7 Summary 20](#_Toc190119876)

[CHAPTER FIVE: 21](#_Toc190119877)

[SYSTEM DESIGN 21](#_Toc190119878)

[5.1 Introduction 21](#_Toc190119879)

[5.2 Architectural Design 21](#_Toc190119880)

[5.3 Data Design 22](#_Toc190119881)

[5.4 User Interface (UI) Design 24](#_Toc190119882)

[5.5 System Security Design 25](#_Toc190119883)

[5.6 Summary 26](#_Toc190119884)

[CHAPTER SIX: 27](#_Toc190119885)

[SYSTEM IMPLEMENTATION 27](#_Toc190119886)

[6.1 Introduction 27](#_Toc190119887)

[6.2 Coding Standards and Guidelines 27](#_Toc190119888)

[6.3 Development Tools and Environment 28](#_Toc190119889)

[6.4 Integration and Testing 28](#_Toc190119890)

[6.4.1 System Integration 28](#_Toc190119891)

[6.4.2 Testing Methods 29](#_Toc190119892)

[6.4.3 Test Cases and Results 29](#_Toc190119893)

[6.5 Deployment 29](#_Toc190119894)

[6.5.1 Local Deployment 29](#_Toc190119895)

[6.5.2 Online Hosting 29](#_Toc190119896)

[6.6 Post-Implementation Review 30](#_Toc190119897)

[6.7 Summary 30](#_Toc190119898)

[CHAPTER SEVEN: 31](#_Toc190119899)

[LIMITATIONS, CONCLUSIONS, AND RECOMMENDATIONS 31](#_Toc190119900)

[7.1 Introduction 31](#_Toc190119901)

[7.2 Limitations 31](#_Toc190119902)

[7.2.1 Technical Limitations 31](#_Toc190119903)

[7.2.2 Resource Limitations 31](#_Toc190119904)

[7.2.3 Data Limitations 32](#_Toc190119905)

[7.2.4 User Limitations 32](#_Toc190119906)

[7.3 Recommendations 32](#_Toc190119907)

[7.3.1 Technical Recommendations 32](#_Toc190119908)

[7.3.2 Resource Recommendations 32](#_Toc190119909)

[7.3.3 Data Recommendations 33](#_Toc190119910)

[7.3.4 User Recommendations 33](#_Toc190119911)

[7.4 Conclusion 33](#_Toc190119912)

[7.4.1 Project Achievement 33](#_Toc190119913)

[7.4.2 Future Prospects 33](#_Toc190119914)

[7.4.3 Final Thoughts 34](#_Toc190119915)

[REFERENCES 35](#_Toc190119916)

[APPENDICES 36](#_Toc190119917)

[Appendix A: Database Structure 36](#_Toc190119918)

[Appendix B: Code Samples 37](#_Toc190119919)

[Appendix C: System Screenshots 40](#_Toc190119920)

# TABLE OF FIGURES AND TABLES

[Figure 1: usecase 16](#_Toc190119760)

[Figure 2: Level 0 DFD 17](#_Toc190119761)

[Figure 3: Level 1 DFD 18](#_Toc190119762)

[Figure 4: User Registration Flowchart 19](file:///C:\Users\JUMANJI\OneDrive\Documents\NEW%20NOTES%20FINAL%20YR\Soko%20Langu%20Final%20Report.docx#_Toc190119763)

[Figure 5: Product Listing Flowchart 20](#_Toc190119764)

[Figure 6: HomePage UI 24](#_Toc190119765)

[Figure 7: Farmer Dashboard UI 25](#_Toc190119766)

[Figure 8: Database Schema 36](#_Toc190119767)

[Figure 9: Consumer Messaging Screen 40](#_Toc190119768)

[Figure 10: Farmer Messaging Screen 40](#_Toc190119769)

[Figure 11: Consumer Signup Screen 41](#_Toc190119770)

[Figure 12: Farmer Signup Screen with Validation 42](#_Toc190119771)

[Figure 13: Login Screen 42](#_Toc190119772)

[Figure 14: ProductView Screen 43](#_Toc190119773)

[Figure 15: Search Filter Screen 44](#_Toc190119774)

[Figure 16: Admin Dashboard Screen 44](#_Toc190119775)

# CHAPTER ONE:

# INTRODUCTION

## 1.1 Introduction

Agricultural markets play a crucial role in ensuring food security, economic growth, and sustainable livelihoods for farmers. However, traditional marketplaces often present challenges such as middlemen exploitation, limited market reach, and fluctuating prices, which negatively impact farmers and consumers alike.

The **Soko Langu Market Trade and Marketing Hub System** was developed to address these challenges by providing a digital platform where farmers can list their agricultural produce and consumers can directly access and purchase products based on their region. This system enhances market accessibility, improves efficiency in transactions, and eliminates unnecessary intermediaries, ensuring fair pricing and better market conditions for all stakeholders.

This chapter presents the background of the problem, statement of the problem, research objectives, research questions, significance, scope, limitations, and basic assumptions of the study.

## 1.2 Background of the Problem

The agricultural sector is a key driver of economic growth in many developing countries, including Kenya. However, farmers often face difficulties in accessing markets due to traditional trading systems that involve multiple intermediaries, resulting in reduced profits for producers and increased prices for consumers.

Technological advancements have provided opportunities for digital transformation in various industries, including agriculture. With the increasing adoption of e-commerce and mobile-based solutions, there is a need to modernize agricultural trade by providing farmers with a direct, transparent, and fair marketplace.

The **Soko Langu Market Trade and Marketing Hub System** aimed to bridge this gap by leveraging web-based technology to connect farmers with consumers, ensuring efficient trade, transparency, and accessibility. The system allowed farmers to create accounts, upload product details (such as name, description, price, and category), and receive orders directly from consumers. Consumers, on the other hand, could browse, search, and place orders from farmers within their regions or all over.

By eliminating unnecessary intermediaries, the system sought to reduce transaction costs,ensure fair pricing, and improve farmers’ economic well-being while offering consumers fresh and affordable produce.

## 1.3 Statement of the Problem

Agricultural trade in Kenya remains largely inefficient due to traditional market systems characterized by middlemen control, price fluctuations, and limited access to markets for small-scale farmers. Farmers often sell their produce at significantly lower prices, while consumers face higher costs due to added distribution layers.

Existing digital solutions, such as social media marketplaces and e-commerce platforms, do not cater specifically to regional agricultural trade, making it challenging for small-scale farmers to effectively reach their target consumers.

Thus, there was a need for a dedicated agricultural trade platform that allows farmers to list and sell their produce directly to consumers within their regions, ensuring fair pricing, market transparency, and improved economic benefits. The **Soko Langu Market Trade and Marketing Hub System** was developed to address these inefficiencies by creating a digital marketplace tailored to farmers and consumers, thereby improving market access and reducing dependency on intermediaries.

## 1.4 Objectives of the Study

### 1.4.1 General Objective

The general objective of this study was to develop and implement the **Soko Langu Market Trade and Marketing Hub System**, a digital platform that facilitates direct trade between farmers and consumers to enhance market accessibility, transparency, and efficiency in agricultural transactions.

**1.4.2 Specific Objectives**

The specific objectives of the study were:

1. To design a web-based platform where farmers can list and manage their agricultural produce.
2. To implement a user-friendly interface that allows consumers to search for products based on region and place orders seamlessly.
3. To develop a secure and reliable backend using PHP and MySQL for managing users, products, orders, and transactions.
4. To integrate a messaging feature that enables direct communication between farmers and consumers for order inquiries and confirmations.
5. To evaluate the performance, usability, and effectiveness of the system in enhancing agricultural trade.

## 1.5 Research Questions

The study sought to answer the following questions:

1. How can a web-based marketplace enhance market accessibility for farmers and consumers?
2. What features should be implemented to ensure a user-friendly and secure trading experience?
3. How can a region-based product search feature improve agricultural trade efficiency?
4. What impact does direct farmer-to-consumer trade have on product pricing and availability?
5. How effective is the system in reducing dependency on middlemen and ensuring fair market practices?

## 1.6 Significance of the Study

The **Soko Langu Market Trade and Marketing Hub System** was significant to multiple stakeholders, including:

1. **Farmers** – Provided a direct market to sell produce without intermediaries, ensuring fair pricing and better profits.
2. **Consumers** – Enabled access to fresh, affordable farm produce while eliminating unnecessary price inflations.
3. **Agricultural Stakeholders** – Served as a data-driven platform for analyzing market trends and improving trade policies.
4. **Researchers & Developers** – Provided insights into digital transformation in agriculture and the potential of technology-driven trade solutions.

By implementing a region-based e-commercemodel, the system contributed to economic empowerment, sustainable agriculture, and food security.

## 1.7 Scope of the Study

The study focused on developing and implementing a web-based trading platform that allows farmers to list products and consumers to place orders based on region.

The system was limited to agricultural products, particularly fresh farm produce. It targeted farmers and consumers within regions in Kenya, ensuring localized trade efficiency.

The platform provided account management, product listing, order management, and messaging features but did not include payment processing, leaving transactions to be completed manually between farmers and consumers.

## 1.8 Limitations of the Study

Despite the successful implementation, the study faced several challenges:

1. **Internet Dependency** – Since the platform was web-based, users in remote areas with poor internet connectivity experienced challenges accessing the system.
2. **Limited Payment Integration** – The system did not include an automated payment system, requiring consumers and farmers to finalize transactions manually.
3. **User Adoption** – Farmers with limited digital literacy required training on how to effectively use the system.

## 1.9 Basic Assumptions

The study was based on the following assumptions:

1. Farmers and consumers were willing to adopt digital solutions for agricultural trade.
2. The target users had access to basic internet and smartphones to use the system.
3. Farmers would list genuine products, and consumers would place authentic orders without fraudulent intent.

## 1.10 Operational Definition of Terms

* **E-commerce** – The process of buying and selling goods over the internet.
* **Agri-Tech** – Technology-driven solutions designed to improve agricultural productivity and trade.
* **Middlemen** – Intermediaries who buy products from farmers at lower prices and sell them to consumers at higher prices.
* **User Interface (UI)** – The visual part of a website or application that users interact with.
* **Marketplace Platform** – A digital platform where buyers and sellers meet to conduct business.

# CHAPTER TWO:

# LITERATURE REVIEW

This chapter presents an overview of existing research, technological advancements, and previous systems related to digital marketplaces in agriculture. It evaluates existing designs, discusses their strengths and weaknesses, and identifies the gaps that Soko Langu Market Trade and Marketing Hub System aimed to address.

## 2.1 Introduction

Agricultural e-commerce platforms have become increasingly relevant in modern markets due to their ability to bridge the gap between farmers and consumers. Various studies have explored how technology-driven trade can improve efficiency, reduce post-harvest losses, and ensure fair pricing.

This chapter evaluates literature on agricultural market systems, existing solutions, and the technological framework used to develop **Soko Langu**. It further highlights gaps in previous studies and how this project sought to address them.

## 2.2 Discussion of Existing Design Literature

Several digital solutions have been implemented globally to enhance agricultural trade. Some of these platforms include e-commerce websites, mobile applications, and digital marketplaces designed for farmers.

### 2.2.1 Agricultural Digital Marketplaces

**1. M-Farm (Kenya)**

* A **SMS-based agricultural platform** that connects farmers with buyers by providing price transparency and market access.
* **Limitations:** Farmers must manually coordinate transactions, and the platform lacks real-time order processing.

**2. Twiga Foods (Kenya)**

* A **B2B (Business-to-Business) e-commerce platform** that links farmers to retailers by aggregating farm produce and distributing it to urban markets.
* **Limitations:** Twiga focuses on bulk supply to businesses, leaving individual consumers without direct access to farmers.

**3. AgroMarketDay (Uganda)**

* A **mobile-based marketplace** that provides farmers with a platform to list their produce and connect with buyers.
* **Limitations:** Does not support region-based filtering, making it difficult for users to find local produce efficiently.

### 2.2.2 E-commerce in Agricultural Trade

The adoption of e-commerce solutions has significantly transformed agricultural trade by improving accessibility, reducing costs, and eliminating middlemen. However, research highlights challenges such as:

1. Limited digital literacy among farmers, hindering technology adoption.
2. Inadequate regional market filtering, preventing efficient consumer access to locally available produce.
3. High dependency on internet connectivity, limiting access in remote areas.

## 2.3 Evaluation of Proposed Literature

Several studies emphasize the need for localized agricultural e-commerce solutions that enable farmers to sell directly to consumers without intermediaries.

### Key Findings from Literature

* Localized digital marketplaces improve trade efficiency by ensuring faster delivery and lower transaction costs.
* Direct trade between farmers and consumers enhances pricing fairness.
* Mobile-based applications have higher adoption rates than web-only solutions in rural areas.

**Soko Langu Market Trade and Marketing Hub System** was designed to integrate these findings by:

* Providing a region-based filtering system to help consumers find nearby farmers.
* Offering an intuitive web-based platform accessible from both desktop and mobile devices.
* Allowing direct messaging between farmers and consumers to streamline communication and order processing.

## 2.4 Summary of Gaps Identified

Existing agricultural trade platforms have several limitations that **Soko Langu** aimed to address:

|  |  |
| --- | --- |
| Identified Gaps | How Soko Langu Addressed Them |
| Lack of region-based filtering | Implemented a search-by-region feature |
| Dependency on middlemen | Enabled direct farmer-to-consumer transactions |
| No real-time communication | Integrated messaging features for instant farmer-consumer interaction |
| Limited adoption due to complexity | Designed a user-friendly interface with simple navigation |

Table 1: summary of gaps identified

## 2.5 Conclusion

The literature review identified key technological and operational challenges in existing agricultural e-commerce platforms. While previous studies highlight the benefits of digital trade, gaps such as poor region-based search, middleman dependency, and inefficient communication remain unaddressed.

**Soko Langu Market Trade and Marketing Hub System** was developed to bridge these gaps by providing a localized, farmer-to-consumer marketplace that ensures efficient, transparent, and fair agricultural trade.

# CHAPTER THREE:

# SYSTEM METHODOLOGY

This chapter outlines the methodology used in the design and development of the **Soko Langu Market Trade and Marketing Hub System**. It describes the software development model, data collection methods, system requirements, and design techniques used to ensure an effective implementation.

## 3.1 Introduction

Developing an efficient and reliable system requires a structured approach to system analysis, design, development, and testing. The Waterfall Model was adopted as the software development methodology due to its step-by-step approach, which ensures that each phase is completed before proceeding to the next.

This chapter discusses why the Waterfall Model was chosen, the phases of system development, the tools and technologies used, and how the system requirements were defined.

## 3.2 Software Development Model

### 3.2.1 Overview of Software Development Models

Several software development models exist, including:

1. **Waterfall Model** – A structured approach where each phase is completed before the next begins.
2. **Agile Model** – An iterative approach that allows continuous revisions.
3. **Spiral Model** – Combines iterative development with risk analysis.

For this project, the Waterfall Model was selected due to its clear structure and ease of implementation.

### 3.2.2 Justification for Using the Waterfall Model

The Waterfall Model was suitable for **Soko Langu Market Trade and Marketing Hub System** because:

* It follows a sequential approach that ensures proper planning and documentation.
* Each phase is completed before moving to the next, minimizing errors.
* It is well-documented, making future maintenance easier.

The five major phases of the Waterfall Model in this project were:

1. **Requirement Analysis**
2. **System Design**
3. **Implementation (Coding & Development)**
4. **Testing**
5. **Deployment & Maintenance**

## 3.3 Phases of System Development (Waterfall Model)

### 3.3.1 Requirement Analysis

During this phase, the researcher identified functional and non-functional requirements based on the needs of farmers and consumers. Data collection methods used included:

* **Interviews** with farmers to understand market challenges.
* **Observation** of existing trade practices.
* **Review of existing e-commerce systems** to identify gaps.

### 3.3.2 System Design

This phase involved designing the database, user interfaces, and system architecture.

* **Entity-Relationship Diagrams (ERDs)** were used to define database relationships.
* **Data Flow Diagrams (DFDs)** were used to illustrate system processes.
* **Flowcharts** were created to map out the user journey.

### 3.3.3 Implementation

This phase involved coding the system using:

* **Frontend:** React.js
* **Backend:** PHP with MySQL
* **Database Management:** phpMyAdmin
* **Styling & UI Components:** CSS Vanilla

### 3.3.4 Testing

Testing was performed to ensure system stability, performance, and user experience. Testing methods included:

* **Unit Testing** – Each module was tested individually.
* **Integration Testing** – Components were tested together to ensure smooth functionality.
* **User Testing** – Farmers and consumers tested the system and provided feedback.

### 3.3.5 Deployment & Maintenance

The system was deployed in a local development environment before considering hosting on an online server. Future maintenance involved bug fixes, security patches, and feature upgrades.

# 3.4 Tools and Technologies Used

|  |  |
| --- | --- |
| Technology | Purpose |
| React.js | Frontend development (User Interface) |
| PHP | Backend development (Server-side processing) |
| MySQL | Database management system |
| Axios | Handling API requests between frontend and backend |
| CSS | Styling and UI design |
| phpMyAdmin | Managing database interactions |
| GitHub | Version control and code backup |

Table 2: Tools and Technologies used

## 3.5 System Requirements

### 3.5.1 Functional Requirements

The system was required to:  
✅ Allow farmers to create accounts, add products, and receive orders.  
✅ Enable consumers to search for products based on region and place orders.  
✅ Provide a messaging feature for direct communication.  
✅ Support admin management of users and system data.

### 3.5.2 Non-Functional Requirements

The system had to be:  
✅ **User-friendly** – Easy navigation for farmers and consumers.  
✅ **Secure** – Prevent unauthorized access and data breaches.  
✅ **Scalable** – Allow future expansion and improvements.  
✅ **Responsive** – Accessible on both mobile and desktop devices.

## 3.6 Summary

This chapter outlined the methodology used in developing the system. The Waterfall Model was chosen due to its structured approach, ensuring proper requirement analysis, design, implementation, testing, and deployment.

The system was developed using React.js for the frontend, PHP for the backend, and MySQL for database management. Data was collected through interviews, observations, and literature reviews, which helped in designing an efficient system that met farmers' and consumers' needs.

# CHAPTER FOUR:

# SYSTEM ANALYSIS

This chapter presents an in-depth analysis of the **Soko Langu Market Trade and Marketing Hub System**. It explores the **l**imitations of existing solutions, system requirements, data collection techniques, and system models such as DFDs, Use Case Diagrams, and Flowcharts.

## 4.1 Introduction

System analysis involves understanding software limitations, identifying system-related challenges, and proposing necessary improvements. In this project, system analysis focused on:  
✅ Examining the problems farmers and consumers face in agricultural trade.  
✅ Understanding how existing solutions work and their limitations.  
✅ Identifying the functional and non-functional requirements of the new system.  
✅ Designing models that visualize data flow and system interactions.

The Waterfall Model was used, ensuring step-by-step analysis before development.

## 4.2 Understanding Software Limitations & Problems in Existing Systems

Existing agricultural trade methods face several challenges, including:

* **Middlemen Dependency** – Farmers sell at lower prices while consumers buy at higher costs.
* **Lack of Direct Communication** – Farmers and consumers cannot interact efficiently.
* **Limited Regional Trade** – Consumers struggle to find nearby farmers for fresh produce.
* **Manual Order Processing** – Transactions lack automation, causing inefficiencies.

The **Soko Langu System** aimed to address these challenges by eliminating middlemen, improving direct farmer-consumer interactions, and introducing a user-friendly marketplace.

## 4.3 Impact on Organization & Users

The implementation of this system was expected to positively impact various stakeholders:

|  |  |
| --- | --- |
| Stakeholder | Impact |
| Farmers | Increased market access, fair pricing, and direct customer engagement. |
| Consumers | Easy access to fresh produce, cost reduction, and direct farmer interaction. |
| Market System | Enhanced efficiency, transparency, and a structured trade process. |

Table 3: Stakeholders

## 4.4 Data Gathering Techniques

To ensure a well-informed system design, the researcher used various data collection methods:

### 4.4.1 Interviews

✅ Farmers were interviewed to understand their challenges in selling produce.  
✅ Consumers were asked about their experience in purchasing agricultural products.

### 4.4.2 Observations

✅ The researcher observed traditional marketplace operations, noting delays, pricing issues, and communication barriers.

### 4.4.3 Literature Review

✅ Existing systems like **M-Farm, Twiga Foods,** and **AgroMarketDay** were analyzed to identify gaps and opportunities for improvement.

## 4.5 Requirements Determination

### 4.5.1 Functional Requirements

These requirements define what the system must do:  
✅ User registration and login (Farmers & Consumers).  
✅ Farmers can add, edit, and delete products.  
✅ Consumers can search for products by region.  
✅ Consumers can place orders and receive confirmation.  
✅ Messaging system for farmer-consumer communication.

### 4.5.2 Non-Functional Requirements

These ensure the system operates efficiently and securely:  
✅ User-friendly interface for easy navigation.  
✅ Secure authentication using hashed passwords.  
✅ Fast system response time for search and order processing.  
✅ Scalability for future feature expansion.

## 4.6 System Models & Diagrams

### 4.6.1 Use Case Diagram

* **Actors:** Farmer, Consumer, Admin.
* **Interactions:** Farmers manage products, consumers place orders, admin oversees system operations.

As shown in the figure below:



Figure 1: usecase

### 4.6.2 Data Flow Diagrams (DFDs)

* **Level 0 DFD:** High-level flow between Farmers, Consumers, and the System.
* **Level 1 DFD:** Detailed breakdown of user interactions, data processing, and system storage.

As shown in the below figures:

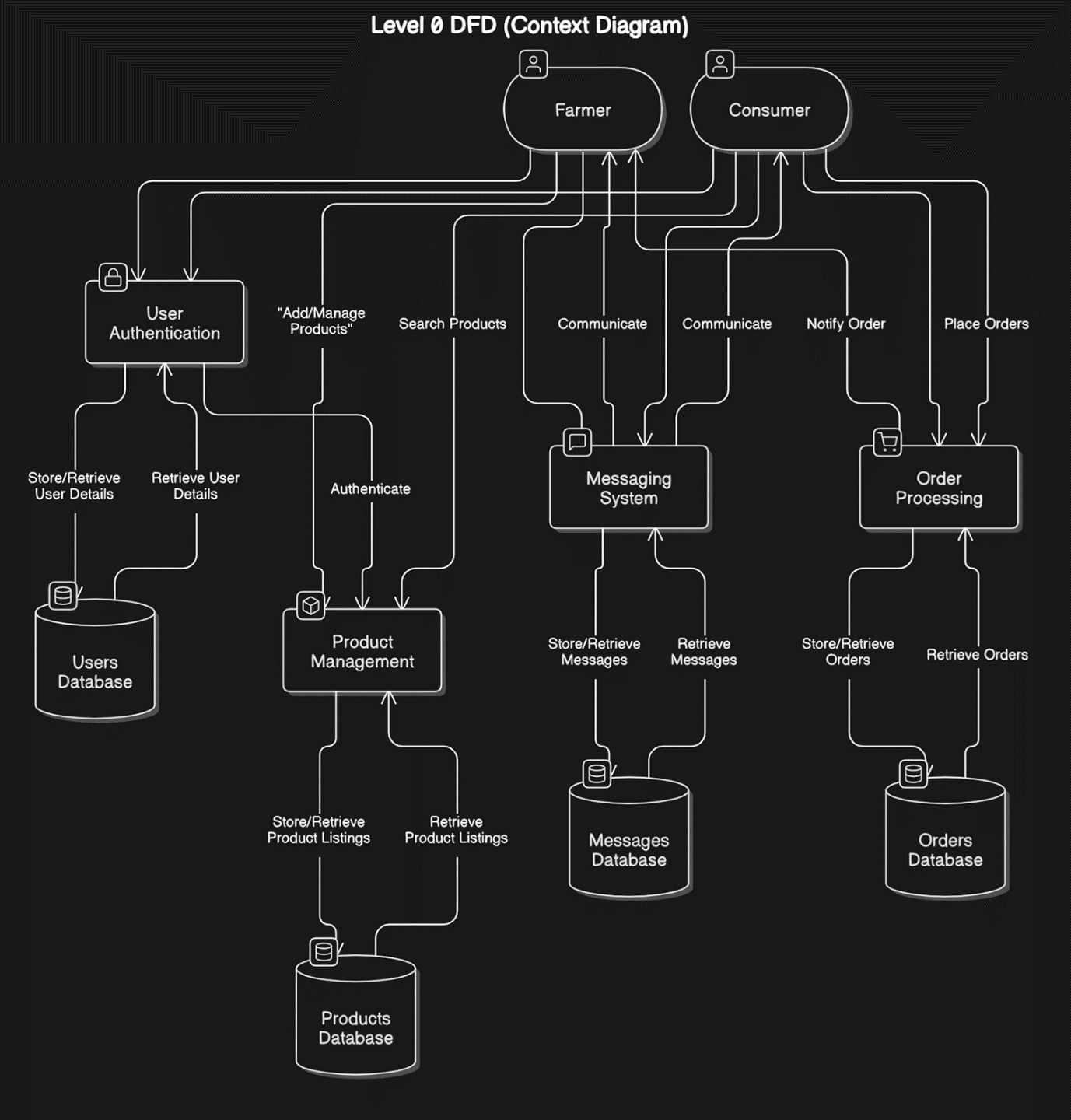


Figure 2: Level 0 DFD

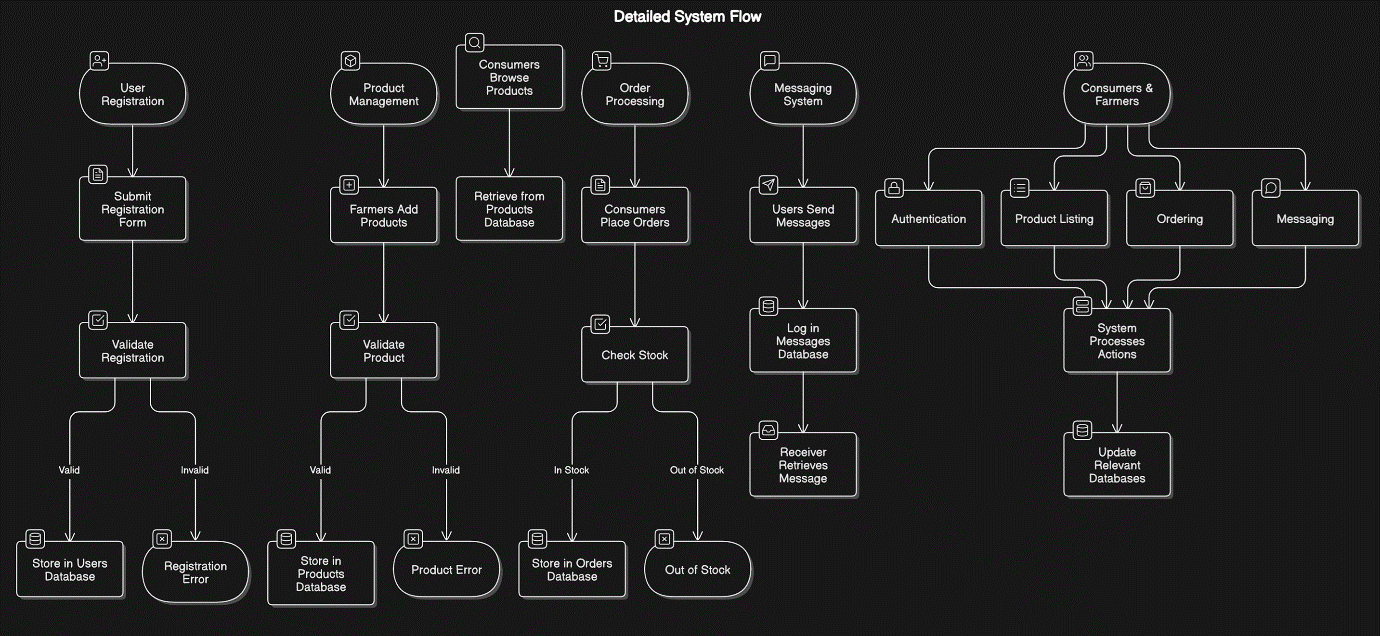


Figure 3: Level 1 DFD

### 4.6.3 Flowcharts

* **User Registration Flowchart** – Steps from sign-up to database storage.
* **Product Listing Flowchart** – Farmer uploads a product, and the system verifies and saves it.
* **Order Placement Flowchart** – Consumer selects a product, places an order, and the system updates records.
* **Messaging Flowchart** – User sends a message, and the system delivers it to the recipient.

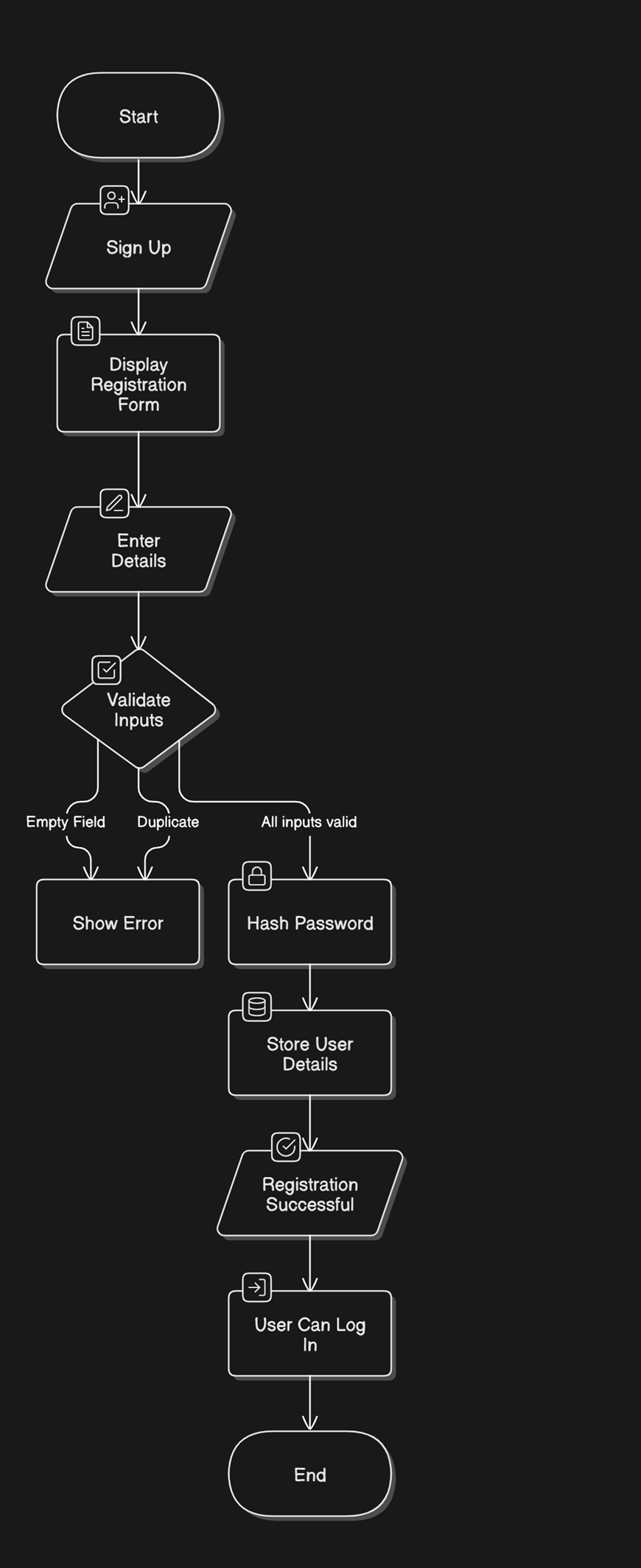
As shown in the below figures:

Figure 4: User Registration Flowchart

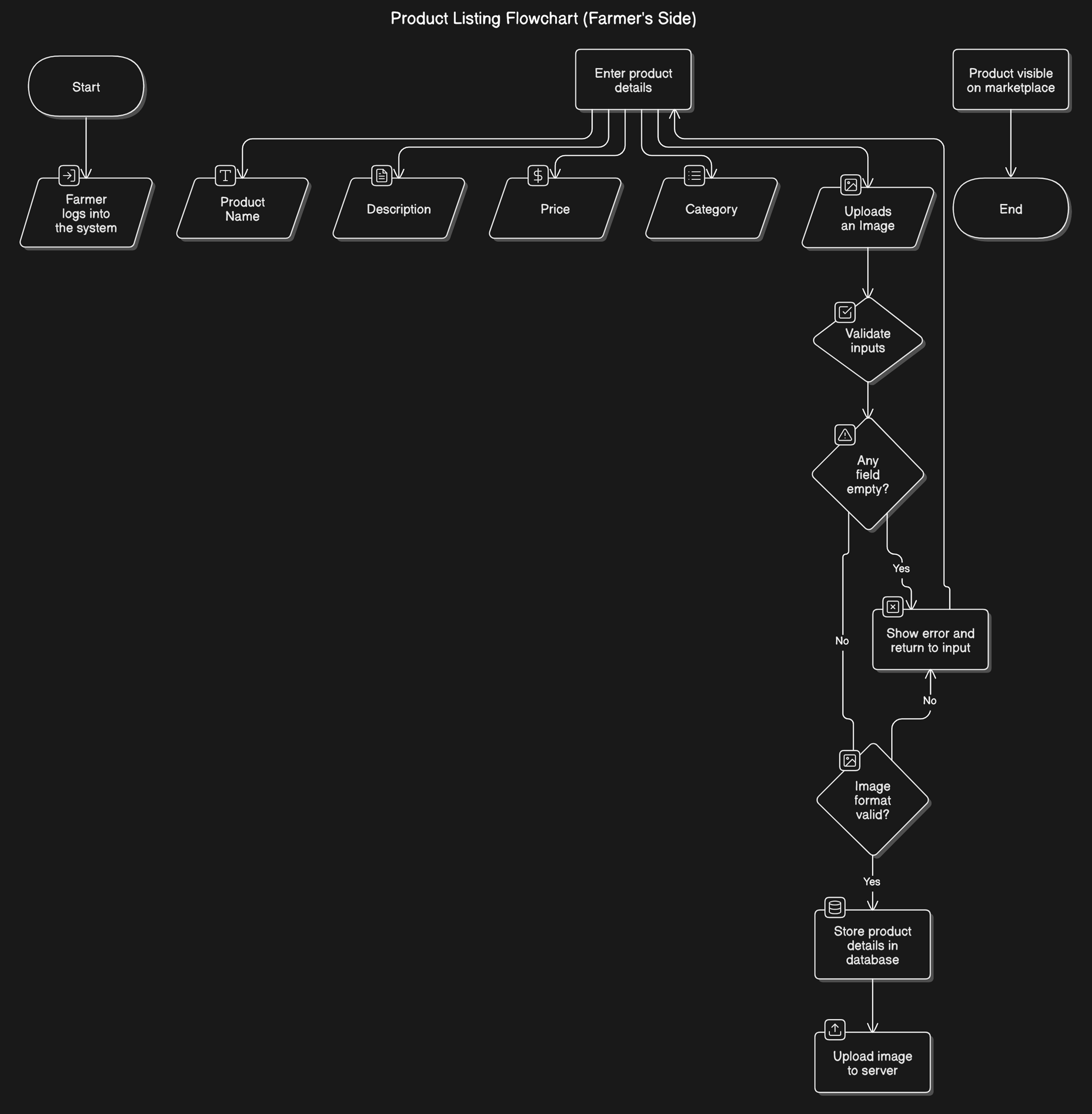


Figure 5: Product Listing Flowchart

## 4.7 Summary

This chapter focused on analyzing existing agricultural trade challenges, gathering requirements, and visualizing system interactions using DFDs, Use Cases, and Flowcharts.

The **Soko Langu System** was designed to simplify agricultural trade by enabling direct transactions between farmers and consumers while ensuring efficiency, security, and usability.

# ****CHAPTER FIVE:****

# ****SYSTEM DESIGN****

This chapter presents the **technical design of the Soko Langu Market Trade and Marketing Hub System**. It includes **architectural design, database structure, user interface design, and security measures** to ensure efficiency, usability, and security in system operations.

## ****5.1 Introduction****

System design involves defining the **overall structure, data organization, and security protocols** of an application. The **Soko Langu System** was designed using **a three-tier architecture** to ensure a **clear separation of concerns** between the **frontend, backend, and database.**

This chapter discusses the **system architecture, database design, user interface design, and security features** implemented to ensure a robust and scalable system.

## ****5.2 Architectural Design****

The **Soko Langu System** follows a **three-tier architecture**, ensuring **modular development and easy maintenance:**

**1. Presentation Layer (Frontend)**  
✅ Developed using **React.js**, providing a **user-friendly and responsive interface.**  
✅ Handles **user input, navigation, and API requests**.

**2. Business Logic Layer (Backend)**  
✅ Implemented using **PHP**, responsible for **processing requests, handling authentication, and managing data.**  
✅ Uses **RESTful APIs** to communicate with the frontend.

**3. Data Storage Layer (Database)**  
✅ Built using **MySQL**, storing information on **users, products, orders, and messages**.  
✅ Managed using **phpMyAdmin** for **data retrieval and updates**.

## ****5.3 Data Design****

The **database design** ensures **efficient data storage, retrieval, and integrity**. The system consists of **five key tables**:

**1. Users Table** – Stores user information.

|  |  |  |
| --- | --- | --- |
| ****Field**** | ****Type**** | ****Description**** |
| UserID | VARCHAR(200) | Unique identifier for users |
| Username | VARCHAR(50) | User's chosen username |
| FirstName | VARCHAR(50) | User's first name |
| LastName | VARCHAR(20) | User's last name |
| PhoneNo | VARCHAR(100) | User's contact number |
| Email | VARCHAR(100) | User's email (unique) |
| Region | VARCHAR(50) | User's location |
| Role | VARCHAR(20) | Defines user type (Farmer/Consumer) |
| Password | VARCHAR(255) | Hashed password |
| JoinedDate | TIMESTAMP | Account creation date |
| Status | VARCHAR(20) | Online/Offline status |
| Photo | VARCHAR(255) | Profile picture path |

Table 4: Users table

**2. Products Table** – Stores product details.

|  |  |  |
| --- | --- | --- |
| ****Field**** | ****Type**** | ****Description**** |
| ProductID | VARCHAR(200) | Unique identifier for products |
| UserID | VARCHAR(200) | ID of the farmer who owns the product |
| Image | VARCHAR(255) | Product image path |
| Name | VARCHAR(100) | Product name |
| Description | VARCHAR(500) | Product details |
| Price | VARCHAR(100) | Product price (stored as text for flexibility) |
| Category | VARCHAR(200) | Type of product |
| UploadDate | VARCHAR(200) | Date product was added |

Table 5: Products table

**3. Orders Table** – Tracks purchases.

|  |  |  |
| --- | --- | --- |
| ****Field**** | ****Type**** | ****Description**** |
| OrderID | VARCHAR(200) | Unique order identifier |
| UserID | VARCHAR(200) | ID of the consumer placing the order |
| ProductName | VARCHAR(200) | Name of the ordered product |
| Quantity | VARCHAR(11) | Number of items ordered |
| TotalPrice | VARCHAR(50) | Total cost of order |
| OrderStatus | VARCHAR(50) | Status (Pending/Completed) |
| OrderDate | TIMESTAMP | Date of order |

Table 6: Orders table

**4. Messages Table** – Handles messaging between users.

|  |  |  |
| --- | --- | --- |
| ****Field**** | ****Type**** | ****Description**** |
| MsgID | VARCHAR(200) | Unique message ID |
| SenderID | VARCHAR(200) | ID of the sender |
| ReceiverID | VARCHAR(200) | ID of the recipient |
| Message | VARCHAR(1000) | Message content |
| Seen | VARCHAR(10) | Status (Yes/No) |
| Date | TIMESTAMP | Time of message |

Table 7: Messages table

**5. Wishlist Table** – Stores consumer wishlists.

|  |  |  |
| --- | --- | --- |
| ****Field**** | ****Type**** | ****Description**** |
| WishID | VARCHAR(200) | Unique wishlist entry ID |
| UserID | VARCHAR(200) | Consumer ID |
| ProductID | VARCHAR(200) | Product added to wishlist |
| Date | TIMESTAMP | Date added |

Table 8: Wishlist table

**6. Reset\_Temp Table** – Stores token which are temporary when reset password is sent to email.

|  |  |  |
| --- | --- | --- |
| ****Field**** | ****Type**** | ****Description**** |
| email | VARCHAR(200) | Unique wishlist entry ID |
| token | VARCHAR(255) | Consumer ID |
| expDate | DATETIME | Date added |

Table 9: Reset\_temp table

## ****5.4 User Interface (UI) Design****

This section includes some of the visual looks of the system:

* **Homepage design** – Showcasing product categories.
* **Farmer dashboard** – Product management & order tracking.
* **Consumer Product View** – Product viewing & order placement.
* **Messaging system** – Chat interface for consumer-farmer communication.
* **Search Filtering screens** – Order placement and confirmation views.

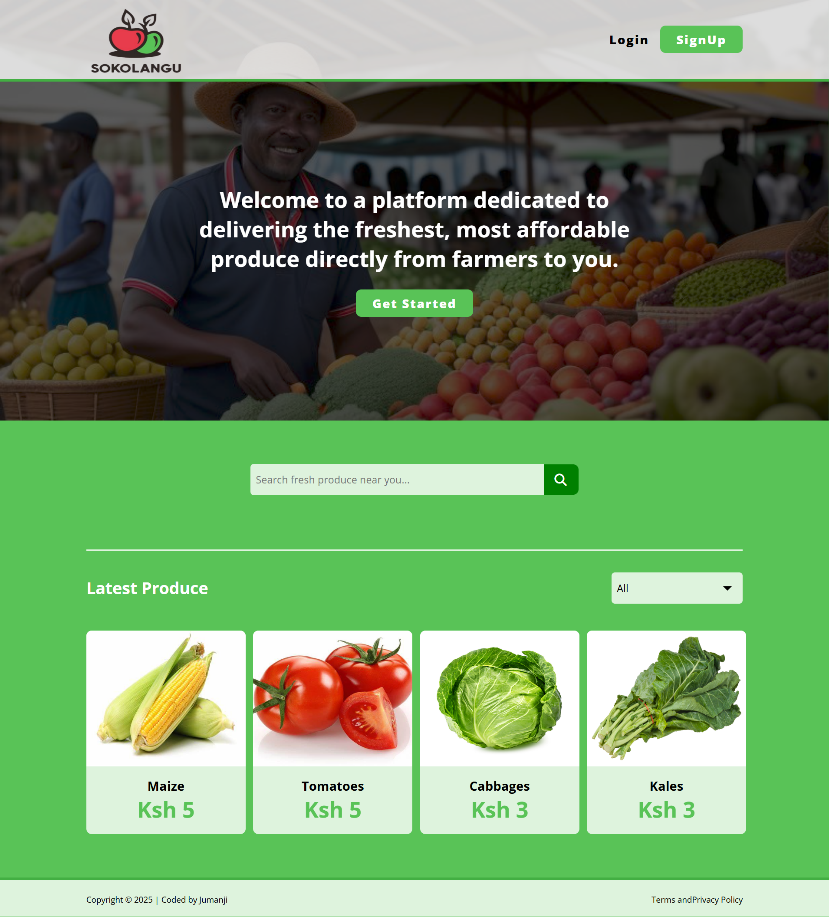
As shown below a few of them:  


Figure 6: HomePage UI

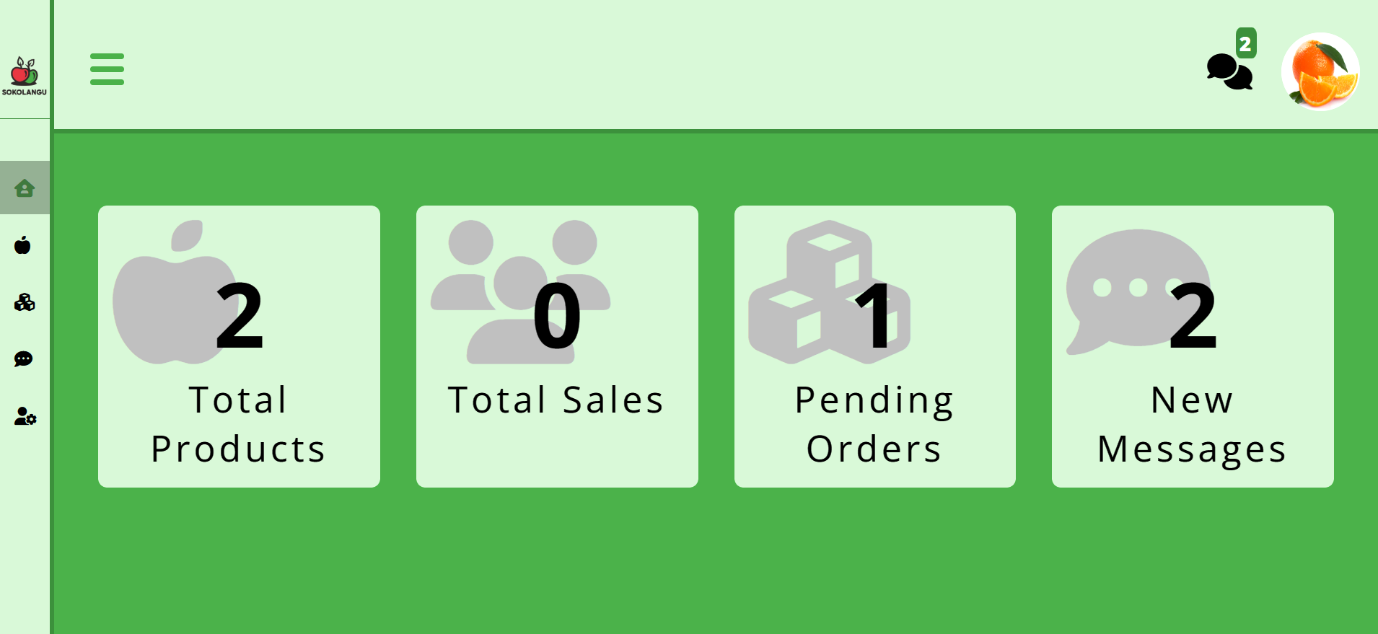


Figure 7: Farmer Dashboard UI

## ****5.5 System Security Design****

Security was a **major priority** in the development of **Soko Langu**. The following measures were implemented:

**1. Authentication & Authorization**  
✅ Passwords stored using **bcrypt hashing**.  
✅ Email and username enforced as **unique identifiers** to prevent duplication.  
✅ **Session-based authentication** for secure user access.

**2. Data Validation & Input Sanitization**  
✅ **SQL Injection Prevention** – All inputs sanitized using mysqli\_real\_escape\_string().  
✅ **Cross-Site Scripting (XSS) Protection** – Prevented by filtering input before displaying on the frontend.

**3. Secure File Uploads**  
✅ **Restricted file types** to only allow images (jpg, jpeg, png).  
✅ **Limited file size** to **5MB** to prevent server overload.  
✅ **Stored images in a secure folder** outside public access.

**4. User Activity Logging**  
✅ Recorded **user registration, logins, and key transactions** for security audits.

**5. Access Control**  
✅ **Role-based access control (RBAC)** – Farmers, Consumers, and Admin had **different permissions.**✅ **Consumers could only browse and purchase**, while **Farmers could list and manage products**.

## ****5.6 Summary****

This chapter described the **technical aspects of Soko Langu’s design**, covering **system architecture, database structure, UI components, and security measures**. The **three-tier architecture** ensured modularity, while **security features like password hashing, data validation, and access control** safeguarded system integrity.

# ****CHAPTER SIX:****

# ****SYSTEM IMPLEMENTATION****

This chapter details the **development and deployment of the Soko Langu Market Trade and Marketing Hub System**. It covers the **coding standards, development tools, system integration, testing processes, and deployment procedures.**

## ****6.1 Introduction****

System implementation involves **coding, integrating different modules, testing, and deploying** the final product. The **Soko Langu System** was developed using **React.js for the frontend, PHP for the backend, and MySQL for database management**.

This chapter outlines the **development approach, tools used, system integration, testing procedures, and deployment strategies.**

## ****6.2 Coding Standards and Guidelines****

To ensure **code readability, maintainability, and efficiency**, the following standards were followed:

**1. Naming Conventions**  
✅ Variables and functions were named using **camelCase** in JavaScript (orderTotal) and **snake\_case** in PHP (order\_total).  
✅ Database table names used **lowercase with underscores** (products\_table).

**2. Code Structuring**  
✅ **Frontend** was divided into **components** (Header.jsx, ProductView.jsx).  
✅ **Backend** had separate **API endpoints** (getOrders.php, getProducts.php).

**3. Security Best Practices**  
✅ **User authentication was implemented** with **password hashing (bcrypt)**.  
✅ **SQL queries were parameterized** to prevent **SQL injection attacks**.

**4. Comments and Documentation**  
✅ Each function and API endpoint was **documented** for future reference.

## ****6.3 Development Tools and Environment****

The system was built using the following tools:

|  |  |
| --- | --- |
| ****Tool**** | ****Purpose**** |
| **React.js** | Frontend development (User Interface) |
| **PHP** | Backend development (Server-side logic) |
| **MySQL** | Database management system |
| **Axios** | API requests between frontend and backend |
| **CSS** | Styling and responsive design |
| **phpMyAdmin** | Database management interface |
| **XAMPP** | Local server for PHP and MySQL |
| **Postman** | API testing tool |
| **GitHub** | Version control and collaboration |

Table 10: Development Tools and Environments

The **system was developed in a local environment using XAMPP** before being deployed.

## ****6.4 Integration and Testing****

6.4.1 System Integration  
After coding, the system’s modules were integrated in a **step-by-step approach**:  
✅ **Frontend and backend integration** was done using **RESTful APIs**.  
✅ **Database integration** was tested to ensure smooth **data storage and retrieval**.

6.4.2 Testing Methods  
The following testing techniques were used to ensure **system stability and functionality:**

✅ **Unit Testing** – Individual components (e.g., Product Listing, Order Processing) were tested separately.  
✅ **Integration Testing** – Verified smooth **communication between frontend, backend, and database.**✅ **User Testing** – Farmers and consumers tested the system to identify usability improvements.

### ****6.4.3 Test Cases and Results****

|  |  |  |  |
| --- | --- | --- | --- |
| ****Test Case**** | ****Expected Result**** | ****Actual Result**** | ****Status**** |
| User Registration | New users should register successfully | Successfully registered | ✅ Pass |
| Login Authentication | Valid users should log in | Successfully logged in | ✅ Pass |
| Product Upload | Farmers should add products | Products listed correctly | ✅ Pass |
| Order Placement | Consumers should place orders | Orders stored in database | ✅ Pass |
| Messaging Feature | Users should exchange messages | Messages delivered correctly | ✅ Pass |

Table 11: Test cases and Results

## ****6.5 Deployment****

6.5.1 Local Deployment  
Before launching online, the system was first tested in a **local environment using XAMPP**.

6.5.2 Online Hosting  
The system can be deployed on a **web hosting service** supporting **PHP and MySQL**, such as:  
✅ **InfinityFree** (for free hosting)  
✅ **TrueHost Kenya**  
✅ **Hostinger** (for premium hosting)

## ****6.6 Post-Implementation Review****

After deployment, the system was evaluated based on **performance, usability, and security**:

✅ **Performance:** The system **loaded pages quickly** and handled **multiple users efficiently**.  
✅ **Usability:** Farmers and consumers **easily navigated the platform**, and feedback was positive.  
✅ **Security:** Authentication and data validation mechanisms worked as expected.

## ****6.7 Summary****

This chapter explained how **Soko Langu Market Trade and Marketing Hub System** was implemented, including **coding standards, tools, testing, deployment, and post-launch review.**

The system was successfully developed using **React.js, PHP, and MySQL**, tested for functionality, and prepared for deployment.

# ****CHAPTER SEVEN:****

# ****LIMITATIONS, CONCLUSIONS, AND RECOMMENDATIONS****

This chapter presents the **limitations of the Soko Langu Market Trade and Marketing Hub System**, the **conclusions drawn from the study**, and **recommendations for future improvements.**

## ****7.1 Introduction****

No system is entirely free from challenges. While **Soko Langu** successfully streamlined farmer-to-consumer trade, certain **technical, resource, and usability limitations** were encountered.

This chapter highlights these **limitations**, presents **recommendations** for future improvements, and concludes with the **overall impact and future prospects** of the system.

## ****7.2 Limitations****

The system faced **several constraints** during development and implementation:

### ****7.2.1 Technical Limitations****

✅ **Internet Dependency** – The system requires a **stable internet connection**, making it **less accessible in rural areas** with poor connectivity.  
✅ **No Mobile App** – The system was designed as a **web platform** and lacks a **dedicated mobile app** for farmers and consumers.  
✅ **Manual Payments** – Since **payment integration was not included**, transactions must be completed **offline**, reducing automation.

### ****7.2.2 Resource Limitations****

✅ **Limited Hosting Resources** – The system was developed in a **local environment**, and deploying it to a **live server** requires **funding for domain and hosting services**.  
✅ **Farmer Training Needs** – Some farmers may **lack digital literacy**, requiring **training** before they can efficiently use the system.

### ****7.2.3 Data Limitations****

✅ **No Automated Inventory Management** – Farmers must **manually update product availability,** which may lead to **delays in reflecting stock changes**.  
✅ **Limited Data Insights** – The system lacks **advanced analytics** for **market trends and sales tracking,** which could benefit farmers.

### ****7.2.4 User Limitations****

✅ **Adoption Challenges** – Some farmers and consumers may be **hesitant to adopt online trading**, preferring **traditional methods**.  
✅ **Trust Issues** – Since payments are handled **offline, trust between buyers and sellers** remains a challenge.

## ****7.3 Recommendations****

To improve the **Soko Langu System**, the following **enhancements are recommended**:

### ****7.3.1 Technical Recommendations****

✅ **Develop a Mobile App** – Creating an **Android/iOS app** would improve **accessibility and usability** for farmers and consumers.  
✅ **Integrate Online Payments** – Implementing **M-Pesa or card payments** would **streamline transactions and improve security.**✅ **Optimize for Low Connectivity** – Adding **offline functionality** (such as SMS-based order confirmations) would **enhance accessibility** in rural areas.

### ****7.3.2 Resource Recommendations****

✅ **Host on a Reliable Server** – Deploying the system on a **cloud-based server** would improve **availability and scalability**.  
✅ **Organize Digital Training** – Conducting **farmer training sessions** on system usage would **increase adoption and efficiency.**

### ****7.3.3 Data Recommendations****

✅ **Automate Inventory Updates** – Implementing **real-time stock updates** would reduce errors and improve user experience.  
✅ **Add Sales Analytics** – Providing **data insights on consumer demand trends** would help **farmers make informed decisions.**

### ****7.3.4 User Recommendations****

✅ **Enhance User Verification** – Implementing **KYC (Know Your Customer) verification** would help **build trust** between buyers and sellers.  
✅ **Create a Review System** – Allowing consumers to **rate and review farmers** would improve **market transparency and credibility.**

## ****7.4 Conclusion****

### ****7.4.1 Project Achievement****

The **Soko Langu Market Trade and Marketing Hub System** successfully **bridged the gap between farmers and consumers** by providing a **direct and transparent agricultural marketplace.** The system:  
✅ Allowed **farmers to list and sell produce directly to consumers**.  
✅ Enabled **consumers to search for farm products based on region.**  
✅ Integrated **messaging features for order coordination.**  
✅ Provided an **easy-to-use platform for agricultural trade.**

### ****7.4.2 Future Prospects****

The system can be **expanded in the future** by:  
✅ Developing **a mobile application** for easier access.  
✅ Introducing **AI-powered recommendations** for consumers.  
✅ Partnering with **agricultural cooperatives** to increase farmer adoption.

### ****7.4.3 Final Thoughts****

The **Soko Langu System** demonstrated **how digital platforms can revolutionize agricultural trade** by ensuring **fair pricing, transparency, and market efficiency**. Future improvements will focus on **enhancing security, automation, and scalability** to maximize the system’s impact.

# ****REFERENCES****

This final section includes **references to sources used** in the research

1. **Books & Articles:**
   * Smith, J. (2020). E-commerce in Agriculture: A Digital Transformation Guide. Oxford University Press.
   * Anderson, R. (2018). Agricultural Market Trends and Digital Solutions. Cambridge University Press.
2. **Web Sources & Reports:**
   * Food and Agriculture Organization (FAO). (2023). Digital Agriculture: Market Challenges and Opportunities. Retrieved from [www.fao.org](https://www.fao.org)
   * Kenya National Bureau of Statistics (KNBS). (2022). Agricultural Market Report 2022. Retrieved from www.knbs.or.ke
3. **Software & Technologies Used:**
   * React.js Documentation. (n.d.). Retrieved from <https://react.dev>
   * PHP & MySQL Guide. (2023). Retrieved from <https://www.php.net>

# ****APPENDICES****

This section provides **supporting materials** related to the **Soko Langu System**, such as **database structure, system screenshots, and sample code.**

## ****Appendix A: Database Structure****

The **database schema** used in the system are as follows:

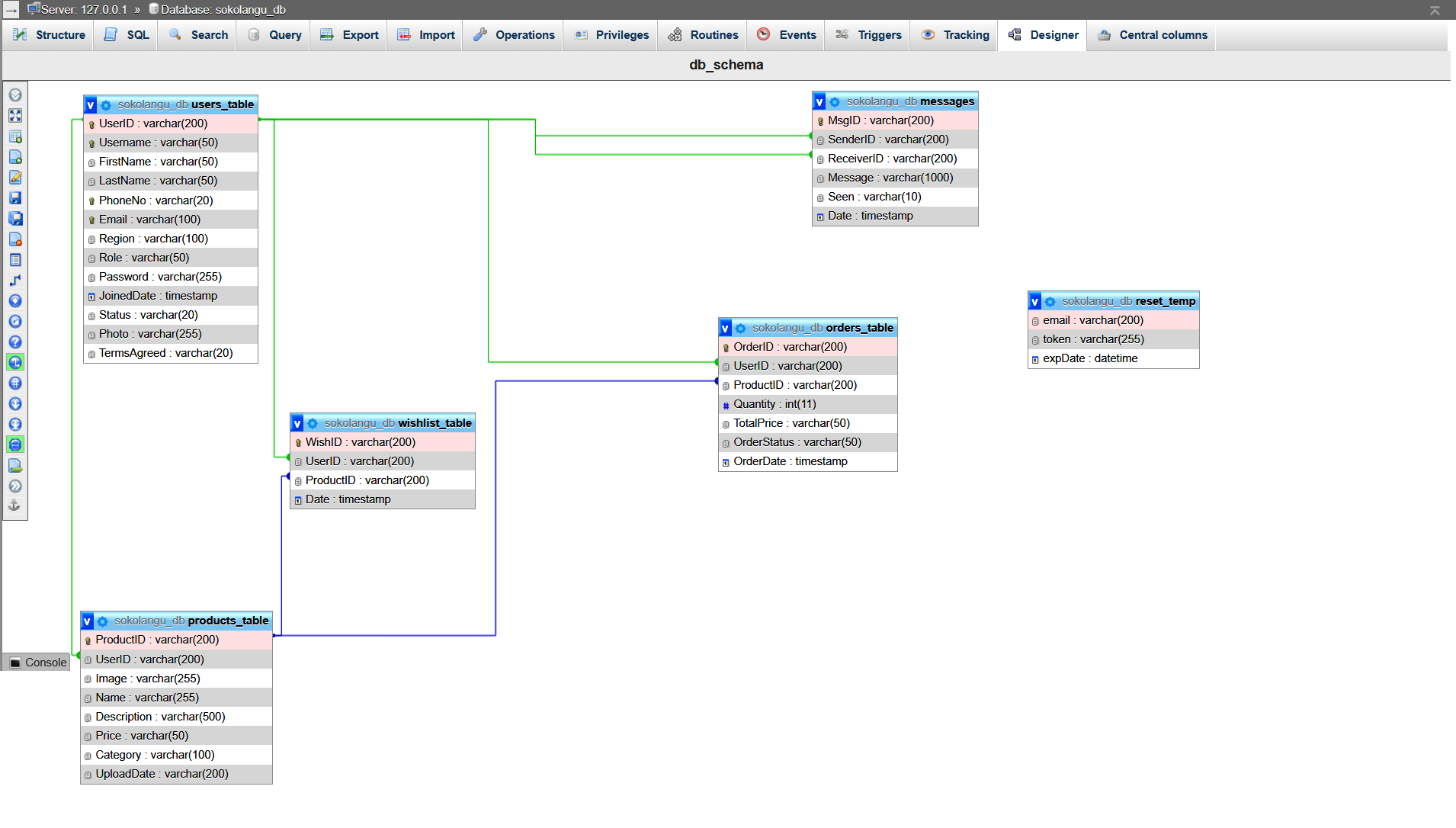


Figure 8: Database Schema

## ****Appendix B: Code Samples****

**User Registration (PHP)**

<?php

    include "config.php";

    $username = $\_POST['username'];

    $email = $\_POST['email'];

    $password = $\_POST['password'];

    $first\_name = $\_POST['first\_name'];

    $last\_name = $\_POST['last\_name'];

    $phone\_no = $\_POST['phone\_no'];

    $region = $\_POST['region'];

    $role = $\_POST['role'];

    $terms\_check = $\_POST['terms\_check'];

    function generateUserID() {

        $randomPart = generateRandomString(8);

        return "user\_" . $randomPart;

    }

    function generateRandomString($length) {

        $characters = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';

        $charLength = strlen($characters);

        $randomString = '';

        for ($i = 0; $i < $length; $i++) {

            $randomString .= $characters[rand(0, $charLength - 1)];

        }

        return $randomString;

    }

    if (!empty($username) && !empty($email) && !empty($password) && !empty($first\_name) && !empty($last\_name) && !empty($phone\_no) && !empty($region) && $terms\_check) {

        $query = $conn->prepare("SELECT \* FROM `users\_table` WHERE username = (?) OR email = (?)");

        $query->bind\_param("ss", $username, $email);

        $query->execute();

        $result = $query->get\_result();

        $num = mysqli\_num\_rows($result);

if ($num == 1) {

            echo "Username/Email Exists";

        } else {

            $user\_id = generateUserID();

            date\_default\_timezone\_set('Africa/Nairobi');

            $joined\_date = date('Y-m-d H:i:s');

            $terms\_check = "Agreed";

            $password = password\_hash($password, PASSWORD\_DEFAULT); // Hash password for security

            $photo = "avatar.png"; //Default photo of the new user

            $query = $conn->prepare("INSERT INTO `users\_table` (UserID, Username, FirstName, LastName, PhoneNo, Email, Region, Role, Password, JoinedDate, Photo, TermsAgreed)

                                     VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)");

            $query->bind\_param("ssssssssssss", $user\_id, $username, $first\_name, $last\_name, $phone\_no, $email, $region, $role, $password, $joined\_date, $photo, $terms\_check);

            $query->execute();

            if ($query) {

                echo 1; // Success response

            } else {

                echo "Error, Please Try Again!";

            }

        }

    } else {

        echo "All fields Required!";

    }

?>

**Fetching Products (React.js - Axios Request)**

useEffect(() => {

        //Axios to get products data

        if (LoggedIn) {

            axios({

                method: "get",

                params: { user\_id: user },

                url: api\_url + "/consumer/getProducts.php",

            })

                .then((response) => {

                    setProductsData(response.data);

                    setFilteredProductsData(response.data); // Initialize the filtered data with all products

                })

                .catch((e) => {

                    console.log(e);

                });

        } else {

            axios({

                method: "get",

                url: api\_url + "/consumer/productsData.php",//fetch All by default regardless of region

            })

                .then((response) => {

                    setProductsData(response.data);

                    setFilteredProductsData(response.data); // Initialize the filtered data with all products

                })

                .catch((e) => {

                    console.log(e);

                });

        }

        setTimeout(() => {

            setIsLoading(false);

        }, 3000);

    }, [])

## ****Appendix C: System Screenshots****

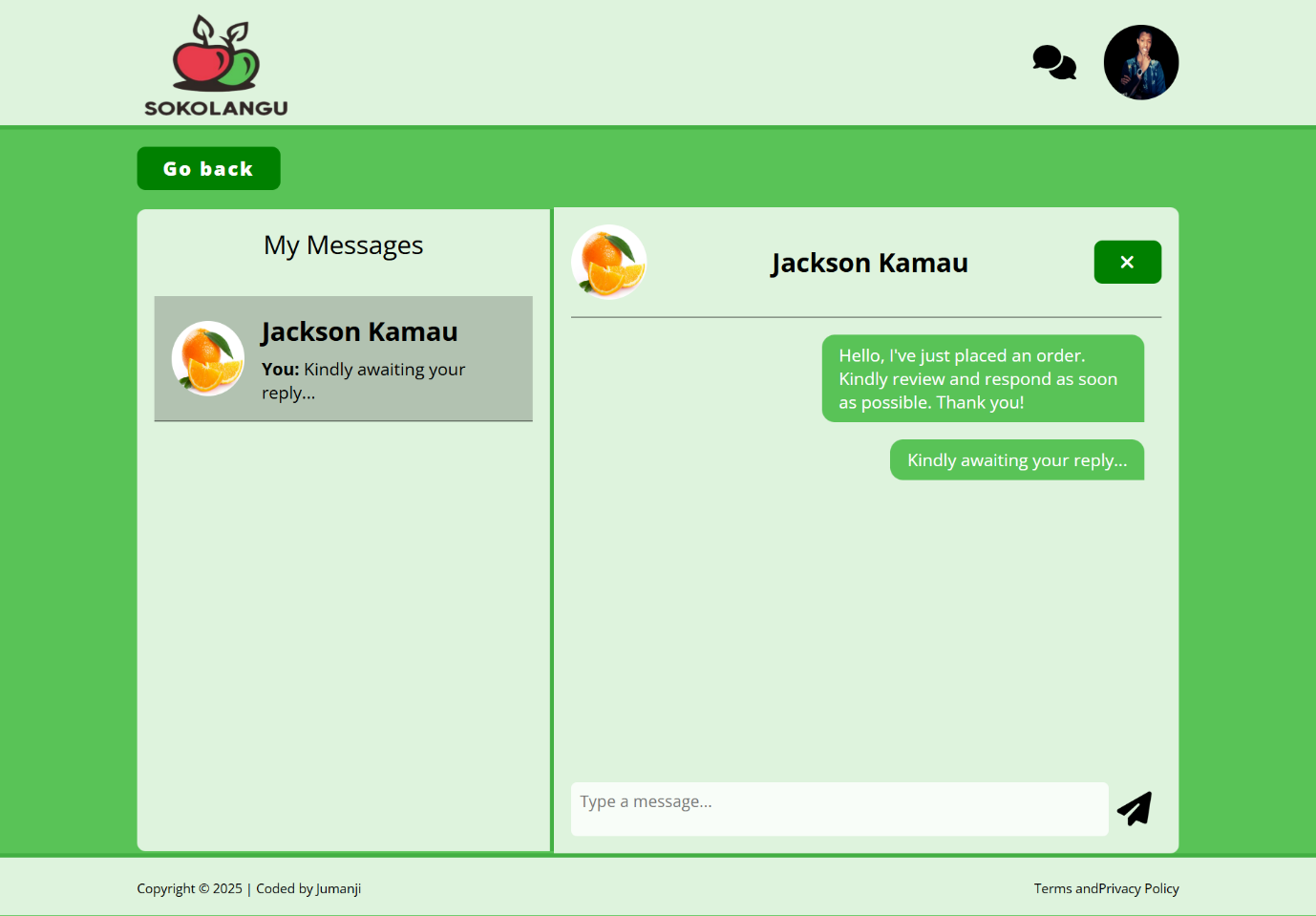


Figure 9: Consumer Messaging Screen

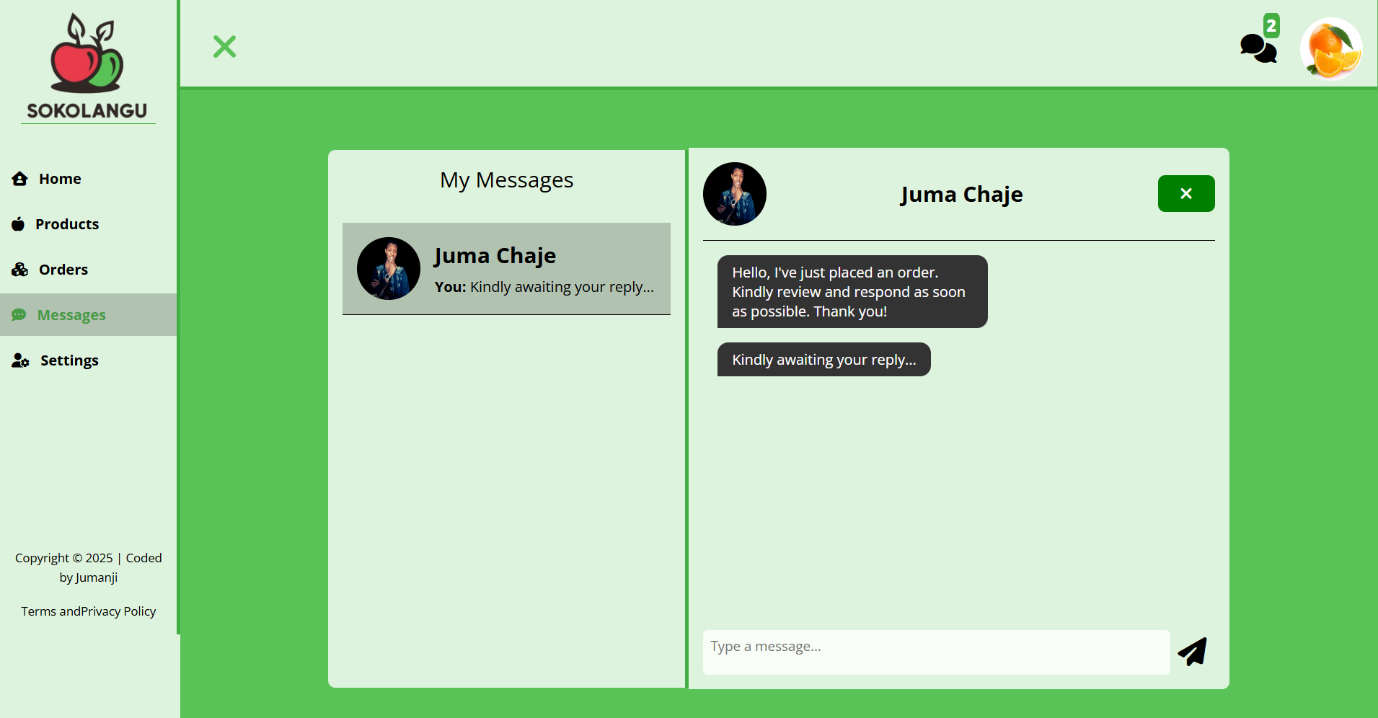


Figure 10: Farmer Messaging Screen

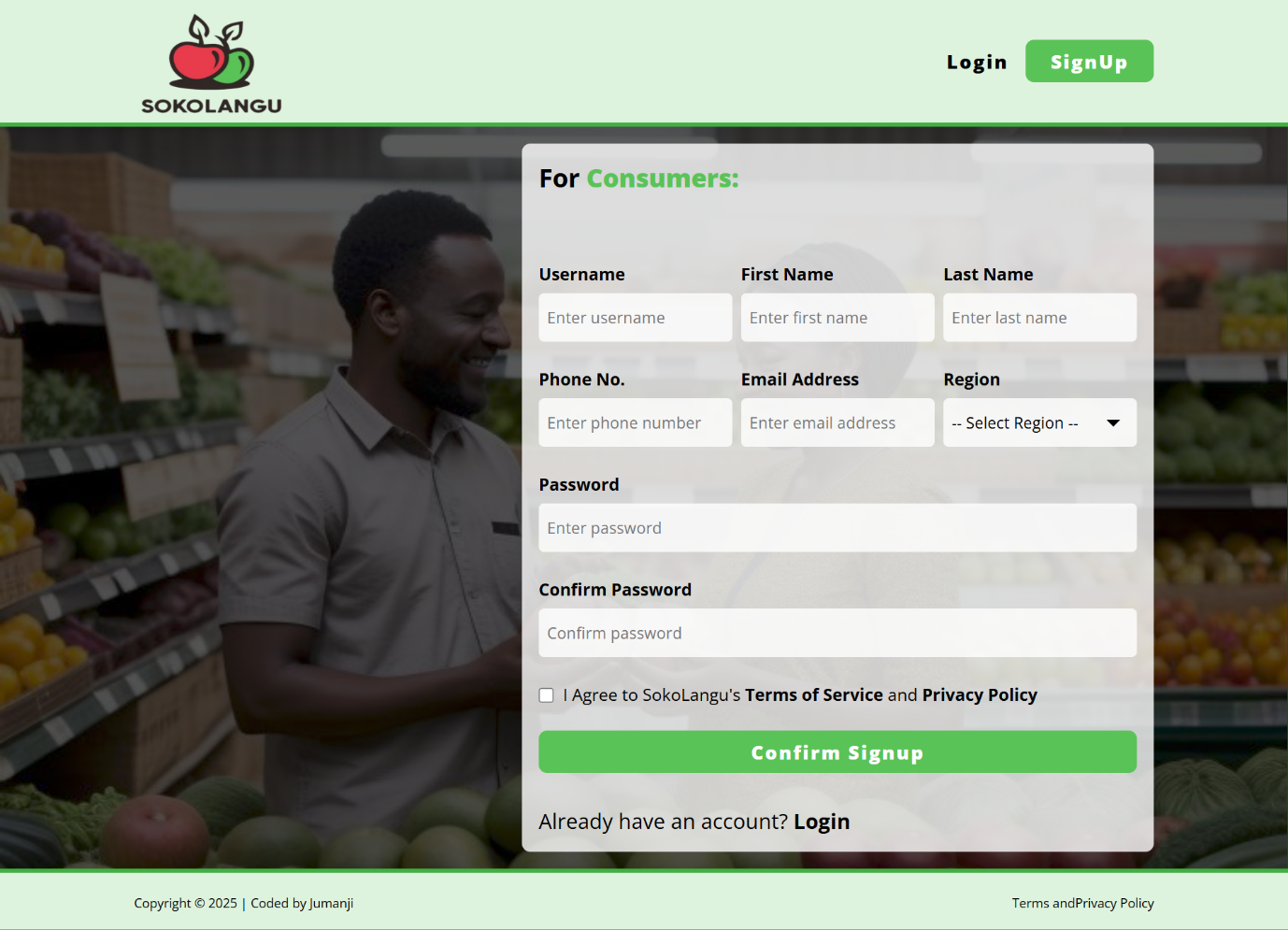


Figure 11: Consumer Signup Screen

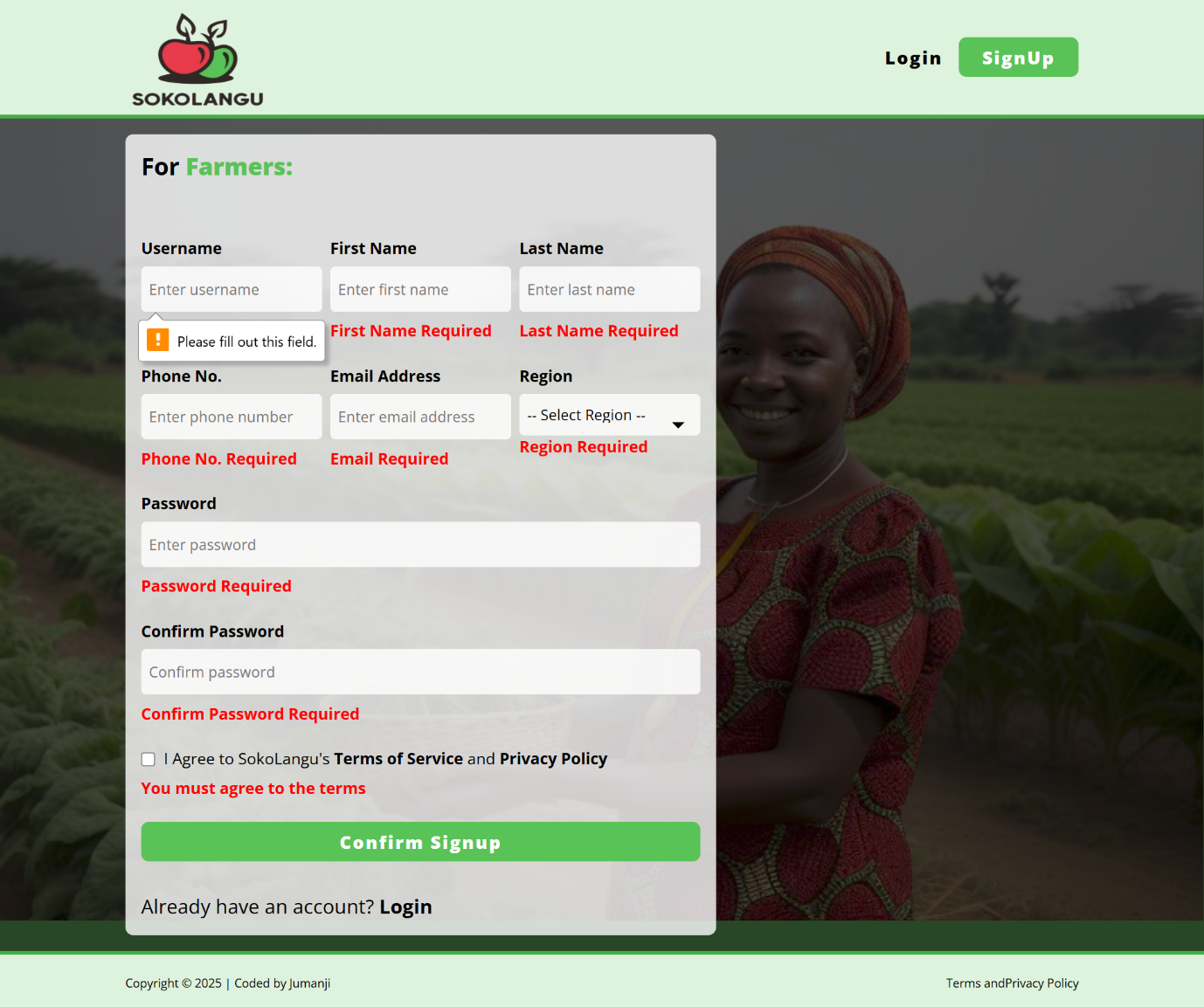


Figure 12: Farmer Signup Screen with Validation

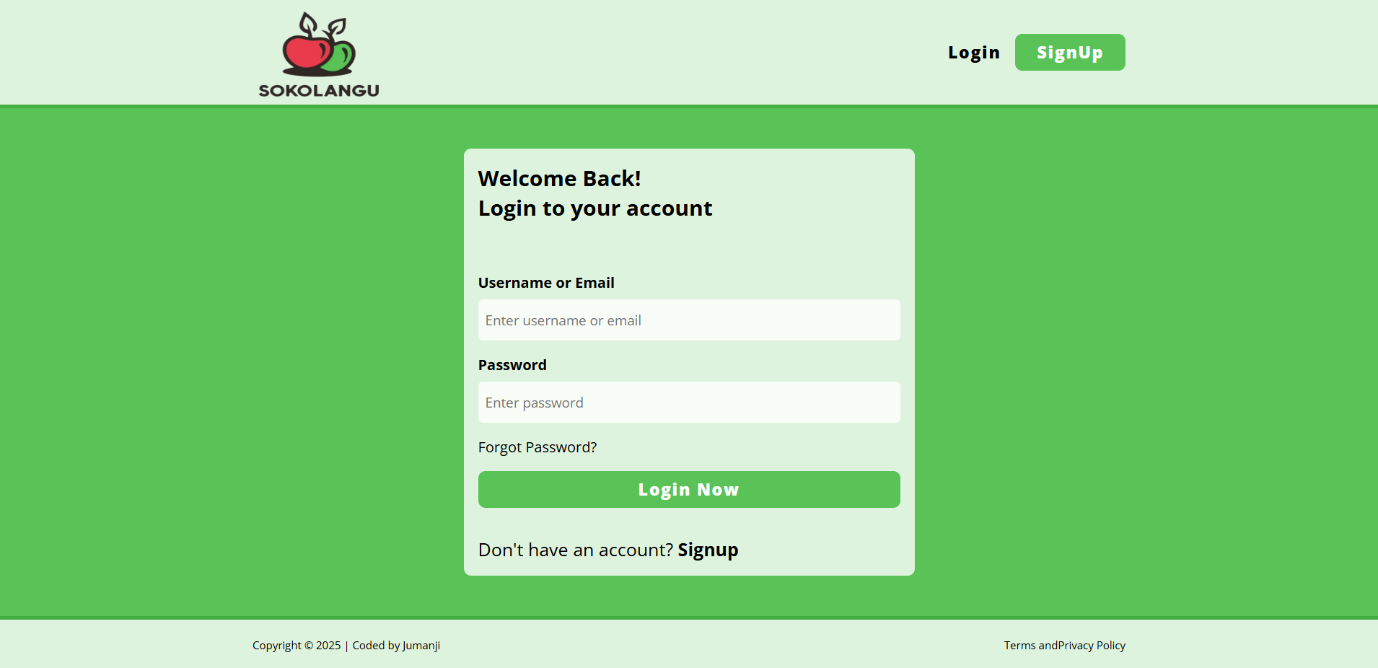


Figure 13: Login Screen

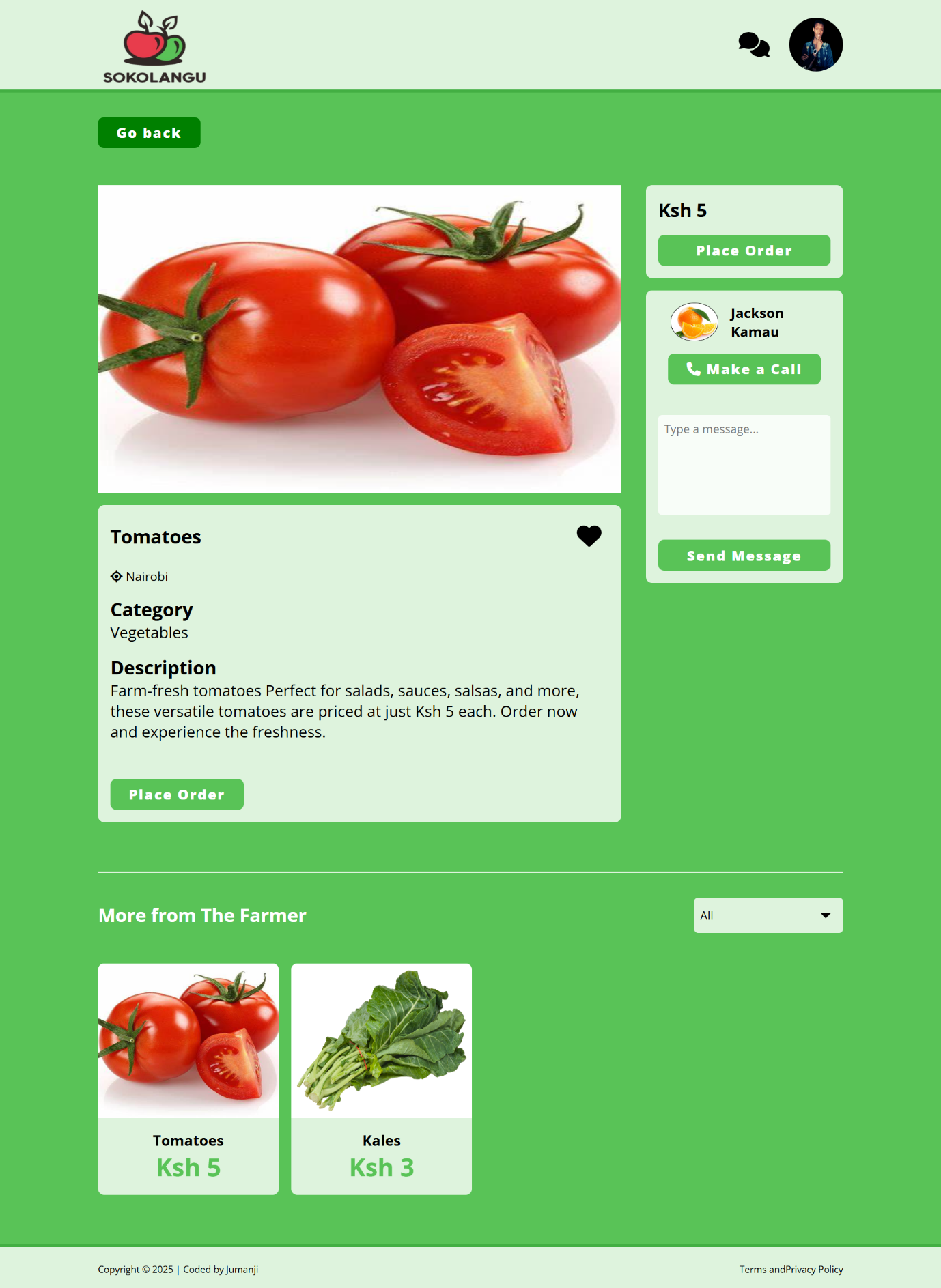


Figure 14: ProductView Screen

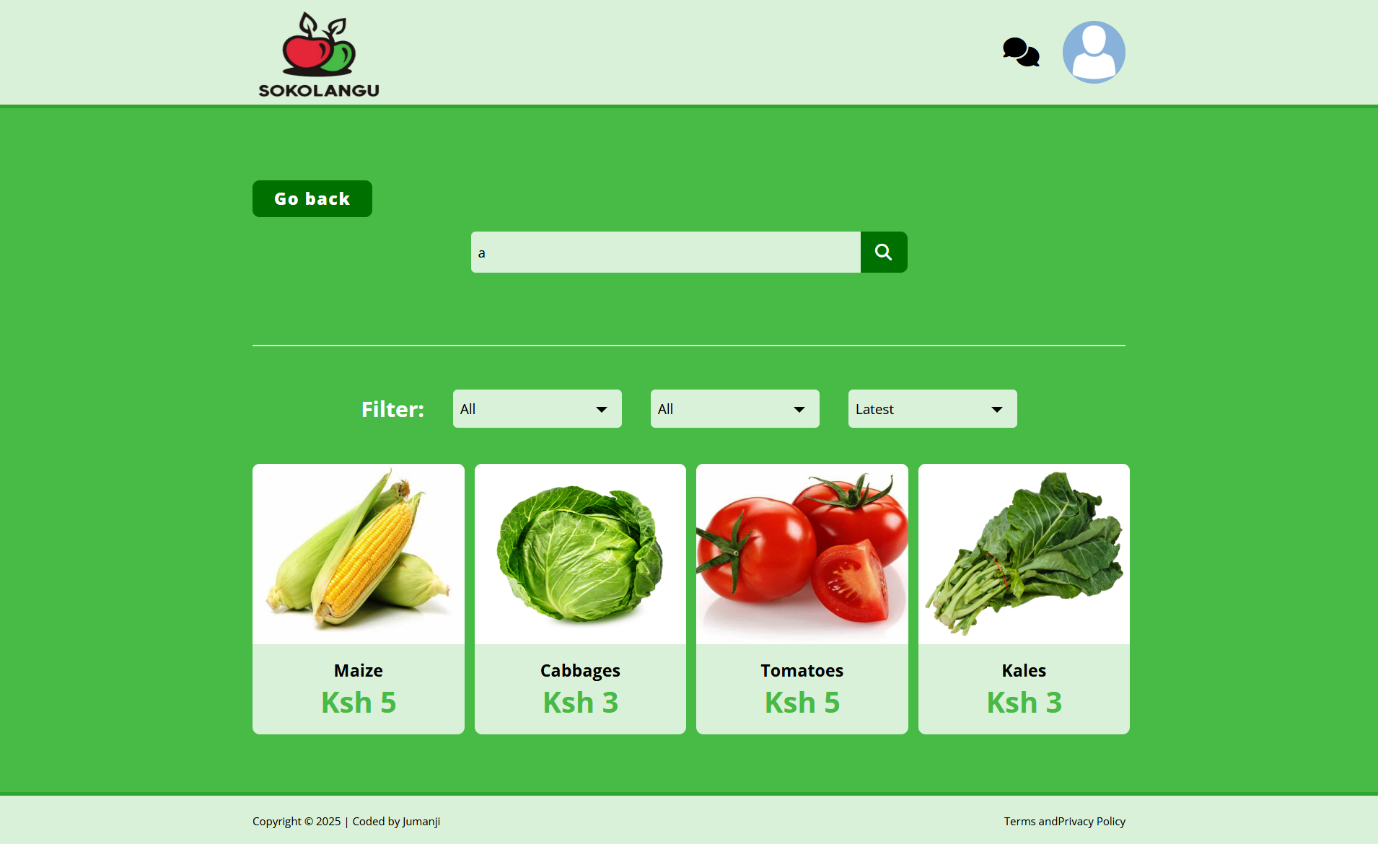


Figure 15: Search Filter Screen



Figure 16: Admin Dashboard Screen