EFFECTIVE MANAGEMENT FOR BLOCKCHAIN- BASED AGRI-FOOD SUPPLY CHAINS

Submitted by

ARUNKUMAR J (211420104028)

VIGNESH S (211420104342)

DHANUSHRAJ R C (211420104060)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonmous Institution, Affiliated to Anna University, Chennai)

APRIL 2024

PANIMALAR ENGINEERING COLLEGE

(An Autonmous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **EFFECTIVE MANAGEMENT FOR BLOCKCHAIN BASED AGRI-FOOD SUPPLY CHAINS** is the bonafide work of "Arunkumar J Vignesh s Dhanush R C**"** who carried out the project work under my supervision.

Signature of the HOD with date	**Signature of the Supervisor with date**
DR L.JABASHEELA M.E.,Ph.D.,	**Dr.M.S Vinmathi M.E.,Ph.D.,**
Professor and Head,	**Professor**
Department of Computer Science and Engineering,	Department of Computer Science and Engineering,
Panimalar Engineering College,	Panimalar Engineering College,
Chennai - 123	Chennai – 123

Submitted for the Project Viva – Voce examination held on _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION BY THE STUDENT

We Arunkumar J [211420104028] Vignesh S [211420104342] Dhanush [211420104060] hereby declare that this project report titled "EFFECTIVE MANAGEMENT FOR BLOCKCHAIN- BASED AGRI-FOOD SUPPLY CHAINS", under the guidance of Dr. N.Pughazendi is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

Arunkumar J

Vignesh S

Dhanush R C

ACKNOWLEDGEMENT

Our profound gratitude is directed towards our esteemed Secretary and Correspondent, **Dr. P. CHINNADURAI, M.A., Ph.D**., for his benevolent words and fervent encouragement. His inspirational support proved instrumental in galvanizing our efforts, ultimately contributing significantly to the successful completion of this project

We want to express our deep gratitude to our Directors, **Tmt. C. VIJAYARAJESWARI, Dr. C. SAKTHI KUMAR, M.E., Ph.D., and Dr. SARANYASREE SAKTHI KUMAR,B.E., M.B.A., Ph.D.,** for graciously affording us the essential resources and facilities for undertaking of this project.

Our gratitude is also extended to our Principal, **Dr. K. MANI, M.E., Ph.D.,** whose facilitation proved pivotal in the successful completion of this project.

We express my heartfelt thanks to **Dr. L. JABASHEELA M.E., Ph.D.,** Head of the Department of Computer Science and Engineering, for granting the necessary facilities that contributed to the timely and successful completion of project.

We would like to express our sincere thanks to Project Coordinator and Project **Dr.N.PUGHAZENDI,M.E., Ph.D.,** Guide **Dr M.S. Vimathi M.E., Ph.D.,** and all the faculty members of the Department of CSE for their unwavering support for the successful completion of the project.

ARUNKUMAR J

VIGNESH S

DHANUSHRAJ RC

PROJECT COMPLETION CERTIFICATE

ABSTRACT

In agri-food supply chains (ASCs), ensuring product safety and profitability are paramount concerns for both consumers and farmers. However, the complex and dynamic nature of ASCs poses significant challenges to effective traceability and management of agri-food products. Existing solutions often fall short of meeting the requirements of traceability and management in ASCs. To address these challenges, this research presents a novel approach that integrates blockchain technology for product traceability and deep reinforcement learning (DR-SCM) for supply chain management.

The proposed blockchain-based ASC framework aims to provide a robust and decentralized solution for product traceability. By leveraging blockchain technology, a transparent and tamper-proof record of each agri-food product's journey through the supply chain is maintained. This ensures accountability and transparency, thereby enhancing consumer confidence in product safety and quality. Furthermore, the decentralized nature of blockchain provides a secure platform for storing and sharing traceability data, mitigating the risk of data tampering or manipulation.

In parallel, the DR-SCM method is introduced to optimize decision-making in agri-food supply chains. Deep reinforcement learning techniques are employed to train agents to make sequential decisions regarding production, inventory management, and distribution. By observing the state of the supply chain and taking appropriate actions, the system learns to maximize profitability over time. Extensive simulation experiments are conducted to evaluate the performance of the proposed method under various ASC environments.

APPENDIX 3

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In recent years, the problems of agri-food safety and farmer income have received widespread attention. The issues of agri-food safety may occur in each part of agri-food supply chains (ASCs), while inefficient management strategies of ASCs would lead to low profits. However, many factors may constrain the normal work of ASCs. First, due to the complex structure of ASCs, it is hard to record the full circulation information of agri-food products while ensuring that the information will never be tampered with. Second, the shift of consumer preferences has become the main barrier of precisely determining the production and storage of agri-food products with the consideration of profit maximization. Such uncertainties and dynamics undoubtedly increase the toughness of designing an efficient ASC framework.

To address these problems, the effective traceability and management for agri-food products in ASCs have become urgently necessary. On one hand, to guarantee the agri-food safety, the information of agri-food products in ASCs including production, processing, storage, distribution, and retail should be collected and recorded when establishing a mechanism of product traceability. However, most of the traditional traceability solutions of ASCs rely on a centralized system maintained by a trusted third party, which may suffer from the potential single-node failure and security threats such as data tampering and information leakage. Blockchain, a distributed, append-only, and tamper-proof ledger, offers an effective architecture for reliable transactions on the Bitcoin network without the control of a centralized third party.

Each information recorded in a blockchain should be verified by the majority of participants to reach a global consensus, which ensures the information source with auditability and transparency. Moreover, there is no need for a blockchain-based traceability solution to connect to a remote cloud data center because it only requires a stable connection among adjacent participants. Therefore, the blockchain technology can be used to realize a reliable product traceability in supply chains, which has recently become a new research direction and attracted many research interests., respectively.

1.1 PROBLEM DEFINITION

The traditional agricultural supply chain involves multiple intermediaries, resulting in inefficiencies, lack of transparency, and increased costs for both farmers and consumers. Farmers often face challenges in accessing fair markets, while consumers may struggle to trace the origins and quality of the produce they purchase. To address these issues, there's a need for a Farmer-to-Customer Direct Crop Sale platform leveraging blockchain technology. The following problems need to be addressed Current agricultural supply chains lack transparency, making it difficult for consumers to trace the journey of produce from farm to table. Farmers also face challenges in proving the authenticity and quality of their product The involvement of multiple intermediaries in the supply chain leads to increased costs for both farmers and consumers. Farmers often receive low prices for their produce due to the profit margins taken by intermediaries Small-scale and local farmers struggle to access fair markets for their produce. They often rely on middlemen who offer low prices, leading to reduced profitability.Consumers often face uncertainties regarding the quality and freshness of the produce they purchase. Lack of transparent information about farming practices and handling procedures contributes to this issue. Traditional payment systems in agricultural transactions may be slow and prone to errors. Farmers may face delays in receiving payments, affecting their cash flow.Without robust traceability mechanisms, it's challenging to identify the origin of agricultural products and ensure adherence to food safety standards and regulations.

CHAPTER 2

LITERATURE REVIEW

1.Title: Blockchain and Its Impacts on Agri-Food Supply Chain Network Management

Author: Michael Paul Kramer, Linda Bitsch and Jon Hanf

ABSTRACT: Blockchain is an emerging meta-technology and considered a new institutional technology with the potential to change the governance of vertically integrated food supply chains. This paper investigates the effects on coordination mechanisms in vertically cooperating agri-food networks that result from the implementation of different blockchain technology platform types (BCTPT). The research is based on an extensive literature overview and exploratory use cases of BCTPT implementations in the agri-food industry which are presented to illustrate the applicability of the findings.

2.Title: Blockchain-Based Agri-Food Supply Chain: A Complete Solution

Author: Affaf Shahid; Ahmad Almogren; Nadeem Javaid; Fahad Ahmad Al-Zahrani; Mansour Zuair

ABSTRACT: Supply chains are evolving into automated and highly complex networks and are becoming an important source of potential benefits in the modern world. At the same time, consumers are now more interested in food product quality. However, it is challenging to track the provenance of data and maintain its traceability throughout the supply chain network. The traditional supply chains are centralized and they depend on a third party for trading. These centralized systems lack transparency, accountability and auditability. In our proposed solution, we have presented a complete solution for blockchain-based Agriculture and Food (Agri-Food) supply chain.

3.Title : An agri-food supply chain traceability system for China based on RFID & blockchain technology

Author : Feng Tian

ABSTRACT: For the past few years, food safety has become an outstanding problem in China. Since traditional agri-food logistics pattern cannot match the demands of the market anymore, building an agri-food supply chain traceability system is becoming more and more urgent. In this paper, we study the utilization and development situation of RFID (Radio-Frequency IDentification) and blockchain technology first, and then we analyze the advantages and disadvantages of using RFID

4.Title: A peer-to-peer electronic cash system

Author: Satoshi Nakamoto

ABSTRACT: A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending.

5.Title: Food safety traceability system based on blockchain and EPCIS

Author: H. Feng, X. Wang, Y. Duan, J. Zhang, and X. Zhang

ABSTRACT: Raceability plays a vital role in food quality and safety management. Traditional Internet of Things (IoT) traceability systems provide the feasible solutions for the quality monitoring and traceability of food supply chains. However, most of the IoT solutions rely on the centralized server-client paradigm that makes it difficult for consumers to acquire all transaction information and to track the origins of products.

6.Title: Agri-Food Supply Chain Management

Author: C. Ganesh Kumar Pachayappan Murugaiyan G. Madanmohan

ABSTRACT: The purpose of this paper is to present a critical review of prior literature relating to agri-food supply chain management. An in-depth analysis has been carried out to identify the influential information from the literature. This paper has identified gaps to be explored about agricultural supply chain management (SCM) practices which may be used by researchers to enrich theory construction and practitioners may concentrate on establishing the extent and frontiers of agri-food SCM.

7.Title: Nutritional Quality and Safety Traceability System for China's Leafy Vegetable Supply Chain Based on Fault Tree Analysis and QR Code

Author: Y. Dong, Z. Fu, S. Stankovski, S. Wang, and X. Li

ABSTRACT: Leafy vegetables are consumed in most daily diets worldwide. As living standards improve, food quality, safety requirements, and nutrition are becoming increasingly important to consumers when purchasing leafy vegetables. This study proposes an evaluation and traceability method that can be used to track the nutritional quality of leafy vegetables. Employing the principles of the Hazard Analysis and Critical Control Point (HACCP) system combined with fault tree analysis (FTA).

8.Title: A meta-analysis of intercropping in Africa: Impacts on crop yield, farmer income, and integrated pest management effects

Author: J. Himmelstein, A. Ares, D. Gallagher, and J. Myers

ABSTRACT: Poverty and hunger in Africa are prevalent and will increase in absolute terms with population growth and continued land degradation. Therefore, there is a need for sustainable agricultural strategies, such as conservation agriculture (CA) and integrated pest management (IPM). Among CA practices, intercropping holds the promise of providing benefits to smallholders through increased crop yields and income as well as improved resource use.

CHAPTER 3

THEORETICAL BACKGROUND

3.1 IMPLEMENTATION ENVIROMENT

HARDWARE REQUIREMENT

System **:** Intel i3
Hard Disk **:** 40 GB
Floppy Drive **:** 1.44 Mb
Monitor **:** 15 VGA Colour
Mouse **:** Logitech
Ram **:** 512 Mb

SOFTWARE REQUIREMENT

Operating system **:** Windows XP
Coding Language **:** JAVA
Data Base **:** MYSQL
IDE **:** Netbeans IDE

BLOCKCHAIN TECHNOLOGIES:

Blockchain technology is a decentralized digital ledger system that enables the recording, verification, and storage of transactions across a network of computers. It was first introduced in 2008 by an anonymous person or group of people using the pseudonym Satoshi Nakamoto as a core component of the digital currency Bitcoin. However, its applications have since expanded far beyond cryptocurrencies.Here are the key components and concepts of blockchain technology:

Decentralization: Unlike traditional centralized systems where a single entity has control over data and transactions, blockchain operates on a decentralized network of computers (nodes). Each node maintains a copy of the entire blockchain ledger, and all transactions are collectively validated by the network.Immutable Ledger: Blockchain records transactions in blocks, which are cryptographically linked to form a continuous chain. Once a transaction is recorded on the blockchain, it cannot be altered or deleted. This feature ensures the integrity and immutability of the ledger, making blockchain a trusted source of truth.

Cryptographic Security: Blockchain uses cryptographic techniques to secure transactions and protect the integrity of the data. Each block contains a unique cryptographic hash of the previous block, creating a chain of blocks that is resistant to tampering and fraud.

Consensus Mechanisms: Consensus mechanisms are protocols used to achieve agreement among nodes on the validity of transactions and the order in which they are added to the blockchain. Popular consensus mechanisms include Proof of Work (PoW), Proof of Stake (PoS), and variants such as Delegated Proof of Stake (DPoS) and Practical Byzantine Fault Tolerance (PBFT).

Smart Contracts: Smart contracts are self-executing contracts with the terms of the agreement directly written into code.. Smart contracts run on blockchain platforms like Ethereum, enabling programmable and automated interactions without the need for intermediaries.

3.2 SYSTEM ARCHITECTURE

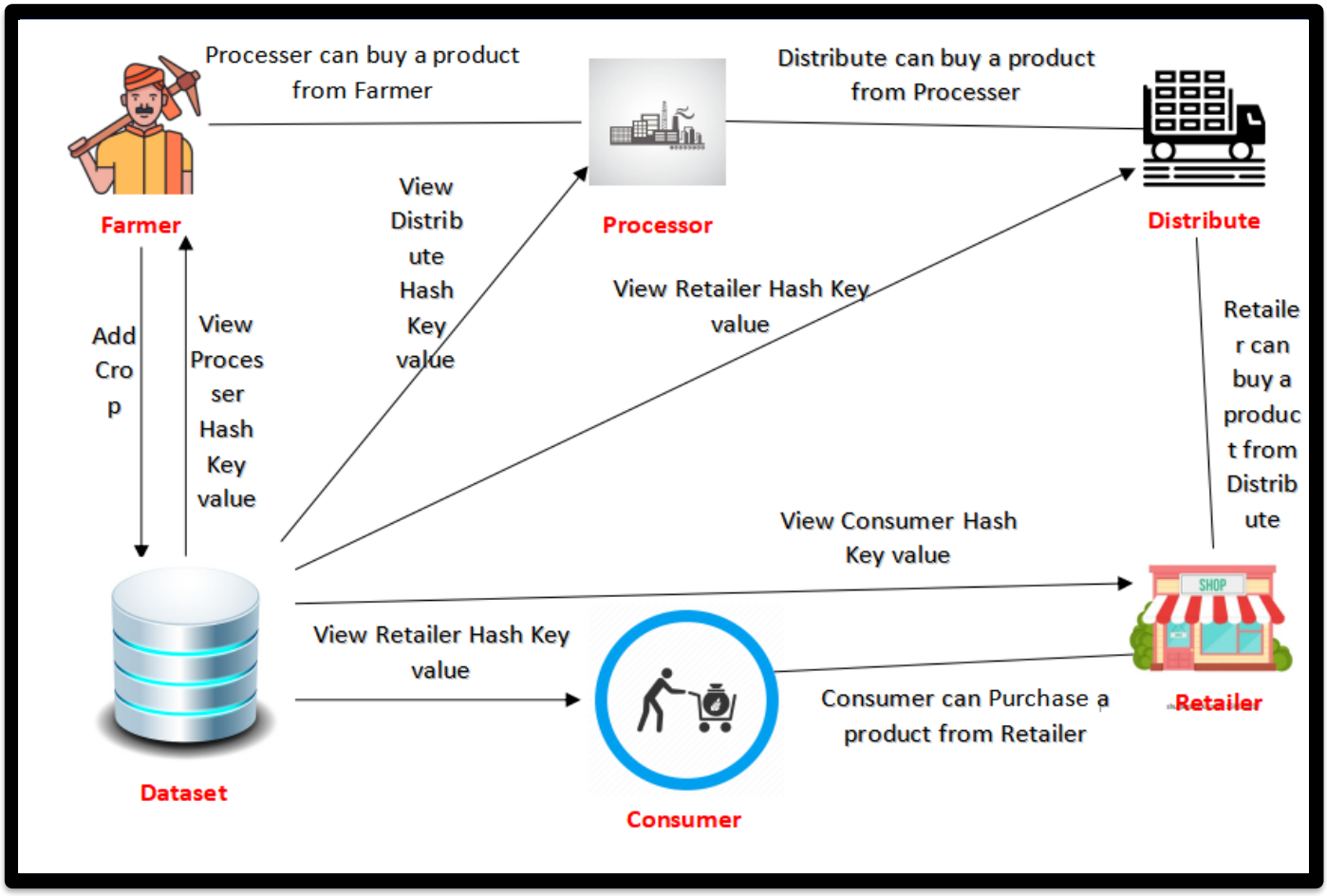


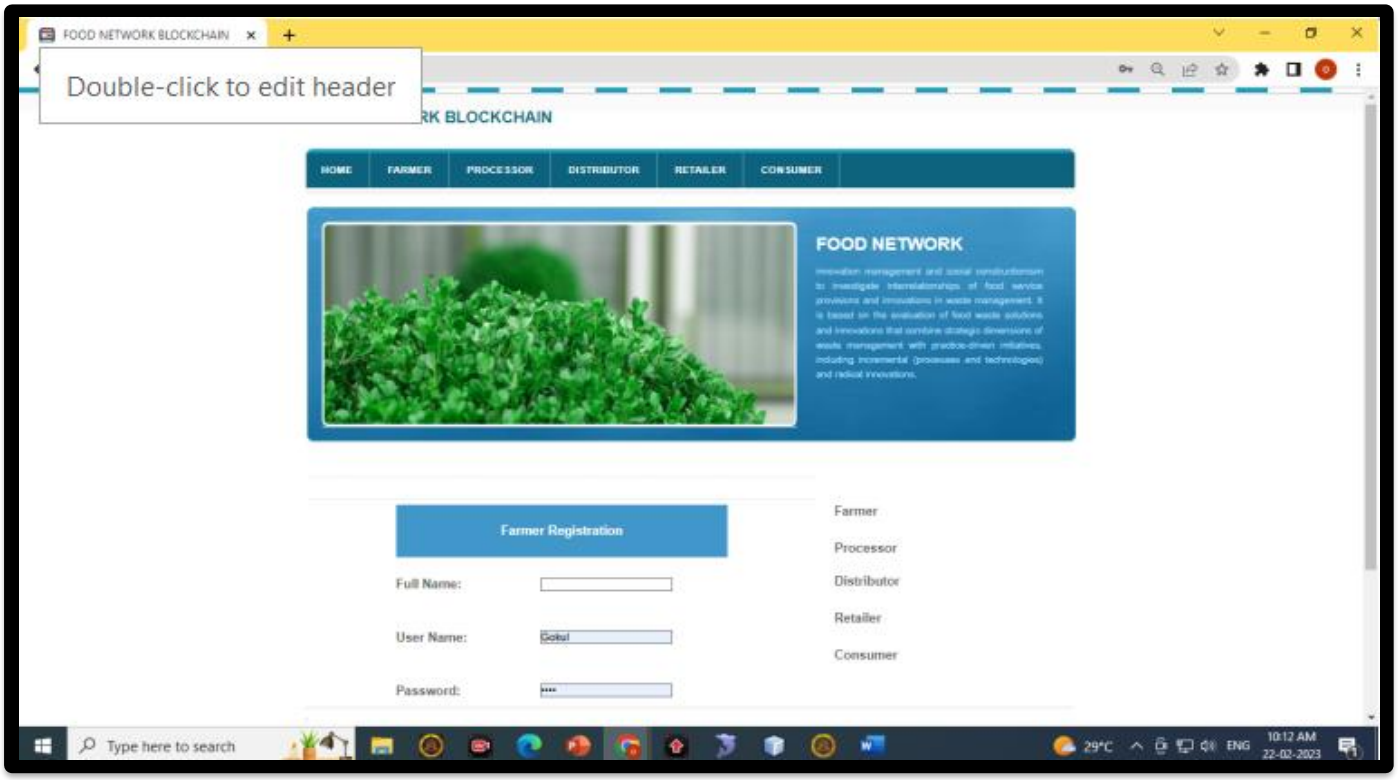
Fig 3.2.1 Architecture diagram for effectivemanagement for block chain based Agri food chain supply

3.3 PROPOSED METHADOLOGY

PROPOSED SYSTEM

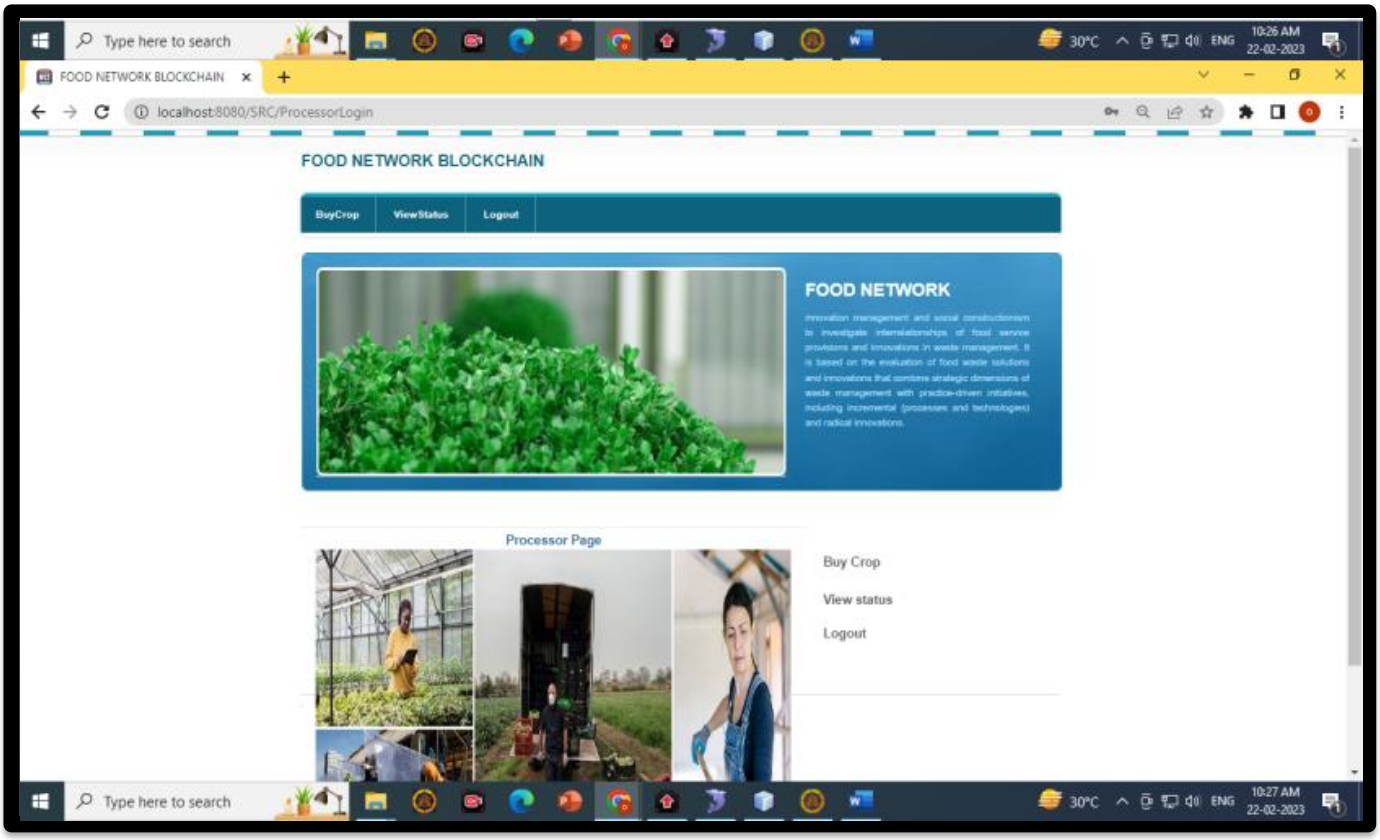
To evaluate how well the suggested DR-SCM method for managing ASCs performs by contrasting it with two established techniques: the computational cost of hashes with various numbers of blocks and Q-learning methods.

Displays the benefits (i.e. profits) of various ASC management strategies in various scenarios. With the exception of the heuristic method, the benefits of the various strategies all rise as the number of episodes rises.



3.3.2 Input Design

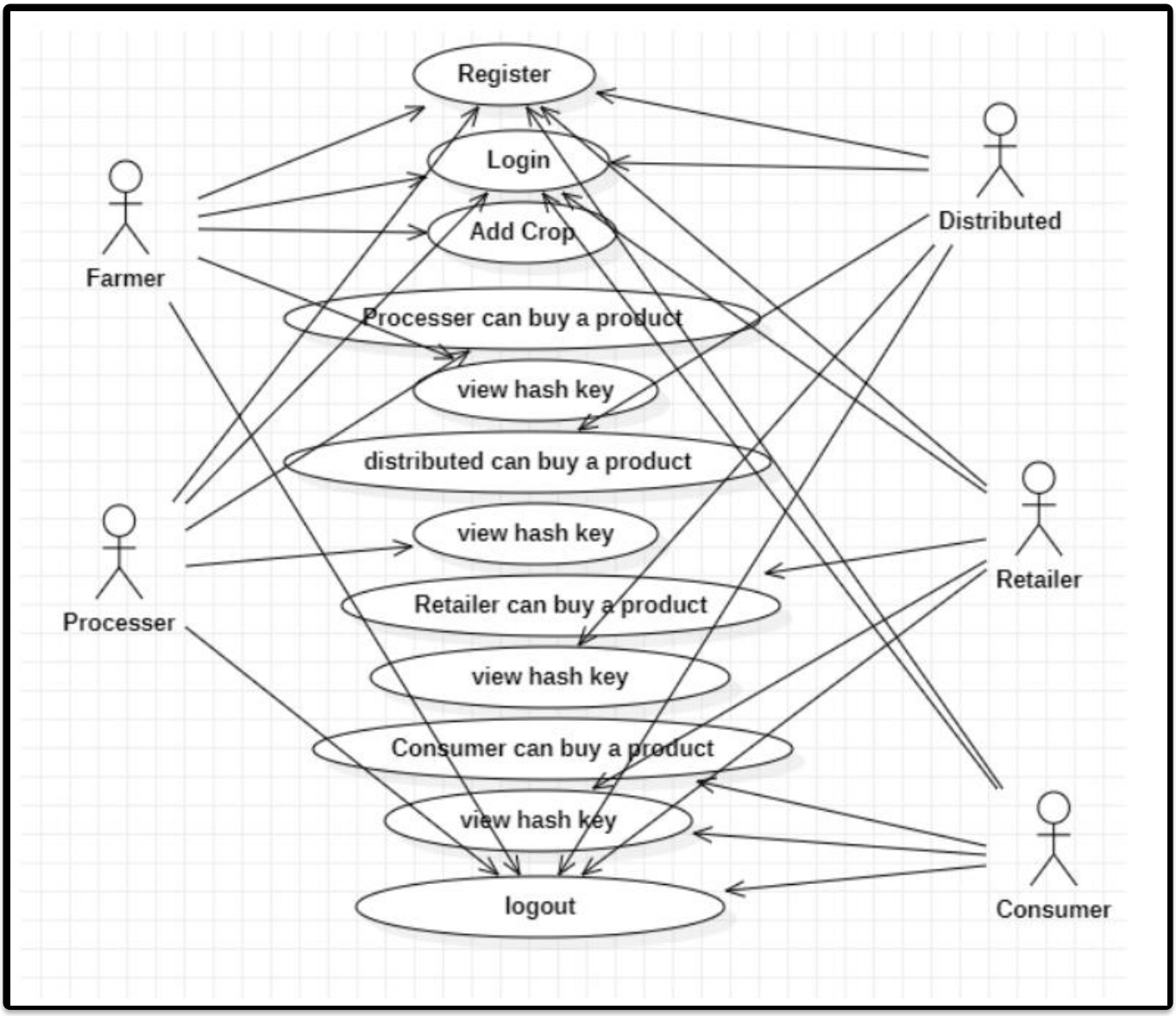
Fig 3.3.2 shows the Input Design involves determining how users interact with the system to input data ensuring user-friendly interfaces .



3.3.3 Module Design

Fig 3.3.3 shows the module Design for processor page status with report

USE CASE DIAGRAM:



3.3.3.1 Use case diagram

The Use case diagram in Fig 3.3.3.1 Use case diagram: Illustrates interactions between actors and the blockchain-based agri-food supply chain system, certification verification, and transaction management.

SEQUENCE DIAGRAM:

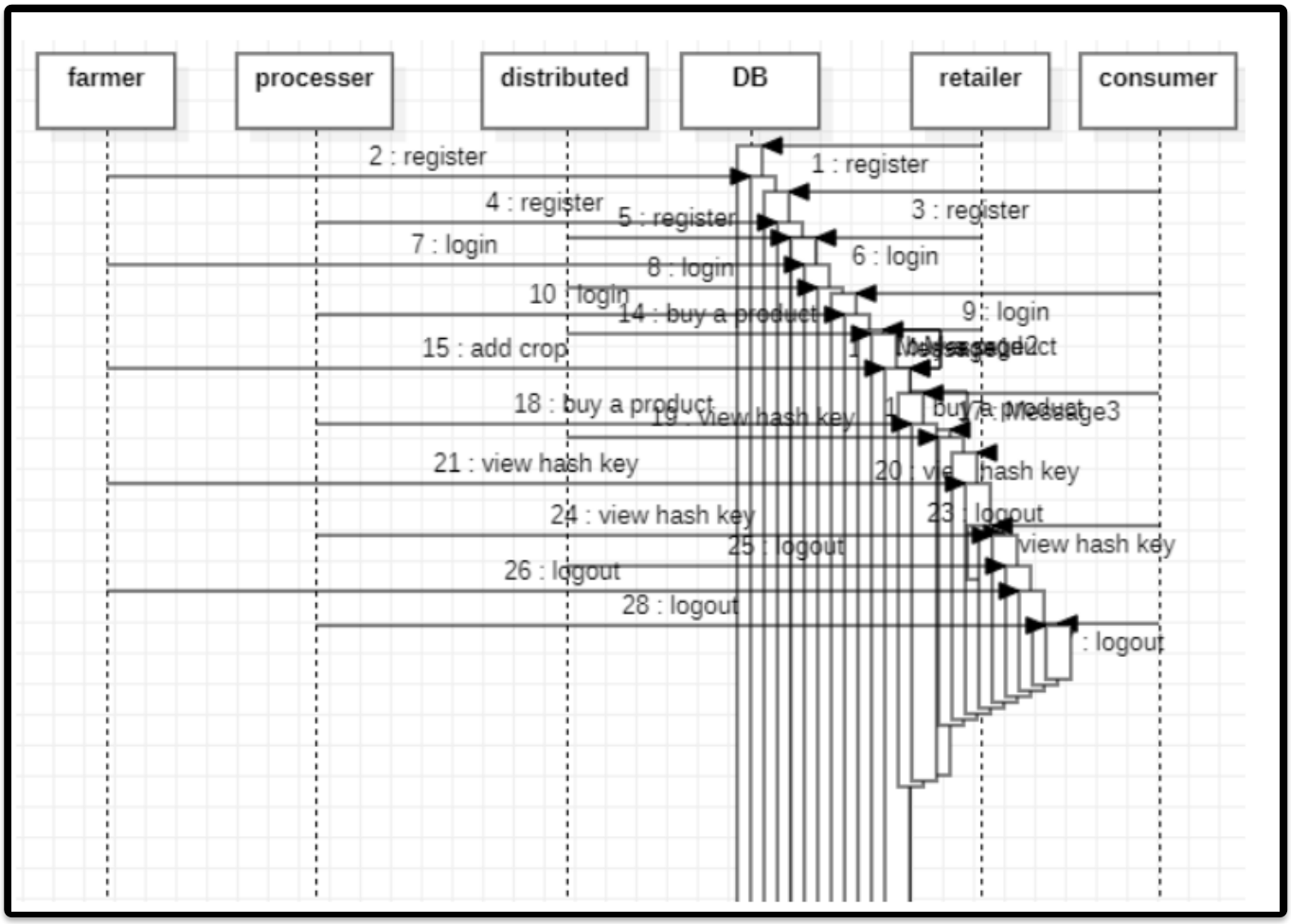


Fig 3.3.3.2

Fig 3.3.3.2 Sequence diagram Demonstrates the flow of interactions between system components and actors, outlining steps such as farmer registration, product certification, order placement, within the blockchain-enabled agri-food supply

COLLABORATION DIAGRAM:

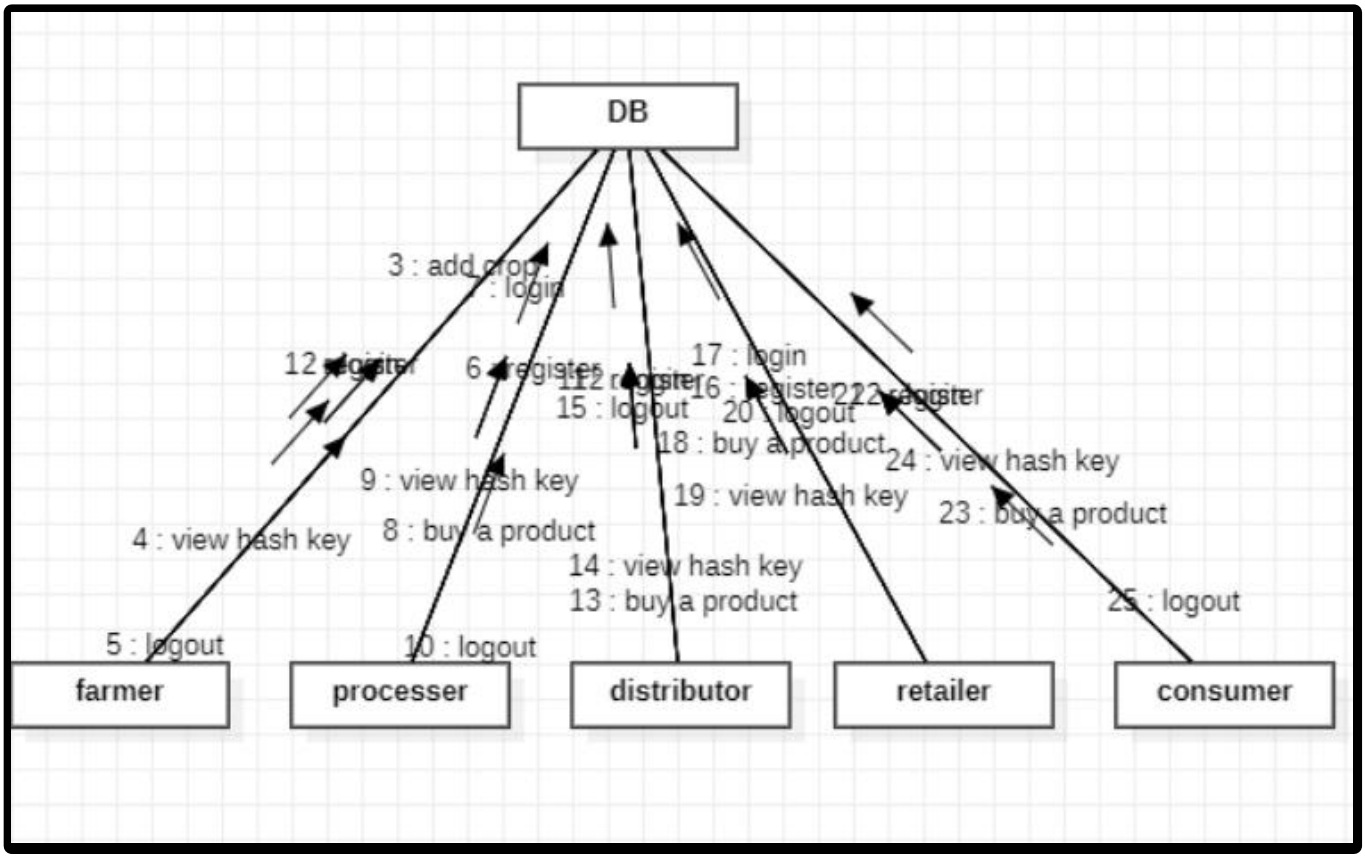


Fig 3.3.3.3

Fig 3.3.3.3 Collabration diagram for Visualizes how different components, such as farmers, distributors, consumers, and smart contracts, interact and communicate to execute processes like product authentication, inventory management, and payment settlement in the blockchain-integrated agri-food supply chain.

DEPLOYMENT DIAGRAM:

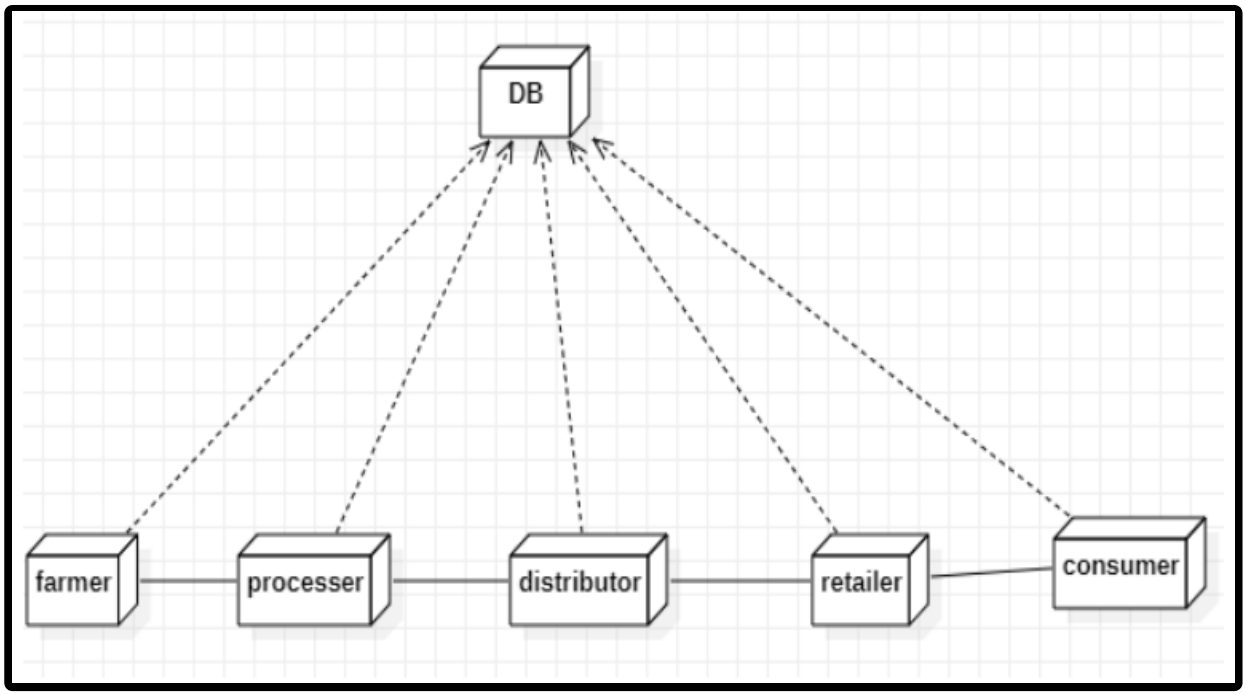
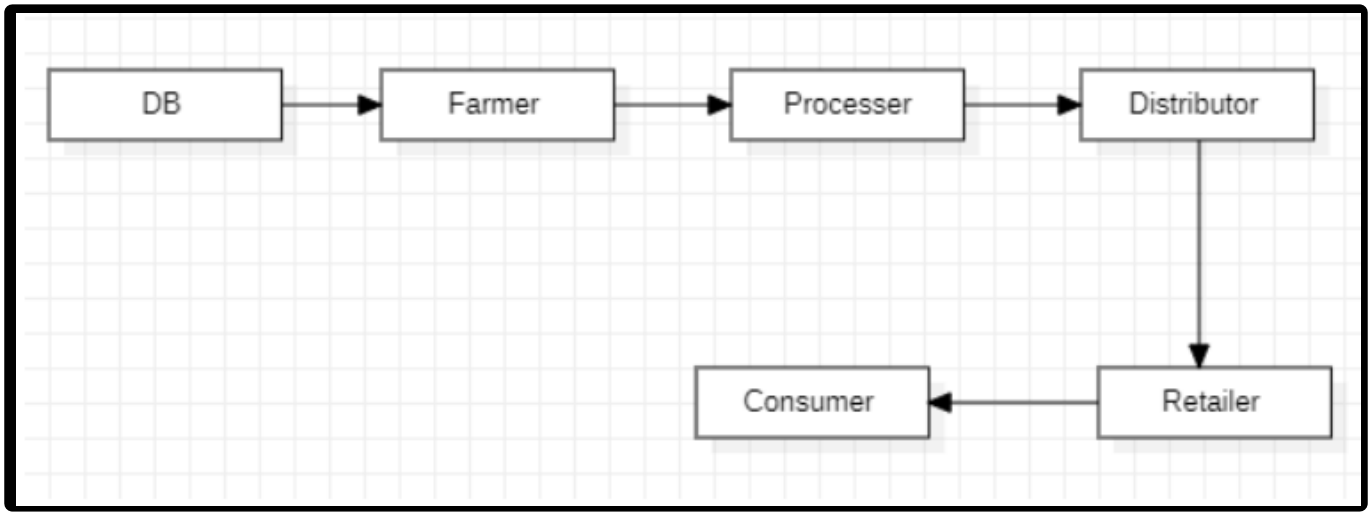


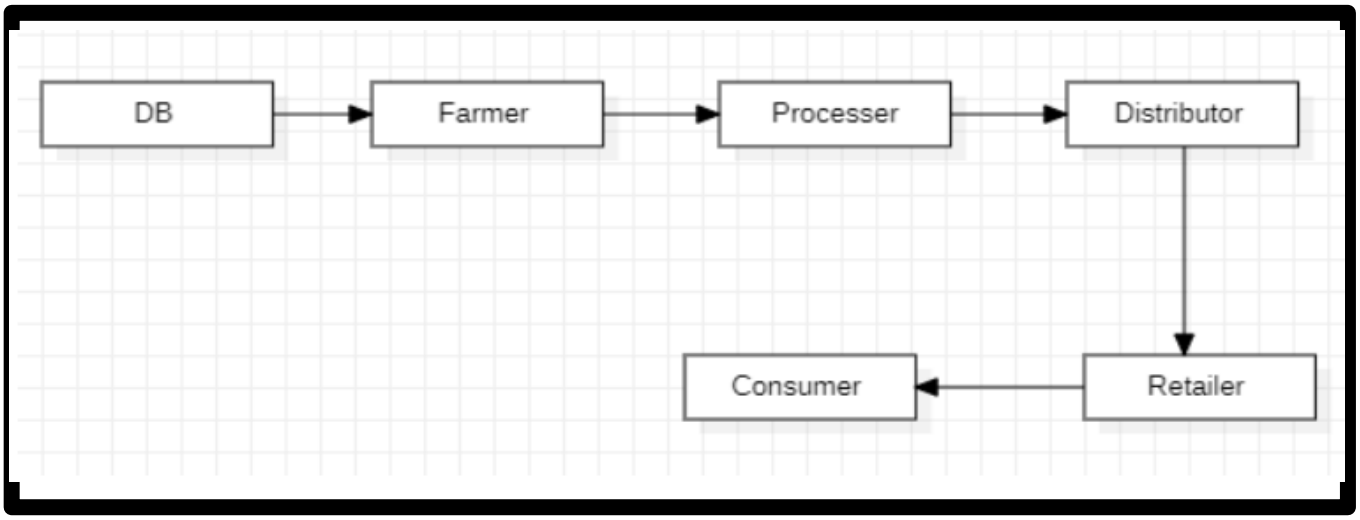
Fig 3.3.3.4 Deployment Diagram for Effective Management for Illustrates the physical deployment of system components like blockchain nodes, smart contracts, web servers, and databases across various network nodes, ensuring scalability, redundancy, and fault tolerance in the agri-food supply chain infrastructure.

DATAFLOW DIAGRAM:

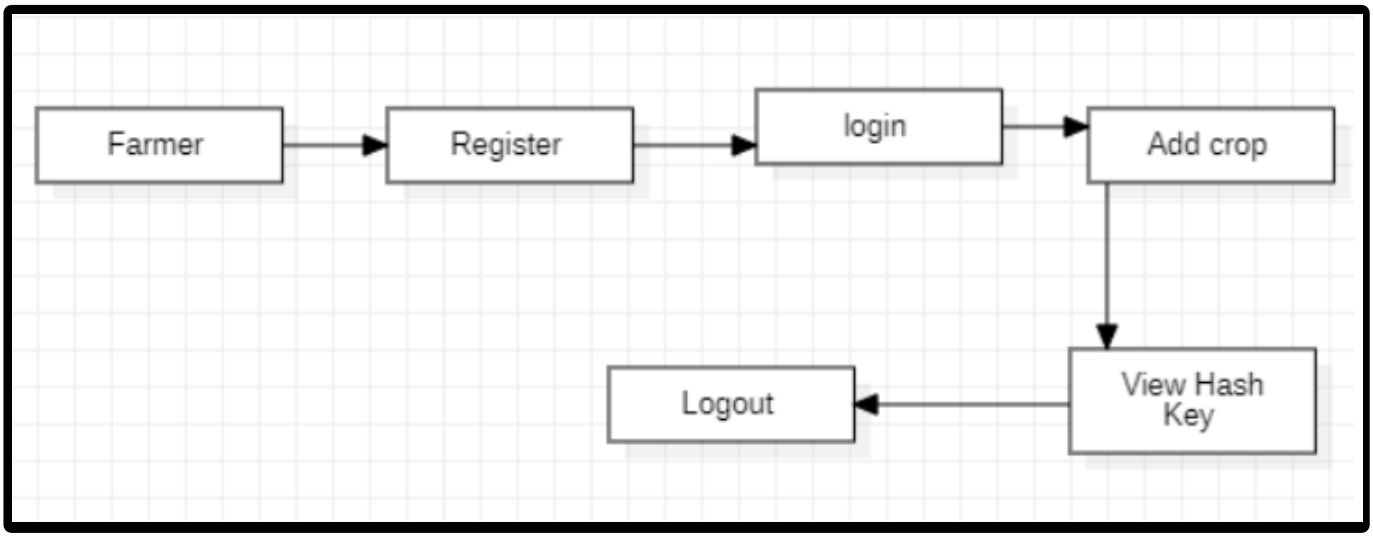
LEVEL 0:



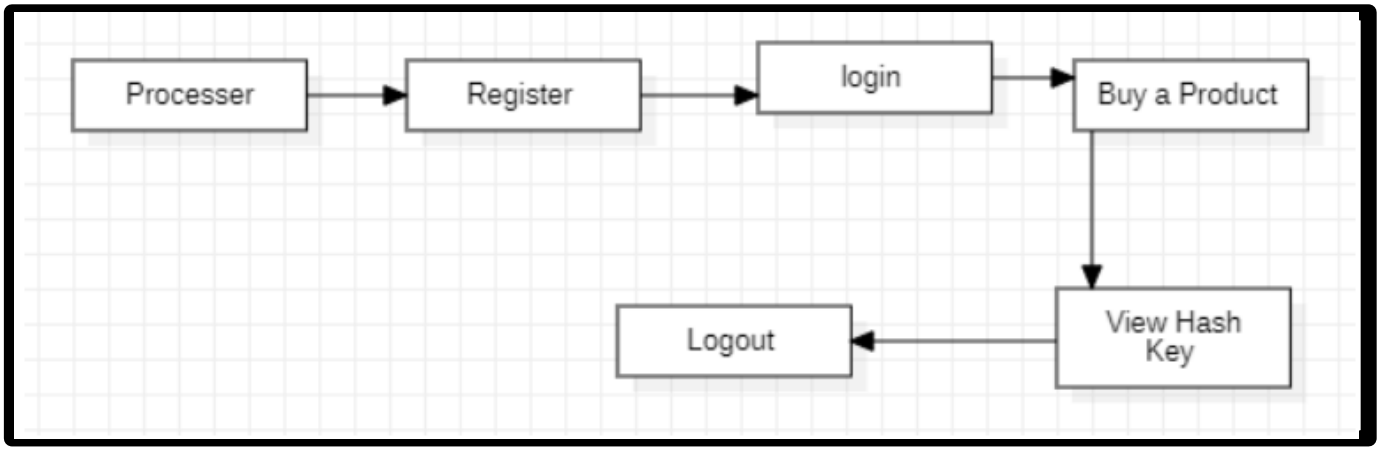
LEVEL 1:



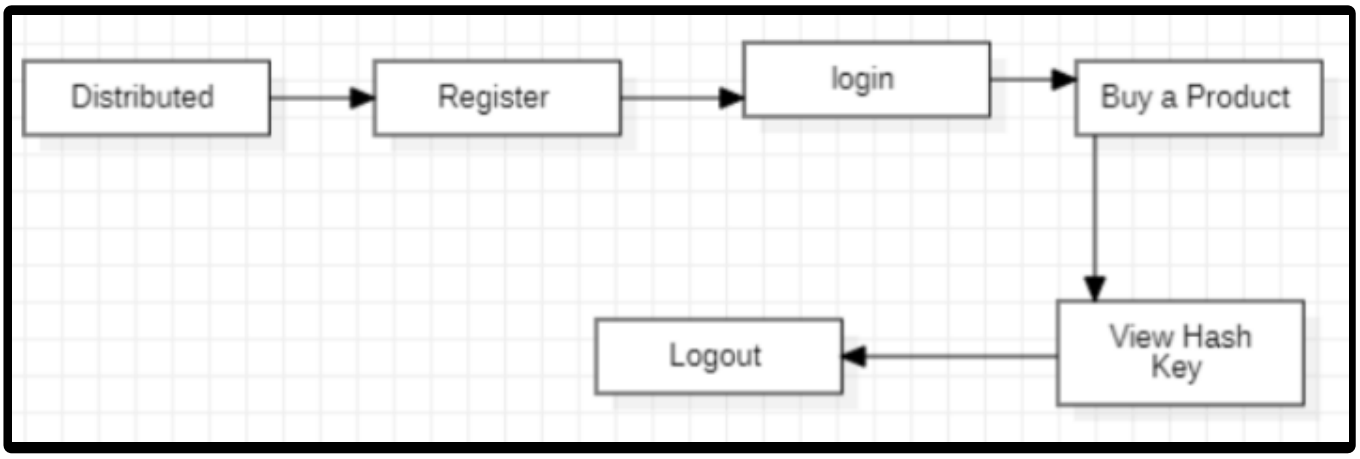
LEVEL 2:



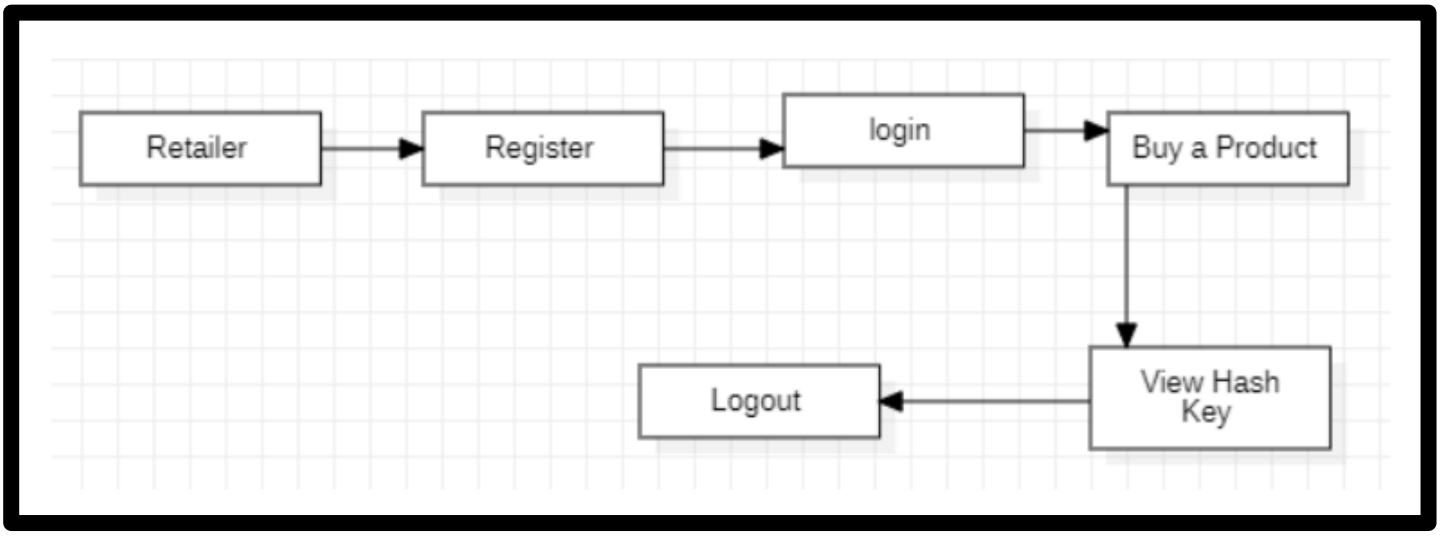
LEVEL 3:



LEVEL 4:



LEVEL 5:



LEVEL 6:

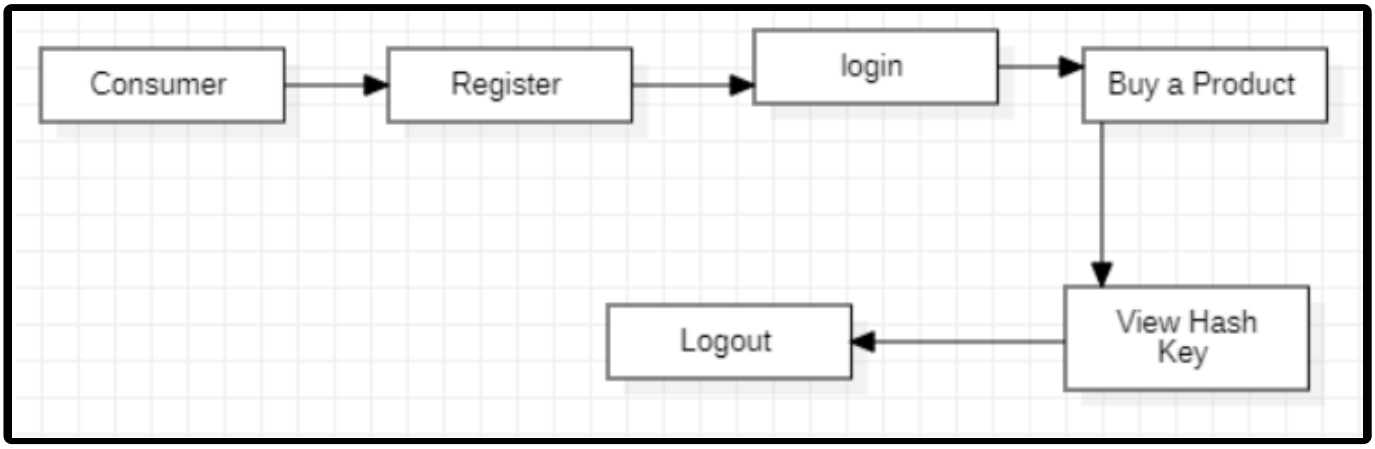


Fig 3.3.3.5

Fig 3.3.3.5 Data flow diagram for effective management for agri based food supply chain

ER DIAGRAM:

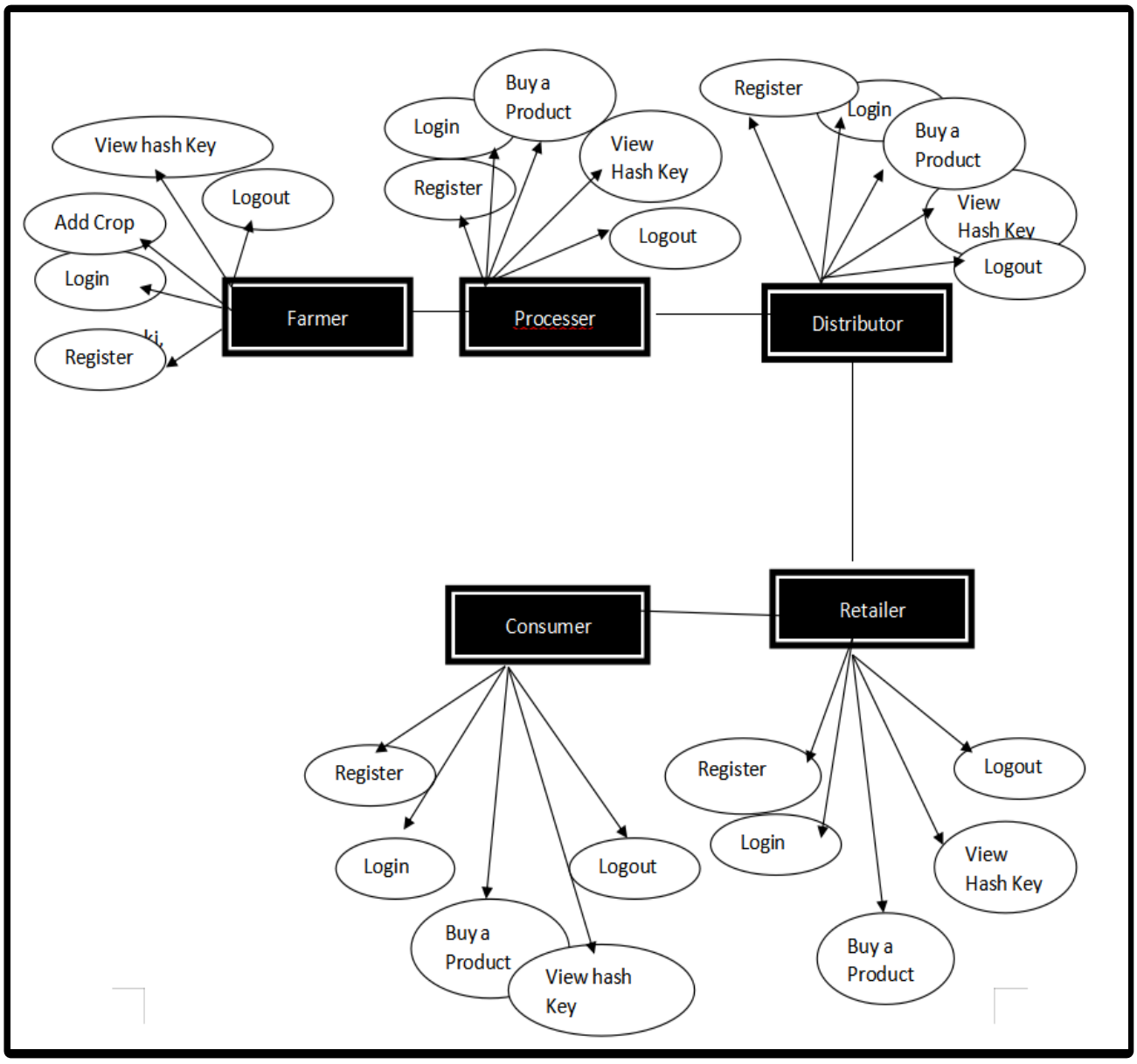


Fig 3.3.3.6

Fig 3.3.3.6 ER diagram for effective management for agri based food supply chain

CHAPTER 4
SYSTEM IMPLEMENTATION

Image classification consists of 5 main modules.

- **FARMER**
- **PROCESSER**
- **DISTRIBUTER**
- **RETAILERS**
- **CUSTOMERS**

4.1 FARMER

Farmers can utilize blockchain-based supply chain traceability tools to track the movement of their products from farm to fork. By recording key information such as product origin, production methods, handling practices, and transportation details on the blockchain, farmers can provide transparent and verifiable information to consumers, retailers, and other stakeholders. This enhances trust, ensures food safety, and supports sustainability initiatives.Smart contracts are utilized to automate and enforce agreements between farmers and other supply chain participants. Farmers can create smart contracts for various purposes, such as sales agreements, payment terms, delivery schedules, and quality standards. Smart contracts execute predefined conditions automatically, eliminating the need for intermediaries and reducing transaction costs and administrative burdens for farmers.

The Farmer Management module integrates with blockchain-based marketplaces that connect farmers directly with buyers, retailers, and consumers. Farmers can list their products on these marketplaces, set prices, and negotiate terms using smart contracts. This enables farmers to access new markets, improve market visibility, and receive fair compensation for their products while bypassing traditional intermediaries.Farmers can utilize blockchain technology to streamline compliance and certification processes related to food safety standards.

STEPS INVOLES IN FARMERS:

Farmer

Register the account with the basic information

Login account with the correct username and password

Farmer can add crop

Once the processer can buy the product. Farmer can view the hash key value view status

4.2 PROCESSER

Processors receive comprehensive education and training on blockchain technology, its applications in agri-food supply chains, and the benefits of adopting blockchain-based systems. This training empowers processors with the knowledge and skills required to leverage blockchain effectively in their operations.
Supply Chain Traceability and Transparency:

The module enables processors to track and trace the movement of raw materials, ingredients, and finished products throughout the supply chain using blockchain technology. Processors can access real-time visibility into product origins, production processes, quality control checks, and distribution channels. This enhances transparency, facilitates compliance with regulatory requirements, and ensures product authenticity and safety.
Quality Control and Assurance:

Processors can implement robust quality control and assurance processes using blockchain technology to maintain product quality and consistency. Processors record quality control checks, certifications, and compliance documentation on the blockchain, creating transparent and immutable records for auditing and verification purposes.

Inventory and Production Management:

The module provides processors with blockchain-based tools for inventory and production management. Processors can optimize production scheduling, monitor equipment utilization, and manage inventory levels efficiently. By recording production data and inventory transactions on the blockchain, processors can improve operational visibility, reduce waste, and enhance resource utilization.

STEPS INVOLES IN PROCESSER:

Processor

Register the account with the basic information

Login account with the correct username and password

Processer can buy the product

Once the Distributor can buy the product. Processor can view the Hash key value view status

4.3 DISTRIBUTER

Distributors receive comprehensive education and training on blockchain technology, its applications in agri-food supply chains, and the benefits of adopting blockchain-based systems. This training equips distributors with the knowledge and understanding required to leverage blockchain effectively in their operations.The module enables distributors to track and trace the movement of products throughout the supply chain using blockchain technology. Distributors can access real-time visibility into product origins, production processes, transportation routes, and storage conditions. This enhances transparency, facilitates recall management, and ensures compliance with regulatory requirements.Distributors can utilize blockchain-based inventory management tools to optimize stock levels, monitor product availability, and manage inventory across multiple locations. By recording inventory transactions on the blockchain, distributors can improve accuracy, reduce stockouts, and minimize excess inventory holding costs.Distributors can implement quality assurance and compliance protocols using blockchain technology to ensure that products meet specified standards and regulatory requirements. Distributors can record quality control checks, certifications, and compliance documentation on the blockchain, providing transparent and auditable records for regulatory audits and inspections.

STEPS INVOLES IN DISTRIBUTOR

Distributed

Register the account with the basic information

Login account with the correct username and password

Distributor can buy the product

Once the Retailer can buy the product. Distributor can view the Hash key value view status

4.4 RETAILER

Retailers receive comprehensive education and training on blockchain technology, its applications in agri-food supply chains, and the benefits of adopting blockchain-based systems. This training empowers retailers with the knowledge and skills required to leverage blockchain effectively in their operations.The module enables retailers to trace the origin and journey of products throughout the supply chain using blockchain technology. Retailers can access real-time visibility into product origins, production methods, quality control checks, and transportation routes. This enhances transparency, facilitates compliance with food safety standards, and ensures product authenticity and safety.Retailers can utilize blockchain-based tools for inventory management and shelf life tracking. Retailers can monitor inventory levels, track product expiration dates, and optimize shelf space allocation. By recording inventory transactions and product attributes on the blockchain, retailers can reduce waste, improve product freshness, and enhance customer satisfaction.Smart contracts are utilized to automate and enforce agreements between retailers and suppliers, distributors, and other supply chain partners. Retailers can create smart contracts for procurement, pricing terms, delivery schedules, and quality specifications. Smart contracts execute predefined conditions automatically, reducing transaction costs, minimizing disputes, and ensuring compliance with contractual obligations.Retailers can implement robust quality assurance and product authentication processes using blockchain technology. Retailers record product certifications, labeling information, and authenticity verification data on the blockchain, creating transparent and immutable records for authentication and verification purposes. This enhances consumer trust

and confidence in the products offered by retailers.

STEPS INVOLES IN RETAILERS

Retailer

Register the account with the basic information

Login account with the correct username and password

Retailer can buy the product

Once the Consumer can buy the product. Retailer can view the Hash key value view status

4.5 CONSUMERS

Consumers can access transparent and verifiable information about the origin, production methods, and journey of products throughout the supply chain using blockchain technology. By scanning QR codes or accessing blockchain-based platforms, consumers can trace the origin of products, verify authenticity, and ensure compliance with food safety standards and ethical sourcing practices.The module facilitates consumer engagement and feedback by providing consumers with opportunities to interact with brands, producers, and supply chain stakeholders. Consumers can provide feedback, ratings, and reviews on products, share their preferences and concerns, and participate in surveys and loyalty programs. This fosters trust, transparency, and accountability within the supply chain ecosystem.Consumers can access information about sustainability initiatives, ethical sourcing practices, and environmental impact assessments related to the products they consume. Blockchain-based platforms provide transparent and verifiable data on factors such as carbon footprint, water usage, and labor practices, enabling consumers to support brands and products aligned with their values and preferences.Consumers can verify product labels, certifications, and claims using blockchain technology to ensure accuracy and authenticity. Blockchain-based platforms provide transparent and immutable records of product certifications, organic labels, fair trade certifications, and other quality

indicators, enabling consumers to make informed decisions based on reliable information.

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 TESTING

TEST CASE ID	TESTCASE/ ACTION TO BE PERFORMED	EXPECTED RESULT	ACTUAL RESULT	PASS/FAIL
1.	Login id for user with password	Login success	Login success	Pass
2.	Login id for faremer with password	Login success	Login success	Pass
3.	Farmer adding crops And details	Added successfully	Added successfully	Pass
4.	User selects distributor	selected	selected	Pass
5.	User selects which type and number of crops	Added successfully	Added successfully	Pass
6.	View status of the crop	Viewing status box	Viewing status box	Pass

5.1 Test Result

In table 5.1 test cases form a comprehensive suite to verify the function

5.2 Results & Discussion

Participants have successfully leveraged blockchain to track the journey of food products from farm to fork, providing consumers with unprecedented visibility into the production and distribution process.

By recording every transaction and movement of goods on the blockchain, participants have established a transparent and immutable audit trail, reducing the risk of fraud, counterfeit products, and supply chain disruptions.The adoption of blockchain technology has streamlined supply chain processes, reducing paperwork, manual interventions, and administrative overheads.Participants have optimized inventory management, logistics, and procurement processes, leading to cost savings and improved resource utilization.Blockchain has facilitated greater collaboration and information sharing among stakeholders, fostering trust and cooperation across the agri-food ecosystem.Through shared access to blockchain-based platforms, participants have enabled real-time communication and collaboration, leading to faster decision-making and problem-solving.Participants have implemented blockchain solutions that comply with regulatory standards and requirements, ensuring food safety, quality, and compliance with labeling and traceability regulations.By maintaining accurate and verifiable records on the blockchain, participants have facilitated audits, inspections, and compliance reporting, reducing the risk of non-compliance and associated penalties.

Blockchain-enabled transparency and traceability have instilled confidence in consumers, leading to increased trust and loyalty towards brands and products.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

Design a blockchain-based framework to guarantee the agri-food safety with product traceability in ASC systems. DR-SCM method to make decisions on the production and storage of agri-food products for optimizing product profits in ASCs. The extensive simulation experiments verify the effectiveness of the proposed blockchain-based framework and the DR-SCM method for ASC optimization. More specifically, the results show that the proposed blockchain-based ASC framework can well guarantee reliable product traceability

6.2 FUTURE ENHANCEMENTS

In the future, employing increasingly sophisticated ASC management situations with demands created using real-world data, adding more cutting-edge DRL-based algorithms (such as asynchronous advantage actor-critic) will be implemented. In the meanwhile, assess the robustness and potential advancements of these algorithms and investigate their viability in actual ASC situations are done

REFERENCES

[1] D.-Y. Lin, C.-J. Juan, and C.-C. Chang, ‘‘Managing food safety with pricing, contracts and coordination in supply chains,’’ IEEE Access, vol. 7, pp. 150892–150909, 2019.

[2] H. Fan, ‘‘Theoretical basis and system establishment of China food safety intelligent supervision in the perspective of Internet of Things,’’ IEEE Access, vol. 7, pp. 71686–71695, 2019.

[3] M. Toledo-Hernández, T. Tscharntke, A. Tjoa, A. Anshary, B. Cyio, and T. C. Wanger, ‘‘Hand pollination, not pesticides or fertilizers, increases cocoa yields and farmer income,’’ Agricult., Ecosyst. Environ., vol. 304, Dec. 2020, Art. no. 107160.

[4] J. Himmelstein, A. Ares, D. Gallagher, and J. Myers, ‘‘A meta-analysis of intercropping in Africa: Impacts on crop yield, farmer income, and integrated pest management effects,’’ Int. J. Agricult. Sustainability, vol. 15, no. 1, pp. 1–10, Jan. 2017.

[5] Y. Dong, Z. Fu, S. Stankovski, S. Wang, and X. Li, ‘‘Nutritional quality and safety traceability system for China’s leafy vegetable supply chain based on fault tree analysis and QR code,’’ IEEE Access, vol. 8, pp. 161261–161275, 2020.

[6] C. Ganeshkumar, M. Pachayappan, and G. Madanmohan, ‘‘Agri-food supply chain management: Literature review,’’ Intell. Inf. Manage., vol. 9, no. 2, pp. 68–96, 2017.[

[7] Q. Lin, H. Wang, X. Pei, and J. Wang, ‘‘Food safety traceability system based on blockchain and EPCIS,’’ IEEE Access, vol. 7, pp. 20698–20707, 2019.

[8] H. Feng, X. Wang, Y. Duan, J. Zhang, and X. Zhang, ‘‘Applying blockchain technology to improve agri-food traceability: A review of development methods, benefits and challenges,’’ J. Cleaner Prod., vol. 260, Jul. 2020, Art. no. 121031.

[9] S. Nakamoto, ‘‘Bitcoin: A peer-to-peer electronic cash system,’’ White Paper, 2008. Accessed: Jun. 26, 2020. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[10] F. Tian, ‘‘An agri-food supply chain traceability system for China based on RFID & blockchain technology,’’ in Proc. 13th Int. Conf. Service Syst. Service Manage. (ICSSSM), Jun. 2016, pp. 1–6.

APPENDICES

A.1 SDG GOALS

SDG 1: No Poverty: By facilitating direct sales between farmers and customers, the project can help increase the income of small-scale farmers, thus reducing poverty in rural areas.

SDG 2: Zero Hunger: By improving access to markets and ensuring fair prices for farmers, the project can contribute to reducing hunger and improving food security.

SDG 3: Good Health and Well-being: Ensuring transparency and traceability in the food supply chain can lead to safer food products, promoting better health outcomes for consumers.

SDG 4: Decent Work and Economic Growth: By cutting out intermediaries and enabling farmers to receive fair prices for their produce, the project can create economic opportunities and promote sustainable livelihoods in rural communities.

SDG 5: Industry, Innovation, and Infrastructure: Leveraging blockchain technology for direct crop sales represents an innovative approach to improving agricultural supply chains and infrastructure, thereby contributing to economic growth and development.

A.2 Source Code

```
import java.io.IOException;
 import java.io.PrintWriter;
 import java.net.InetAddress;
 import java.net.UnknownHostException;
 import java.sql.Connection;
 import java.sql.DriverManager;
 import java.sql.ResultSet;
 import java.sql.Statement;
 import java.text.DecimalFormat;
 import java.util.Random;
 import javax.servlet.RequestDispatcher;
 import javax.servlet.ServletException;
 import javax.servlet.annotation.WebServlet;
 import javax.servlet.http.HttpServlet;
 import javax.servlet.http.HttpServletRequest;
 import javax.servlet.http.HttpServletResponse;
 import javax.servlet.http.HttpSession;

 /**
  *
  * @author java.2
  */
 @WebServlet(name = "Register", urlPatterns = {"/Register"})
 public class AddCrop extends HttpServlet {

 Connection con=null;
    Statement st=null,st1=null;
    ResultSet rs=null;
    RequestDispatcher rd=null;
         public void doPost(HttpServletRequest req, HttpServletResponse res) throws
 IOException,ServletException {


            HttpSession sn = req.getSession(true);
            String username=sn.getAttribute("username").toString();
             String cropname= req.getParameter("cropname");
            String quantity= req.getParameter("quantity");
           String price= req.getParameter("price");

           RequestDispatcher rd;
          Random generator1 = new Random();
            int seckey = generator1.nextInt(1000000);
```

```
            try {


              Class.forName("com.mysql.jdbc.Driver");
              con =
        DriverManager.getConnection("jdbc:mysql://localhost:3306/foodnetworkblockchain",
        "root","root");
             String processor="nil";
              String nil="nil";
              st1 = con.createStatement();
            int add=st1.executeUpdate("insert into
        cropdetails(username,cropname,quantity,price,processor,hashkey1,distributor,hashkey
        2,retailer,hashkey3,consumer,hashkey4)
        values('"+username+"','"+cropname+"','"+quantity+"','"+price+"','"+processor+"','"+ni
        l+"','"+nil+"','"+nil+"','"+nil+"','"+nil+"','"+nil+"','"+nil+"')");
           rd=req.getRequestDispatcher("viewstatus.jsp");
              rd.forward(req,res);

           } catch(Exception e2)
            {
              System.out.println("Exception : "+e2.toString());
           }
       }
    }
```

```
import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;
  public class ConsumerLogin extends HttpServlet {
   String username="";
   String password="";
   String type="";
   Connection con=null;
   Statement st=null;
   ResultSet rs=null;
```

```
   public void doPost(HttpServletRequest req, HttpServletResponse res) throws
IOException,ServletException {
       username = req.getParameter("username");
       password = req.getParameter("password");

          HttpSession sn = req.getSession(true);
    sn.setAttribute("username",username);

                    RequestDispatcher rd = null;
       System.out.println("User login works"+type);
       try {

          Class.forName("com.mysql.jdbc.Driver");
          con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/foodnetworkblockchain","root","r
oot");
          st = con.createStatement();
          rs = st.executeQuery("select * from consumerdetails where username='"+username+"'
&& password='"+password+"'");
          if(rs.next()) {
              rd=req.getRequestDispatcher("consumerpage.jsp");
          }
              else {

             rd=req.getRequestDispatcher("failure.jsp");

          }
    rd.forward(req,res);


       }
       catch(Exception e2)
        {

          System.out.println("Exception : "+e2.toString());

       }
   }
}
```

```
import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.lang.*;
import java.security.MessageDigest;
import java.text.DecimalFormat;
import java.text.SimpleDateFormat;
  // import de.daslaboratorium.machinelearning.classifier.Classification;
///import de.daslaboratorium.machinelearning.classifier.Classifier;
public class ConsumerProcess extends HttpServlet {

Connection con=null;
    Statement st=null,st1=null;
    ResultSet rs=null;
    RequestDispatcher rd=null;
          public void doPost(HttpServletRequest req, HttpServletResponse res) throws
IOException,ServletException {


          HttpSession sn = req.getSession(true);


             String username= sn.getAttribute("username").toString();
           String id= req.getParameter("id");
```

```
   String status="approved";

        RequestDispatcher rd;

        try {


        Class.forName("com.mysql.jdbc.Driver");
        con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/foodnetworkblockchain","root","r
oot");

        int id1=0;
        String prevdata="";
         st = con.createStatement();
        ResultSet rs1 = st.executeQuery("select * from blockchain");
        while(rs1.next()) {
        id1=rs1.getInt(1)+1;
        prevdata=rs1.getString(2);
        }


            java.util.Date date = new java.util.Date() ;
                        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-
dd HH:mm") ;
                        dateFormat.format(date);
                        System.out.println(dateFormat.format(date));
            String data1 = prevdata+dateFormat.format(date)+username;



                                StringBuffer sb=new StringBuffer();
                                MessageDigest md = MessageDigest.getInstance("MD5");
                                    md.update(String.valueOf(data1).getBytes());

                                    byte byteData[] = md.digest();

                                    //convert the byte to hex format method 1

                                    for (int i = 0; i < byteData.length; i++) {
                                     sb.append(Integer.toString((byteData[i] & 0xff) + 0x100,
16).substring(1));
                                    }
```

```
                    File file = new File("C://blockchain//block"+id1+".txt");

//Create the file
if (file.createNewFile())
{
   System.out.println("File is created!");
} else {
   System.out.println("File already exists.");
}

//Write Content
FileWriter writer = new FileWriter(file);
writer.write(sb.toString());
writer.close();

  Statement st1 = con.createStatement();
        int add1=st1.executeUpdate("insert into blockchain
values('"+id1+"','"+sb.toString()+"')");



         String query = "update cropdetails set consumer = ?,hashkey4 = ? where id = ?";
     PreparedStatement preparedStmt = con.prepareStatement(query);
     preparedStmt.setString  (1, username);
    preparedStmt.setString(2, sb.toString());
     preparedStmt.setString(3, id);
     // execute the java preparedstatement
     preparedStmt.executeUpdate();

         rd=req.getRequestDispatcher("consumerstatus.jsp");
         rd.forward(req,res);
       } catch(Exception e2) {
         // rd=req.getRequestDispatcher("failure.jsp");
         System.out.println(e2);
       }
   }
}
```

```
import java.io.IOException;
import java.io.PrintWriter;
import java.net.InetAddress;
import java.net.UnknownHostException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.text.DecimalFormat;
import java.util.Random;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

/**
 *
 * @author java.2
 */
@WebServlet(name = "Register", urlPatterns = {"/Register"})
public class ConsumerRegistration extends HttpServlet {

Connection con=null;
   Statement st=null,st1=null;
   ResultSet rs=null;
   RequestDispatcher rd=null;
        public void doPost(HttpServletRequest req, HttpServletResponse res) throws
IOException,ServletException {


        HttpSession sn = req.getSession(true);

          String fullname= req.getParameter("fullname");
        String username= req.getParameter("username");
       String password= req.getParameter("password");
        String company= req.getParameter("company");
                String mobile= req.getParameter("mobile");
                String place= req.getParameter("place");

        RequestDispatcher rd;
```

```
        Random generator1 = new Random();
          int seckey = generator1.nextInt(1000000);

          try {


          Class.forName("com.mysql.jdbc.Driver");
          con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/foodnetworkblockchain","root","r
oot");

          st1 = con.createStatement();
         int add=st1.executeUpdate("insert into consumerdetails
values('"+fullname+"','"+username+"','"+password+"','"+company+"','"+place+"','"+mobile+"')
");
       rd=req.getRequestDispatcher("success.jsp");
          rd.forward(req,res);

       } catch(Exception e2)
        {
          System.out.println("Exception : "+e2.toString());
       }
   }
}
```

```
import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;
  public class DistributorLogin extends HttpServlet {
   String username="";
   String password="";
   String type="";
   Connection con=null;
   Statement st=null;
```

```
   ResultSet rs=null;

   public void doPost(HttpServletRequest req, HttpServletResponse res) throws
IOException,ServletException {
       username = req.getParameter("username");
       password = req.getParameter("password");

          HttpSession sn = req.getSession(true);
    sn.setAttribute("username",username);

                    RequestDispatcher rd = null;
       System.out.println("User login works"+type);
       try {

          Class.forName("com.mysql.jdbc.Driver");
          con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/foodnetworkblockchain","root","r
oot");
          st = con.createStatement();
          rs = st.executeQuery("select * from distributordetails where username='"+username+"'
&& password='"+password+"'");
          if(rs.next()) {
              rd=req.getRequestDispatcher("distributorpage.jsp");
          }
              else {

             rd=req.getRequestDispatcher("failure.jsp");

          }
    rd.forward(req,res);


       }
       catch(Exception e2)
        {

          System.out.println("Exception : "+e2.toString());

       }
   }
}
```

```
import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.lang.*;
import java.security.MessageDigest;
import java.text.DecimalFormat;
import java.text.SimpleDateFormat;
  // import de.daslaboratorium.machinelearning.classifier.Classification;
///import de.daslaboratorium.machinelearning.classifier.Classifier;
public class DistributorProcess extends HttpServlet {

Connection con=null;
    Statement st=null,st1=null;
    ResultSet rs=null;
    RequestDispatcher rd=null;
          public void doPost(HttpServletRequest req, HttpServletResponse res) throws
IOException,ServletException {


          HttpSession sn = req.getSession(true);
```

```
String username= sn.getAttribute("username").toString();
String id= req.getParameter("id");

String status="approved";

RequestDispatcher rd;

try {


Class.forName("com.mysql.jdbc.Driver");
con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/foodnetworkblockchain","root","r
oot");

int id1=0;
String prevdata="";
st = con.createStatement();
ResultSet rs1 = st.executeQuery("select * from blockchain");
while(rs1.next()) {
id1=rs1.getInt(1)+1;
prevdata=rs1.getString(2);
}


java.util.Date date = new java.util.Date() ;
SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-
dd HH:mm") ;
dateFormat.format(date);
System.out.println(dateFormat.format(date));
String data1 = prevdata+dateFormat.format(date)+username;



StringBuffer sb=new StringBuffer();
MessageDigest md = MessageDigest.getInstance("MD5");
md.update(String.valueOf(data1).getBytes());

byte byteData[] = md.digest();

//convert the byte to hex format method 1

for (int i = 0; i < byteData.length; i++) {
sb.append(Integer.toString((byteData[i] & 0xff) + 0x100,
```

```
16).substring(1));
                                        }

                        File file = new File("C://blockchain//block"+id1+".txt");

//Create the file
if (file.createNewFile())
{
   System.out.println("File is created!");
} else {
   System.out.println("File already exists.");
}

//Write Content
FileWriter writer = new FileWriter(file);
writer.write(sb.toString());
writer.close();

  Statement st1 = con.createStatement();
        int add1=st1.executeUpdate("insert into blockchain
values('"+id1+"','"+sb.toString()+"')");



          String query = "update cropdetails set distributor = ?,hashkey2 = ? where id = ?";
     PreparedStatement preparedStmt = con.prepareStatement(query);
     preparedStmt.setString  (1, username);
    preparedStmt.setString(2, sb.toString());
     preparedStmt.setString(3, id);
     // execute the java preparedstatement
     preparedStmt.executeUpdate();

          rd=req.getRequestDispatcher("distributorstatus.jsp");
          rd.forward(req,res);
       } catch(Exception e2) {
          // rd=req.getRequestDispatcher("failure.jsp");
          System.out.println(e2);
       }
   }
}
```

```
import java.io.IOException;
import java.io.PrintWriter;
import java.net.InetAddress;
import java.net.UnknownHostException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.text.DecimalFormat;
import java.util.Random;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

/**
 *
 * @author java.2
 */
@WebServlet(name = "Register", urlPatterns = {"/Register"})
public class DistributorRegistration extends HttpServlet {

Connection con=null;
    Statement st=null,st1=null;
    ResultSet rs=null;
    RequestDispatcher rd=null;
          public void doPost(HttpServletRequest req, HttpServletResponse res) throws
IOException,ServletException {


          HttpSession sn = req.getSession(true);

            String fullname= req.getParameter("fullname");
          String username= req.getParameter("username");
         String password= req.getParameter("password");
          String company= req.getParameter("company");
                    String mobile= req.getParameter("mobile");
```

```
                String place= req.getParameter("place");

          RequestDispatcher rd;
        Random generator1 = new Random();
          int seckey = generator1.nextInt(1000000);

          try {


          Class.forName("com.mysql.jdbc.Driver");
          con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/foodnetworkblockchain","root","r
oot");

          st1 = con.createStatement();
         int add=st1.executeUpdate("insert into distributordetails
values('"+fullname+"','"+username+"','"+password+"','"+company+"','"+place+"','"+mobile+"')
");
       rd=req.getRequestDispatcher("success.jsp");
          rd.forward(req,res);

       } catch(Exception e2)
        {
          System.out.println("Exception : "+e2.toString());
       }
   }
}
```

```
import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;
  public class FarmerLogin extends HttpServlet {
   String username="";
   String password="";
```

```
    String type="";
    Connection con=null;
    Statement st=null;
    ResultSet rs=null;

    public void doPost(HttpServletRequest req, HttpServletResponse res) throws
IOException,ServletException {
        username = req.getParameter("username");
        password = req.getParameter("password");

            HttpSession sn = req.getSession(true);
     sn.setAttribute("username",username);

                    RequestDispatcher rd = null;
        System.out.println("User login works"+type);
        try {

            Class.forName("com.mysql.jdbc.Driver");
            con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/foodnetworkblockchain","root","r
oot");
            st = con.createStatement();
            rs = st.executeQuery("select * from farmerdetails where username='"+username+"' &&
password='"+password+"'");
            if(rs.next()) {
                rd=req.getRequestDispatcher("farmerpage.jsp");
           }
               else {

              rd=req.getRequestDispatcher("failure.jsp");

            }
     rd.forward(req,res);


        }
        catch(Exception e2)
         {

            System.out.println("Exception : "+e2.toString());

        }
    }
}
```

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;
   public class FarmerLogin extends HttpServlet {
    String username="";
    String password="";
    String type="";
    Connection con=null;
    Statement st=null;
    ResultSet rs=null;

    public void doPost(HttpServletRequest req, HttpServletResponse res) throws
IOException,ServletException {
```

```
        username = req.getParameter("username");
        password = req.getParameter("password");

            HttpSession sn = req.getSession(true);
     sn.setAttribute("username",username);

                        RequestDispatcher rd = null;
        System.out.println("User login works"+type);
        try {

            Class.forName("com.mysql.jdbc.Driver");
            con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/foodnetworkblockchain","root","r
oot");
            st = con.createStatement();
            rs = st.executeQuery("select * from farmerdetails where username='"+username+"' &&
password='"+password+"'");
            if(rs.next()) {
                rd=req.getRequestDispatcher("farmerpage.jsp");
            }
                else {

               rd=req.getRequestDispatcher("failure.jsp");

            }
     rd.forward(req,res);


        }
        catch(Exception e2)
         {

            System.out.println("Exception : "+e2.toString());

        }
    }
}
```

```
import org.jfree.chart.ChartPanel;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.JFreeChart;
import org.jfree.ui.ApplicationFrame;
import org.jfree.ui.RefineryUtilities;
import org.jfree.chart.plot.PlotOrientation;
import org.jfree.data.category.DefaultCategoryDataset;

public class LineChart1 extends ApplicationFrame {

   public LineChart1( String applicationTitle , String chartTitle ) {
      super(applicationTitle);
      JFreeChart lineChart = ChartFactory.createLineChart(
         chartTitle,
         "Review Strength Length","Mining Time(s)",
         createDataset(),
         PlotOrientation.VERTICAL,
         true,true,false);

      ChartPanel chartPanel = new ChartPanel( lineChart );
      chartPanel.setPreferredSize( new java.awt.Dimension( 560 , 367 ) );
      setContentPane( chartPanel );
   }

   private DefaultCategoryDataset createDataset( ) {
      DefaultCategoryDataset dataset = new DefaultCategoryDataset( );

      String c1 = "18";
     // String c2 = "10";
      String c3 = "22";
     // String c4 = "30";
```

```
        String c5 = "28";
       // String c6 = "50";
        String c7 = "34";
      //  String c8 = "70";
        //String c9 = "50";

       String series1 = "";


        dataset.addValue( 12 , series1 , c1 );
        //dataset.addValue( 25 , series1 , c2 );
        dataset.addValue( 6 , series1 , c3 );
       // dataset.addValue( 44 , series1 , c4 );
        dataset.addValue( 8 , series1 , c5 );
       // dataset.addValue( 62 , series1 , c6 );
        dataset.addValue( 7 , series1 , c7 );
       // dataset.addValue( 76 , series1 , c8 );




        return dataset;
    }

    public static void main( String[ ] args ) {

         LineChart1 chart = new LineChart1("Performance Evaluation" , "");
        chart.pack( );
        RefineryUtilities.centerFrameOnScreen( chart );
        chart.setVisible( true );



    }
}
```

```
import java.awt.Color;
import java.awt.Dimension;
import java.awt.GradientPaint;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartPanel;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.axis.CategoryAxis;
import org.jfree.chart.axis.CategoryLabelPositions;
import org.jfree.chart.axis.NumberAxis;
import org.jfree.chart.plot.CategoryPlot;
import org.jfree.chart.plot.PlotOrientation;
import org.jfree.chart.renderer.category.BarRenderer;
import org.jfree.data.category.CategoryDataset;
import org.jfree.data.category.DefaultCategoryDataset;
import org.jfree.ui.ApplicationFrame;
import org.jfree.ui.RefineryUtilities;

/**
 * A simple demonstration application showing how to create a bar chart.
 */
public class PerfGraph extends ApplicationFrame {

    /**
     * Creates a new demo instance.
     *
     * @param title  the frame title.
     */
    public PerfGraph(String title) {

        super(title);
        CategoryDataset dataset = createDataset();
        JFreeChart chart = createChart(dataset);
        ChartPanel chartPanel = new ChartPanel(chart, false);
        chartPanel.setPreferredSize(new Dimension(500, 270));
        setContentPane(chartPanel);

    }

    /**
     * Returns a sample dataset.
     *
     * @return The dataset.
     */
```

```
private static CategoryDataset createDataset() {

    // row keys...
    String series1 = "";

    // column keys...
    String category1 = "18";
    String category2 = "22";
    String category3 = "28";
    String category4 = "34";


    // create the dataset...
    DefaultCategoryDataset dataset = new DefaultCategoryDataset();

    dataset.addValue(52300, series1, category1);
    dataset.addValue(52700, series1, category2);
    dataset.addValue(53050, series1, category3);
    dataset.addValue(53400, series1, category4);


    return dataset;

}

/**
 * Creates a sample chart.
 *
 * @param dataset  the dataset.
 *
 * @return The chart.
 */
private static JFreeChart createChart(CategoryDataset dataset) {

    // create the chart...
    JFreeChart chart = ChartFactory.createBarChart(
        "Performance Chart",         // chart title
        "Review String Length",              // domain axis label
        "Gas Consumption (gas)",                 // range axis label
        dataset,                  // data
        PlotOrientation.VERTICAL, // orientation
        true,                     // include legend
        true,                     // tooltips?
        false                     // URLs?
    );
```

```
// NOW DO SOME OPTIONAL CUSTOMISATION OF THE CHART...

// set the background color for the chart...
chart.setBackgroundPaint(Color.white);

// get a reference to the plot for further customisation...
CategoryPlot plot = chart.getCategoryPlot();
plot.setBackgroundPaint(Color.lightGray);
plot.setDomainGridlinePaint(Color.white);
plot.setDomainGridlinesVisible(true);
plot.setRangeGridlinePaint(Color.white);

// set the range axis to display integers only...
final NumberAxis rangeAxis = (NumberAxis) plot.getRangeAxis();
rangeAxis.setStandardTickUnits(NumberAxis.createIntegerTickUnits());

// disable bar outlines...
BarRenderer renderer = (BarRenderer) plot.getRenderer();
renderer.setDrawBarOutline(false);

// set up gradient paints for series...
GradientPaint gp0 = new GradientPaint(
    0.0f, 0.0f, Color.blue,
    0.0f, 0.0f, new Color(0, 0, 64)
);
GradientPaint gp1 = new GradientPaint(
    0.0f, 0.0f, Color.green,
    0.0f, 0.0f, new Color(0, 64, 0)
);
GradientPaint gp2 = new GradientPaint(
    0.0f, 0.0f, Color.red,
    0.0f, 0.0f, new Color(64, 0, 0)
);
renderer.setSeriesPaint(0, gp0);
renderer.setSeriesPaint(1, gp1);
renderer.setSeriesPaint(2, gp2);

CategoryAxis domainAxis = plot.getDomainAxis();
domainAxis.setCategoryLabelPositions(
    CategoryLabelPositions.createUpRotationLabelPositions(Math.PI / 6.0)
);
// OPTIONAL CUSTOMISATION COMPLETED.

return chart;
```

```
    }

    /**
     * Starting point for the demonstration application.
     *
     * @param args  ignored.
     */
    public static void main(String[] args) {

        PerfGraph demo = new PerfGraph("Performace Evaluation");
        demo.pack();
        RefineryUtilities.centerFrameOnScreen(demo);
        demo.setVisible(true);

    }

}
```

```
import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;
   public class ProcessorLogin extends HttpServlet {
    String username="";
    String password="";
    String type="";
    Connection con=null;
    Statement st=null;
    ResultSet rs=null;

    public void doPost(HttpServletRequest req, HttpServletResponse res) throws
IOException,ServletException {
        username = req.getParameter("username");
        password = req.getParameter("password");

            HttpSession sn = req.getSession(true);
     sn.setAttribute("username",username);

                    RequestDispatcher rd = null;
        System.out.println("User login works"+type);
```

```
        try {

            Class.forName("com.mysql.jdbc.Driver");
            con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/foodnetworkblockchain","root","r
oot");
            st = con.createStatement();
            rs = st.executeQuery("select * from processordetails where username='"+username+"'
&& password='"+password+"'");
            if(rs.next()) {
                 rd=req.getRequestDispatcher("processorpage.jsp");
              }
                 else {

                rd=req.getRequestDispatcher("failure.jsp");

              }
        rd.forward(req,res);


        }
        catch(Exception e2)
         {

            System.out.println("Exception : "+e2.toString());

        }
    }
}
```

```
import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;
  public class ProcessorLogin extends HttpServlet {
   String username="";
   String password="";
   String type="";
   Connection con=null;
   Statement st=null;
   ResultSet rs=null;

   public void doPost(HttpServletRequest req, HttpServletResponse res) throws
IOException,ServletException {
       username = req.getParameter("username");
       password = req.getParameter("password");

          HttpSession sn = req.getSession(true);
    sn.setAttribute("username",username);

                    RequestDispatcher rd = null;
       System.out.println("User login works"+type);
       try {

          Class.forName("com.mysql.jdbc.Driver");
          con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/foodnetworkblockchain","root","r
oot");
          st = con.createStatement();
          rs = st.executeQuery("select * from processordetails where username='"+username+"'
&& password='"+password+"'");
          if(rs.next()) {
              rd=req.getRequestDispatcher("processorpage.jsp");
          }
```

```
            else {

            rd=req.getRequestDispatcher("failure.jsp");

          }
    rd.forward(req,res);


      }
      catch(Exception e2)
       {

         System.out.println("Exception : "+e2.toString());

      }
   }
}
```

```
import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;
  public class ProcessorLogin extends HttpServlet {
   String username="";
   String password="";
   String type="";
   Connection con=null;
   Statement st=null;
   ResultSet rs=null;

   public void doPost(HttpServletRequest req, HttpServletResponse res) throws
IOException,ServletException {
       username = req.getParameter("username");
       password = req.getParameter("password");

          HttpSession sn = req.getSession(true);
    sn.setAttribute("username",username);

                    RequestDispatcher rd = null;
       System.out.println("User login works"+type);
       try {

          Class.forName("com.mysql.jdbc.Driver");
          con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/foodnetworkblockchain","root","r
oot");
          st = con.createStatement();
          rs = st.executeQuery("select * from processordetails where username='"+username+"'
&& password='"+password+"'");
          if(rs.next()) {
              rd=req.getRequestDispatcher("processorpage.jsp");
          }
              else {

             rd=req.getRequestDispatcher("failure.jsp");

          }
    rd.forward(req,res);


       }
       catch(Exception e2)
        {
```

```
            System.out.println("Exception : "+e2.toString());

        }
    }
}
```

```
import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;
   public class ProcessorLogin extends HttpServlet {
    String username="";
    String password="";
    String type="";
    Connection con=null;
    Statement st=null;
    ResultSet rs=null;
```

```
   public void doPost(HttpServletRequest req, HttpServletResponse res) throws
IOException,ServletException {
       username = req.getParameter("username");
       password = req.getParameter("password");

          HttpSession sn = req.getSession(true);
    sn.setAttribute("username",username);

                    RequestDispatcher rd = null;
       System.out.println("User login works"+type);
       try {

          Class.forName("com.mysql.jdbc.Driver");
          con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/foodnetworkblockchain","root","r
oot");
          st = con.createStatement();
          rs = st.executeQuery("select * from processordetails where username='"+username+"'
&& password='"+password+"'");
          if(rs.next()) {
              rd=req.getRequestDispatcher("processorpage.jsp");
          }
              else {

             rd=req.getRequestDispatcher("failure.jsp");

          }
    rd.forward(req,res);


       }
       catch(Exception e2)
        {

          System.out.println("Exception : "+e2.toString());

       }
   }
}
```

```
import java.io.*;
  import java.sql.*;
  import javax.servlet.*;
  import javax.servlet.http.*;
  import java.lang.*;
import java.security.MessageDigest;
import java.text.DecimalFormat;
import java.text.SimpleDateFormat;
 // import de.daslaboratorium.machinelearning.classifier.Classification;
///import de.daslaboratorium.machinelearning.classifier.Classifier;
public class RetailerProcess extends HttpServlet {

Connection con=null;
   Statement st=null,st1=null;
   ResultSet rs=null;
   RequestDispatcher rd=null;
        public void doPost(HttpServletRequest req, HttpServletResponse res) throws
IOException,ServletException {


        HttpSession sn = req.getSession(true);


           String username= sn.getAttribute("username").toString();
```

```
 String id= req.getParameter("id");

   String status="approved";

RequestDispatcher rd;

try {


Class.forName("com.mysql.jdbc.Driver");
con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/foodnetworkblockchain","root","r
oot");

int id1=0;
String prevdata="";
 st = con.createStatement();
ResultSet rs1 = st.executeQuery("select * from blockchain");
while(rs1.next()) {
id1=rs1.getInt(1)+1;
prevdata=rs1.getString(2);
}


    java.util.Date date = new java.util.Date() ;
                SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-
dd HH:mm") ;
                dateFormat.format(date);
                System.out.println(dateFormat.format(date));
    String data1 = prevdata+dateFormat.format(date)+username;



                        StringBuffer sb=new StringBuffer();
                        MessageDigest md = MessageDigest.getInstance("MD5");
                    md.update(String.valueOf(data1).getBytes());

                    byte byteData[] = md.digest();

                    //convert the byte to hex format method 1

                    for (int i = 0; i < byteData.length; i++) {
                     sb.append(Integer.toString((byteData[i] & 0xff) + 0x100,
16).substring(1));
                    }
```

```
File file = new File("C://blockchain//block"+id1+".txt");

//Create the file
if (file.createNewFile())
{
   System.out.println("File is created!");
} else {
   System.out.println("File already exists.");
}

//Write Content
FileWriter writer = new FileWriter(file);
writer.write(sb.toString());
writer.close();

  Statement st1 = con.createStatement();
        int add1=st1.executeUpdate("insert into blockchain
values('"+id1+"','"+sb.toString()+"')");



          String query = "update cropdetails set retailer = ?,hashkey3 = ? where id = ?";
     PreparedStatement preparedStmt = con.prepareStatement(query);
     preparedStmt.setString  (1, username);
    preparedStmt.setString(2, sb.toString());
     preparedStmt.setString(3, id);
     // execute the java preparedstatement
     preparedStmt.executeUpdate();

          rd=req.getRequestDispatcher("retailerstatus.jsp");
          rd.forward(req,res);
       } catch(Exception e2) {
          // rd=req.getRequestDispatcher("failure.jsp");
          System.out.println(e2);
       }
   }
}
```

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

import java.io.IOException;
import java.io.PrintWriter;
import java.net.InetAddress;
import java.net.UnknownHostException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.text.DecimalFormat;
import java.util.Random;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

/**
 *
 * @author java.2
 */
@WebServlet(name = "Register", urlPatterns = {"/Register"})
public class RetailerRegistration extends HttpServlet {

Connection con=null;
   Statement st=null,st1=null;
   ResultSet rs=null;
   RequestDispatcher rd=null;
        public void doPost(HttpServletRequest req, HttpServletResponse res) throws
IOException,ServletException {


        HttpSession sn = req.getSession(true);

          String fullname= req.getParameter("fullname");
        String username= req.getParameter("username");
```

```
        String password= req.getParameter("password");
          String company= req.getParameter("company");
                  String mobile= req.getParameter("mobile");
                  String place= req.getParameter("place");

          RequestDispatcher rd;
        Random generator1 = new Random();
          int seckey = generator1.nextInt(1000000);

          try {


          Class.forName("com.mysql.jdbc.Driver");
          con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/foodnetworkblockchain","root","r
oot");

          st1 = con.createStatement();
         int add=st1.executeUpdate("insert into retailerdetails
values('"+fullname+"','"+username+"','"+password+"','"+company+"','"+place+"','"+mobile+"')
");
       rd=req.getRequestDispatcher("success.jsp");
          rd.forward(req,res);

       } catch(Exception e2)
        {
          System.out.println("Exception : "+e2.toString());
       }
    }
}
```

A.3Screenshots

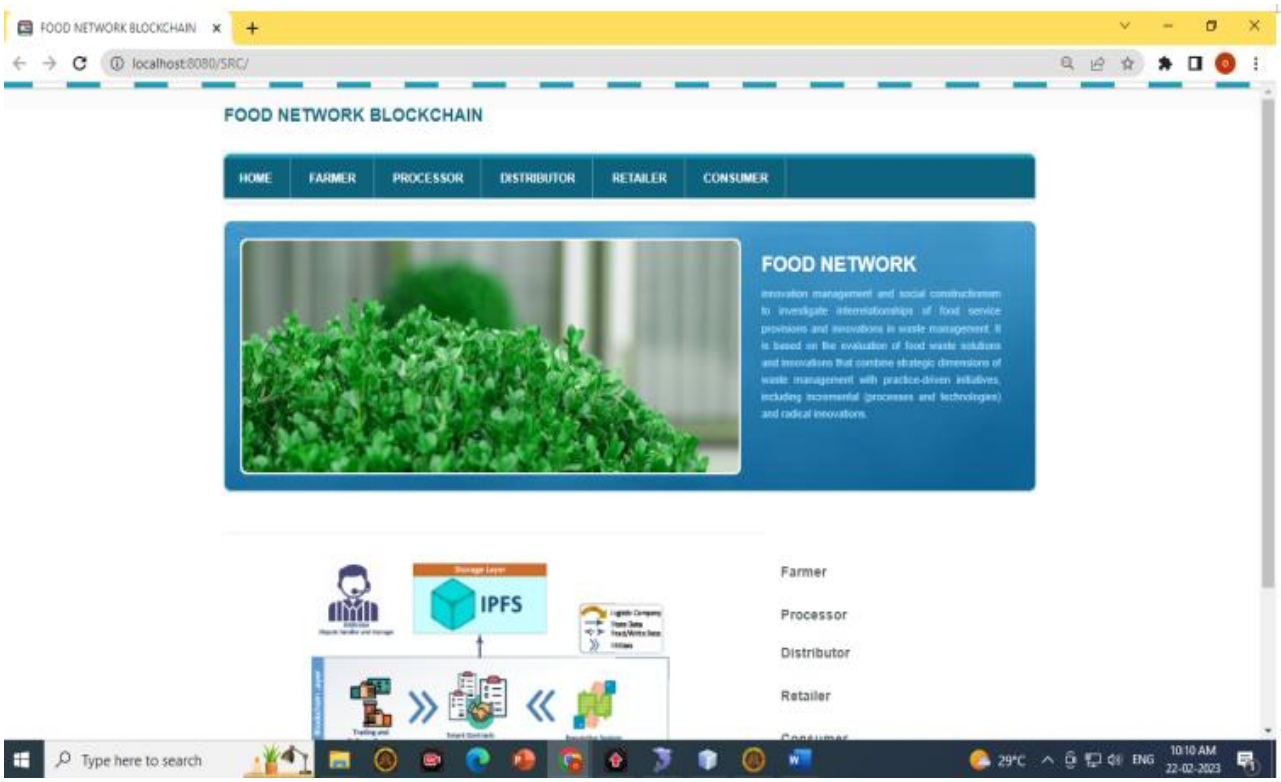


Fig 8.1 Home page

.

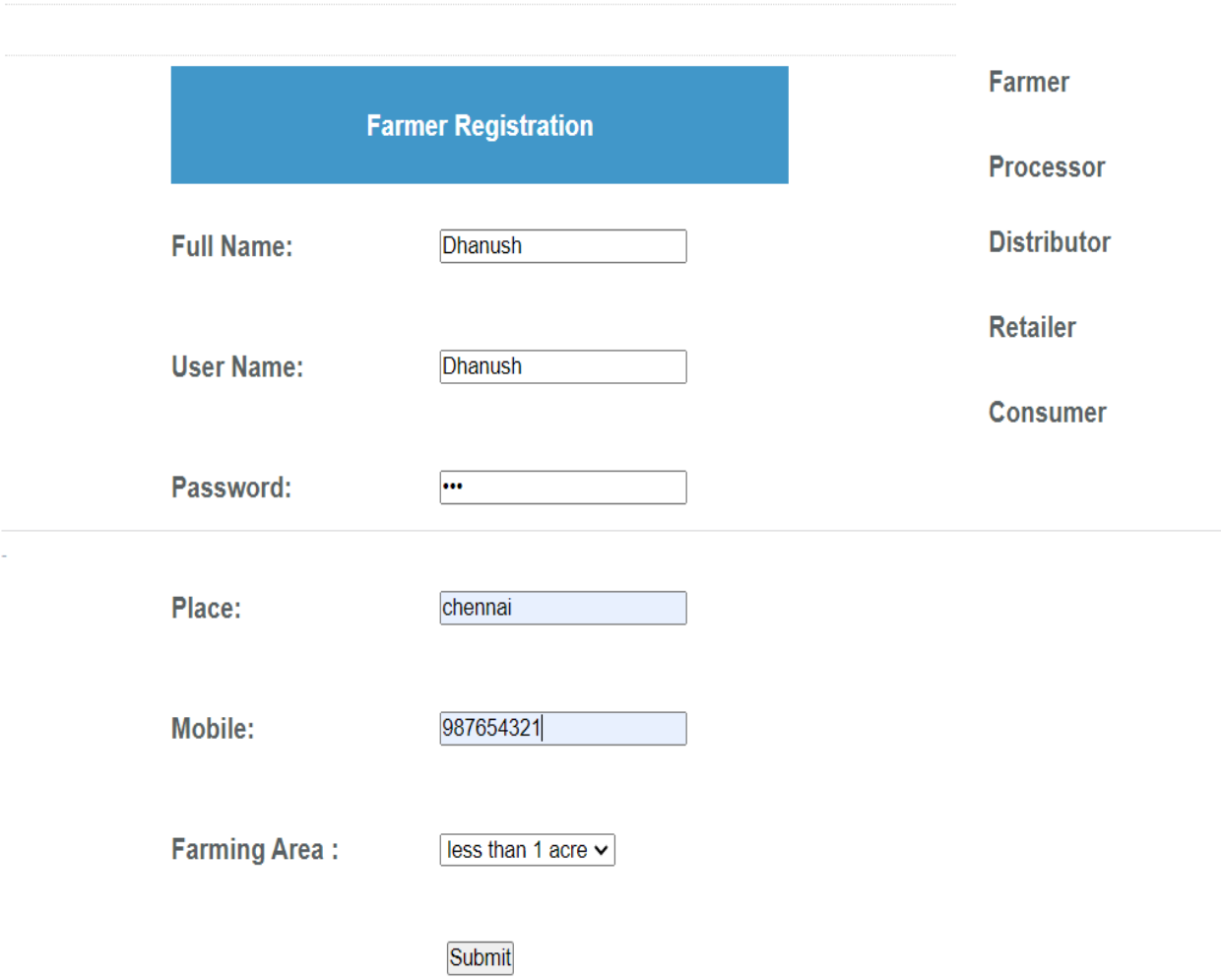


Fig 8.2 Farmer registration page

In fig 8.2 Farmer registration page shows the how user login to the login page by giving the full name,user name ,password and essential details

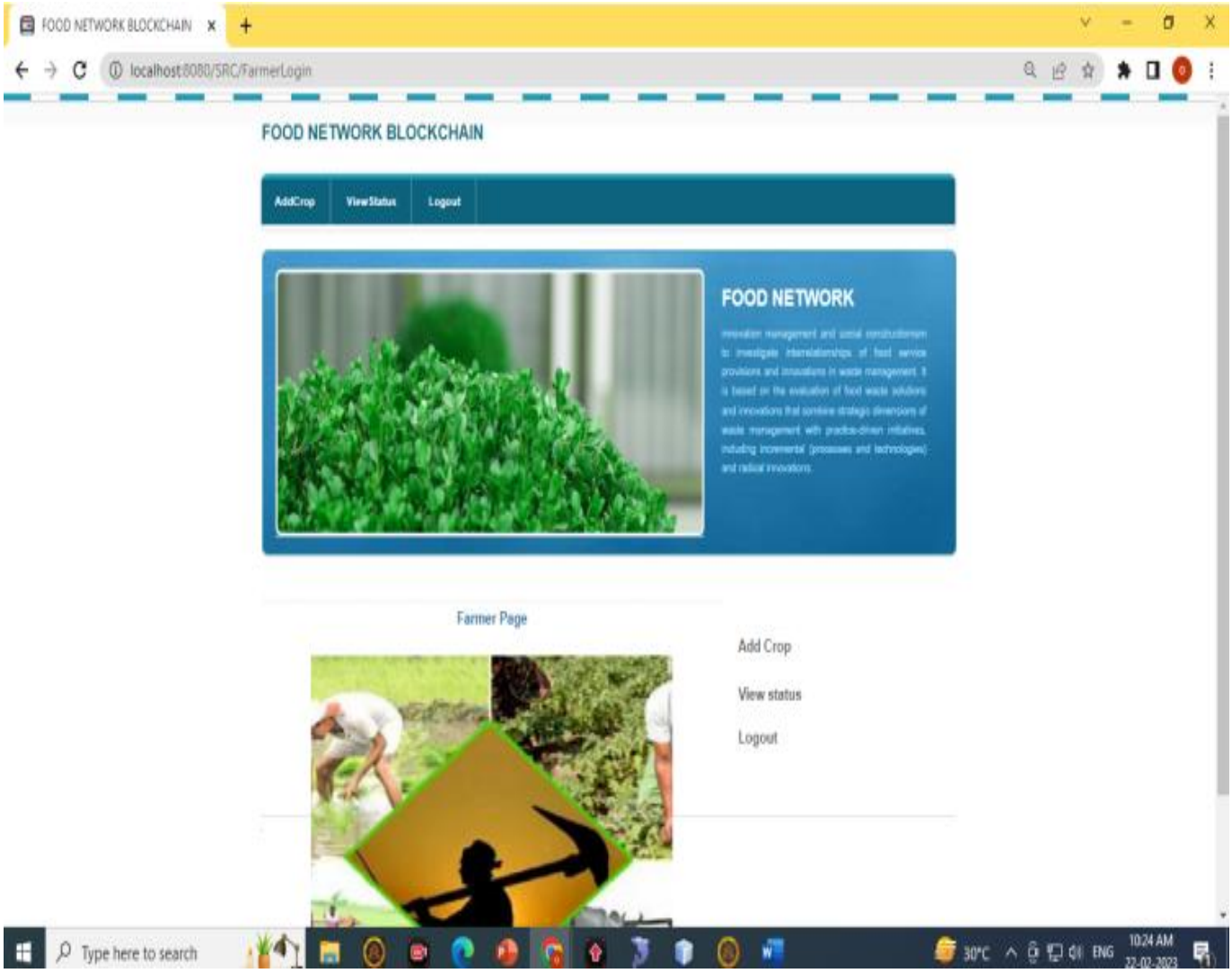


Fig 8.3 Farmer page

Fig 8.3 shows the registration page for farmer to add crop details and viewing status if the farmer adds the crop the processor and distributor can view it

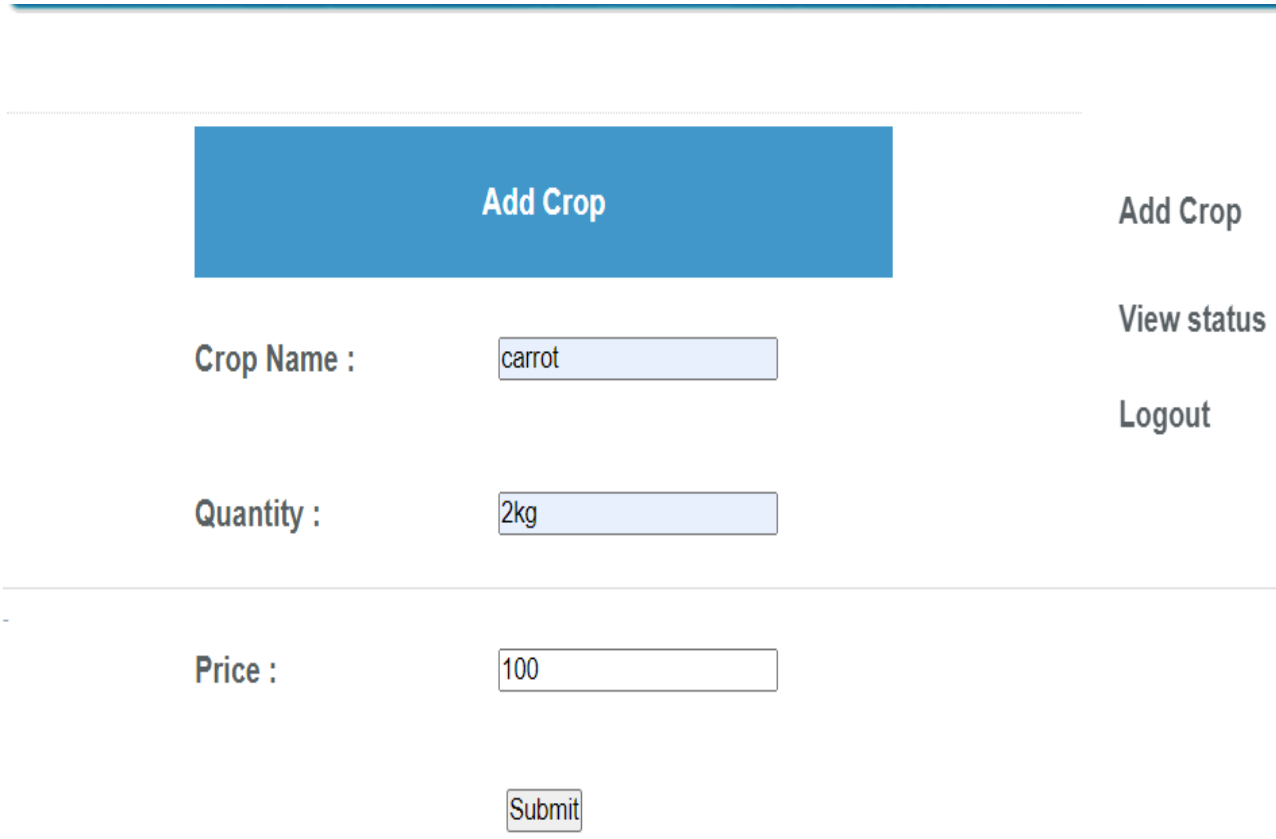


Fig 8.4 Adding crop

In Fig 8.4 shows the crops details to be given by farmer if the crops are added it will be shown to customer