

# AGRICULTURE DOCS CHAIN

## PROJECT REPORT

Date	28 October 2023
Team ID	NM2023TMID03811

## TEAM MEMBERS :

Desika.k
Anuja.M
Gandhimathi.B
Noorul Rashana.A

## INDEX

- 1. INTRODUCTION**
  - Project Overview
  - Purpose
- 2. LITERATURE SURVEY**
  - Existing problem
  - References
  - Problem Statement Definition
- 3. IDEATION & PROPOSED SOLUTION**
  - Empathy Map Canvas
  - Ideation & Brainstorming
- 4. REQUIREMENT ANALYSIS**
  - Functional requirement
  - Non-Functional requirements
- 5. PROJECT DESIGN**
  - Data Flow Diagrams & User Stories
  - Solution Architecture
- 6. PROJECT PLANNING & SCHEDULING**
  - Technical Architecture
  - Sprint Planning & Estimation
  - Sprint Delivery Schedule
- 7. CODING & SOLUTIONING (Explain the features added in the project along with code)**
  - Feature 1
  - Feature 2
  - Database Schema (if Applicable)
- 8. PERFORMANCE TESTING**
  - Performance Metrics
- 9. RESULTS**
  - Output Screenshots
- 10. ADVANTAGES & DISADVANTAGES**
- 11. CONCLUSION**
- 12. FUTURE SCOPE**
- 13. APPENDIX**
  - Source Code
  - GitHub & Project Demo Link

# 1.INTRODUCTION

## 1.1 PROJECT OVERVIEW :

*The Agriculture DocsChain project aims to create a blockchain-based solution for storing and managing agricultural data. It focuses on providing a secure, transparent, and decentralized platform for farmers, agricultural organizations, and stakeholders to store, add, query, and update crucial agricultural information. This system will improve data integrity, accessibility, and trust within the agricultural sector.*

### ***Key Objectives:***

#### ***Decentralized Data Storage:***

*The core feature of Agriculture DocsChain is a decentralized ledger that stores agricultural data in a tamper-proof and transparent manner. All participants in the network will have access to the same data, reducing the risk of data manipulation or fraud.*

#### ***Data Accessibility:***

*Farmers, agricultural researchers, government agencies, and other stakeholders can access the data stored on the blockchain, promoting data sharing and collaboration within the agricultural community.*

#### ***Data Integrity:***

*Data entered into the blockchain is cryptographically secured and immutable. This ensures that historical records and information remain trustworthy and cannot be altered without consensus.*

#### ***User-Friendly Interface:***

*The platform will have a user-friendly interface for users to add, query, and update data. It should be accessible to users with varying levels of technical expertise.*

#### ***Smart Contracts:***

*Implement smart contracts for automated processes, such as crop insurance claims, supply chain tracking, and payments, to streamline operations and reduce human errors*

#### ***Data Standardization:***

*Develop standardized data formats and protocols to ensure compatibility and consistency across the network.*

#### ***Security and Privacy:***

*Implement robust security measures to protect sensitive data and provide users with control over their information.*

### ***Key Components:***

#### *Blockchain Infrastructure:*

*Choose an appropriate blockchain platform Ethereum for building the DocsChain. Ensure scalability and consensus mechanisms that suit the agriculture industry's needs.*

#### *Data Entry and Query Interfaces:*

*Create user-friendly web or mobile applications for data entry, querying, and updating. Implementing a user-friendly interface is essential for broad adoption.*

#### *Smart Contracts:*

*Develop smart contracts to automate processes, including data updates, payments, and validation of transactions.*

#### *Data Standardization:*

*Define data standards and protocols for various types of agricultural data, ensuring interoperability among different stakeholders.*

#### *Identity Management:*

*Implement robust identity and access management systems to protect user data and maintain privacy.*

#### *API Integration:*

*Enable easy integration with existing agricultural systems and data sources, ensuring a seamless flow of information into and out of the blockchain.*

#### *Consensus Mechanisms:*

*Determine the consensus algorithm that best fits the agricultural data use case, considering factors like transaction speed and data integrity.*

#### *Documentation and Training:*

*Provide comprehensive documentation and training materials for users and developers to promote adoption and development on the platform.*

## **1.2 PURPOSE :**

*The blockchain technology allows peer-to-peer transactions to take place transparently and without the need for an intermediary like a bank (such as for cryptocurrencies) or a middleman in the agriculture sector. By eliminating the need for a central authority, the technology changes the way that trust is granted – instead of trusting an authority, trust is placed in cryptography and peer-to-peer architecture. It thus helps restore the trust between producers and consumers, which can reduce the transaction costs in the agri-food market.*

*The blockchain technology offers a reliable approach of tracing transactions between anonymous participants. Fraud and malfunctions can thus be detected quickly. Moreover, problems can be reported in real-time by incorporating smart contract. This helps address the challenge of tracking products in the wide-reaching supply chain due to the complexity of the agri-food system. The technology thus provides solutions to issues of food quality and safety, which are highly concerned by consumers, government, etc.*

*The blockchain technology provides transparency among all involved parties and facilitates the collection of reliable data. Blockchain can record every step in a product's value chain, ranging a product's creation to its death. The reliable data of the farming process are highly valuable for developing data-driven facilities and insurance solutions for making farming smarter and less vulnerable. Moreover we can register and update the food products easily by the owner and query to get the product details by others in this project.*

## 2. LITERATURE SURVEY

### 2.1 EXISTING PROBLEM :

*Lack of Infrastructure: In many rural areas, there is inadequate digital infrastructure, making it challenging to implement blockchain technology effectively.*

*Data Standardization: Agricultural data comes in various formats, making it difficult to standardize and record on a blockchain. This lack of uniformity can lead to inefficiencies.*

*Trust and Adoption: Farmers and other stakeholders may be hesitant to trust and adopt blockchain technology, as they might not fully understand it or have concerns about data security and ownership.*

*Interoperability: Different blockchains might not be compatible with each other, causing issues when multiple entities are involved in the agriculture supply chain.*

*Smart Contract Complexity: Creating and managing smart contracts for various agricultural processes can be complex and may require technical expertise.*

*Costs: Implementing and maintaining a blockchain system can be costly, and it might not be affordable for small-scale farmers or underfunded agricultural organizations.*

*Regulatory Challenges: Regulatory frameworks for blockchain technology in agriculture might not be well-established, causing uncertainty and potential legal issues.*

*Data Ownership and Privacy: Determining who owns and controls the data on the blockchain can be a contentious issue, and ensuring data privacy is a concern.*

*Scalability: As the volume of data and users grows, ensuring that the blockchain can scale without performance issues can be a significant problem.*

*Education and Training: Adequately training farmers and other stakeholders to use the blockchain system can be time-consuming and may face resistance.*

## **2.2 REFERENCES :**

- *Blockchain Technology in Current Agricultural Systems: From Techniques to Applications* Institute of Agricultural Economics and Rural Development, Guangzhou 510640, ChinaF.
- *Tian, "A supply chain traceability system for food safety based on HACCP, blockchain & Internet of Things," in Proc. Int. Conf. service Syst. service Manage., 2017, pp. 1–6*
- *.G. Zhao, S. Liu, C. Lopez, H. Lu, S. Elgueta, H. Chen, and B. M. Boshkoska, "Blockchain technology in agri-food value chain management: A synthesis of applications, challenges and future research directions," Comput. Ind., vol. 109, pp. 83–99, Aug. 2019.*
- *H. Xiong, T. Dalhaus, P. Wang, and J. Huang, "Blockchain technology for agriculture: Applications and rationale," Frontiers Blockchain, vol. 3, p. 7, Feb. 2020.*
- *F. Antonucci, S. Figorilli, C. Costa, F. Pallottino, L. Raso, and P. Menesatti, "A review on blockchain applications in the agri-food sector," J. Sci. Food Agricult., vol. 99, no. 14, pp. 6129–6138, 2019*
- *M. Lezoche, J. E. Hernandez, M. D. M. E. Alemany Díaz, H. Panetto, and J. Kacprzyk, "Agri-food 4.0: A survey of the supply chains and technologies for the future agriculture," Comput. Ind., vol. 117, May 2020, Art. no. 103187.*
- *G. Sylvester, "E-agriculture in action: Blockchain for agriculture: Opportunities and challenges," Bangkok Int. Telecommun. Union (ITU), 2019, pp. 27–29.*

### **2.3 PROBLEM STATEMENT :**

*In the agricultural industry, the management and verification of critical documentation, including land records, crop data, supply chain information, and certifications, are often inefficient, prone to fraud, and lack transparency.*

*Currently all systems rely on paper-based records, multiple intermediaries, and manual data entry, which can lead to errors, disputes, and a lack of trust within the ecosystem. Additionally, there is a growing need for secure and transparent data sharing between farmers, suppliers, regulators, and consumers.*

*The unique and best technology is needed to solve these problems.*

### 3. IDEATION & PROPOSED SOLUTION

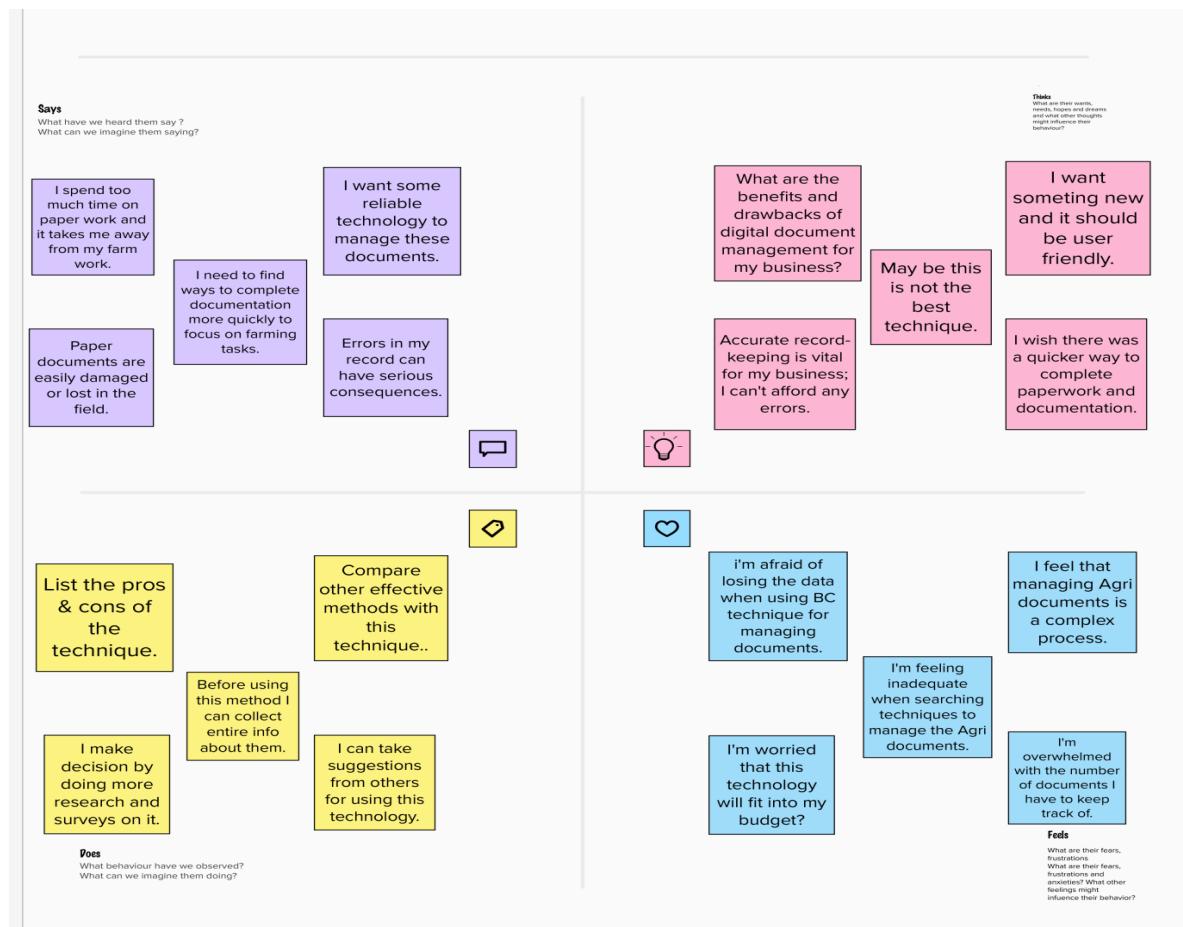
#### 3.1 EMPATHY MAP CANVAS :

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes.

It is a useful tool to help teams better understand their users.

Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.

Reference: <https://www.mural.co/templates/empathy-map-canvas>



## 3.2 IDEATION & BRAINSTROMING :

- Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.
- Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

Reference: <https://www.mural.co/templates/empathy-map-canvas>

The screenshot shows the 'Brainstorm & idea prioritization' template from Mural. The interface is divided into several sections:

- Header:** A circular icon containing a lightbulb inside a speech bubble.
- Title:** 'Brainstorm & idea prioritization'
- Description:** 'Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.'
- Preparation Time:** '10 minutes to prepare', '1 hour to collaborate', '2-8 people recommended'.
- Step 1: Define your problem statement:** A section titled 'Define your problem statement' with a sub-section 'PROBLEM'. It contains a box with text about food production challenges and a note: 'In farming and food production, there are problems with tracking documents and papers in a safe and reliable way. This makes it hard for people to trust the information and can cause mistakes and delays in getting food to stores. To solve this, we can use a system like AgriDocChain that uses technology to keep all the important documents safe and easy to access.' A timer icon indicates '5 minutes'.
- Key rules of brainstorming:** A section titled 'Key rules of brainstorming' with the subtitle 'To run a smooth and productive session'. It lists six rules with icons:
  - Stay in topic.
  - Encourage wild ideas.
  - Defer judgment.
  - Listen to others.
  - Go for volume.
  - If possible, be visual.
- Feedback and sharing:** Buttons for 'Share template feedback' and 'Weather-Integrated Records Link your...'.

## 2

### Brain Storm

write down any ideas that comes to mind that address your problem statement.

#### Person 1

AI can be applied for document classification, data extraction, and processing.

Using Mobile apps to capture and upload documents.

Invest in a dedicated document management software system that is tailored to the unique requirements of the agriculture industry.

Establish an online platform where farmers can share their agricultural documents, experiences, and best practices with one another.

#### Person 2

Cloud-based document management systems allow for easy access to documents from anywhere,

Sensors for soil, weather, and plant health can transmit data directly to document management systems.

Using a CMS like WordPress or Joomla to organize and manage documents on a website.

Set up audit logs to record and monitor all activities related to document access and management

#### Person 3

Using Voice Recognition Technology

GIS technology can be used to map and visualize agricultural data,

Utilize farm management software platforms that centralize data on crop planting, cultivation, and harvest.

Regularly back up agricultural documents and data to prevent data loss in case of system failures or cyberattacks.

#### Person 4

Create a blockchain based platform for agriculture documents ensuring data security & transparency.

Implementing Scanner & OCR to convert physical DOC into Digital format.

Install firewalls and intrusion detection systems to protect the sensitive data,

Develop a customized database to store and manage agricultural documents.

## Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you can break it up into smaller sub-groups.

⌚ 20 minutes

Create a blockchain based platform for agriculture documents ensuring data security & transparency.

Using Mobile apps to capture and upload documents.

Using a CMS like WordPress or Joomla to organize and manage documents on a website.

Implementing Scanner & OCR to convert physical DOC into Digital format.

**Using Voice Recognition Technology**

Sensors for soil, weather, and plant health can transmit data directly to document management systems.

Establish an online platform where farmers can share their agricultural documents, experiences, and best practices with one another.

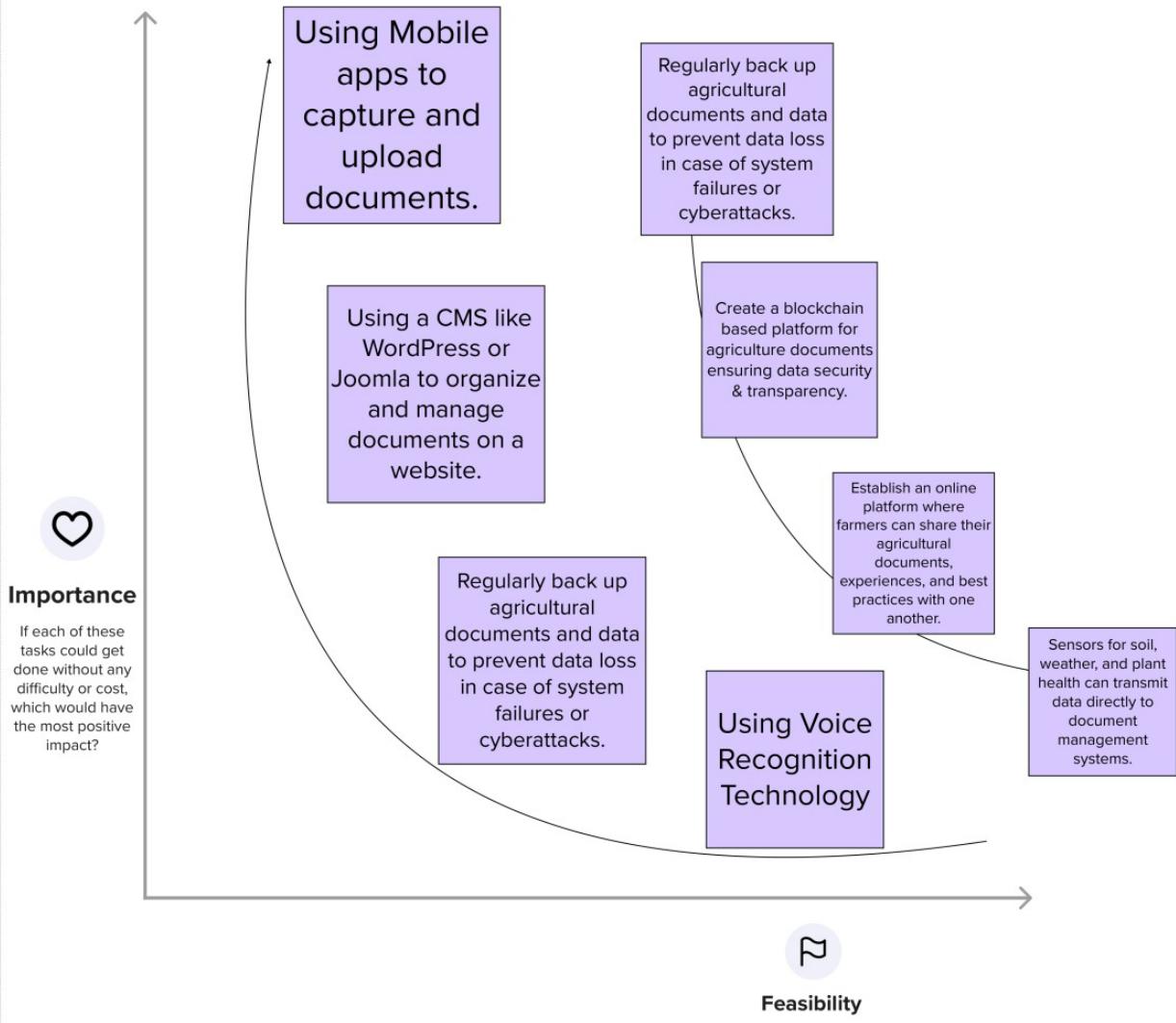
Regularly back up agricultural documents and data to prevent data loss in case of system failures or cyberattacks.

4

### Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

⌚ 20 minutes



## 4. REQUIREMENT ANALYSIS

Project Design Phase-II

### 4.1 Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	<p>Users should be able to register and create accounts.</p> <p>Users should be able to log in securely.</p> <p>Implement role-based access control for different types of users (e.g., farmers, suppliers, regulators).</p>
FR-2	Secure Transactions and Payments	<p>Facilitate secure financial transactions, including payments between stakeholders.</p> <p>Implement encryption and security measures to protect sensitive financial data.</p>
FR-3	Notification and Alerts	<p>Send notifications and alerts to users for critical events, such as contract updates, compliance violations, or supply chain disruptions.</p>
FR-4	User Support	<p>Offer customer support and resources for users to resolve issues or answer questions.</p>

## 4.2 Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

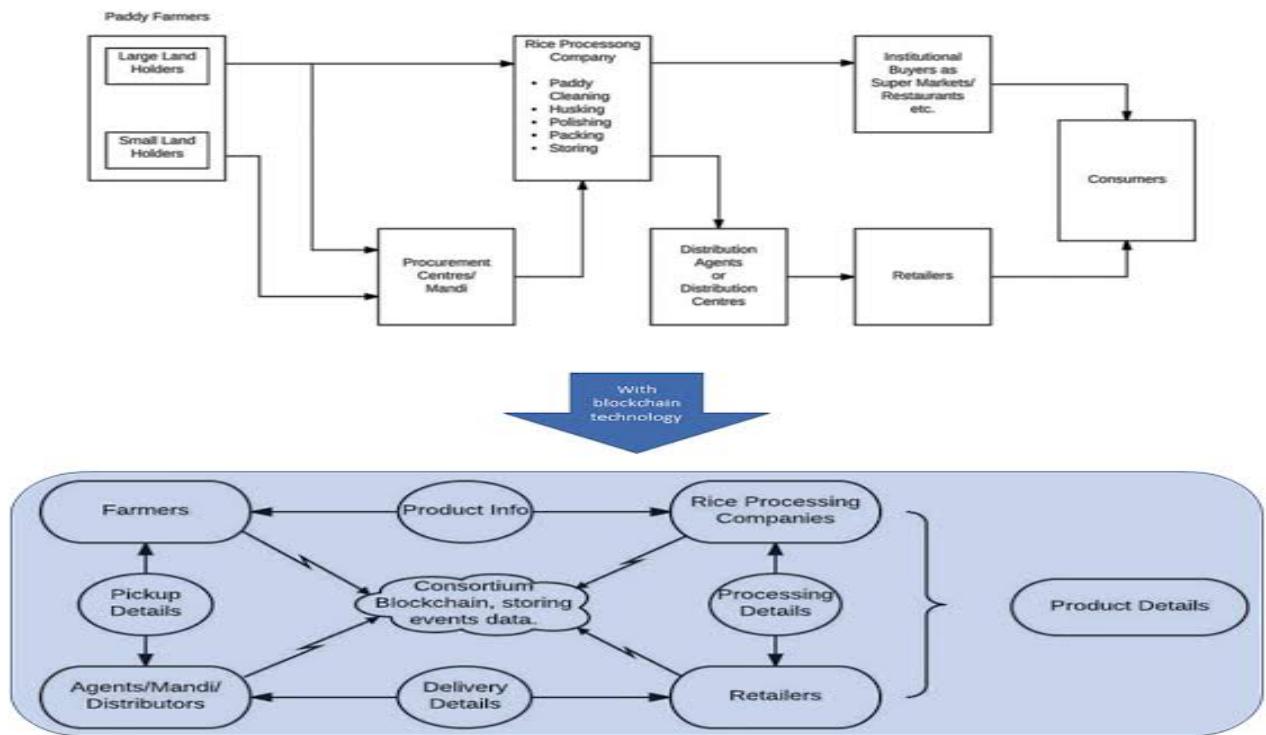
FR No.	Non-Functional Requirement	Description
NFR-1	<b>Usability</b>	<p>Usability requirements focus on the user-friendliness of the system.</p> <p>Ensure that the system has an intuitive and easy-to-use interface for all types of users, including those with varying levels of technical expertise</p>
NFR-2	<b>Security</b>	<p>Implement strong encryption for data at rest and data in transit.</p> <p>Enforce access controls and authentication mechanisms to prevent unauthorized access.</p>
NFR-3	<b>Reliability</b>	<p>Reliability requirements address the system's ability to perform consistently and predictably.</p> <p>Ensure that the system operates without unexpected downtime or data loss.</p>
NFR-4	<b>Performance</b>	<p>Define the system's speed, responsiveness, and resource utilization.</p> <p>Specify acceptable response times for various operations, such as document retrieval, contract execution, and data analysis.</p>
NFR-5	<b>Availability</b>	<p>Availability requirements describe the system's uptime and accessibility.</p> <p>Set a target availability percentage (e.g., 99.9%) and downtime limitations.</p>
NFR-6	<b>Scalability</b>	<p>Focus on the system's ability to handle increased load as it grows.</p> <p>Design the system architecture to be scalable, enabling it to accommodate a growing user base and data volume.</p>

## 5. PROJECT DESIGN

### 5. 1 DATA FLOW DIAGRAMS & USER STORIES:

#### *Data Flow Diagram:*

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



### **USER STORIES :**

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Team Member
Farmer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Desika
		USN-2	As a user, I will receive confirmation email once I have registered for the application.	I can receive confirmation email & click confirm	High	Anuja
		USN-3	As a user, I can register for the application through Gmail.	I can get notification	Low	Gandhimathi
	Login	USN-4	As a user, I can log into the application through my email & password.	The login page must provide fields for entering a username and password	High	Noorul Rashana
		USN-5	The login page should have a "Remember Me" checkbox	Allows the user to remain logged in between sessions.	High	Desika
	Dashboard	USN-6	I want to have a dashboard that provides an overview of relevant information and features to help me manage my account or access important data.	The dashboard should include widgets or components that offer at-a-glance information, such as recent activity, account status etc	Medium	Anuja
Customer (Web user)		USN-7	Users should have the ability to perform common tasks directly from the dashboard.	Tasks such as creating new records, viewing recent notifications	High	Noorul Rashana
Supply Chain developer	Username	USN-8	As a user of the application, I want to have the ability to set or change my username, ensuring it's unique and reflective of my	Users should have the option to change their username after	High	Gandhimathi

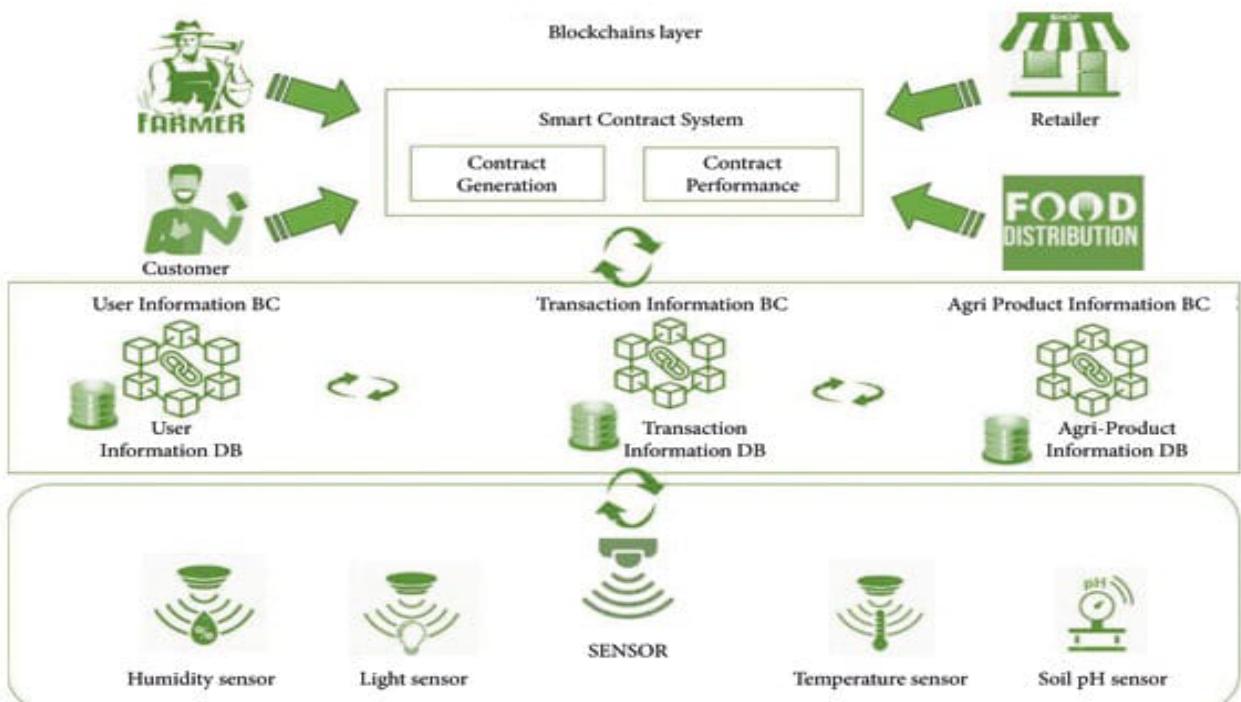
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Team Member
			identity within the system.	registration.		
		USN-9	Users should receive immediate feedback during the username creation or change process.	Indicating whether their chosen username is valid and available.	High	Anuja
Administrator	Membership	USN-10	I want to manage the membership records of our farmer members on AgriDocchain.	It facilitate information sharing among them	Medium	Gandhimathi

## 5.2 SOLUTION ARCHITECTURE:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

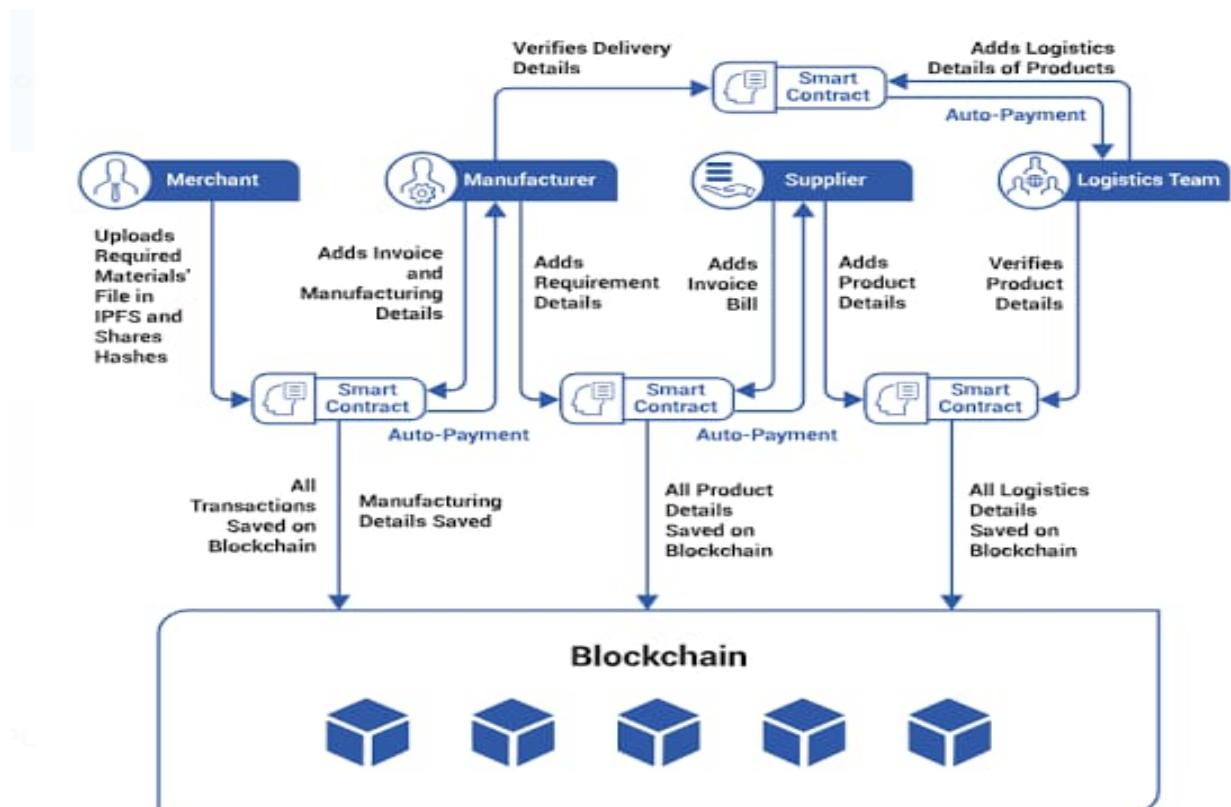
**Solution Architecture Diagram**



## 6. PROJECT PLANNING & SCHEDULING

### 6.1 TECHNICAL ARCHITECTURE:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2



**FIG: TECHNICAL ARCHITECTURE**

**Table-1 : Components & Technologies:**

S.No	Component	Description	Technology
1.	User Interface	Develop user-friendly web interfaces	HTML, CSS, and JavaScript.
2.	Database	The core of the blockchain, it stores all transactions and data.	Distributed Ledger
3.	Cloud Services	Host your application on cloud platforms like for high availability	AWS, Azure, or Google Cloud
4.	File Storage	It will depend on the specific use cases, data volume, and regulatory constraints.	Centralized Storage, Cloud Storage, GlusterFS
5.	External API	For integrating with other systems and enabling data exchange.	REST or GraphQL
6.	Data Analytics Tools	Utilize data analytics tools for extracting insights from supply chain data.	Apache Spark or Elasticsearch
7.	Reporting Tools	Implement reporting frameworks for generating visual reports.	Tableau or Power BI
8.	Infrastructure	Involves a combination of hardware, software, and network components to support the application's functionality and performance	BC Nodes, Firewalls and Security, Kubernetes, etc.

**Table-2: Application Characteristics:**

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Agridocschain should utilize open-source blockchain frameworks and this provides transparency, community support.	Ethereum, Hyperledger Fabric & Corda
2.	Security Implementations	Security is paramount, especially in a Doc chain management system. It should implement robust security measures.	SHA-256, Encryptions, Digital Certificates
3.	Scalable Architecture	This involves the ability to handle increased data and transaction loads without compromising performance or security.	Load Balancers, Docker and Kubernetes
4.	Availability	Implementing redundancy, failover mechanisms,	Cloud Service, Content Delivery

S.No	Characteristics	Description	Technology
		and 24/7 support is essential to ensure continuous availability.	Networks (CDNs)
5.	Performance	This includes efficient transaction processing, low latency, and the ability to handle a large number of concurrent users and data without slowdowns.	Caching Mechanism, Database Sharding & CDN

## 6. 2 SPRINT PLANNING & EXECUTION:

*Sprint planning and execution in blockchain Agri Docs chain involves following steps to be taken to finish an overall quality project on time.*

S.NO	SPRINTS	DESCRIPTION	EXECUTE BY
01	Ideation and proposed solution	Empathy map Canvas, Ideation & Brainstroming.	Gandhimathi.B
02	Requirements analysis	Functional Requirements Non Functional Requirements	Anuja.M
03	Project Design	Dataflow diagrams & User Stories Solution Architecture	Gandhimathi.B
04	Project planning	Technical Architecture	Anuja.M
05	Tools Requirements	Download VS code Node.js Metamask	Noorul Rashana.A
06	Zip File Execution	Download zipfile & extract it & open it in VScode	Desika.k
07	Remix and Metamask	Execution of Solidity code and adding Metamask Extension to Remix.	Desika.k Noorul Rashana.A
08	Implementation of Smart Contract & Interaction of frontend	Using file Connector.js Interaction with frontend .	Desika.k Noorul Rashana.A
09	Performance testing	Testing of codes and project output	Noorul Rashana.A Desika.K
10	Project Review	Review of output & results	Anuja.M Desika.k Gandhimathi.B Noorul Rashana.A

## 6. 3 SPRINT DELIVERY SCHEULE:

S.NO	SPRINTS	START DATE	END DATE	STATUS
01	Ideation and proposed solution	20.10.2023	22.10.2023	Finished
02	Requirements analysis	22.10.2023	23.10.2023	Finished
03	Project Design	24.10.2023	26.10.2023	Finished
04	Project planning	27.10.2023	28.10.2023	Finished
05	Tools Requirements & tools download	20.10.2023	22.10.2023	Finished
06	Zip File download & Extraction	22.10.2023	23.10.2023	Finished
07	Solidity code and adding Metamask Extention	24.10.2023	26.10.2023	Finished
08	Implementation of Smart Contract & Interaction of frontend	26.10.2023	28.10.2023	Finished
09	Performance testing Project Review	28.10.2023	29.10.2023	Review
10	Demo video creation	29.10.2023	29.10.2023	On Progress

## 7. CODING AND SOLUTIONING

### 7.1 FEATURE 1:

#### *Software – Remix Etherium IDE*

Remix is a software application, specifically an Integrated Development Environment (IDE), designed for Ethereum smart contract development. It provides a web-based platform for Ethereum developers to write, compile, deploy, and test smart contracts, making it easier to create decentralized applications (DApps) on the Ethereum blockchain.

It provides full support for the Solidity programming language, which is commonly used for Ethereum smart contract development. Remix can compile Solidity code into Ethereum bytecode, which is essential for deploying contracts on the Ethereum blockchain.

Remix provides a testing environment for writing and running unit tests for smart contracts, helping developers ensure their code functions as expected

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract AgricultureRegistry {
    struct foodProduct {
        string name;
        string description;
        uint256 quantity;
        address owner;
    }

    mapping(uint256 => foodProduct) public products;
    uint256 public productCount;

    event ProductAdded(uint256 productId, string name, string description, uint256 quantity, address owner);
    event ProductUpdated(uint256 productId, string name, string description, uint256 quantity);

    modifier onlyOwner(uint256 _productId) {
        require(products[_productId].owner == msg.sender, "Only the owner can perform this action");
        -
    }
}
```

```

function addProduct(uint256 ProductId, string memory _name, string memory _description, uint256
_qty) external {

    products[ProductId] = foodProduct(_name, _description, _qty, msg.sender);
    productCount++;
    emit ProductAdded(productCount, _name, _description, _qty, msg.sender);
}

function updateProduct(uint256 _productId, string memory _name, string memory _description,
uint256 _qty) external onlyOwner(_productId) {
    foodProduct storage product = products[_productId];
    product.name = _name;
    product.description = _description;
    product.quantity = _qty;
    emit ProductUpdated(_productId, _name, _description, _qty);
}

function getProductDetails(uint256 _productId) external view returns (string memory name, string
memory description, uint256 quantity, address owner) {
    foodProduct memory product = products[_productId];
    return (product.name, product.description, product.quantity, product.owner);
}
}

```

## 7.2 FEATURE 2:

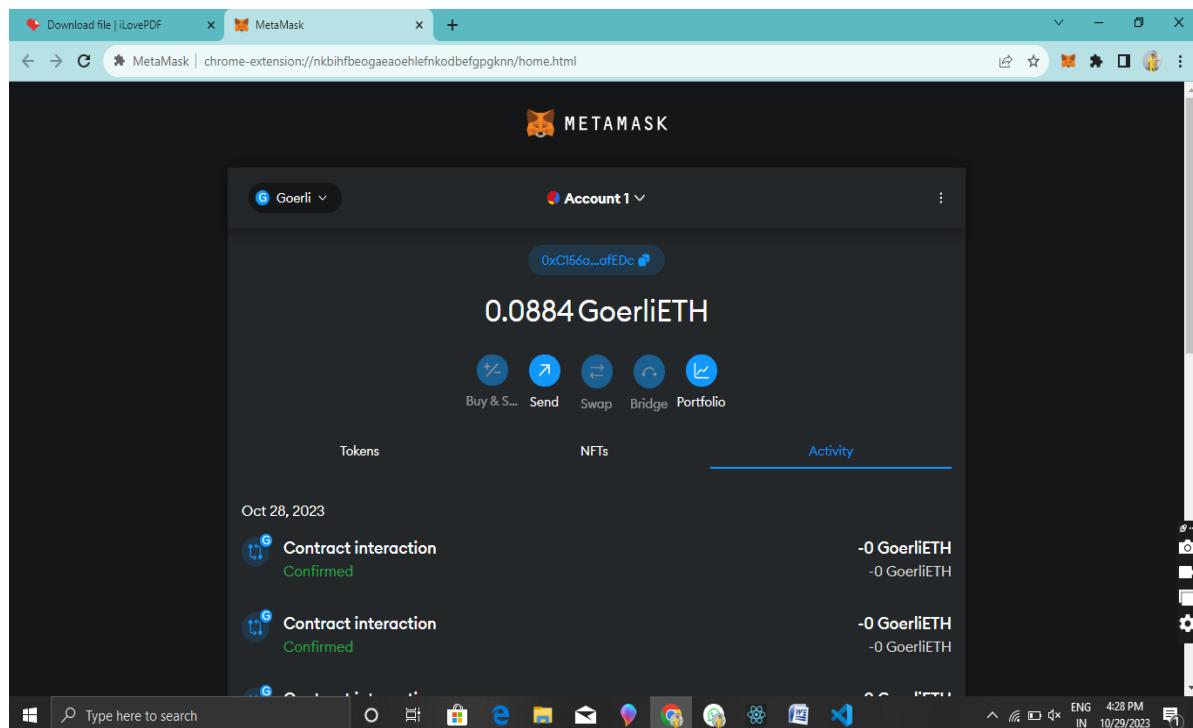
### **Metamask :**

MetaMask is a popular cryptocurrency wallet and decentralized application (DApp) browser extension. It allows users to manage their Ethereum-based assets, interact with Ethereum-based DApps, and store private keys securely.

Users can access their wallet and interact with decentralized applications directly from their web browser.

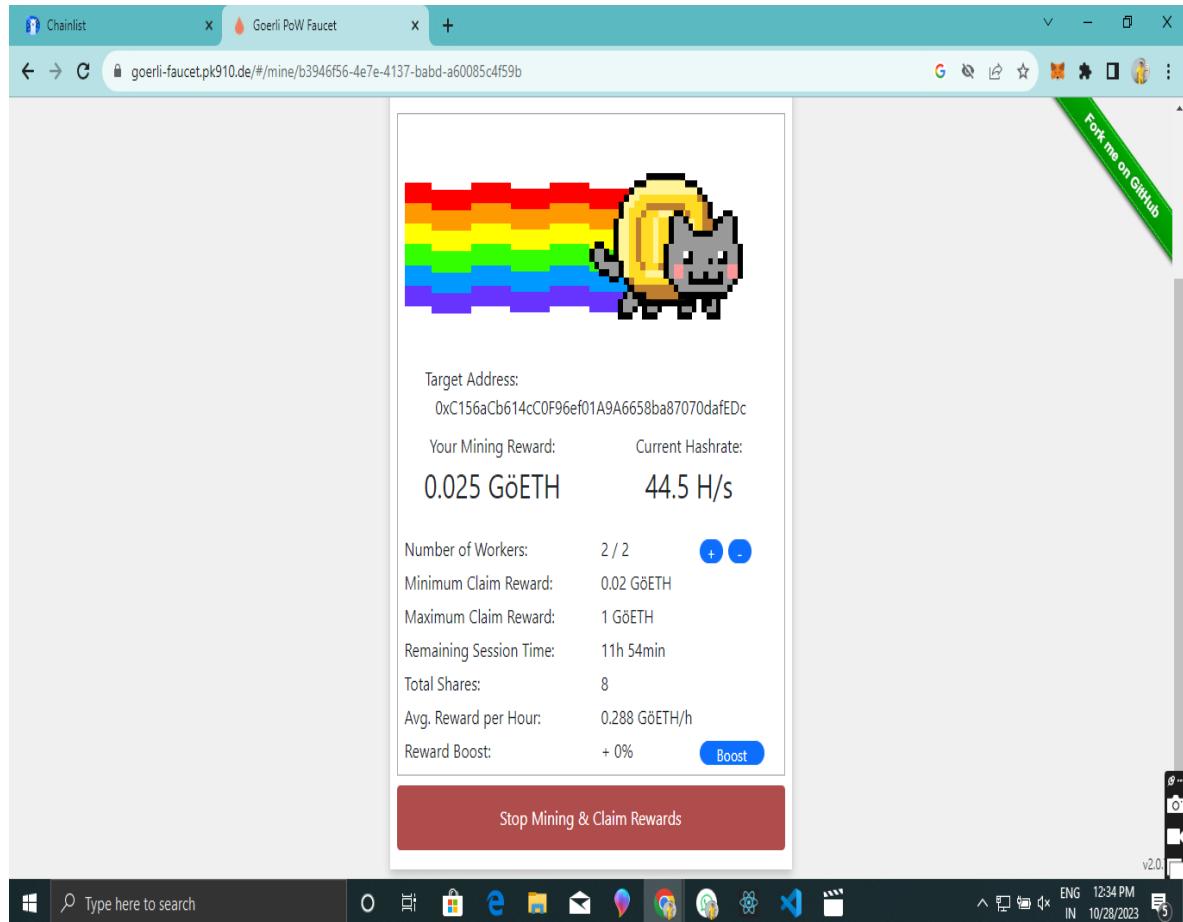
Can be easily manage Ethereum-based tokens within MetaMask, including sending, receiving, and swapping tokens.

It's available as a browser extension for major browsers like Chrome, Firefox, and Brave, and as a mobile app for iOS



## **Goerli PoW Faucet:**

- This is an Ethereum Faucet running on the Goerli Testnet.
- To prevent malicious actors from exhausting all available funds or accumulating enough Ether to mount long running spam attacks, this faucet requires some mining work to be done in exchange for free testnet funds
- Just enter your ETH Address and start mining. When you've collected enough ETH, stop mining and claim your rewards.



## **CODINGS:**

### **Frontend Codings:**

CSS code: (App.css)

```
.App {  
  text-align: center;  
}  
  
.App-logo {  
  height: 40vmin;  
  pointer-events: none;  
}  
  
@media (prefers-reduced-motion: no-preference) {  
  .App-logo {  
    animation: App-logo-spin infinite 20s linear;  
  }  
}  
  
.App-header {  
  background-color: #282c34;  
  min-height: 100vh;  
  display: flex;  
  flex-direction: column;  
  align-items: center;  
  justify-content: center;  
  font-size: calc(10px + 2vmin);  
  color: white;  
}  
  
.App-link {  
  color: #61dafb;  
}  
  
@keyframes App-logo-spin {  
  from {  
    transform: rotate(0deg);  
  }  
  to {  
    transform: rotate(360deg);  
  }  
}
```

### CSS code: (index.css)

```
body {  
    margin: 0;  
    font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',  
    'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',  
    sans-serif;  
    -webkit-font-smoothing: antialiased;  
    -moz-osx-font-smoothing: grayscale;  
}  
  
code {  
    font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',  
    monospace;  
}
```

### JAVA SCRIPTS CODE: ( connector.js )

```
const { ethers } = require("ethers");  
  
const abi = [  
    {  
        "anonymous": false,  
        "inputs": [  
            {  
                "indexed": false,  
                "internalType": "uint256",  
                "name": "productId",  
                "type": "uint256"  
            },  
            {  
                "indexed": false,  
                "internalType": "string",  
                "name": "name",  
                "type": "string"  
            },  
            {  
                "indexed": false,
```

```
"internalType": "string",
"name": "description",
"type": "string"
},
{
"indexed": false,
"internalType": "uint256",
"name": "quantity",
"type": "uint256"
},
{
"indexed": false,
"internalType": "address",
"name": "owner",
"type": "address"
}
],
{
"name": "ProductAdded",
"type": "event"
},
{
"anonymous": false,
"inputs": [
{
"indexed": false,
"internalType": "uint256",
"name": "productId",
"type": "uint256"
},
{
"indexed": false,
"internalType": "string",
"name": "name",
"type": "string"
},
{
"indexed": false,
"internalType": "string",
"name": "description",
"type": "string"
},
{
"indexed": false,
"internalType": "uint256",
"name": "quantity",
"type": "uint256"
}
```

```

    },
],
"name": "ProductUpdated",
"type": "event"
},
{
"inputs": [
{
"internalType": "uint256",
"name": "ProductId",
"type": "uint256"
},
{
"internalType": "string",
"name": "_name",
"type": "string"
},
{
"internalType": "string",
"name": "_description",
"type": "string"
},
{
"internalType": "uint256",
"name": "_quantity",
"type": "uint256"
}
],
"name": "addProduct",
"outputs": [],
"stateMutability": "nonpayable",
"type": "function"
},
{
"inputs": [
{
"internalType": "uint256",
"name": "_productId",
"type": "uint256"
}
],
"name": "getProductDetails",
"outputs": [
{
"internalType": "string",
"name": "name",

```

```

    "type": "string"
},
{
  "internalType": "string",
  "name": "description",
  "type": "string"
},
{
  "internalType": "uint256",
  "name": "quantity",
  "type": "uint256"
},
{
  "internalType": "address",
  "name": "owner",
  "type": "address"
}
],
"stateMutability": "view",
"type": "function"
},
{
  "inputs": [],
  "name": "productCount",
  "outputs": [
    {
      "internalType": "uint256",
      "name": "",
      "type": "uint256"
    }
  ],
  "stateMutability": "view",
  "type": "function"
},
{
  "inputs": [
    {
      "internalType": "uint256",
      "name": "",
      "type": "uint256"
    }
  ],
  "name": "products",
  "outputs": [
    {
      "internalType": "string",

```

```

    "name": "name",
    "type": "string"
},
{
  "internalType": "string",
  "name": "description",
  "type": "string"
},
{
  "internalType": "uint256",
  "name": "quantity",
  "type": "uint256"
},
{
  "internalType": "address",
  "name": "owner",
  "type": "address"
}
],
"stateMutability": "view",
"type": "function"
},
{
  "inputs": [
    {
      "internalType": "uint256",
      "name": "_productId",
      "type": "uint256"
    },
    {
      "internalType": "string",
      "name": "_name",
      "type": "string"
    },
    {
      "internalType": "string",
      "name": "_description",
      "type": "string"
    },
    {
      "internalType": "uint256",
      "name": "_quantity",
      "type": "uint256"
    }
  ],
  "name": "updateProduct",

```

```
"outputs": [],
"stateMutability": "nonpayable",
"type": "function"
}
]

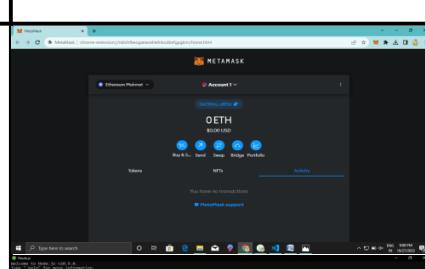
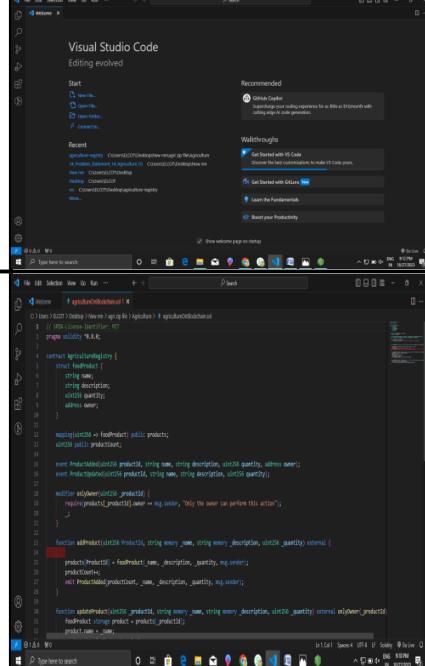
if (!window.ethereum) {
  alert('Meta Mask Not Found')
  window.open("https://metamask.io/download/")
}

export const provider = new ethers.providers.Web3Provider(window.ethereum);
export const signer = provider.getSigner();
export const address = "0xC62527f5e9D4ca3b0f2A1c808B1a172f158a557"

export const contract = new ethers.Contract(address, abi, signer)
```

## 8. PERFORMANCE TESTING

### 8. 1 PERFORMANCE METRICS :

S.No.	Parameter	Values	Screenshot
1.	Information gathering	Setup all the Prerequisite: 1.node.js 2.vs code 3.metamask	
2.	Extract the zip files	Open to vs code	

### 3. Remix Ide platform exploring

Deploy the smart contract code

Deploy and run the transaction. By selecting the environment - inject the MetaMask.

The image consists of three vertically stacked screenshots of the Remix IDE interface, showing the process of deploying and running a smart contract named "AgricreateMarket".

- Screenshot 1:** Shows the "DEPLOY & RUN TRANSACTIONS" tab. It displays the contract source code in Solidity. The "GAS LIMIT" is set to 3000000. The "ACCOUNT" dropdown shows "0x54...". The "CONTACT" section shows the contract address "0x54...". The "Transactions recorded" section is empty. The "Deployed Contracts" section also shows the deployed contract.
- Screenshot 2:** Shows the same interface after deployment. The "GAS LIMIT" is now set to 1000000. The "ACCOUNT" dropdown shows "0x54... (MetaMask)". The "CONTACT" section shows the contract address "0x54...". The "Transactions recorded" section shows one transaction with hash "0x54...". The "Deployed Contracts" section shows the deployed contract.
- Screenshot 3:** Shows the "DEPLOY & RUN TRANSACTIONS" tab again. The "GAS LIMIT" is set to 3000000. The "ACCOUNT" dropdown shows "0x54... (MetaMask)". The "CONTACT" section shows the contract address "0x54...". The "Transactions recorded" section shows two transactions with hashes "0x54..." and "0x54...". The "Deployed Contracts" section shows the deployed contract.

4.

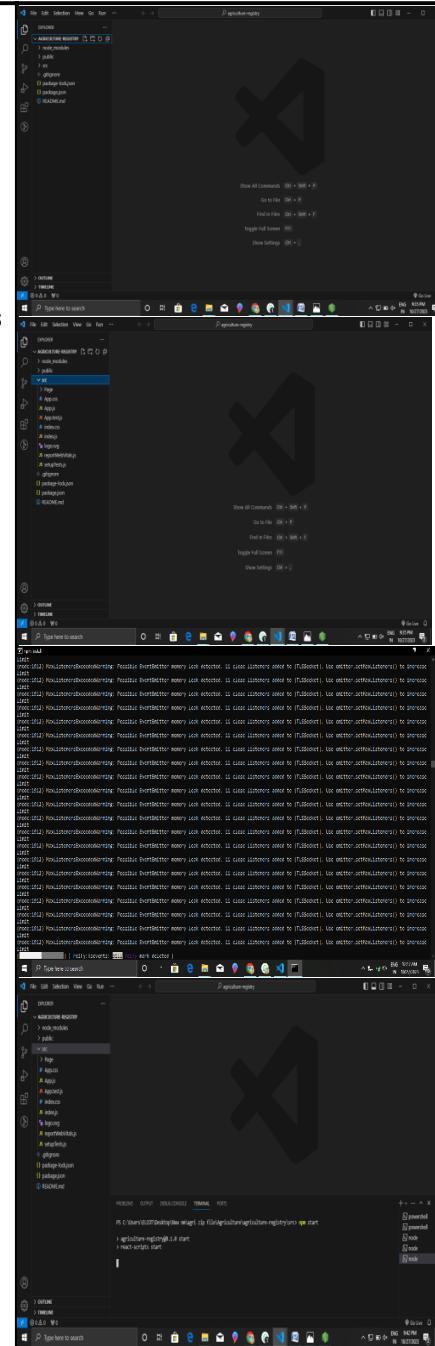
Open file explorer

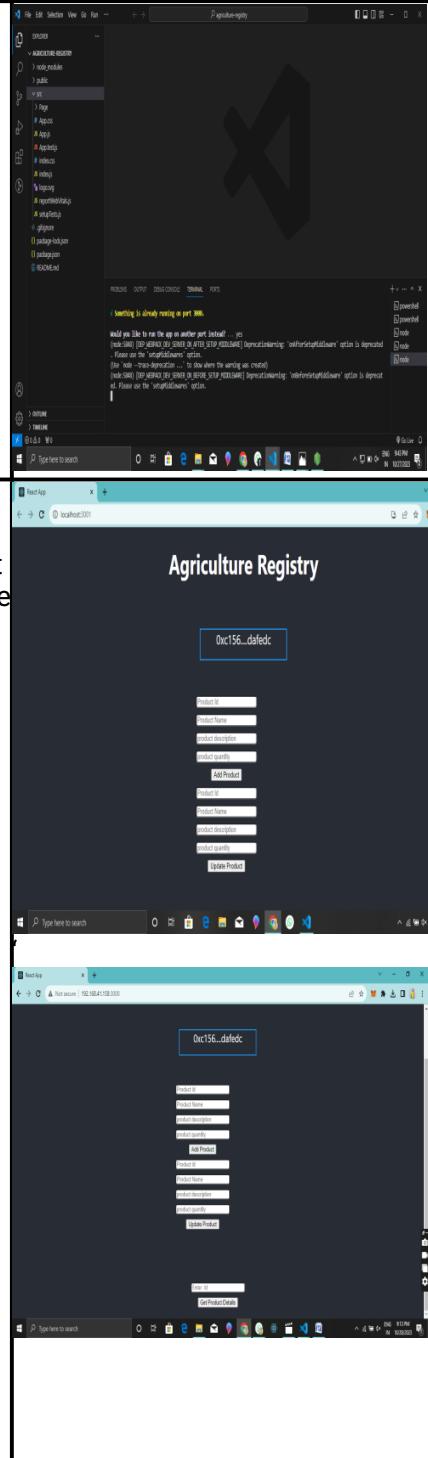
Open the extracted file and click on the folder.

Open src, and search for  
utiles.

Open cmd enter commands

- 1.npm install
  - 2.npm bootstrap
  - 3.npm start

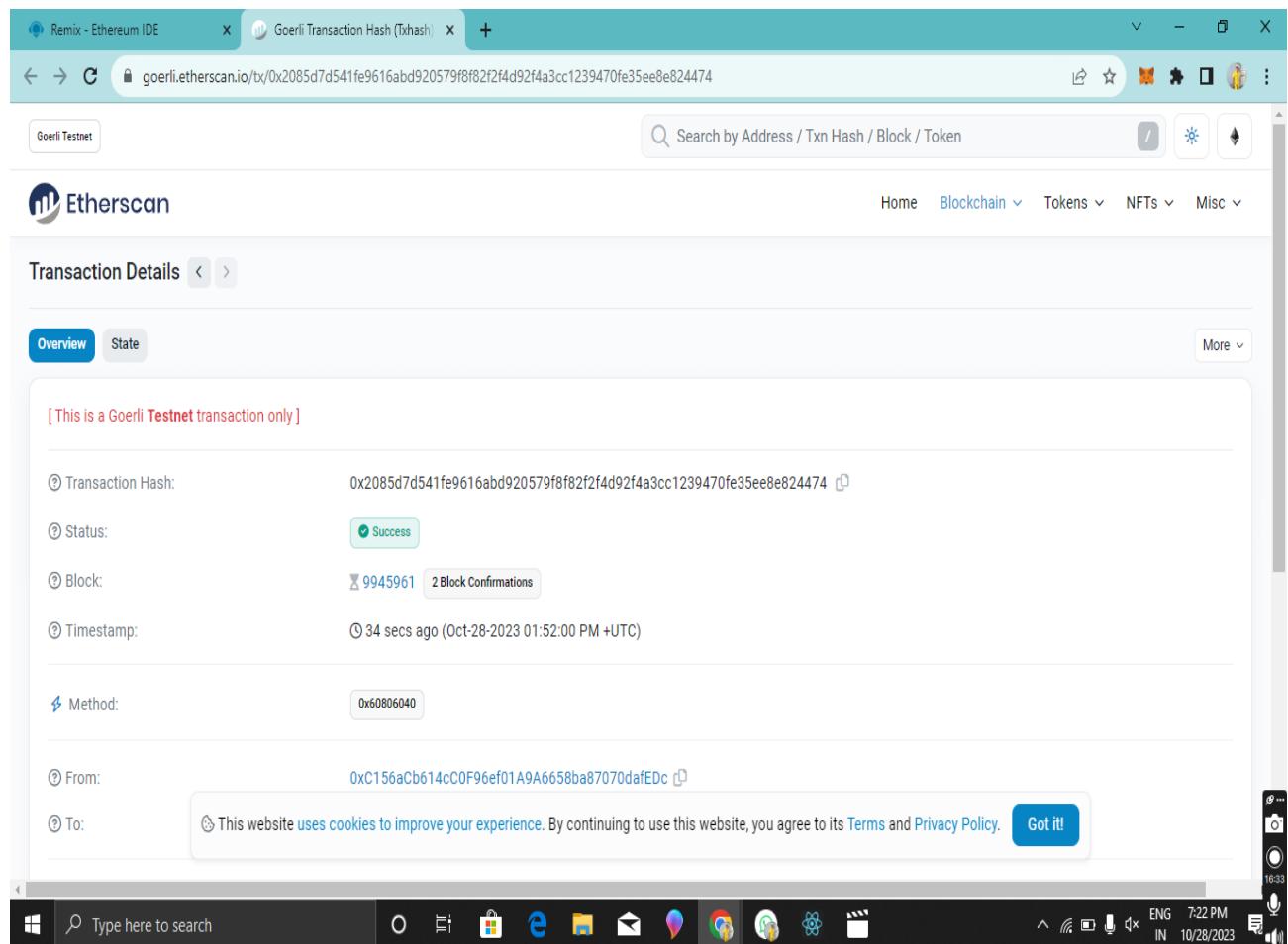


5.	{LOCALHOST IP ADDRESS}	<p>copy the address and open it to chrome so you can see the front end of your project.</p>  <p>The screenshot shows the Windows Start Menu with the 'Agriculture Registry' application pinned. Below the Start Menu, two browser windows are displayed. Both windows show the same user interface for managing products. The top window has the URL 'localhost:3001' in the address bar, and the bottom window has the URL '192.168.1.108:3001' in the address bar. The interface includes fields for Product ID, Product Name, Product Description, Product Quantity, and buttons for Add Product, Get Product, Get Product Details, and Update Product.</p>

## 9. RESULTS

### 9.1 OUTPUT SCREENSHOTS

(Smart contract creation output)



Remix - Ethereum IDE X Goerli Transaction Hash (Txhash) X +

goerli.etherscan.io/tx/0x2085d7d541fe9616abd920579f8f82f2f4d92f4a3cc1239470fe35ee8e824474

Goerli Testnet Search by Address / Txn Hash / Block / Token

This is a Goerli Testnet transaction hash.

Transaction Hash: 0x2085d7d541fe9616abd920579f8f82f2f4d92f4a3cc1239470fe35ee8e824474

Status: Success

Block: 9945961 2 Block Confirmations

Timestamp: 34 secs ago (Oct-28-2023 01:52:00 PM +UTC)

Method: 0x60806040

From: 0xC156aCb614cC0F96ef01A9A6658ba87070dafEDc

To: [ 0xfc62527f5e9d4ca3b0f2a1c808b1a172f158a557 Created ]

Value: 0 ETH (\$0.00)

Transaction Fee: 0.00192152500691749 ETH (\$0.00)

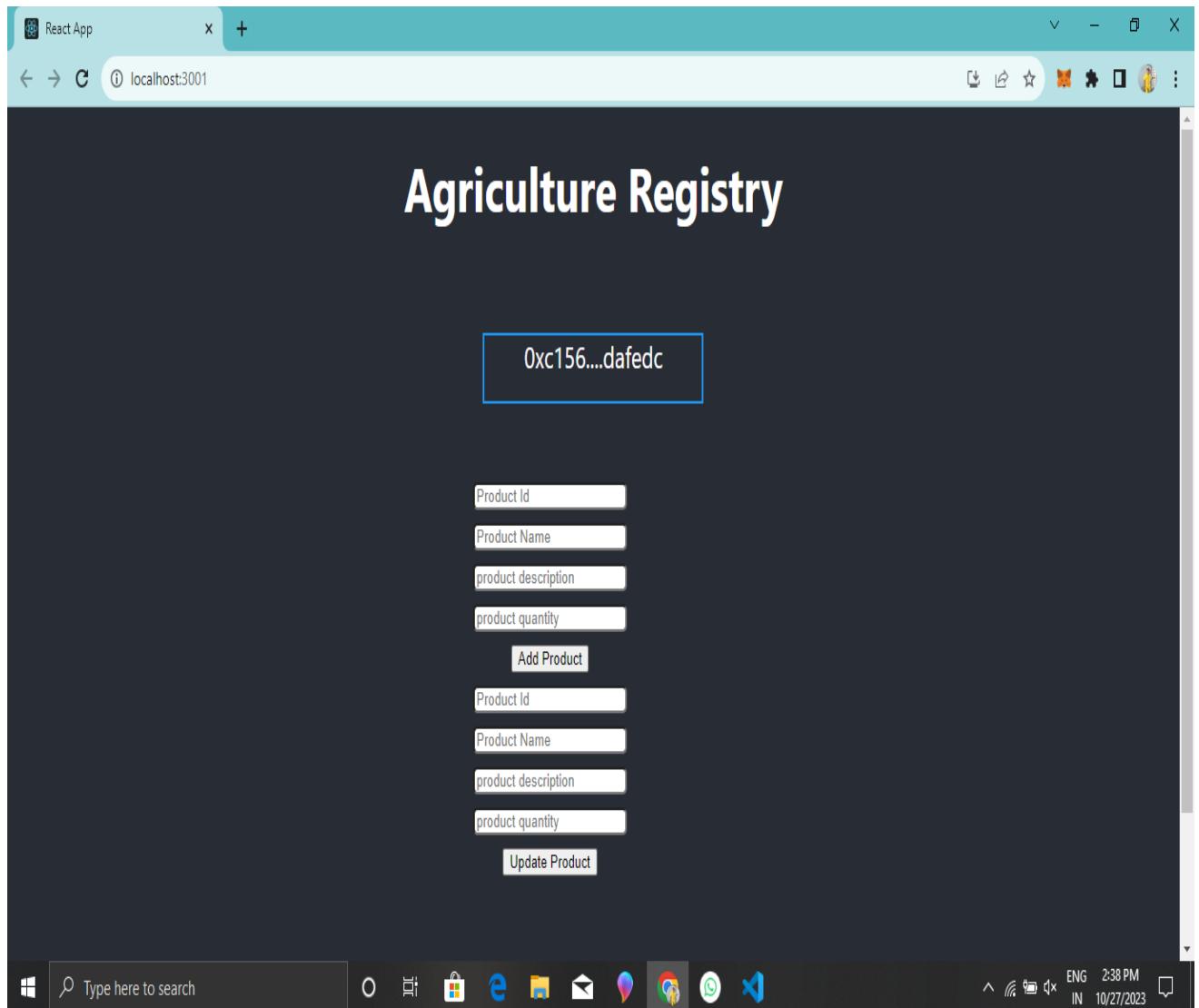
Gas Price: 2.50000009 Gwei (0.0000000250000009 ETH)

More Details: This website uses cookies to improve your experience. By continuing to use this website, you agree to its [Terms](#) and [Privacy Policy](#). Got it!

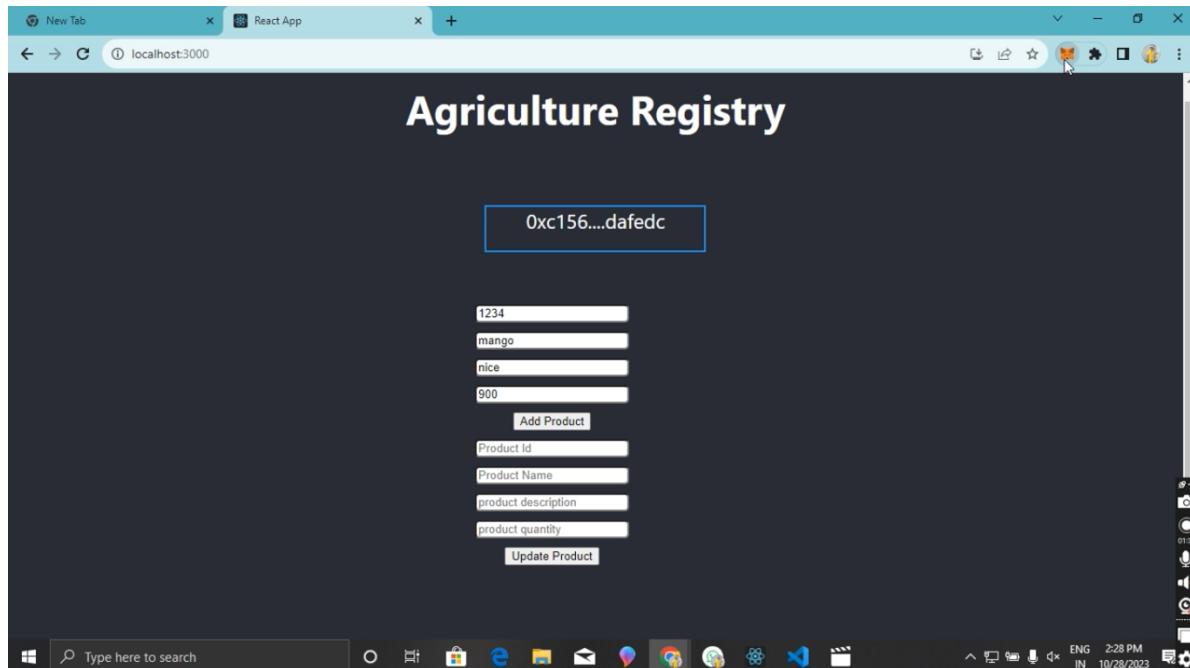
Type here to search

7:23 PM  
10/28/2023

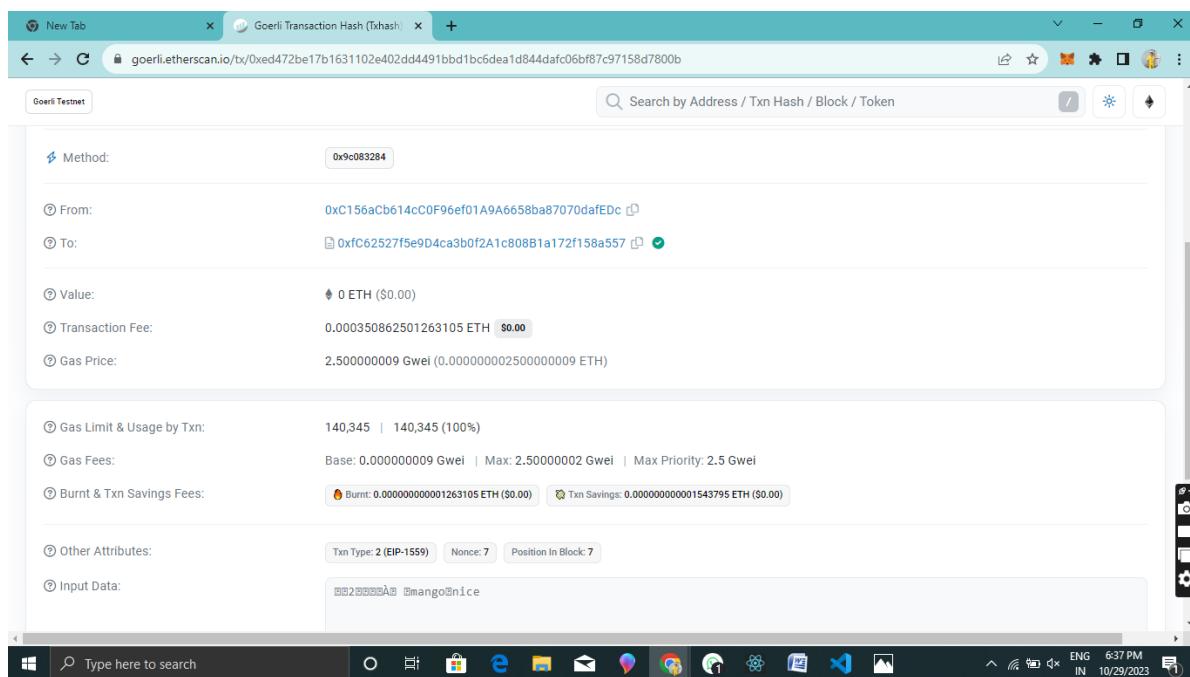
*(Opening of local host ip address in chrome)*



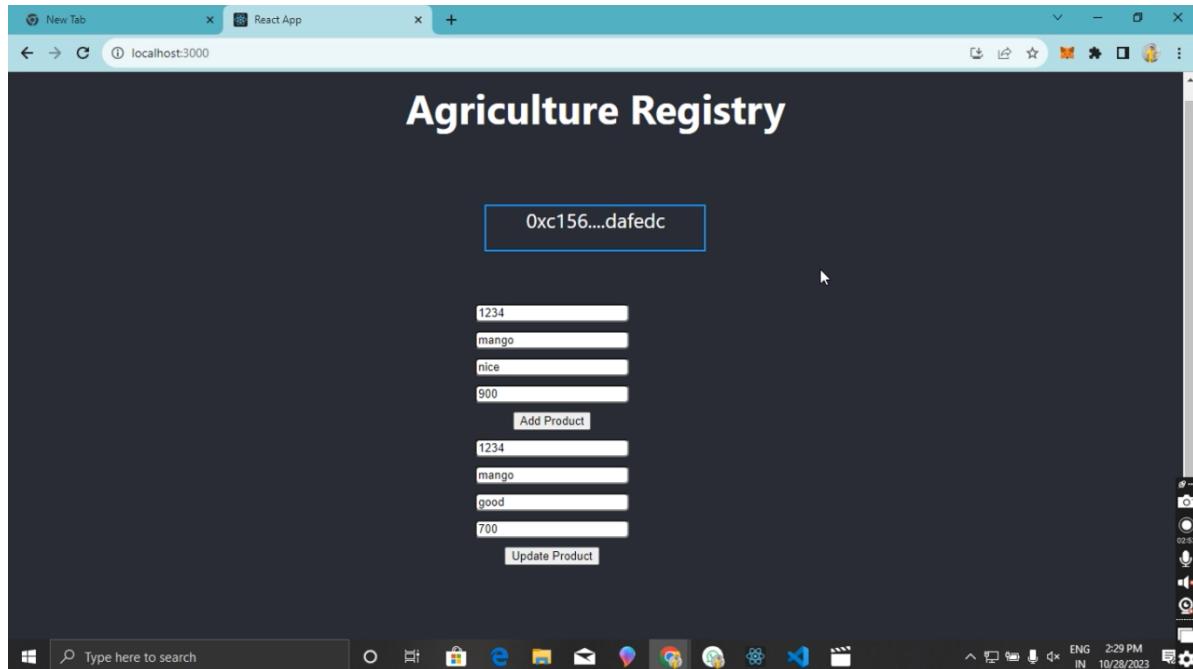
*(input data to add product )*



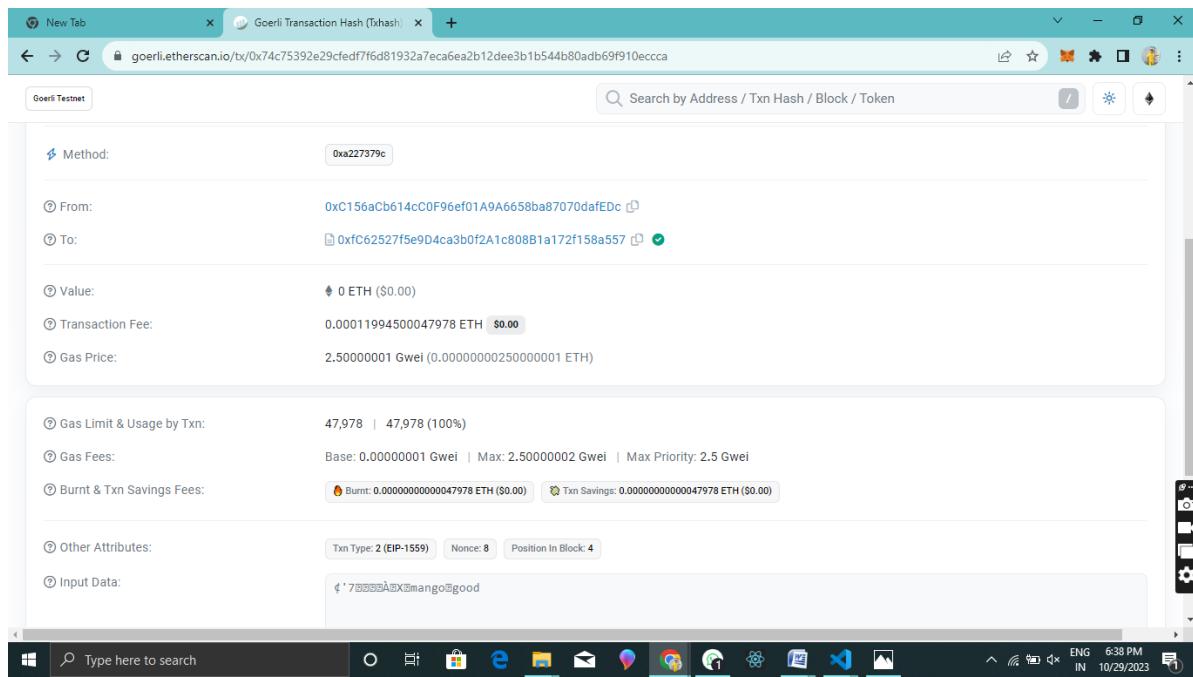
*(output )*



*(input data to update product)*



*(output )*



*(Transaction history)*

Transaction Hash	Method	Block	Age	From	To	Value	Txn Fee
0xdf68e0f38252decdf...	0xa227379c	9946040	22 hrs 47 mins ago	0xC156aCb614cC0F96ef01A9A6658ba87070dafEDc	0xfc6252...f158a557	0 ETH	0.00007632
0x36656be139622cde...	0x9c083284	9946026	22 hrs 50 mins ago	0xC156aCb614cC0F96ef01A9A6658ba87070dafEDc	0xfc6252...f158a557	0 ETH	0.00018493
0x74c75392e29cfedf7...	0xa227379c	9945985	23 hrs 2 mins ago	0xC156aCb614cC0F96ef01A9A6658ba87070dafEDc	0xfc6252...f158a557	0 ETH	0.00011994
0xed472be17b163110...	0x9c083284	9945978	23 hrs 4 mins ago	0xC156aCb614cC0F96ef01A9A6658ba87070dafEDc	0xfc6252...f158a557	0 ETH	0.00035086
0x2085d7d541fe9616a...	0x60806040	9945961	23 hrs 8 mins ago	0xC156aCb614cC0F96ef01A9A6658ba87070dafEDc	Contract Creation	0 ETH	0.00192152
0xd6ea6097b840982c...	0x60806040	9945938	23 hrs 14 mins ago	0xC156aCb614cC0F96ef01A9A6658ba87070dafEDc	Contract Creation	0 ETH	0.00192152
0x36d5a61e21ea2b9b...	Transfer*	9944788	1 day 4 hrs ago	0xC156aCb614cC0F96ef01A9A6658ba87070dafEDc	0xd9145C...43F39138	0 ETH	0.00003343
0xec7a254eb90a9945...	Transfer*	9944784	1 day 4 hrs ago	0xC156aCb614cC0F96ef01A9A6658ba87070dafEDc	0xd9145C...43F39138	0 ETH	0.00003343
0x86bddefc8540e31ff6...	Transfer*	9944748	1 day 4 hrs ago	0xC156aCb614cC0F96ef01A9A6658ba87070dafEDc	0xd9145C...43F39138	0 ETH	0.00003345
0xb2bc0546f73765e8c...	Transfer*	9944541	1 day 5 hrs ago	0xC156aCb614cC0F96ef01A9A6658ba87070dafEDc	0xd9145C...43F39138	0 ETH	0.00003343
0xb2198f1ae9c15516a...	Transfer*	9944532	1 day 5 hrs ago	0xC156aCb614cC0F96ef01A9A6658ba87070dafEDc	0xd9145C...43F39138	0 ETH	0.00003343

## 10. ADVANTAGES AND DISADVANTAGES

### **ADVANTAGES :**

*Incorporating blockchain technology into agricultural registries has the potential to revolutionize the sector by providing a more secure, efficient, and transparent way to manage and share critical agricultural data.*

- *Data Integrity*
- *Transparency*
- *Security among agricultural data*
- *Efficiency*
- *Reduced intermediaries*
- *Global Accessibility*

### **DIS ADVANTAGES :**

*While blockchain technology offers many advantages for agriculture registries, it's essential to consider the disadvantages and challenges:*

- *Complexity*
- *Scalability issues*
- *Energy Consumption*
- *Costs*
- *Regularity Challenges*
- *Human error*

## 11. CONCLUSION

*This Agriculture DocsChain project, utilizing blockchain technology, represents a significant step forward in addressing key challenges and opportunities within the agricultural sector. This innovative solution provides a secure, transparent, and decentralized platform for storing and managing agricultural data, with the ability to add, query, and update details.*

*This project facilitates easy data sharing and collaboration within the agricultural community. Farmers, researchers, and organizations can efficiently access, add, query, and update agricultural information, leading to more informed decision-making.*

*Blockchain technology provides a historical record of all data and transactions. This traceability is particularly valuable in tracking the origin and journey of agricultural products, which is essential for quality control and food safety. The registry can incorporate strong identity and access management systems, allowing users to control who can access their data. This feature protects sensitive information while still allowing for data sharing when necessary.*

## 12. FUTURE SCOPE

*The future scope of Agriculture DocsChain is not limited to data management but extends to reshaping the entire agricultural landscape by providing secure, transparent, and efficient data handling. As technology continues to evolve and as the agricultural sector faces new challenges, the platform has the potential to adapt and grow in ways that benefit all stakeholders in agriculture. The platform's ability to provide reliable and real-time data can revolutionize the way decisions are made in agriculture. Farming practices, crop planning, and supply chain management can become more data-driven, leading to increased productivity and sustainability.*

## **13. APPENDIX**

**Source Code :**

[https://drive.google.com/file/d/1J6ulm\\_XoqdAnD3\\_9D-gwgTVXrlIGrY3q/view?  
usp=sharing](https://drive.google.com/file/d/1J6ulm_XoqdAnD3_9D-gwgTVXrlIGrY3q/view?usp=sharing)

**GitHub Link :**

[https://github.com/NOORUL-A/Agriculture-Docs-Chain\\_NM2023TMID03811](https://github.com/NOORUL-A/Agriculture-Docs-Chain_NM2023TMID03811)

**Project Demo Link :**

[https://drive.google.com/folderview?id=1JCAaZBgxErdQUciujGtZr5WNA  
Gt7w7Ee](https://drive.google.com/folderview?id=1JCAaZBgxErdQUciujGtZr5WNA Gt7w7Ee)