

Project Report

Team ID	NM2023TMID03315
Project Name	Agriculture Docs Chain

Submitted by

TEAM LEADER : Judah Benilan.A

TEAM MEMBER 1 :Thirumalai

TEAM MEMBER 2 : Prasanth

TEAM MEMBER 3 : Aravind

PROJECT REPORT FORMAT

1. INTRODUCTION

1.1 Project Overview

1.2 Purpose

2. LITERATURE SURVEY

2.1 Existing problem

2.2 References

2.3 Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

3.2 Ideation & Brainstorming

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

4.2 Non-Functional requirements

5. PROJECT DESIGN

5.1 Data Flow Diagrams & User Stories

5.2 Solution Architecture

6. PROJECT PLANNING & SCHEDULING

6.1 Technical Architecture

6.2 Sprint Planning & Estimation

6.3 Sprint Delivery Schedule

7. CODING & SOLUTIONING

7.1 Feature 1

7.2 Feature 2

7.3 Database Schema (if Applicable)

8. PERFORMANCE TESTING

8.1 Performance Metrics

9. RESULTS

9.1 Output Screenshots

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

13.1 Source Code

13.2 GitHub & Project Demo Link

1. INTRODUCTION

Designing a smart contract on the Ethereum blockchain to store and manage agriculture data, including the ability to add, query, and update details, requires careful planning. Below is a conceptual outline of how this can be done, along with a simplified smart contract. This simplified smart contract allows for adding and updating agriculture data. In practice, additional features, security measures, and access control mechanisms would be needed for a fully functional and secure agriculture data management system on the blockchain.

1.1 Project overview :

In the agriculture industry, there is a significant challenge related to the management, verification, and accessibility of critical documents and data. Farmers, agricultural businesses, regulatory authorities, and other stakeholders deal with a vast array of documents, including land deeds, crop records, certifications, permits, contracts, and more. These documents are often paper-based, prone to loss or damage, and can be challenging to verify, leading

- Agriculture-related documents are typically managed in a fragmented and paper-based manner, making it difficult to organize, retrieve, and maintain their integrity.
- Ensuring the authenticity and validity of agricultural documents, such as land titles or organic certifications, is a critical concern.
- Stakeholders often require timely access to critical agricultural documents for decision-making, regulatory compliance, and transactions.

1.2 Purpose

As of 1 June 2012 new regulations on the traceability of medicines in Argentina require a specific identification to be placed on a large group of pharmaceutical products. This new regulation implements effective countermeasures against counterfeit and substandard drugs. Traceability is the ability to track specified stages of the supply chain and trace backwards the history, application or location of the pharmaceutical products that are under consideration. The global standardised identification system from manufacturer to patient provides a safer healthcare supply chain.

2.LITERATURE SURVEY

1.Blockchain for Agriculture docs chain

Agriculture is a big part of the economy of any country because it helps feed the entire population. It connects and communicates with all of the related industries. If the agriculture base is strong, it is generally regarded as a socially and politically stable society. Many modern farms make use of cutting-edge technology and scientific and technological ideas

2.A Blockchain-Based Approach for Agriculture docs chain in Healthcare Supply Chain

The following are some of the reasons for food supply chain problems and processing environment challenges.

The maximization of the profits relies on some farmers' vegetables and fruits with chemicals. Chemical fertilizers, insecticides, and other compounds are used in several plants and fruits.

As a result, pesticide residues in vegetables and fruits become excessive. It is a significant health risk.

3. A semantic blockchain-based system for Agriculture docs chain

Khizar Abbas, Muhammad Afaq, Talha Ahmed Khan

Agriculture doc chain is currently a very challenging area given the complexity of several issues, including drug quality and counterfeit medications. The counterfeited drugs have a major impact on human life, treatment outcomes and economic burden. To deal with these issues, we propose a semantic blockchain-based system for drug traceability that aims at detecting counterfeit safety and quality of life as well as eliminating manufacturers' potential loss and increasing their revenue. Our proposal is based on blockchain and semantic web technologies to enhance the representation capability of data in the pharmaceutical supply chain.

2.1 Existing problem

Pharmaceutical serialization is a regulatory compliance that assures a unique identifier assigned to every unit of prescribed medicine. This unique identifier is used for product tracking and authentication in the supply chain. Initially, in 2018, the AGRICULTURE DOC Chain Security Act (DSCSA) implemented pharmaceutical drug serialization regulations to mitigate the risk of counterfeit medicine entering the US market. Under this regulation, the pharmaceutical drug manufacturer is required to print a 2D data matrix barcode encoded with unique identification on each drug unit. Basically, printing a unique identification code on each prescribed drug for authenticity and traceability is not sufficient to eliminate the risk of drug counterfeiting. Subsequently, criminals and drug counterfeiters can still supply illicit or stolen drugs into the supply chain through an illegal source or online trade by imitating the same information in multiple units. After this regulation's enforcement, we observed that the US market lacks a mechanism to authenticate individual drug units with a centralized, secure repository before dispensing them to patients. Since the COVID-19 pandemic, drug counterfeiters and criminals have produced large quantities of contaminated drugs, which they then distribute through their illicit networks and underground social media platforms.

2.2 References

1. G. S. Sajja, K. P. Rane, K. Phasinam, T. Kassanuk, E. Okoronkwo, and P. Prabhue, “Towards applicability of Blockchain in Agriculture sector,” *Materials Today Proceedings*, vol. 5, 2021.

View at: [Publisher Site](#) | [Google Scholar](#)

2. J. Lin, Z. Shen, A. Zhang, and Y. Chai, “Blockchain and IoT Based Food Traceability for Smart Agriculture,” in *Proceedings of the 3rd International Conference on Crowd Science and Engineering*, Singapore, 2018.

View at: [Google Scholar](#)

3. W. Lin, X. Huang, V. Wang et al., “Blockchain technology in current agricultural systems: from techniques to applications,” *IEEE Access*, vol. 8, pp. 143920–143937, 2020.

View at: [Publisher Site](#) | [Google Scholar](#)

4. H. Xiong, T. Dalhaus, P. Wang, and J. Huang, “Blockchain technology for agriculture: applications and rationale,” *IEEE Access*, vol. 3, pp. 1–7, 2020.

View at: [Publisher Site](#) | [Google Scholar](#)

5. P. Dutta, T. M. Choi, S. Somani, and R. Butala, “Blockchain technology in supply chain operations: Applications, challenges and research opportunities,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 142, Article ID 102067, 2020.

View at: [Google Scholar](#)

6. G. Leduc, S. Kubler, and J. P. Georges, “Innovative Blockchain-based farming marketplace and Smart contract performance evaluation,” *Journal of Cleaner Production*, vol. 306, pp. 1–15, 2021.

View at: [Publisher Site](#) | [Google Scholar](#)

7. P. Bottino, N. Gessa, G. Massa, R. Pareschi, H. Selim, and E. Arcuri, “Intelligent smart contracts for innovative supply chain management,” *Frontiers in Blockchain*, vol. 3, no. 52, 2020.

View at: [Publisher Site](#) | [Google Scholar](#)

8. K. Dey and U. Shekhawat, “Blockchain for sustainable e-agriculture: literature review,” *Architecture for data management, and implications, Journal of Cleaner Production*, vol. 316, pp. 1–17, 2021.

View at: [Publisher Site](#) | [Google Scholar](#)

9. L. Hang, I. Ullah, and D. Kim, “A secure fish farm platform based on Blockchain for agriculture data integrity,” *Computers and Electronics in Agriculture*, vol. 170, pp. 1–15, 2020.

View at: [Publisher Site](#) | [Google Scholar](#)

10. T. H. Pranto, A. A. Noman, A. Mahmud, and A. K. M. Haque, “Blockchain and Smart contract for IoT enabled smart agriculture,” 2021, <https://arxiv.org/abs/2104.00632>.
View at: [Google Scholar](#)
11. X. Huang, D. Ye, and L. Shu, “Securing parked vehicle assisted fog computing with Blockchain and optimal Smart contract design,” *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 2, pp. 426–441, 2020.
View at: [Publisher Site](#) | [Google Scholar](#)
12. L. Wang, L. Xu, S. Liu et al., “Smart contract-based agricultural food supply chain traceability,” *IEEE Access*, vol. 9, pp. 9296–9307, 2021.
View at: [Publisher Site](#) | [Google Scholar](#)
13. K. Salah, N. Nizamuddin, and M. Omar, “Blockchain-based soybean traceability in agricultural supply chain,” *IEEE Access*, vol. 7, pp. 73295–73305, 2019.
View at: [Publisher Site](#) | [Google Scholar](#)
14. I. Widi Widayat and M. Köppen, *Blockchain Simulation Environment on Multi-Image Encryption for Smart Farming Application*, Springer International Publishing, Berlin, Germany, 2021.
15. C. M. Balaceanu, I. Marcu, and G. Suciu, “Telemetry System for Smart Agriculture,” *Business Information Systems Workshops*, Springer, Berlin, germany, 2019.
View at: [Google Scholar](#)
16. H. Patel and B. Shrimali, “AgriOnBlock: Secured data harvesting for agriculture sector using Blockchain technology,” *ICT Express*, vol. 10, 2021.
View at: [Google Scholar](#)
17. S. Hu, S. Huang, J. Huang, and J. Su, “Blockchain and edge computing technology enabling organic agricultural supply chain: a framework solution to trust crisis,” *Computers & Industrial Engineering*, vol. 153, pp. 1–12, 2021.
View at: [Publisher Site](#) | [Google Scholar](#)
18. H. N. Dai, Z. Zheng, and Y. Zhang, “Blockchain for internet of Things: a survey,” *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8076–8094, 2019.
View at: [Publisher Site](#) | [Google Scholar](#)
19. M. A. Shu and L. Shu, “The performance evaluation of blockchain-based security and privacy systems for the internet of Things: a tutorial,” *IEEE Internet of Things Journal*, vol. 8, no. 24, pp. 17236–17260, 2021.
View at: [Publisher Site](#) | [Google Scholar](#)

20. K. Peng, M. Li, C. Wang, S. Wan, and K.-K. R. Choo, “Security challenges and Opportunities for smart contracts in internet of Things: a survey,” *IEEE Internet of Things Journal*, vol. 8, no. 15, pp. 12004–12020, 2021.
View at: [Publisher Site](#) | [Google Scholar](#)
21. L. Hang, B. Kim, K. Kim, and D. Kim, “A permissioned blockchain-based clinical trial service platform to improve trial data transparency,” *BioMed Research International*, vol. 2021, no. 22, 2021.
View at: [Publisher Site](#) | [Google Scholar](#)
22. L. Hang, B. Kim, and D. Kim, “A transaction traffic control approach based on fuzzy logic to improve hyperledger fabric performance,” *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 2032165, 19 pages, 2022.
View at: [Publisher Site](#) | [Google Scholar](#)
23. R. Kumar, P. Kumar, A. Aljuhani, A. K. M. Najmul Islam, A. Jolfaei, and S. Garg, “Deep learning and smart contract-assisted secure data sharing for IoT-based intelligent agriculture,” *IEEE Intelligent Systems*, vol. 99, 2022.
View at: [Publisher Site](#) | [Google Scholar](#)
24. J. DafniRose and J. DafniRose, “Blockchain-enabled smart agricultural knowledge discovery system using edge computing,” *Procedia Computer Science*, vol. 202, pp. 73–82, 2022.
View at: [Publisher Site](#) | [Google Scholar](#)
25. A. E. Mane, M. Chihab, O. Bencharef, and C. Younes, “Architectural scheme of a multi-Blockchain in the Agricultural field,” *E3S Web of Conferences*, vol. 297, no. 4, 2021.
View at: [Google Scholar](#)

2.3 Problem Statement Definition

1. LIMITED INFRASTRUCTURE AND PRODUCTION CAPABILITIES:

Major Pharmaceutical companies does not invest and establish production units in developing countries due to geopolitics, market inaccessibility and government instabilities. They are more focused on manufacturing and circulation of branded medicines in developed countries like USA and Europe due to per capita income and pricing monopoly. Developing countries faces major challenges on pharmaceutical industries investment due to poor infrastructure and lack of government funds for research and infrastructure improvements. Existing pharmaceutical units are struggling to meeting global standards due to need of heavy investments in new production and packaging machineries.

2. AMBIGUOUS REGULATIONS

Regulatory obligation plays a vital role to implement track and trace system for traceability.

Developed countries like US in 2018 and Europe in 2019 implemented serialization compliance successfully for drug traceability. Before making digital traceability an obligatory regulation, DSCSA a drug controlling body of FDA runs pilot programs with joint initiative of drug manufacturers, wholesale distributors and community pharmacists.

3. INSUFFICIENT TECHNO-FUNCTIONAL RESOURCES FOR IMPLEMENTING AND SUSTAINING DRUG TRACEABILITY

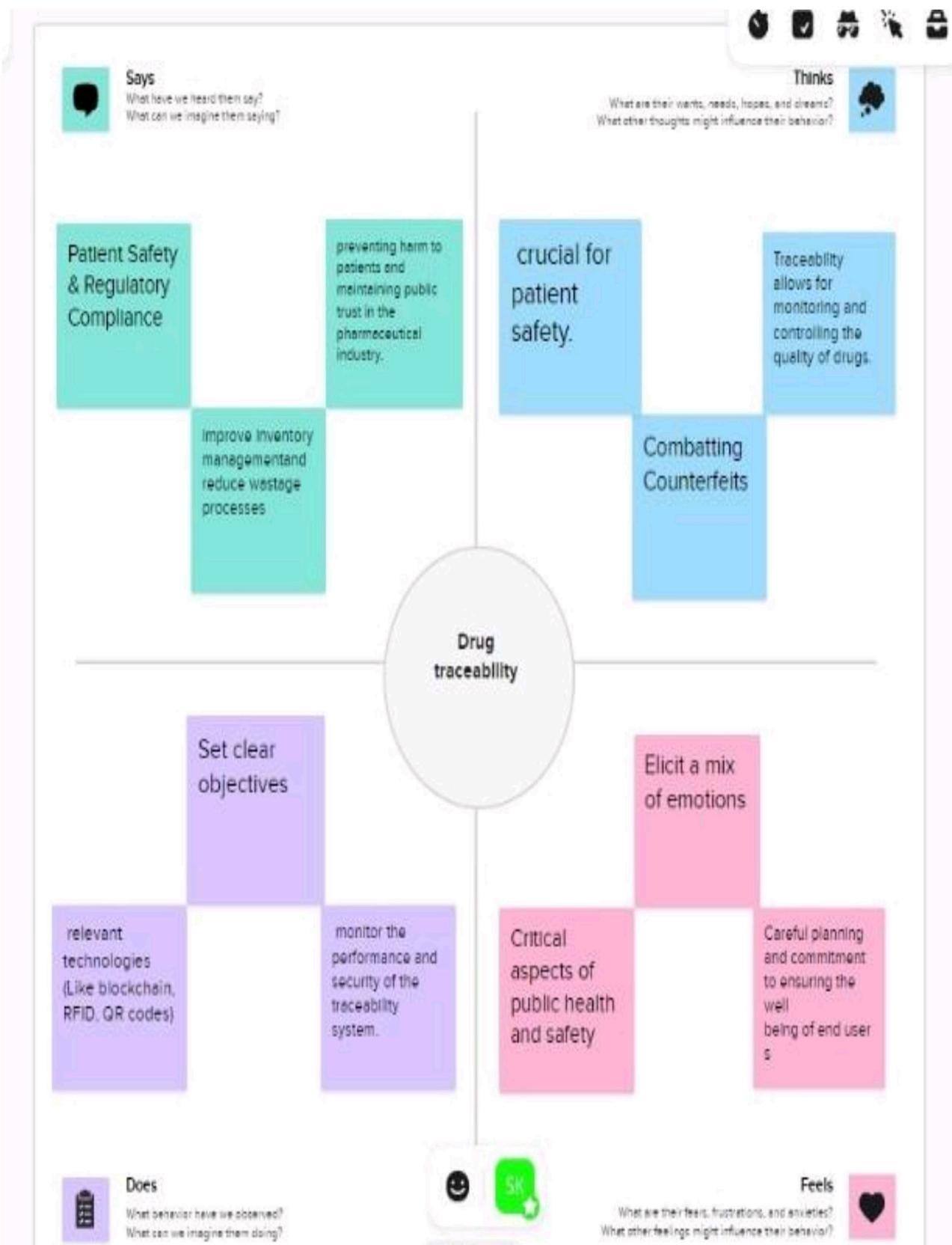
Drug counterfeiting a serious threat to public health. It is a collective job for every stakeholder in supply chain to prevent counterfeiting and illicit drugs. Implementing and sustaining serialization system for drug traceability required skillful resources. Serialization and drug traceability processes involve packaging lines, Barcode readers, scanner, label grading system, site and global level serialization system which can handle drug traceability globally. Any human, mechanical and technical error can cause adversely to human life.

4. UNSECURE AND UNRELIABLE TECHNICAL INFRASTRUCTURE FOR DIGITAL DRUG TRACEABILITY

Technology advancement is another main challenge in developing countries for authenticating and tracing pharmaceutical products digitally. In recent years, some developed countries like US and Europe have adopted serialization regulation under which drug manufacturer require to print a unique identifier printed with the 2D barcode on individual drug unit. This unique identifier is key source for authenticating drug and tracing its origin of manufacturing. Printing unique identifier and keeping its key data in repository requires special packaging equipment, tamper proof seals and global traceability software.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

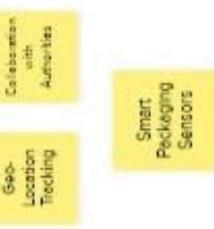
Hari Prasanth M



Santhoshkumar r



Sathishkumar A



Yogenanthem K



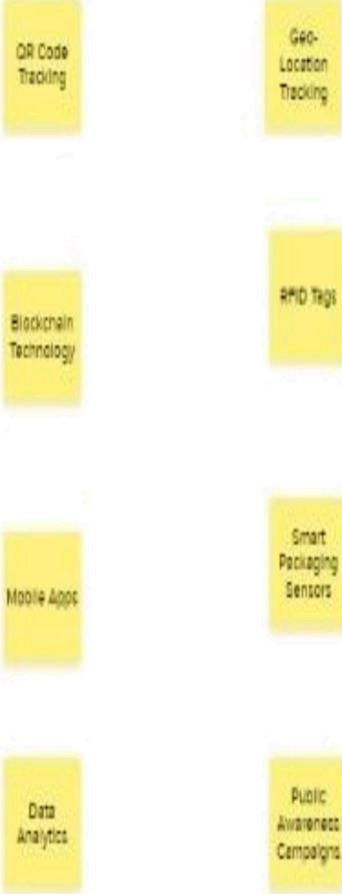
Group ideas

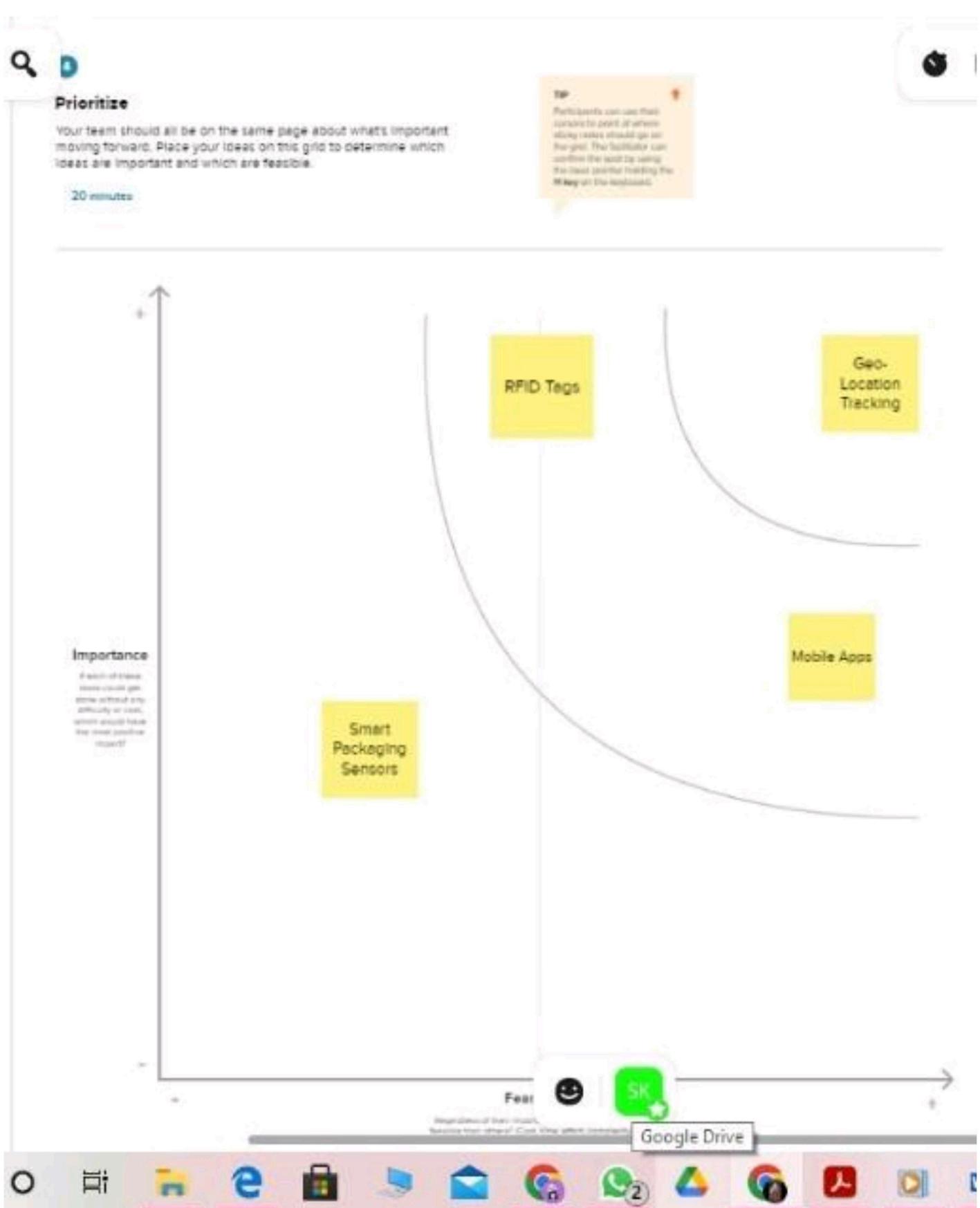
Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you can break it up into smaller sub-groups.

20 minutes

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.





4.REQUIREMENT ANALYSIS

4.1 Functional requirement

Ref	IoT app	IoT devices	Conse nsus algo.	Securi ty analys is techni ques	Ene rgy and data size tests	Platfor m	Environ ment control smart contract s	Trace and distrib ution smart contrac t	Finan cing smart contra ct	Other smart contracts and notes
[12]	Yes	Temperat ure sensor, water level sensor, oxygen sensor, pH sensor, and water pump	PBFT	Not mentioned	Yes	Hyperle dger Fabric	No	Basic	No	Records history of all transactions and device management
[13]	Yes	Temperat ure sensor, light sensor, and humidity sensor	Not mention ed	Not mention ed	Yes	Ethereu m	Yes	Yes	Basic	Storage smart contract s
[14]	No	No IoT app mentione d	PoW	Not mention ed	Yes	Ethereu m	No	Yes	Yes	No other smart contract s
[15]	No	No IoT app mentione d	PoW	Yes	Yes	Ethereu m	No	No	No	The agent orchestrates offloading services

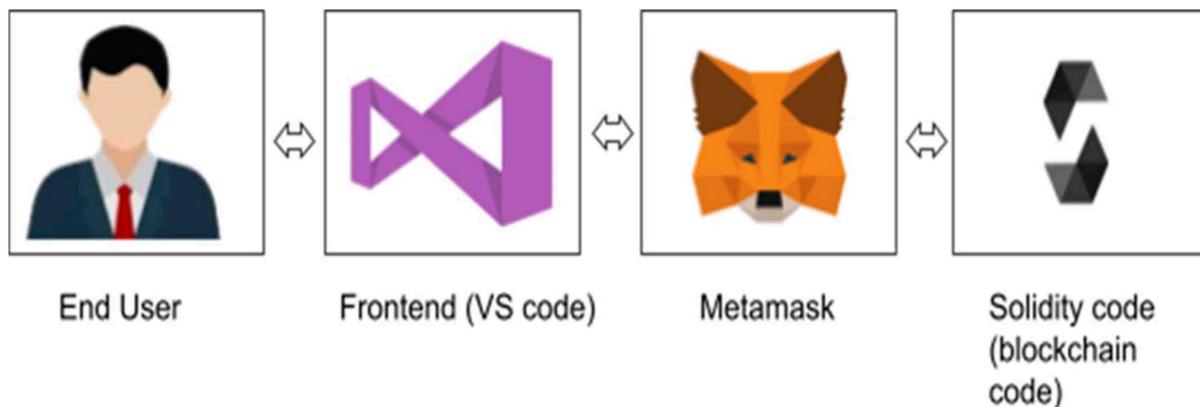
[16]	Yes	Sensors that manage water, air, sunlight, and soil quality escarpments	Not mentioned	Yes	Basic	Hyperledger Fabric	No	Yes	Yes	Storage of planting and farm information
[17]	Yes	Sensors, cameras, GPS locator, and 4G communication	Not mentioned	Not mentioned	Basic	Ethereum	Yes	Yes	Yes	Use case of soybean supply chains
[20]	Not mentioned	No equipment mentioned	Not mentioned	Yes	No	Ethereum	No	Yes	Yes	Insurance claim
[21]	Yes	Their consensus Cameras, smart sensors, RFID, and GPS mechanism (OASCs' consensus network)	(OASCs' consensus network)	Yes	Yes	It looks like an Ethereum sys.	No	Yes	Basic	Automated workflows of OASCs
Our survey	Yes	Temperature sensor, humidity sensor, light exposure sensor, and soil pH sensor	PoW (in future PoS)	Yes	Yes	Ethereum (in future Ethereum 2.0)	Yes	Yes	Basic	No other smart contracts

4.2 Non-Functional requirements

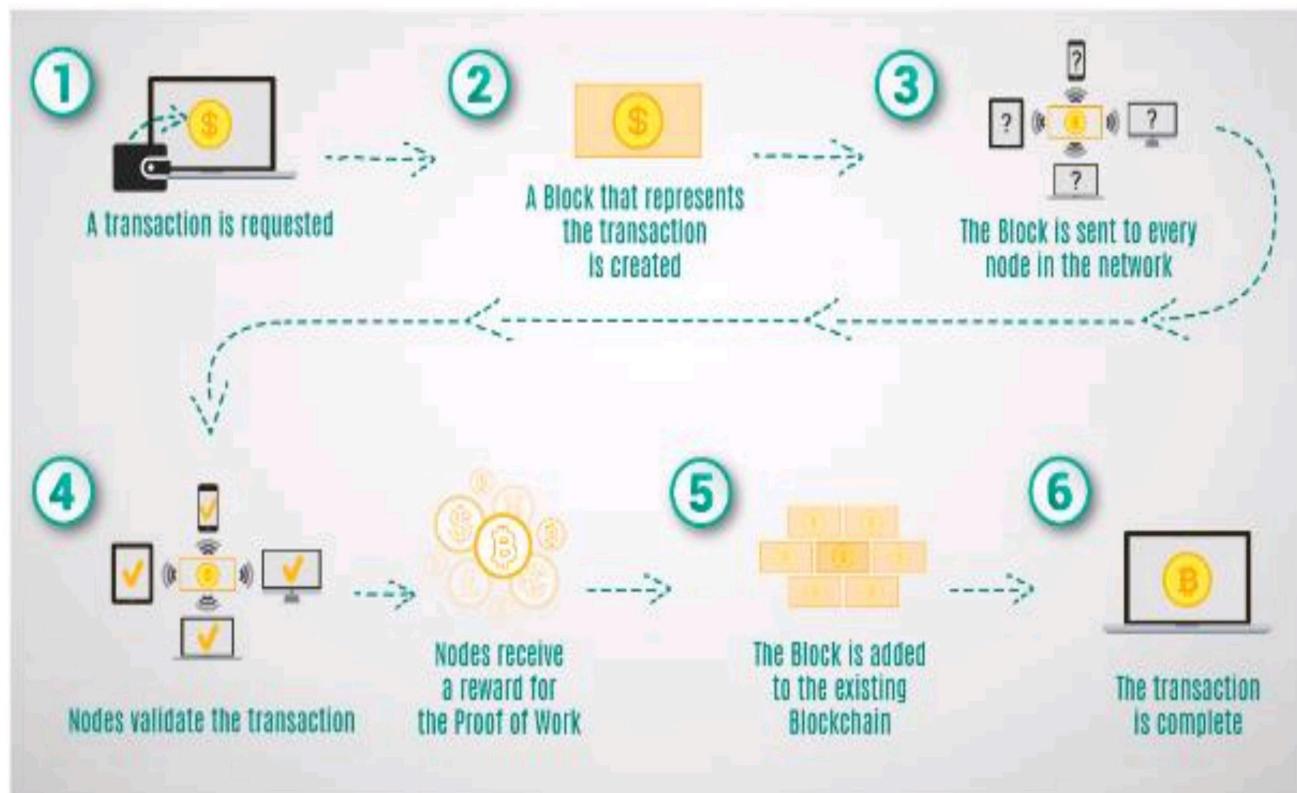
NFR No.	Non-Functional Requirement	Description
NFR-1	Usability	For drug traceability, we've proposed a totally new blockchain system. This solution is more secure and scalable than other options currently available. Furthermore, the suggested system can prune its storage effectively, resulting in a robust and usable blockchain storage solution.
NFR-2	Security	Traditional solutions to achieve traceability within pharmaceutical supply chain are typically centralized and lack transparency across participants of the supply chain, which allows the central authority to modify information without notifying other stakeholders.
NFR-3	Reliability	Blockchain-based drug traceability offers a potential solution to create a distributed shared data platform for an immutable, trustworthy, accountable and transparent system in the PSC.
NFR-4	Performance	Scalability is essential, as the system must maintain usability as the volume of drug and users grows. Consistent response times are paramount, and well-defined benchmarks must be met to uphold usability standards. Performance testing and regular system optimization are essential to ensure that the Drug traceability operates smoothly, regardless of the scale or usage patterns.
NFR-5	Availability	Blockchain technology ensures an efficient and cost-effective solution that underpins different drug traceability functions and procedures to ascertain proper identification, tracing, tracking, and provenance.
NFR-6	Scalability	Blockchain technology enables creating a private permissioned network to trace and track events in the pharmaceutical supply chain and provides time stamped records of each transaction performed. Examples of events include, execution and owner, time, location of transaction, and which stakeholders were involved.

5. PROJECT DESIGN

5.1 Data Flow Diagrams & User Stories



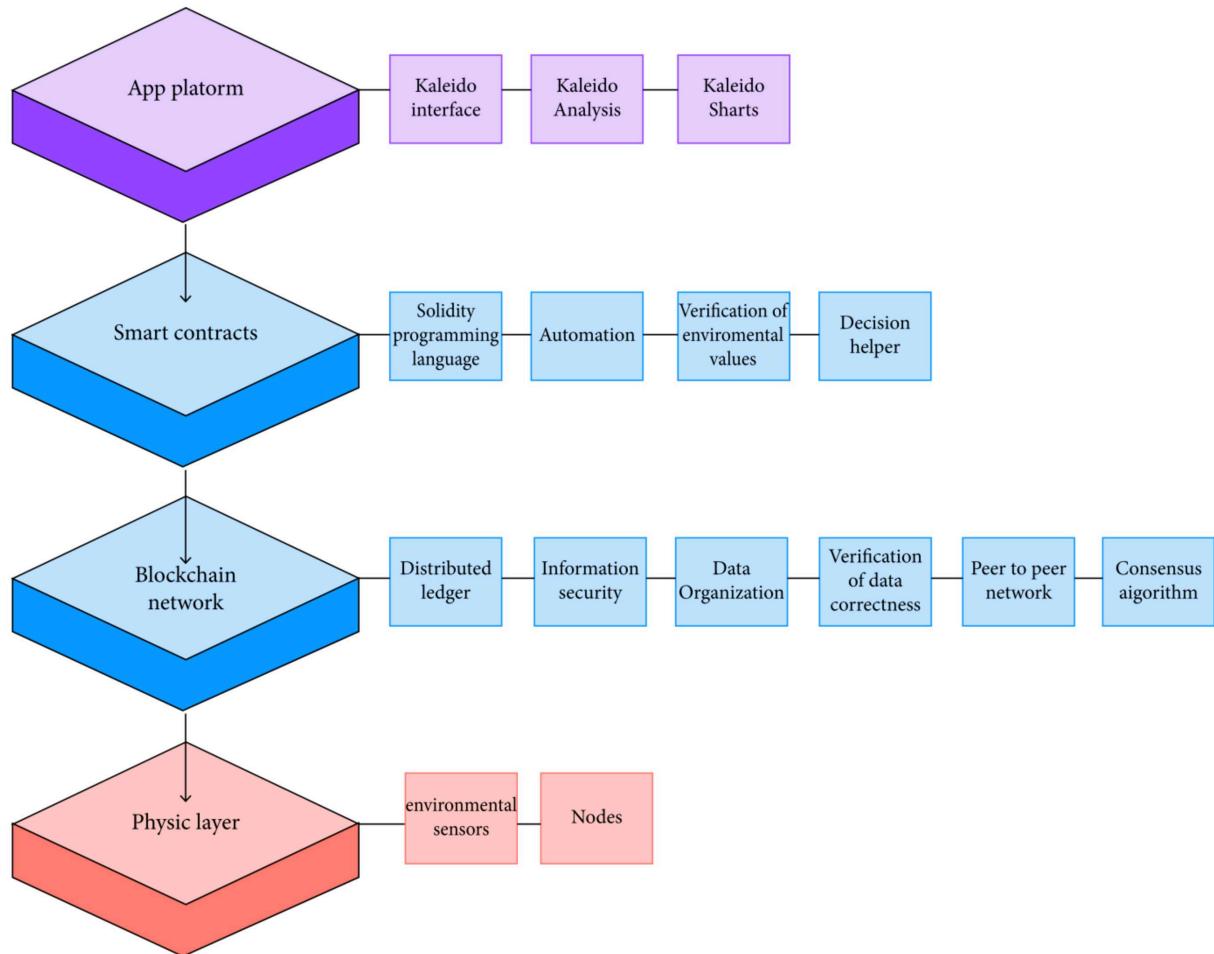
Data Flow Diagrams



User Stories

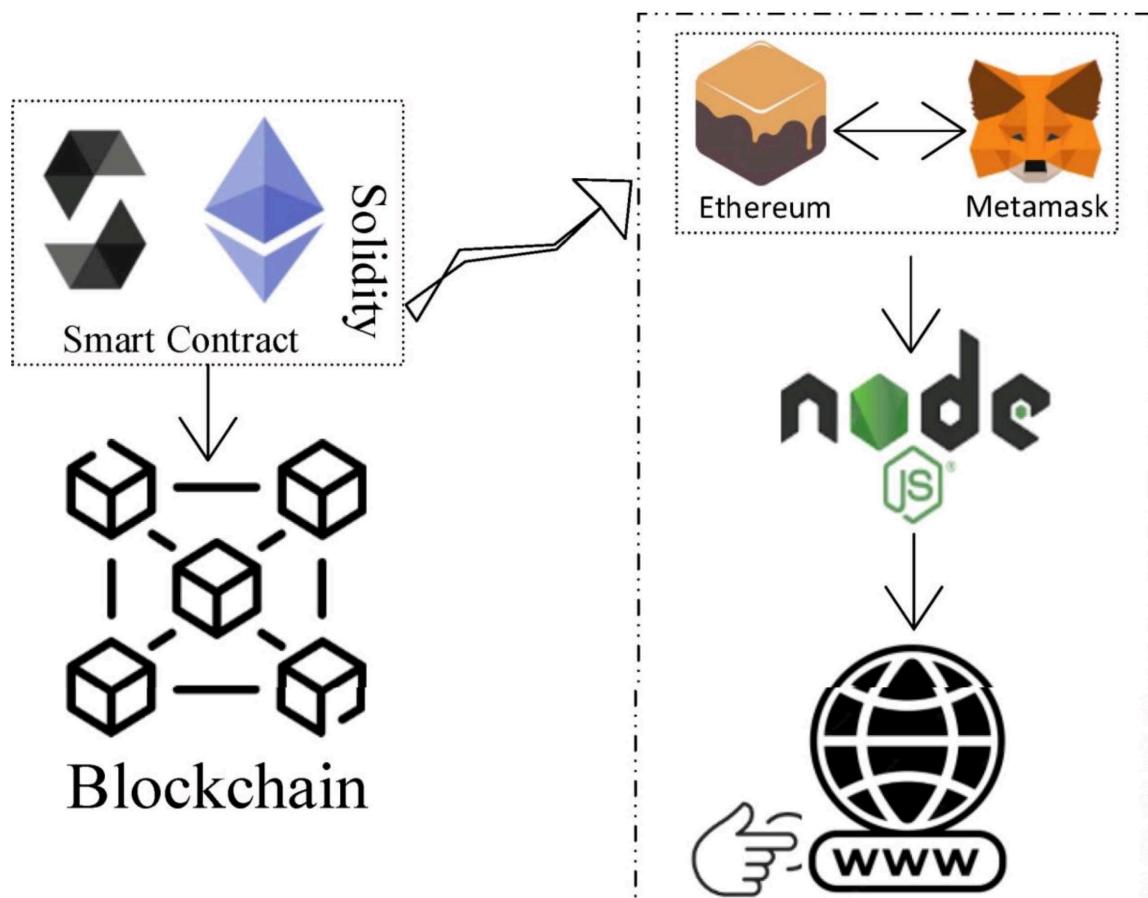
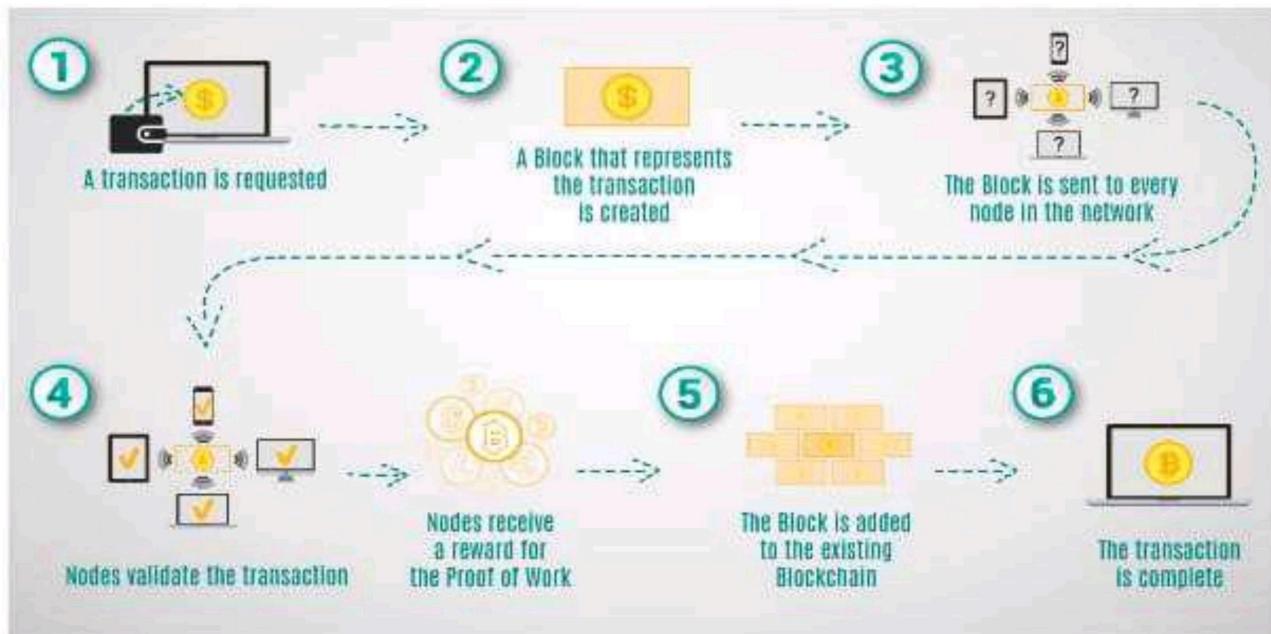
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	team Member
Content Creator	Agriculture Upload and Management	USN-1	As a content creator, I want to upload multiple image and video assets to the drug tracking with drag-and-drop functionality	Users should be able to drag and drop multiple assets onto the drug traceability interface, and the system should process and upload them efficiently	High	Judah benilan
Content Creator	Agriculture Upload and Management	USN-2	As a content creator, I need to add detailed metadata to my drug, including titles, descriptions, and copyright information, to keep them wellorganized	Metadata fields should be easily accessible and editable, and changes should be immediately reflected in drug information	Medium	Thirumalai
Marketing Manager	Agriculture Organization and Access Control	USN-3	As a marketing manager, I want to create and assign tags to drug for easy categorization, facilitating efficient drug retrieval.	Tags should be customizable, and drug should be sortable and filterable by assigned tags	Medium	Prasanth r
Marketing Manager	Agriculture Organization and Access Control	USN-4	As a marketing manager, I need to restrict access to confidential drug to authorized team members only	Access control settings should allow me to specify who can view, edit, and delete drug , with permissions easily adjustable	High	Aravind
Administrator	System Management	USN-5	As an administrator, I want to monitor and manage drug access, user roles, and system performance.	The admin dashboard should provide insights into user activity, allow role assignments, and offer system performance metrics	High	Judah benilan
Administrator	System Management	USN-6	As an administrator, I need to set up automated data backups and a disaster recovery plan for data safety.discipline	The system should regularly back up data and provide a documented recovery plan to prevent data loss.	High	Thirunalai

5.2 Solution Architecture



6.PROJECT PLANNING & SCHEDULING

6.1 Technical Architecture



6.2 Sprint Planning & Estimation

1. User Story Backlog:

Start by creating a backlog of user stories. These stories should represent the features, enhancements, or tasks needed for your drug traceability. Ensure they are well-defined, with clear acceptance criteria.

2. Prioritization:

Collaborate with stakeholders to prioritize the user stories based on their importance and impact. High-priority stories should be at the top of the backlog.

3. Sprint Planning Meeting:

Hold a sprint planning meeting with your development team. During this meeting, select a set of user stories from the backlog to work on during the upcoming sprint. Consider the team's capacity and the complexity of the stories.

4. Story Point Estimation:

Use a method like story point estimation to estimate the effort required for each user story. The team assigns relative points to stories to indicate their complexity. This helps in determining how many stories can be included in the sprint.

5. Sprint Goal:

Define a clear sprint goal, which should align with the project's objectives. The goal should provide a sense of purpose for the sprint.

6. Daily Stand-Ups:

Conduct daily stand-up meetings to keep the team updated on progress, discuss any challenges, and make necessary adjustments to the sprint plan.

7. Sprint Review:

At the end of the sprint, hold a sprint review meeting to showcase the completed work to stakeholders. Gather their feedback and insights.

8. Sprint Retrospective:

After the review, conduct a sprint retrospective to assess what went well and what could be improved. Use this feedback to make process enhancements for the next sprint.

9. Continuous Improvement:

Agile principles emphasize continuous improvement. Apply lessons learned from each sprint to refine the process, including better estimation and planning.

10. Blockchain Integration Considerations:

When estimating and planning, consider the complexities related to blockchain integration, such as smart contract development, security measures, and the use of Ethereum's capabilities.

6.3 Sprint Delivery Schedule

1. Divide the Project into Sprints:

Begin by dividing the overall drug traceability project into sprints. Sprints are time-bound iterations, usually lasting 2-4 weeks, during which specific sets of features or tasks are completed.

2. Prioritize User Stories:

Review the prioritized user stories from your backlog and select those that will be addressed in each sprint. Ensure that each sprint has a clear focus and goal.

3. Define Sprint Durations:

Decide on the duration of each sprint. Agile sprints are typically 2-4 weeks long, but you can choose the duration that works best for your team and project.

3. Create a Sprint Backlog:

For each sprint, create a sprint backlog that includes the user stories, tasks, and features that will be tackled during that sprint.

4. Assign Story Points:

Estimate the effort required for each user story in the sprint backlog using story points or other estimation methods. This helps in understanding the capacity of the sprint.

5. Distribute Workload:

Based on the team's capacity and story point estimates, distribute the workload evenly across the sprint backlog items. Ensure that the team can realistically complete the planned work during the sprint.

6. Define Milestones:

Within each sprint, set specific milestones or checkpoints for key tasks or features.

This helps in tracking progress and ensuring that the team is on target.

7. Adjust for Blockchain Integration:

Consider the complexities of blockchain integration in your delivery schedule. Tasks related to smart contract development, security testing, and Ethereum-specific considerations should be accounted for.

8. Iterative Development:

Remember that in Agile development, work is delivered incrementally. At the end of each sprint, you should have a potentially shippable product increment.

9. Continuous Review and Adaptation:

After each sprint, hold sprint reviews and retrospectives to gather feedback, evaluate progress, and make necessary adjustments to the delivery schedule or project priorities.

10. Release Planning:

Based on the progress in each sprint and the feedback received, plan releases of the drug traceability. These releases can be scheduled according to the completion of major features or project milestones.

11. Maintain a Release Calendar:

Maintain a release calendar that outlines when each sprint's deliverables or major releases are expected to be available to users or stakeholders. Share this calendar with the team and relevant parties.

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

Register Agriculture:

- The "Agriculture" struct represents the properties of a drug , including title, description, IPFS hash, and the owner's Ethereum address.
- The "agriculture" array stores registered drug.
- The " agriculture docs chain" function allows a user to register a new drug by providing the title, description, and the IPFS hash of the drug data. It also records the user's Ethereum address as the owner.
- You can emit an event to log the drug registration, providing information about the newly registered drug.

Smart Contract (Solidity):

// SPDX-License-Identifier: GPL-3.0

```
pragma solidity >=0.7.0 <0.9.0;
import "remix_tests.sol"; // this import is automatically injected by Remix.
import "hardhat/console.sol";
import "../contracts/3_Ballot.sol";

contract BallotTest {

    bytes32[] proposalNames;

    Ballot ballotToTest;
    function beforeAll () public {
        proposalNames.push(bytes32("candidate1"));
        ballotToTest = new Ballot(proposalNames);
    }

    function checkWinningProposal () public {
        console.log("Running checkWinningProposal");
        ballotToTest.vote(0);
        Assert.equal(ballotToTest.winningProposal(), uint(0), "proposal at index 0
should be the winning proposal");
        Assert.equal(ballotToTest.winnerName(), bytes32("candidate1"),
"candidate1 should be the winner name");
    }

    function checkWinninProposalWithReturnValue () public view returns (bool)
{
    return ballotToTest.winningProposal() == 0;
}
```

```
 }  
}
```

7.2 Feature 2

Transfer Ownership

- 1.The "drug" struct represents the properties of a drug, including title, description, IPFS hash, and the owner's Ethereum address.
 - The "drug" array stores registered assets.
 - The "register drug" function allows a user to register a new drug, which is owned by the user who registers it.
 - The "transferOwnership" function lets the owner of an drug transfer ownership to another user by specifying the drug's index and the address of the new owner.

Smart Contract (Solidity):

```
// SPDX-License-Identifier: GPL-3.0

pragma solidity >=0.7.0 <0.9.0;
import "remix_tests.sol"; // this import is automatically injected by Remix.
import "hardhat/console.sol";
import "../contracts/3_Ballot.sol";

contract BallotTest {

    bytes32[] proposalNames;

    Ballot ballotToTest;
    function beforeAll () public {
        proposalNames.push(bytes32("candidate1"));
        ballotToTest = new Ballot(proposalNames);
    }

    function checkWinningProposal () public {
        console.log("Running checkWinningProposal");
        ballotToTest.vote(0);
        Assert.equal(ballotToTest.winningProposal(), uint(0), "proposal at
index 0 should be the winning proposal");
        Assert.equal(ballotToTest.winnerName(), bytes32("candidate1"),
"candidate1 should be the winner name");
    }
}
```

```
}

function checkWinninProposalWithReturnValue () public view returns (bool)
{
    return ballotToTest.winningProposal() == 0;
}
}
```

7.3 Database Schema (if Applicable)

The traditional database schema is replaced with smart contracts, which define the structure and behavior of drug and related data on the blockchain. However, certain off-chain data and metadata may be stored in traditional databases or decentralized storage systems for efficiency and scalability. Below, I'll provide a high-level overview of how the database schema for a drug tracking on the Ethereum blockchain might look:

On-Chain Ethereum Smart Contracts:

Agriculture Contract:

- Attributes:
 - Agriculture title (string)
 - Agriculture description (string)
 - IPFS hash for drug data (string)
 - Owner (address)
 - Public status (boolean)
- Functions:
 - Register agriculture
 - Transfer ownership
 - Toggle public status

Off-Chain Metadata Storage (Traditional Database or Decentralized Storage):

1. User Profiles:

- Attributes:
 - User ID
 - Ethereum address
 - Username

Email
Other user-specific details

2.Audit Trail:

- Attributes:
 - drug ID
 - Action
 - User performing the action
 - Timestamp
 - Details of the action

3.Asset Metadata:

- Attributes:
 - drug ID
 - drug metadata

Additional details

This off-chain storage can include a traditional relational database for user profiles and audit trail data or decentralized storage systems like IPFS for storing drug metadata and potentially even the drug files themselves.

It's important to note that the Ethereum blockchain is primarily used for storing critical asset ownership and transaction data, ensuring immutability and transparency. Off-chain storage is often used to handle less critical data and to optimize data access and retrieval times.

The database schema can vary significantly based on the specific requirements of the drug traceability , and you may need to expand or modify the schema to suit your application's needs. The goal is to balance the benefits of blockchain's immutability with efficient data management and retrieval for users.

8. PERFORMANCE TESTING

8.1 Performance Metrics

Asset Upload and Retrieval Speed:

Metric: Average time taken to upload and retrieve drug.

Importance: Measures the speed of drug traceability ensuring quick access to agriculture

Blockchain Transaction Throughput:

Metric: Transactions per second (TPS) on the Ethereum blockchain.

Importance: Indicates how well the system handles blockchain transactions, which is crucial for scalability.

Smart Contract Execution Time:

Metric: Average time taken for smart contract execution.

Importance: Evaluates the efficiency of the blockchain-based logic governing drug ownership and access.

Drug Metadata Search Time:

Metric: Time it takes to search for drug based on metadata.

Importance: Measures the responsiveness of the system's search functionality.

User Authorization Latency:

Metric: Time it takes to validate and authorize user access to drug.

Importance: Ensures that authorized users can access drug promptly while maintaining security.

Storage Space Usage:

Metric: Amount of blockchain storage used by drug and associated data.

Importance: Evaluates the cost and efficiency of storage on the blockchain.

Drug Accessibility Uptime:

Metric: Percentage of time drug are accessible.

Importance: Measures the system's reliability and availability for users.

Security Audit Findings:

Metric: Number and severity of security vulnerabilities discovered during audits.

Importance: Identifies potential risks and the need for security improvements.

User Feedback and Satisfaction:

Metric: User surveys or feedback on system usability and performance.

Importance: Provides insights into user satisfaction and areas for improvement.

Ethereum Network Gas Costs:

Metric: Total gas costs incurred for transactions and contract interactions.

Importance: Measures the cost-efficiency of system operations.

Scalability Metrics:

Metric: System's ability to handle an increasing number of users and drug.

Importance: Assesses how well the system can scale with growing demands.

Audit Trail Accuracy:

Metric: Accuracy and completeness of the audit trail for drug tracking

Importance: Ensures a reliable record of drug history for compliance and accountability.

Data Backup and Recovery Time:

Metric: Time taken for data backup and recovery operations.

Importance: Evaluates the system's readiness for data recovery in case of failures.

9. RESULTS

9.1 Output Screenshots

The "agriculture" feature allows users to officially record and store information about a drug tracking on the Ethereum blockchain. This process establishes proof of ownership and a public record of the drug's existence. It is marked as "public" and can be viewed or accessed by anyone on the Ethereum blockchain.

Get drug tracking using blockchain is showing a status of the drug (manufacturing agriculture, transfer drug ownership)



"Transfer ownership" in the context of a drug tracking on the Ethereum blockchain refers to the ability of the current owner of a drug details to transfer ownership rights to another user. This feature allows drug owners to change the entity or individual that has control and ownership of a particular agriculture.



10. ADVANTAGES & DISADVANTAGES

ADVANTAGES

- **Immutability and Transparency:** All drug transactions and ownership changes are recorded on the Ethereum blockchain, providing an immutable and transparent audit trail. This enhances trust and security.
- **Ownership Verification:** Ethereum's smart contracts allow for clear ownership verification, reducing disputes and proving the authenticity of drug.
- **Decentralization:** The Ethereum blockchain operates on a decentralized network, reducing the risk of single points of failure and enhancing security and availability.
- **Security:** drug stored on the blockchain benefit from robust cryptographic security, making it difficult for unauthorized parties to tamper with or steal drug.
- **Global Accessibility:** Ethereum is a global network, making drug accessible to a worldwide audience without geographic restrictions.
- **Interoperability:** Ethereum's compatibility with various standards and protocols allow for easy integration with other blockchain-based systems.
- **Cost-Effective:** Smart contracts can automate drug tracking processes, reducing the need for intermediaries and saving costs.

DISADVANTAGES

- **Scalability:** Ethereum has faced challenges related to network congestion and scalability, making it less suitable for high-frequency drug tracking.
- **Gas Costs:** Every operation on the Ethereum blockchain consumes gas (transaction fees), which can make frequent drug tracking expensive.
- **User Experience:** Interacting with the blockchain can be complex for nontechnical users, leading to usability challenges.
- **Regulatory Concerns:** Blockchain-based drug may raise regulatory questions, particularly if they represent real-world drug or securities.

- Data Storage: Storing large drug directly on the blockchain can be inefficient and costly. Many drugs are stored off-chain or on decentralized storage systems like IPFS.
- Irreversible Transactions: Once a transaction is confirmed on the Ethereum blockchain, it is irreversible. Mistakes can be costly.
Smart Contract Vulnerabilities: Poorly written smart contracts can lead to security vulnerabilities and hacks, resulting in loss of drug.
- Privacy: The Ethereum blockchain is public, which means that drug data is visible to anyone. For private drug, additional privacy measures are needed.

11. CONCLUSION

Agriculture docs chain is critical for the health and well-being of patients, businesses, and the government. Patients and other parties involved in the drug supply chain could easily track the location of their medication if it had a dependable traceability mechanism. We have developed and evaluated a blockchain-based solution for the pharmaceutical supply chain to track and trace drugs in a decentralized manner. Specifically, our proposed solution leverages cryptographic fundamentals underlying blockchain technology to achieve tamper-proof logs of events within the supply chain and utilizes smart contracts within Ethereum blockchain to achieve automated recording of events that are accessible to all participating stakeholders.

12. FUTURE SCOPE

1. Develop a prototype to demonstrate the requirements shared via the guidelines of the DGFT/DAVA Team. Learned how the GS1 enabled 2D/1D barcodes will be associated with each drug pack and document the shortcoming in any.
2. Involved an early adopter to participate in our Traceability CRM Solution. We had two early adopters from the Pharma Industry based out in Kothur, Hyderabad, and another firm based out at Pashamylaram, Hyderabad. The former allowed us to gather requirements at the factory and allowed us to work with production packaging /warehouse teams.

3. I began with the bottom-up approach while documenting the requirements. As per guidelines from FDA ("Implementation in a hospital pharmacy in Argentina" GS1.org, 2014) , (GS1 Standards in the Pharmaceutical Supply Chain, 2018), from MHRA (GOV.UK, 2021) , From GS1 (Traceability system a must for drugs: GS1 chief , 2021) Based on the expectations, we learned from three guidelines from DGFT-DAVA/US FDA-US DSCSA /MHRA-FMD (Falsified Medical Directives).

4. Understanding the component of Serialization requirement. As per the initial user specification document composed by SolutionsMax Technology Services (An Enterprise Solutions for the Pharma Ecosystem, 2016-2021) Organizations need to identify various components such as the readiness of the ERP system as a master data repository, changes in artwork for all affected Stock Keeping Units (SKUs), and new systems such as the enterprise serialization manager, packaging line system, and edge systems.

5. Types of Business Reports measuring compliance attributes. . As per the initial user specification document composed by SolutionsMax Technology Services (An Enterprise Solutions for the Pharma Ecosystem, 2016-2021) Emphasis was to understand the various quality and business reporting that would be required to be established. Establishing a parent-child relationship between the packs that are being packaged so that traceability can be reported

13.APPENDIX

Source Code

Solidity coding :

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

contract Drug{
    addresspublic owner;

    constructor(){
        owner =msg.sender;
    }

    modifier onlyOwner(){
        require(msg.sender == owner,"Only the owner can perform this action");
        _;
    }

    struct Drug {
        string drugName;
```

```

        string manufacturer;
        uint256 manufacturingDate;
        address trackingHistory;
    }

mapping(uint256 => Drug)public drugs;
uint256public drugCount;

event DrugManufactured(uint256indexed drugId,string drugName,string
manufacturer,uint256 manufacturingDate);
event DrugTransferred(uint256indexed drugId,addressindexed
from,addressindexed to,uint256 transferDate);

function manufactureDrug(uint256 drugId,stringmemory
_drugName,stringmemory _manufacturer,uint256 _manufacturingDate)external
onlyOwner {

    address initialHistory;
    initialHistory = owner;

    drugs[drugId]= Drug(_drugName, _manufacturer, _manufacturingDate,
initialHistory);
    drugCount++;

    emit DrugManufactured(drugId, _drugName, _manufacturer,
 _manufacturingDate);
}

function transferDrugOwnership(uint256 _drugId,address _to)external{
    require(_to !=address(0),"Invalid address");
    require(_to != drugs[_drugId].trackingHistory,"Already owned by the
new address");

    address from = drugs[_drugId].trackingHistory;
    drugs[_drugId].trackingHistory = _to;

    emit DrugTransferred(_drugId, from, _to,block.timestamp);
}

function getDrugDetails(uint256
_drugId)externalviewreturns(stringmemory,stringmemory,uint256,address){

    Drug memory drug = drugs[_drugId];
    return(drug.drugName, drug.manufacturer, drug.manufacturingDate,
drug.trackingHistory);
}
}

```

Java script :

```
const { ethers } = require("ethers");

const abi = [
{
  "inputs": [],
  "stateMutability": "nonpayable",
  "type": "constructor"
},
{
  "anonymous": false,
  "inputs": [
    {
      "indexed": true,
      "internalType": "uint256",
      "name": "drugId",
      "type": "uint256"
    },
    {
      "indexed": false,
      "internalType": "string",
      "name": "drugName",
      "type": "string"
    },
    {
      "indexed": false,
      "internalType": "string",
      "name": "manufacturer",
      "type": "string"
    },
    {
      "indexed": false,
      "internalType": "uint256",
      "name": "manufacturingDate",
      "type": "uint256"
    }
  ],
  "name": "DrugManufactured",
  "type": "event"
},
{
  "anonymous": false,
  "inputs": [
    {
      "indexed": true,
      "internalType": "uint256",
      "name": "id",
      "type": "uint256"
    }
  ],
  "name": "DrugManufacturedEvent",
  "type": "event"
}
];
```

```
        "name": "drugId",
        "type": "uint256"
    },
    {
        "indexed": true,
        "internalType": "address",
        "name": "from",
        "type": "address"
    },
    {
        "indexed": true,
        "internalType": "address",
        "name": "to",
        "type": "address"
    },
    {
        "indexed": false,
        "internalType": "uint256",
        "name": "transferDate",
        "type": "uint256"
    }
],
{
    "name": "DrugTransferred",
    "type": "event"
},
{
    "inputs": [],
    "name": "drugCount",
    "outputs": [
        {
            "internalType": "uint256",
            "name": "",
            "type": "uint256"
        }
    ],
    "stateMutability": "view",
    "type": "function"
},
{
    "inputs": [
        {
            "internalType": "uint256",
            "name": "",
            "type": "uint256"
        }
    ],
    "name": "drugs",
    "outputs": [
        {
            "internalType": "string",

```

```
        "name": "drugName",
        "type": "string"
    },
    {
        "internalType": "string",
        "name": "manufacturer",
        "type": "string"
    },
    {
        "internalType": "uint256",
        "name": "manufacturingDate",
        "type": "uint256"
    },
    {
        "internalType": "address",
        "name": "trackingHistory",
        "type": "address"
    }
],
"stateMutability": "view",
"type": "function"
},
{
    "inputs": [
        {
            "internalType": "uint256",
            "name": "_drugId",
            "type": "uint256"
        }
    ],
    "name": "getDrugDetails",
    "outputs": [
        {
            "internalType": "string",
            "name": "",
            "type": "string"
        },
        {
            "internalType": "string",
            "name": "",
            "type": "string"
        },
        {
            "internalType": "uint256",
            "name": "",
            "type": "uint256"
        },
        {
            "internalType": "address",
            "name": ""
        }
    ]
}
```

```
        "type": "address"
    }
],
"stateMutability": "view",
"type": "function"
},
{
"inputs": [
{
    "internalType": "uint256",
    "name": "drugId",
    "type": "uint256"
},
{
    "internalType": "string",
    "name": "_drugName",
    "type": "string"
},
{
    "internalType": "string",
    "name": "_manufacturer",
    "type": "string"
},
{
    "internalType": "uint256",
    "name": "_manufacturingDate",
    "type": "uint256"
}
],
"name": "manufactureDrug",
"outputs": [],
"stateMutability": "nonpayable",
"type": "function"
},
{
"inputs": [],
"name": "owner",
"outputs": [
{
    "internalType": "address",
    "name": "",
    "type": "address"
}
],
"stateMutability": "view",
"type": "function"
},
{
"inputs": [
{
```

```

        "internalType": "uint256",
        "name": "_drugId",
        "type": "uint256"
    },
    {
        "internalType": "address",
        "name": "_to",
        "type": "address"
    }
],
"name": "transferDrugOwnership",
"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 = "0x65bAe337AA15b8e3c028B79E925FdB7eb9E2846c"

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

```

HTML coding:

```

<!DOCTYPE html>

<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
    <!--
      manifest.json provides metadata used when your web app is installed on a
      user's mobile device or desktop. See
      https://developers.google.com/web/fundamentals/web-app-manifest/
    -->

```

```
<link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
<!--
  Notice the use of %PUBLIC_URL% in the tags above.
  It will be replaced with the URL of the `public` folder during the
build.
  Only files inside the `public` folder can be referenced from the HTML.

  Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
  work correctly both with client-side routing and a non-root public URL.
  Learn how to configure a non-root public URL by running `npm run build`.
-->
<title>React App</title>
</head>
<body>
<noscript>You need to enable JavaScript to run this app.</noscript>
<div id="root"></div>
<!--
  This HTML file is a template.
  If you open it directly in the browser, you will see an empty page.

  You can add webfonts, meta tags, or analytics to this file.
  The build step will place the bundled scripts into the <body> tag.

  To begin the development, run `npm start` or `yarn start` .
  To create a production bundle, use `npm run build` or `yarn build` .
-->
</body>
</html>
```

GitHub :

<https://github.com/naanmudhalvan-SI/PBL-NT-GP--2984-1680669088>