Project Report

Team ID	NM2023TMID03276
Project Name	Drug Traceability

Submitted by

**TEAM LEADER : ** SUNMATHI S

TEAM MEMBER 1: SNEGA N

TEAM MEMBER 2: DIVYA M

TEAM MEMBER 3: SUMITHA R

PROJECT REPORT FORMAT

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**
   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

The "Drug Traceability Smart Contract on Ethereum Blockchain" is a pioneering solution poised to revolutionize pharmaceutical supply chain management. In response to the urgent need for enhanced transparency and security in drug tracking, this smart contract leverages blockchain technology. It establishes a decentralized network of nodes, underpinned by non-repudiation and robust security features, to ensure the verifiable traceability of pharmaceuticals. Traditional drug management systems often struggle with issues of authenticity and accountability, making them susceptible to counterfeit drugs and inefficiencies. In contrast, this Ethereum-based smart contract offers a transformative approach. It enables stakeholders to monitor the entire lifecycle of drugs, from production to distribution and ultimately to the end-user, with unparalleled transparency and trust.

Project overview:

The proposed Ethereum-based drug traceability will revolutionize the way individuals and organizations manage their drug. Leveraging the power of Ethereum's blockchain, the project will enable users to create, register, and trade various drug, ranging from Non-Fungible Tokens (NFTs) to fungible tokens, in a trust less and decentralized environment. Smart contracts will be at the core of this system, ensuring secure ownership management and stream lined drug transactions.

Every human being has a fundamental right to access to healthcare. It is the duty of the government to provide its citizens with high-quality infrastructure and healthcare services. Government agencies and the healthcare sector have been working to reduce the negative effects that bogus drugs have on people's health for the past few decades. According to the World Health Organization, 4 out of 10 medicines in emerging and underdeveloped nations are either fraudulent or may be contaminated. The international economy suffers from counterfeit drug costs in the billions of dollars, and organizations are forced to spend less on research and development (R&D).

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 underconsideration. The global standardized identification system from manufacturer to patient provides a safer healthcare supply chain.

2. LITERATURE SURVEY

1.Blockchain for drug traceability

Mueen Uddin, Khaled Salah, Raja Jayaraman,Sasa Pesic Samer, Ellahham

Pharmaceutical supply chain (PSC) consists of multiple stakeholders including raw materialsuppliers, manufacturers, distributors, regulatory authorities, pharmacies, hospitals, and patients. The complexity of product and transaction flows in PSC requires an effective traceability system to determine the current and all previous product ownerships. In addition, digitizing track and trace process provides significant benefit for regulatory oversight and ensures product safety. We propose two potential blockchain based decentralized architectures, Hyperledger Fabric and Besu to meet critical requirements for drug traceability such as privacy, trust, transparency, security, authorization and authentication, and scalability.

2.A Blockchain-Based Approach for Drug Traceability in Healthcare SupplyChain

Ahmad Musamih, Khaled Salah, Junaid Arshad

Healthcare supply chains are complex structures spanning across multiple organizational and geographical boundaries, providing critical backbone to services vital for everyday life. The inherent complexity of such systems can introduce impurities including inaccurate information, lack of transparency and limited data anend-to-end product tracking system across the pharmaceutical supply chain is paramount to ensuring product safety and eliminating counterfeits. Most existing track and trace systems are centralized leading to data privacy, transparency and authenticity issues in healthcare supply chains.

3. A semantic blockchain-based system for drug traceability

Khizar Abbas, Muhammad Afaq, Talha Ahmed Khan

Drug traceability 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 proposalis based on blockchain and semantic web technologies to enhance the representation capability of data in the pharmaceutical supply chain.

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 Drug Supply 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 throughan 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 amechanism 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 andunderground social media platforms.

References

[1] Ahmad Musamih, Khaled Salah, Raja Jayaraman, Junaid Arshad, Mazin Debe, Yousof Al- Hammad, Samer Ellahham—A Blockchain Based Approach for Drug Traceability in HealthcareSupply Chain‖ IEEE Access, vol 09, pp 9728 – 9743, 2021.

[2] Peng Nie; Xinxing Zhou; Chong Wang; Huilin Zheng; Yubing Zeng, —Design and Implementation of Coronavirus Vaccines Information Traceability System based onBlockchain‖, vol. 32, pp 122-124, 2021.

[3] Ching-Ling Chen, Yong-Yuan Deng, Chun-Ta li, Shunzhi Zhu, Yi-Jui Chiu, Pei – ZhiChen, —An IoT-Based Traceable Drug Anticounterfeiting Management System‖, IEEE Access, vol. 08, pp 224532-224546, 2020.

[4] Ahmad Musamih, Raja Jayaraman, Khaled Salah, Haya Hasan, Ibrar Yaqoob, Yousof Al- Hammadi, —Blockchain Based Solution for the Administration of Controlled Medication‖,IEEE Access, vol. 09, pp 145397-145412, 2021.

[5] Monalisa Sahoo, Sunil Samanta Singhar, Biswojit Nayak, Bhabendu Kumar Mohanta, —A Blockchain Based Framework Secured by ECDSA to Curb Drug Counterfeiting‖, IEEE - 45670,vol. 10, 2019.

[6] Tejaswini S; Karthik S; Roopashree H B; Trupti K N; Sushma M P, —Med Secure: A Blockchain based Authenticated System for Counterfeit Medicine in Decentralized Peer to Peer Network‖, 2021

7] Randhir Kumar, Rakesh Tripathi, ―Traceability of Counterfeit Medicine Supply ChainThrough Blockchain‖, COMSNETS, vol 19, pp 568 – 569, 2019.

[8] Zhengfei Wang,Lai Wang,Fu'an Xiao, Qingsong Chen, Liming Lu, Jiaming Hong, ―A Traditional Chinese Medicine Traceability System Based on Lightweight Blockchain‖ Journal of Medical Internet Research vol. 23, iss 6, pp 1-7, 2021.

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 percapita income and pricing monopoly. Developing countries faces major challenges on pharmaceutical industries investment due to poor infrastructure and lack of government funds for search 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 ofdrug manufacturers, wholesale distributors and community pharmacists.

3. INSUFFICIENT TECHNO-FUNCTIONAL RESOURCES FOR IMPLEMENTINGANDSUSTAINING DRUGTRACEABILITY

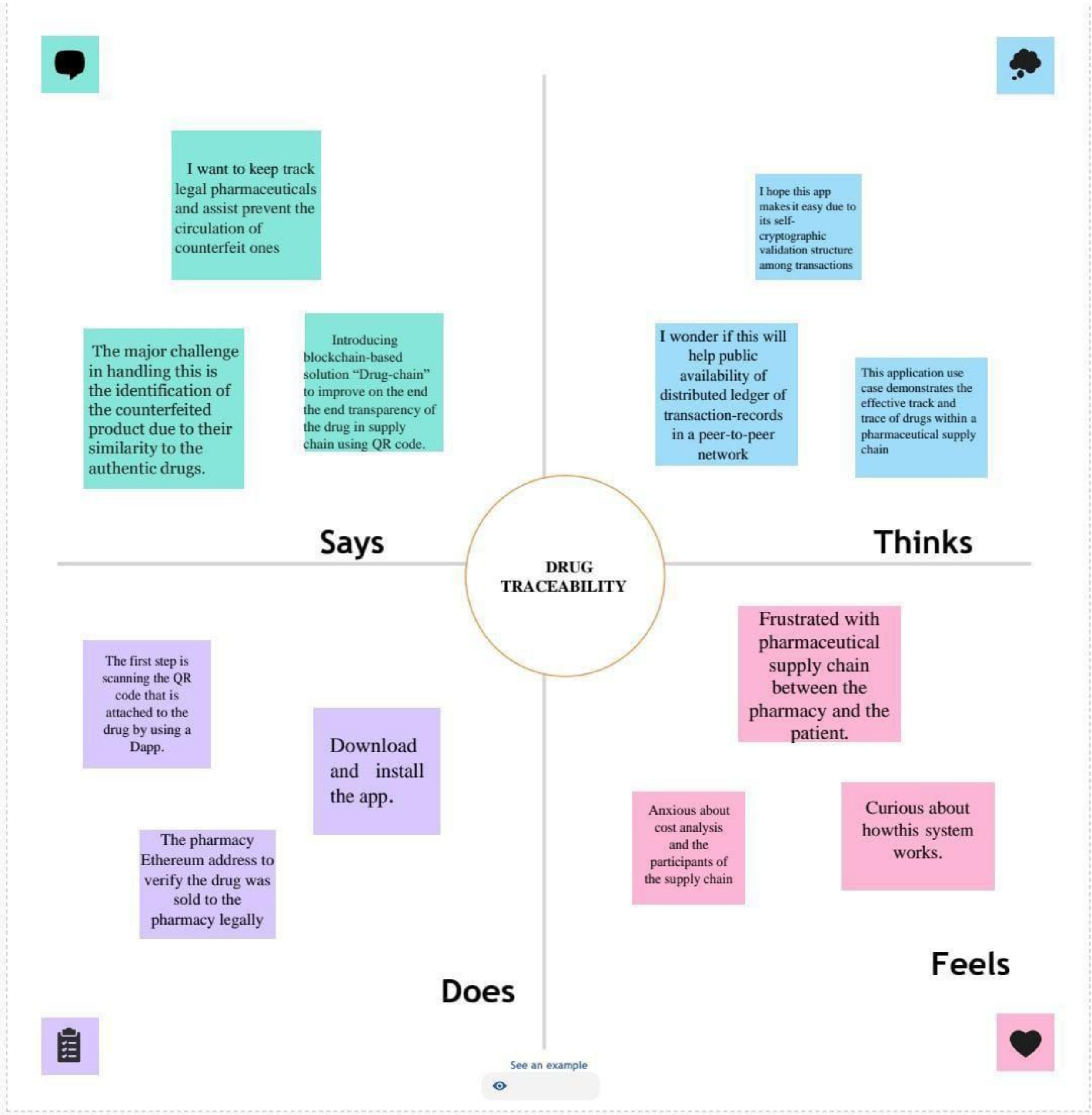
Drug counterfeiting a serious threat to public health. It is a collective job for every stakeholder in supply chain to prevent counterfeiting and illicit drugs. Implementing and sustaining serialization system fordrug traceability required skillful resources. Serialization and drug traceability processes involve packaginglines, 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 humanlife.

4.UNSECURE AND UNRELIABLE TECHNICAL INFRASTRUCTURE FOR DIGITAL DRUGTRACEABILITY

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 haveadopted serialization regulation under which drug manufacturer require to print a unique identifier printedwith the 2D barcode on individual drug unit. This unique identifier is key source for authenticating drug andtracing its origin of manufacturing. Printing unique identifier and keeping its key data in repository requiredspecial packaging equipment, tamper proof seals and global traceabilitysoftware.

3. IDEATION & PROPOSED SOLUTION

Empathy Map Canvas



Ideation & Brainstorming

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 and 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.

QR Code Tracking

Geo-Location Tracking

Blockchain Technology

RFID Tags

Mobile Apps

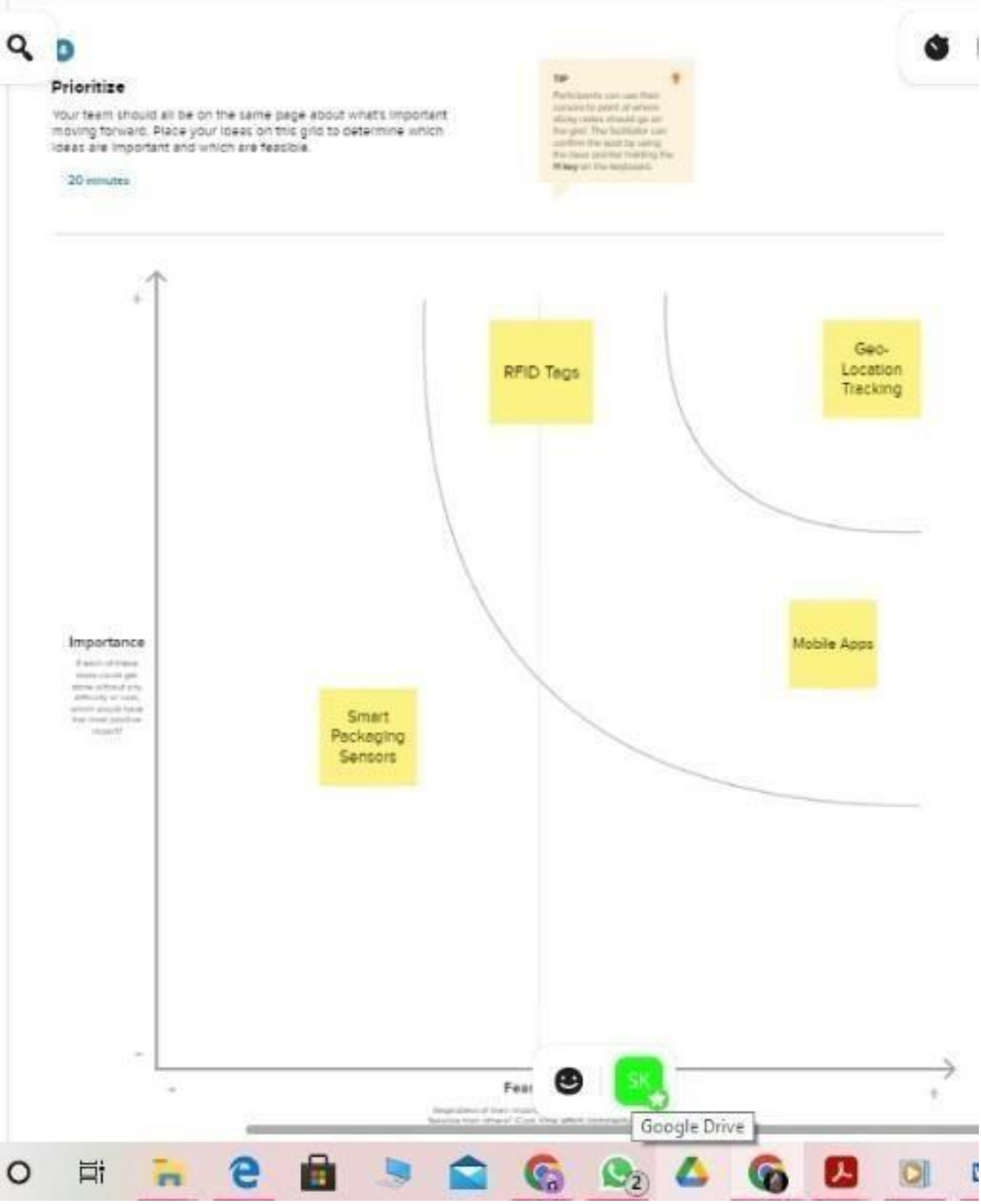
Smart Packaging Sensors

Data Analytics

Public Awareness Campaigns

SK

emp


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
RFID Tags
Geo-
Location
Tracking
Importance
Mobile Apps
Smart
Packaging
Sensors
Fea
SK
Google Drive

4. REQUIREMENT ANALYSIS

Functional requirement

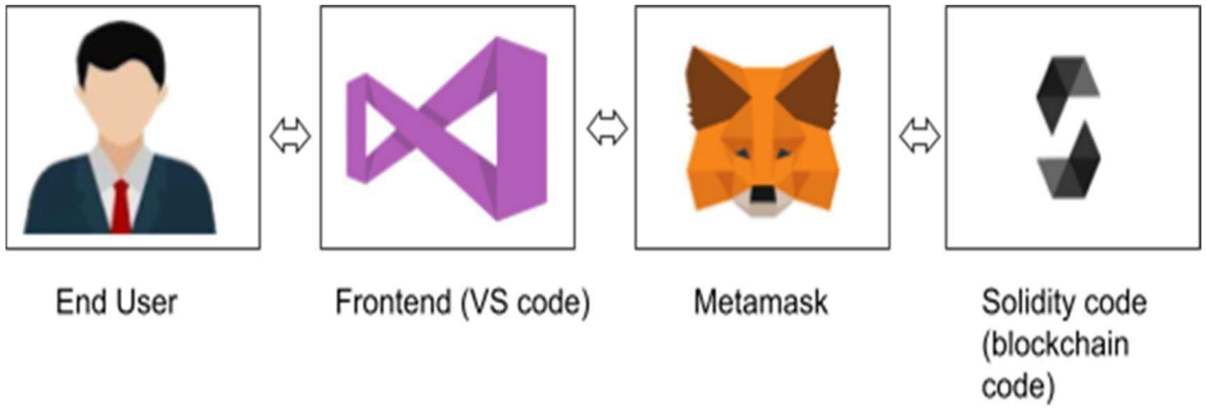
FR No.	Functional Requirement	Description
FR- 1	Drug Creation and Registration	Users should be able to create and register drug on the Ethereum blockchain. Drug registration should include metadata such as title, description, author, date, and any Other relevant information.
FR- 2	Drug Storage and Encryption	drug should be securely stored on the blockchain, with data encryption to protect their integrity and confidentiality.
FR- 3	Drug Tracking and Metadata Management	Users should have the ability to update drug metadata, including tags, categories, and descriptions. The system should support searching and filtering drug based on metadata.
FR- 4	Drug Ownership and Transfer	drug should be associated with specific owners, and ownership should be transferable through blockchain transactions. Ownership transfers should be securely recorded on the blockchain.
FR- 5	Access Control and Permissions	Define access control and permissions for drug viewing, editing, and transfer. Implement role-based access control (RBAC) for different users or user groups.
FR- 6	Smart Contracts:	Utilize smart contracts for managing drug ownership, transfers, and permissions. Implement contract functionality for executing predefined rules and logic.
FR- 7	Interoperability with Other Systems	Ensure interoperability with other drug tracking or blockchain platforms. Support importing and exporting drug and metadata to and from other systems.
FR- 8	Content Preview and Playback	Provide the ability to preview or play drug directly within the system. Ensure compatibility with various file formats and viewers.
FR- 9	Reporting and Analytics	Generate reports on drug usage, ownership changes, and access patterns. Provide analytics to help users understand drug performance and trends.
FR- 10	Auditing and Compliance	Maintain an audit trail of all drug -related activities. Ensure compliance with relevant data protection and copyright regulations.

Non-Functional requirements

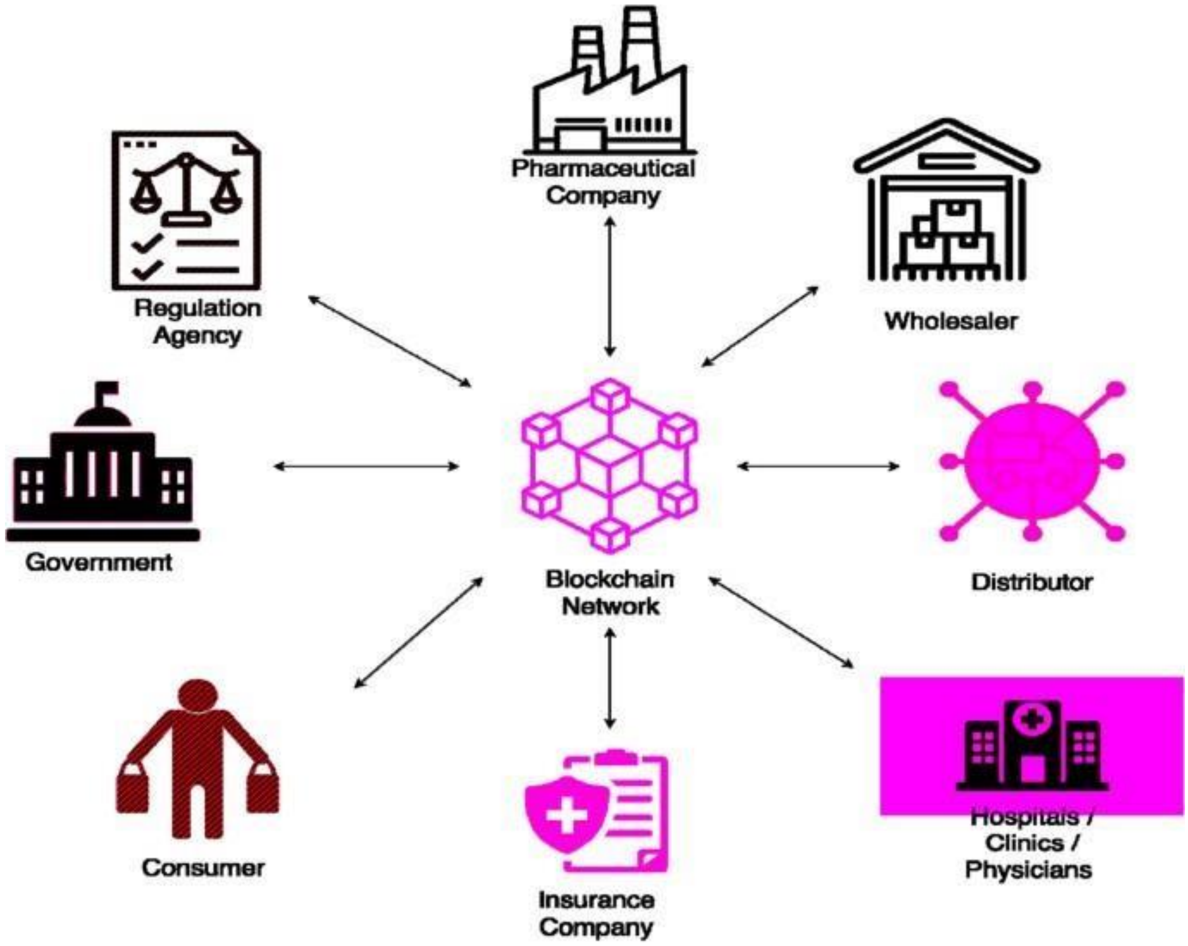
NFR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	For drug traceability, we've proposed a totally new blockchain system. This solution is more secure and scalable than other options currently available. Furthermore, the suggested system can prune its storage effectively, resulting in 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 includes, execution and owner, time, location of transaction, and which stakeholders were involved.

5. PROJECT DESIGN

Data Flow Diagrams & User Stories



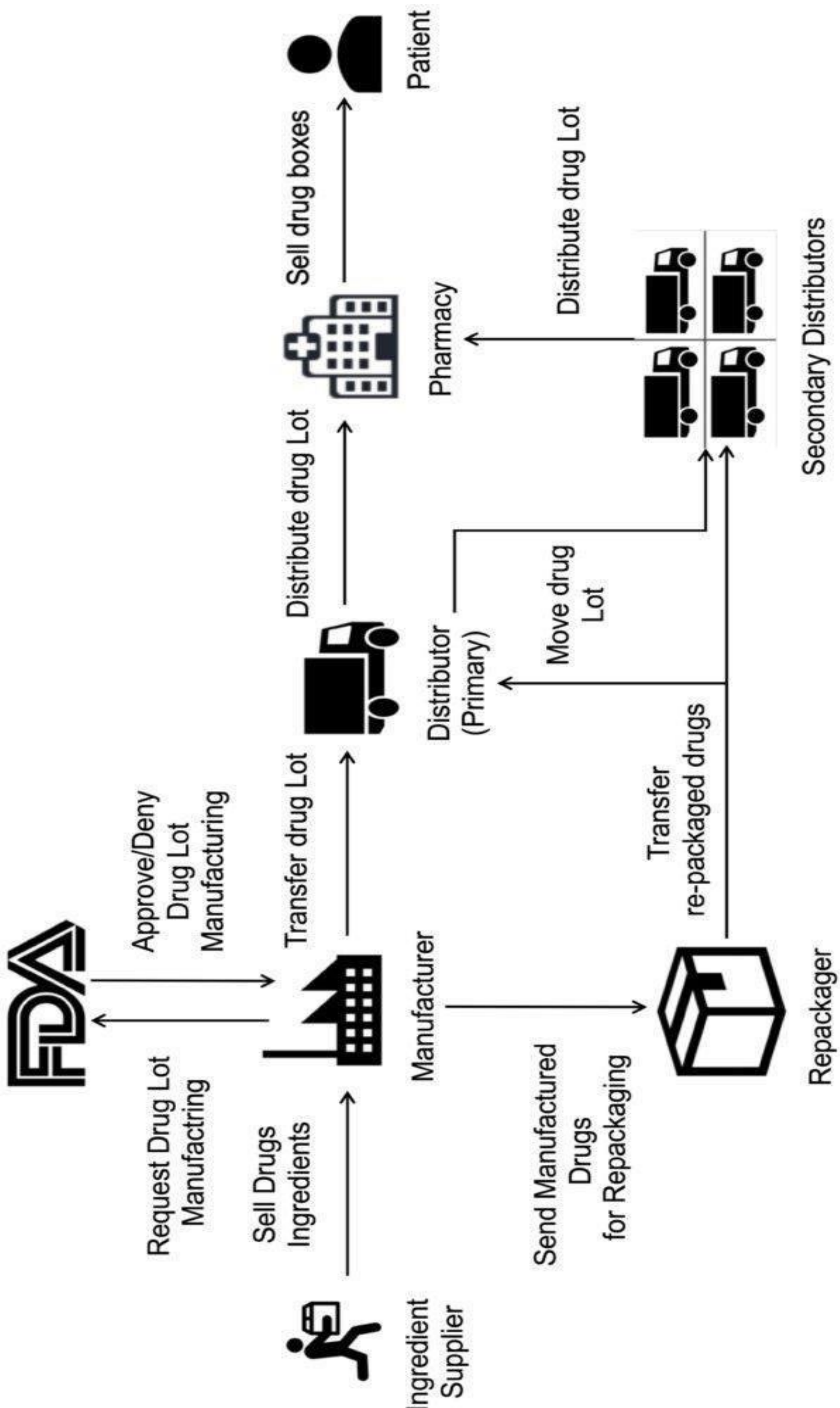
Data Flow Diagrams



User Stories

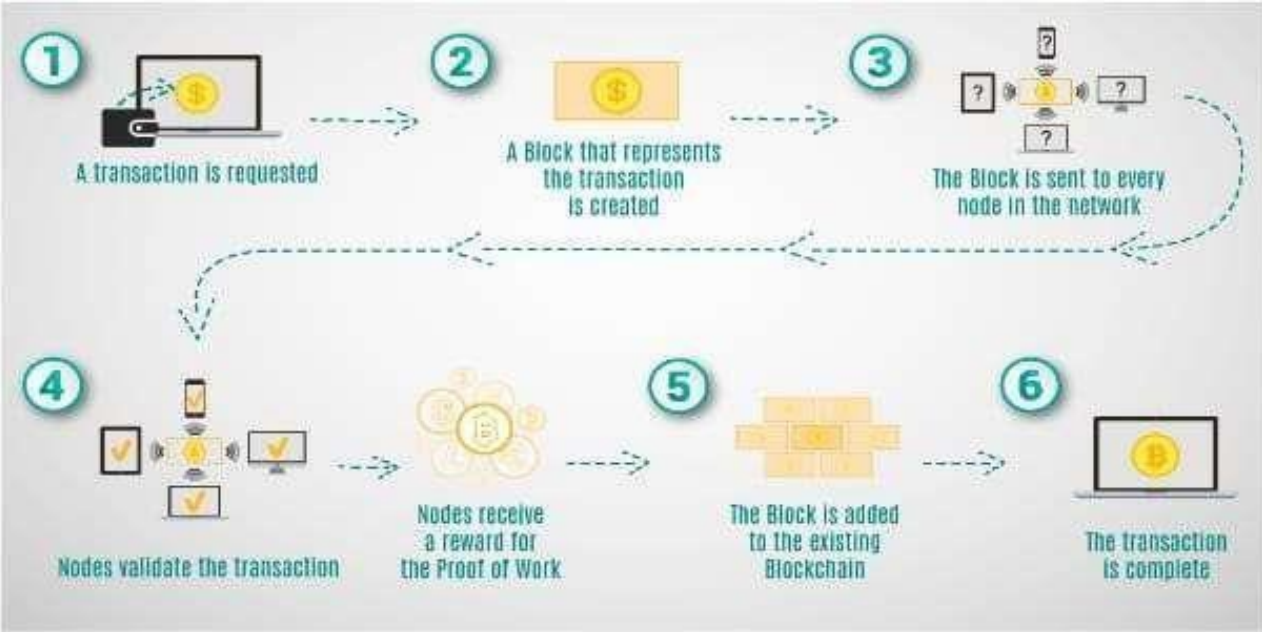
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	team Member
Content Creator	Drug Upload and Management	USN-1	As a content creator, I want to upload multiple image and video assets to the drug tracking with drag-and-drop functionality	Users should be able to drag and drop multiple assets onto the drug traceability interface, and the system should process and upload them efficiently	High	Sunmathi Snega
Content Creator	drug 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 well organized	Metadata fields should be easily accessible and editable, and changes should be immediately reflected in drug information	Medium	Sunmathi
Marketing Manager	Drug 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	Divya Sumitha
Marketing Manager	drug 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	Sumitha Sunmathi
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	Divya Snega
Administrator	System Management	USN-6	As an administrator, I need to set up automated data backups and a disaster recovery plan for data safety.disciplie	The system should regularly back up data and provide a documented recovery plan to prevent data loss.	High	Snega

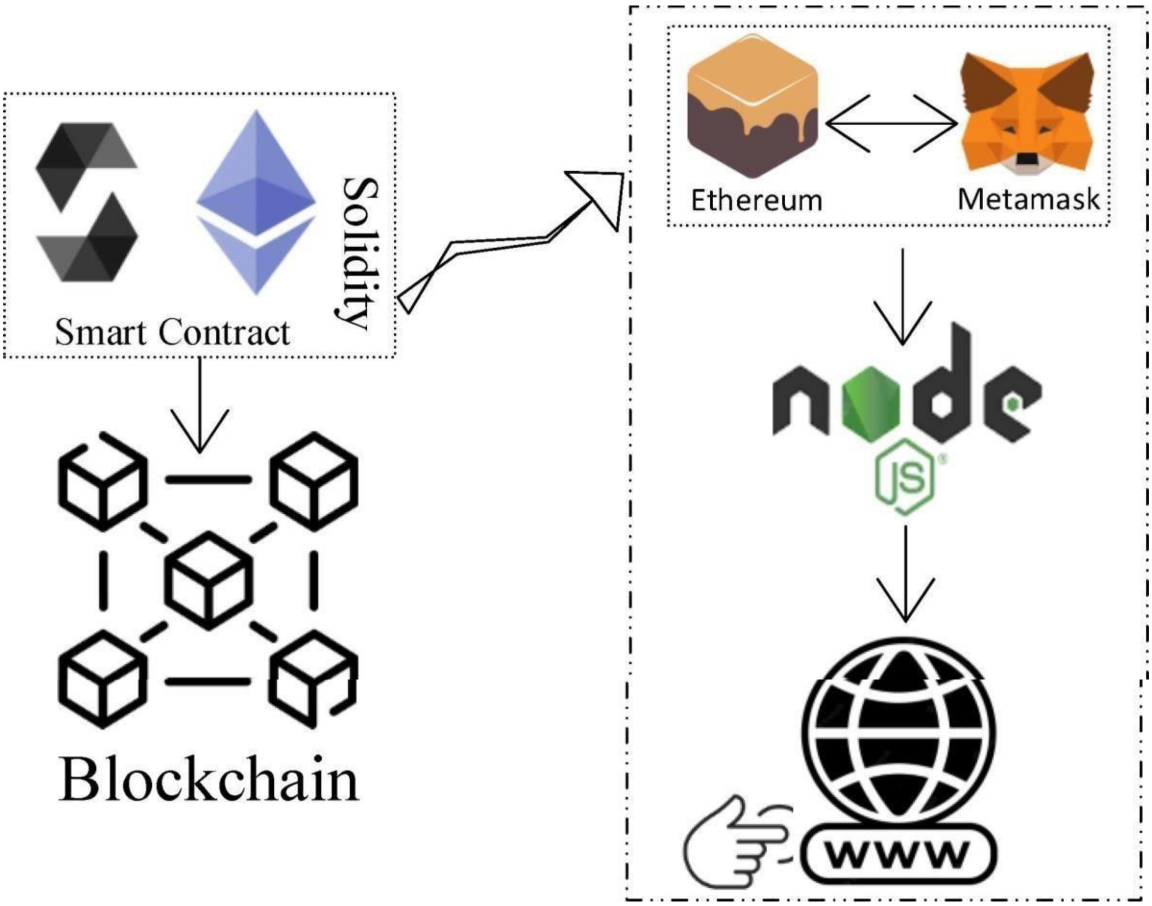
Solution Architecture



6. PROJECT PLANNING & SCHEDULING

Technical Architecture





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.

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 Ethereumspecific 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 projectalong with code)

7.1 Feature 1

Register drug

- 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 drug.
- The " register drug" 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: MIT
pragma solidity ^0.8.0;

contract Drug{
   address public 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;
    uint256 public drugCount;

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

    function manufactureDrug(uint256 drugId, string memory _drugName,
string memory _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) external view returns (string
memory, string memory, uint256, address) {

        Drug memory drug = drugs[_drugId];
```

return (drug.drugName, drug.manufacturer, drug.manufacturingDate, drug.trackingHistory);
    }
}

7.2 Feature 2

Trasfor 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: MIT
pragmasolidity^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);
    }
}
```

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:

drugContract:

- Attributes:
  - drug title (string)
  - drug description (string)
  - IPFS hash for drug data (string)
  - Owner (address)
  - Public status (boolean)
- Functions:
  - Register drug
  - 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 drug.

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 drugare 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:

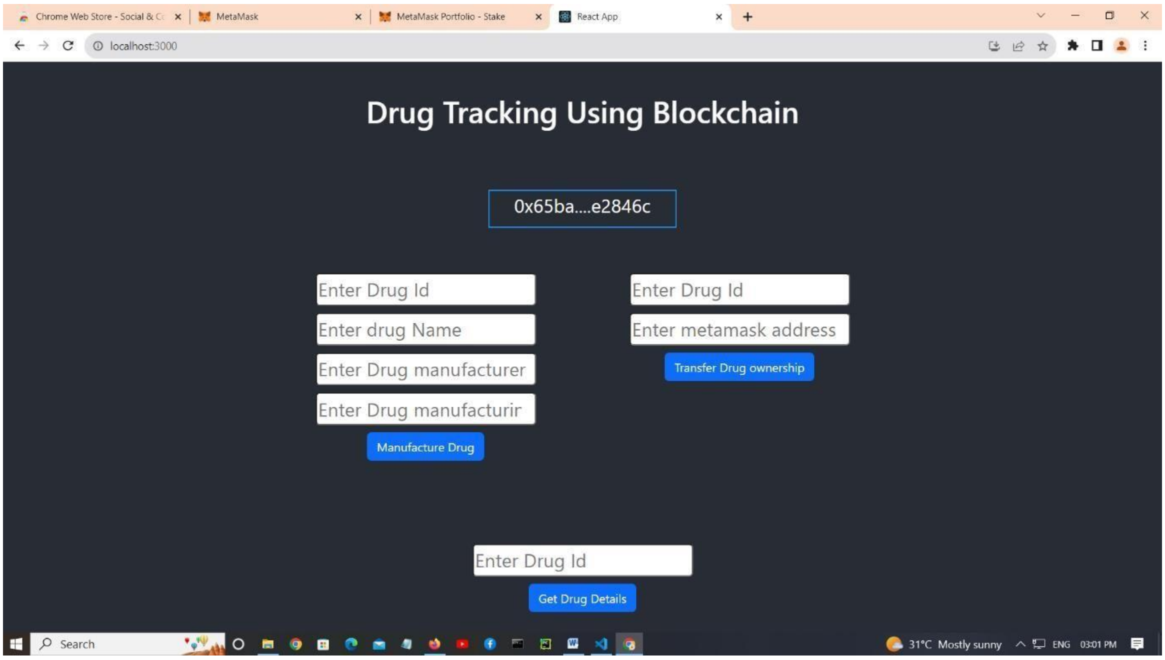
Metric: Time taken for data backup and recovery operations.
Importance: Evaluates the system's readiness for data recovery in case of failures.

9. RESULTS

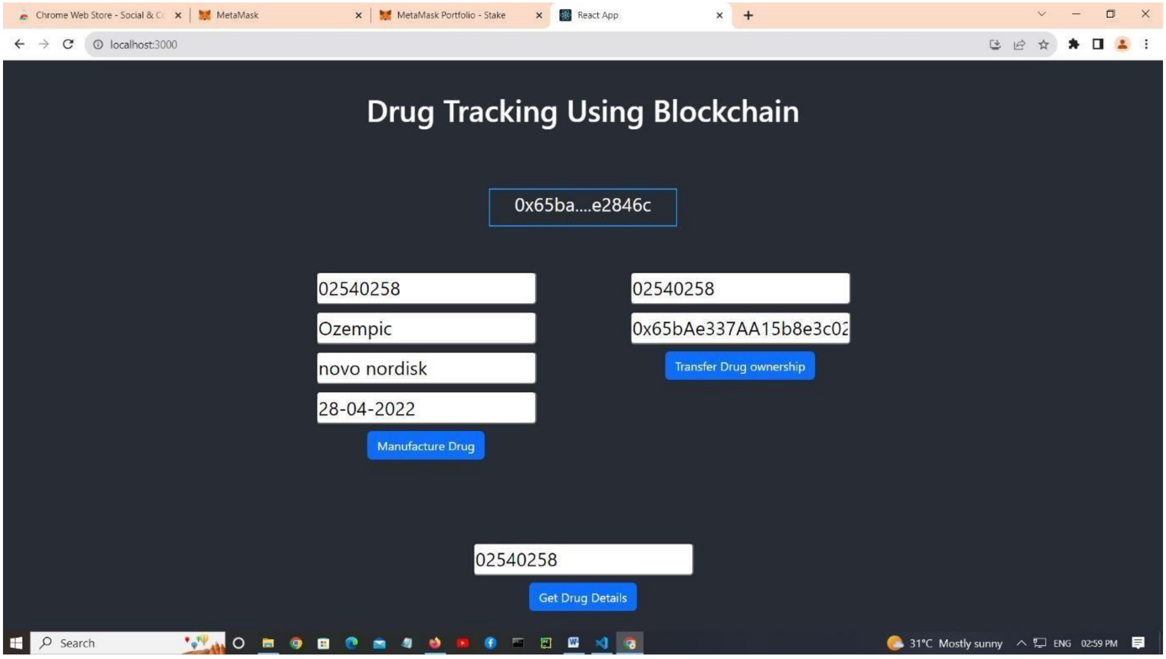
9.1 Output Screenshots

The " drug" 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 drug, transfer drug ownership)



"Transfer ownership" in the context of a drug trackingon the Ethereum blockchain refers to the ability of the current owner of a drugdetails 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 drug.



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 nontechnicalusers, 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 drug 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

Drug traceability 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 beestablished. 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
pragmasolidity^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);
    }
}
```

<u>Java script :</u>

```
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": "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 = " 0xAB7123389d20Aa927dA44A1C7981A89BbEcE1eb3"

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/SunmathiS/Nan-mudhalvan-

Project Video Demo Link:

https://www.mediafire.com/file/2bti743o8kyzj20/VID-20231031-WA0005.mp4/file