Project Report

Team ID	**NM2023TMID03267** **C617C160A2C5FECA4EF4CA7DD8E81935**
Project Name	Food tracking system

SUBMITTED BY

TEAM LEADER : K.SINDHUSA
TEAM MEMBER 1 : A.ROJA
TEAM MEMBER 2 : M.YUGA BHARATH
TEAM MEMBER 3 : N.GOKUL

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**
   Performace Metrics
9. **RESULTS**
   Output Screenshots
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**

Source Code

GitHub & Project Demo Link

1. INTRODUCTION

The awareness of protecting human health, which has increased on a global scale in recent years, has also shown itself in the food industry, and it has gained great importance that food be safe in order to lead a healthy life. Access to and the consumption of safe food is a right that every person should have. Food safety covers the whole process from the production stage of the food until it reaches the consumer [1]. More than 60% or about 1 billion tons of food is wasted within the supply chain while harvesting, processing, shipping, and storing [2]. For instance, nearly 492 million tons of perishable food were wasted in the year 2011 because of the ineffective and poor management of the food tracking systems [3].

According to the FAO (UN Food and Agriculture Organization), food security is defined as the ability of every person to have access to sufficient, safe, and nutritious food at all times to lead an active and healthy life [4]. Food safety is possible by taking every step of the food under control in the whole process, starting from the raw material until it reaches our table. Especially with the spread of digitalization, it is expected that the number and success of food tracking systems will increase [5]. The benefits of digitalization will be most clearly and largely achieved through the use of emerging technologies.

With the maturation and spread of emerging technologies, it has started to become a part of our daily life, shaping life and paving the way for digitalization [6]. One of the main reasons for digitalization is to reduce or even eliminate the need for manpower [7]. Undoubtedly, one of the prominent technologies at this point is blockchain technology, which is the infrastructure of cryptocurrencies such as Bitcoin, and many application areas have begun to emerge with the possibility of making transactions without intermediaries. Due to the advantages offered by blockchain technology, it has recently gained the notion of being the technological basis on which many applications are developed. It can be used in many different areas, such as smart city applications, IoT, health, energy management, and land registration systems [8]. Blockchain-based security approaches in remote patient monitoring using IoT devices are seen in the literature. In addition, blockchain-based cryptographic technologies could be used for deployment in the IoT.

Project overview :

The proposed Ethereum-based drug traceability will revolutionize the way individuals and organizations manage their drug.[1]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.[2] 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

Contributions of Proposed Study The novelty and contributions of this proposed study are: • A total of 0.038 s for latency was gathered with the proposed system, which is 435 times better than Ethereum, one of the most popular blockchain infrastructures. • A transmission per second value of 285, reception per second value of 335, and CPU load value of 19.22 are obtained with the proposed blockchain-based system. • Through the proposed blockchain-based system to be established, suppliers that make unfair price increases in the case of a food shortage, which will become a bigger problem in the coming periods due to the COVID-19 pandemic, will be prevented. • It is the first study in which the live use of the blockchain-based food tracking system is carried out and the satisfaction survey is carried out. • A total of 75.31% of the users who use the application liked the interface of the application; 97.54% of the users stated that they found the application extremely useful and that they would like to use it again in the future.

2. LITERATURE SURVEY

1.Blockchain for Food tracking system

One of the foremost blockchain-based food tracking systems is the "Food Trust" system developed by IBM. Announced for the first time in 2017, Food Trust has provided traceability in the food supply chain to 80 different brands so far by using blockchain technology. With this traceability, the supply process from producers to consumers can be followed in detail. IBM's open-source technology based on Hyperledger Fabric allows companies to set their own rules on the system. It is argued that the traceability offered by the Food Trust not only helps food safety but also helps producers with food freshness, sustainability, and waste. Announcing that more than 5 million food products already on the shelves are included in the system, IBM seems confident that this platform will grow strongly. Among the companies using this application are giants such as Dile, Kroger, McCormick and Company, Nestle, Tyson Foods, and Unilever [11]. Walmart has used blockchain to record where every piece of meat it buys from China comes from, where it is processed, where it is stored, and all transactions related to its sale, along with its historical course. All detailed information about the farm where the meat comes from, the factory where it is processed, the batch number of the product, the storage temperature of the product, and transportation can be tracked on the blockchain. In addition to the benefits of processing speed, information sharing, and transparency, the main purpose is summarized as increasing food safety [12]. Provenance has conducted a blockchain-based pilot project in Indonesia to transparently track the movement of products from sea to table in the fishing industry. The seafood trade consists of a very large fishing network, and it is a very difficult sector to control quality. There is no reliable audit in the sector. This project aims to help stop illegal, excessive, harmful to the sea and the environment, and non-sanitary fishing violations in the tuna fish industry. Thus, consumers will be able to view the source of the food they supply transparently, and a legal basis will be established to combat illegal fishing. With the use of this example, the aim is that the use of blockchain technology will facilitate transparency, tracking, and auditing, thus ensuring the safety of food products, preventing illegal and excessive fishing, and preventing damage to the environment [13]. Kim proposes a blockchain-based traceability system with different ontologies, where each one could accomplish and be part of certain transactions. He offers the use of smart contracts. Ethereum, with the Solidity programming language, was used in his study [14]. Feng Tian et al. propose a blockchain solution for agriculture traceability to ensure that the HACCP principles and requirements are addressed during the production, transportation, and preservation of a product [15]. Moreover, Daniel Tse et al. focus on the increasingly serious problem of food safety in China and propose a blockchain solution for the agriculture supply chain, based on the information and transaction security between all the involved parties. In this work, a PEST (political, economic, social, and technological) environment analysis took place to define the challenges and the opportunities of the DLT (Distributed Ledger Technologies) solution [16]. In addition, Francesco Marinello et al. offer a blockchain-based solution focusing on the animal products supply chain in Italy [17]. Kumar et al. propose a rice supply chain system that uses blockchain technology to assure the safety of rice during its flow through the supply chain [18]. Maria Elena Latino et al. propose another interesting idea regarding the agriculture supply chain and the use of Industry 4.0 principles [19]. They refer to the idea of food democracy, according to which consumers are considered citizens and the food is not a good but a civil right. The authors advertise the idea of voluntary traceability and combine it with Industry 4.0 technologies. The significance of voluntary traceability is highlighted, focusing on the volume and the quality of the data collected for each product, as well as the need for a big data platform to handle them. Islam and others published work about the visualization of food supply chain management. Their research aims to propose a new visualization approach that allows supply Electronics 2022, 11, 2491 4 of 26 chain operators to collaborate effectively in the design process of FTSs capable of maintaining streamlined information flow, minimizing information loss, and improving supply chain performance [20]. Bahga et al. proposed work to monitor the food supply chain tracking system on a cloud-based architecture. The proposed system, called CloudTrack, provides the global information of the entire fleet of food supply vehicles and is proposed to be used to track and monitor a large number of vehicles in real time [21]. Caro et al. propose an integrated solution of a blockchain platform named AgriBlockIoT in the agriculture supply chain [22]. AgriBlockIoT is a fully distributed system that uses blockchain technology in combination with IoT devices to collect and distribute traceability data. The proposed solution was tested with two Ethereum and Hyperledger Sawtooth blockchain platforms. AgriBlockIoT enables the integration of IoT and blockchain technologies, creating transparent, fault-tolerant, immutable, and auditable records which can be used for an agri-food traceability system.

References

[1] Ahmad Musamih, Khaled Salah, Raja Jayaraman, Junaid Arshad, Mazin Debe, Yousof Al-Hammad, Samer Ellahham—A BlockchainBased Approach for food tracking system 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 on Blockchain‖, 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, —BlockchainBased Solution for the Administration of Controlled Medication‖, IEEE Access, vol. 09, pp 145397-145412, 2021.

[5] Monalisa Sahoo, Sunil SamantaSinghar, 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

Problem Statement Definition

Imagine you meet a small startup company planning to launch an Online Food Delivery System like Swiggy/Zomato that allows consumers & service providers to interact in real-time. They want to support both Mobile App and Web-based applications. Like many small start-ups, they are confident that they will be the next big thing and expect significant, rapid growth in the next few months. With this in mind, they are concerned about the following:

Scaling to meet the demand, but they are not sure when and how the demand will grow — they are very concerned about buying too much infrastructure too soon or not enough too late!
Effective distribution of load
The ability for Service Providers to send notifications to consumer
configurable database/s and data access layer to yield high performance and throughput
Allocate food delivery personal efficiently
Design for easy onboarding and searchability of restaurants
Prediction of the food delivery time for order
Proposed Solution
Proposed below is a high-level reference architecture for Online Food Delivery systems.

This proposed architecture is generic and it can be deployed to any of cloud provider like AWS/GCP/Azure

Assumptions
Delivery partners are provided with devices that have inbuilt GPS. This will help to locate their current location accurately.
Integration with map provider is there and we get details about routes, traffic and commute time
Proposed Architecture Diagram

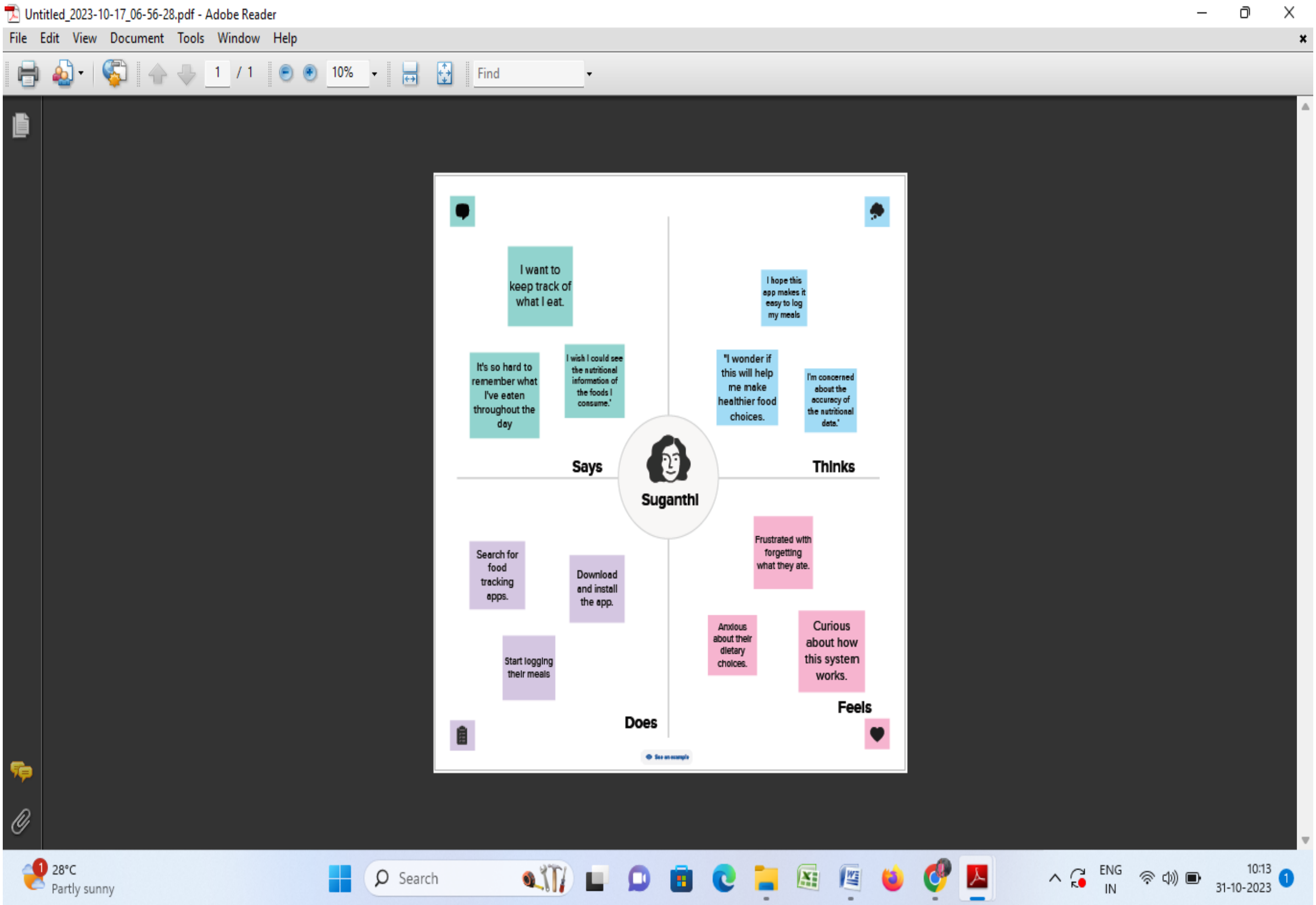
Possible Tech Stack

Solution Details
Core Features of Customer App
Searching menu: Allow your users to search for different restaurants, cafes by location, and cuisines. Using the search filter, users can easily find their favorite eating places, list menu, offers, etc.
Order placement: The user can place an order of selected dishes and food. They just need to cross-verify their preferred dish, delivery time, and proceed check-out.
Tracking Delivery Partners: With real-time tracking features, it becomes easy for users to track delivery drivers and know their real-time location information. Users can check the time taken by the food delivery executive to deliver their parcel.
Payment gateway integration: You provide the users with multiple payment options like credit/debit cards, different wallets like Google Pay, Paytm, Phonepe, UPI, etc
Core Features of Delivery Partner
Delivery Partner's profile: Through this feature, a driver can keep his profile update. It contains his full name, address, email, contact number, photo, and other personal information.
Notification for orders: Through push notifications, drivers can get constant updates & alerts for new orders. It will help in the accurate delivery service of your restaurant.

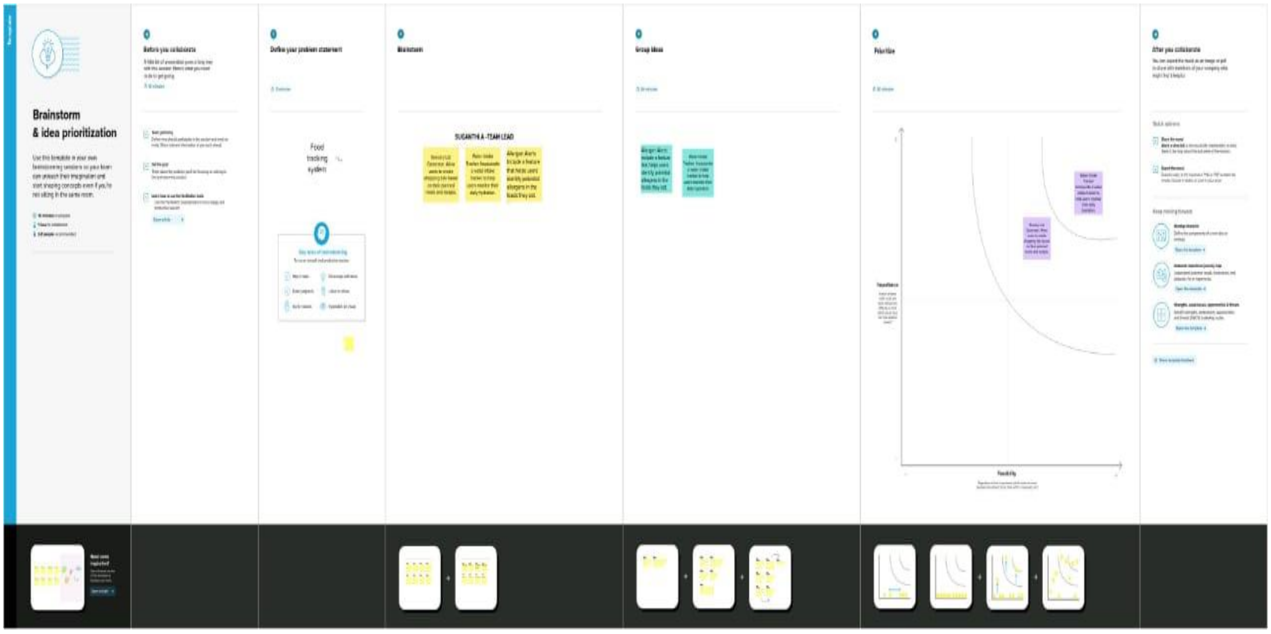
Map for the delivery route: Integrate Google Map or other providers and allow drivers to choose the shortest and fastest routes to reach the location.
Core Features of Food Partners/ Restaurants Restaurant Profile/Menu: Through this feature, a restaurant owner can add their restaurant details, menu and its availability, price, preparation times, etc
Notification for orders: Through push notifications, Restaurants can get constant updates & alerts for new orders. It will help in the accurate delivery service of your restaurant.
Notifications for Pickup Partners: They will get alerts about delivery partners, their location when they will pick up, etc.
Payment Details: Information about the payment received from the food delivery system for their orders Core Features for Food Ordering System Admin
Restaurant management: Being on the admin panel, one can directly manage all the restaurants by adding, updating, and removing any eating joint from the list. He can also check active restaurant status.
Analytics & report generation: Using analysis and report features, you can get real-time insights of reports and other accounting informati…

3. IDEATION & PROPOSED SOLUTION

Empathy Map Canvas



Ideation & Brainstorming



REQUIREMENT ANALYSIS

Functional requirement

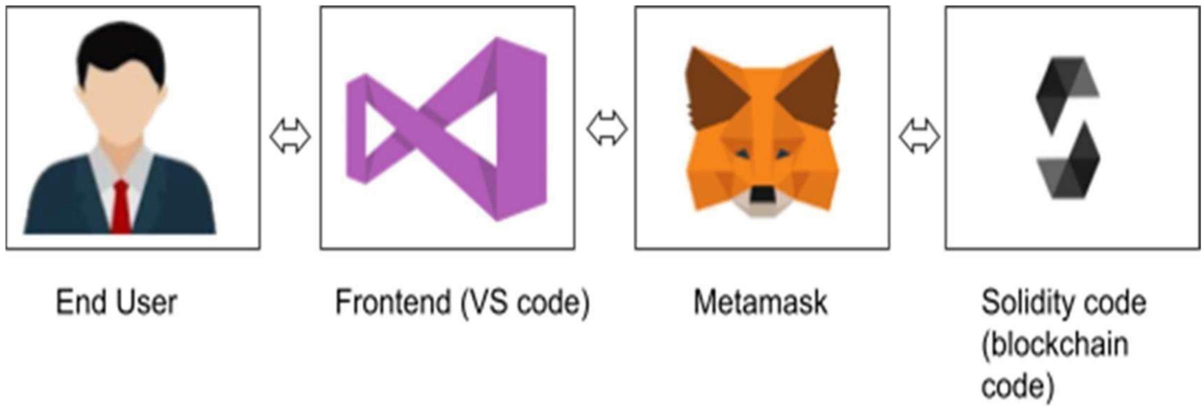
FR No.	Functional Requirement	Description
FR- 1	Food Creation and Registration	Users should be able to create and register drug on the Ethereum blockchain. food registration should include metadata such as title, description, author, date, and any otherrelevant information.
FR- 2	Foods torage and Encryption	Food should be securely stored on the blockchain, with data encryption to protect theirintegrity and confidentiality.
FR- 3	Food Tracking and Metadata Management	Users should have the ability to update drug metadata, including tags, categories, and descriptions.The system should support searching and filteringdrug based on metadata.
FR- 4	food Ownership and Transfer	drug should be associated with specific owners, and ownership should be transferable through blockchain transactions. Ownership transfers should be securely recordedon 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) fordifferent 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 andviewers.
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 protectionand copyright regulations.

Non-Functional requirements

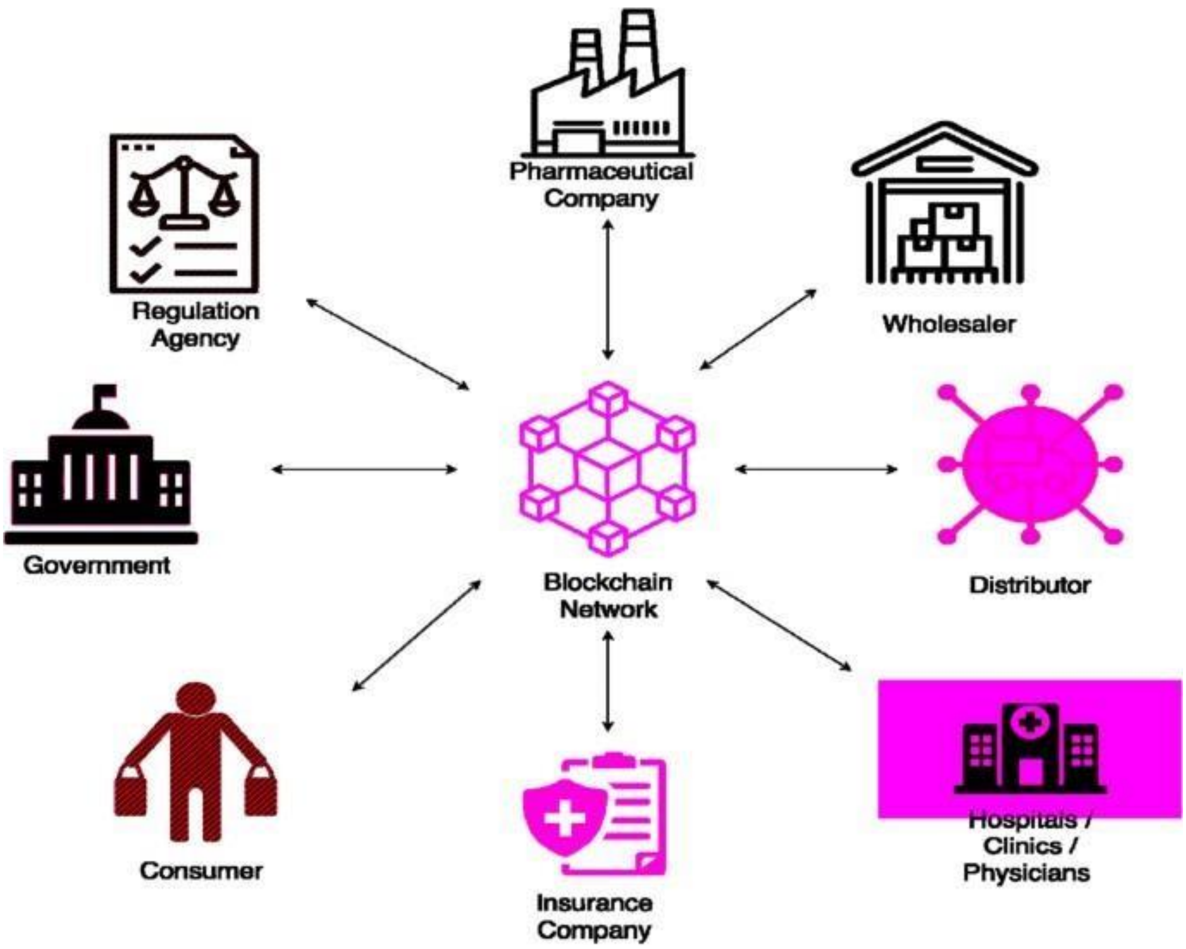
NFR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	For drug traceability, we've proposed a totally new blockchain system. This solution is more secure and scalable than other options currently available. Furthermore, the suggested system can prune its storage effectively, resulting in 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, regardlessof 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.

4. PROJECT DESIGN

Data Flow Diagrams & User Stories



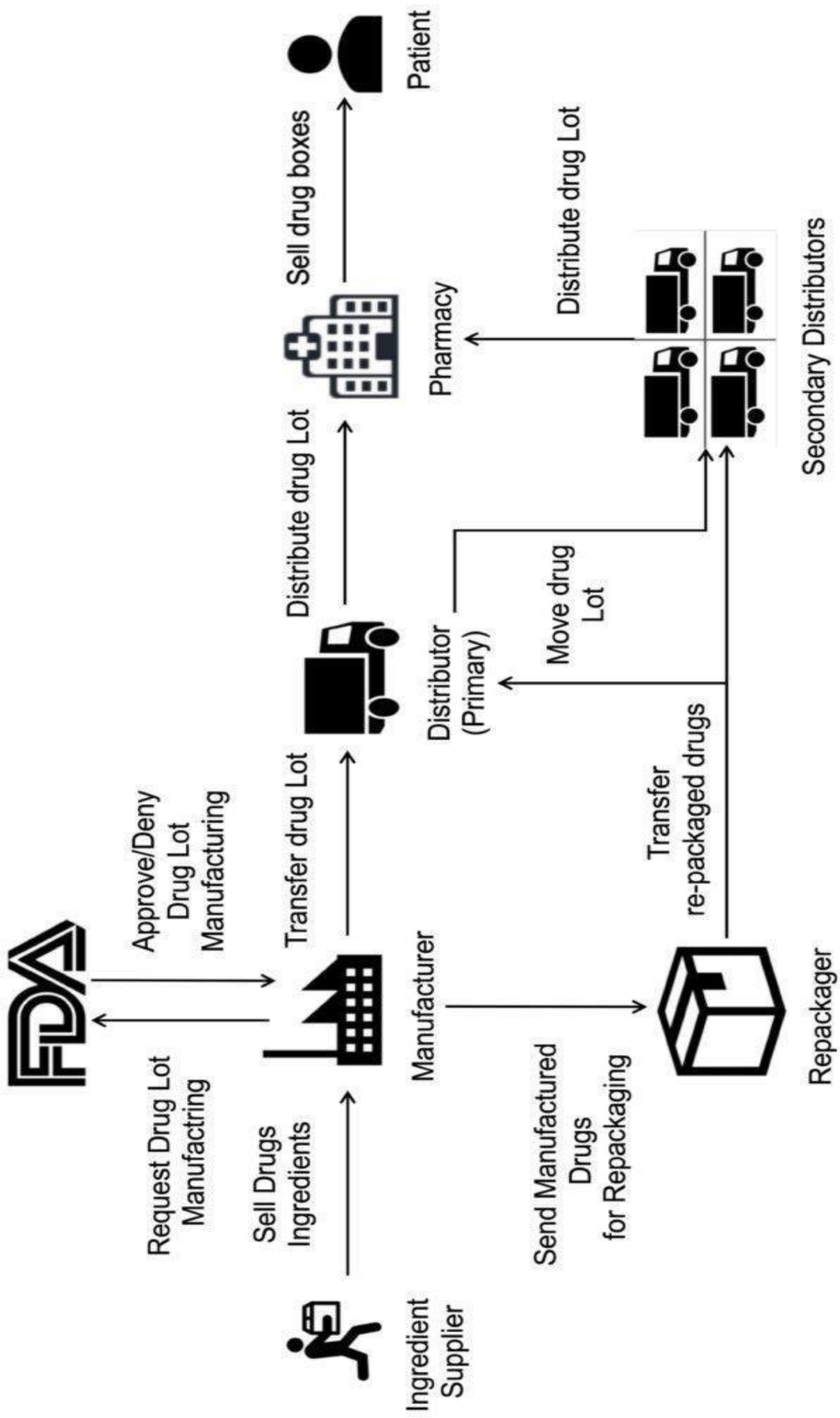
Data Flow Diagrams



User Stories

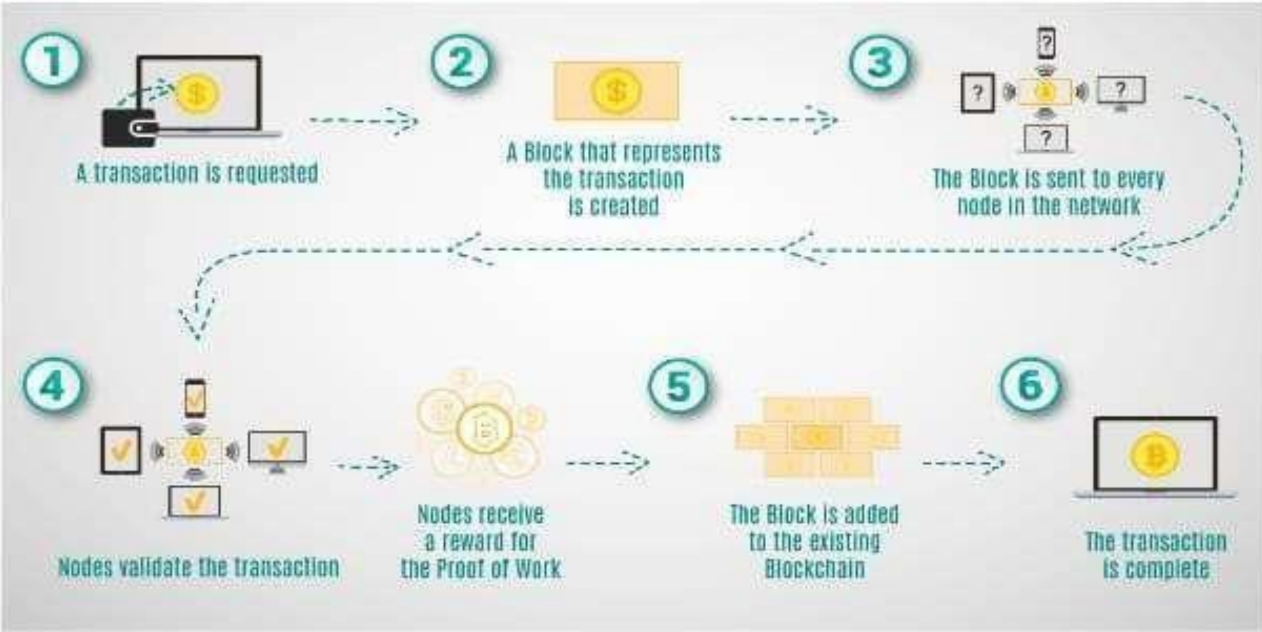
User Type	Functional Requireme nt (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	team Member
Content Creator	Drug Upload and Management	USN-1	As a content creator, I want to upload multiple image and video assets to the drug tracking with drag-and-drop functionality	Users should be able to drag and drop multiple assets onto the drug traceability nterface, and the system should process and upload them efficiently	High	Sindhusa
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 wellorganized	Metadata fields should be easily accessible and editable, and changes should be immediately reflected in drug information	Medium	gokul
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	roja
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	yugabharath
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	sindhusa
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	roja

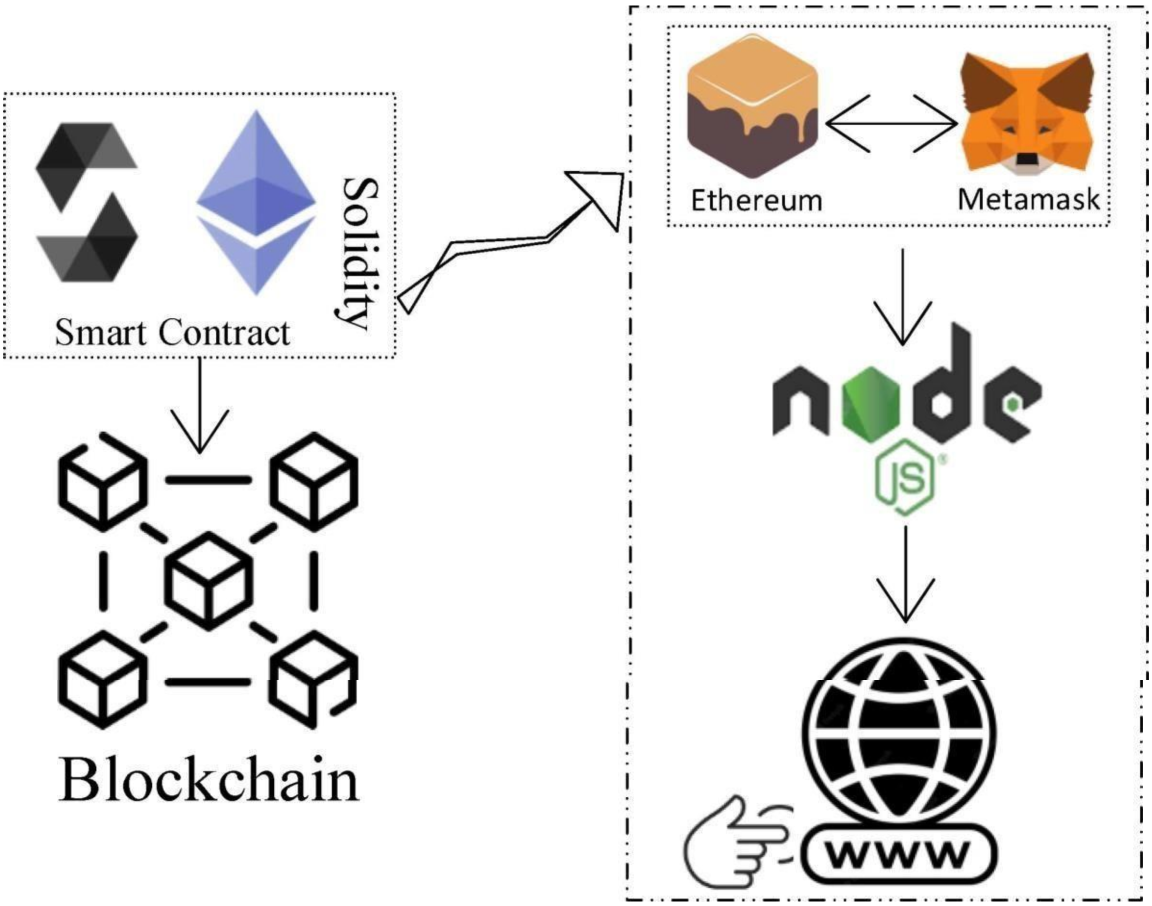
Solution Architecture



5. 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.

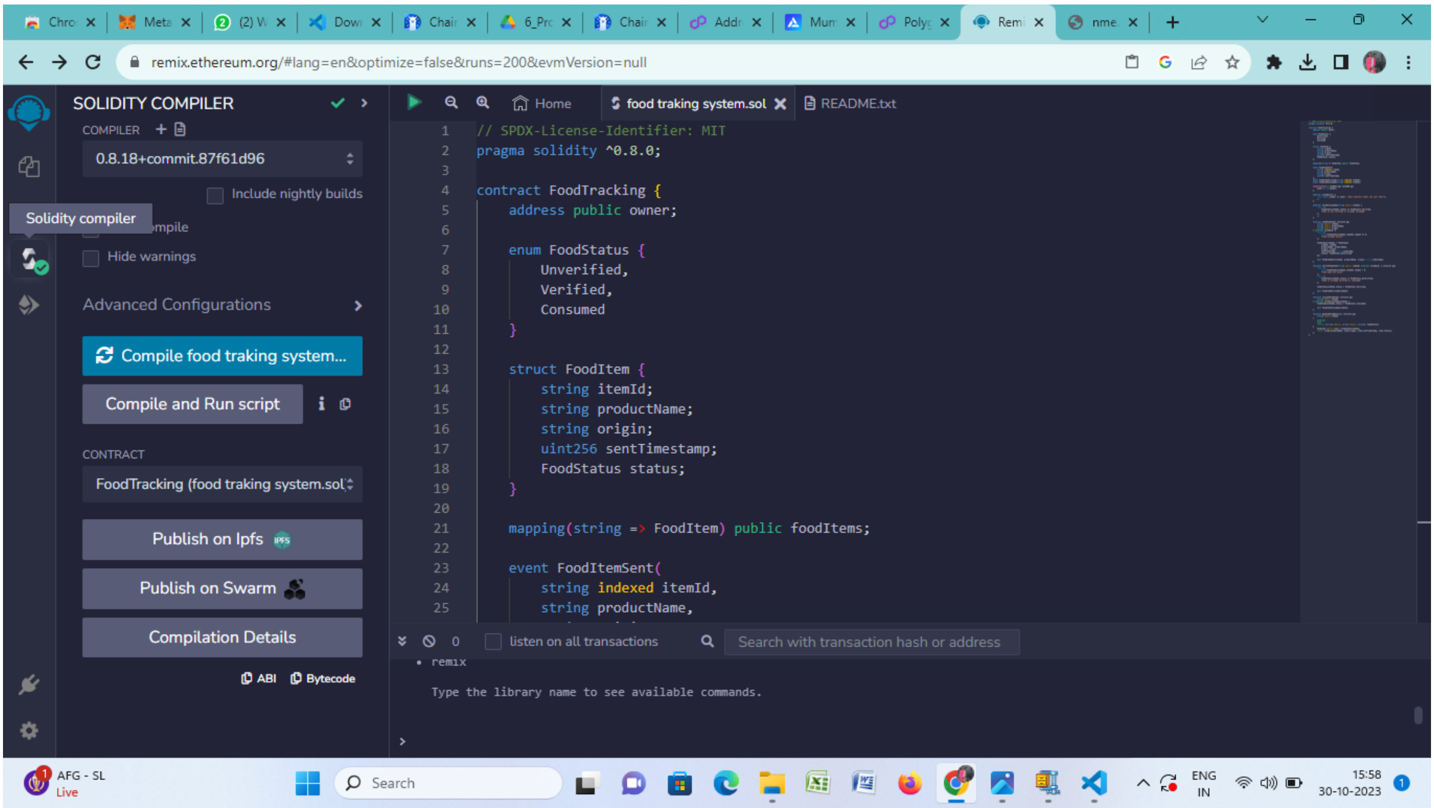
11. Maintain a Release Calendar:

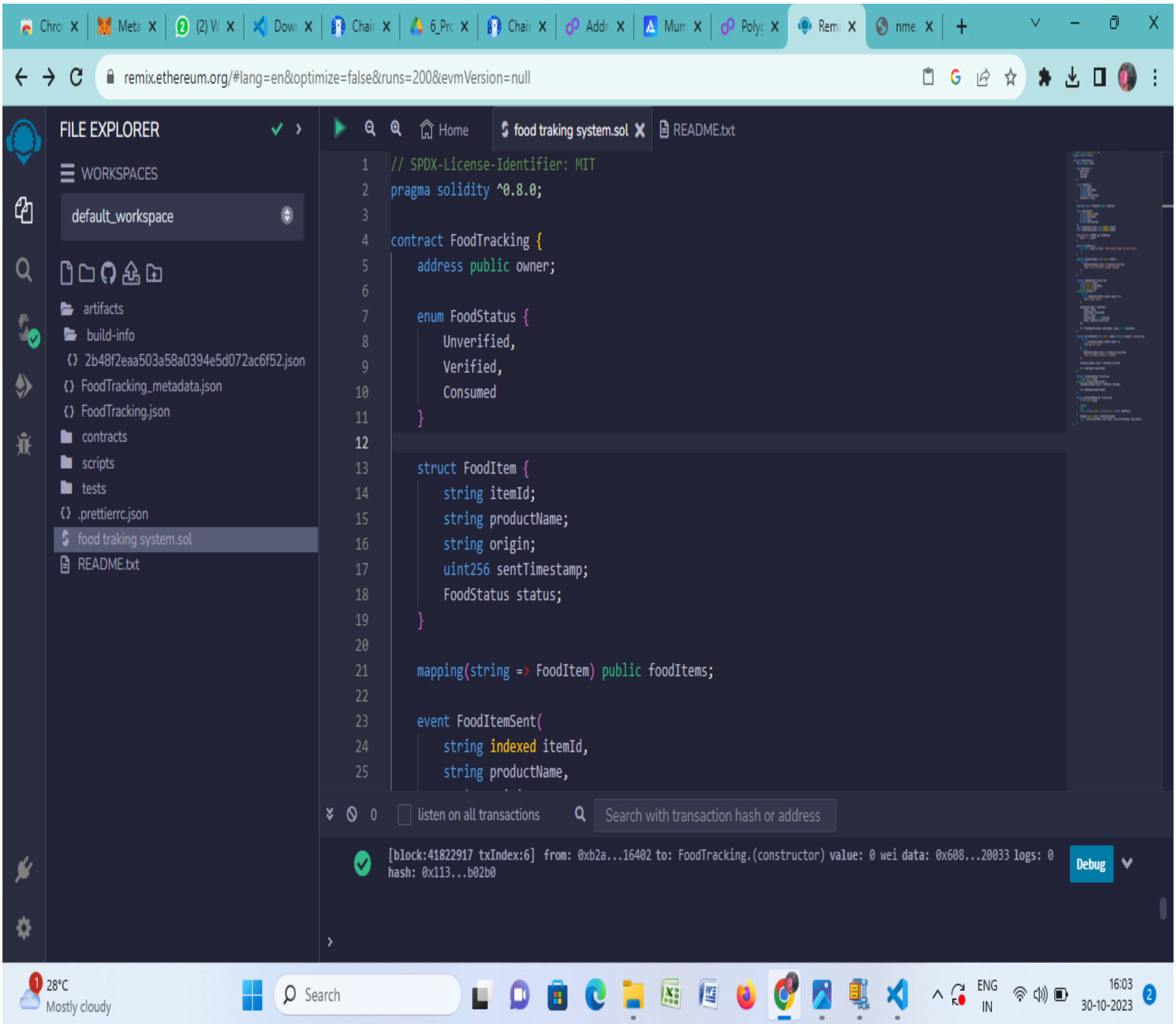
Maintain a release calendar that outlines when each sprint's deliverables or major releases are expected to be available to users or stakeholders. Share this calendar with the team and relevant parties.

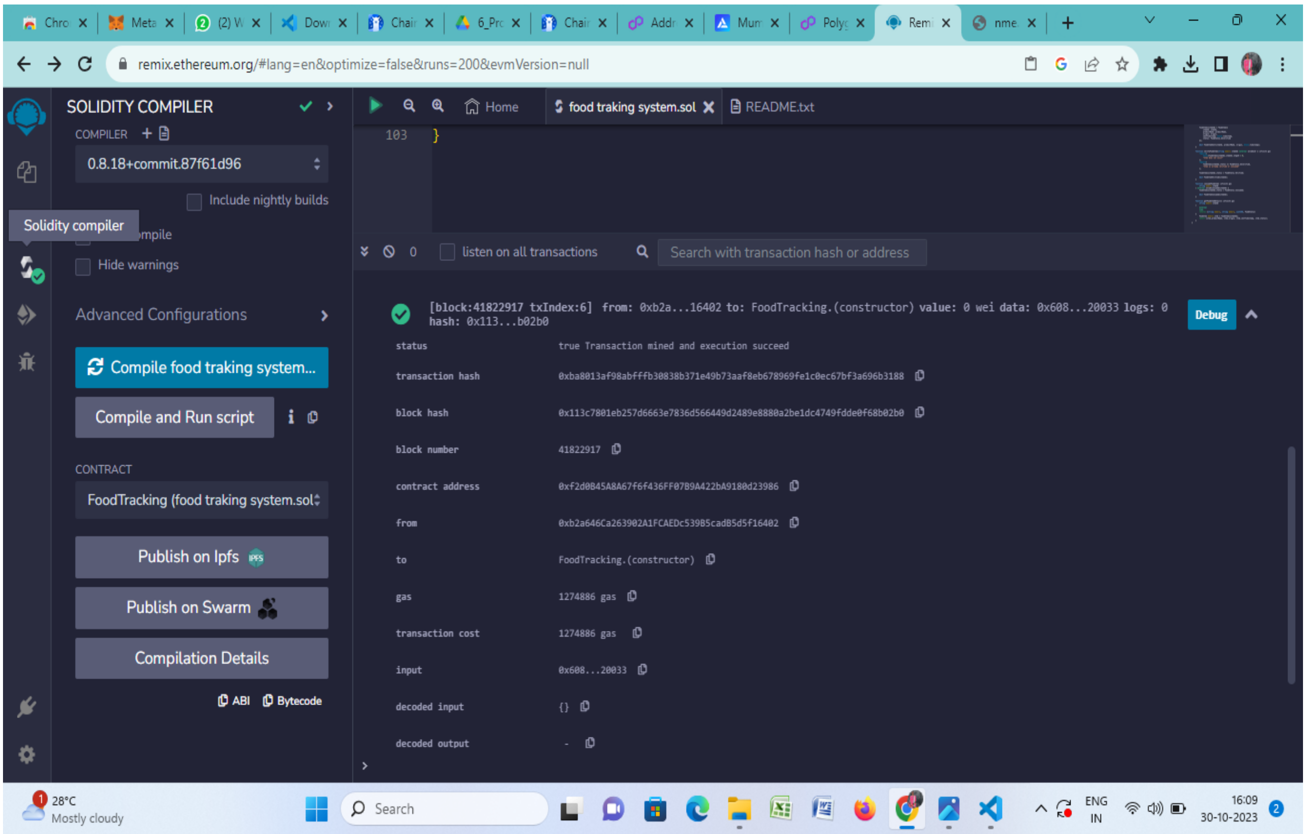
7. CODING & SOLUTIONING (Explain the features added in the projectalong with code)

Feature 1

Smart Contract (Solidity):




remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null
FILE EXPLORER
WORKSPACES
default_workspace
artifacts
build-info
2b48f2eaa503a58a0394e5d072ac6f52.json
FoodTracking_metadata.json
FoodTracking.json
contracts
scripts
tests
.prettierrc.json
food traking system.sol
README.txt
Home
food traking system.sol
README.txt
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;
contract FoodTracking {
address public owner;
enum FoodStatus {
Unverified,
Verified,
Consumed
}
struct FoodItem {
string itemId;
string productName;
string origin;
uint256 sentTimestamp;
FoodStatus status;
}
mapping(string => FoodItem) public foodItems;
event FoodItemSent(
string indexed itemId,
string productName,
listen on all transactions
Search with transaction hash or address
[block:41822917 txIndex:6] from: 0xb2a...16402 to: FoodTracking.(constructor) value: 0 wei data: 0x608...20033 logs: 0
hash: 0x113...b02b0
Debug
28°C
Mostly cloudy
Search
ENG
IN
16:03
30-10-2023


remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null
SOLIDITY COMPILER
COMPILER
0.8.18+commit.87f61d96
Include nightly builds
Solidity compiler
Hide warnings
Advanced Configurations
Compile food traking system...
Compile and Run script
CONTRACT
FoodTracking (food traking system.sol)
Publish on Ipfs
Publish on Swarm
Compilation Details
ABI
Bytecode
Home
food traking system.sol
README.txt
listen on all transactions
Search with transaction hash or address
[block:41822917 txIndex:6] from: 0xb2a...16402 to: FoodTracking.(constructor) value: 0 wei data: 0x608...20033 logs: 0 hash: 0x113...b02b0
Debug
status
true Transaction mined and execution succeed
transaction hash
0xba8013af98abfffb30838b371e49b73aaf8eb678969fe1c0ec67bf3a696b3188
block hash
0x113c7801eb257d6663e7836d566449d2489e8880a2be1dc4749fdde0f68b02b0
block number
41822917
contract address
0xf2d0B45A8A67f6f436FF07B9A422bA9180d23986
from
0xb2a646Ca263902A1FCAEDc539B5cadB5d5f16402
to
FoodTracking.(constructor)
gas
1274886 gas
transaction cost
1274886 gas
input
0x608...20033
decoded input
{}
decoded output
-
28°C
Mostly cloudy
Search
ENG
IN
16:09
30-10-2023

Additional details

This off-chain storage can include a traditional relational database for userprofiles 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 storingcritical 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 suityour 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

Performace 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.

FOOD 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.

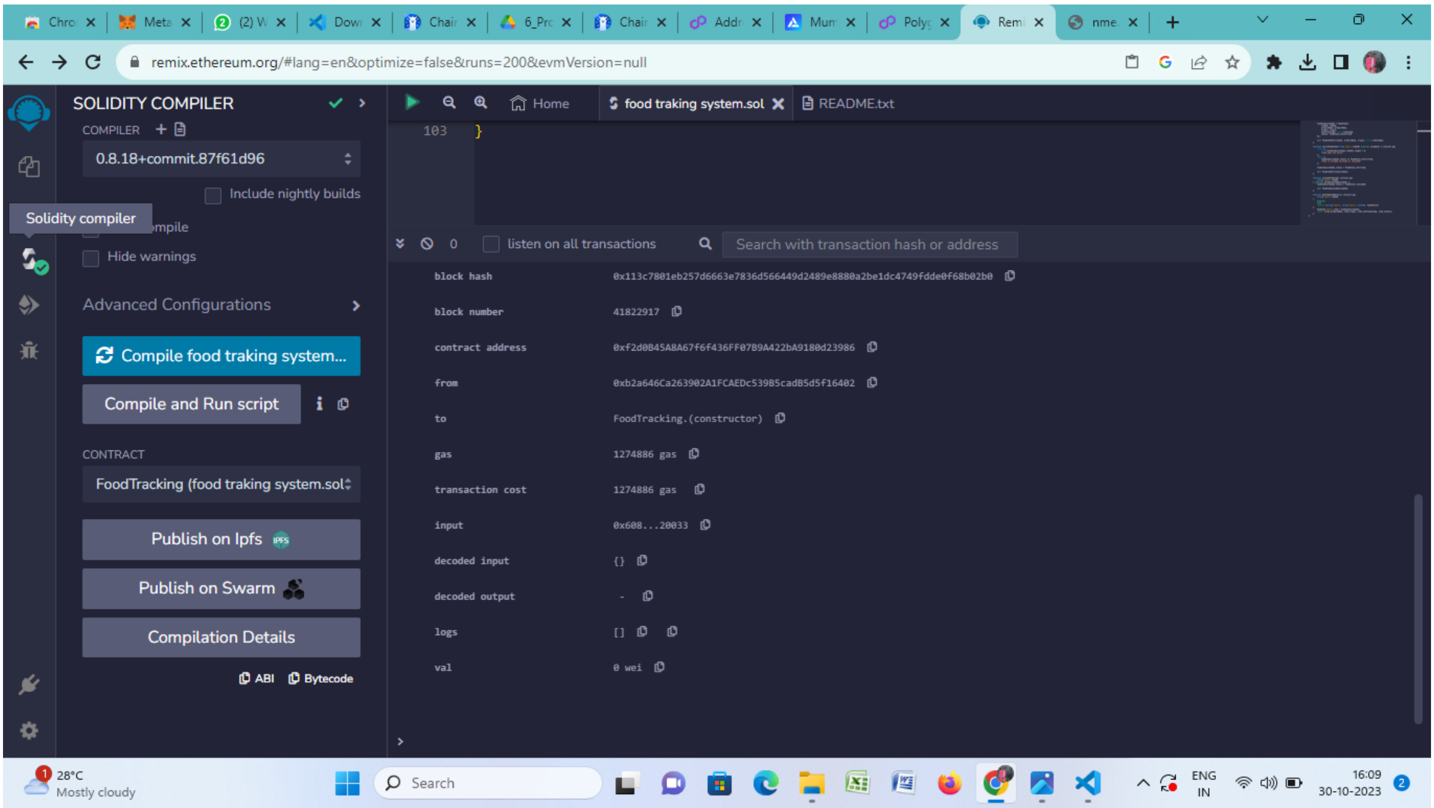
9. RESULTS

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)



Results :

Before using the blockchain-based food tracking system, the performance data of the system were obtained. In this way, it will be necessary to prevent problems such as scalability and to stop the work if it is foreseen that the blockchain-based system to be used will not reach the desired performance values. The performance values of Ethereum and Hyperledger Sawtooth are used to benchmark the values obtained from the proposed system. A simulation environment has been set up to collect and compare these data using Matlab. The latency (s), Net Tx (bytes), Net Rx (bytes), and CPU load (%) values are the variables that keep the data obtained in this simulation environment. With the data obtained in this simulation environment, the aim is to reveal the difference with other platforms clearly and concretely. The latency (s) value in the proposed system was obtained as 0.038. The transmission per second value is 285, the reception per second value is 335, and the CPU load rate value is 19.22. Especially when we evaluate the latency times, the obtained value is at a very good level compared to Ethereum. When it is compared with Hyperledger Sawtooth, it is seen that there is a little more delay. The main reason for this is that the system architecture is more complicated, and the data size obtained is high. This is also evident from the fact that the transmission per second and reception per second values are much higher than Hyperledger Sawtooth. It has been observed that a rate of 19.22 was achieved in the CPU usage rate (Table 4). As a result, it is seen that the performance data obtained have a serious advantage over Ethereum, especially in terms of latency, and it has started to converge in other values (Figure 13). Considering that the real-time operation of the installed system is extremely important, the choice of Hyperledger Fabric has once again emerged as the right decision.

10. CONCLUSION

In this study, the establishment of a blockchain-based food tracking system in Turkey, its performance comparison, the operation of the system, and the results are discussed. The flow of a food tracking system has been demonstrated in Turkey, and accordingly, the 12-step system flow required to develop a blockchain-based food tracking system has been obtained. Comparing the performance data of the established blockchain-based system with other blockchain infrastructures, a value of 0.038 s for latency is 435 times better than Ethereum, one of the most popular blockchain infrastructures. A transmission per second value of 285, reception per second value of 335, and CPU load rate value of 19.22 are obtained with the proposed system. Because it is not currently possible to put such a system into use throughout the country, choosing a pilot region and operating the system in this region and taking their feedback is essential for obtaining solid evidence to show that the users of the system are looking for such a system to use. For this, a survey study was conducted on the users of the system. We can say that the results obtained are concrete proof of how much the system is needed and that it is favored by the public. The system was used for three months in the selected pilot study area. A total of 7828 users viewed the application. A total of 72.03% of them (5560 users) logged into the application and had a user experience. As a result of the two-question survey directed to these participants, 75.31% of the users who use the application like the interface of the application, while the others have low satisfaction. Considering that this developed application is not a commercial product but a proof of concept (PoC) study, it is obvious that there will be some development needs if it is turned into a commercial product.

11. 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).

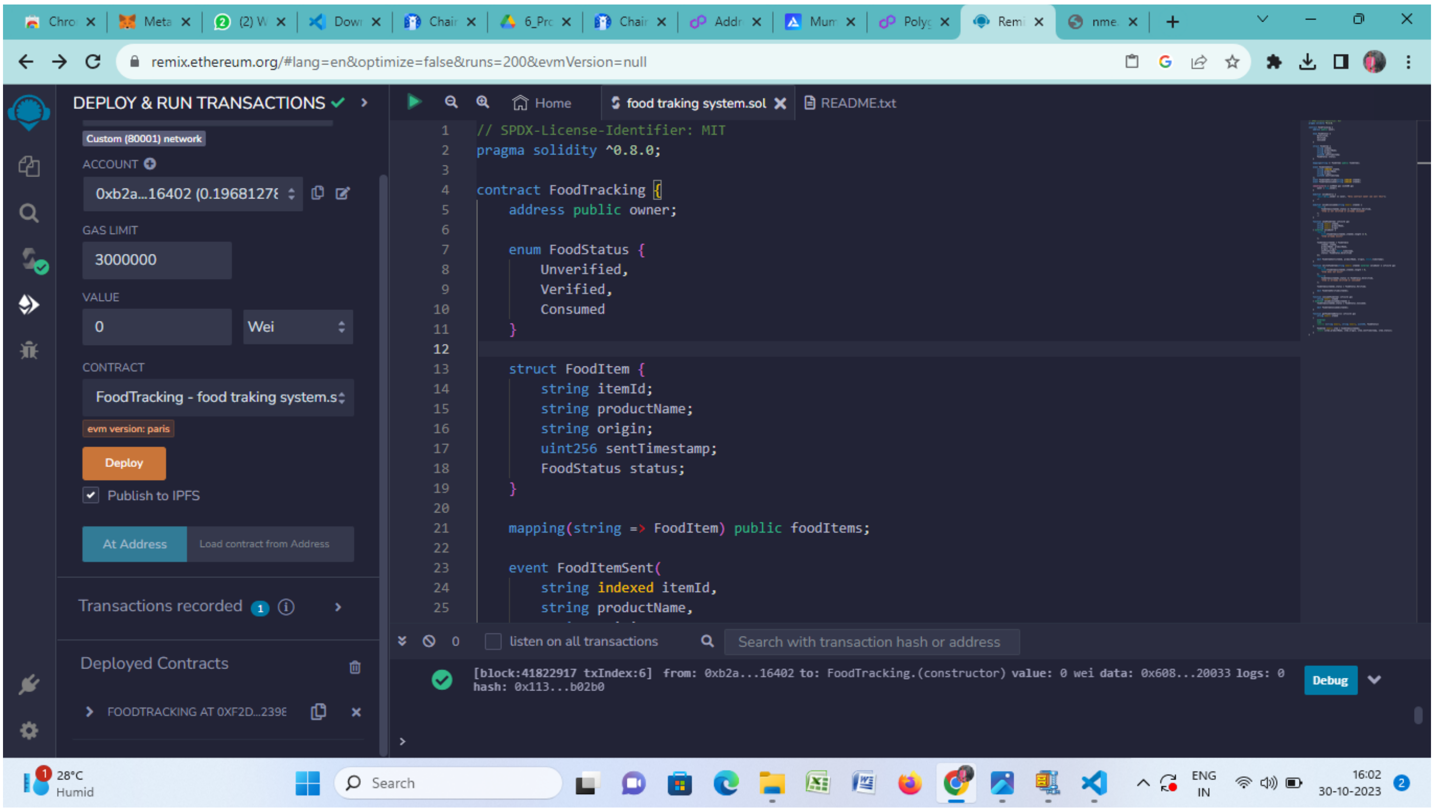
4. Understanding the component of Serialization requirement. As per the initial user specification document composed by SolutionsMax Technology Services (An Enterprise Solutions for the Pharma Ecosystem, 2016-2021) Organizations need to identify various components such as the readiness of the ERP system 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 :



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": "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 = “0xb2a646Ca263902A1FCAEDc539B5cadB5d5f16402"

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/Sindhusa.k/Naan-mudhalvan-

Project Video Demo Link :

https://github.com/sindhusa/Naanmudhalvan/commits?author=Sindhusa.K