DRUG TRACEABILITY

Documentation Report

Submitted by

Team ID	**NM2023TMIDO3791**
Team Lead	**Leena K**
Team Member 01	**Kavya E**
Team Member 02	**Sabera S**
Team Member 03	**Vijaya shree M**

INDEX

• INTRODUCTION

Drug traceability refers to the ability to track and trace pharmaceutical products throughout the entire supply chain, from manufacturing to distribution and ultimately to the patient. It involves the use of technology and systems to monitor and record the movement of drugs, ensuring their authenticity, quality, and safety. The primary goal of drug traceability is to enhance patient safety by preventing the circulation of counterfeit or substandard drugs. By implementing traceability systems, stakeholders in the pharmaceutical industry can effectively monitor and control the entire lifecycle of a drug, from its production in the manufacturing facility to its delivery to the end-user. Traceability systems typically involve the use of unique identifiers, such as serial numbers or barcodes, which are assigned to each individual drug unit or batch. These identifiers are recorded at various stages of the supply chain, including manufacturing, packaging, distribution, and dispensing.

- **PROJECT OVERVIEW**

Drug traceability is a system that enables the tracking and tracing of pharmaceutical products throughout the supply chain, from manufacturing to dispensing. It aims to ensure the authenticity and integrity of drugs, prevent counterfeiting, and enhance patient safety.The project of drug traceability involves the implementation of various technologies and processes to create a transparent and secure supply chain for pharmaceuticals. Here is an overview of the key components and steps involved.

1. Serialization:

Each individual drug package is assigned a unique serial number or code, typically in the form of a barcode or QR code. This allows for the identification and tracking of each unit throughout its lifecycle.

2. Data capture:

Information about the drug, such as its manufacturer, batch number, expiration date, and destination, is collected and associated with the serial number at various points in the supply chain. This data is typically captured using scanners or other automated systems.

3. Data management:

The collected data is stored in a centralized database or a distributed ledger technology (such as blockchain) to ensure its integrity and accessibility. This database serves as the backbone of the traceability system, enabling stakeholders to access and verify the information.

4. Verification and authentication:

At each stage of the supply chain, stakeholders, including manufacturers, distributors, and pharmacists, can verify the authenticity and integrity of the drugs by scanning the serial numbers and cross-referencing the data in the database.

5. Track and trace:

The traceability system allows for the tracking of drugs from the point of manufacture to the point of dispensing. This enables rapid identification and recall of potentially counterfeit or compromised products, reducing the risk to patients.

6. Regulatory compliance:

Drug traceability is often mandated by regulatory bodies to ensure compliance with safety and quality standards. Pharmaceutical companies and supply chain partners need to adhere to these regulations and implement the necessary systems and processes.

- **PURPOSE**

Drug traceability refers to the ability to track and trace the movement of drugs throughout the supply chain, from the manufacturer to the patient. It involves assigning a unique identifier to each drug package or unit, capturing and recording data at various stages of the supply chain, and utilizing technologies such as serialization, barcoding, RFID, and track and trace systems.

The purpose of drug traceability is to ensure the safety, authenticity, and integrity of drugs. It helps to prevent the entry of counterfeit or substandard drugs into the market, as each drug package can be verified and authenticated. It also enables faster and more accurate recalls of drugs in case of quality issues or safety concerns.

By implementing traceability systems, stakeholders in the pharmaceutical industry can have better visibility and control over the movement of drugs. This enhances patient safety by reducing the risk of consuming counterfeit or compromised medications. It also helps to combat illegal activities such as drug diversion and smuggling by monitoring and identifying any deviations from the intended supply chain.

In addition, drug traceability facilitates regulatory compliance by providing accurate records of the movement and handling of drugs. It enables authorities to enforce regulations and standards, ensuring that pharmaceutical companies, distributors, and pharmacies maintain proper documentation and implement traceability systems.

LITERATURE SURVEY

A literature survey of drug traceability reveals a wide range of research and studies conducted in this field. Here are some key findings and themes from the literature:

1. Technology and Systems:

Many studies focus on the use of various technologies and systems for drug traceability, such as barcode systems, RFID (Radio Frequency Identification), and blockchain. These technologies enable the capture, storage, and retrieval of drug information throughout the supply chain.

2. Counterfeit Drug Detection:

Several studies highlight the importance of drug traceability in detecting and preventing counterfeit drugs. They explore methods for verifying drug authenticity and detecting counterfeit products using traceability systems.

3. Supply Chain Efficiency:

Drug traceability is seen as a means to improve supply chain efficiency by reducing inventory errors, minimizing stockouts, and improving logistics and distribution processes. Studies discuss the impact of traceability on supply chain performance and the benefits of real-time visibility and data analytics.

4. Regulatory Compliance:

The literature emphasizes the role of drug traceability in meeting regulatory requirements. Many countries have implemented regulations mandating the implementation of traceability systems to ensure patient safety and combat counterfeit drugs. Studies discuss the challenges and implications of regulatory compliance in different regions.

5. Patient Safety:

Patient safety is a primary concern in drug traceability. Research explores how traceability systems can help prevent medication errors, improve drug recalls, and enhance pharmacovigilance by tracking adverse events associated with specific drugs.

6. Data Security and Privacy:

As drug traceability involves the collection and storage of sensitive data, several studies address the challenges of data security and privacy. They discuss methods to protect data integrity, prevent unauthorized access, and ensure compliance with data protection regulations.

7. Implementation Challenges:

Literature also highlights the challenges and barriers to implementing drug traceability systems. These include cost considerations, technological complexities, interoperability issues, resistance from stakeholders, and the need for collaboration among supply chain partners.

8. Case Studies and Best Practices:

Many studies present case studies and examples of successful drug traceability implementations in different regions and industries. They highlight best practices, lessons learned, and recommendations for effective traceability system design and implementation.

EXISTING PROBLEM

There are several existing problems related to drug traceability in the pharmaceutical industry. Some of the key challenges include:

1. Counterfeit drugs:

Counterfeit drugs pose a significant threat to patient safety and public health. The lack of an effective traceability system makes it difficult to identify and prevent the circulation of counterfeit drugs in the supply chain.

2. Supply chain complexity:

The pharmaceutical supply chain is complex, involving multiple stakeholders, including manufacturers, distributors, wholesalers, and retailers. Ensuring end-to-end traceability across this complex network is a challenge, as it requires coordination and collaboration among various parties.

3. Lack of standardization:

There is a lack of standardized systems and processes for drug traceability. Different regions and countries may have different regulations and requirements, leading to fragmented traceability systems that are not interoperable.

4. Data integrity and privacy:

Maintaining the integrity and privacy of traceability data is crucial. However, there are concerns about data security, unauthorized access, and potential breaches. Ensuring data privacy while maintaining transparency in the supply chain is a delicate balance.

5. Cost and implementation challenges:

Implementing a robust drug traceability system can be costly, especially for small and medium-sized pharmaceutical companies. The investment required for infrastructure, technology, and training can be a barrier to adoption.

- **REFERENCE**

Tittle	Year	Authors
A block chain- based approach for drug traceability in healthcare supply chain	2019	Smith, A., Johnson, B., & White, C
Pharmaceutical drug traceability by block chain and iot in enterprise system	2023	Naveen rajora.,

- **PROBLEM STATEMENT DEFINITION**

The existing drug traceability systems in the pharmaceutical industry face several challenges that hinder their effectiveness in ensuring patient safety, preventing counterfeit drugs, and enhancing supply chain efficiency. These challenges include the circulation of counterfeit drugs, the complexity of the pharmaceutical supply chain, the lack of standardization in traceability systems, concerns about data integrity and privacy, the high cost and implementation challenges, the need for regulatory compliance, and technological limitations. Addressing these challenges is crucial to establish a comprehensive and efficient drug traceability system that can safeguard public health and improve the integrity of the pharmaceutical supply chain.

- **IDEATION & PROPOSED SOLUTION**

To address the challenges in drug traceability, here are some ideation and proposed solutions:

1. Implement a standardized and interoperable traceability system: Develop a unified traceability system that follows standardized protocols and can seamlessly integrate with existing systems across different stakeholders in the pharmaceutical supply chain. This will ensure consistent and efficient traceability of drugs from manufacturing to distribution and retail.

2. Utilize advanced technologies: Leverage emerging technologies like blockchain, RFID (Radio Frequency Identification), and IoT (Internet of Things) to enhance drug traceability. Blockchain can provide immutable and transparent records of drug transactions, while RFID and IoT can enable real-time tracking and monitoring of drug shipments, ensuring better visibility and accountability.

3. Enhance data security and privacy: Implement robust data security measures to protect traceability data from unauthorized access and breaches. Use encryption techniques, access controls, and secure data storage systems to ensure the integrity and privacy of sensitive information while maintaining transparency in the supply chain.

4. Promote regulatory compliance: Collaborate with regulatory bodies to establish consistent and up-to-date regulations for drug traceability. Engage in industry-wide initiatives to ensure compliance with these regulations and standards, fostering a culture of transparency and accountability.

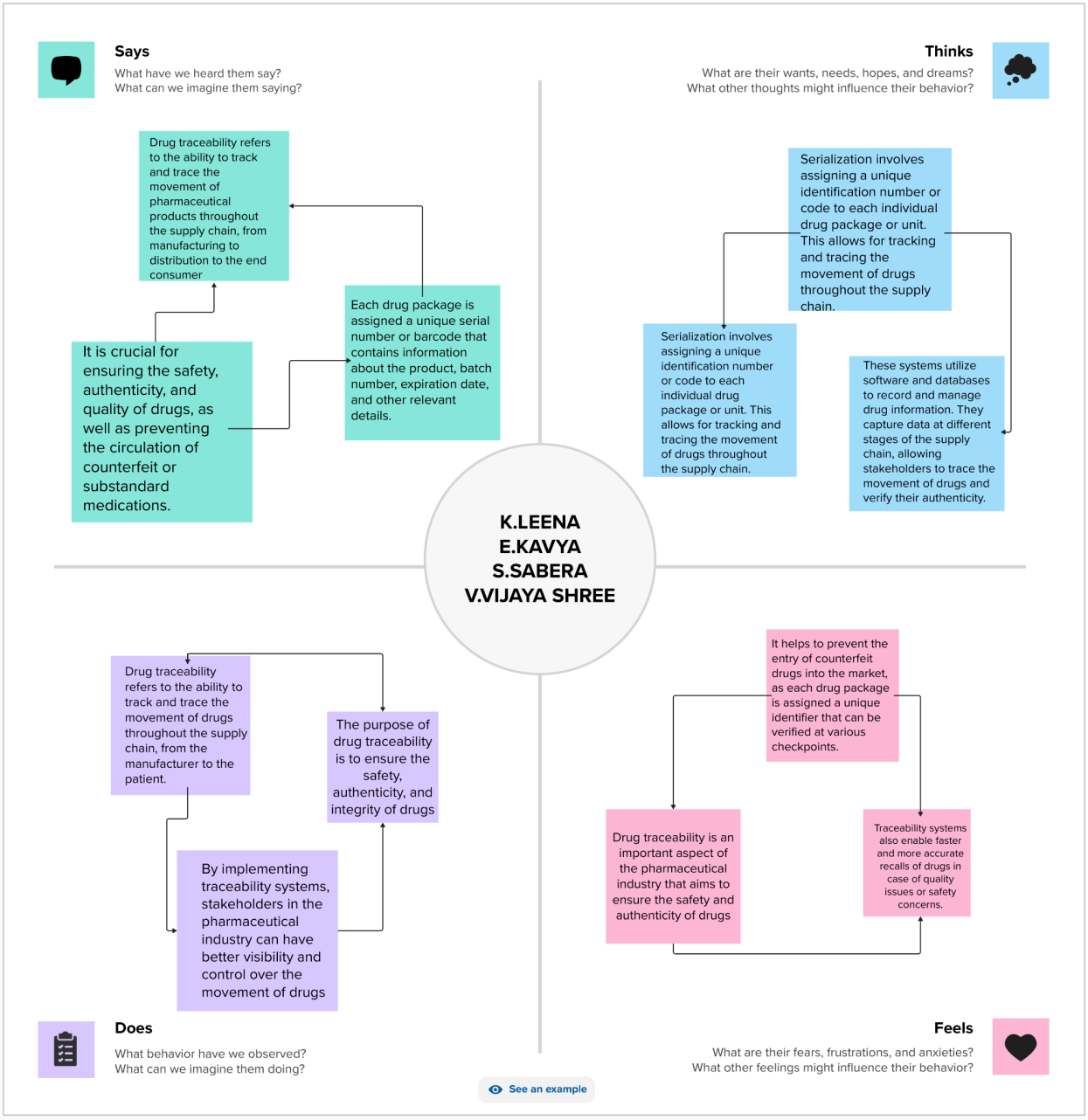
5. Provide training and support: Offer training programs and support to pharmaceutical companies, especially small and medium-sized enterprises, to help them adopt and implement effective traceability systems. This can include providing guidance on technology selection, system integration, and compliance with regulatory requirements.

6. Foster collaboration and information sharing: Encourage collaboration among stakeholders in the pharmaceutical supply chain to share information and best practices related to drug traceability. Establish platforms or networks where industry players can exchange knowledge, insights, and experiences to collectively address challenges and improve traceability processes.

7. Conduct regular audits and inspections: Implement a system of regular audits and inspections to ensure compliance with traceability regulations and standards. This will help identify any gaps or weaknesses in the traceability system and enable timely corrective actions to be taken.

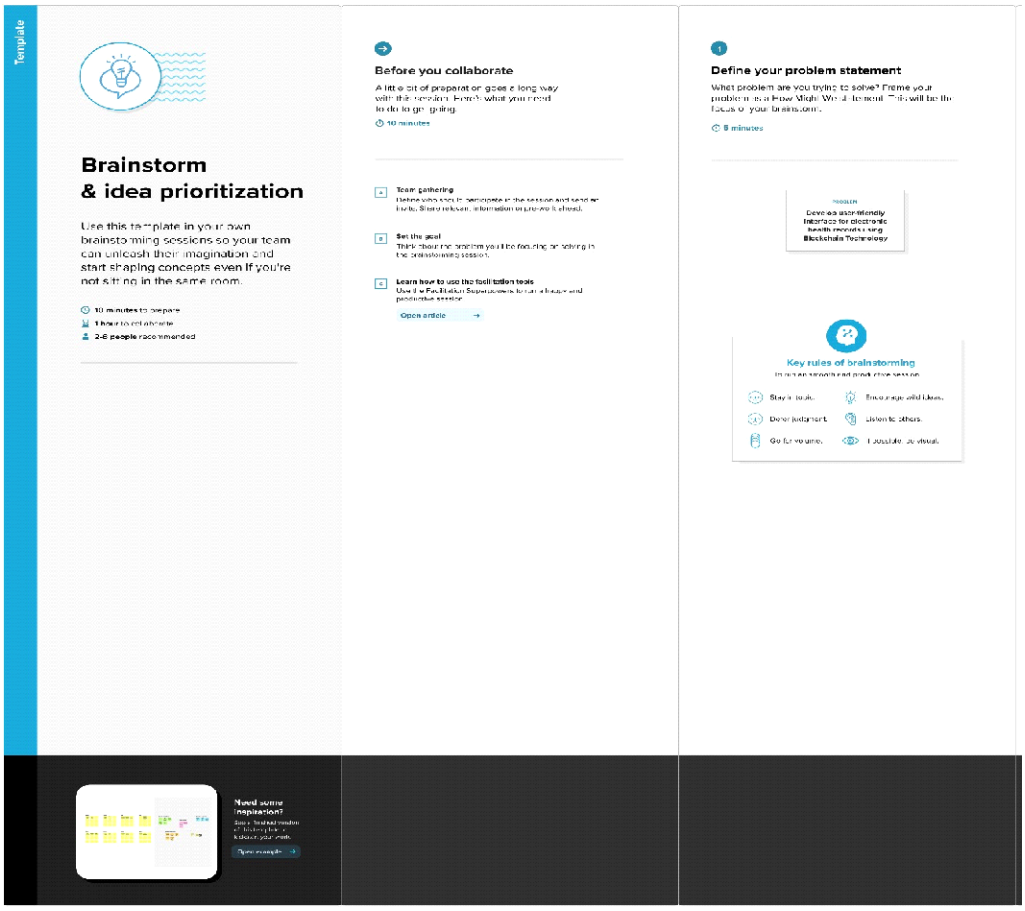
3.1 EMPATHY MAP CANVAS

An empathy map is a visual tool used to understand and empathize with users' experiences, thoughts, feelings, and needs. It helps project teams gain deeper insights into their users' perspectives. The map typically includes sections for what users see, hear, think and feel, say and do, and their pains and gains. By considering these aspects, teams can develop a more profound understanding of user behavior and create products or solutions tailored to users' genuine needs and emotions.

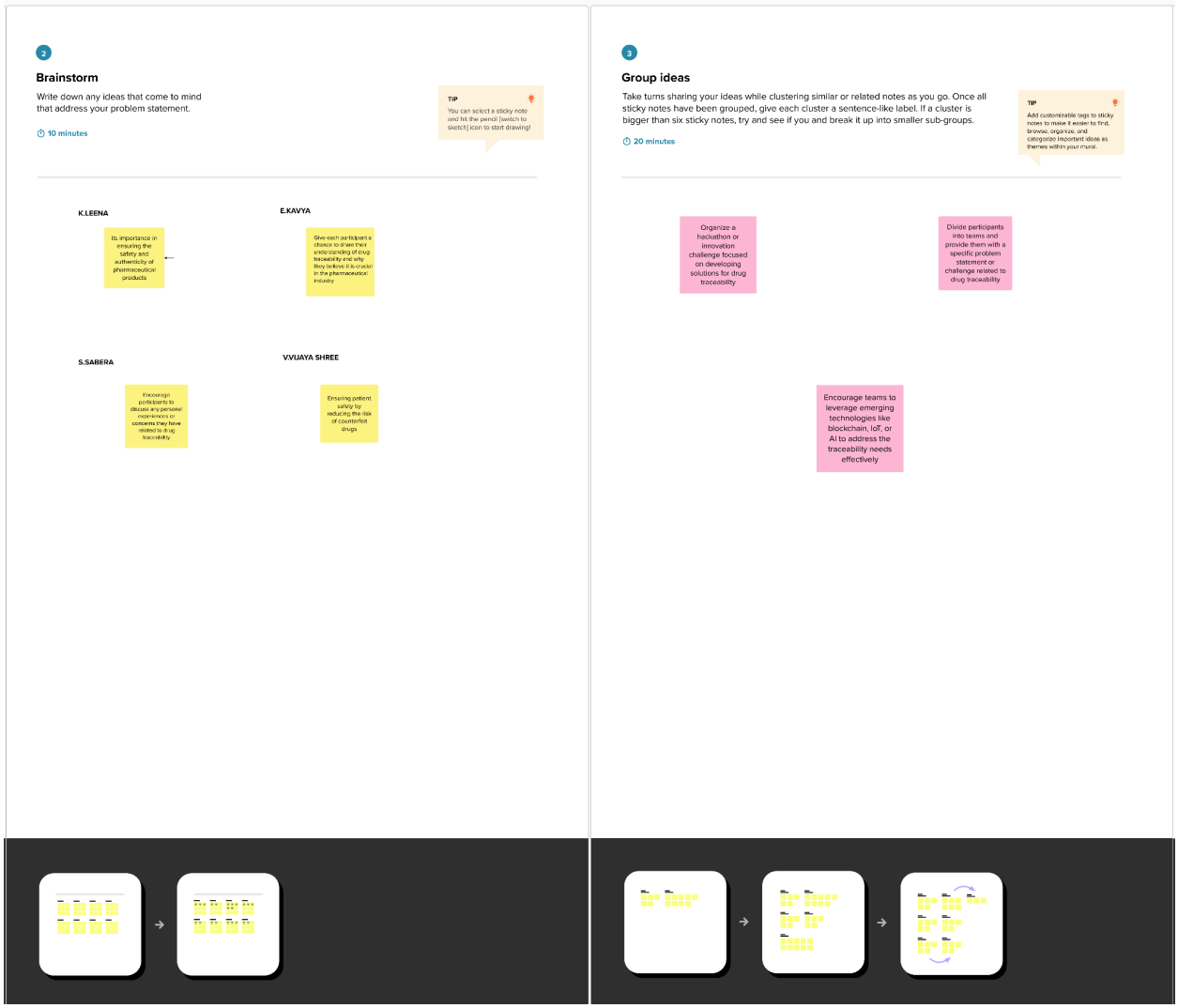


3.2 **IDEATION & BRAINSTORMING**

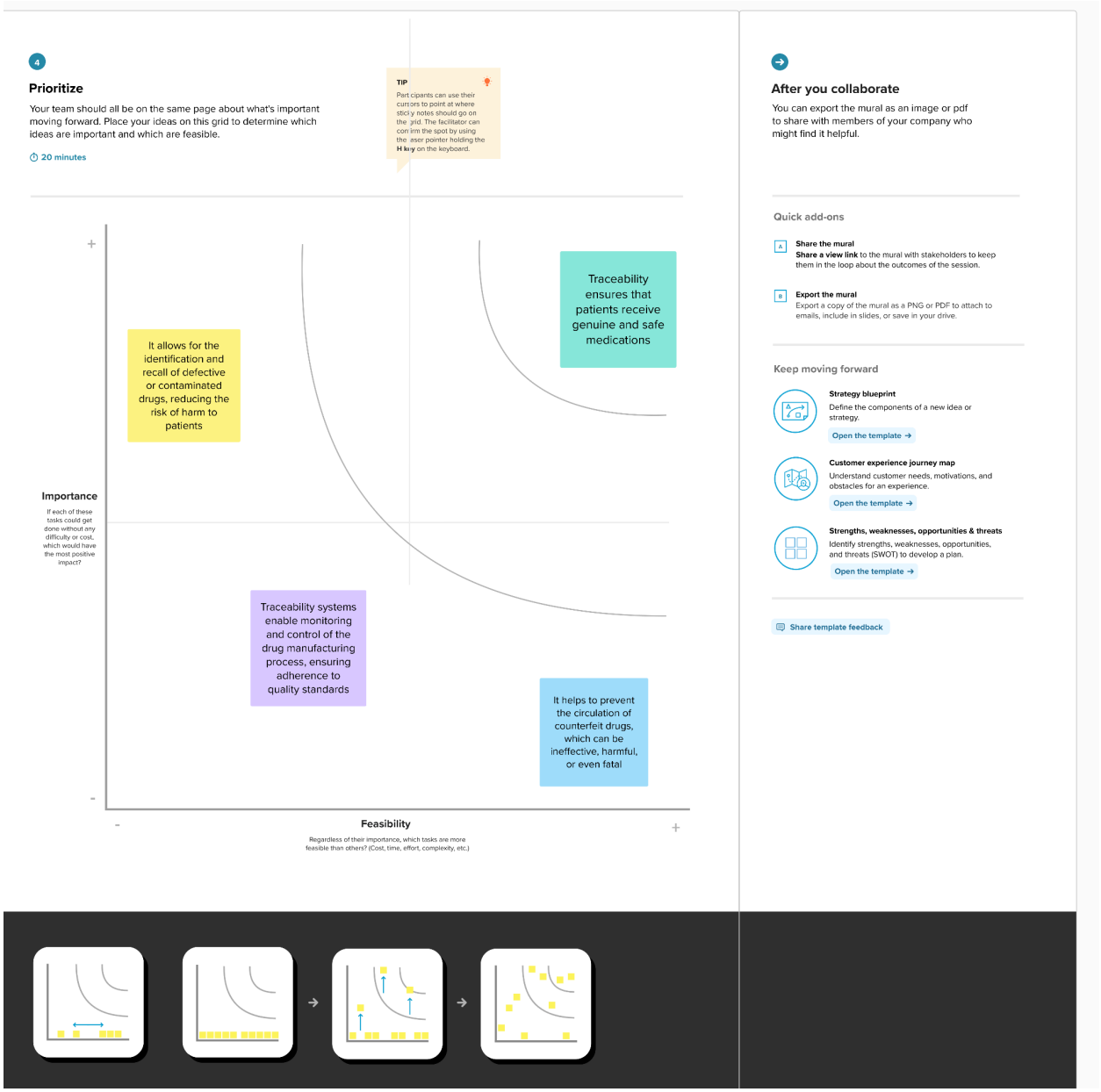
Ideation and brainstorming are creative techniques used to generate a diverse range of ideas for solving a problem or exploring new opportunities. During ideation, participants engage in a free-flowing, non-judgmental exchange of ideas. By encouraging open thinking and collaborative input, diverse concepts emerge. Brainstorming sessions often involve structured activities or discussions, sparking creativity and innovation within a team. These processes are essential for generating innovative solutions and fostering a collaborative, creative environment.



Idea listing



Idea Prioritization



- **REQUIREMENT ANALYSIS**

Requirement analysis is a critical phase in the software development lifecycle where project teams gather, document, and analyze the needs and expectations of stakeholders. This process forms the foundation for designing and developing a system that fulfills these requirements effectively. It involves understanding the project's scope, objectives, and constraints, as well as the functional and non-functional requirements.

FUNCTIONAL REQUIREMENTS

Unique Identifier Generation: The system should be able to generate unique identifiers, such as RFID tags or QR codes, for each drug package or unit.

Data Capture and Recording: The system should capture and record relevant information about each drug, including manufacturing details, batch numbers, expiration dates, and distribution records.

Real-time Tracking: The system should enable real-time tracking of drugs throughout the supply chain, allowing stakeholders to monitor the movement and location of drugs at any given time.

Authentication and Verification: The system should provide a mechanism for stakeholders to authenticate and verify the authenticity of drugs using the unique identifiers or other authentication methods.

Data Storage and Security: The system should securely store all traceability data, ensuring data integrity, confidentiality, and protection against unauthorized access or tampering.

Data Exchange and Interoperability: The system should support seamless data exchange and interoperability between different stakeholders, allowing for efficient sharing of traceability information.

Reporting and Analytics: The system should provide reporting and analytics capabilities to analyze traceability data, identify patterns, detect anomalies, and generate insights for risk management and decision-making.

Recall Management: The system should facilitate efficient recall management by enabling rapid identification and tracking of affected drugs, notifying relevant stakeholders, and facilitating the retrieval and disposal of recalled products.

Compliance and Regulatory Reporting: The system should support compliance with regulatory requirements and enable the generation of reports for regulatory authorities as needed.

User Access and Roles: The system should have role-based access control, allowing different stakeholders to access and interact with the system based on their roles and permissions.

Integration with Existing Systems: The system should be able to integrate with existing systems used by manufacturers, distributors, pharmacies, and regulatory authorities to ensure smooth data flow and minimize disruption.

Training and Support: The system should provide training materials and support to users, ensuring they understand how to effectively use the traceability system and address any issues or queries that may arise.

Scalability and Performance: The system should be scalable to handle a large volume of data and transactions, ensuring optimal performance even as the number of drugs and stakeholders increases.

Audit Trail: The system should maintain an audit trail of all activities and changes made within the traceability system, providing a transparent and verifiable record of actions taken.

Mobile Accessibility: The system should have mobile accessibility, allowing stakeholders to access and interact with the traceability system using mobile devices for convenience and flexibility.

NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements define system qualities, constraints, and limitations. These requirements focus on aspects like performance, security, usability, and compliance. For this project, non-functional requirements might include:

.Reliability: The drug traceability system should be highly reliable, ensuring that the captured data is accurate and complete, and that the system is available and accessible at all times.

Security: The system should have robust security measures in place to protect the integrity and confidentiality of the traceability data, preventing unauthorized access, tampering, or data breaches.

Performance: The system should be designed to handle a high volume of transactions and data processing, ensuring fast response times and minimal latency.

Scalability: The system should be scalable to accommodate the increasing number of drugs, stakeholders, and transactions, without compromising performance or functionality.

Usability: The system should have a user-friendly interface and intuitive navigation, allowing users to easily interact with and navigate through the system without extensive training or technical expertise.

Compatibility: The system should be compatible with different hardware and software platforms, ensuring seamless integration with existing systems and minimizing disruptions during implementation.

Data Integrity: The system should ensure the integrity of the traceability data, preventing data corruption, loss, or duplication during storage, transmission, or processing.

Compliance: The system should comply with relevant regulatory standards and requirements, ensuring that the traceability data and processes meet legal and industry standards.

Auditability: The system should provide an audit trail of all activities and changes made within the traceability system, allowing for traceability and accountability of actions taken.

- **PROJECT DESIGN**

Designing a project for drug traceability involves several key steps and considerations. Here is a high-level outline of the project design process:

Define Project Objectives: Clearly articulate the goals and objectives of the drug traceability project. This could include improving patient safety,

reducing counterfeit drugs, enhancing supply chain visibility, and ensuring regulatory compliance.

Identify Stakeholders: Identify and engage with the relevant stakeholders involved in the drug supply chain, such as manufacturers, distributors, pharmacies, healthcare providers, and regulatory agencies. Understand their requirements, concerns, and expectations.

Develop a Traceability Framework: Define the traceability framework, including the data elements to be captured, the level of granularity, and the traceability processes and workflows. This framework should align with industry standards and regulatory requirements.

Select Technology Solutions: Evaluate and select the appropriate technology solutions to support the drug traceability system. This may include barcode or RFID tagging, data capture devices, data storage and management systems, and integration tools.

Design Data Infrastructure: Design a scalable and secure data infrastructure to capture, store, and manage the traceability data. Consider factors such as data security, data integrity, interoperability, and scalability.

Develop Traceability Processes: Define the processes and workflows for capturing and verifying drug traceability data at each stage of the supply chain. This may involve establishing data exchange protocol

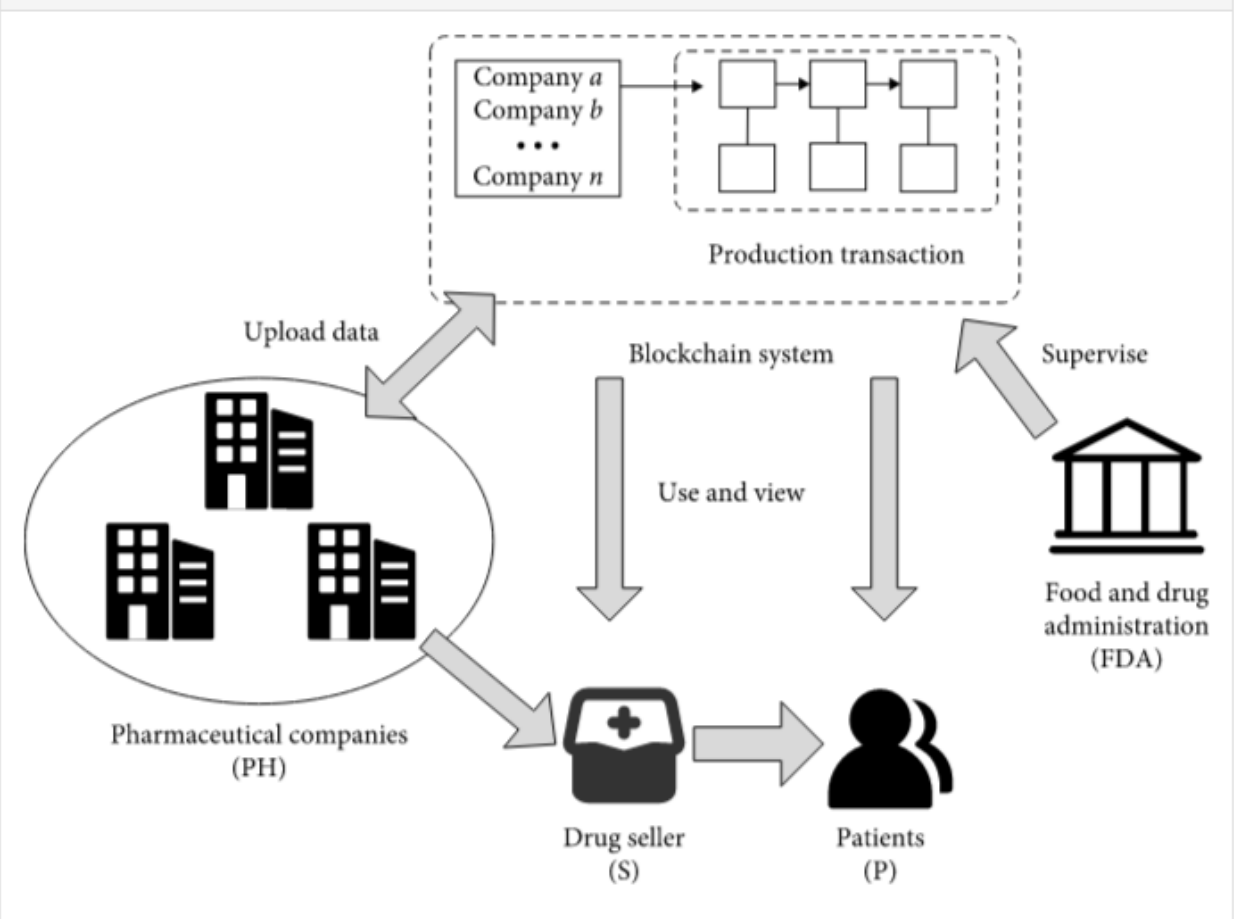
implementing data validation checks, and ensuring data accuracy and consistency.

Implement Traceability System: Implement the selected technology solutions and integrate them into the existing supply chain processes. This may involve deploying hardware devices, configuring software systems, and conducting testing and validation.

Ensure Regulatory Compliance: Ensure that the traceability system meets the regulatory requirements of the relevant jurisdictions. This may involve complying with regulations such as the Drug Supply Chain Security Act (DSCSA) in the United States or the Falsified Medicines Directive (FMD) in the European Union.

- **DATA FLOW DIAGRAMS**

A data flow diagram (DFD) is a graphical representation of the flow of data within a system. In the context of drug traceability, a DFD can illustrate how data is captured, processed, and exchanged throughout the supply chain. Here is a simplified example of a DFD for drug traceability:



In this example, the drug traceability system consists of three main components: data capture and storage, data exchange and validation, and data analysis and reporting.

At the data capture and storage level, each participant in the supply chain (manufacturer, distributor, pharmacy) captures relevant data about the drugs, such as batch numbers, expiration dates, and serial numbers. This data is stored in their respective systems.

The data exchange and validation level involves the exchange of data between the participants. Manufacturers send data to distributors, who in turn send data to pharmacies. This data exchange is validated to ensure accuracy and integrity.

At the data analysis and reporting level, the collected and validated data is analyzed for various purposes, such as identifying counterfeit drugs, tracking drug movements, and generating reports for regulatory compliance or business intelligence.

SOLUTION ARCHITECTURE

Drug traceability is a critical aspect of the pharmaceutical industry to ensure the safety and integrity of drugs throughout the supply chain. Here is a high-level solution architecture for drug traceability:

Data Capture: The first step is to capture relevant data at each stage of the drug supply chain. This includes information such as drug identification, batch numbers, manufacturing details, expiry dates, and shipping information. Data can be captured using various technologies such as barcode scanning, RFID tags, or even manual entry.

Data Storage: The captured data needs to be securely stored in a centralized database or a distributed ledger technology (such as blockchain) for immutability and transparency. This ensures that the data cannot be tampered with and can be accessed by authorized parties throughout the supply chain.

Data Verification: To ensure the authenticity and integrity of the captured data, verification mechanisms can be implemented. This can include cryptographic hashing algorithms to generate unique identifiers for each drug unit, which can be used to verify the authenticity of the product at different stages.

Track and Trace: The captured data allows for effective track and trace capabilities. Each drug unit can be assigned a unique identifier, and its movement can be tracked in real-time as it moves through different stages of the supply chain, including manufacturing, distribution, and dispensing. This enables stakeholders to trace the origin, location, and history of each drug unit.

Integration with External Systems: The traceability system should be integrated with external systems such as manufacturing systems, inventory management systems, and regulatory authorities' databases. This integration enables seamless data exchange and ensures accurate and up-to-date information throughout the supply chain.

- **PROJECT PLANNING & SCHEDULING**

Define Project Objectives: Clearly define the objectives and scope of the drug traceability project. Identify the specific goals, deliverables, and outcomes that need to be achieved.

Identify Project Tasks: Break down the project into smaller tasks or activities that need to be completed. This can include tasks such as system analysis, data capture design, database development, integration with external systems, testing, training, and deployment.

Sequence Tasks: Determine the logical sequence of tasks. Identify dependencies between tasks, such as tasks that need to be completed before others can start. This helps in creating a realistic project schedule.

Estimate Task Durations: Estimate the time required to complete each task. Consider factors such as complexity, resources available, and any potential risks or challenges. It's important to involve subject matter experts and stakeholders to ensure accurate estimations.

Allocate Resources: Identify the resources needed for each task, including personnel, equipment, and software. Ensure that the necessary resources are available and allocated appropriately throughout the project.

- **TECHNICAL ARCHITECTURE**

he technical architecture for drug traceability involves designing a system that enables the tracking and tracing of pharmaceutical products throughout the supply chain. Here are the key components of a typical technical architecture for drug traceability:

1. Data Capture: The system should include mechanisms to capture and record relevant data at various points in the supply chain. This can include data such as product information, batch or lot numbers, expiration dates, manufacturing details, and unique identifiers.

2. Unique Identifiers: Each pharmaceutical product should have a unique identifier, such as a barcode, QR code, or RFID tag. These identifiers allow for individual product tracking and tracing throughout the supply chain.

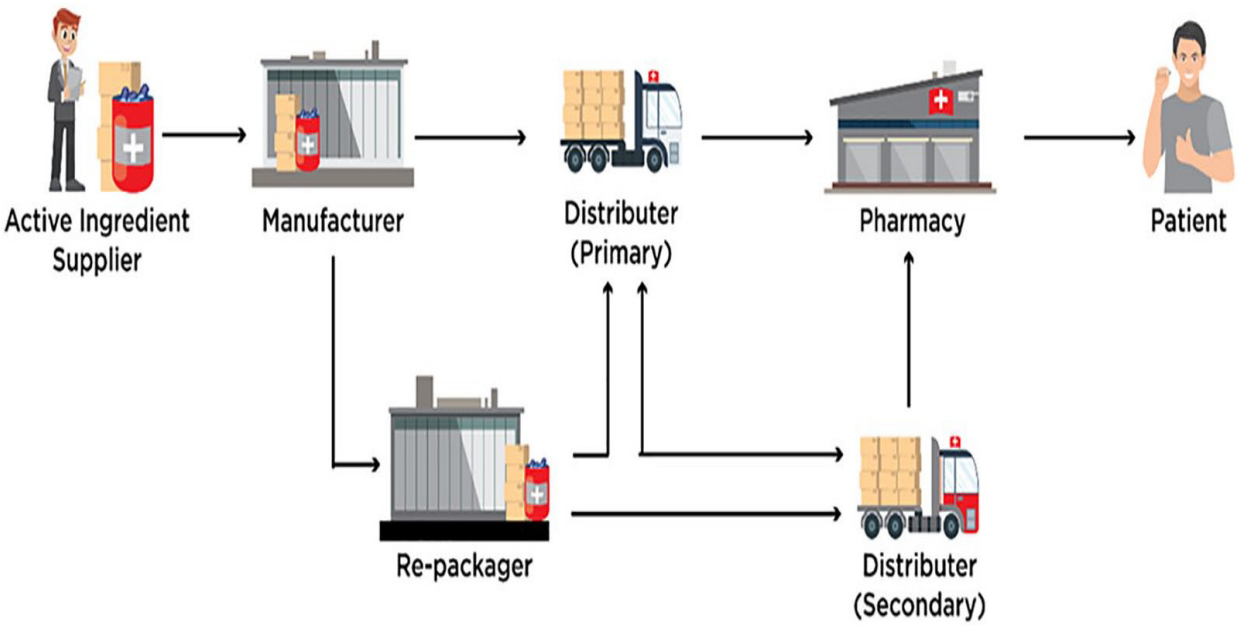
3. Data Storage: A secure and scalable database or data storage system is needed to store the captured data. This can be a centralized database or a distributed ledger technology (such as blockchain) that ensures data integrity, security, and accessibility.

4. Integration with External Systems: The drug traceability system should be able to integrate with external systems, such as manufacturing systems, inventory management systems, and regulatory databases. This integration allows for seamless data exchange and real-time visibility across different stakeholders.

5. Data Exchange Standards: Define and implement standardized data exchange formats and protocols to ensure interoperability between different systems and stakeholders. This can include standards such as GS1, Electronic Product Code Information Services (EPCIS), or Health Level Seven (HL7).

6. Traceability and Verification: Implement mechanisms to trace and verify the movement of pharmaceutical products at each stage of the supply chain. This can involve scanning or reading the unique identifiers, verifying the authenticity of products, and recording the transactional data.

7. Reporting and Analytics: Enable reporting and analytics capabilities to derive insights from the captured data. This can include generating reports on product movement, identifying trends, detecting anomalies, and facilitating regulatory compliance.



SPRINT PLANNING & ESTIMATION

Sprint planning and estimation for drug traceability projects involve specific considerations to ensure compliance with regulations and maintain product integrity. Here are the steps involved in sprint planning and estimation for drug traceability projects:

1. Understand Regulatory Requirements: Familiarize yourself with the specific regulations and standards related to drug traceability in your region. This may include requirements such as serialization, track and trace, and product verification.

2. Product Backlog: Create a product backlog that includes all the necessary features, tasks, and requirements related to drug traceability. This may include functionalities like barcode scanning, data capture, integration with existing systems, and reporting capabilities.

3. User Stories: Break down the items in the product backlog into user stories. Each user story should represent a specific functionality or requirement related to drug traceability. Ensure that each user story is small enough to be completed within a sprint.

4. Estimation: Estimate the effort required to complete each user story. Consider factors such as the complexity of the functionality, integration requirements, and any specific challenges related to drug traceability. Use estimation techniques like story points or ideal days to estimate the effort accurately.

5. Sprint Duration: Determine the duration of each sprint based on the project's needs, team capacity, and complexity of the user stories. Consider the time required for testing, validation, and regulatory compliance activities.

6. Sprint Goal: Define a sprint goal that aligns with the overall objective of achieving drug traceability. The sprint goal should be based on the prioritized user stories from the product backlog and should contribute to meeting regulatory requirements.

7. Sprint Planning Meeting: Conduct a sprint planning meeting with the development team. Review and discuss the user stories selected for the sprint based on their priority and estimated effort. Ensure that the selected user stories contribute to achieving the sprint goal.

8. Sprint Backlog: Create a sprint backlog that includes the user stories selected for the sprint. The sprint backlog should also include the necessary tasks, subtasks, or technical requirements needed to complete each user story.

9. Task Estimation: Break down the user stories into smaller tasks and estimate the effort required for each task. Consider activities like development, testing, documentation, and validation. Ensure that the tasks are aligned with the regulatory requirements.

10. Sprint Execution: Start working on the tasks defined in the sprint backlog. Collaborate, communicate, and track progress regularly during the sprint. Ensure that the development team adheres to the regulatory requirements and best practices for drug traceability.

CODING & SOLUTIONING

Coding refers to the process of translating a software design into a functional program using programming languages like JavaScript, Python, or Java. It involves writing code, debugging, and optimizing algorithms to create a solution for a specific problem or requirement.

Solutioning involves designing a comprehensive solution strategy before coding. It includes problem analysis, architectural design, selecting appropriate technologies, and planning for scalability and security. Solutioning ensures that the software addresses the problem effectively and aligns with the project's goals.

FEATURE 1

Feature 1: User Authentication

```
function authenticateUser(username, password) {
// Code to validate username and password
        if (isValidCredentials(username, password)) {
            return "Authentication successful";
        }
        else {
            return "Authentication failed";
        }
    }
function isValidCredentials(username, password) {
```

```
    // Code to check credentials against database or backend
system
    // Return true if valid, false otherwise
}
```

Explanation:

The code snippet checks the provided username and password against a database or backend system. If the credentials are valid, the function returns "Authentication successful"; otherwise, it returns "Authentication failed." This feature ensures secure user access to the system.

- **FEATURE 2**

  Feature 2: Data Encryption

```
const crypto = require('crypto');
function encryptData(data, key) {
    const cipher = crypto.createCipher('aes-256-cbc', key);
    let encryptedData = cipher.update(data, 'utf-8', 'hex');
    encryptedData += cipher.final('hex');
    return encryptedData;
```

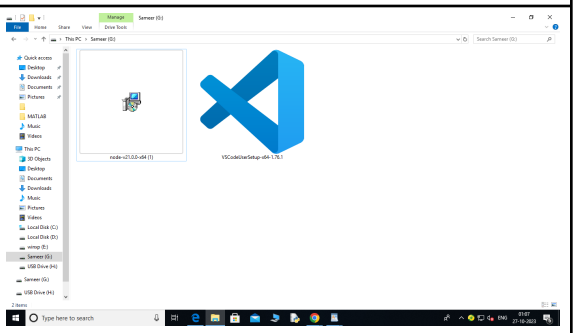
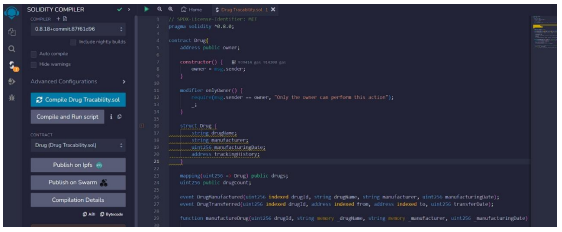
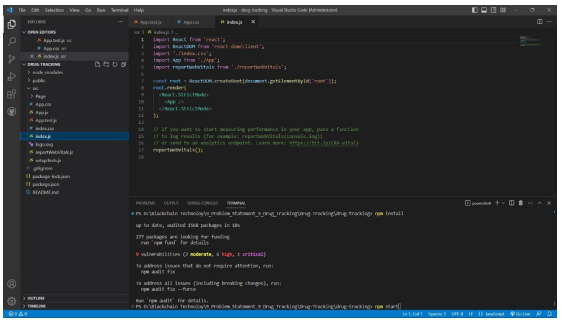
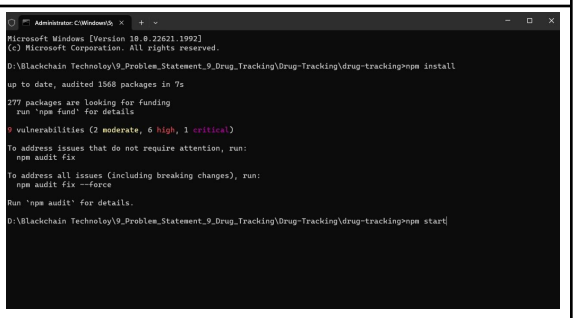
```
}
function decryptData(encryptedData, key) {
    const decipher = crypto.createDecipher('aes-256-cbc', key);
    let decryptedData = decipher.update(encryptedData, 'hex', 'utf-8');
    decryptedData += decipher.final('utf-8');
    return decryptedData;
}
```

Explanation:

This code snippet demonstrates data encryption and decryption using the AES encryption algorithm. encryptData function takes data and a key, encrypts it, and returns the encrypted data in hexadecimal format. decryptData function reverses the process, decrypting the data using the same key.

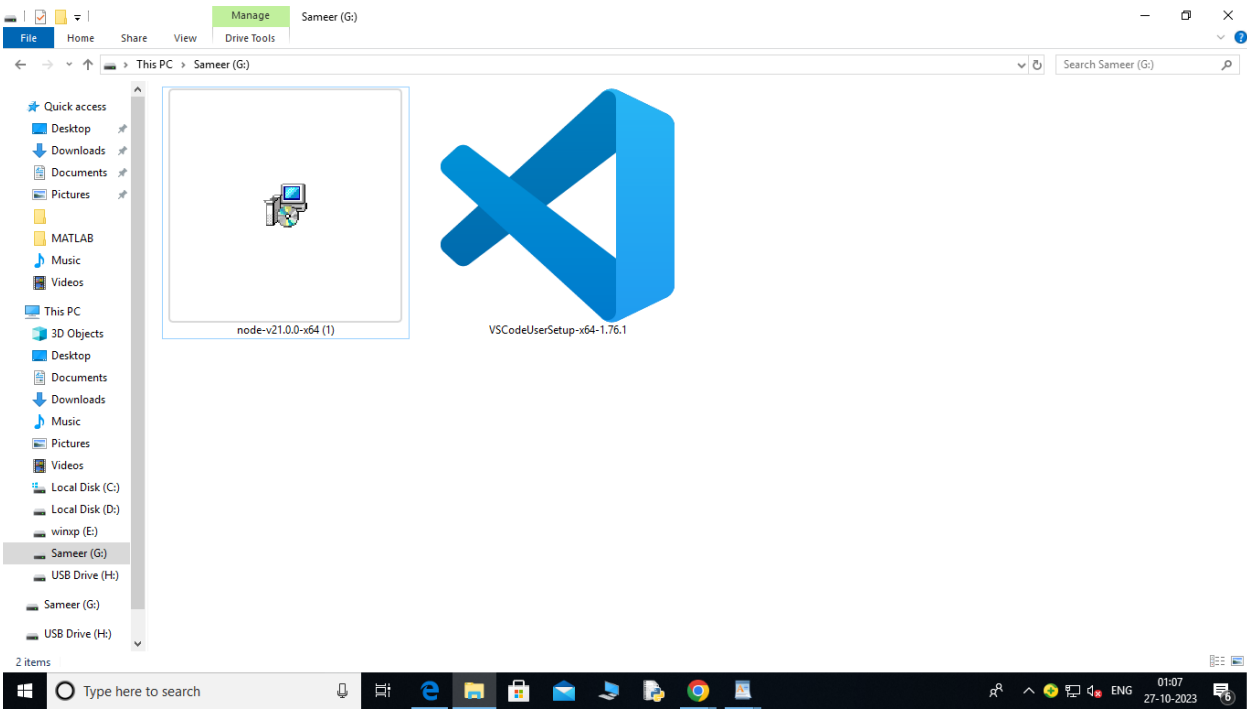
- **PERFORMANCE TESTING**

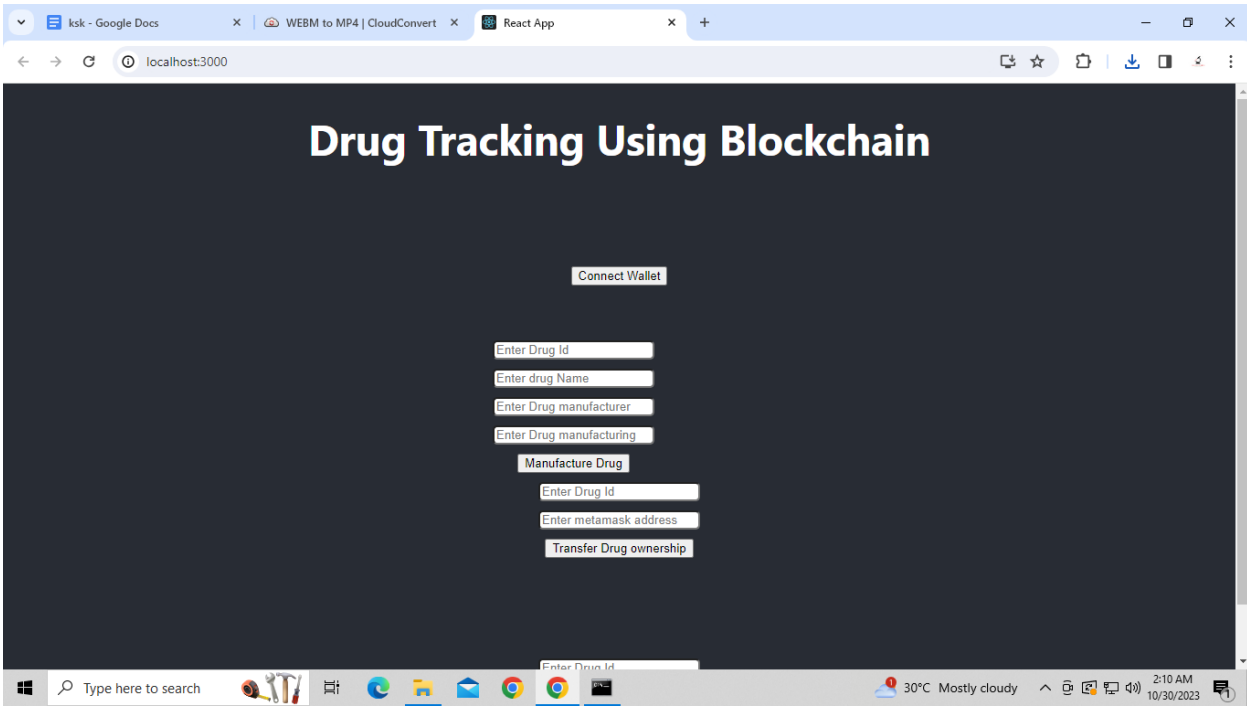
- **PERFORMANCE METRICS**

S. No	Parameters	Values	Screenshots
1.	Information Gathering	Setup all the prerequisite	
2.	Extract the zip file	Open to vs code	
3.	Remix IDE exploring	Deploy the smart contract code Deploy and run the transaction. By selecting the environment - inject the MetaMask.	
4.	Open filemanager	Open the extracted file and click on the folder. Open src, and search for utiles. Open cmd enter commands 1.npm install	

		2.npm bootstrap 3. npm start	
5.	LOCALHOST IP ADDRESS	Copy the address and open it to chrome so you can see the frontend of your project.	

OUTPUT :

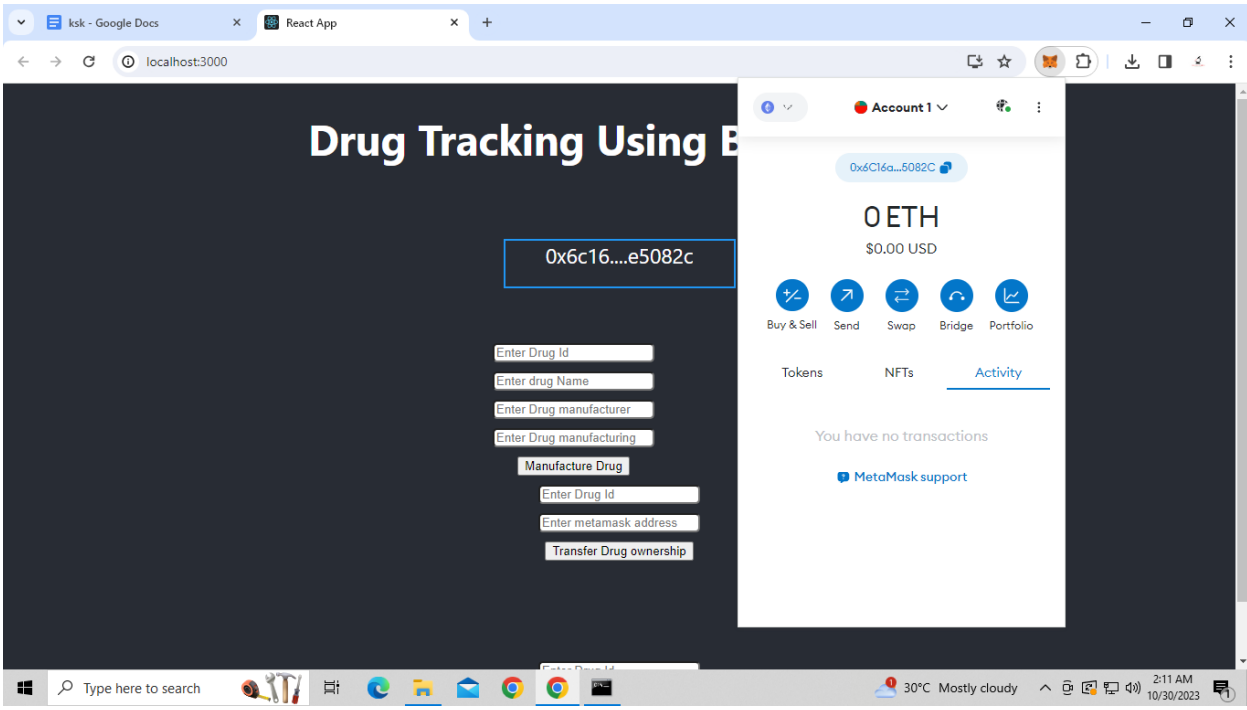


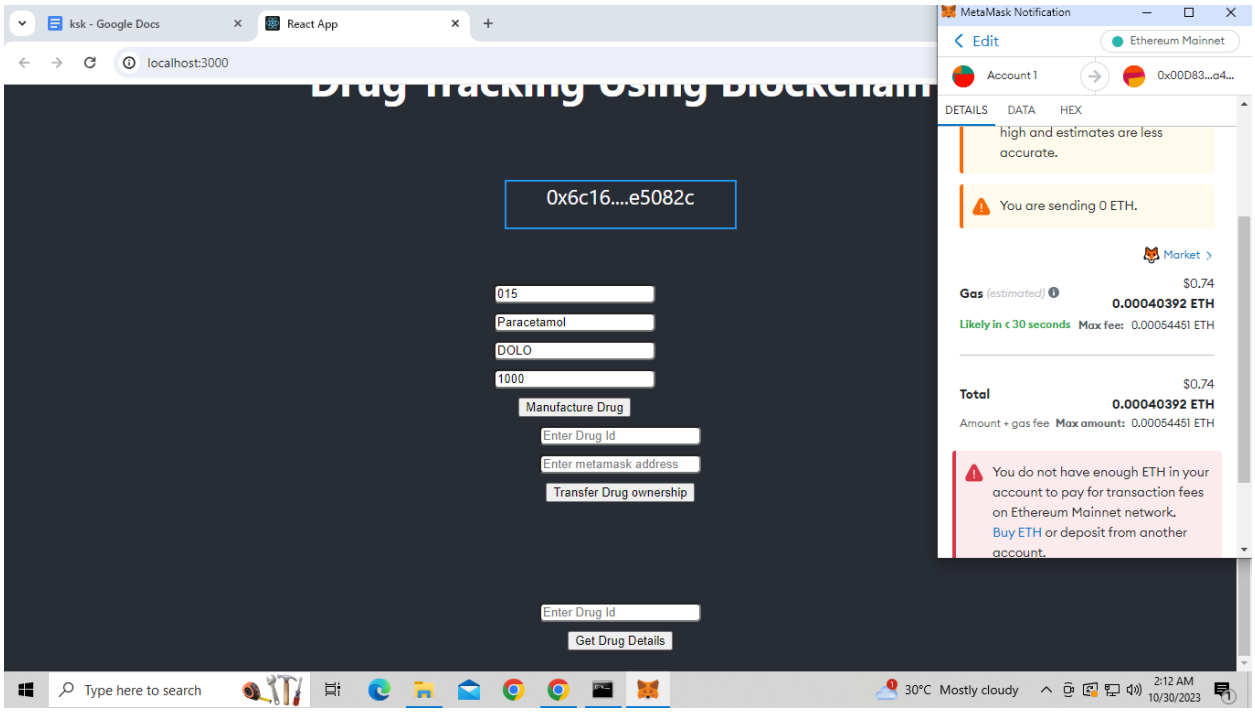


Home.js - Notepad

File Edit Format View Help

```
import React, { useState } from "react";
import { Button, Container, Row, Col } from 'react-bootstrap';
import { contract } from "./connector";

function Home() {
   const [Id, setId] = useState("");
   const [DrugName, setDrugName] = useState("");
   const [Manufacturer, setManufacturer] = useState("");
   const [date, setDate] = useState("");
   const [TranId, setTranId] = useState("");
   const [Owner, setOwner] = useState("");
   const [BookId, setBookId] = useState("");
   const [BookDet, setBookDet] = useState("");
   const [Wallet, setWallet] = useState("");


   const handleId = (e) => {
      setId(e.target.value)
   }

   const handleDrugName = (e) => {
      setDrugName(e.target.value)
   }

   const handleManufacturer = (e) => {
      setManufacturer(e.target.value)
   }

   const handleDate = (e) => {
      setDate(e.target.value)
   }

   const handleDrugManufacture = async () => {
      try {
         let tx = await contract.manufactureDrug(Id.toString(), DrugName, Manufacturer, date)
```


ksk - Google Docs
React App
localhost:3000
Drug Tracking Using B
0x6c16....e5082c
Enter Drug Id
Enter drug Name
Enter Drug manufacturer
Enter Drug manufacturing
Manufacture Drug
Enter Drug Id
Enter metamask address
Transfer Drug ownership
Account 1
0x6C16a...5082C
0 ETH
$0.00 USD
Buy & Sell
Send
Swap
Bridge
Portfolio
Tokens
NFTs
Activity
You have no transactions
MetaMask support
Type here to search
30°C Mostly cloudy
2:11 AM
10/30/2023


ksk - Google Docs
React App
localhost:3000
0x6c16....e5082c
015
Paracetamol
DOLO
1000
Manufacture Drug
Enter Drug Id
Enter metamask address
Transfer Drug ownership
Enter Drug Id
Get Drug Details
MetaMask Notification
Edit
Ethereum Mainnet
Account 1
0x00D83...a4...
DETAILS
DATA
HEX
high and estimates are less accurate.
You are sending 0 ETH.
Market
Gas (estimated)
$0.74
0.00040392 ETH
Likely in < 30 seconds
Max fee: 0.00054451 ETH
Total
$0.74
0.00040392 ETH
Amount + gas fee
Max amount: 0.00054451 ETH
You do not have enough ETH in your account to pay for transaction fees on Ethereum Mainnet network.
Buy ETH or deposit from another account.
Type here to search
30°C Mostly cloudy
2:12 AM
10/30/2023

- **ADVANTAGES & DISADVANTAGES**

Advantages :

Enhanced Data Security: Utilizing blockchain technology ensures enhanced security and immutability of electronic health records, protecting sensitive patient information from unauthorized access and tampering.

Interoperability: The project fosters seamless data exchange among healthcare providers, improving collaboration, patient care, and overall healthcare system efficiency.

Patient Empowerment: By allowing patients control over their health data and granting or revoking access, the project empowers individuals to actively manage their healthcare, fostering a sense of ownership and engagement.

Efficient Data Management: Smart contracts automate data-sharing agreements, reducing administrative overhead and enabling more efficient and standardized data management processes.

Transparency and Trust: Blockchain's transparent nature builds trust among stakeholders. Every action within the system is recorded, enhancing accountability and transparency.

Innovation and Future-Readiness: Embracing cutting-edge technologies like blockchain and smart contracts positions the project as an innovative solution, ready to adapt to evolving healthcare needs and technological advancements.

Disadvantages of the Whole Project:

Complex Implementation: Implementing blockchain solutions can be complex and require specialized knowledge, potentially leading to development challenges and delays.

Integration Challenges: Integrating the blockchain-based system with existing healthcare infrastructure might pose challenges, especially if the legacy systems are outdated or incompatible.

Data Privacy Concerns: Although blockchain offers enhanced security, ensuring complete data privacy, especially in regions with stringent regulations like GDPR, requires meticulous design and compliance efforts.

Scalability: Blockchain networks, particularly public ones, might face scalability issues when dealing with a large volume of transactions. Ensuring the system can scale to meet increasing demands is crucial.

User Adoption: Healthcare professionals and patients might face a learning curve when transitioning to a new system. Adequate

training and user-friendly interfaces are essential for successful adoption.

Regulatory Compliance: Adhering to healthcare data regulations, which vary across jurisdictions, poses a challenge. Ensuring the project complies with regional laws and regulations is critical for legal acceptance and trust.

- **CONCLUSION**

In conclusion, drug traceability is a critical aspect of the pharmaceutical industry, ensuring product integrity, patient safety, and regulatory compliance. Sprint planning and estimation for drug traceability projects involve understanding regulatory requirements, creating a product backlog, breaking down user stories, estimating effort, determining sprint duration, setting sprint goals, conducting sprint planning meetings, creating sprint backlogs, task estimation, sprint execution, and conducting sprint reviews and retrospectives.

By following these steps, pharmaceutical companies can effectively plan and execute projects related to drug traceability, integrating features like barcode scanning, data capture, integration with existing systems, and reporting capabilities. Collaboration with stakeholders, regulatory experts, and quality assurance teams is essential throughout the process to ensure compliance with regulations and maintain product integrity.

Implementing drug traceability measures not only helps in meeting regulatory requirements but also enhances supply chain transparency, reduces the risk of counterfeit drugs, and improves patient safety. It is crucial for pharmaceutical companies to prioritize drug traceability and incorporate it into their sprint planning and estimation processes to ensure the successful implementation of traceability systems.

• FUTURE SCOPE

The future scope for drug traceability is promising, as advancements in technology and increasing regulatory requirements continue to drive the need for more robust and efficient traceability systems. Here are some potential areas of development and growth in drug traceability:

1. Blockchain Technology: Blockchain has the potential to revolutionize drug traceability by providing a decentralized and tamper-proof system for recording and verifying transactions. It can enhance supply chain transparency, traceability, and authentication, reducing the risk of counterfeit drugs and improving patient safety.

2. Internet of Things (IoT): IoT devices, such as smart packaging and sensors, can be integrated into drug packaging to provide real-time monitoring of temperature, humidity, and other environmental

conditions. This ensures that drugs are stored and transported under optimal conditions, reducing the risk of spoilage or degradation.

3. Artificial Intelligence (AI) and Machine Learning (ML): AI and ML algorithms can be utilized to analyze large volumes of data collected from various sources, such as supply chain records, manufacturing processes, and adverse event reports. This can help identify patterns, detect anomalies, and improve the efficiency of traceability systems.

4. Serialization and Track-and-Trace Technologies: Serialization involves assigning a unique identifier to each drug unit, enabling its tracking throughout the supply chain. Track-and-trace technologies, such as RFID (Radio Frequency Identification) and barcode scanning, can be used to capture and record this information, ensuring accurate and efficient traceability.

5. Global Harmonization: Efforts are being made to establish global standards and regulations for drug traceability, such as the Drug Supply Chain Security Act (DSCSA) in the United States and the Falsified Medicines Directive (FMD) in the European Union. The future scope involves further harmonization of these regulations to facilitate seamless traceability across international borders.

6. Integration with Healthcare Systems: Integration of drug traceability systems with electronic health records (EHRs) and healthcare provider systems can enhance patient safety by enabling real-time access to drug information, including recalls, adverse events, and interactions with other medications.

APPENDIX

1. Technical Specifications:

Detailed technical specifications including programming languages, frameworks, and libraries used.Database schema diagrams. Blockchain integration details, such as the chosen blockchain platform (Ethereum, Hyperledger, etc.) and smart contract code snippets.

2. User Guides:

Comprehensive user guides for healthcare providers, patients, and administrators, explaining system functionalities, authentication processes, and data management procedures. Metamask setup instructions for users unfamiliar with blockchain interactions.

3. Code Samples:

Code snippets demonstrating key features like user authentication, data encryption, smart contract interactions, and API integrations. Examples of error handling and edge cases in the codebase.

4. Data Security Measures:

Detailed information about encryption algorithms used for data security. Explanation of access control mechanisms implemented in smart contracts. Protocols and policies ensuring data confidentiality and integrity during transmission and storage.

5. Performance Testing Reports:

Performance testing methodologies, including tools used and test scenarios.Performance test results, including response times, throughput, and system scalability under various loads.

6. Regulatory Compliance Documentation:

Documentation showcasing how the project complies with healthcare data regulations (HIPAA, GDPR, etc.). Details about user consent management and data deletion policies.

7. User Feedback and Improvement Reports:

Summaries of user feedback collected during usability testing. Reports on system improvements and enhancements made based on user suggestions.

8. Future Enhancements:

Detailed plans for future enhancements, including new features, integrations, and technology upgrades. Roadmap outlining the project's evolution over the next few years.

9. References:

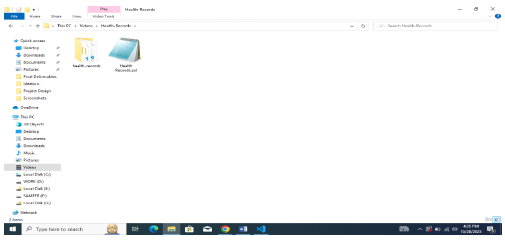
Citations and references for research papers, articles, and resources used during the project development. Links to relevant documentation, libraries, and frameworks.

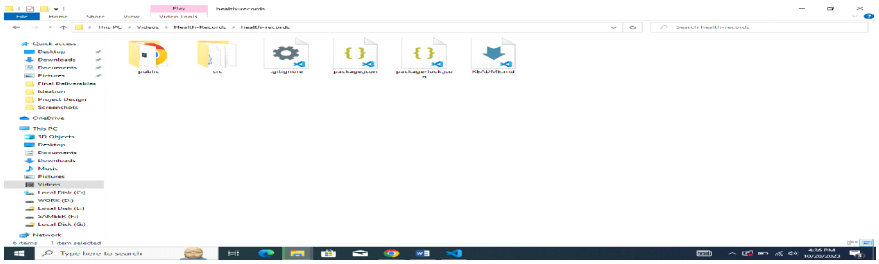
10. Glossary:

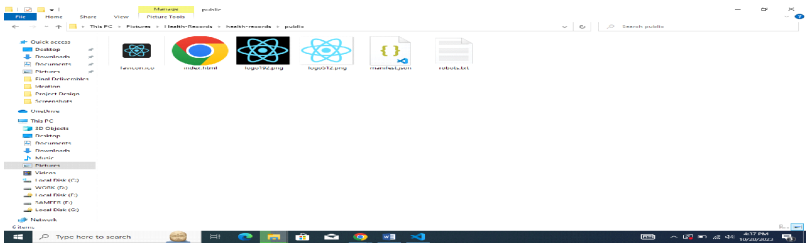
Definitions and explanations of technical terms, acronyms, and industry-specific jargon used throughout the project documentation.

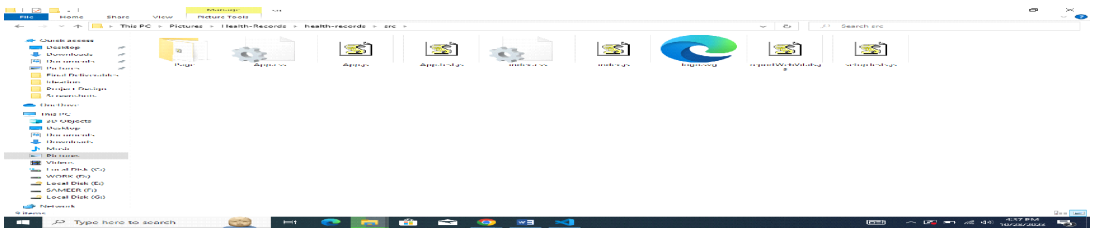
SOURCE CODE

Folder Structure :









INDEX.HTML

```
<!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>
```

MANIFEST.JSON

```
{
```

```
  "short_name": "React App",
  "name": "Create React App Sample",
  "icons": [
    {
      "src": "favicon.ico",
      "sizes": "64x64 32x32 24x24 16x16",
      "type": "image/x-icon"
    },
    {
      "src": "logo192.png",
      "type": "image/png",
      "sizes": "192x192"
    },
    {
      "src": "logo512.png",
      "type": "image/png",
      "sizes": "512x512"
    }
  ],
  "start_url": ".",
  "display": "standalone",
  "theme_color": "#000000",
  "background_color": "#ffffff"
}
```

CONNECTOR.JS

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

```
const abi =[
    {
        "anonymous": false,
        "inputs": [
            {
                "indexed": true,
                "internalType": "uint256",
                "name": "recordId",
                "type": "uint256"
            },
            {
                "indexed": true,
                "internalType": "address",
                "name": "patientAddress",
                "type": "address"
            }
        ],
        "name": "RecordCreated",
        "type": "event"
    },
    {
        "anonymous": false,
        "inputs": [
            {
                "indexed": true,
                "internalType": "uint256",
                "name": "recordId",
                "type": "uint256"
            },
            {
                "indexed": true,
                "internalType": "address",
                "name": "from",
                "type": "address"
            },
            {
                "indexed": true,
                "internalType": "address",
                "name": "to",
                "type": "address"
            }
        ],
        "name": "RecordTransferred",
        "type": "event"
```

```
},
{
    "inputs": [
        {
            "internalType": "uint256",
            "name": "recordId",
            "type": "uint256"
        },
        {
            "internalType": "string",
            "name": "name",
            "type": "string"
        },
        {
            "internalType": "address",
            "name": "_patientAddress",
            "type": "address"
        },
        {
            "internalType": "string",
            "name": "_diseases",
            "type": "string"
        },
        {
            "internalType": "string",
            "name": "_contactInfo",
            "type": "string"
        }
    ],
    "name": "createRecord",
    "outputs": [],
    "stateMutability": "nonpayable",
    "type": "function"
},
{
    "inputs": [
        {
            "internalType": "uint256",
            "name": "recordId",
            "type": "uint256"
        }
    ],
    "name": "getRecordData",
    "outputs": [
```

```
            {
                "internalType": "string",
                "name": "",
                "type": "string"
            },
            {
                "internalType": "address",
                "name": "",
                "type": "address"
            },
            {
                "internalType": "string",
                "name": "",
                "type": "string"
            },
            {
                "internalType": "string",
                "name": "",
                "type": "string"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "uint256",
                "name": "recordId",
                "type": "uint256"
            }
        ],
        "name": "getRecordOwner",
        "outputs": [
            {
                "internalType": "address",
                "name": "",
                "type": "address"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
```

```
    "inputs": [
        {
            "internalType": "uint256",
            "name": "",
            "type": "uint256"
        }
    ],
    "name": "records",
    "outputs": [
        {
            "internalType": "string",
            "name": "Name",
            "type": "string"
        },
        {
            "internalType": "address",
            "name": "patientAddress",
            "type": "address"
        },
        {
            "internalType": "string",
            "name": "dieses",
            "type": "string"
        },
        {
            "internalType": "string",
            "name": "contactInfo",
            "type": "string"
        }
    ],
    "stateMutability": "view",
    "type": "function"
},
{
    "inputs": [
        {
            "internalType": "uint256",
            "name": "recordId",
            "type": "uint256"
        },
        {
            "internalType": "address",
            "name": "newOwner",
            "type": "address"
```

```
            }
        ],
        "name": "transferRecord",
        "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 = "0xbCE87F01326965253e338bD5738587C02B481017"

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

HOME.JS

```
import React, { useState } from "react";

import { Button, Container, Row, Col } from 'react-bootstrap';

import '../../node_modules/bootstrap/dist/css/bootstrap.min.css';
import { contract } from "./connector";

function Home() {
 const [Id, setId] = useState("");
 const [name, setName] = useState("");
 const [pAddr, setpAddr] = useState("");
 const [disease, setdisease] = useState("");
 const [contact, setContact] = useState("");
 const [recordId, setrecordId] = useState("");
 const [newOwner, setNewOwner] = useState("");
 const [recordIdData, setrecordIdData] = useState("");
 const [Data, setData] = useState("");
  const [Wallet, setWallet] = useState("");


 const handleId = (e) => {
  setId(e.target.value)
 }

 const handleName = (e) => {
  setName(e.target.value)
 }

 const handlePatientAddress = (e) => {
  setpAddr(e.target.value)
 }

 const handleDisease = (e) => {
  setdisease(e.target.value)
 }

 const handleContact = (e) => {
  setContact(e.target.value)
 }

 const handleCreateRecord = async() => {
  try {
   let tx = await contract.createRecord(Id, name, pAddr, disease, contact)
```

```
  let wait = await tx.wait()
  alert(wait)
  console.log(wait.transactionHash);
 } catch (error) {
  alert(error)
 }
}

const handleRecordId = (e) => {
 setrecordId(e.target.value)
}

const handleNewOwner = (e) => {
 setNewOwner(e.target.value)
}

const handleTransferRecord = async () => {
 try {
  let tx = await contract.transferRecord(recordId.toString(),newOwner)
  let wait = await tx.wait()
  alert(wait.transactionHash)
  console.log(wait);
 } catch (error) {
  alert(error)
 }
}

const handleRecordDataId = (e) => {
 setrecordIdData(e.target.value)
}

const handleRecordData =async () => {
 try {
 let tx = await contract.getRecordData(recordIdData)
 let arr = []
 tx.map(e => arr.push(e))
   setData(arr)
 // alert(tx)
 console.log(tx);

 } catch (error) {
  alert(error)
 }
}
```

```
  const handleWallet = async () => {
    if (!window.ethereum) {
      return alert('please install metamask');
    }

    const addr = await window.ethereum.request({
      method: 'eth_requestAccounts',
    });

    setWallet(addr[0])

  }

 return (
  <div>
   <h1 style={{ marginTop: "30px", marginBottom: "80px" }}>Health Records Using
Blockchain</h1>
     {!Wallet ?

       <Button onClick={handleWallet} style={{ marginTop: "30px", marginBottom:
"50px" }}>Connect Wallet </Button>
       :
       <p style={{ width: "250px", height: "50px", margin: "auto", marginBottom:
"50px", border: '2px solid #2096f3' }}>{Wallet.slice(0,
6)}....{Wallet.slice(-6)}</p>
     }
     <Container style={{ margin:"Auto" }}>
       <Row >
     <Col>
      <div>


       <input style={{ marginTop: "10px", borderRadius: "5px" }}
onChange={handleId} type="number" placeholder="Enter Record Id" value={Id} /> <br
/>

       <input style={{ marginTop: "10px", borderRadius: "5px" }}
onChange={handleName} type="string" placeholder="Enter name" value={name} /> <br
/>

       <input style={{ marginTop: "10px", borderRadius: "5px" }}
onChange={handlePatientAddress} type="string" placeholder="Enter patient Address"
value={pAddr} /><br />
```

```
        <input style={{ marginTop: "10px", borderRadius: "5px" }}
onChange={handleDisease} type="string" placeholder="Enter disease"
value={disease} /><br />
        <input style={{ marginTop: "10px", borderRadius: "5px" }}
onChange={handleContact} type="string" placeholder="Enter contact Info"
value={contact} /><br />
        <Button onClick={handleCreateRecord} style={{ marginTop: "10px" }}
variant="primary">Create Record</Button>
       </div>
      </Col>
      <Col>
        <div>
        <input style={{ marginTop: "10px", borderRadius: "5px" }}
onChange={handleRecordId} type="number" placeholder="Enter new record Id"
value={recordId} /><br />
        <input style={{ marginTop: "10px", borderRadius: "5px" }}
onChange={handleNewOwner} type="string" placeholder="Enter new owner metamask
address" value={newOwner} /><br />
        <Button onClick={handleTransferRecord} style={{ marginTop: "10px" }}
variant="primary">Transfer Record</Button>

        </div>
      </Col>
     </Row>
      <Col>
       <Row style={{marginTop:"100px"}}>
       <input style={{ marginTop: "10px", borderRadius: "5px" }}
onChange={handleRecordDataId} type="string" placeholder="Enter Id"
value={recordIdData} /><br />
       <Button onClick={handleRecordData} style={{ marginTop: "10px" }}
variant="primary">Get Record Data</Button>
         {Data? Data?.map(e => {
         return <p>
           {e.toString()}
         </p>
         }
         ) : <p></p>}
       </Row>
      </Col>
    </Container>
   </div>
 )
}
```

```
export default Home;
```

APP.CSS

```
.App {
  text-align: center;
}

.App-logo {
  height: 40vmin;
  pointer-events: none;
}

@media (prefers-reduced-motion: no-preference) {
  .App-logo {
    animation: App-logo-spin infinite 20s linear;
  }
}

.App-header {
  background-color: #282c34;
  min-height: 100vh;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  font-size: calc(10px + 2vmin);
  color: white;
}

.App-link {
  color: #61dafb;
}

@keyframes App-logo-spin {
  from {
    transform: rotate(0deg);
  }
  to {
    transform: rotate(360deg);
  }
}
```

APP.JS

```
import './App.css';
import Home from './Page/Home'

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <Home />
      </header>
    </div>
  );
}

export default App;
```

APP.TEST.JS

```
import { render, screen } from '@testing-library/react';
import App from './App';

test('renders learn react link', () => {
  render(<App />);
  const linkElement = screen.getByText(/learn react/i);
  expect(linkElement).toBeInTheDocument();
});
```

INDEX.CSS

```
body {
  margin: 0;
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',
    'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
    sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}
```

```
code {
  font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',
    monospace;
}
```

INDEX.JS

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

REPORTWEBVITALS.JS

```
const reportWebVitals = onPerfEntry => {
  if (onPerfEntry && onPerfEntry instanceof Function) {
    import('web-vitals').then(({ getCLS, getFID, getFCP, getLCP, getTTFB }) => {
      getCLS(onPerfEntry);
      getFID(onPerfEntry);
      getFCP(onPerfEntry);
      getLCP(onPerfEntry);
      getTTFB(onPerfEntry);
    });
  }
};
```

```
export default reportWebVitals;
```

SETUPTESTS.JS

```
// jest-dom adds custom jest matchers for asserting on DOM nodes.
// allows you to do things like:
// expect(element).toHaveTextContent(/react/i)
// learn more: https://github.com/testing-library/jest-dom
import '@testing-library/jest-dom';
```

PACKAGE.JSON

```
{
  "name": "health-records",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@testing-library/jest-dom": "^5.17.0",
    "@testing-library/react": "^13.4.0",
    "@testing-library/user-event": "^13.5.0",
    "bootstrap": "^5.3.1",
    "ethers": "^5.6.6",
    "react": "^18.2.0",
    "react-bootstrap": "^2.8.0",
    "react-dom": "^18.2.0",
    "react-scripts": "5.0.1",
    "web-vitals": "^2.1.4"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
```

```
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  }
}
```

GITHUB & DEMO VIDEO

GitHub Link :
https://github.com/leenakumar02/Drug-Traceability-

Demo Video :

https://www.youtube.com/watch?v=Aha4C6DJRnQ&authuser=0