K S R INSTITUTE FOR ENGINEERING AND TECHNOLOGY

TIRUCHENGODE : 637 215

Computer Science and Engineering

NAAN MUDHALVAN

SB8024 – Blockchain Development

by Naan Mudhalvan Scheme - 2023

TEAM ID : NM2023TMID11701

PROJECT DOMAIN : BLOCKCHAIN TECHNOLOGY

PROJECT TITLE : BLOCKCHAIN TECHNOLOGY FOR ELECTRONIC HEALH RECORD

Submitted by

Name	Register Number
DEVADHARSHINI S	(731620104008)
INDHUMATHI M	(731620104022)
KAVISHNI S	(731620104028)
SOWMIYA G	(731620104051)

TABLE OF CONTENT

1.INTRODUCTION

In today's world we carry our photos, video, emails, and event banking services on our mobile device. But we are still unable to hold on to our medical records. As life events take patients away from one provider's data record into another, they leave data dispersed across various health institutions. The healthcare ecosystem is increasingly complex, with multiple stakeholders involved in complex and sensitive data interactions. This can lead to privacy challenges, data insecurity, and operational inefficiencies. Thus, the problem of Data Health Interoperability remains open. The main question is how to put patients at the center of their healthcare data and share medical data with known and unknown stakeholders while ensuring the protection of patient privacy, data integrity, and avoiding data misusage. In this article, we explore how blockchain and smart contracts can be applied to improve the way electronic health record (EHR) are handled across various health institutions. To do that, we will first present the blockchain methodology used to handle electronic EPiC Series then, we will propose an architecture that is able to improve the current EHR systems, we will discuss and analyze the usability of our software implementation for EHR using blockchain, and finally we will discuss some implementation challenges. By empowering patients with ownership and control over their health data, and facilitating secure and standardized data sharing across healthcare providers and systems, blockchain technology aims to enhance data management, reduce administrative overhead, and ultimately improve the quality of healthcare services. This makes the integration of blockchain technology into EHR systems a promising avenue for transforming healthcare data management and fostering a more patient-centric healthcare ecosystem.

1.1 PROJECT OVERVIEW

A blockchain technology project for electronic health records (EHRs) aims to revolutionize healthcare data management by leveraging the key features of blockchain, such as decentralization, immutability, transparency, and cryptographic security. The primary goal is to enhance data security, ensure interoperability across different healthcare providers and systems, empower patients with data ownership and control, maintain data accuracy and trust, and prevent fraud in healthcare. The project involves the development of a blockchain-based EHR system where patient health data is securely recorded on the blockchain, and authorized parties, such as healthcare providers and patients, can access and share this data in a secure and standardized manner. The project also addresses challenges related to scalability, privacy, integration with existing systems, regulatory compliance, and user adoption. This is achieved through careful planning, research, stakeholder engagement, and the integration of advanced cryptographic techniques.

The primary objectives of a blockchain technology project for electronic health records (EHRs) include enhancing the security, privacy, and integrity of health data, promoting interoperability and data sharing across different healthcare providers and systems, and empowering patients with ownership and control over their health data. The successful implementation of this project has the potential to transform the healthcare sector by providing a secure, interoperable, and patient-centric EHR system.

1.2 PURPOSE

The most popular usage of blockchain technologies was in the cryptocurrencies, such as Bitcoin [Nakamoto 2008]. Blockchain technology is grounded on the notion of a distributed ledger, which acts like a database containing information about the history of transactions involving those agents. It is constantly audited by groups of agents (selected according to different policies, depending on the application domain). The result of each auditing is stored in a block and broadcast to the network. Blocks are sequentially appended to the ledger, forming a cryptographically linked chain. Attempts to tamper the blocks or to alter their order can be easily detected.

The whole community may accept or reject the reliability of any block, according to a predefined set of rules. If an agent receives several valid additions to their local copy of the ledger, they always choose the longest chain of valid blocks (or the earliest one, if they have the same length), ignoring other conflicting and less relevant chains. This conceptually simple procedure ensures that consensus is eventually reached, even in scenarios where propagation is slow due to high network latency. If auditing is approved by the community, then the ledger – possibly containing recent, previously unverified transactions – is replicated across the agents. Otherwise, the largest accepted portion of the ledger is replicated with information about dissonances and corresponding actions to be taken – whose effects are then registered as new transactions to be audited in future rounds of verifications.

2. LITERATURE SURVEY

1) "Blockchain Technology: Principles and Applications in Medical Research" (2018) by Fatemeh Saheb Taramsari and Javad Dargahi: This study explores the principles of blockchain technology and its potential applications in medical research, particularly in ensuring data integrity, security, and confidentiality.
2) "Blockchain for Healthcare Data Management: Challenges, Opportunities, and a Conceptual Model" (2018) by Kudakwashe Dandajena and Yacine Atif: This paper discusses the challenges and opportunities of using blockchain for healthcare data management and proposes a conceptual model for a blockchain-based EHR system.
3) "A Survey on the Security of Blockchain Systems" (2017) by Xiaoqi Li, et al.: This comprehensive survey provides an overview of the security aspects of blockchain technology, including data integrity, privacy, and access control, which are relevant to EHR systems.
4) "Blockchain as an Enabling Technology for E-Health: A Systematic Review" (2019) by Daniel Diaz, et al.: This systematic review explores the use of blockchain as an enabling technology for e-health, including applications in EHR systems, and highlights the potential benefits and challenges.
5) "Blockchain in Healthcare: A Systematic Literature Review, Synthesis, and Framework Development" (2020) by Christian Esposito, et al.: This study provides a systematic literature review of blockchain applications in healthcare and proposes a framework for evaluating the feasibility and impact of blockchain-based EHR systems.

2.1 EXISTING PROBLEM

1. Data Security and Privacy: Traditional EHR systems are centralized and can be susceptible to hacking, data breaches, and unauthorized access. Blockchain technology, with its decentralized and secure nature, can help in ensuring data integrity, security, and privacy.

2.Interoperability: EHRs are often stored in silos with different healthcare providers, making it challenging to share and access records across different organizations. Blockchain can enable interoperability by creating a secure and unified system where data can be shared and accessed in a standardized manner.

3.Data Ownership and Control: Patients often have limited control over their health data and how it is used. Blockchain technology can provide a mechanism for patients to own and control their health data, and decide who has access to it.

4.Data Accuracy and Trust: Traditional EHR systems can be prone to errors and discrepancies. Blockchain technology can help ensure data accuracy and trust by maintaining an immutable ledger of all transactions, making it difficult for any single entity to alter the data.

5.Fraud Prevention: Blockchain technology can help in preventing fraud in healthcare by creating an immutable record of all transactions, making it difficult for any malicious actors to manipulate or alter data.

2.2 REFERENCES

- https://faucet.polygon.technology/

- https://app.mural.co/t/project1551/m/project1551/1698215591656/6d03fda4c0219ed67 6d60dd0a7c5956ea07415c8?sender=u2c629690e8195d281d960306

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

- https://remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&v ersion=soljson-v0.8.18+commit.87f61d96.js

2.3 PROBLEM STATEMENT DEFINITION

Blockchain is a technology designed to manage patient data that has the potential to support transparency and accountability. A blockchain is a ledger of transactions where an identical copy is visible to all the members of a computer network. Network members validate the data entered into the ledger, and once entered, the data is immutable. Create a solution where you can store the electronic health record of the patients in a distributed and decentralized network. You should be able to query and change the ownership of the record as necessary.

3. IDEATION & PROPOSED SOLUTION

The ideation for a blockchain technology project in electronic health records (EHR) stems from the growing need for secure, interoperable, and patient-centric healthcare data management. Traditional EHR systems face challenges related to data security, integrity, and efficient sharing among healthcare providers.

Proposed Solution:

Our proposed solution is to develop a blockchain-based EHR platform that addresses the following key objectives:

- **Security and Privacy:** Implement a robust and tamper-proof data storage system, ensuring that patients have full control over who can access their health records. Encryption and access control mechanisms will be integral to safeguarding patient data.

- **Interoperability:** Enable seamless sharing of EHRs among healthcare providers, allowing for a comprehensive view of a patient's medical history. Smart contracts can facilitate automated data sharing consent and revocation.

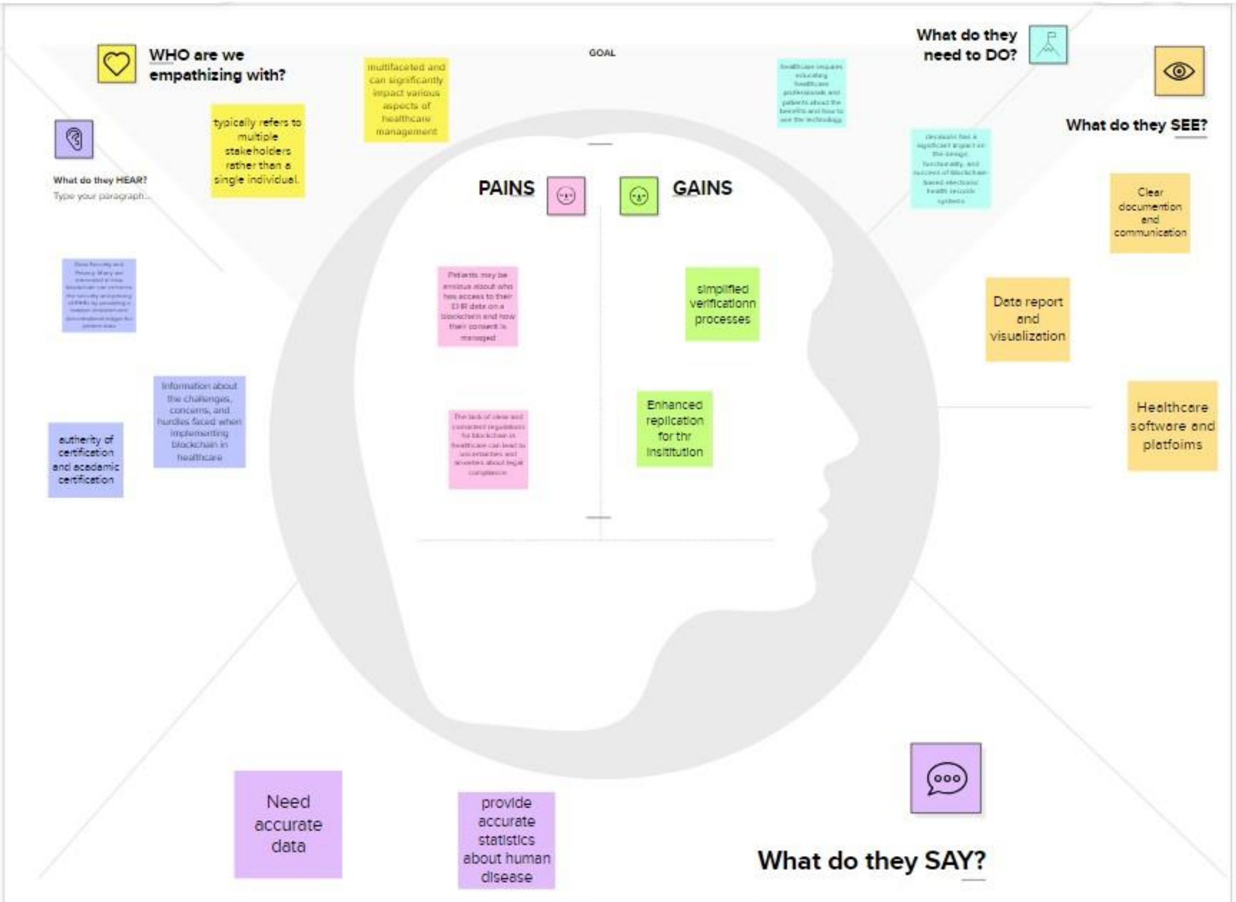
- **Data Integrity:** Ensure that EHRs are unaltered and accurate over time by leveraging blockchain's immutability. Any changes or updates will be recorded in an auditable manner.

- **Efficiency:** Reduce administrative burdens and redundancies by streamlining data exchange, ultimately improving healthcare delivery and reducing costs.

- **Compliance:** Ensure adherence to relevant data privacy regulations such as HIPAA in the United States or GDPR in Europe.

3.1 EMPATHY MAP CANVAS:

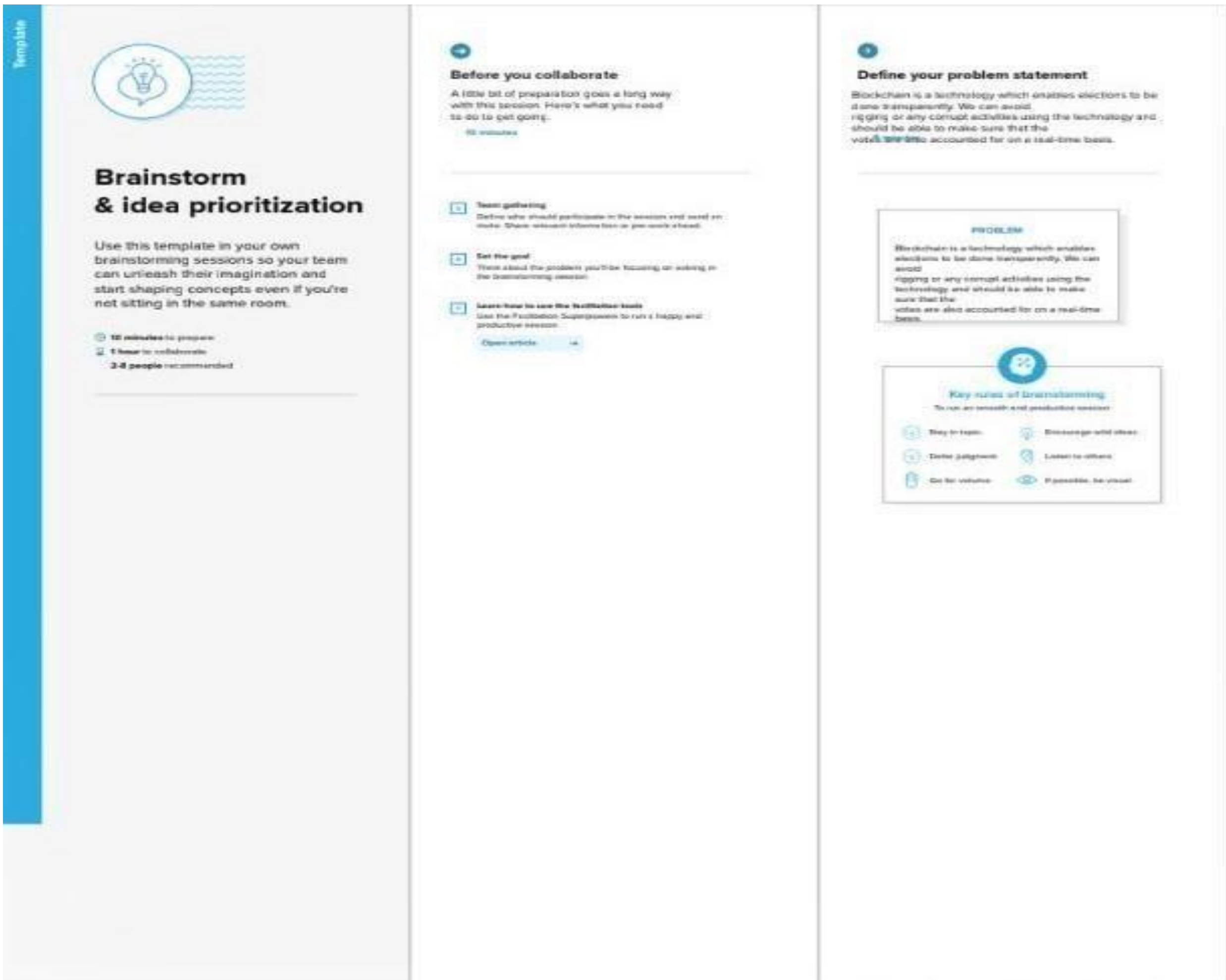
The Empathy Map Canvas can help developers to understand the needs of voters and to design a system that meets those needs. By considering the thoughts, feelings, and behaviors of voters, developers can create a system that is user-friendly and easy to use. The Empathy Map Canvas is a visual tool used to help teams better understand their customers or users. It is often used in design thinking and user experience (UX) research. By filling out the canvas, teams can gain a deeper understanding of the user's perspective and use that information to create more effectiv9e products and services.



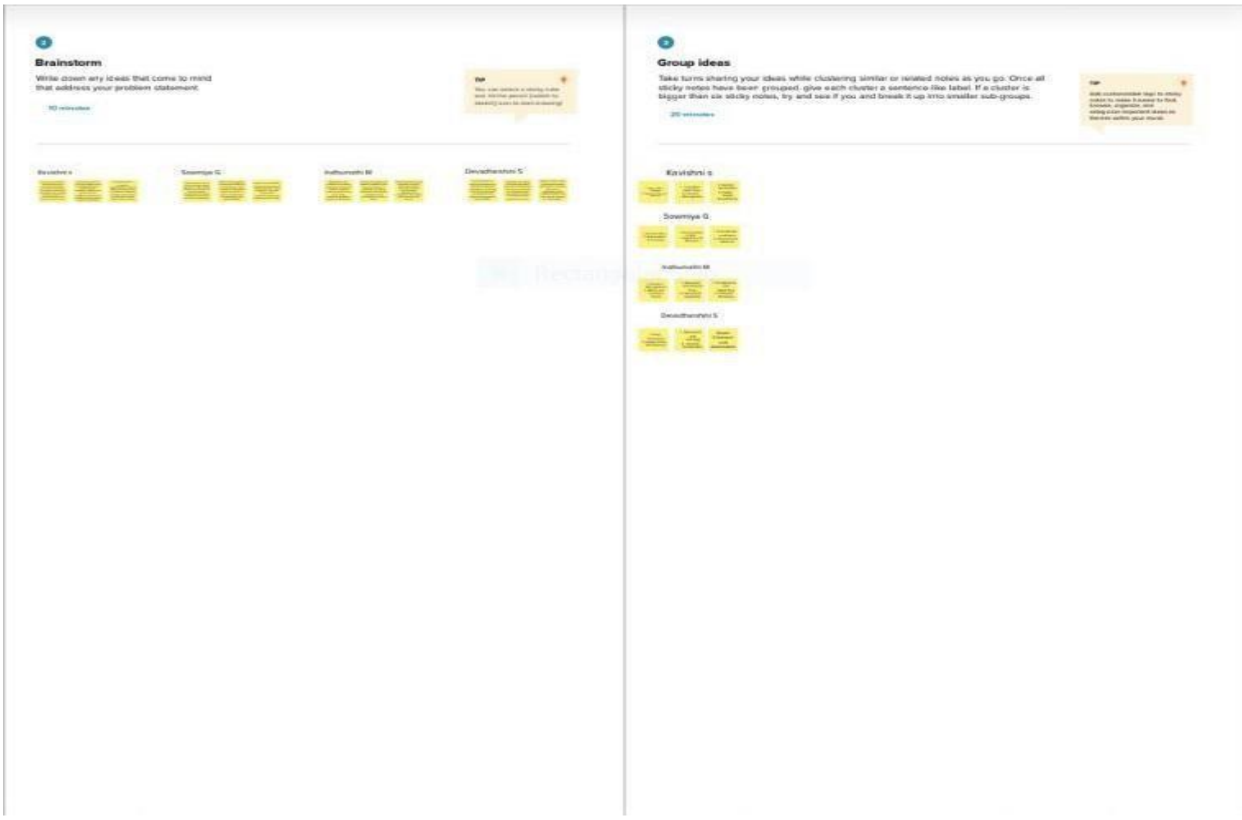
3.2 IDEATION & BRAINSTORMING :

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

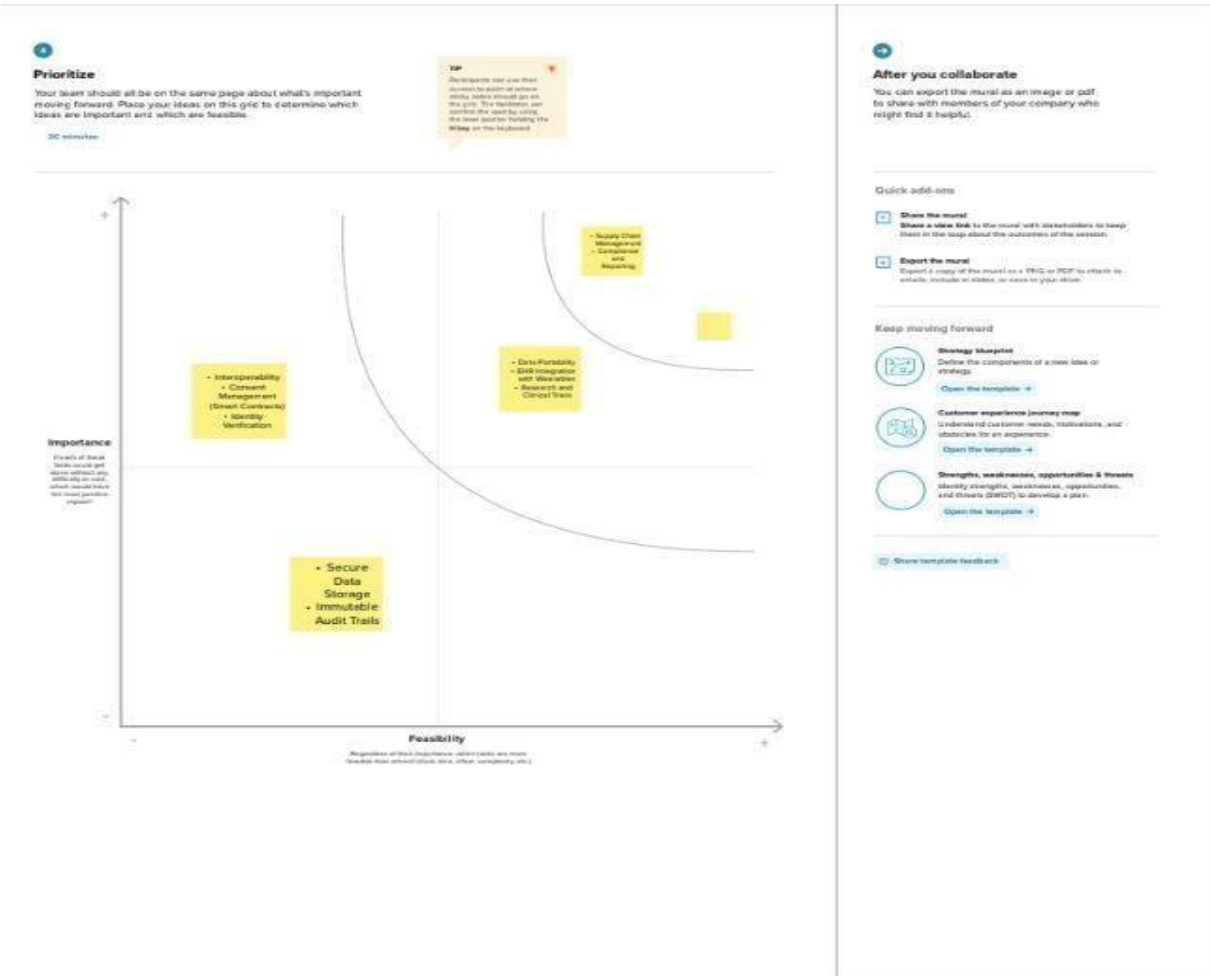
Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization



4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

A blockchain-based electronic health records (EHR) system must satisfy several functional requirements to be effective and user-friendly. Firstly, it must ensure data integrity and immutability, meaning once a record is added, it cannot be altered or deleted.

Data Encryption: The EHR system should be capable of encrypting sensitive health data to ensure privacy and confidentiality.

Identity Management: The system should provide a robust identity management mechanism to authenticate and authorize users, such as patients, healthcare providers, and administrators.

Access Control: The system should implement fine-grained access control to ensure that only authorized users can access or modify specific health data.

Data Sharing: The system should enable secure and efficient data sharing across different healthcare providers and institutions, facilitating interoperability.

Data Integrity: The system should ensure data integrity by maintaining an immutable ledger of all transactions, making it difficult to alter the data.

Compliance: The system should comply with relevant regulations and standards, such as HIPAA in the United States, to ensure the protection of patient data.

User-Friendly Interface: The system should provide an easy-to-use interface for patients and healthcare providers to interact with the EHR system.

4.2 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements for implementing blockchain technology in electronic health records (EHR) systems include:

Performance: Ensure the blockchain network can handle a large volume of transactions without significant delays, to meet the real-time demands of healthcare professionals.

Scalability: The system should be able to scale horizontally to accommodate growing amounts of data and increased user activity.

Availability: Ensure high availability of the EHR system, minimizing downtime for maintenance or upgrades.

Reliability: The system should be highly reliable, with minimal risk of failures, to ensure consistent access to critical patient data.

Security: Provide robust security measures, including encryption, identity management, and access control, to protect patient data from unauthorized access or cyberattacks.

Interoperability: Support interoperability standards to enable the exchange of health data with other systems and organizations seamlessly.

Usability: Ensure that the system is user-friendly for healthcare professionals, patients, and other authorized users.

Auditability: Maintain a comprehensive and immutable audit trail to facilitate accountability and compliance checks.

Data Backup and Recovery: Implement robust data backup and recovery mechanisms to prevent data loss in case of system failures or disasters.

Data Privacy: Guarantee the privacy of patient data, ensuring that sensitive information is not exposed to unauthorized parties.

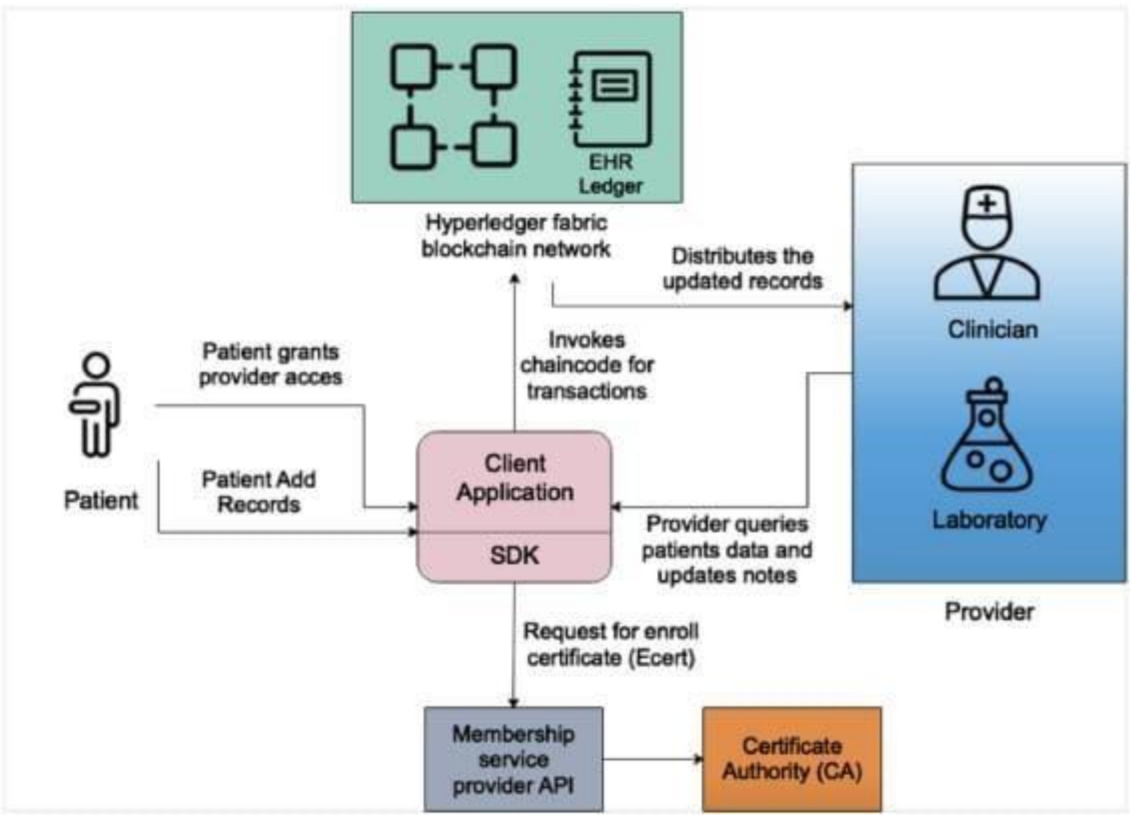
5.PROJECT DESIGN

5.1 DATA FLOW DIAGRAM & USER STORIES:

A data flow diagram (DFD) for a blockchain-based Electronic Health Records (EHR) project, focused on the interaction between healthcare providers and patients, plays a crucial role in illustrating the flow of data and information within the system. The Keys are,

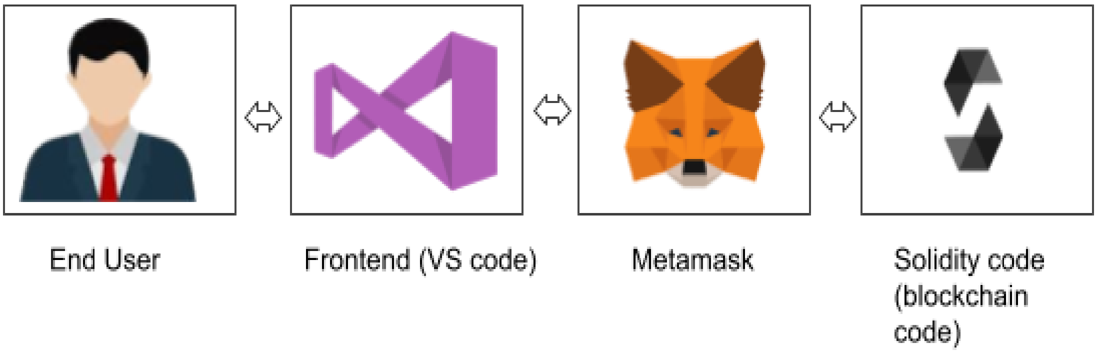
Providers: This category includes hospitals, clinics, physicians, nurses, and any healthcare professionals involved in patient care. Providers have the responsibility to create and update patient EHRs, maintaining accurate and up-to-date medical information

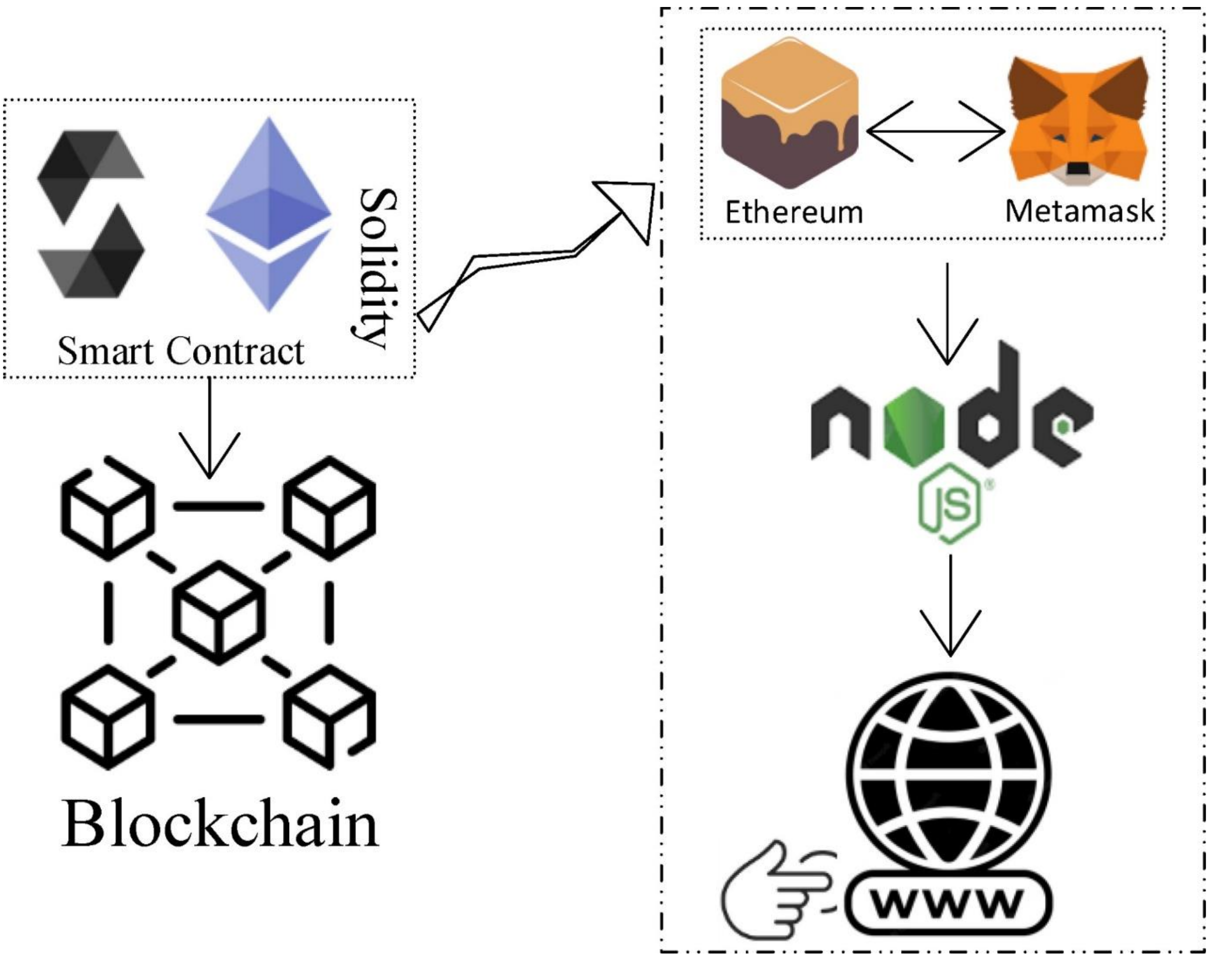
Patients: Patients are the primary subjects of the EHRs and have the right to access and manage their own health records. They can also authorize healthcare providers to access their data.



5.2 SOLUTION ARCHITECTURE :

The solution architecture for a blockchain-based Electronic Health Records (EHR) project, designed to facilitate secure data management and sharing between healthcare providers and patients, comprises a robust and innovative framework.



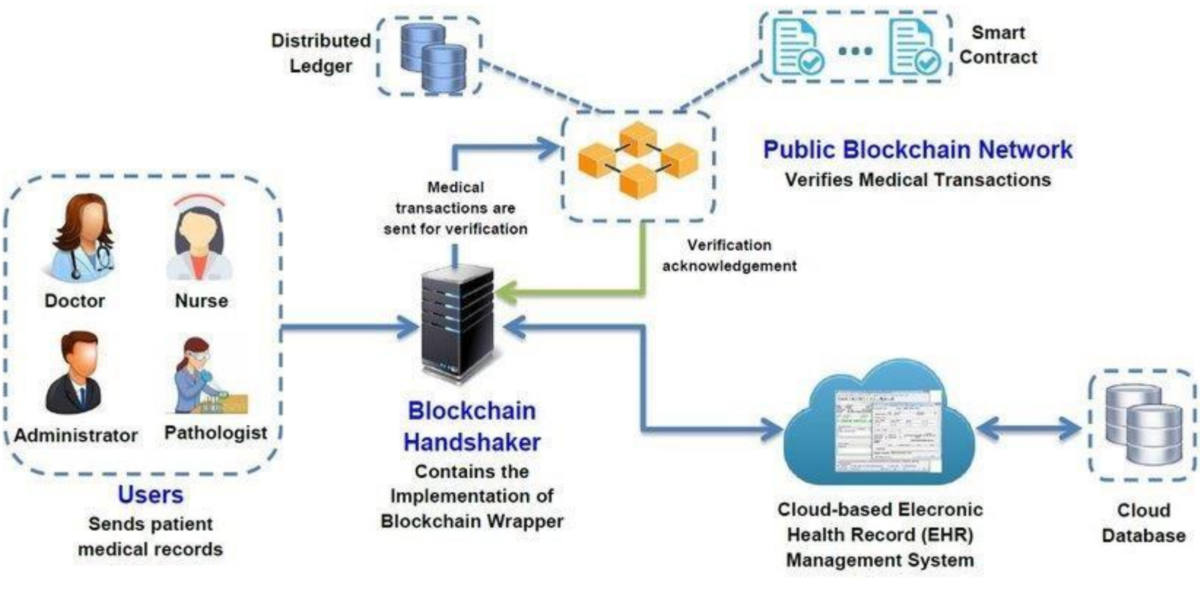


6 .PROJECT PLANNING AND SCHEDULING

6.1 TECHNICAL ARCHITECTURE:

The technical architecture of a blockchain-based Electronic Health Records (EHR) system for the interaction between healthcare providers and patients is a carefully designed framework that leverages cutting-edge technologies to ensure data security, reliability, and scalability.

The technical architecture also includes robust security measures, such as identity management and access control systems, to authenticate and authorize users. It incorporates cryptographic techniques for secure communication between the user interfaces and the blockchain network, ensuring data transmission remains confidential. The user interfaces, accessible through web applications or mobile apps, are designed with a focus on user experience and ease of interaction .



6.2 SPRINT PLANNING AND ESTIMATION:

Sprint planning and estimation for an Electronic Health Records system using blockchain can be a complex process, and it's essential to break down the project into manageable tasks and allocate resources effectively. Here's a high-level overview of how you can approach sprint planning and estimation for such a project:

Define Sprint Goal:

The Sprint Goal is a clear, concise statement of what the team aims to achieve during the sprint. It provides a common understanding of the objective and helps guide the team's work.

User Stories:

User stories are short descriptions of a feature from an end-user perspective. They define what needs to be done and serve as the basis for task breakdown and estimation.

Task Breakdown:

Tasks are the specific activities required to complete a user story. Breaking down user stories into tasks helps identify the work needed and provides a clear action plan for team members**.**

Story Points:

Story points are a relative measure of effort required to complete a user story or task. They are used to estimate the complexity and workload of each item. **Sprint Capacity:**

Sprint Capacity represents the amount of work the team can commit to completing in the sprint. It considers the team's velocity and available resources.

6.3 SPRINT DELIVERY SCHEDULE:

Developing blockchain technology for electronic health records (EHRs) requires a carefully planned and iterative approach. A sprint delivery schedule can help manage the project efficiently. Below is a hypothetical sprint schedule for a blockchain-based EHR system:

- **Sprint 1 (2 weeks - Discovery and Planning):**
  - Define project goals and objectives. Identify the EHR data elements to be stored on the blockchain. Research regulatory requirements (e.g., HIPAA compliance).
  - Select the blockchain platform and technologies (e.g., Ethereum, Hyperledger Fabric). Formulate a high-level architecture design.
- **Sprint 2 (2 weeks - Proof of Concept):**
  - Set up the selected blockchain platform. Create a small-scale proof of concept (PoC) to validate the chosen technologies.
  - Develop a prototype for a basic EHR data structure on the blockchain. Establish a consensus mechanism and data storage protocols.
- **Sprint 3 (3 weeks - MVP Development):**
  - Build the Minimum Viable Product (MVP) of the EHR system. Develop smart contracts for data storage and access control.
  - Implement identity verification and access management features. Ensure encryption and privacy measures are in place.
- **Sprint 4 (3 weeks - Security and Compliance):**
  - Conduct security audits and vulnerability assessments. Address any security concerns and implement necessary enhancements.
  - Perform a compliance assessment to ensure adherence to healthcare data regulations.
  - Integrate data encryption and decryption capabilities.

- **Sprint 5 (3 weeks - User Interface Development):**
  - Design and develop a user-friendly EHR interface for healthcare providers and patients.
  - Implement access controls and permission management within the interface.
  - Include features for data retrieval, updates, and sharing. Ensure data is presented in a clear and concise manner.
- **Sprint 6 (2 weeks - Testing and Quality Assurance):**
  - Conduct extensive testing, including functional, integration, and security testing. Address any bugs, issues, or discrepancies identified during testing.
  - Perform load testing to assess the system's scalability and performance.
  - Prepare documentation for users and administrators.
- **Sprint 7 (2 weeks - Deployment and User Training):**
  - Deploy the blockchain-based EHR system in a controlled environment.
  - Train healthcare providers, administrators, and patients on system usage and best practices.
  - Monitor system performance and make any necessary adjustments.
- **Sprint 8 (2 weeks - User Feedback and Refinement):**
  - Gather user feedback on the system's usability and functionality.
  - Implement improvements and refinements based on user input.
  - Address any additional user requests or identified issues.
- **Sprint 9 (1 week - Final Testing and Quality Assurance):**
  - Perform a final round of testing to ensure all issues have been addressed.
  - Conduct security testing and risk assessments.
  - Verify compliance with healthcare data regulations.
- **Sprint 10 (1 week - Go-Live):**
  - Officially launch the blockchain-based EHR system for production use.
  - Monitor system performance, security, and user feedback. Continue refining the system and addressing any emerging issues.

7.CODING AND SOLUTION:

7.1 FEATURE 1:

Utilize natural language processing (NLP) techniques to extract relevant keywords, phrases, or patterns from textual audit data.

Apply statistical analysis or rule-based algorithms to identify potential fraud indicators based on historical fraud cases or known patterns.

Develop machine learning models, such as text classification or topic modeling, to automatically detect fraud-related content in textual data.

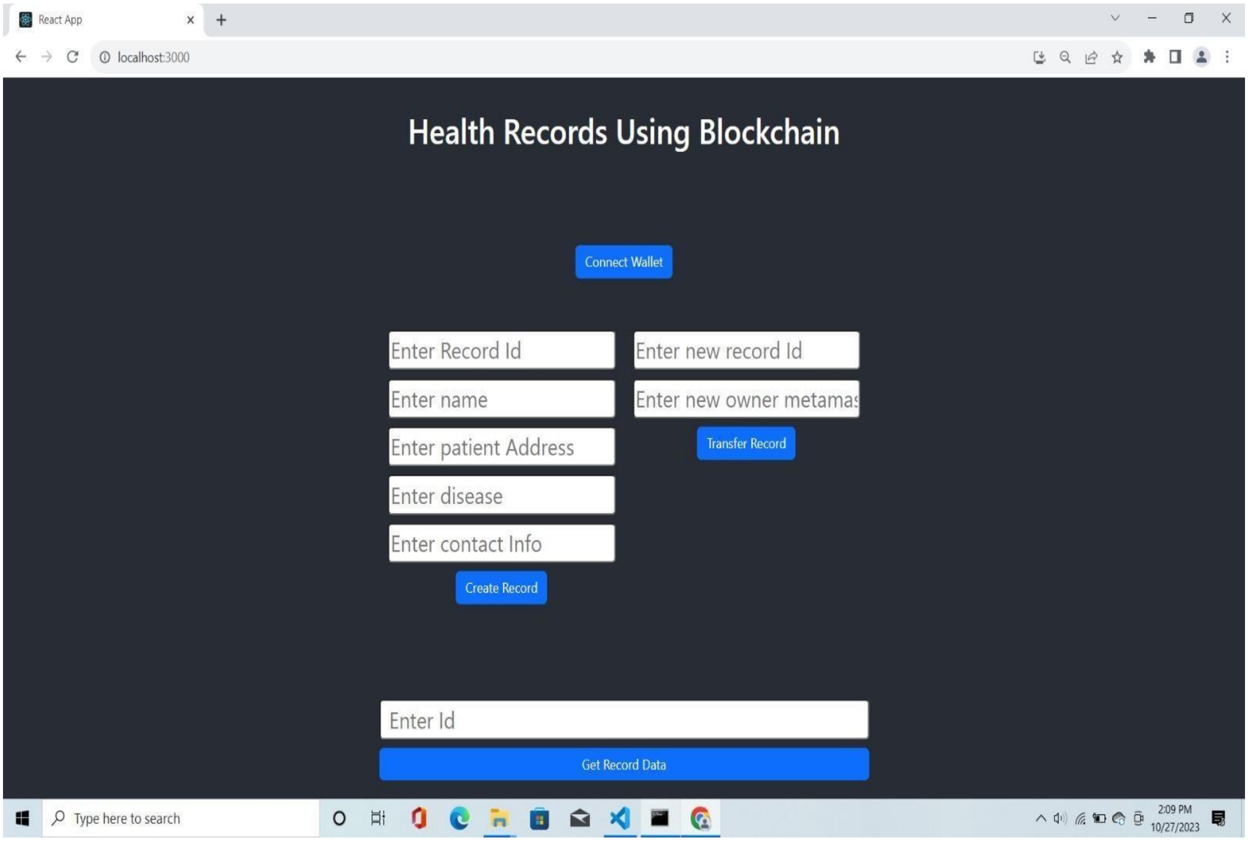
Coding:

```
contract HealthRecords {
   struct PatientRecord {
 string Name;
address patientAddress;
string dieses;
string contactInfo;
    }
 mapping(uint256 => PatientRecord) public records;
     event RecordCreated(uint256 indexed recordId, address indexed
patientAddress);
 event RecordTransferred(        uint256 indexed recordId,        address indexed
from,  address indexed to);
  modifier onlyOwner(uint256 recordId) {
        require(msg.sender == records[recordId].patientAddress,"Only contract owner
can call this");
        _;
    }
```

```
 function createRecord( uint256 recordId, string memory name, address
_patientAddress, string memory _diseases, string memory _contactInfo
    ) external {  records[recordId].Name = name;
        records[recordId].patientAddress = _patientAddress;
        records[recordId].dieses = _diseases;
records[recordId].contactInfo = _contactInfo;
  emit RecordCreated(recordId, _patientAddress);
    }
 function transferRecord(uint256 recordId, address newOwner) external
onlyOwner(recordId) {
          require(records[recordId].patientAddress == newOwner, "New Owner
should have different Address");
          require(records[recordId].patientAddress == msg.sender, "Only record owner
can transfer");
          records[recordId].patientAddress = newOwner;
         emit RecordTransferred(recordId, records[recordId].patientAddress,
newOwner);
    }
    function                  getRecordData(
uint256 recordId
    ) external view returns (string memory, address, string memory,string memory) {
return                         (records[recordId].Name,records[recordId].patientAddress,
records[recordId].dieses, records[recordId].contactInfo);
    }
    function getRecordOwner(uint256 recordId) external view returns (address) {
return records[recordId].patientAddress;
    }
}
```

7.2 FEATURE 2:

- Define risk factors and indicators based on fraud detection guidelines or industry standards.
- Assign weights or importance to each risk factor based on its relevance and impact on fraud detection.
- Calculate risk scores by aggregating the weighted risk factors and indicators for each audit entity or transaction.
- Categorize the risk levels (e.g., low, medium, high) based on predefined thresholds or risk thresholds specific to the audit engagement.



8.PERFORMANCE TESTING

8.1 PERFORMANCE METRICES

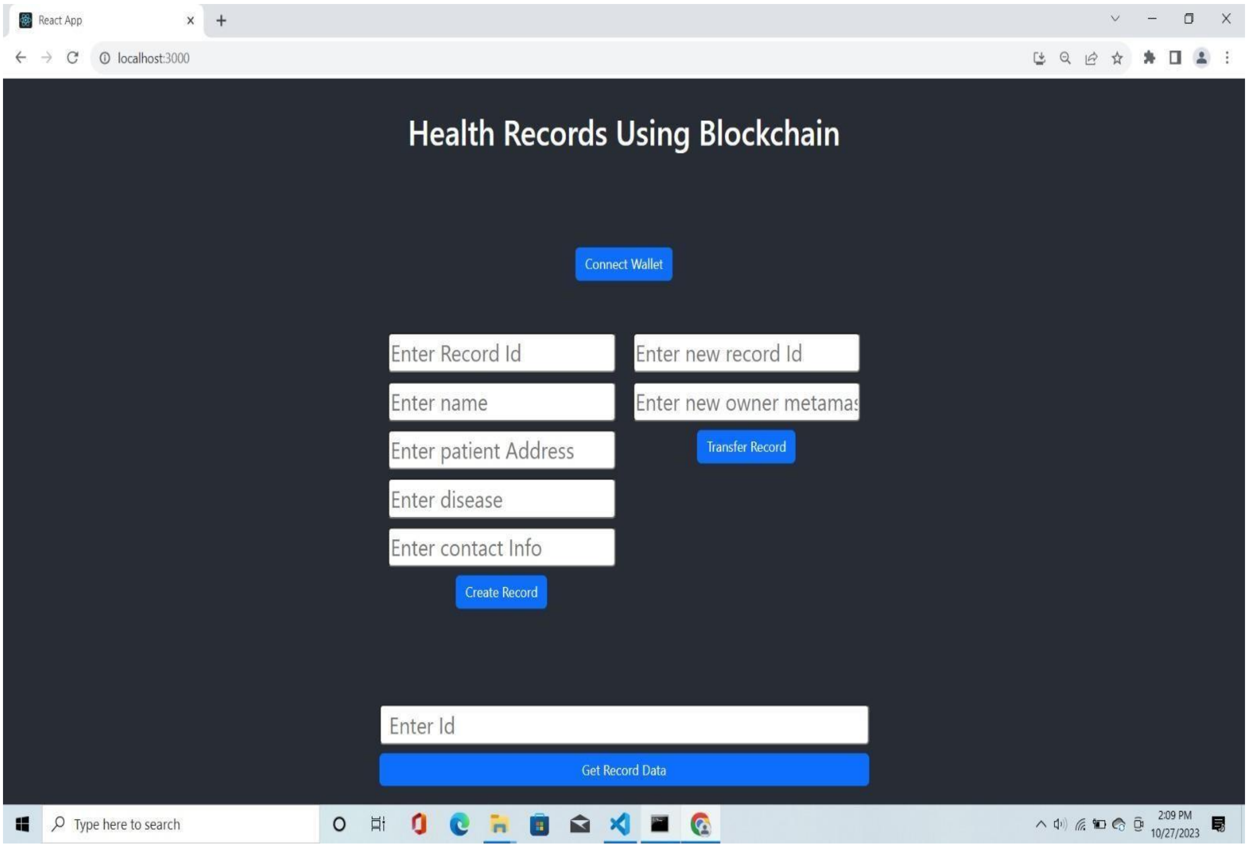
Performance metrics are used to evaluate the effectiveness of the Audit AI system in detecting fraud and assessing its overall performance. Here are some commonly used performance metrics for fraud detection:

- **Transaction Speed:** This metric measures the duration between when a transaction is initiated and when it is added to the blockchain. A faster transaction speed is crucial for real-time data access in a healthcare context.
- **Throughput:** This represents the total number of transactions that can be processed per unit of time. High throughput is necessary to accommodate large-scale health data transactions and to ensure that the system remains responsive and efficient.
- **Scalability:** This metric evaluates how well the system can scale horizontally or vertically to accommodate growth in data and users. Good scalability is critical for maintaining performance as the system grows.
- **Latency:** This measures the delay between the initiation of a transaction and its propagation across all nodes in the network. Lower latency is important to maintain data consistency across the blockchain network.
- **Data Integrity:** This is the measure of how accurately the data is recorded and maintained on the blockchain. Higher data integrity is essential to ensure that health records remain reliable and tamper-proof.
- **Storage Efficiency:** This refers to the efficiency with which the system utilizes storage space for the blockchain data. Efficient storage use is key for ensuring the system remains cost-effective as data grows.
- **False Negative Rate (Miss Rate or Type II Error):** It measures the proportion of fraudulent instances incorrectly identified as non-fraudulent.

9. RESULTS:

Blockchain technology has gained significant attention as a promising solution for electronic health records (EHRs). EHRs contain sensitive and critical patient information, making security and privacy paramount. Blockchain's decentralized and immutable ledger system enhances data security by reducing the risk of unauthorized access and tampering. Additionally, it can streamline data sharing among healthcare providers, improving the efficiency of patient care. Patients themselves can have greater control over their EHRs and grant access as needed. However, challenges such as scalability, interoperability, and regulatory concerns still need to be addressed to fully realize the potential of blockchain in healthcare. Nonetheless, it holds great promise in revolutionizing the way electronic health records are managed and shared, ultimately benefiting both patients and healthcare providers.

9.1 Output Screenshot



10.ADVANTAGES AND DISADVANTAGES :

Advantages:

- **Enhanced Fraud Detection:** Audit AI leverages machine learning algorithms to analyze audit data and identify patterns or anomalies indicative of fraud. It can detect complex fraud schemes that may be difficult for traditional rule-based approaches to identify, leading to improved fraud detection rates.
- **Automation and Efficiency:** By automating the fraud detection process, Audit AI reduces the reliance on manual reviews and enhances efficiency. It can process large volumes of audit data in a shorter time, allowing auditors to focus on investigating flagged transactions or entities.
- **Improved Accuracy:** Machine learning models used in Audit AI can learn from historical fraud cases and patterns, making them capable of detecting subtle fraud signals. This improves the accuracy of fraud detection by reducing false positives and false negatives.
- **Real-time Monitoring:** Audit AI can provide real-time or near real-time monitoring of audit data, allowing for proactive fraud detection and prevention. It enables auditors to identify potential fraud cases promptly and take appropriate actions to mitigate risks.
- **Scalability:** Audit AI can scale to handle large and diverse datasets, making it suitable for organizations with extensive audit data. It can adapt to changing data volumes and complexity, ensuring the system remains effective as the organization grows.
- **User Satisfaction:** This gauges how satisfied users (patients, healthcare providers, administrators, etc.) are with the system's performance, usability, and features. High user satisfaction indicates that the system meets user needs and expectations.

Disadvantages:

- **Scalability Issues:** Blockchain networks, especially those that use proof-of work consensus mechanisms, can face scalability issues. This can result in slower transaction times and higher costs as the network grows.
- **Data Privacy Concerns:** Even though blockchain provides a secure and tamper-proof ledger, the transparency of blockchain can pose data privacy challenges. Sensitive health data needs to be carefully handled to ensure compliance with privacy regulations.
- **Integration with Existing Systems:** Many healthcare providers already have established EHR systems in place. Integrating a new blockchain-based system with existing infrastructure can be complex and costly.
- **Complexity and Learning Curve:** Blockchain technology is relatively new and can be complex to understand and implement. There may be a steep learning curve for healthcare providers and administrators.
- **Regulatory Compliance:** The healthcare industry is heavily regulated. Ensuring that a blockchain-based EHR system complies with all relevant regulations and standards can be challenging.
- **Initial Costs:** Setting up a blockchain-based EHR system can be costly, especially in terms of developing or adapting the technology, training staff, and ensuring compliance with regulations.
- **Network Consensus Issues:** Achieving consensus on a blockchain network can sometimes be challenging, especially in a consortium or permissioned blockchain where multiple parties need to agree on the validity of transactions.

11.CONCLUSION:

In conclusion, blockchain technology holds tremendous promise for electronic health records (EHRs). Its decentralized and secure nature can ensure the integrity, privacy, and accessibility of patient data, ultimately improving healthcare quality and reducing fraud. However, successful implementation requires overcoming technical, regulatory, and interoperability challenges. With continued innovation and collaboration, blockchain has the potential to revolutionize the management of EHRs, enhancing patient care and data security in the healthcare industry.

Nowadays, despite recent advances in IT, few health institutions provide data integration between units. This work exploits the main healthcare scenarios of care coordination and health information management and proposes an architecture for secure data exchange. In our proposal, all data is owned by the patients. And it relies on blockchain technology and well-diffused cloud storage services to obtain security, high availability, fault tolerance and improved trust. Blockchain technology has emerged as a transformative solution for electronic health records.

In the future, after creating a reliable health data network, we could delegate to the blockchain the activity of data access grant, opening new frontiers to data health management. This innovation has the potential to revolutionize the health care industry, fostering trust among stack holders, ensuring data security, and stream lining data sharing. As health care providers, institutions and regulators continue to explore and address the challenges of implementing block chain in EHRs, it is evident that this technology offers a promising path towards improving patients care and health care data management.

12.FUTURE SCOPE:

- **Decentralized Autonomous Organizations (DAOs):** With the evolution of blockchain, we could see the emergence of DAOs for managing healthcare data, where governance and decision-making are automated and executed via smart contracts.
- **Integration with Artificial Intelligence (AI) and Machine Learning (ML):** Integrating AI and ML with blockchain-based EHR systems could facilitate advanced data analytics and predictive modeling. This could aid in personalized medicine, population health management, and improving healthcare outcomes.
- **Universal Health IDs:** Blockchain could be used to create a universal health ID system, which would simplify patient identification and streamline the process of accessing and sharing health records across different healthcare providers and systems.
- **Enhanced Data Security:** As blockchain technology matures, we could see more advanced cryptographic techniques being integrated into EHR systems, further enhancing data security and privacy.
- **Improved Interoperability:** Continued development of blockchain could lead to standardized protocols and frameworks for data exchange, improving interoperability between different EHR systems and healthcare organizations.
- **Tokenization of Health Data:** Blockchain could facilitate the tokenization of health data, enabling patients to have more control over their data and potentially monetize it by sharing it with research organizations and pharmaceutical companies.
- **Smart Contracts for Health Insurance Claims:** Blockchain-based smart contracts could be used to automate health insurance claims processing, reducing administrative costs and speeding up the reimbursement process for healthcare providers and patients.

13. APPENDIX :

Source Code:

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" />
        <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
       <title>React App</title>
   </head>
 <body>
         <noscript>You need to enable JavaScript to run this app.</noscript>
        <div id="root">
        </div>
</body>
</html>
```

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;
```

Health Records.sol

```
contract HealthRecords {
    struct     PatientRecord     {
        string Name;        address
        patientAddress;      string
        dieses;                  string
        contactInfo;
    }
   mapping(uint256 => PatientRecord) public records;
      event RecordCreated(uint256 indexed recordId, address indexed
patientAddress);
    event  RecordTransferred(   uint256  indexed  recordId,address  indexed  from,
address indexed to );
 modifier onlyOwner(uint256 recordId) {
        require(msg.sender == records[recordId].patientAddress,"Only contract owner
can call this");
        _;
    }

   function                createRecord(
uint256 recordId,
        string memory name, address _patientAddress, string memory _diseases, string
memory _contactInfo
    ) external {
        records[recordId].Name = name;
        records[recordId].patientAddress = _patientAddress;
        records[recordId].dieses = _diseases;       records[recordId].contactInfo
= _contactInfo;
        emit RecordCreated(recordId, _patientAddress);
```

```
    }

    function transferRecord(uint256 recordId, address newOwner) external
onlyOwner(recordId) {
        require(records[recordId].patientAddress == newOwner, "New Owner should
have different Address");
        require(records[recordId].patientAddress == msg.sender, "Only record owner
can transfer");
        records[recordId].patientAddress = newOwner;
  emit RecordTransferred(recordId, records[recordId].patientAddress, newOwner);
    }

    function getRecordData(
uint256 recordId
    ) external view returns (string memory, address, string memory,string memory) {
return (records[recordId].Name, records[recordId].patientAddress,
records[recordId].dieses,  records[recordId].contactInfo);
    }

    function getRecordOwner(uint256 recordId) external view returns (address) {
return records[recordId].patientAddress;
    }
}
```

GITHUB AND PROJECT DEMO LINK:

Github link:https://github.com/Kavishni06/NM-BLOCKCHAIN-DEVELOPMENT-ELECTRONIC-HEALTH-RECORDS
Project demo link:
https://drive.google.com/file/d/1pjzYv_EsdbVPAAo53ByrFBUTqnpUAF58/view?usp=drivesdk