

PROJECT REPORT

Date	25.10.2023
Team id	NM2023TMID01514
Project name	Block chain Technology For Electronic Health Records

1. Introduction:

Project overview:

Implementing blockchain technology for electronic health records (EHRs) to enhance data security, privacy, and interoperability. Key objectives include creating a decentralized ledger for tamper-proof EHR storage, utilizing smart contracts for access control, and improving patient access to their records. The technology stack includes blockchain platforms, encryption, and user-friendly interfaces. The project is divided into phases, with a focus on compliance with healthcare data regulations. The benefits include increased data security, streamlined data sharing, and patient empowerment, while challenges include scalability and user adoption.

Purpose:

The purpose of using blockchain technology for electronic health records

1. Enhance Security: Blockchain ensures data integrity and security by creating a tamper-proof ledger, reducing the risk of unauthorized access and data breaches.
2. Improve Privacy: By encrypting and decentralizing EHRs, patient data privacy is enhanced, giving patients more control over who accesses their health information.
3. Enable Interoperability: Blockchain can facilitate data exchange between different healthcare providers and systems, improving the overall efficiency of healthcare management.
4. Empower Patients: Patients can have easier access to and control over their health records, fostering greater engagement and collaboration in their healthcare.

5. Simplify Data Management: Streamlined and secure data management can reduce administrative burden and enhance the accuracy and accessibility of health records.

6. Ensure Regulatory Compliance: Blockchain can help healthcare organizations comply with data protection regulations like HIPAA and GDPR, which are crucial in the healthcare industry.

2. Literature survey:

Existing problem:

- Scalability
- User Adoption
- Privacy Concerns
- Cost and Energy Consumption
- Interoperability
- Data Recovery
- Technical Expertise
- Legal and Regulatory Challenges
- Smart Contracts

References :

- Blockchain: A Healthcare Perspective ;Barkha Kakkar;Prashant Johri 2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART)
- A Patient-Centric Healthcare Framework Reference Architecture for Better Semantic Interoperability Based on Blockchain, Cloud, and IoT,Ahmad N. Gohar.
- This paper presents A secure and efficient framework based on Blockchain, Cloud, and IoT named Patient-Centric Healthcare Framework (PCH) for better healthcare systems interoperability. A tiered-based architecture (5 tiers) with collaboration is designed for the feasible realization of PCH.
- Digital Secure Storage for Healthcare Data Information Using Blockchain,H.

Naveenkumar;P. Ranjana.2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)

- Evaluating the Impact of Blockchain Models for Secure and Trustworthy Healthcare Records ,Mohammad Zarour.
- Block Chain and Distributed Computing Aided with Cloud Technology- A Specific Reference to Security Issues of Healthcare Industry,C. Murugumani.

Problem statement definit on:

Blockchain technology for Electronic Health Records (EHR) aims to leverage the benefits of blockchain, such as security, transparency, and data integrity, to revolutionize the healthcare industry. However, the successful integration of blockchain into EHR faces significant challenges that need to be addressed. This problem statement defines these challenges and provides a clear understanding of the issues that must be overcome to ensure the effective implementation of BlockchainH in EHR systems.

The aim of this project is to develop a blockchain technology for electronic health records that addresses the following key challenges are;

Data Integrity and Security

Interoperability

Patron Privacy

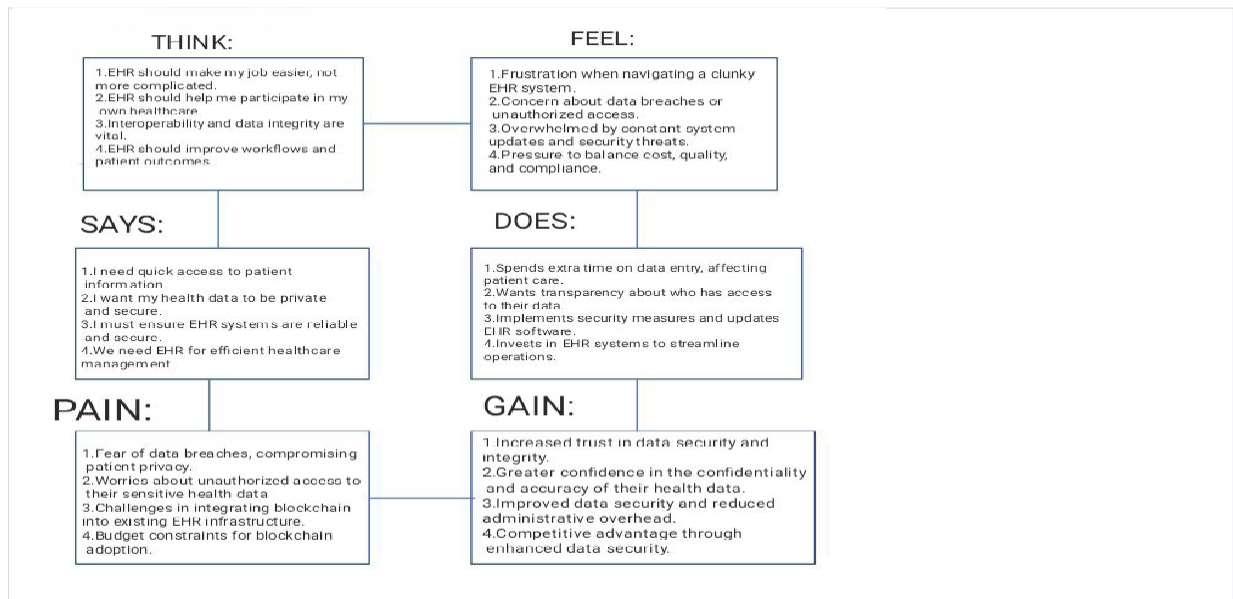
Decentralization and RedundancyUser

Experience

Scalability and Cost

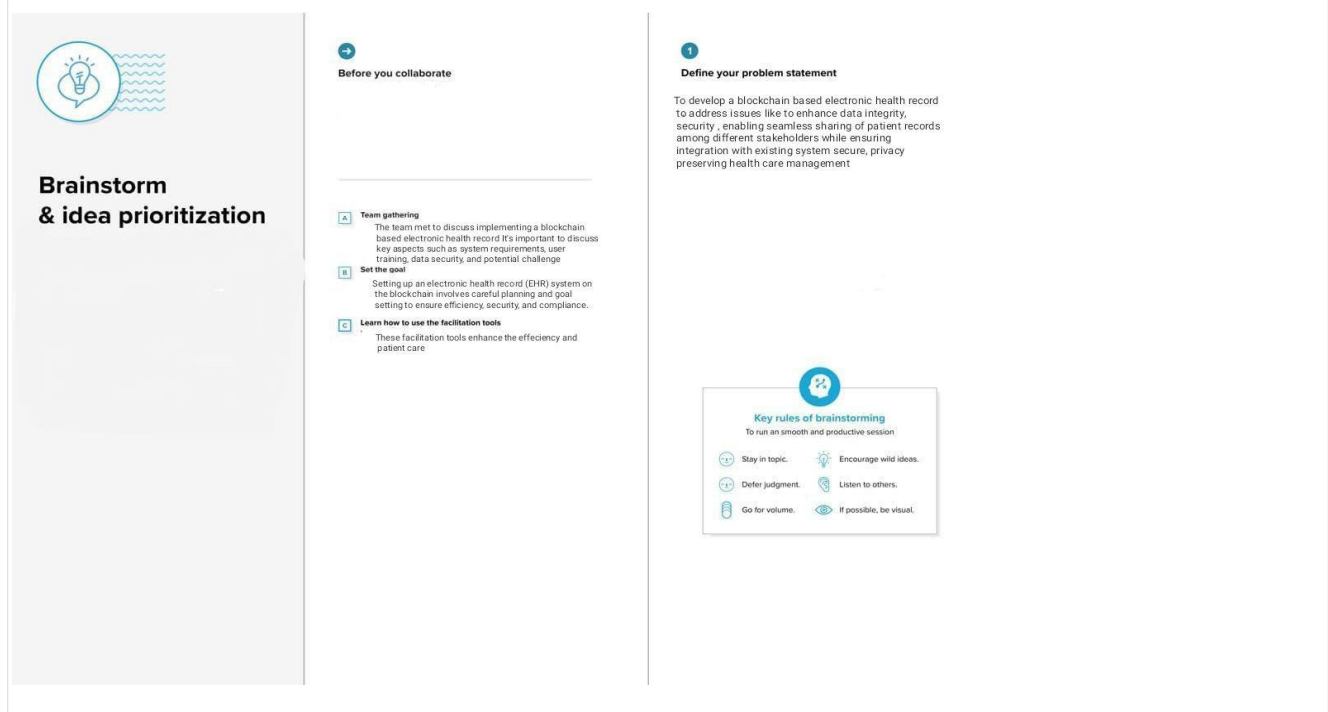
3. Ideation and proposed solution:

Empathy map:




Brainstorming

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



Step-2:Brainstorm,Idea List ng and Grouping




Brainstorm & idea prioritization

Before you collaborate

- A Team gathering**
The team met to discuss implementing a blockchain based electronic health record. It's important to discuss key aspects such as system requirements, user training, data security, and potential challenge.
- B Set the goal**
Setting up an electronic health record (EHR) system on the blockchain involves careful planning and goal setting to ensure efficiency, security, and compliance.
- C Learn how to use the facilitation tools**
These facilitation tools enhance the efficiency and patient care.

Define your problem statement

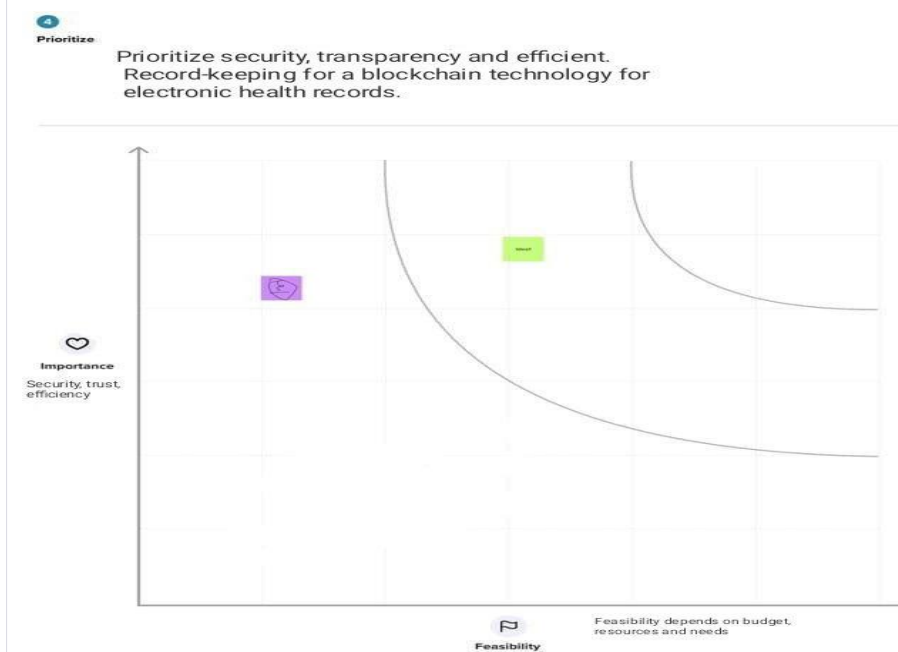
To develop a blockchain based electronic health record to address issues like to enhance data integrity, security, enabling seamless sharing of patient records among different stakeholders while ensuring integration with existing system secure, privacy preserving health care management



Key rules of brainstorming
To run a smooth and productive session

Stay in topic.	Encourage wild ideas.
Defer judgment.	Listen to others.
Go for volume.	If possible, be visual.

Step-3:Idea Prioritization



4. Requirement analysis:

Functional Requirements:

1. User Authentication and Access Control:

- Users must be able to securely log in with strong authentication methods.
- Role-based access control should restrict access to EHRs based on user roles.

2. EHR Data Storage:

- The system should securely store various types of EHR data, including patient demographics, medical history, diagnoses, and treatment records.

3. Blockchain Data Management:

- Create, update, and delete operations on EHR data should be recorded on the blockchain.
- The system should ensure data immutability and transparency.

4. Smart Contracts:

- Implement smart contracts to automate access control, consent management, and data sharing.
- Define the conditions under which data can be accessed or shared.

5. Data Sharing and Interoperability:

- Enable secure data sharing between healthcare providers, patients, and authorized parties.

- Support industry standards (e.g., HL7 FHIR) for interoperability.

6. Patient Portal:

- Provide a user-friendly interface for patients to access and manage their EHRs.
- Patients should be able to grant and revoke access to their data.

7. Audit Trails:

- Maintain detailed audit trails of all interactions with EHR data.
- Include timestamps, user IDs, and transaction records.

8. Consent Management:

- Allow patients to provide informed consent for data sharing.
- Patients should have the ability to specify the scope and duration of consent.

9. Data Encryption:

- Implement encryption protocols to protect EHR data both in transit and at rest.
- Ensure compliance with data protection regulations.

10. Data Recovery:

- Develop mechanisms for data recovery in case of system failures.
- Implement backups and redundancy.

Non-Functional Requirements:

1. Security:

- The system must guarantee data security and privacy, ensuring compliance with healthcare data regulations (e.g., HIPAA, GDPR).

2. Performance:

- The blockchain should handle a high volume of transactions and data storage efficiently.
- Response times for data retrieval and smart contract executions should be fast.

3. Scalability:

- The system should be scalable to accommodate the increasing volume of EHR data and users.
- Scalability solutions like sharding or sidechains may be necessary.

4. Reliability:

- The system should be available 24/7 with minimal downtime.
- Implement robust disaster recovery and redundancy measures.

5. Interoperability:

- Ensure the system can seamlessly integrate with existing EHR systems used by healthcare providers.
- Conform to interoperability standards to facilitate data exchange.

6. Compliance:

- Adhere to healthcare data regulations, and allow for regular audits to demonstrate compliance.

7. User Experience (UX):

- Provide an intuitive and user-friendly interface for healthcare professionals and patients.
- Ensure a positive user experience to encourage adoption.

8. Testing and Quality Assurance:

- Conduct comprehensive testing, including security assessments and penetration testing, to identify vulnerabilities.
- Verify system performance under various conditions.

9. Training and User Adoption:

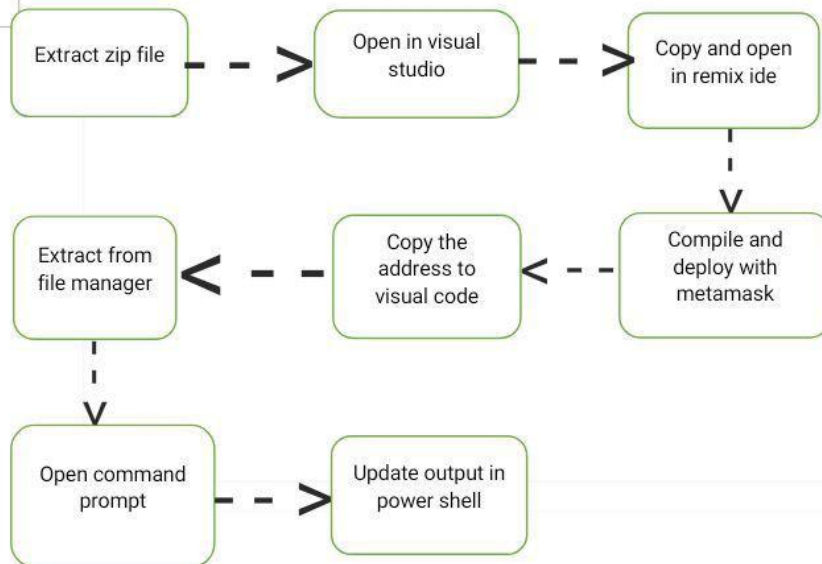
- Develop a training plan to educate healthcare providers and patients on using the system effectively.
- Encourage user adoption through clear communication and support.

10. Cost and Resource Efficiency:

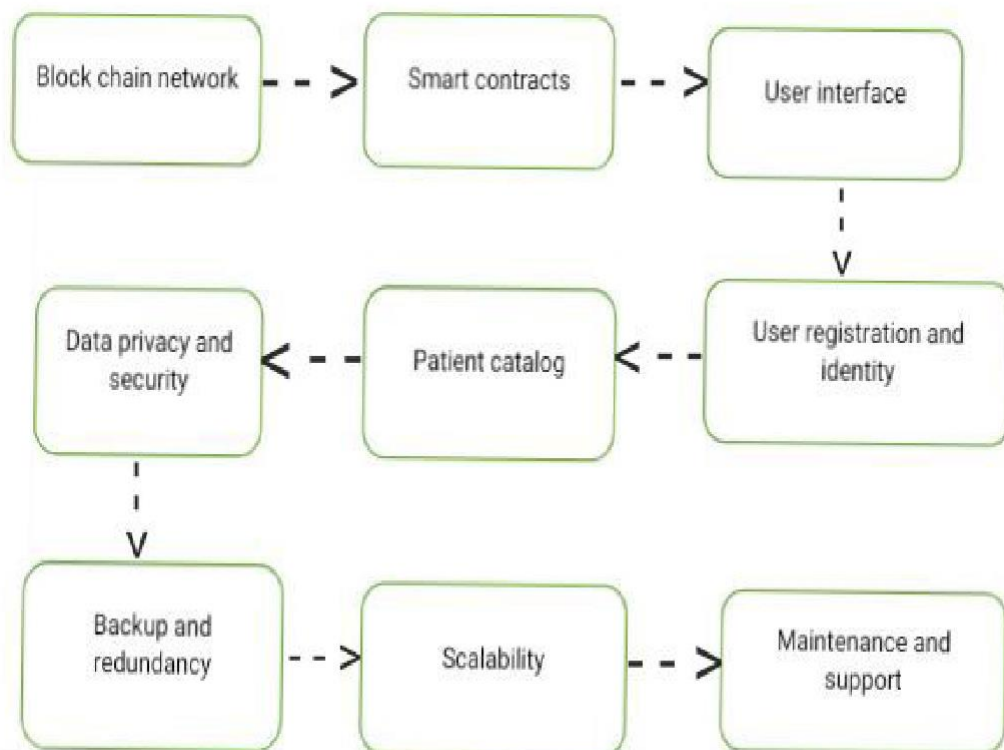
- Budget constraints and resource availability should be considered in system design and development.
- Aim for an efficient use of resources while achieving project goals.

5. Project design:

Data flow diagrams and user stories:

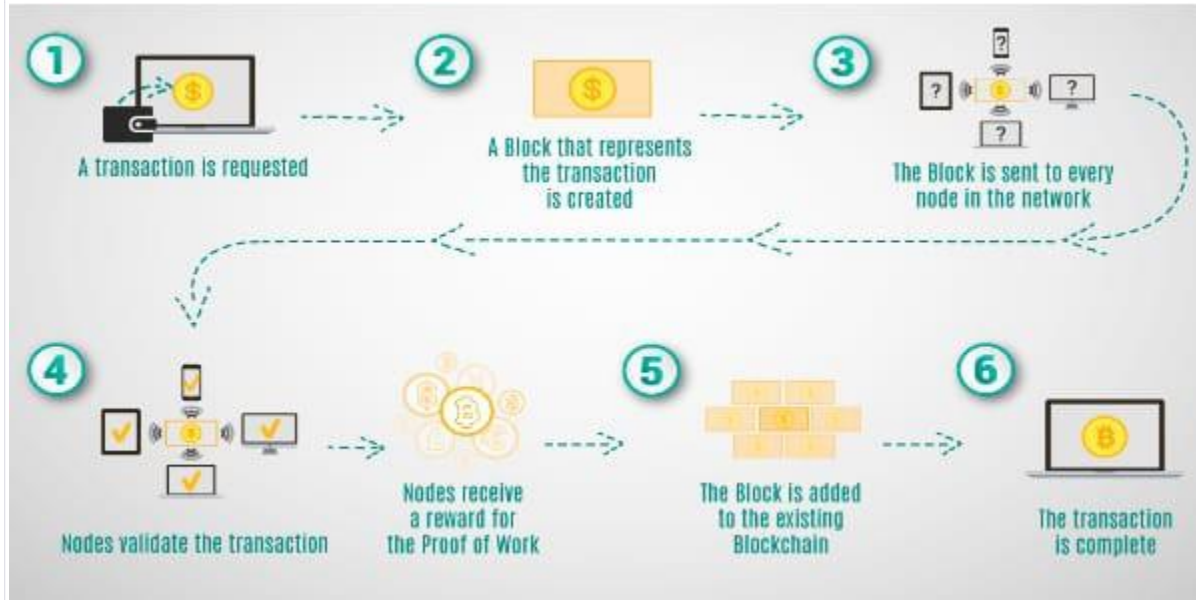


Solut on architecture:



6. Project planning and scheduling

Technical architecture:



Sprint planning and estimation:

Tools required are installed before deadlines. The tools are meta mask, visual studio and node JS

Remix code were built

Using file connector.js

All requirements were built before deadlines

Sprint delivery schedule:

Team members do all steps in developing the project Each

member are separated to do different works Works are done

on date fixed

7. Coding and solut oning:

Feature1

SPDX License Identifier: This is a comment indicating the SPDX license identifier for the contract. In this case, it's specified as MIT, which is a permissive open-source license.

Solidity Version Pragma: The contract specifies the Solidity version to be used, in this case, version 0.8.0. This pragma ensures that the contract is compiled using the specified version of the Solidity compiler.

Feature 2:

Owner Management:

The contract has an owner variable that represents the owner's Ethereum address. The owner is set to the address that deploys the contract in the constructor.

There's a only Owner modifier that restricts certain functions to be callable only by the owner. This modifier checks if the sender of the transaction is the owner before allowing the function to proceed.

8. Performance test ng:

Performance metrics:

1. Transaction Throughput:

- Measure the number of transactions the blockchain can handle per second. High throughput is essential to support the volume of EHR data and user interactions.

2. Latency and Response Time:

- Evaluate the time it takes for a transaction or data retrieval request to be processed. Low latency and fast response times are critical for user satisfaction.

3. Scalability:

- Assess how well the system scales as the volume of EHR data and

users increases. Monitor the blockchain's ability to handle increased loads without a significant decrease in performance.

4. Consensus Mechanism:

- Analyze the energy and computational requirements of the chosen consensus mechanism (e.g., Proof of Work, Proof of Stake). Evaluate its impact on performance.

5. Data Storage and Retrieval Speed:

- Measure the time it takes to store and retrieve EHR data from the blockchain. Fast data access is essential for timely patient care.

6. Smart Contract Execution Time:

- Evaluate the time required for smart contracts to execute, especially those related to access control and consent management. Efficient contract execution is crucial.

7. Block Confirmation Time:

- Determine the time taken for a block to be added to the blockchain. A shorter confirmation time enhances transaction speed.

8. Blockchain Size and Storage Efficiency:

- Monitor the growth of the blockchain size and assess storage efficiency. Efficient data storage is essential for long-term usability.

9. Security Metrics:

- Measure the effectiveness of security measures, including the prevention of unauthorized access, data breaches, and fraud.

10. Availability and Uptime:

- Ensure that the blockchain system is highly available with minimal downtime. Monitor system uptime to prevent interruptions in healthcare

operations.

11. Fault Tolerance and Disaster Recovery:

- Assess the system's ability to recover from failures and ensure data integrity in case of system disruptions.

12. Interoperability:

- Evaluate the ease of integration with existing EHR systems used by healthcare providers. Ensure data exchange occurs smoothly.

13. Consent Management Efficiency:

- Monitor the time it takes to update and manage consent settings for data sharing. Efficient consent management is crucial for patient privacy.

14. Compliance and Auditing:

- Ensure that the system meets healthcare data regulations (e.g., HIPAA, GDPR). Regular audits and compliance checks should confirm adherence.

15. User Adoption and User Satisfaction:

- Gather feedback from healthcare providers and patients to gauge user satisfaction and adoption rates. A positive user experience is vital.

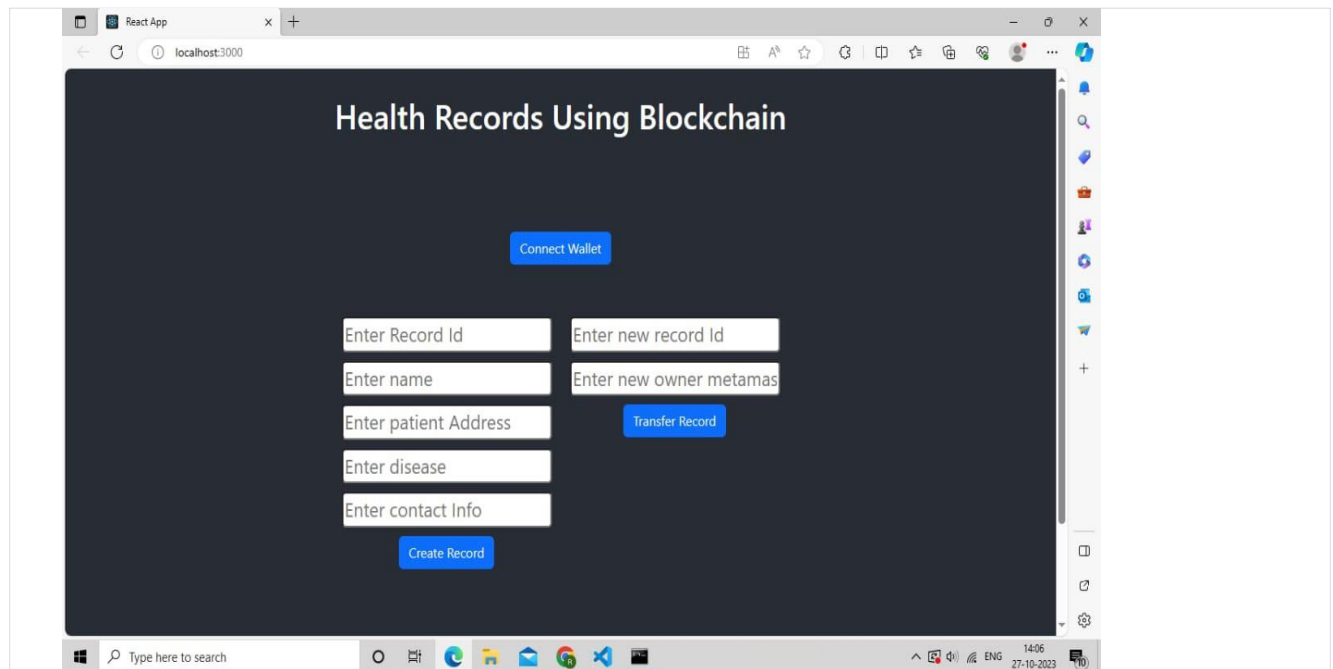
16. Energy Efficiency:

- Evaluate the energy consumption of the blockchain network, especially if using a Proof of Work consensus mechanism. Seek ways to minimize environmental impact.

17. Resource Utilization:

- Monitor resource utilization, such as CPU, memory, and network usage, to optimize resource allocation and minimize wastage.

9.Results:



10.Advantages and Disadvantages:

Advantage:

1. Enhanced Security: Blockchain provides robust security through cryptographic hashing and decentralized data storage, reducing the risk of data breaches and unauthorized access.
2. Data Privacy: Patients can have greater control over who accesses their EHRs and grant or revoke consent for data sharing, improving data privacy and compliance with regulations.
3. Data Integrity: Once recorded on the blockchain, EHR data is immutable, making it tamper-proof and ensuring the integrity of medical records.
4. Interoperability: Blockchain can facilitate data exchange and interoperability between different healthcare providers and systems, streamlining healthcare management.
5. Transparency: The transparent nature of the blockchain ensures that all authorized parties can view the same data, reducing errors and improving the quality of care.

6. **Efficient Data Sharing:** Smart contracts can automate access control and consent management, making data sharing more efficient and reducing administrative overhead.
7. **Patient Empowerment:** Patients have direct access to and control over their health records, enabling them to participate more actively in their healthcare decisions.
8. **Auditability:** Detailed audit trails and transaction records on the blockchain can assist in compliance with healthcare regulations and auditing processes.

Disadvantages:

1. **Scalability Challenges:** Blockchain networks can face scalability issues as they grow, potentially leading to slower transaction processing and higher costs.
2. **Complexity:** Implementing blockchain in healthcare requires understanding complex technologies and regulatory requirements, which can be challenging.
3. **Energy Consumption:** Some blockchain systems, particularly those using Proof of Work consensus, can be energy-intensive, potentially contributing to environmental concerns.
4. **Adoption Hurdles:** Convincing healthcare providers and institutions to adopt blockchain technology for EHRs can be a significant challenge, as it requires changes to established processes and systems.
5. **Regulatory Compliance:** Achieving and maintaining compliance with healthcare data regulations like HIPAA and GDPR is a complex task, and blockchain solutions must ensure adherence.
6. **Costs:** Implementing and maintaining a blockchain-based EHR system can be costly, including development, infrastructure, and ongoing maintenance expenses.
7. **Data Recovery:** In the case of lost private keys or errors in smart contracts.

11. Conclusion:

A blockchain technology for electronic health records (EHRs) holds great promise in revolutionizing the healthcare industry. By enhancing data security, privacy, and interoperability, it addresses critical issues that have persisted in traditional EHR systems. Patients gain greater control over their health information, and healthcare providers benefit from streamlined data access and sharing. However, the adoption of blockchain technology in healthcare comes with challenges, including scalability,

complexity, and regulatory compliance.

Successful implementation of blockchain for EHRs requires a well-thought-out strategy, robust security measures, and a user-friendly interface. It also necessitates ongoing vigilance in adhering to healthcare data regulations. Despite the challenges, the potential for improved patient care, data integrity, and transparency makes blockchain technology a compelling solution for the future of electronic health records. As the technology matures and adoption increases, it has the potential to significantly transform healthcare management for the better.

12.Future scope:

- Block chain can facilitate increased transparency alongside reducing the costs and risks associated with supply chain management.
- Better end-to-end traceability of materials in the supply chain for compliance with corporate standards.
- Limitations on product counterfeiting and fraud.
- Improved transparency and visibility into outsourced contract management with better control over compliance.
- Lower paperwork and reduction in administrative costs.

13.Appendix:

Source code:

```
// SPDX-License-Identifier: MIT

pragma solidity ^0.8.0; 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, stringmemory
_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].diseases,
        records[recordId].contactInfo);
    }

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

```

