ELECTRONIC HEALTH RECORDS

Abstract:

Blockchain has been an interesting research area for a long time and the benefits it provides have been used by several various industries. Similarly, the healthcare sector stands to benefit immensely from blockchain technology due to security, privacy, confidentiality and decentralization. Nevertheless, the Electronic Health Record (EHR) systems face problems regarding data security, integrity and management. In this paper, we discuss how blockchain technology can be used to transform the EHR systems and could be a solution to these issues. We present a framework that could be used for the implementation of blockchain technology in the healthcare sector for EHR. The aim of our proposed framework is firstly to implement blockchain technology for EHR and secondly to provide secure storage of electronic records by defining granular access rules for the users of the proposed framework. Moreover, this framework also discusses the scalability problem faced by blockchain technology in general via use of off-chain storage of the records. This framework provides the EHR system with the benefits of having a scalable, secure and integral blockchain-based solution.

1. INTRODUCTION

1.1 Project Overview

The recent advent in technology is affecting all parts of human life and is changing the way we use and perceive things previously. Just like the changes technology has offered in various other sectors of life, it is also finding new ways for improvement in healthcare sector. The main benefits that advancement in technology is offering are to improve security, user experience and other aspects of healthcare sector. These benefits were offered by Electronic Health Record (EHR) and Electronic Medical Record (EMR) systems. However, they still face some issues regarding the security of medical records, user ownership of data, data integrity etc. The solution to these issues could be the use of a novel technology, i.e., Blockchain. This technology offers to provide a secure, temperproof platform for storing medical records and other healthcare-related information.

Before the advent of modern technology, healthcare sector used paper based system to store the medical records, i.e., using handwritten mechanism. This paper-based medical record system was inefficient, insecure, unorganized and was not temper-proof. It also faced the issue of data- duplication and redundancy as all the institutions that patient visited had various copies of patient's medical records.

The healthcare sector faced a trend shift towards EHR systems that were designed to combine paper-based and electronic medical records (EMR). These systems were used to store clinical notes and laboratory results in its multiple components. They were proposed to enhance the safety aspect of the patients by preventing errors and increasing information access . The goal of EHR systems was to solve the problems faced by the paper-based healthcare records and to provide an efficient system that would transform the state of healthcare sector .

The EHR systems have been implemented in a number of hospitals around the world due the benefits it provides, mainly the improvement in security and its cost-effectiveness. They are considered a vital part of healthcare sector as it provides much functionality to the healthcare [4]. These functionalities are electronic storage of medical records, patients' appointment management, billing and accounts, and lab tests. They are available in many of the EHR system being used in the healthcare sector. The basic focus is to provide secure, temper-proof, and shareable medical records across different platforms. Despite the fact that notion behind usage of EHR systems in the hospitals or healthcare was to improve the quality of healthcare, these systems faced certain problems and didn't meet the expectations associated with them [3]. A study was conducted in Finland to find the experiences of nursing staff with the EHR, it was concluded that EHR systems faced the problems related to them being unreliable and having a poor state of user-friendliness.

1.2 Purpose

A. Interoperability

It is the way for different information systems to exchange information between them. The information should be exchangeable and must be usable for further purposes. An important aspect of EHR systems is its Health Information Exchange (HIE) or in general data sharing aspect. With a number of EHR systems being deployed in various hospitals they have a varying level of terminologies, technical and functional capabilities which makes it to have no universally defined standard. Moreover, at technical level the medical records being exchanged should be interpretable, and that interpreted piece of information could be further used.

B. Information Asymmetry

Today the greatest problem in healthcare sector defined by the critics is information asymmetry which refers to one party having better access to information than the other party. In case of EHR systems, or in general healthcare sector is suffering from this problem as doctors or hospitals have access to the patient's records, thus making it central. If a patient wants to access his medical records he would have to follow a long and tedious process to access them. The information is centralized to only a single healthcare organization and its control is only provided to the hospitals or organizations.

C. Data Breaches

Data breaches in healthcare sector also calls for the need of a better platform. A study was done for analyzing the data breaches in EHR systems and it depicted that 173 million data entries have been compromised in these systems since October 2009. Another study conducted by Argaw *et al.* explains that hospitals have become a target of cyber-attacks and an increasing trend has been witnessed by the researchers while conducting this study that a lot of research work has been done in this domain.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

In the existing system the records are stored and maintained under the organization. So that, the patient can't able to access these records for further references. When the particular server(database) gets crashed then all the records will be spoiled. To overcome these drawbacks the proposed system is developed. Electronic Records (EHRs) provide a convenient record storage service, which promotes traditional patient medical records on paper to be electronically accessible on the web. In the current situation, patients scatter their EHRs across the different areas during life events, causing the EHRs to move from one service provider database to another. Therefore, the patient may lose control of the existing healthcare data, while the service provider usually maintains the primary stewardship. Patient access permissions to EHRs are very limited, and patients are typically unable to easily share these data with researchers or providers. Interoperability challenges between different providers, hospitals. The patient should have right to access his EHRs for managing and sharing them independently Institutions, etc.

2.2 References

- G. Jetley and H. Zhang, "Electronic health records in IS research: Quality issues essential thresholds and remedial actions", *Decis. Support Syst.*, vol. 126, pp. 113-137, Nov. 2019.

- K. Wisner, A. Lyndon and C. A. Chesla, "The electronic health record's impact on nurses' cognitive work: An integrative review", *Int. J. Nursing Stud.*, vol. 94, pp. 74-84, Jun. 2019.

- T. Vehko, H. Hyppönen, S. Puttonen, S. Kujala, E. Ketola, J. Tuukkanen, et al., "Experienced time pressure and stress: Electronic health records usability and information technology competence play a role", *BMC Med. Inform. Decis. Making*, vol. 19, no. 1, pp. 160, Aug. 2019.

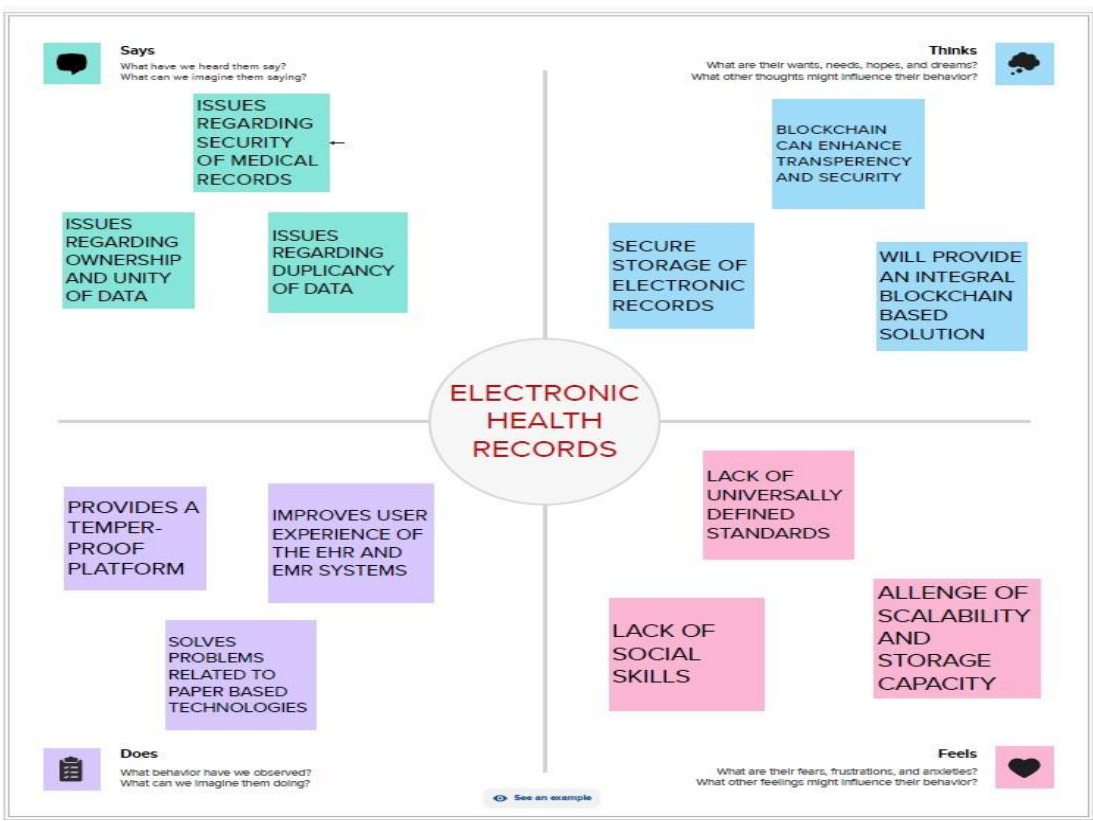
- W. W. Koczkodaj, M. Mazurek, D. Strzałka, A. Wolny-Dominiak and M. Woodbury-Smith, "Electronic health record breaches as social indicators", *Social Indicators Res.*, vol. 141, no. 2, pp. 861-871, Jan. 2022.

2.3 Problem Statement Definition

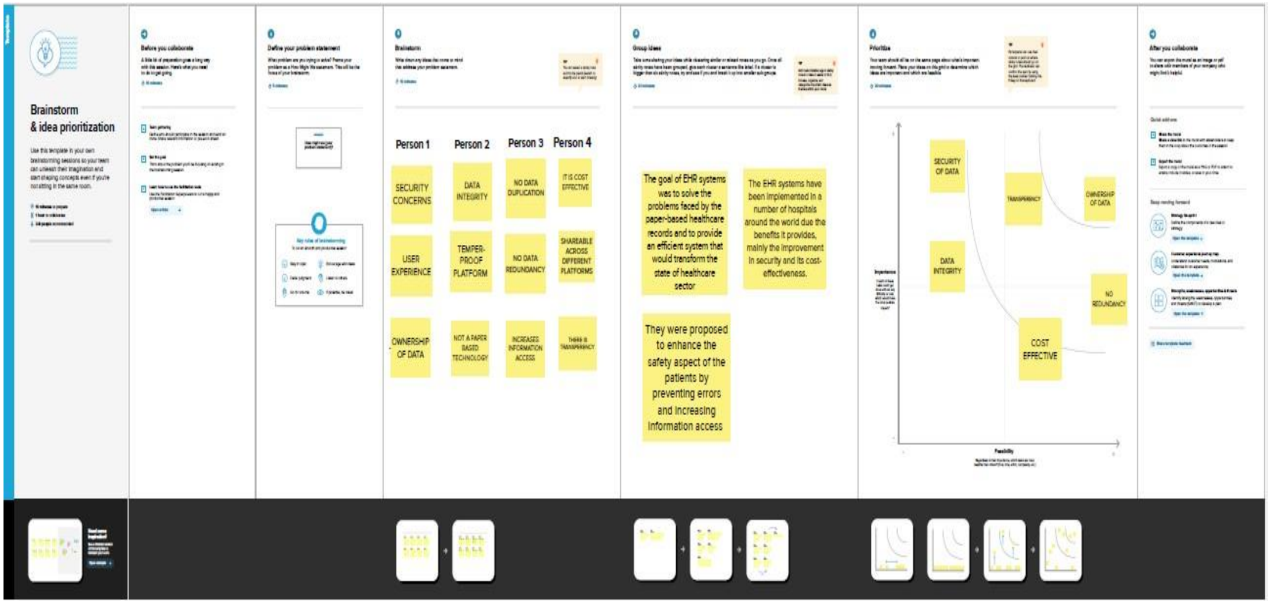
The recent advent in technology is affecting all parts of human life and is changing the way we use and perceive things previously. Just like the changes technology has offered in various other sectors of life, it is also finding new ways for improvement in healthcare sector. The main benefits that advancement in technology is offering are to improve security, user experience and other aspects of healthcare sector. These benefits were offered by Electronic Health Record (EHR) and Electronic Medical Record (EMR) systems. However, they still face some issues regarding the security of medical records, user ownership of data, data integrity etc. The solution to these issues could be the use of a novel technology, i.e., Blockchain. This technology offers to provide a secure, temper-proof platform for storing medical records and other healthcare related information.

3. IDEATION AND PROPOSED SOLUTIONS

3.1 Empathy Map Canvas



3.2 Ideation and Brainstorming



4. REQUIREMENT ANALYSIS

4.1 Functional Requirements

Analyzing the requirements for implementing electronic health records (EHR) using blockchain technology is a critical step to ensure the success and security of such a system. Here are some related points for EHR using blockchain requirement analysis.

Data Security and Privacy: Ensure that patient health data is secure and private, with strict access control mechanisms and encryption. Specify the need for HIPAA compliance or other relevant data protection regulations.

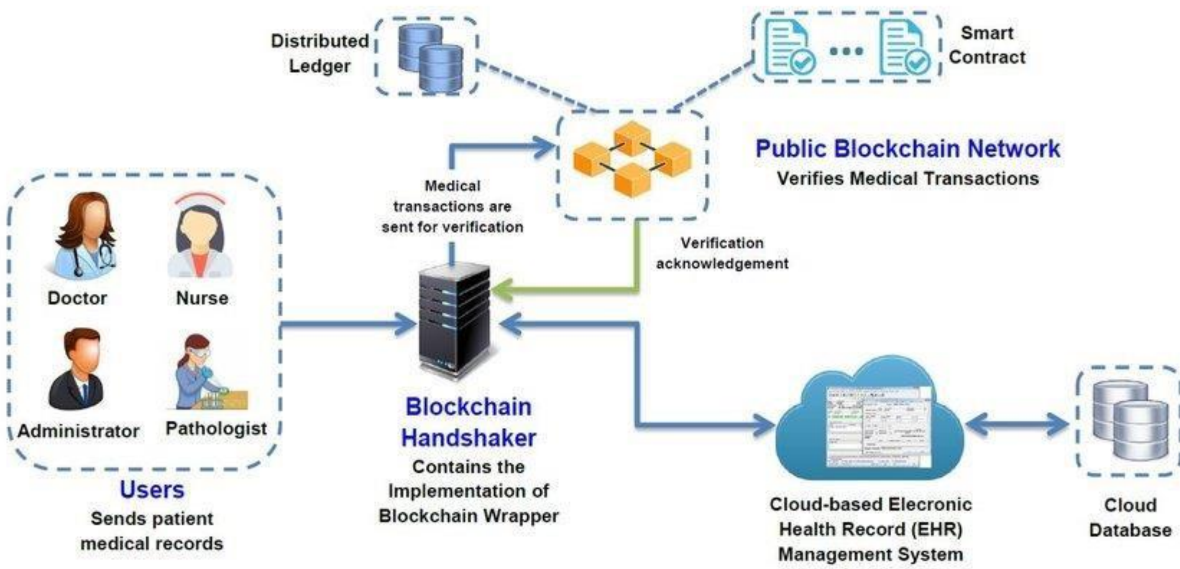
Immutable Data: Identify the requirement for maintaining an immutable ledger of health records. This ensures data integrity and auditability.

Interoperability: Specify the need for data interoperability standards to enable the exchange of health information between different healthcare providers and systems.

Scalability: Assess the scalability requirements for the blockchain network to handle a growing volume of health records and transactions.

5. PROJECT DESIGN

5.1 Data Flow Diagram & User Stories



The Healthcare Blockchain System framework is designed for remote monitoring of patients and access to their health records. It uses a private blockchain based on the Ethereum protocol to protect patients' privacy . At the same time, it maintains a secure record of who has initiated activities on the blockchain and provides details about every data transaction. Notifications are delivered to all involved parties to address security vulnerabilities in remote patient monitoring. Combined with blockchain, MPC (Secure Multi-Party Computing), Indicator-Centric Schema (ICS), and Healthcare Data Gateway (HGD) frameworks all guarantee patients' rights to own, control, and share their own data easily and securely. MeDShare employs smart contracts and access control to track the behaviors of patients' privacy data and detect the possible invasion of patients' privacy. More specifically, MeDShare closely monitors all actions performed on the data through smart contracts and keys attached to the contracts, so that if a malicious user tries to steal data privacy or tamper with reports generated by the smart contract, their actions will be exposed, and access will be restricted or even revoked. In addition to the above solutions, there are many more studies using blockchain technology to store, track, and manage medical records. Researchers propose an IoMT-based platform for EHR based on the blockchain. The method combines IoMT (Internet of Medical Things) and blockchain to encrypt and save the user's health information. First, multiple smart sensors collect the user's health recording, and then encrypted health data will be stored in the nodes of the Ethereum blockchain, thus protecting the privacy of users.

User Stories:

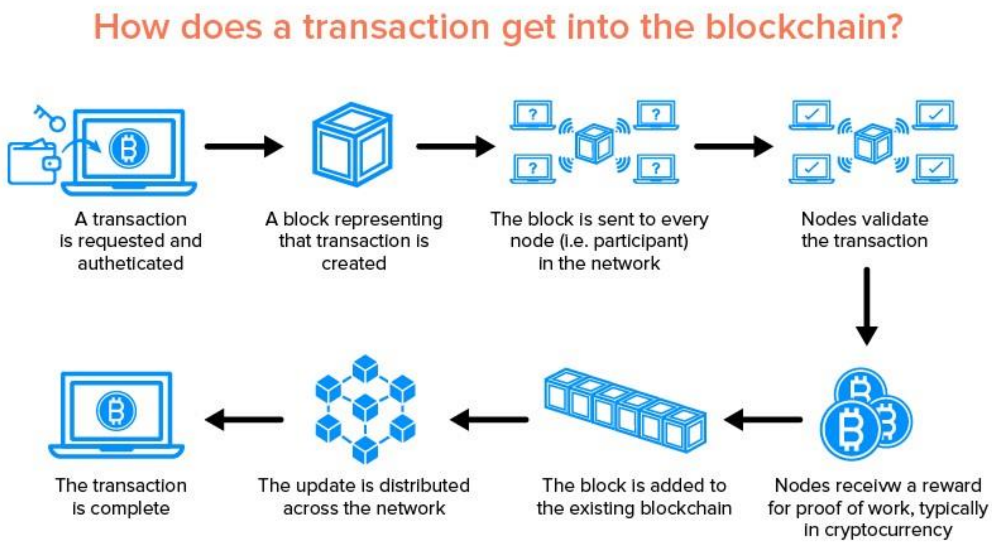
User Story 1:

I am a General Practitioner. I want to change the patient's information entered on the system to correct inaccuracies in the report. The information was noted by the nurse when the patient was admitted. The patient's name is misspelled by the nurse. The patient forgot to mention a health condition they were treated for several years back. Both these errors have to be corrected on the system to maintain an accurate record of the patient's history.

User Story 2:

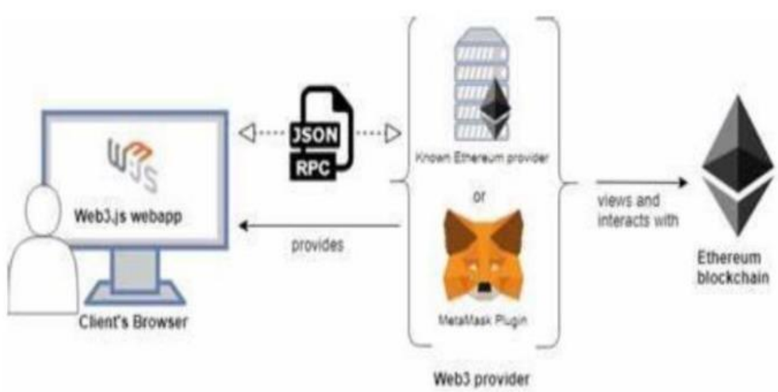
I am a patient. I want to provide my physician with regular updates on my sugar test results, so he can monitor my health progress. I conduct these tests at home. My doctor has to be reported about the test results on a monthly basis to monitor my health progress. If my sugar test results indicate an imbalance, the doctor will call me for a visit. I want to receive this notification from my doctor in advance, so I can schedule my visit conveniently.

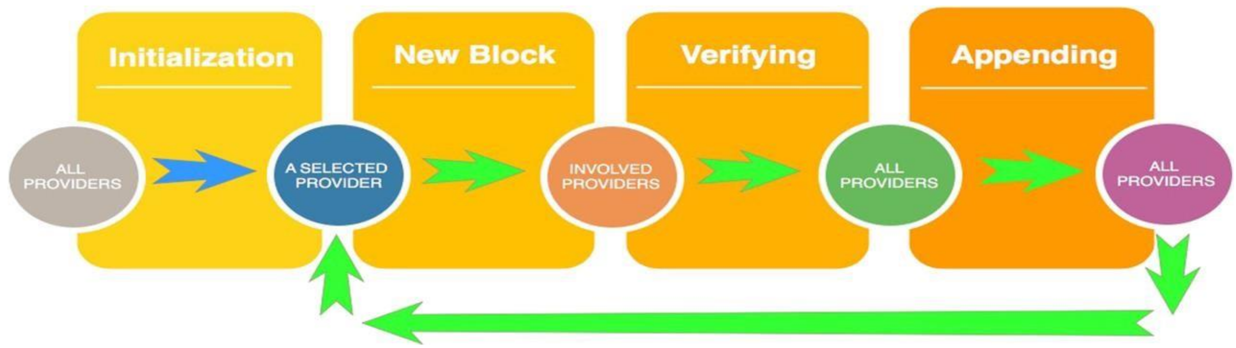
5.2 Solution Architecture



6. PROJECT PLANNING AND SCHEDULING

6.1 Technical Architecture





6.2 Sprint Planning & Estimation

- Intervention:

The Sprints intervention had 3 primary components: (1) training clinicians to use existing EHR features more efficiently, (2) redesigning the multidisciplinary workflow within the clinic, and (3) building new specialty specific EHR tools. Sprint was a quality improvement intervention, and we prioritized continuous process improvement over consistent data collection. Thus, we adjusted some survey questions over time to provide better insight, and thus the total number of respondents to some questions will differ from others. For each of the 6 clinics, the most updated survey at the beginning of that Sprint was sent to *all* clinicians in that clinic both before and after Sprint.

- Agile :

We use Agile methodology as a guiding strategy for Sprint. Agile methodology improves software delivery by focusing on the voice of the customer and making rapid incremental improvements in a short time period. We held daily huddles and prioritized tasks on the basis of feedback from clinicians and clinic staff. We used scrum boards to depict progress in fulfilling requested EHR changes. When possible, we facilitated conversations between specialists in the clinic undergoing Sprint and specialists at similar clinics in our organization to build consensus on specialty specific EHR tools.

- Building new EHR Tools:

Once EHR tool requests were approved by clinic leaders, the Sprint project manager placed them in a visual chart under the following headings: Backlog, To Do, In Process, Done, and Parking Lot. These items were reprioritized and repositioned daily. In this way, clinicians and clinic staff could see the status of all requests, including Parking Lot items that would not be addressed during Sprint.

- Multidisciplinary Workflow Design:

Some Sprint team members met individually and in small groups with Non clinician staff to observe the multidisciplinary workflow and patient flow in the clinic. Using this information, trainers taught EHR best practices to staff, and the clinic manager redesigned common workflows so that all clinicians and staff did things the same way. For examples of redesigned workflows, see the Discussion section.

6.3 Sprint Delivery Schedule

Week 1: Establishing the core

- Set up the basic blockchain infrastructure.
- Implement a minimal user registration and an authentication system.
- Focus on the basic security measures to be carried out or needed to be performed.

Week 2: Expand and Enhance

- Extend health recording to support more recording types.
- Add basic updates to show recording status of the health.
- Enhance authentication and security measures.
- Begin developing a dashboard.

Week 3 : Finalize and Prepare

- Conduct basic testing and issue resolution.
- Complete the patients dashboard with additional features for better interactivity.

7. CODING & SOLUTIONING

7.1 Feature 1

Smart Contract(Solidity)

```
   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;
    }
}
```

The main language used in Ethereum is solidity – which is a JavaScript-like language developed specifically for writing smart contracts. Solidity is statically typed, supports inheritance, libraries and complex user-defined types among other features.

The solidity compiler turns code into EVM bytecode, which can then be sent to the Ethereum network as a deployment transaction. Such deployments have more substantial transaction fees than smart contract interactions and must be paid by the owner of the contract.

7.2 Feature 2 Front

end(JavaScript)

```
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>
) }
e
x
p
o
rt
d
ef
a
ul
t
H
o
m
e;
```

8. PERFORMANCE TESTING

8.1 Performance Metrics

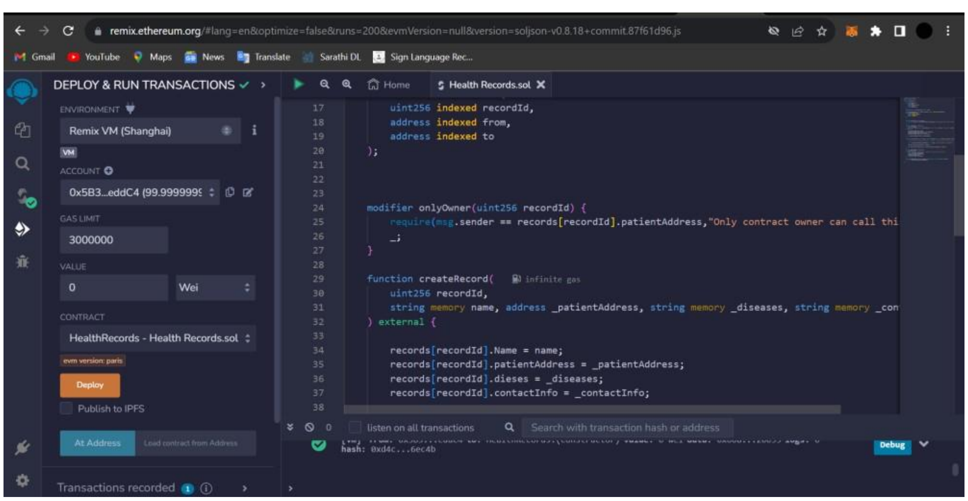
To understand the performance of an EHR we need to look at performance metrics:

- Performance metrics should be determined with input from all stakeholders
- Performance metrics need to be both meaningful and agreed upon by all stakeholders.
- For performance metrics to be meaningful they should be reasonable, collectable, and measurable.
- Performance metrics should be real time (network/server monitoring) or near real time (EUD metrics)
- Performance metrics should be analyzed and acted upon to be meaningful
- Performance metrics center around availability, reliability, and latency (speed)

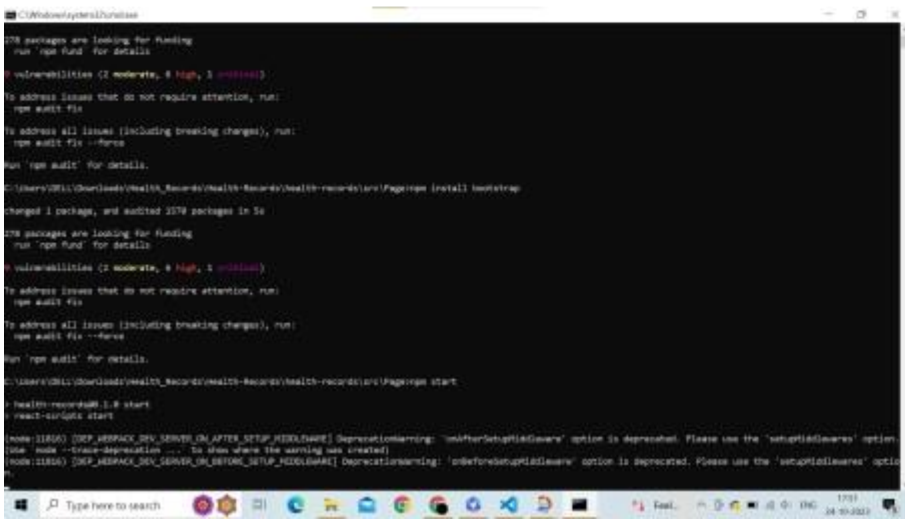
9. RESULTS

9.1 Output Screenshots

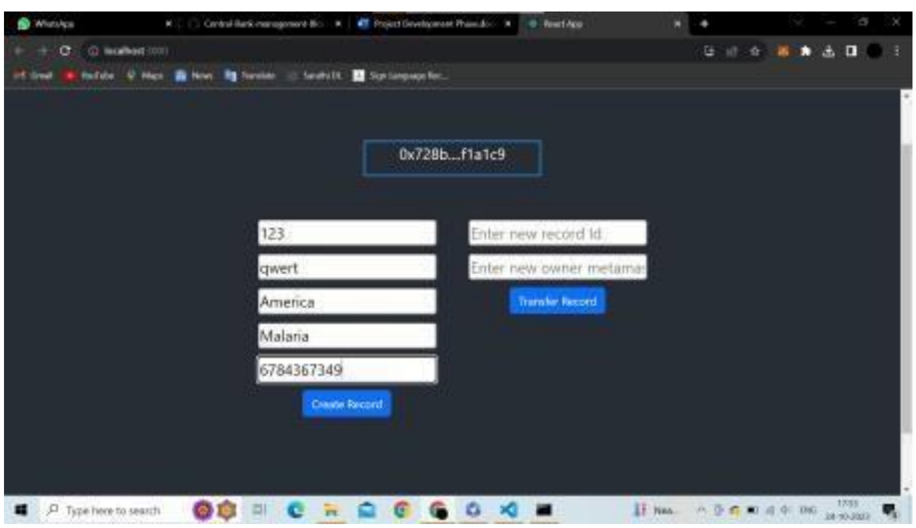
SMART CONTRACT



INSTALLING DEPENDENCIES



OUTPUT SCREEN



10. ADVANTAGES AND DISADVANTAGES

10.1 Advantages

- The main benefits that advancement in technology is offering are to improve security, user experience and other aspects of healthcare sector. These benefits were offered by Electronic Health Record (EHR) and Electronic Medical Record (EMR) systems.

- This technology offers to provide a secure, temper-proof platform for storing medical records and other healthcare-related information.

- Before the advent of modern technology, the healthcare sector used paperbased systems to store medical records, i.e., using handwritten mechanisms. This paper-based medical record system was inefficient, insecure, unorganized and was not temper-proof. It also faced the issue of data- duplication and redundancy as all the institutions that patient visited had various copies of patient's medical records.

- The EHR systems have been implemented in several hospitals around the world due to the benefits they provide, mainly the improvement in security and its cost-effectiveness.

10.2 Disadvantages

- Interoperability

An important aspect of EHR systems is its Health Information Exchange (HIE) or in general data sharing aspect. With a number of EHR systems being deployed in various hospitals they have a varying level of terminologies, technical and functional capabilities which makes it to have no universally defined standard

- Information Asymmetry oday the greatest problem in healthcare sector defined by the critics is information asymmetry which refers to one party having better access to information than the other party. In case of EHR systems, or in general healthcare sector is suffering from this problem as doctors or hospitals have access to the patient's records, thus making it central.

- Data Breaches

Data breaches in healthcare sector also calls for the need of a better platform. Hospitals have become a target of cyber-attacks and an increasing trend has been witnessed by the researchers while conducting this study that a lot of research work has been done in this domain.

11. CONCLUSION

In this paper we discussed how blockchain technology can be useful for healthcare sector and how it can be used for electronic health records. Despite the advancement in the healthcare sector and technological innovation in EHR systems they still faced some issues that were addressed by this novel technology, i.e., blockchain. Our proposed framework is a combination of secure record storage along with the granular access rules for those records. It creates such a system that is easier for the users to use and understand. Also, the framework proposes measures to ensure the system tackles the problem of data storage as it utilizes the off-chain storage mechanism of IPFS. And role-based access also benefits the system as the medical records are only available to trusted and related individuals. This also solves the problem of information asymmetry of EHR system.

12. FUTURE SCOPE

For the future, we plan to implement the payment module in the existing framework. For this we need to have certain considerations as we need to decide how much a patient would pay for consultation by the doctor on this decentralized system functioning on the blockchain. We would also need to define certain policies and rules that comply with the principles of the healthcare sector.

Blockchain technology might be a future suitable solution for common problems in the healthcare field, such as EHR interoperability, establishing sharing trust between healthcare providers, auditability, privacy, and granting of health data access control by patients, which would enable them to choose whom they want to trust and with whom to share their medical records. However, additional research, trials, and experiments must be carried out to ensure that a secure and established system is implemented prior to using Blockchain technology on a large scale in healthcare, since a patient's health data are personal, highly sensitive, and critical information.

This study may serve as a basis or inspiration for future works and studies. Our answered research questions and taxonomy may contribute to the proposition of an architecture or model that addresses the challenges that are discussed in this article. In addition, a possible direction for future work is to survey the combination of Blockchain and the Internet of the things (IoT) in healthcare, with the objective of realizing network scalability improvements by supporting low-end devices.

13. APPENDIX

Source Code

Smart Contract

Smart Contract(Solidity)

```
   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;
    }
}
```

Connector.js

```
{
   "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 = "0x4B3F652c74e2faA9C53DF29541F065e0b6bBd039"

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;
```

GitHub:

https://github.com/nitinsatheesan12/Blockchain-Technology-for-ElectronicHealth-Records/tree/main

Project Demo Link:

https://drive.google.com/file/d/1IVEiG6VeJrnYMzOSK2apfgQ07YzgWT92/view?usp=drivesdk