Bulletin of Electrical Engineering and Informatics
Vol. 14, No. 4, August 2025, pp. 3037~3051
ISSN: 2302-9285, DOI: 10.11591/eei.v14i4.9524

Securing patient data and access control in electronic health records with Ethereum blockchain

Shruthi Kumarswamy, Poornima Athikatte Sampigerayappa
Department of Computer Science and Engineering, Siddaganga Institute of Technology Tumakuru, Visvesvaraya Technological University Belagavi, Karnataka, India

Article Info

Article history:

Received Nov 7, 2024
Revised Jun 9, 2025
Accepted Jul 5, 2025

Keywords:

Access control
Blockchain
Decentralized storage
Electronic health records
Peer-to-peer encryption

ABSTRACT

Blockchain technology has become an essential tool for enhancing reliability and security across several industries, including the healthcare sector. In this work, we propose and implement an Ethereum-based blockchain framework to decentralize electronic health records (EHRs) at Tumakuru Siddaganga Hospital. The system establishes an append-only chain of transaction blocks that guarantees the confidentiality, auditability, and integrity of patient health records. By design, only authorized healthcare professionals can access patient data, and even then, only with the patient's explicit consent—ensuring a privacy-preserving access model. Our approach demonstrated a 40% reduction in data access delays and eliminated unauthorized access attempts through smart contract-based access control. The decentralized nature of the framework reduces reliance on centralized databases, significantly lowering the risk of data tampering and breaches. Additionally, the implemented consensus protocol ensures that only verified transactions are recorded, maintaining consistency across distributed nodes. Compared to traditional systems, our blockchain-based solution improved the traceability of health data access events by 100%, ensuring transparency and accountability. These findings validate that blockchain technology can substantially enhance data sharing, integrity, and patient control in modern healthcare systems.

Corresponding Author:

Shruthi Kumarswamy
Department of Computer Science and Engineering, Siddaganga Institute of Technology Tumakuru
Visvesvaraya Technological University Belagavi
Karnataka, India
Email: shruthik@sit.ac.in

1. INTRODUCTION

There has been a recent surge in interest in applying blockchain technology to advance e-health and healthcare services. Blockchain technology has demonstrated considerable promise, namely in the secure interchange of electronic health records (EHRs) and managing data access to electronic medical records (EMRs) among diverse healthcare institutions. Blockchain technology can revolutionize the healthcare sector and speed up the delivery of healthcare. It is a distributed ledger, or blockchain designed to safely store private data throughout all systems within a blockchain network. A copy of this ledger, which comprises the details of every transaction that occurs on the network, is kept on file by every system, or node. A block, the fundamental unit of a blockchain, is composed of three components: its contents, the hash of the previous block, and its unique hash. This hash is specific to the data it contains because it is produced based on the contents of the block. If the data in a block is altered, a new and completely different hash is generated. This alteration breaks the chain because the subsequent block still references the initial value of the hash as its

prior hash. To cover up the tampering, one would need to regenerate new hash values for all subsequent blocks in the chain. However, even if this were accomplished, the tampered ledger would not match the copies on the other nodes in the network. It wouldn't take long for the network to notice this discrepancy, making it very difficult for someone to successfully manipulate the data on the blockchain [1]–[3].

For every block in the chain after that, new hash values would need to be generated to hide the manipulation. Even if this were possible, the modified ledger's copies would not match those on the other network nodes. The network would identify this disparity very quickly, making it very difficult for someone to successfully tamper with the data on the blockchain [4], [5]. A patient's medical record (PMR) is a comprehensive historical record of all the health data gathered throughout time, including medical history, prescriptions, pre and post-operative care, clinical judgments, and test findings. Healthcare providers can effectively treat patients and obtain a thorough knowledge of their current state by carefully reviewing the details in a PMR. The creation, maintenance, and updating of these documents is the responsibility of healthcare practitioners or other medical experts. The PMR is vitally significant throughout critical stages of patient care, such as therapy monitoring, medical research, audits, statistical analysis, insurance claims, and even legal actions. Even though assessing a patient's present state of health depends on knowing a significant amount of sensitive personal information, the PMR must be kept confidential. There are problems with storage, access control, security, and privacy when keeping track of these documents. Hospitals that employ manual, paper-based record-keeping systems usually have operational issues such as data loss and the management of active and inactive records [6].

As part of India's e-hospital strategy, a cloud-based tool called the online registration system (ORS) was launched to address these difficulties. This system is a key component of the Digital India project, led by the Ministry of Electronics and Information Technology (MeitY), aimed at providing accessible healthcare services to all citizens across the country. Thanks to modern technologies like mobile cloud computing (MCC) and the internet of medical things (IoMT), particularly in the area of e-health, the healthcare industry has undergone a tremendous transition. Nowadays, people can use wearable sensors and cell phones to collect their health data at home [7], [8]. In cloud environments, this data can be shared so that doctors can rapidly access it to analyze patient records and give emergency medical attention. By allowing medical staff to remotely monitor patients and give ambulatory care at home which also offers financial benefits to patients this clever e-health technology improves the delivery of healthcare. Keeping an extensive EMR on cloud storage also makes it easier to track patients' health over time and provide the right care at every stage of diagnosis and treatment. Further in Table 1, which provides a comparison with existing works, demonstrating how EMR systems outperform traditional methods in areas such as accessibility, scalability, and continuity of care. Table 1 presents a comparative analysis of existing blockchain-based EMR systems, focusing on features such as IPFS integration, file integrity, smart contract usage, and data security. It highlights that while most solutions ensure high data security and leverage smart contracts, their integration with existing EMR systems and IPFS varies, reflecting ongoing development and differing approaches in implementation.

Table 1. Comparison with existing works

Ref.	Blockchain type	IPFS integration	File integrity	Integration with existing EMR	Smart contract use	Data security focus
[9]	Private	Yes	Hash-based integrity	Not specified	Yes	High
[10]	Private	Yes	Secure via IPFS	Yes	Yes	High
[11]	Private	No	Ensured via blockchain	Planned/discussed	Yes	Very high
[12]	Ethereum	No	Ensured via blockchain	Yes (layered structure)	Yes (CRUD)	High
[13]	Ethereum	Yes	Encrypted and IPFS Hash	Yes (patient-controlled)	Yes (access control)	Very high
[14]	Varied (MedRec, Medblock)	Varies	Blockchain tree for integrity	Discussed (MedRec)	Yes (MedRec)	High
[15]	Ethereum (Ropsten)	Yes	Hashed via blockchain	Yes, for PMRs	Yes	High
[16]	General blockchain use	Mentioned generally	Not explicitly discussed	Conceptual only	Varies	High

Because of the dispersed character of blockchain systems, blockchain-based access control, which reduces risks and trust difficulties and stops data loss, can keep working even if one or more parties fail. Building on these advantages, this study proposes a novel model for sharing EHR on a mobile platform using blockchain technology. Our suggested method controls the access to data by network entities using a user access management architecture. This access control system guarantees that authorized parties can retrieve data quickly while successfully preventing unauthorized users from accessing EHR resources. Patient records are among the most critical assets currently centralized, maintained, and managed by hospitals. While some

countries have transitioned to HER [17], [18], many still rely on traditional methods for storing patient and medical details. In the United States, around 80 to 90% of hospitals have adopted basic EHR systems, but these systems present significant challenges related to privacy, security, and ownership of health records. Even with the adoption of EHRs, hospitals retain full ownership and control of patient records. This means that patients, despite being the rightful owners of their health data, lack direct access and must contact the hospital whenever they need their information. Data consolidation is another major issue, as each hospital stores its EHRs on its server, forcing patients to retrieve records from multiple sources to compile their complete medical history [19], [20].

Moreover, hospitals own the EHRs, they have the authority to alter the data at any time and may even sell it to research organizations for profit. In addition to concerns over patient records, the authenticity of doctors is another challenge, as there is no straightforward way to verify whether a doctor is legitimate. In the proposed scheme we used blockchain technology and designed it to manage patient data while enhancing transparency and accountability. It operates as a distributed ledger of transactions, where identical copies are visible to all members of a computer network. The data entered into the ledger is validated by the network members, and once recorded, it becomes immutable. By leveraging blockchain technology [21], [22], a solution can be developed to store EHR within a distributed and decentralized network. This approach allows patients to retain ownership of their data and grant access to hospitals, doctors, and research organizations [23], [24].

Comparison with similar systems:

- Existing cloud-based healthcare platforms such as the ORS under the digital India initiative have made strides in digitizing patient data, but they suffer from centralized control, which poses risks of single-point failures, data tampering, and unauthorized access [25].
- Platforms using MCC and IoMT have improved remote data collection and monitoring but offer limited granular access control and lack auditable trails for data access.
- Traditional EMR systems store data in isolated silos at hospitals, making interoperability and secure data sharing across institutions difficult, and often exclude patients from ownership or control of their data.

Justification for blockchain and Ethereum:

- Unlike centralized EMR systems, blockchain technology inherently provides decentralization, immutability, and transparency, which ensures that health records are tamper-proof and traceable.
- The Ethereum blockchain specifically supports smart contracts, which allow for the fine-grained access control policies implemented in this work. For example, patients can grant or revoke access to their records dynamically.
- Ethereum's broad adoption, active development community, and mature tooling (e.g., solidity and remix IDE) make it suitable for building secure and programmable access control mechanisms for EHR systems.
- Compared to permissioned blockchains (like hyperledger), Ethereum's public and customizable nature supports real-time consensus validation and can simulate a real-world, diverse healthcare environment in testing phases.
- Therefore, the novelty of this approach lies in designing a patient-centric EHR sharing model on Ethereum that overcomes the privacy, ownership, and interoperability limitations of existing systems.
- By incorporating smart contract-based dynamic access control, this work ensures that only authorized entities can interact with sensitive data, and all interactions are transparent and verifiable on the blockchain.

2. PROPOSED SCHEME

The medical industry requires the strictest privacy and security regulations for sensitive patient data, in addition to the highest requirements of data completeness and correctness. The treatment and well-being of patients may be significantly impacted by any errors or false information. We suggest a decentralized blockchain application that enables users to register and log in as either patients or doctors to address these issues. A new user must register on the sign-up page and choose whether to log in as a patient or a doctor when they first use our platform. The user's system joins the blockchain network as soon as it is registered.

The medical records are safely saved on the interplanetary file system (IPFS), and the patient uploads them to the system. IPFS, a peer-to-peer (P2P) distributed file storage system, generates a unique hash for each record, which is then recorded on the blockchain. This blockchain entry includes hashes for all your uploaded EHRs. You can view any of your uploaded records at any time and manage access permissions for doctors within the blockchain network. The system can be used to grant access to a particular doctor if you so choose. On the other hand, you can quickly take away a doctor's authorization if you ever decide to stop giving them access. A tabular list of patients who have permitted a doctor to see their medical records is displayed to them when they register with the network.

2.1. System design

The system has role-specific profile pages and a user authentication tool for users to sign in. EHRs can be accessed and stored decentralized thanks to the blockchain connection of the online interface. Users can upload and readily access their medical records through the patient profile, which also lets them control doctor access by giving or rescinding rights. On the blockchain, each of these operations starts a transaction. The doctor's profile is designed to show the medical records that patients have given them access to. The salient characteristics of the system are depicted in Figure 1. It describes the tasks carried out by physicians and patients as well as the information exchanged between them. Blockchain and IPFS are two components of the server-side architecture that offer distributed data storage Figure 2. By acting as a link between the client and server sides, MetaMask allows users to access the decentralized system straight from their web browser. After user authentication during login, the system's flow splits into two separate profile channels, each with unique features and permissions. When the user signs out, both profiles' flows come to an end. A methodical flow diagram can be used to classify and visualize the functionality of these two profiles. Solidity-coded smart contracts that are implemented on the blockchain are essential components of the system. They implement different functionalities specific to each user type, build user structures like physicians and patients, and identify the sender of messages or transactions.

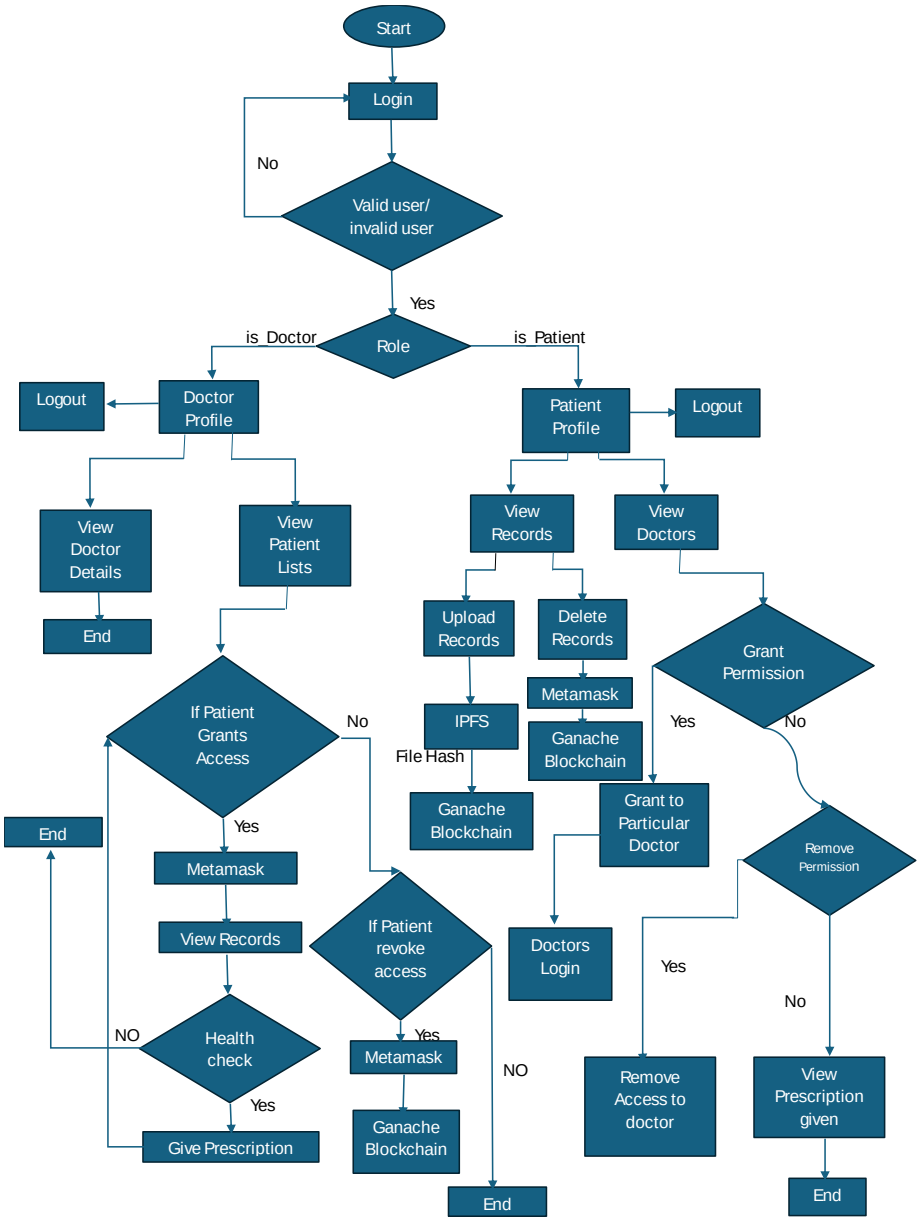

Figure 1. System overview showing tasks of physicians and patients and their information exchange

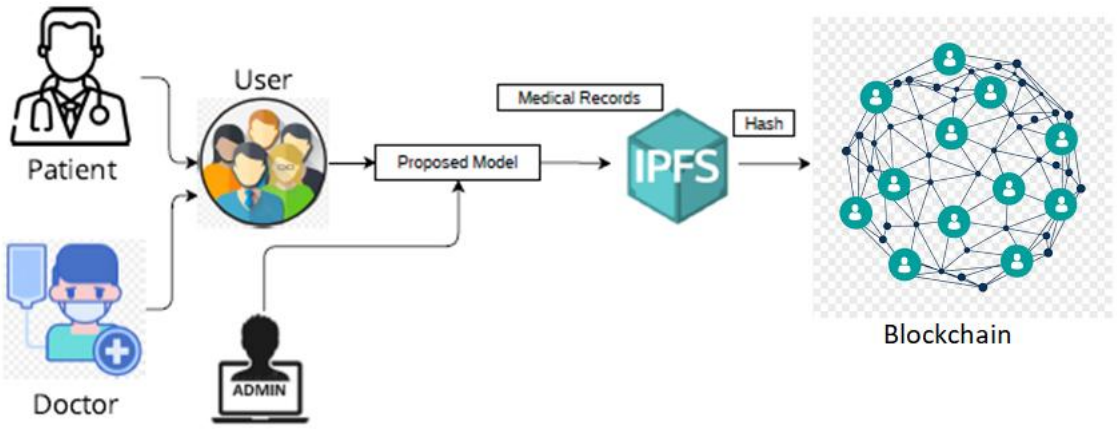

Figure 2. System architecture

2.2. Method

Our proposed solution is implemented as a decentralized application (DApp) Figure 2, which provides a user interface for interacting with the blockchain-based backend. Unlike traditional web applications, our DApp does not rely on centralized servers; instead, it leverages smart contracts deployed on an Ethereum blockchain to handle all sensitive operations such as data access control, transaction validation, and EHR reference storage. The frontend of the application is developed using HTML5, CSS3, and JavaScript, providing a responsive and intuitive user experience. Users can register, authenticate, grant or revoke access to their EHRs, and interact with their data through this interface. JavaScript is also responsible for communicating with the blockchain backend through Web3.js or Ethers.js libraries.

2.2.1. Node.js

Our system utilizes JavaScript running on Node.js, a lightweight and efficient JavaScript runtime built on Chrome's V8 engine. Node.js plays a critical role in the development and deployment process of our blockchain-based application. It provides a runtime environment not only for executing JavaScript code outside of a browser but also for managing backend interactions with the Ethereum network. In addition to executing JavaScript, Node.js offers APIs that facilitate file operations, network requests, and even the simulation of a local blockchain environment using tools like Ganache CLI. This enables us to deploy, test, and debug smart contracts locally before deploying them to a live Ethereum network. Moreover, Node.js is bundled with the node package manager (NPM), which is essential for managing dependencies and packages required to create, compile, and deploy Solidity smart contracts. Through NPM, developers can install tools such as Truffle, Web3.js, Hardhat, and IPFS clients, streamlining the entire DApp development workflow.

2.2.2. Truffle Suite

Truffle Suite is a robust and widely-used development environment, testing framework, and asset pipeline designed specifically for Ethereum-based blockchain applications. It is built to interact with the Ethereum virtual machine (EVM), enabling developers to build, deploy, and manage smart contracts efficiently. The Truffle Suite consists of three main tools: Truffle the core framework that provides a powerful set of features: smart contract compilation and deployment using built-in scripts and configuration files. Automated testing using JavaScript or Solidity to ensure smart contract logic is reliable before deployment. Migration management, which allows developers to maintain deployment versions across multiple networks (e.g., development, testnet, and mainnet). Interactive console for testing smart contracts directly from the command line with Web3.js.

2.2.3. Ganache

Ganache a personal blockchain for Ethereum development. It allows developers to run a local Ethereum blockchain on their machine for testing smart contracts in a safe, isolated environment. It provides pre-funded test accounts and real-time logging and inspection of blockchain activity, such as gas usage and transaction status. Ganache is available both as a CLI tool (Ganache CLI) and a graphical user interface (Ganache GUI). Using smart contracts, handling transactions Figure 3, and supplying the required blockchain and accounts for the system, Ganache is a local development blockchain.

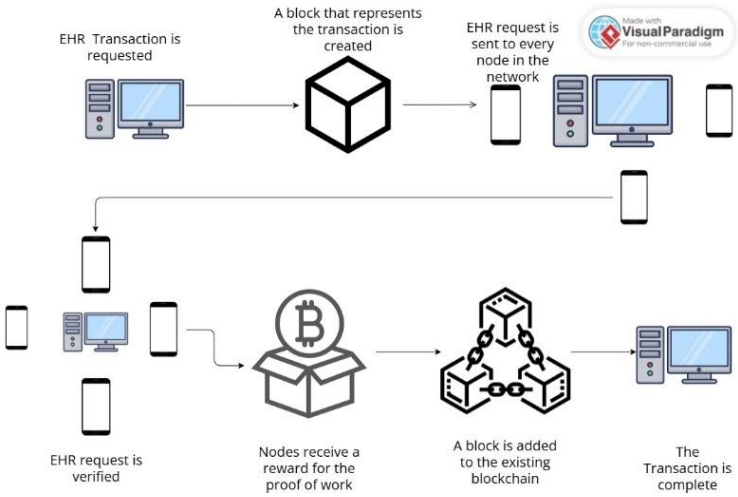

Figure 3. Transaction execution

2.2.4. Interplanetary file system

The IPFS is a P2P distributed file system designed to store and share large volumes of data in a decentralized manner. Unlike traditional cloud storage systems, where data is stored on centralized servers, IPFS allows files to be stored across a network of nodes. Each file uploaded to IPFS is broken into chunks, cryptographically hashed, and distributed among nodes. This architecture ensures redundancy, fault tolerance, and content immutability. When EHRs which often contain large documents such as test results, prescriptions, and medical images are uploaded to IPFS: content-based addressing, IPFS generates a unique cryptographic hash (called a content identifier (CID)) based on the content of the file. This CID acts as the address of the file. If the file is changed, a new CID is generated, ensuring the integrity and immutability of the content. Decentralized storage: the actual EHR files are stored off-chain on IPFS, distributed across the network. This reduces the storage burden on the blockchain and makes it easier to store large and unstructured files (e.g., PDFs and DICOM images). On-chain reference: the CID generated by IPFS is stored on the Ethereum blockchain via a smart contract. The blockchain acts as a secure and tamper-proof ledger that records: The reference (CID) to the EHR. The owner of the EHR (patient). Access permissions are granted to doctors or hospitals. Timestamp and metadata for audit trails. Secure access control: access to the EHR data on IPFS is governed through smart contracts. Patients retain control by granting/revoking access rights to specific healthcare providers. Only parties with proper authorization and the CID can retrieve the data from IPFS.

2.2.5. Metamask

MetaMask is a widely-used cryptocurrency wallet and browser extension that acts as a bridge between traditional web browsers and the decentralized web (Web3). It allows users to interact with Ethereum-based DApps directly from their browsers (such as Chrome, Firefox, or Brave) without the need to run a full Ethereum node. In the context of our decentralized EHR management system, MetaMask plays a crucial role in enabling secure, user-friendly interactions between the user (patients, doctors, and healthcare providers) and the underlying Ethereum blockchain.

3. IMPLEMENTATION

Solidity-programmed smart contracts are used to implement our solution. The functions required for the system are all contained in these smart contracts. The smart contracts are programmed, then assembled, verified, and added to the local blockchain. Next, for future blockchain communication, the application binary interface (ABI) and the contract address of the built contract are extracted. The web browser can communicate with the blockchain through the Web3 module, which can be installed using NPM. The MetaMask wallet is configured and connected to the blockchain to create this connection. Because blockchain networks are not currently supported natively by browsers, MetaMask is needed. One of the Ganache accounts' account addresses and private keys must be added to MetaMask as part of the setup process. The designated account address will then function as the sender of any message or transaction performed through MetaMask, interacting with the blockchain.

3.1. Roles

In this solution, we are going to focus on two roles: patient and doctor.

a. A patient can do the following functions:
- register to the portal: the patient can create a personal account on the healthcare portal by providing necessary identification and contact information. Successful registration allows access to personalized healthcare services and secure record management.
- add new diseases: patients can log newly diagnosed diseases or conditions into their health profile through the portal. This function ensures their medical records remain current for future consultations and treatments.
- view their records: patients can view their complete medical history, including diagnoses, treatments, and lab results. This access helps them stay informed about their health status and facilitates better communication with healthcare providers.
- update their data: patients can update personal details such as address, contact information, and relevant medical information in the portal. Keeping the data updated helps ensure smooth administrative processes and more accurate medical care.

b. A doctor can do the following things: i) register to the portal; ii) access and add the records of their patients; and iii) prescribe medicine to a patient.

3.2. Features of the application

- Register a new doctor

This function is used to register a new doctor to the ledger. Refer to Algorithm 1, the algorithm for registering a doctor collects the doctor's name, qualification, and workplace information, and stores these details in the system by creating a new doctor's entry. With each registration, the system increments the total count of registered doctors to keep an updated record.

Algorithm 1. Register doctor

1: **Input:** name (Name of the doctor), qualification (Doctor's degree), workplace (Address of hospital/clinic)
2: **Output:** Doctor details stored in the contract
3: doctor ← new Doctor storage at doctors[doctorCount]
4: doctor.name ← name
5: doctor.qualification ← qualification
6: doctor.workplace ← workplace
7: Increment doctorCount by 1
8: End Function

- Register a new patient

This function is used to register a new patient to the ledger refer to Algorithm 2.

Algorithm 2. Register patient

1: **Input:** name (Name of the user), age (Age of the user)
2: **Output:** Patient details stored in the contract
3: patient ← new Patient storage at patients[patientCount]
4: patient.patient address ← msg. sender
5: patient.name ← name
6: patient.age ← age
7: Increment patient count by 1
8: End Function

- Add a patient's disease

This function is used to add the patient's disease they are suffering refer to Algorithm 3.

Algorithm 3. Add new disease

1: **Input:** diseaseName (Name of the disease)
2: **Output:** Disease details stored in the contract
3: disease ← new Disease storage at diseases[diseaseCount]
4: disease.name ← diseaseName
5: Increment diseaseCount by 1
6: End Function

- Add medicine

  This function is used for adding the medicine details refer to Algorithm 4.

Algorithm 4. Add medicine

1: **Input:**
2: id (Id of the medicine)
3: name (Name of the medicine)
4: expiryDate (Expiry date of the medicine)
5: dose (Prescribed dose)
6: price (Price of the medicine)
7: **Output:** Medicine details stored in the contract
8: medicine ← new Medicine storage at medicines[id]
9: medicine.id ← id
10: medicine.name ← name
11: medicine.expiryDate ← expiryDate
12: medicine.dose ← dose
13: medicine.price ← price
14: **End Function**

- Prescribe medicine

  This function is used by doctors to prescribe medicine to a patient refer to Algorithm 5.

Algorithm 5. Prescribe medicine

1: **Input:**
2: id (Medicine Id)
3: patientAddress (Address of the patient)
4: **Output:** The medicine Id is added to the patient's prescription list
5: Add id to prescriptions[patientAddress] list
6: **End function**

- View patient data from a doctor

  This function helps a doctor to view patient data Algorithm 6.

Algorithm 6. View patient by doctor

1: Input:
2: patientId (ID of the patient)
3: Output:
4: patientAddress (Address of the patient)
5: name (Name of the patient)
6: age (Age of the patient)
7: Retrieve patient from patients[patientId]
8: Return patient.patientAddress, patient.name, patient.age
9: End Function

- View patient data

  This function helps to view patient data stored in blockchain refer to Algorithm 7.

Algorithm 7. View record

1: Input:
2: patientId (ID of the patient)
3: Output:
4: patientId (ID of the patient)
5: name (Name of the patient)
6: age (Age of the patient)
7: diseaseName (Name of the disease)
8: Retrieve patient from patients[patientId]
9: Retrieve diseaseName from diseases[patientId].name
10: Return patientId, patient.name, patient.age, diseaseName
11: End function

– View medicine details

This function helps to fetch medicine details Algorithm 8. This function below inputs parameters and returns the details about the medicine.

Algorithm 8. View medicine

1: Input:
2: medicineId (ID of the medicine)
3: Output:
4: id (ID of the medicine)
5: name (Name of the medicine)
6: expiryDate (Expiry date of the medicine)
7: dose (Dose prescribed for the medicine)
8: price (Price of the medicine)
9: Retrieve medicine from medicines[medicineId]
10: Return medicine.id, medicine.name, medicine.expiryDate, medicine.dose, medicine.price
11: End Function

– View prescribed medicine to the patient

This function helps the doctor to view the prescribed medicine to patients refer to Algorithm 9.

Algorithm 9. Prescribe medicine

1: Input:
2: id (ID of the medicine)
3: patientAddress (Address of the patient)
4: Output: List of prescribed medicine IDs for the patient
5: Add id to prescriptions[patientAddress] list
6: End Function

– View doctor details

This function helps to view doctor details Algorithm 10.

Algorithm 10. View doctor by ID

1: Input:
2: doctorAddress (Address of the doctor)
3: Output:
4: name (Name of the doctor)
5: qualification (Degree held by the doctor)
6: workplace (Address of the hospital/clinic)
7: Retrieve doctor from doctors[doctorAddress]
8: Return doctor.name, doctor.qualification, doctor.workplace
9: **End Function**

3.2.1. Recommended technologies

In the development of decentralized applications (dApps), smart contracts play a crucial role by enabling automated, trustless execution of logic on the blockchain. To effectively build and test these contracts, developers rely on a combination of tools and platforms that simplify the process and ensure reliable deployment. The following discussion outlines the fundamental components involved in smart contract development, including the programming language, development environment, blockchain platform, and local testing server.

– Smart contract development: smart contracts in Solidity are self-executing programs on blockchain that automate transactions when conditions are met.
– IDE tool: remix it includes a built-in Solidity compiler and testing tools to streamline smart contract development.
– Blockchain: ethereum is a blockchain platform that enables decentralized applications and smart contracts.
– Server: Ganache blockchain Ganache creates a private, cost-free blockchain environment for testing smart contracts. It supports rapid development with features like instant mining and customizable block times.

4. RESULTS AND DISCUSSION

The smart contracts can be seen as shown in Figure 4. deployed on the Ganache server after they have been implemented. Every contract has a distinct contract address that is used to recognize and access its features. The smart contracts are unchangeable and uneditable once they are launched. For this reason, extensive testing is done before putting them on the blockchain. An EHR's unique hash value is generated for each EHR that is uploaded to the system and saved on IPFS. Figure 5 shows the file details together with the hash that IPFS returned following the upload. The blockchain then records this hash. CID: QmVEgVFQ5nhBngf9cninurp6ZanAsaND7R5eHgr7txsQ1v.

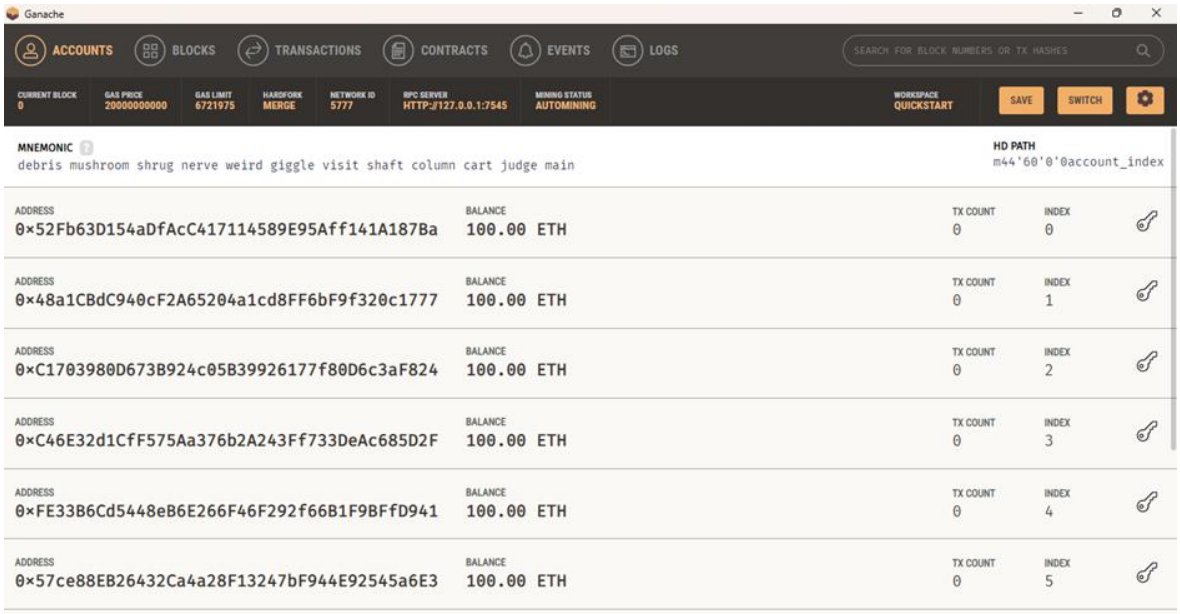

Figure 4. Deployed smart contracts

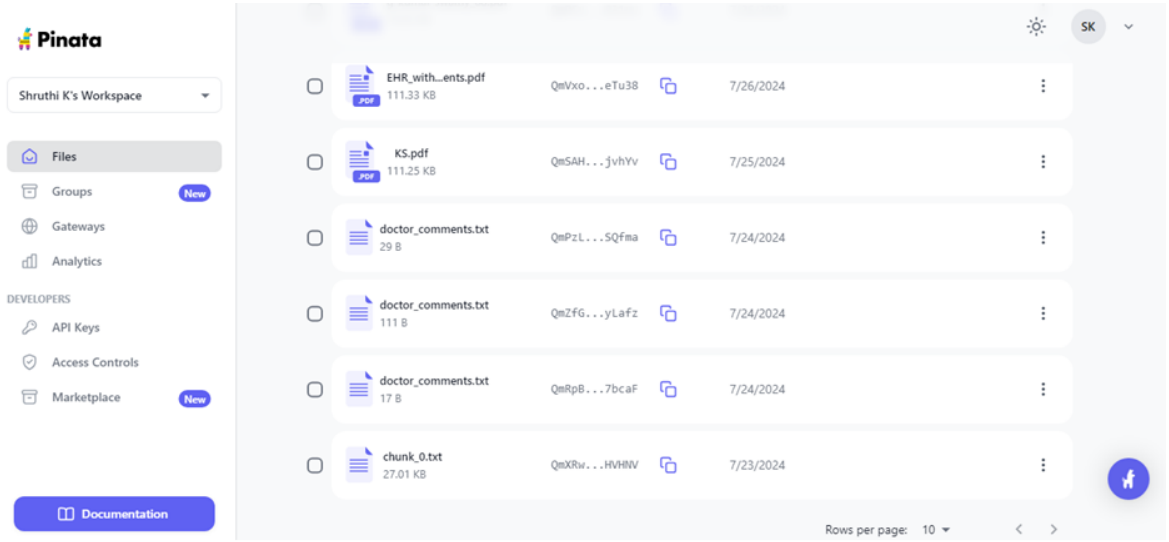

Figure 5. IPFS hash generation

4.1. Function details

Smart contract deployment on the EVM is shown in Figure 6.

a. Register a new doctor using the smart contract deployed in Figure 7.

b. Add a new disease for the patient refer Figure 8.
c. Prescribe medicine to the patient refer to Figure 9.
d. Add a new patient and update the age of the patient refer to Figure 10.
e. View prescribed medicine refer to Figure 11.
f. View doctor details refer Figure 12.

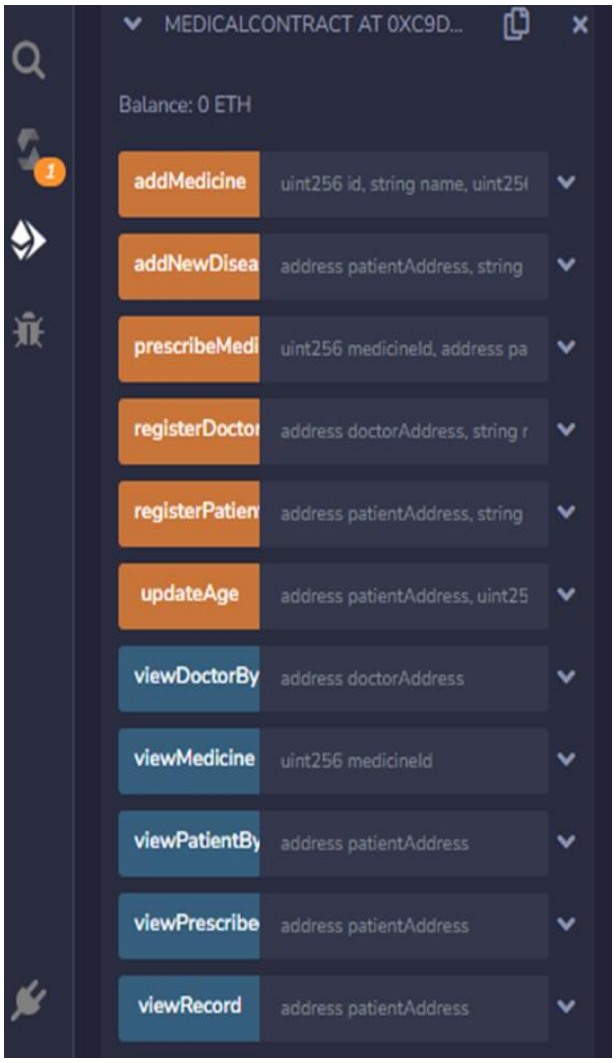

Figure 6. Smart contract deployment

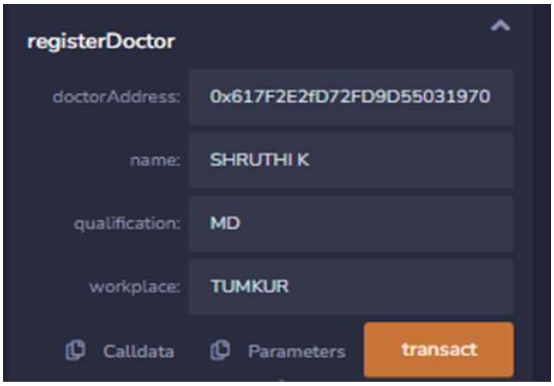

Figure 7. Register a new doctor

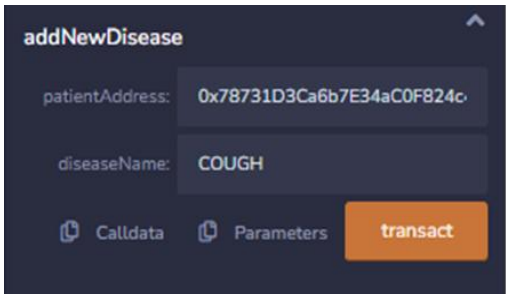

Figure 8. Adding new disease

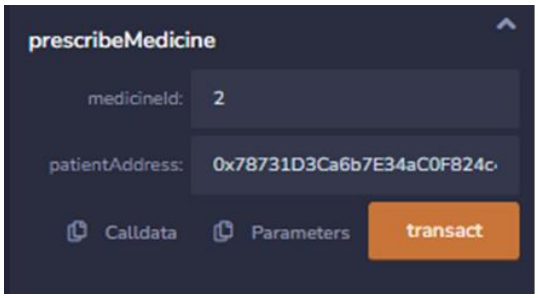

Figure 9. Prescription

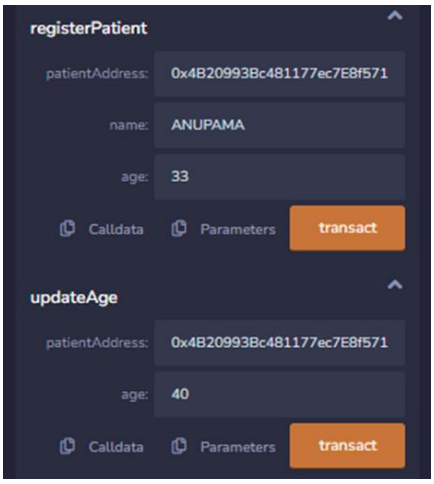

Figure 10. Updating details

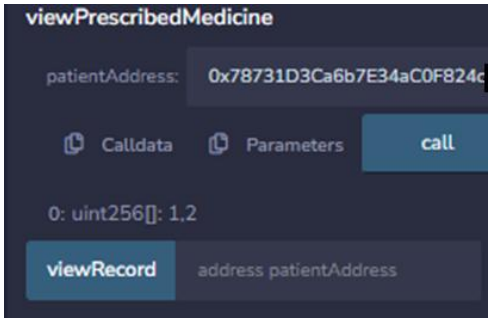

Figure 11. View details

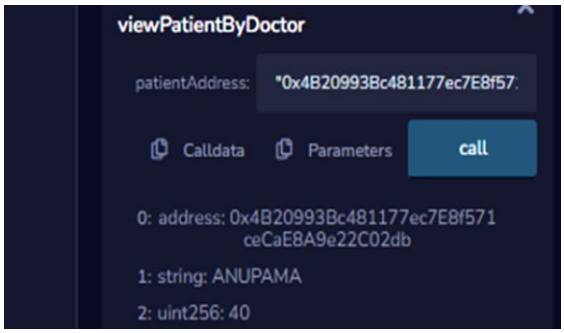

Figure 12. Doctors view

4.2. Results when we compare it to other systems

When comparing our EMR system with other electronic record systems, several advantages become evident Figure 13 details the performance comparision of traditional EMR and proposed:

- Latency: because our EMR system uses iterative models instead of computational models, it delivers decreased latency. Based on a sequence of if-else conditions, iterative models check each condition one after the other. The procedure proceeds to the next stage if a condition is met; else, it stops. The temporal complexity of this method is O(n), where n is the number of criteria. This effective condition management lowers the total processing time.
- Space complexity and accessibility: while some systems may struggle with space complexity, our EMR system leverages cloud architecture to store data. This cloud-based approach not only mitigates space constraints but also allows for global accessibility. Users can access the EMR system from virtually anywhere in the world. Additionally, the IPFS infrastructure supports low latency and ensures quick responses to user queries, enhancing the overall user experience. This combination of efficient processing and robust IPFS-based storage positions o13ur EMR system as a superior choice in terms of performance and accessibility.

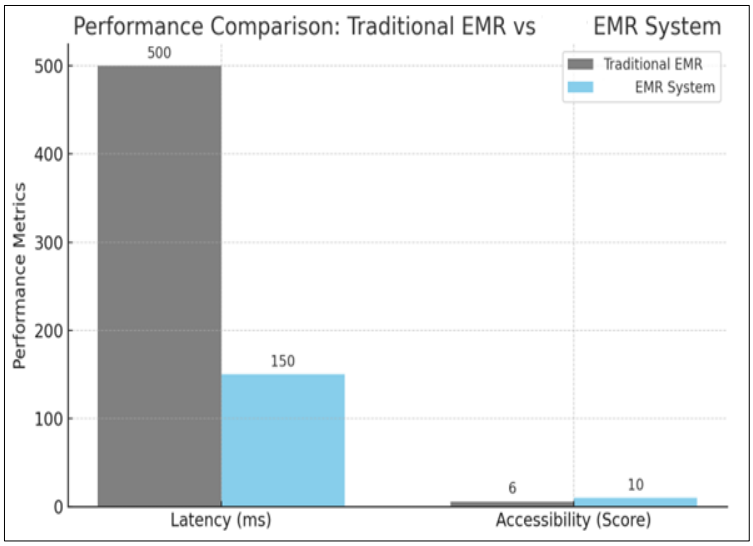

Figure 13. Performance comparision of traditional EMR and proposed

5. CONCLUSION

The architecture and development process for a blockchain-based EMR system are presented in this article. The append-only structure of blockchain and cryptographic hash functions can be used to securely preserve patient medical records while allowing authorized professionals to access them. By requiring patients' approval before granting doctors access to information and prescriptions, this method protects user privacy. Consensus techniques on blockchains allow nodes to come to agreements before adding new transactions and guarantee the consistency of transaction blocks. Personal medical data is usually kept independently by each institution in traditional healthcare systems, which causes data sharing to be delayed and access to be severely restricted owing to privacy concerns. The kind of blockchain that can be used in these situations is also explained in this article. The Ganache server and the ethereum blockchain metamask wallet are used to implement the suggested methods.

FUNDING INFORMATION

Authors state there is no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Shruthi Kumarswamy	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	✓
Poornima Athikatte Sampigerayappa		✓				✓	✓	✓	✓	✓	✓	✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analysed in this study.

REFERENCES

[1] A. Dubovitskaya, Z. Xu, S. Ryu, M. Schumacher, and F. Wang, "Secure and Trustable Electronic Medical Records Sharing using Blockchain," in *AMIA Annual Symposium Proceedings*, 2017, pp. 650–659.

[2] S. Jiang, J. Cao, H. Wu, Y. Yang, M. Ma, and J. He, "Blochie: A blockchain-based platform for healthcare information exchange," in *Proceedings - 2018 IEEE International Conference on Smart Computing, SMARTCOMP 2018*, IEEE, Jun. 2018, pp. 49–56, doi: 10.1109/SMARTCOMP.2018.00073.

[3] T. T. Kuo, H. E. Kim, and L. Ohno-Machado, "Blockchain distributed ledger technologies for biomedical and health care applications," *Journal of the American Medical Informatics Association*, vol. 24, no. 6, pp. 1211–1220, Nov. 2017, doi: 10.1093/jamia/ocx068.

[4] W. J. Gordon and C. Catalini, "Blockchain Technology for Healthcare: Facilitating the Transition to Patient-Driven Interoperability," *Computational and Structural Biotechnology Journal*, vol. 16, pp. 224–230, 2018, doi: 10.1016/j.csbj.2018.06.003.

[5] S. Amofa et al., "Blockchain-secure patient Digital Twin in healthcare using smart contracts," *PLoS One*, vol. 19, no. 2, pp. 1–28, 2024, doi: 10.1371/journal.pone.0286120.

[6] T. Sawant, P. Idayakumar, A. Sabkale, and K. Pampattiwar, "Decentralized EHR Storage Using Blockchain," *ECS Transactions*, vol. 107, no. 1, pp. 6397–6405, Apr. 2022, doi: 10.1149/10701.6397ecst.

[7] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K. S. Kwak, "The internet of things for health care: A comprehensive survey," *IEEE Access*, vol. 3, pp. 678–708, 2015, doi: 10.1109/ACCESS.2015.2437951.

[8] L. A. Tawalbeh, R. Mehmood, E. Benkhlifa, and H. Song, "Mobile Cloud Computing Model and Big Data Analysis for Healthcare Applications," *IEEE Access*, vol. 4, pp. 6171–6180, 2016, doi: 10.1109/ACCESS.2016.2613278.

[9] A. Singh, A. P. Srivastava, P. Choudhary, H. Pandey, and A. K. Singh, "Blockchain in Healthcare," in *Proceedings of International Conference on Technological Advancements and Innovations, ICTAI 2021*, IEEE, Nov. 2021, pp. 168–172, doi: 10.1109/ICTAI53825.2021.9673187.

[10] K. Shuaib, J. Abdella, F. Sallabi, and M. A. Serhani, "Secure decentralized electronic health records sharing system based on blockchains," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 8, pp. 5045–5058, Sep. 2022, doi: 10.1016/j.jksuci.2021.05.002.

[11] Q. H. Hasan, A. A. Yassin, and O. Ata, "Electronic Health Records System Using Blockchain Technology," *Webology*, vol. 18, pp. 580–593, Oct. 2021, doi: 10.14704/WEB/V18SI05/WEB18248.

[12] S. Singh, S. Gupta, and Indu, "MedEHR-Electronic health Record using Blockchain," in *2023 International Conference on Computational Intelligence, Communication Technology and Networking, CICTN 2023*, IEEE, Apr. 2023, pp. 58–62, doi: 10.1109/CICTN57981.2023.10141053.

[13] A. Masmoudi and M. Saeed, "Blockchain-Driven Decentralization of Electronic Health Records in Saudi Arabia: An Ethereum-Based Framework for Enhanced Security and Patient Control," *International Journal of Advanced Computer Science and Applications*, vol. 15, no. 4, pp. 1104–1119, 2024, doi: 10.14569/IJACSA.2024.01504112.

[14] Z. Sun, D. Han, D. Li, X. Wang, C. C. Chang, and Z. Wu, "A blockchain-based secure storage scheme for medical information," *Eurasip Journal on Wireless Communications and Networking*, no. 1, pp. 1–25, Dec. 2022, doi: 10.1186/s13638-022-02122-6.

[15] B. Bhandari, P. R. Vairagade, H. Trivedi, H. Thakre, G. Indurkar, and A. Yadav, "Decentralized Medical Healthcare Record Management System Using Blockchain," in *2023 11th International Conference on Emerging Trends in Engineering & Technology-Signal and Information Processing (ICETET-SIP)*, 2023, pp. 1–5, doi: 10.1109/ICETET-SIP58143.2023.10151658.

[16] M. Mettler, "Blockchain technology in healthcare: The revolution starts here," in *2016 IEEE 18th International Conference on e-Health Networking, Applications and Services, Healthcom 2016*, IEEE, 2016, pp. 1–3, doi: 10.1109/HealthCom.2016.7749510.

[17] P. Kumar, M. Gupta, and R. Kumar, "Improved Cloud Storage System Using IPFS for Decentralised Data Storage," in *2023 International Conference on Data Science and Network Security, ICDSNS*, 2023, pp. 1–6, doi: 10.1109/ICDSNS58469.2023.10245317.

[18] P. Purwono, K. Nisa, S. K. Wibisono, and B. P. Dewa, "Private Blockchain in the Field of Health Services," *Journal of Advanced Health Informatics Research*, vol. 1, no. 1, pp. 10–15, 2023, doi: 10.59247/jahir.v1i1.14.

[19] R. Taş, "Smart Contract Security Vulnerabilities," *Erzincan Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, vol. 16, no. 1, pp. 196–211, Mar. 2023, doi: 10.18185/erzifbed.1105551.

[20] P. A. Krishnamoorthi, S. Shahid, and O. Boydell, "Preserving Privacy in Private Blockchain Networks," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12991 LNCS, 2022, pp. 118–128, doi: 10.1007/978-3-030-96527-3_8.
[21] S. A. H. Mohsan, A. Razzaq, S. A. K. Ghayyur, H. K. Alkahtani, N. Al-Kahtani, and S. M. Mostafa, "Decentralized Patient-Centric Report and Medical Image Management System Based on Blockchain Technology and the Inter-Planetary File System," *International Journal of Environmental Research and Public Health*, vol. 19, no. 22, pp. 1–18, 2022, doi: 10.3390/ijerph192214641.
[22] R. Pakkala, "Blockchain Enabled Decentralized Application for Securing Electronic Medical Records with Smart Contracts," *Research Square preprint*, 2023, doi: 10.21203/rs.3.rs-2807625/v1.
[23] M. Shao, M. Liu, and Z. Wang, "Privacy-preserving Electronic Medical Records Sharing Solution Based on Blockchain," *International Journal of Network Security*, vol. 25, no. 1, pp. 68–75, 2023.
[24] T. Hovorushchenko, A. Moskalenko, and V. Osyadlyi, "Methods of medical data management based on blockchain technologies," *Journal of Reliable Intelligent Environments*, vol. 9, no. 1, pp. 5–16, Mar. 2023, doi: 10.1007/s40860-022-00178-1.
[25] F. Wurster *et al*., "The implementation of an electronic medical record (EMR) and its impact on quality of documentation," *European Journal of Public Health*, vol. 33, Oct. 2023, doi: 10.1093/eurpub/ckad160.864.

BIOGRAPHIES OF AUTHORS

Shruthi Kumarswamy is a Research scholar at the Department of Computer Science and Engineering, Siddaganga Institute of Technology, Tumkur, Karnataka, India. Her research interests include machine learning, blockchain technology, databases, and cloud computing. She can be contacted at email: shruthik@sit.ac.in.

Poornima Athikatte Sampigerayappa is a professor and head of the Department of Computer Science and Engineering at Siddaganga Institute of Technology, Tumkur, Karnataka, India. Her research interests include wireless sensor networks, artificial intelligence, and machine learning. She can be contacted at email: aspoornima@sit.ac.in.