International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 05 | May - 2023 Impact Factor: 8.176 ISSN: 2582-3930

An Innovative Healthcare Records Management System with Ethereum and IPFS

SHAHIB HASAN NITU[1], MAHIN ABRAR[2], ASIBUL HASAN[3]

[1,2,3] *Dept. of Computer Science and Engineering & Anna University, Chennai*

***

Abstract - In an era where data is considered the new currency, secured record management systems are crucial for organizations to operate smoothly and effectively. Pens and papers are becoming obsolete, modern day record management systems that utilize basic databases have limited scalability, security vulnerabilities. Even the EHR that utilizes SHA-3 has drawbacks as it requires custom made codes and firmware for every device. In the medical sector where the value of time is immeasurable and possibilities of deception is higher than before a transparent and immutable solution is required. The implementation of blockchain to the records management solves all these issues single handedly. Storing records using Ethereum blockchain involves hashing the record, adding it to a block on the Ethereum blockchain and then storing the file on IPFS network using CID, with the IPFS hash being recorded on the blockchain as manifest on the decentralized ledger which utilizes Smart Contracts for automated execution of predefined rules. This blockchain based record management already having Keccak-256 built in ensures enhanced data security, network analysis involving studying the structure and behavior of nodes, transactions bottlenecks, data integrity, no data manipulation, improved patient control and privacy. The outcome of the project is to leverage the Ethereum blockchain network to maintain patient healthcare records in a tamper-proof and encrypted format.

***Key Words*:** Blockchain, EHR, Decentralized Ledger, CID, Ethereum, IPFS

I. INTRODUCTION

According to a study by the Ponemon Institute, the average cost of a healthcare data breach in 2020 was $7.13 million. While many hospitals in developing countries still don't have these systems, more hospitals worldwide are beginning to use them. However, traditional electronic health record systems have issues with privacy, security, and ease of use. Hospitals control the records, and if security isn't good enough, someone might misuse them. The records are not the same between hospitals, and doctors might not understand them if they are from a different hospital. This happens because different hospitals use different software systems to create their electronic health records. Lately, hackers have been attacking medical organizations by putting harmful software on their servers, which makes the data inaccessible. They then demand payment to release the data. Two incidents happened at Howard University Hospital in Washington [6], which showed the need for good security measures when it comes to electronic health records. On May 14, 2013, one of the hospital's medical technicians was charged with breaking HIPAA (Health Insurance Portability and Accountability Act) because they sold private patient information. In another incident at the same hospital, a contractor downloaded almost 34,000 patient files to their personal computer. In India, there's a proposal for an organization called the National eHealth Authority (NeHA) [7] that aims to create regulations for eHealthcare. This will ensure that eHealth records are standardized, making it easier to access them safely while also keeping patient information private. This paper's proposed model could be used for this purpose. We have implemented cutting-edge measures to ensure that the privacy and security of our data is beyond reproach. In particular, we have harnessed the power of cryptography, blockchain technology, and IPFS to achieve this. Blockchain technology, in particular, is an immutable list of records stored in blocks that are linked to the preceding block. As a result, all transactions made between users and smart contract accounts are available for public scrutiny, and once they have been stored in a block, they cannot be reversed. We opted to use the Ethereum blockchain, which is highly versatile and enables us to deploy smart contracts to enhance security. Our solution uses a permissioned version where only designated participant nodes can mine, which adds an extra layer of security. We have also leveraged the IPFS, which stores the actual record data off the blockchain and on the network nodes, to provide an additional layer of security.

II. COMPARISON WITH EXISTING SYSTEM

MedRec was one of the first blockchain-based models for managing medical records. It used Ethereum Blockchain and Smart Contracts to store accessibility details of the health record. However, the actual health record was not stored on the blockchain, making it susceptible to attacks or misuse. Our approach differs by storing records in a distributed manner, ensuring the immutability of records using hashes in IPFS, and not relying on a third-party provider.

Another model, Blockchain for Healthcare, stores patient data on the blockchain after being encrypted with the patient's private key. While this data can be decrypted by users such as hospitals and researchers who have permission from the patient, our approach gives patients complete control over who can view their data. Patients' private keys are required to decrypt the data in our model, and we use IPFS to store data rather than the blockchain.

Blockchain for Health Data and Its Potential Use in Health IT and Health Care Related Research stores encrypted medical record details on the blockchain, including where the records are stored and who has access to them. Only the owner of the data can change the access control policies, and a biometric system is suggested for identity authentication. Our approach differs in that we use a distributed system to store data.

Other proposed models, such as the one by Qi Xia et al. and Alevtina Dubovitskaya et al., focus on blockchain-based health record sharing models. While they offer secure and trustworthy methods for sharing records, our approach allows patients to specify which records they want hospitals to gain access to. Overall, our proposed model offers a secure and efficient way to store and access health records in the presence of outside attackers. Moreover, the solution we proposed gives the user to make appointments with the available doctors of the department of their choosing. As time is of the essence saving time will eventually be beneficiary to the patients. The admins have complete control over the management of the doctors. He can monitor the appointments made by the patients. The patients can control over their own medical reports. Medical reports can include from X-rays, MRIs, CT scan, Blood test, prescription and many more depending on the type of tests given by the doctor. Without their login credentials and private key, it is not possible for others to gain access to the uploaded reports. So, our proposed model provides a more complete and enriched experience for the user.

III. PROPOSED SYSTEM

Our proposed system for electronic health records (EHR) management combines the security and immutability of blockchain technology with the distributed file system of IPFS, and the automation of smart contracts. The system will store EHRs in a distributed manner, where each patient owns their own data and has complete control over who can view and access it. The EHRs will be stored on IPFS, which ensures the immutability of records through the use of cryptographic hashes. To ensure the security and privacy of EHRs, we will use a blockchain-based system where access to the EHRs will be granted through smart contracts. Patients will be able to define the access policies for their EHRs and grant access to healthcare providers and researchers through these smart contracts. Access to EHRs will only be granted with the patient's private key, which is securely stored and controlled by the patient. In addition to providing a secure and distributed way to manage EHRs, our system will also automate the process of updating and accessing EHRs.

When a patient visits a healthcare provider, the provider will be able to access the patient's EHR through the smart contract, eliminating the need for manual updates and paperwork. Overall, our proposed system for EHR management using blockchain, IPFS, and smart contracts provides a secure and efficient way to manage and access EHRs, while also giving patients complete control over their own data. In addition to providing a secure and distributed way to manage EHRs, our system will also automate the process of updating and accessing EHRs. When a patient visits a healthcare provider, the provider will be able to access the patient's EHR through the smart contract, eliminating the need for manual updates and paper work.

Overall, our proposed system for EHR management using blockchain, IPFS, and smart contracts provides a secure and efficient way to manage and access EHRs, while also giving patients complete control over their own data. In an electronic health record management system, patients, doctors, and administrators each have their own unique login interfaces to access the system.

- **Patients Functionality**

Patients can log in to view their medical records, update their personal information, and manage their consent preferences for sharing their data with healthcare providers. They can also view their appointments, test results, and medication lists. To ensure security, patients are required to provide their username and password or use biometric authentication methods such as facial recognition or fingerprint scanning.

- **Doctors Functionality**

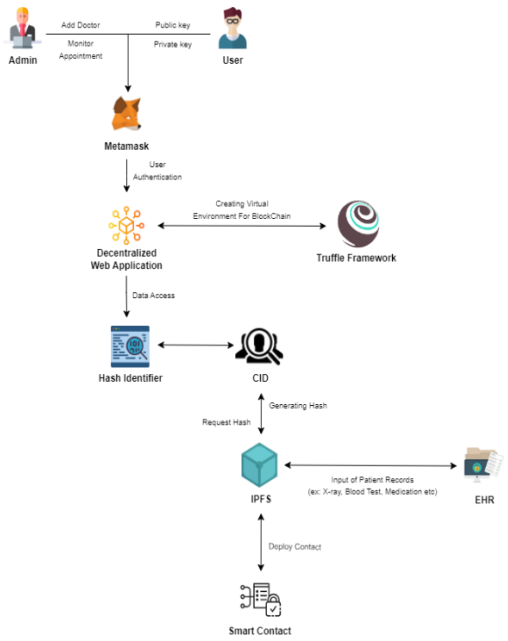
Doctors have access to a different interface where they can view their patients' medical records, add new information, and update the patients' treatment plans. They can also request and view diagnostic tests and order medication prescriptions. Doctors are required to use their own unique username and password to log in to the system, which ensures only authorized medical professionals can access patient information.

- **Admin's Functionality**

Administrators have access to a more comprehensive interface that allows them to manage the entire electronic health record management system. They can view system logs, monitor user activity, and manage user roles and permissions. Administrators are required to use their own unique login credentials to ensure that only authorized personnel have access to system-wide settings.
Each login interface has its own set of features and functions, which are customized to meet the specific needs of patients, doctors, and administrators. All users of the electronic health record management system must adhere to strict data privacy and security standards to protect sensitive patient information from unauthorized access or disclosure.

IV. SYSTEM ARCHITECTURE

In our proposed project, we have designed the system to accommodate different types of users, namely patients, doctors, and administrators. Each user type has a unique set of permissions and access levels to ensure that they can only perform tasks that they are authorized to do.



User Layer

A user is someone who uses a system and its resources. In this proposed system, users could be patients, doctors, or administrative staff. Their main job would be to use the system to create, read, update, and delete medical records. They will access the system using a web browser, which has a graphical interface that shows them the different functions they can use. This interface is like a menu that lets them choose what they want to do. Each user has different permissions based on their role. Behind the graphical interface is the blockchain layer, which is the core technology that powers the system.

Blockchain Layer

This layer allows users to interact with the DApp that operates on the blockchain. This layer has three essential elements:

Blockchain Assets: These are pieces of information that users can send to others or store for future use. They are treated as assets by the blockchain network.

Governance Rules: The blockchain network follows specific consensus rules, which ensure that the transactions are secure and trustworthy. Ethereum uses Proof of Work (PoW) algorithm to maintain the governance of the blockchain.

Network: Ethereum uses a peer-to-peer network, where all the nodes have equal status and rights. This is because the platform aims to be distributed, not centralized.

In simpler terms, the blockchain layer is where the magic happens. It's what allows users to interact with the system and update the medical records securely. It uses special rules and algorithms to make sure that the information is protected and cannot be tampered with. The network is made up of different nodes that work together, rather than having one central control. This means that the system is decentralized, which makes it more secure and trustworthy.

V. TRANSACTION

The system has different types of transactions that can be performed. One is adding records, which creates a patient's medical record in the system. This includes important information like the patient's ID, name, co-morbid conditions, blood group, and an IPFS hash. The IPFS hash is used to store additional medical records or lab results for the patient. Another transaction is updating records, which allows doctors to change basic patient information but not the IPFS hash for security reasons.

Users can also view records, which shows the medical records for a specific patient. This function can be used by both doctors and patients. However, the system ensures that patients can only see their own records by verifying their public account address. The delete records transaction allows doctors to delete a patient's record from the blockchain.

Access to these transactions is limited to specific users based on their role. For example, only doctors and nursing staff are allowed to add or update records. Patients can only view their records and are not given permission to add or update them. This helps to ensure that the system remains secure and that the information stored in it is accurate and up-to-date.

- Add records - lets you create a new medical record for a patient. It has fields for things like their name and blood type. It also includes a special code that links to other medical records about the patient.
- Update records- lets you change some information about a patient's medical record, but not the special code that links to other records. This is to keep the records secure.
- View records- lets you look at a patient's medical records. Patients and doctors can use this to see the records, but the system makes sure each person only sees the records they're supposed to see.
- Delete records- lets doctors remove a patient's medical record from the system.

Only doctors and some other people can "Add" or "Update" records. Patients can only "View" their own records.

VI. SYSTEM IMPLEMENTATION

In this section, we'll take a closer look at how the system was implemented using Ethereum and its related technologies. By delving into the technical details, we'll gain a better understanding of the various functions of the system.

To start with, the system was built on top of the Ethereum blockchain, which allowed us to take advantage of its decentralized and tamper-proof nature. We used the Solidity programming language to write smart contracts, which are self-executing code that automates the execution of certain actions on the blockchain.

The smart contracts were deployed on the Ethereum blockchain using the Truffle framework, which provides a suite of tools for developing and testing smart contracts. We used the Ganache CLI tool for local development and testing, and then deployed the smart contracts to the Rinke by test network.

To interact with the smart contracts, we built a web-based user interface using React.js, a popular JavaScript library for building user interfaces. We used the Web3.js library to connect the user interface to the Ethereum blockchain and interact with the smart contracts.

The system's functionality was divided into different transactions, including adding records, updating records, viewing records, and deleting records. Each of these transactions was implemented as a separate function in the smart contract code, with appropriate access controls to ensure that only authorized users could perform certain actions.

The system also relied on IPFS, a distributed file storage system, to store large medical records files. The IPFS hashes for these files were stored alongside the basic patient records in the Ethereum smart contracts, providing a secure and decentralized storage solution.
In summary, the system was implemented using Ethereum, Solidity, Truffle, Django, Web3.js, and IPFS. By combining these technologies, we were able to build a decentralized and secure medical records management system that provides a high level of privacy and security for patients.

VII. USAGE SCENARIO FOR ALGORITHM

Algorithm 1: Smart Contract for Patient Records

Assign Roles:

function De_ne Roles (New Role, New Account)
add new role and account in
roles mapping
end function
Add Data:
function Add Patient Record (*contains variables to add data*)
if (msg.sender DD doctor) **then**
add data to particular patient's record
else Abort session
end if
end function
Retrieve Data:
functionView Patient Record (patient id)
if (msg.sender DD doctor jj patient) **then**
if (patient id) DD true **then**
retrieve data from speci_ed patient (id)
return (patient record)
to the account that requested the retrieve operation
else Abort session
end if
end if
end function

Update Data:

function Update Patient Record (*contains variables to update data*)
if (msg.sender DD doctor) **then**
if(id DD patient id && name DD patient name) **then**
update data to particular patient's record
return success
else return fail
end if
else Abort session
end if
end function
Delete Data:
function Delete Patient Record (patient id)
if (msg.sender DD doctor) **then**
if (id DD patient id) **then**
delete particular patient's record
return success
else return fail
end if
else Abort session
end if
end function

Algorithm 2: Storing the IPFS hash in the smart contract

Input: IPFS Hash of the record

1: ihash IPFS hash of the record
2: p1 ihash.substringh0; 24i
3: p2 ihash.substringh25; 48i
4: p1 r1 p1+randomCharacter
5: p2 r2 p2+randomCharacter
6: finalpart1 convertToBytes32(p1 r1)
7: finalpart2 convertToBytes32(p2 r2)
8: Store finalpart1 and finalpart2 in smart contract with a mapping to the combined key/patient key

Algorithm 3: Retrieving the IPFS hash from the smart contract

Input: Patient public key/ Combined key

Output: IPFS Hash of the record

1: contracthash hash retrieved from the smart contract
2: finalpart1 contracthash.splitonComma[1]
3: finalpart2 contracthash.splitonComma[2]
4: p1 r1 convertToBase58(finalpart1)
5: p2 r2 convertToBase58(finalpart2)
6: p1 p1 r1-r1
7: p2 p2 r2-r2
8: hash p1+p2

VIII. OPTIMISATION OF STORAGE ON BLOCKCHAIN

To store the mapping between patient/combined key and the ipfs hash values, we used smart contracts. However, this method would not work for a public blockchain because storing a large amount of data in the smart contract would be too costly and slow down the system. Instead, we used IPFS

to store the patient public key to hash mapping and only stored the hash of the map in a smart contract. This way, the smart contract only stores a small amount of data, making it more efficient.

There are three smart contracts in this method. The first stores the hash of the map stored in the IPFS network and is also used by patients to view their records. The second is accessed by patients when they want to grant or revoke access to a hospital and stores the combined patient and hospital key to hash mapping. The third is accessed by hospitals when they create a record and need to store it in IPFS.

We only transferred one of the mappings to IPFS because the other two maps involve entries that require deletion and updation, which would lead to the creation of new maps and waste space. By using IPFS for one mapping and smart contracts for the others, we can store and access data efficiently without slowing down the system.

IX. RESULT

Fig. 1: Onboarding Screen

Fig. 1: shows the onboarding screen of our project with all functionality includes login for all users and admin, appointment booking.

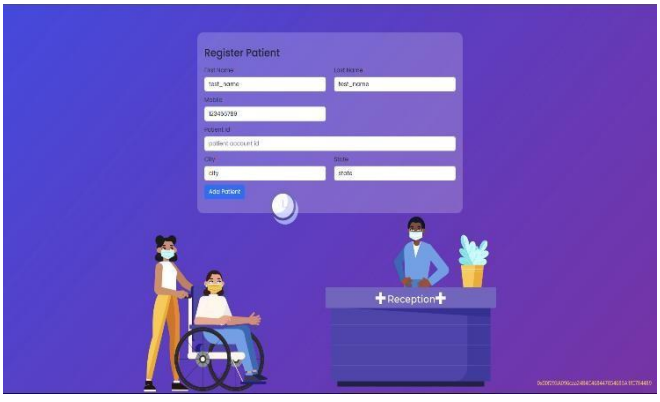


Fig. 1.1: Patients Registration Screen

Fig. 1.1: shows the patients registration form before booking any appointment and getting treatment.

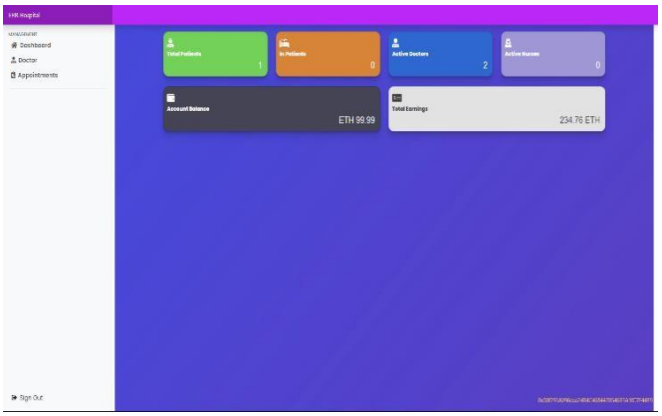


Fig. 1.2: Admin Dashboard

Fig. 1.2: shows the admin dashboard from where An Admin can add doctors, monitoring all the appointments and also the finance management status.

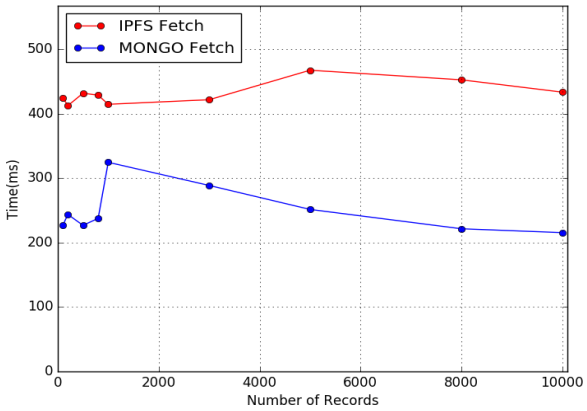


Fig. 2: Record Retrieval Time Comparison

The graph in Fig.2: illustrates the comparison between the time taken to fetch records using IPFS and MongoDB. However, it should be noted that the speed of fetching records from IPFS depends on various factors such as the number of systems that store the records and the proximity of the system containing the record. Due to this variability, it is difficult to accurately analyze IPFS fetch times in a simulation of our system.

Despite the possibility of slightly slower fetch times compared to traditional cloud services, IPFS remains the optimal choice for our model due to its ability to ensure record immutability and decentralization.

Moving on to Fig: 3, we can observe the time taken to retrieve record hashes from smart contracts while varying the number of participants in the network. The test was performed using ganache-cli, and it can be seen that the retrieval time does not significantly vary with the increase in the number of participants.

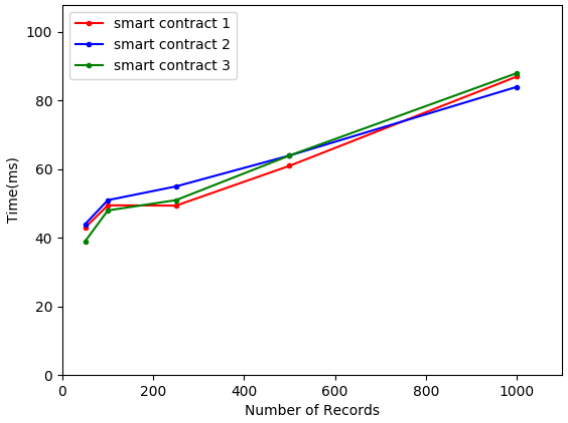


Fig. 3: Hash Retrieval from Smart Contract Time Comparison

X. DISCUSSION

The main challenge in e-Health is confidentiality, which is also a crucial requirement. Although the proposed system ensures that hospitals cannot access a patient's record without their consent, there is no mechanism to prevent hospitals from taking photographs or copying the record's contents when uploading or accessing it.

Another challenge is implementing a digital system in developing countries where many people lack internet access and may not know how to use such a system. Furthermore, in IPFS, data can only be shared if the system containing the record is online. If multiple systems have the record, at least one must be online for transfer to be possible.

The primary advantage of using blockchain in this scenario is its decentralized and immutable logging system, allowing all participants to view who gave permission to access a hospital, who revoked access, who created or accessed which record, etc. This feature enhances security and makes it easier to identify data and record misuse. The use of IPFS, combined with blockchain, creates a fully decentralized system that does not require any organization to store or log data. IPFS ensures record immutability, while blockchain guarantees the immutability of logged transactions. The use of biometric encryption allows access to patient records using their fingerprints, even if they are unable to provide access themselves.

However, certain considerations must be taken into account that may make blockchain and IPFS unsuitable for this system in their current form. In IPFS, data is initially stored on the system that adds it to the IPFS network, and the data is sent through the network to other systems that can act as providers of the data. In this scenario, however, the hospital that creates the record initially stores the data, and only the patient will access the data. Therefore, the replication factor for each record should be at least 20 to allow the IPFS network to function optimally.

Although the blockchain logs patient visits to hospitals, this can be considered a privacy infringement. However, users are identified using their public keys, and their names are not exposed on the system, making it challenging to correlate public keys with patients. It is also impossible for users to delete all copies of a record on the IPFS network, even if they track who stores the record's copy. A viable solution to this problem could be to use IPNS records to point to such records, allowing users to revoke access later. Only the hash of the IPNS record will be registered in the blockchain, not the hash of the medical record directly.

Storing data on a blockchain is not recommended as it limits the scalability of the system. Since IPFS cannot store smart contract contents, a workable solution might involve using cloud storage, which would undermine the system's decentralized nature. In the current system, the hospital is a single entity on the blockchain, which is not ideal. A hospital has several departments with numerous doctors in each department, and one department or doctor should not be able to create or view another department's records without permission.

XI. CONCLUSION

Our paper illustrates how blockchain technology can be utilized in the healthcare sector to improve HER management. Despite technological advancements in EHR systems, they still face challenges such as insurance fraud. Our proposed framework combines secure record storage with granular access rules to address these issues.

Moving forward, we intend to implement an inventory management module within this framework. However, we must consider regulated policies and pricing of individual inventory elements to ensure effective management of each product.

Currently, the system has only been tested using simulation software and test nodes provided by a third-party service. To fully understand the capabilities and limitations of the system, it needs to be tested on a real permissioned blockchain with a properly set up IPFS network. This will help evaluate how the system handles a large number of users trying to store, access, and update data on the IPFS network and smart contracts simultaneously. The goal is to see how the system behaves under real-world conditions with millions of users.

REFERENCES

[1]. A. Shahnaz, U. Qamar and A. Khalid, "Using Blockchain for Electronic Health Records," in IEEE Access, vol. 7, pp. 147782-147795, 2019A. Shahnaz, U. Qamar and A. Khalid, "Using Blockchain for Electronic Health Records," in IEEE Access, vol. 7, pp. 147782-147795, 2019 Jan. 2014, pp. 2716–2724.

[2]. Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, ‘‘An overview of blockchain technology: Architecture, consensus, and future trends,’’ in Proc. IEEE Int. Congr. Big Data (BigData Congr.), Jun. 2017, pp. 557–564.

[3]. A. Dubovitskaya, Z. Xu, S. Ryu, M. Schumacher, and F. Wang, "Secure and trustable electronic medical records sharing using blockchain," in AMIA '17, vol. 2017, 2017, pp. 650-659.

[4]. R. Guo, H. Shi, Q. Zhao, and D. Zheng, "Secure attribute-based signature scheme with multiple authorities for blockchain in electronic health records systems," IEEE Access, vol. 6, pp. 11676-11686,2018.

[5]. “Problems with Medical Records”, Henry M. Tufo and Joseph J. Speidel, Medical Care, Vol. 9, No. 6 (Nov. - Dec., 1971), Published by: Lippincott Williams & Wilkins

[6]. M. Azhagiri, R. Amrita, R. Aparna and B. Jashmitha, "Secured electronic health record management system", Proc. 3rd Int. Conf. Commun. Electron. Syst. (ICCES), pp. 915-919, Oct. 2018.

[7]. ”IPFS Documentation”, Internet: https://docs.ipfs.io/ [July 25 2018]

[8]. V. Buterin, ”A Next-Generation Smart Contract and Decentralized Application Platform, Ethereum White paper”, 2014.

[9]. V. Saini,” Understanding IPFS in Depth (3/6): What is Inter Planetary Naming System (IPNS)?”, Internet: https://hackernoon.com/, Feb 28 2019, [Sep 21 2019]

[10]. Q. Gan and Q. Cao, ‘‘Adoption of electronic health record system: Multiple theoretical perspectives,’’ in Proc. 47th Hawaii Int. Conf. Syst. Sci.,

[11]. N. Atzei, M. Bartoletti, T. Cimoli, S. Lande, and R. Zunino, ‘‘SoK: Unraveling bitcoin smart contracts,’’ in Proc. Int. Conf. Princ. Secur. Trust, Thessaloniki, Greece, 2018, pp. 217–242

[12]. A. McLeod and D. Dolezel, ‘‘Cyber-analytics: Modeling factors associated with healthcare data breaches,’’ Decis. Support Syst., vol. 108, pp. 57–68,
Apr. 2018.

[13]. T.-T. Kuo, H.-E. Kim, and L. Ohno-Machado, ‘‘Blockchain distributed ledger technologies for biomedical and health care applications,’’ J. Amer.
Med. Inform. Assoc., vol. 24, no. 6, pp. 1211–1220, 2017.

BIOGRAPHIES :

Shahib Hasan is a dedicated and passionate student pursuing a Bachelor's degree in Computer Science and Engineering from Anna University, Chennai. With a solid foundation in computer science principles, programming, and problem-solving, he is eager to specialize in human-computer interaction (HCI) through a Master's program. He is fascinated by the integration of human behavior and user experience with technology, and he strives to create innovative and user-friendly solutions. UX design competitions and workshops, honing his technical and teamwork skills. His goal is to contribute to the development of technology that enhances usability and optimizes human-computer interaction.

Mahin Abrar is a motivated and dedicated student pursuing a Bachelor's degree in Computer Science and Engineering from Anna University. He aspires to specialize in cloud engineering through a Master's leveraging his strong foundation in cloud computing, networking, and system design. With a keen eye for detail and a passion for efficiency, he aims to design robust and secure cloud infrastructures that enable businesses to harness the full potential of cloud computing. He actively seeks opportunities to expand his practical knowledge and has gained hands-on experience with industry-leading platforms and tools.

Asibul Hasan is an ambitious student currently pursuing a Bachelor's degree in Computer Science and Engineering at Anna University. With a passion for technology and a keen interest in web development, aspires to he became a skilled web developer and make a mark in the digital world.

In addition to his technical skills, he possesses excellent communication and collaboration abilities. He recognizes the importance of effective teamwork in the web development process and is adept at working harmoniously with others to achieve shared goals. His strong interpersonal skills enable him to effectively communicate complex technical concepts to both technical and non-technical stakeholders.