BLOCKCHAIN TECHNOLOGY

TO IMPROVE SECURITY AND ACCESSIBILITY OF ELECTRONIC HEALTH RECORDS

SUBMITTED BY:

ABHINNA MANANDHAR

COVENTRY ID: 11494793

SUBMITTED TO:

PROJECT SUPERVISOR

MANOJ SHRESTHA

Sage Plagarism Report

Powered by schoolworkspro.com

Abhinna Manandhar

200248

102

Results Found

11.67 %

Match Percentage

Submitted to: Softwarica	**0.76%**
Submitted to: Softwarica	**0.64%**
Submitted to: Softwarica	**0.38%**
Source: How Does Blockchain Work? Everything You Need to Know [Updated] Link: https://www.simplilearn.com/tutorials/blockchain-tutorial/blockchain-technology	**0.27%**
Source: Cryptocurrency Explained With Pros and Cons for Investment Link: https://www.investopedia.com/terms/c/cryptocurrency.asp	**0.25%**
Submitted to: Softwarica	**0.23%**
Submitted to: Softwarica	**0.23%**
Source: Blockchain Architecture Simplified: How It Works | Edureka Link: https://www.edureka.co/blog/blockchain-architecture/	**0.22%**
Submitted to: Softwarica	**0.21%**
Source: Electronic Health Records - ppt video online download Link: https://slideplayer.com/slide/5798471/	**0.21%**
Submitted to: Softwarica	**0.21%**

Concept Diagram

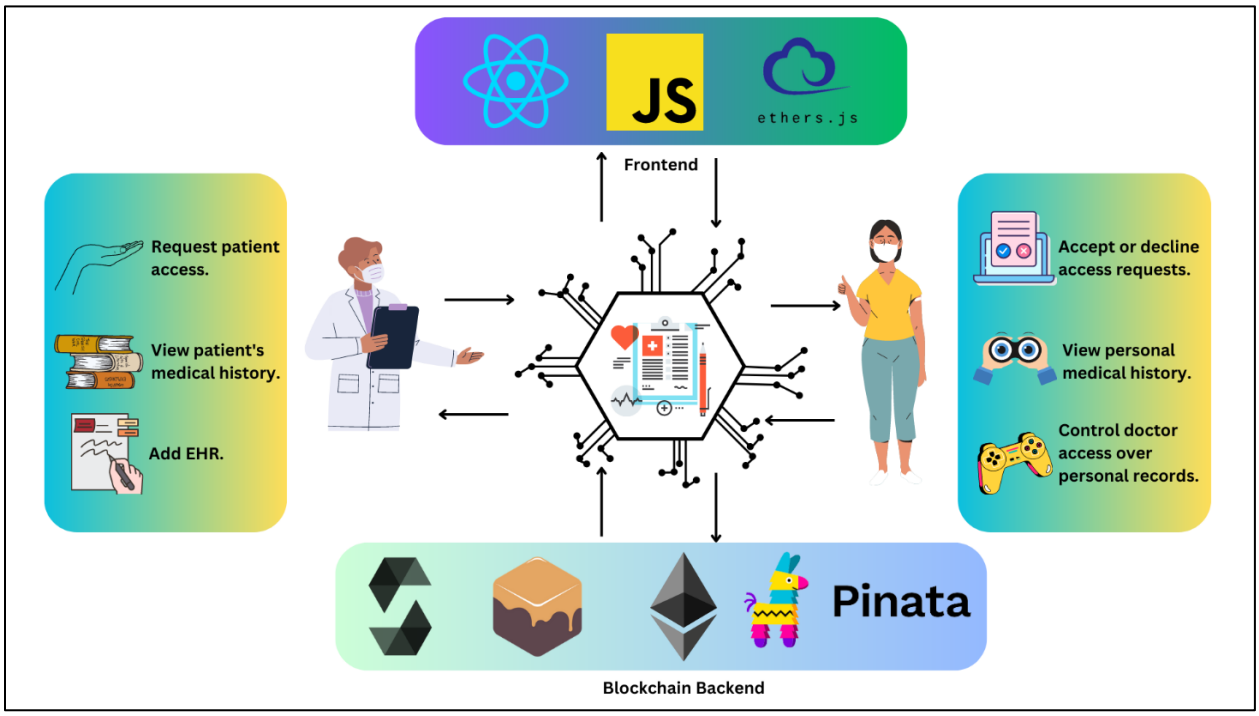

Figure 1: Concept Diagram

Acknowledgment

I would like to express my heartfelt gratitude towards my family, friends, teachers, and everyone who has contributed towards the success of this research. I would like to express great appreciation for our supervisor Mr. Manoj Shrestha for providing all the necessary guidance for the success of this research. I am very grateful for the constant encouragement and moral support provided by my parents, friends, and family. The assistance, mentorship, motivation, and support provided by everyone has played a crucial role in the achievement of this undertaking. I express profound gratitude for your invaluable contributions, which have significantly enhanced the research and facilitated the resolution of various challenges encountered.

Abstract

Electronic medical records are extremely delicate information for patients and healthcare providers that need to be stored with care. This research explores different alternative to the traditional centralized methods of storing these records and studies a more patient-centric approach to provide users with the ownership of their data and solve problems associated with information-asymmetry and data privacy and security. This paper focuses on the use of blockchain technology and compares its potential use cases with the currently popularly used system designs for storing Electronic Health Records. We also analyze the existing technologies and use cases to study and analyze their advantages and places for improvements. Furthermore, the paper also takes into consideration the legal and ethical challenges associated with our solutions. Finally, the research also properly the places for improvement in the solution and discusses the limitations and future works.

Keywords

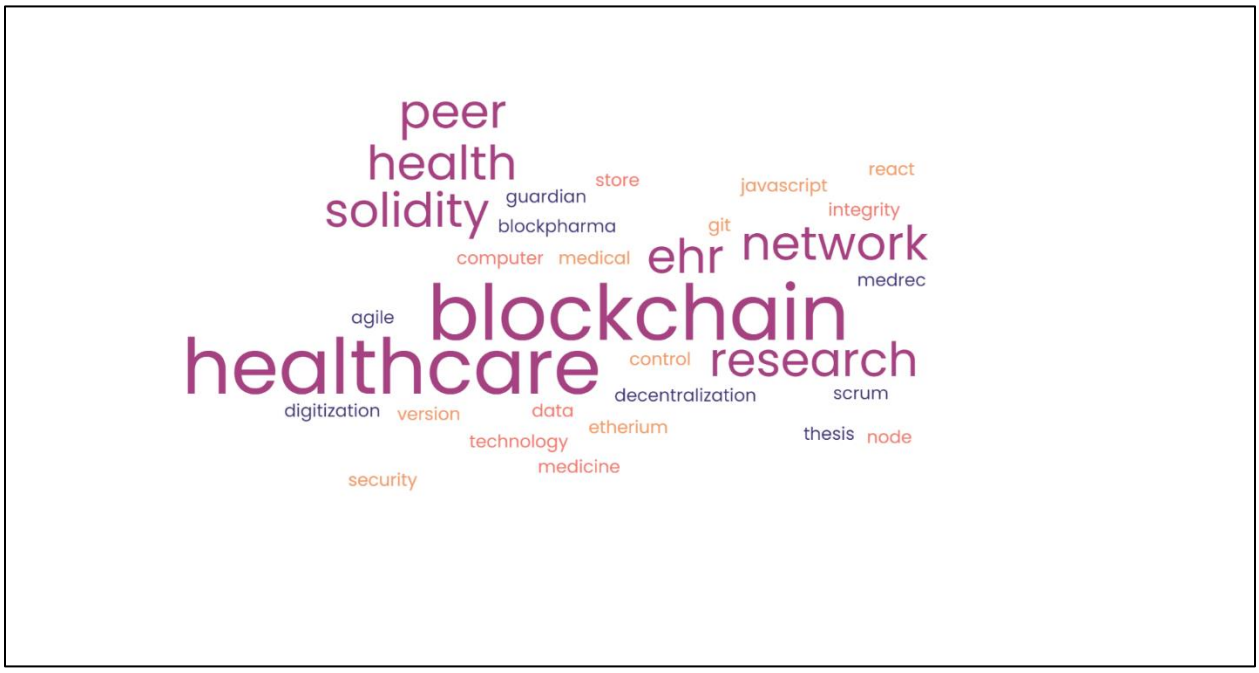

Figure 2: Keywords

Contents

Table Of Figures

Introduction

As one of the world's largest industries, the healthcare sector constitutes different extremely sensitive data associated with organizations and individuals. Therefore, with modern database technologies, the industry has seen many advancements in data storage techniques. The term "Electronic Health Record" represents a digital copy or version of a patient's health records (Seymour et al., Electronic Health Records (EHR) 2012). These Electronic Health Records are traditionally stored and managed by individual groups or healthcare organizations. The method of digitally storing data in databases and networks gained popularity, especially after the rise in favor of the internet. Unsurprisingly, the healthcare industry has also adapted to the trend in the past few decades. Storing healthcare records digitally in databases and networks helped increase security, maintainability and reduce redundancy of the data (Han et al., 2022). However, keeping data in a centralized system may have many disadvantages. Central databases have always been limited to only a selected group of organizations. Despite enabling a high level of privacy and security, centralized databases are usually more vulnerable to attacks and data breaches. Since a single authority control all the data in the system, it also discourages interoperability among multiple organizations. In the context of EHRs, centralized databases limit the access of the records to only a few groups of organizations, individuals, and healthcare providers. This makes interoperability among multiple organizations extremely difficult. This also creates the need for patients to carry a physical or an unnecessary copy of their records for them to share among multiple organizations.

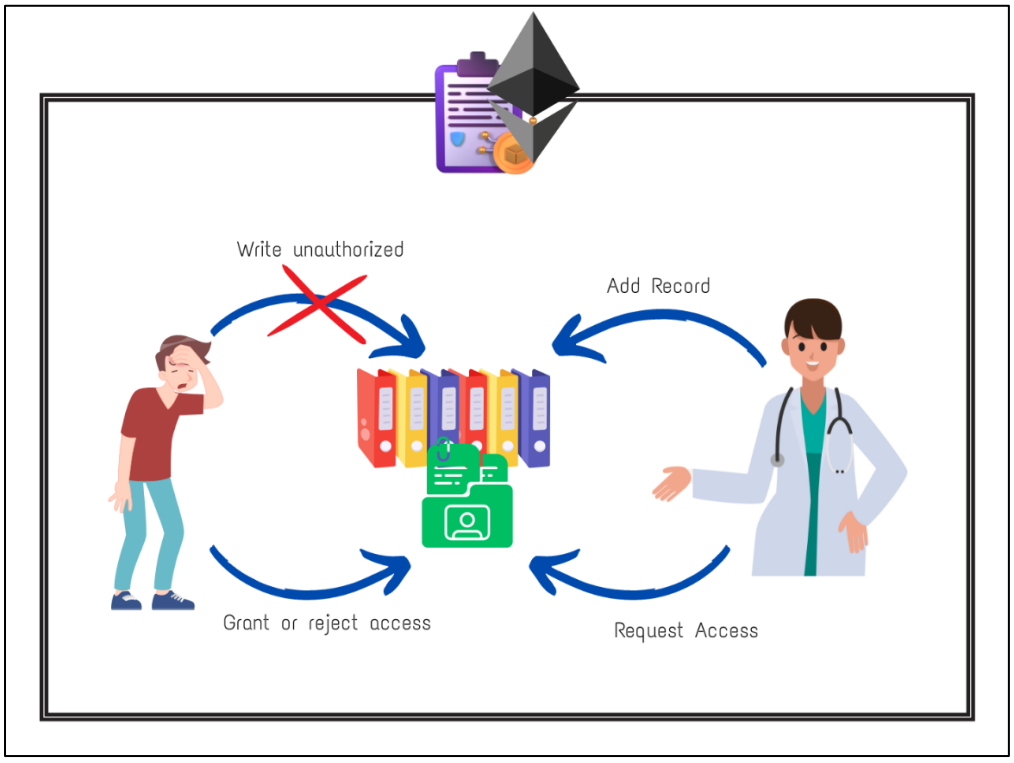

Figure 3: Introduction

A blockchain is a distributed database of multiple records of transactions or events in a network. Like other peer-to-peer networks, it is a decentralized technology for storing data. In technical terms, a blockchain consists of multiple interlinked nodes. Each block in the network holds a copy of the entire chain. Whenever a new node is to be added, it is verified using a consensus method, i.e., all or majority of the nodes must verify the transaction to carry out. The Ethereum blockchain is one of the largest and most popular public blockchain networks. One of the key features of the Ethereum blockchain is smart contracts.

In simple words, smart contracts are digital scripts that automate actions once the requirements are satisfied (Wang et al., 2018). They can be used to ensure trust and transparency among multiple parties. In technical terms, smart contracts are executed whenever a transaction takes place in the blockchain, i.e., when a new node is added to the chain. These smart contracts can also be used to ensure permission control among different members of the blockchain. This technology can be incredibly beneficial for the healthcare industry to store data. As mentioned, the industry consists of various sensitive data. Using smart contracts to store these data in a blockchain can allow organizations and individuals to store their data securely and ensure personal ownership. By making use of smart contracts, individuals can have control over who is allowed to view their records and change them respectively. However, it is challenging to change or update data in a blockchain, and individuals can still permit other individuals or organizations to add to their records. Information such as who is allowed to access and add health records in the blockchain can also be pre-defined in smart contracts.

After gaining massive popularity in the finance industry, people and organizations have explored its potential in many other industries as well (Yli-Huumo et al., 2016). Healthcare is also one of those industries that can benefit intensively from the use of blockchain technology. Promoting decentralization in healthcare can ensure complete ownership of one's medical records and can directly impact the interoperability of medical records among multiple organizations. Instead of creating a centralized data silo of the patient's records and healthcare data, the technology can ensure the implementation of a single repository of all the records for all the organizations to share among (Abimbola, Baatiema, & Bigdeli, 2019). Harnessing the power of smart contracts, individuals can be provided with complete authority over the access control of their data (Abimbola, Baatiema, & Bigdeli, 2019). This provides individuals with a choice of who they want to share their records with. Additionally, smart contracts can also enable a form of authorization for reading and writing records onto the network. For example, write access can only be provided to healthcare providers, while read access can depend on the choice of the owner of the data. Different encryption methods can assure additional security onto the platform, further disabling any security breaches.

Aim

Securely store Electronic Health Records using blockchain technology and ease accessibility for individuals and healthcare providers.

Figure 4: Aim

Objectives

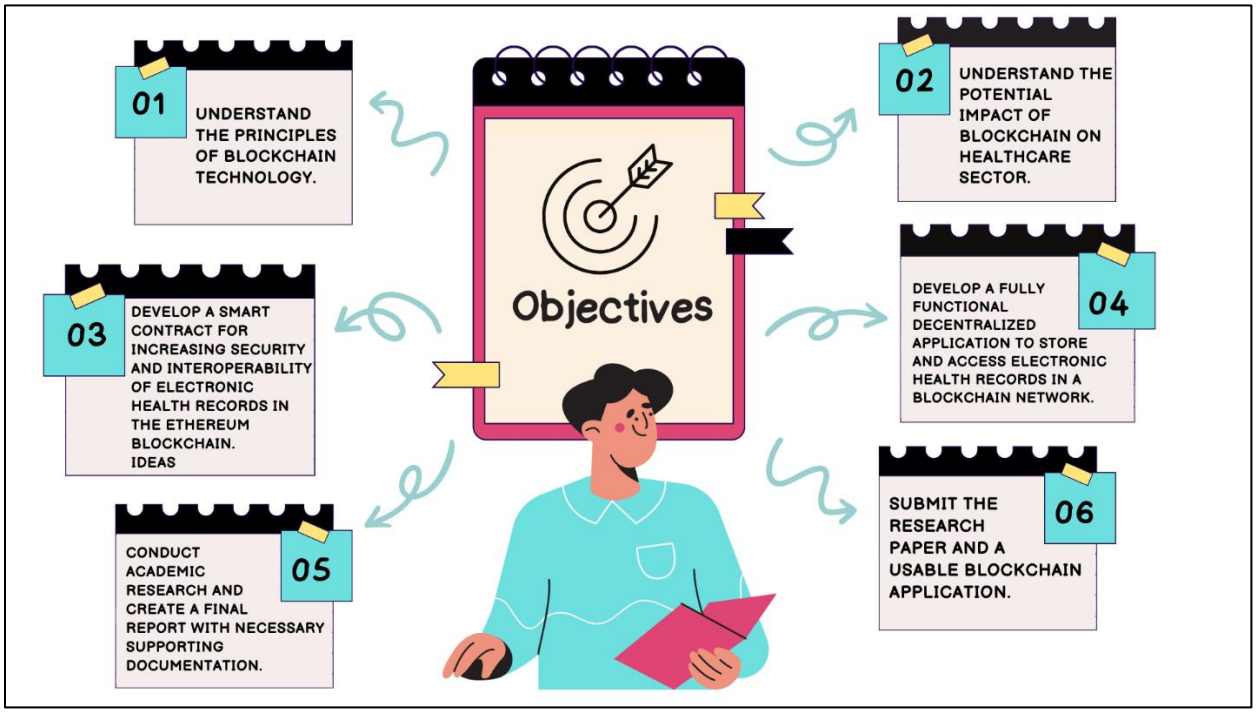

Figure 5: Objectives

Justification

Electronic health records contain very sensitive information that requires high security. With digitization, data storage methods have also seen many changes. However, the conventional methods healthcare organizations use for storing EHR are very limited. The traditional data storage methods follow a particular standard based on a single organization (Murugan et al., 2020). Thus, the records are only limited to one or only a few groups or organizations, making it very difficult for organizations to work together. Being able to share and use information among multiple organizations is referred to as interoperability. When the health records are designed based on the information of only one organization, it may be extremely difficult to maintain interoperability.

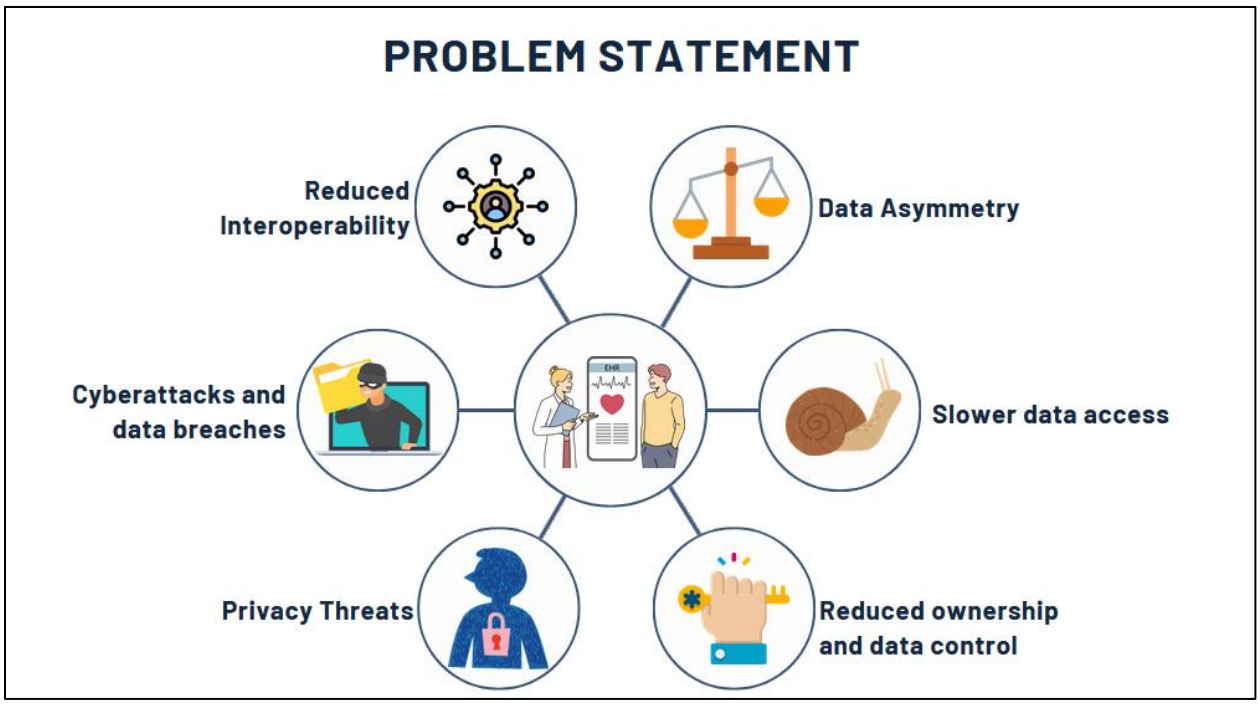

Figure 6: Problem Statement

This is not only a problem for healthcare providers but also for patients who are required to have their own physical or soft copies of their past records. However, complete interoperability cannot be achieved only by standardizing the records. For this, it is also equally important for the data to be used by the end-users, which in our case are healthcare providers and patients (Reisman, 2017). Currently, the majority of organizations and healthcare providers are seen to use centralized data storage infrastructures for storing Health records. Centralized databases provide a simpler architecture to establish a digital operation within one or many organizations (Schreier et al., Year of Publication). They are easier to manage, making it easier to manage and provide good data consistency allowing healthcare providers to better manage the patient's data. A centralized server is also easier to maintain and update. Any new changes in the database or the server of a centralized system can be deployed with much ease, especially with the help of new containerization and other DevOps technologies.

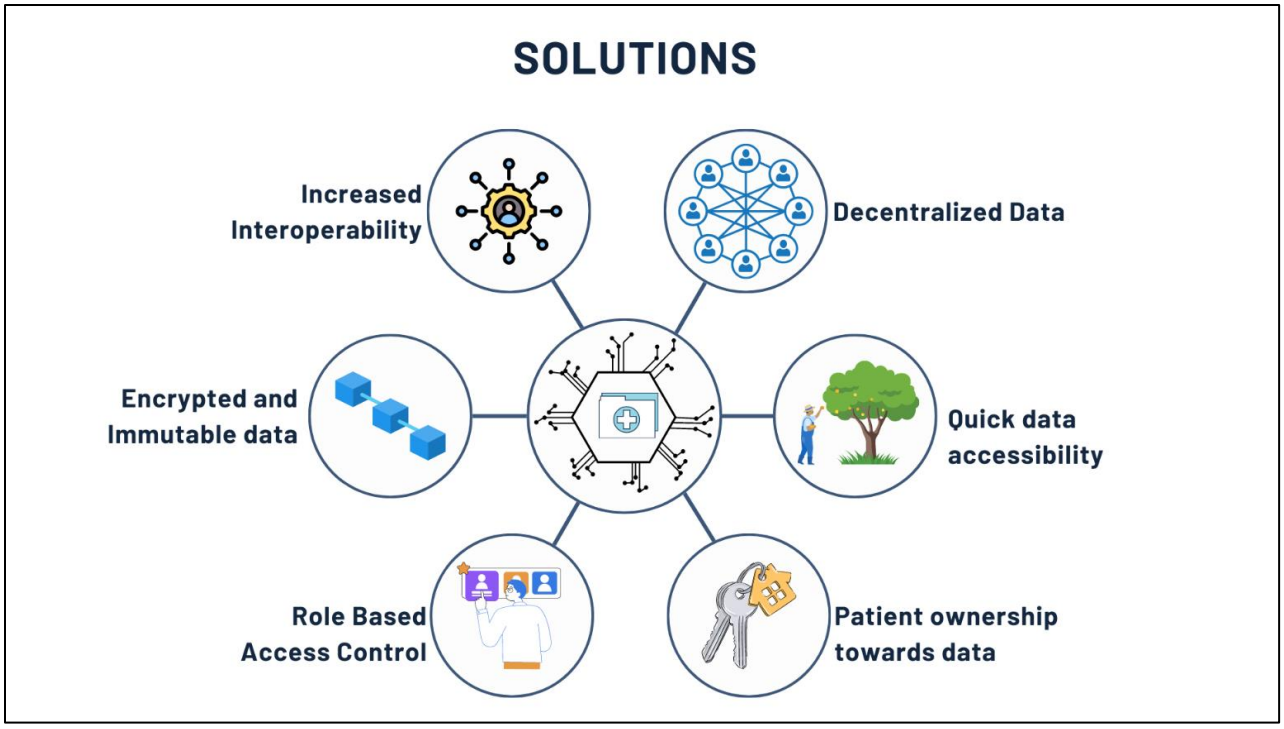

Figure 7: Solutions

However, at the same time, centralized databases are also more vulnerable to data breaches and other cyber-attacks compared to decentralized blockchain networks. Failing to provide proper security to user data can expose patients and healthcare organizations to various privacy-related issues, financial loss, and even legal problems in the future (Murugan et al., 2020). The healthcare industry has also seen a great rise in attempts at unethical cybersecurity attacks (Alharbi, Alharthi, 2021). These attacks include multiple medical information thefts, ransomware attacks, insider threats, and many more. Being a centralized system, these systems highly promote data asymmetry. This term refers to a situation where one party has higher ownership or possession of the information in a transaction. In the case of healthcare providers, centralized systems tend to lean much ownership of the data towards the healthcare providers rather than the patients who actually own the data. This asymmetry in the data further leads to other problems, such as a lack of interoperability. When more control is provided to healthcare providers, the system is usually designed for the ease of one or a few groups of organizations. When compared with a larger number of healthcare providers, a lack of data consistency can be easily recognized. Patients also go through a longer process for accessing and sharing their data with different providers. Especially in the traditional systems, patients may also need to carry physical documents of their medical history to share with other organizations or providers. Having extensive control over the patient's data can also increase the cases of unauthorized access and data forgery(Menachemi & Collum, 2011). There have been multiple cases of false medical records. These cases are mostly seen at the root of the employees, including doctors, nurses, and many other staff members in a

hospital. Unethical manipulation of these data can lead to different healthcare misconducts altering the patient's diagnosis and treatment in the future. As people fail to realize this issue, many hospital employees are seen exploiting their access to the server even today. The reason why one may look forward to forging their medical information may also conclude of multiple reasons, such as to bypass any biometric examinations for legal to transportation regions. Especially during the time of covid-19 pandemic, the creation of false medical reports has seen a massive rise in skipping tests or faking one's vaccination status (Jaclyn Diaz, 2021).

Blockchain technology looks forward to solving most of the issues in a centralized server by rather promoting a decentralized architecture for data storage. Smart contracts solve the problem of data symmetry. The technology enables the secure sharing of data among patients and healthcare providers, therefore, eliminating any exclusive control of any party over the data. Decentralized technology can also enable a consistent standard of data storage, increasing interoperability and also provide the choice of sharing historic medical records to the patients themselves (Shahnaz, Qamar & Khalid, 2019). Immutability is also one of the features of blockchain technology. This means that any data or block added to the network cannot be or is nearly impossible to remove or update (Ahmad et al., 2021). This is because a blockchain network follows a consensus protocol which makes sure that all the blocks in the chain agree on the validity of a transaction. The consensus protocol can highly reduce the unethical act of medical data forgery. And even in the case that they do take place, the digital ledger keeps a fully transparent track of all the historical transactions, which can be used by the legal departments to take measures against the individual staff and patients involved in such unethical acts.

However, decentralized technology also comes along with many challenges. In the current scenario, these changes may be very difficult for healthcare providers and patients to adapt to. Many governing bodies may also not support the concept of decentralized data as it may cause many difficulties during different investigation processes (Shuaib et al., 2022). From a technical point of view, adding new blocks in a public blockchain requires a certain amount of charge known as gas fees. These fees are used to reward the miners in the network. This can further make carrying out transactions in the chain more expensive, further increasing the overall cost of healthcare services. Even though the technology enables high security and data privacy, adapting the technology for daily use is still a great challenge not only for healthcare providers but also for patients and many governing bodies.

Research Questions

Figure 8: Research Questions

Scope

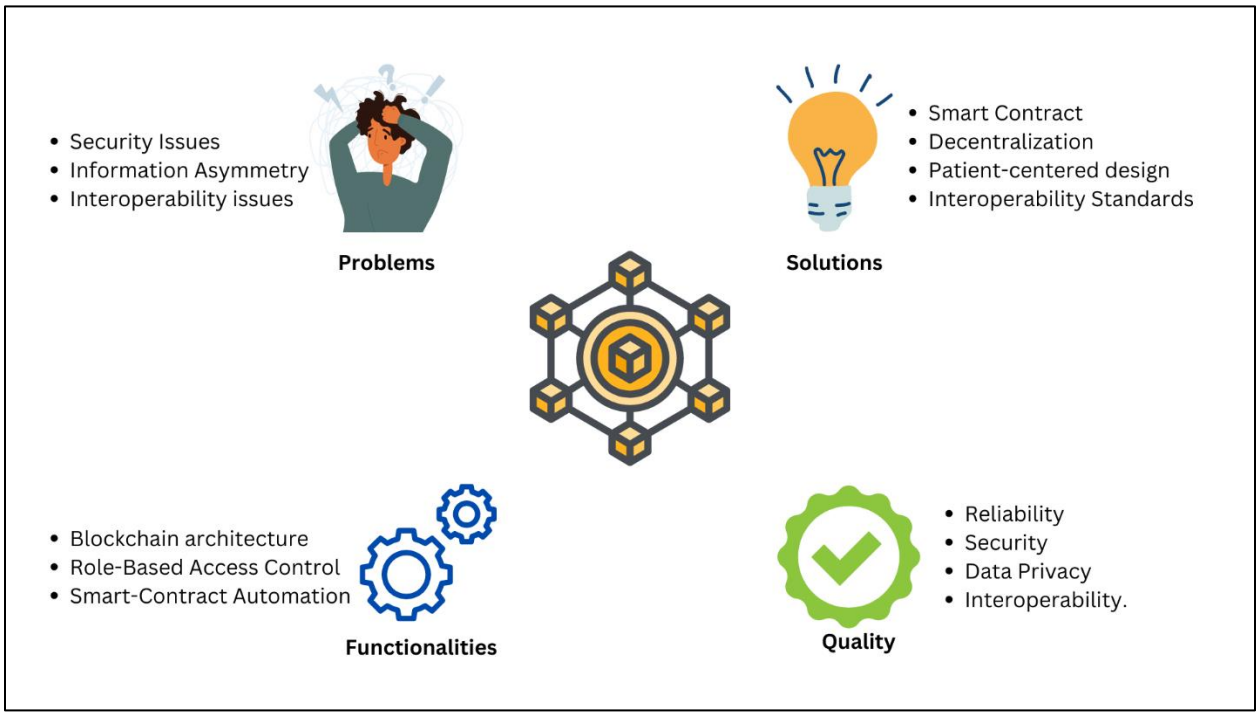

Figure 9: Scope

Ethical Considerations

A set of principles and pre-defined standards in order to ensure the ethicality and reliability of this research. The findings and arguments made during the research are placed with complete honesty and reasoning and do not intend to promote any form of bias. Likewise, all forms of bias in the research are rigorously prevented. This ensures that the readers are not influenced by bias and, consequently, enables them to independently assess the significance of this study. Moreover, efforts are made to minimize mistakes arising from the researcher's carelessness throughout the duration of this study. To provide additional support, the report will undergo periodic reviews to ensure the ongoing credibility of the results.

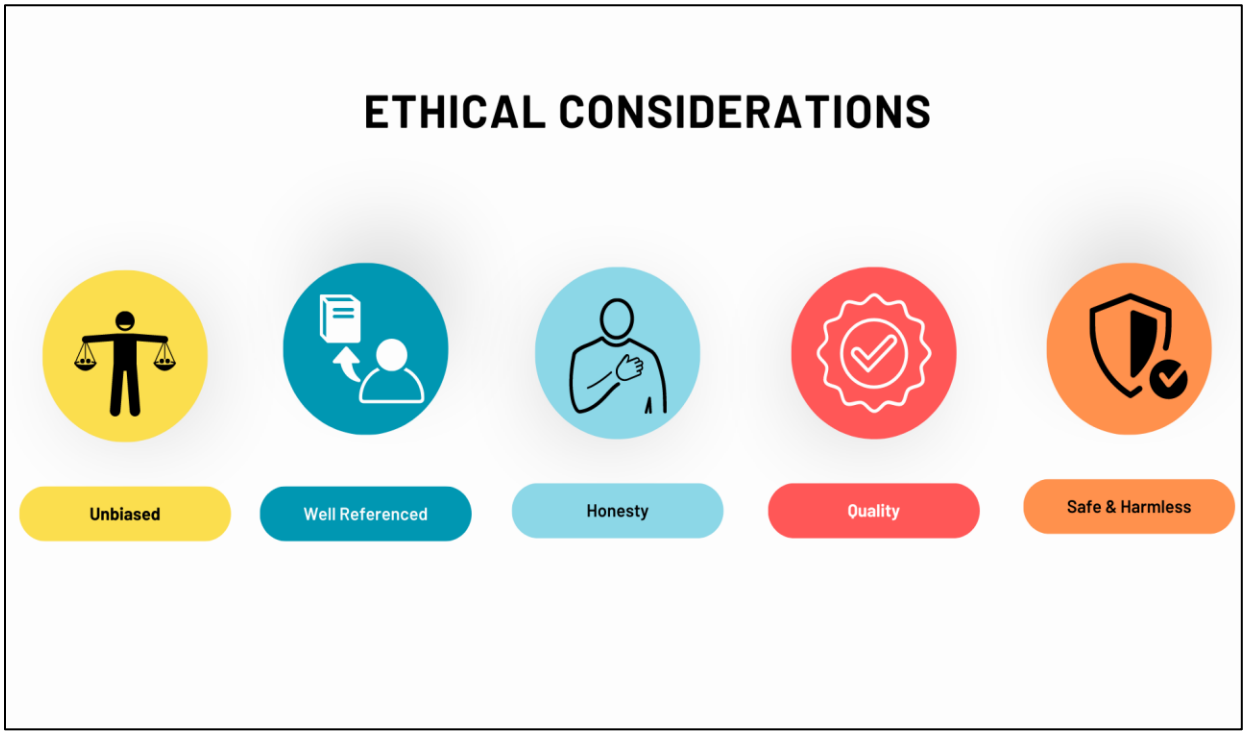

Figure 10: Ethical Considerations

In addition, the research will adhere to principles of intellectual property protection and will take measures to prevent any potential copyright or data privacy concerns. All sources, including articles, past papers, websites, journals, books, and literature, will be appropriately cited, giving credit to the original authors. Furthermore, one of our fundamental principles is to uphold confidentiality, ensuring that any information shared in private will be treated with the utmost respect and that the source's confidentiality will be preserved at all times. In this study, it is ensured that no harm will be inflicted upon any participants, whether it be physical, social, psychological, or in any other way.

Literature Review

Research Methodology

Detailed research is necessary to gain an understanding of the current scenario and existing products to minimize any error and bias in the research. Desk-based research is a methodology that will be followed for the purpose of this research. It refers to the process of conducting a detailed study of the existing resources from trusted data sources such as books, websites, journals, and many more.

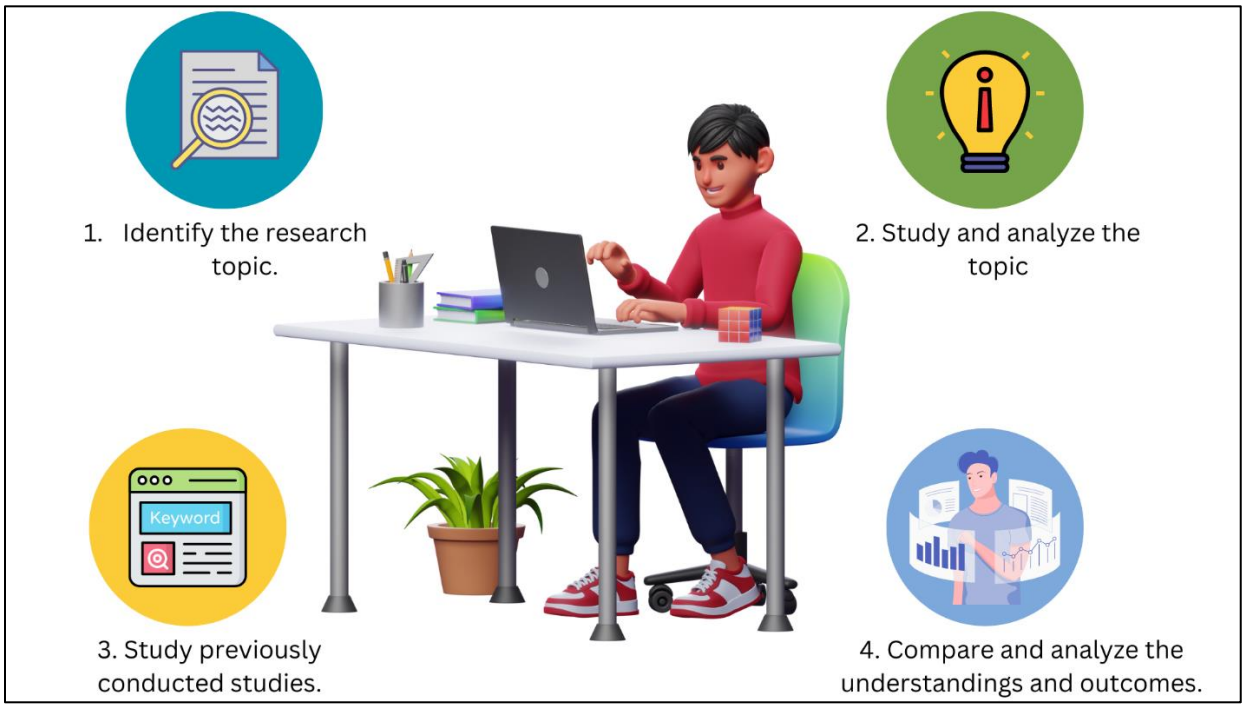

Figure 11: Desk-Based Research

However, a common misconception about desk-based research is that it is the process of collecting data sources which is false (Guerin, Janta, & van Gorp, 2018). Instead, desk-based research is a process of studying the available sources to gain a broader understanding of the topic. This method is also resource friendly and requires much less time and costs when compared to other methodologies. Alternatively, we could also conduct surveys and studies on our own. However, due to the time constraint of our research, desk-based research proves to be the best choice as it not only helps us to gain insight of the existing innovations and usage of blockchain technology, but also is easier to carry out on the basis on easily available resources on the internet at the ease of our devices in very less time. Academic publications and peer-reviewed journals and articles shall also be highly prioritized as they are recognized as the most reliable sources for conducting secondary research. This research process is initiated with an extensive study of different collected resources and multiple case studies of existing research and products. As the research topic aims to identify the implementation of blockchain technology for storing Electronic Health Records, the

case studies shall mostly revolve around products and innovations around digital healthcare systems and blockchain based decentralized applications allowing us to gain a deeper understanding of the domain.

Patientory

Patientory is a blockchain-based healthcare platform that aims to revolutionize the storage of healthcare data. Founded by Chrissa McFarlane in December 2015, the platform was the outcome of a goal to promote a personalized, secure, and consumer-driven management system for healthcare information (Tardif, 2021). McFarlane, the founder and the CEO of Patientory, is also recognized as one of the top women making an impact on the MedTech sector of the health IT industry by Becker's Hospital Review (Ribitzky, Broedl, McFarlane, & Clauson, 2018). Patientory is the proposed solution to having a decentralized repository of all medical records providing users with total access control over their data.

Figure 12: Patientory

The platform aims to make accessing healthcare data easier and faster for all stakeholders, from patients to doctors, hospitals, and healthcare providers. Along with managing medical records, Patientory also serves as a healthcare service provider by enabling users to track their health goals, monitor their progress, and allowing them to communicate securely with their healthcare providers. The use of blockchain technology allows secure storage of data in the network. Moreover, the technology and the use of smart contracts allow Patientory to enable strict Role-Based-Access-Control (RBAC) in the platform. RBAC is a form of access control where the ability of users to access information on a platform is controlled based on their roles (Sandhu, 1998). While role-based access control is one of the most popular forms of access control used by many platforms, Patientory makes use of this feature by providing different roles to users and healthcare

providers. This allows doctors and patients to use the platform on different user interfaces and enable high-level authorization. RBAC allows users to access their personal information, such as medical records, lab results, and healthcare goals. Similarly, Doctors are given access to view and add medical records associated with the respective patients. Additionally, this access control also ensures user permissions in the system. i.e., doctors initially need to request access from the patients to view and add the medical records, and users can likewise allow or decline the requests placed by the doctors and healthcare providers. This form of access control, combined with the secure storage of information using blockchain technology, allows the platform to ensure security and data confidentiality to its users.

Patientory also facilitates seamless interoperability by lowering the barriers across different healthcare systems. The platform provides a well-standardized format for storing medical records that allows information sharing across multiple healthcare providers (Ciampi et al., 2019). Additionally, the platform also provides care plan management services. These are personalized healthcare plans for users to meet their health goals. These care plans are generated based on the user's medical records, health conditions, and other similar factors. This allows users to get a visual representation of their current health status and also track their progress to meet their goals (Ciampi et al., 2019). This is one of the features that attract users to use Patientory.

Patientory is a HIPAA-compliant platform that adheres to all security and storage requirements for personal health information. The platform offers analytics and reporting functionalities to healthcare organizations to assist them with care and coordination among the users. The use of blockchain technology further enables security and confidentiality among the parties. As mentioned, Patientory is HIPAA-compliant, thus providing safe communication between patients, doctors, hospitals, and other stakeholders and healthcare professionals.

Moreover, the healthcare platform also empowers its own cryptocurrency, known as PTOY tokens (Dedeturk et al., 2021). As releasing a cryptocurrency is one of the forms of raising investment capital for most blockchain startups, PTOY tokens are also a similar approach made by Patientory. These tokens are ERC-20-compliant cryptocurrencies built on the Ethereum blockchain (Dedeturk et al., 2021). Being a utility token, all the transactions in the Patientory platform are carried out by using these tokens. These tokens allow secure transactions in the platform and also incentivize network participation. Additionally, they also allow users to access selective features in the platform. The tokens were initially launched in an initial coin offering in 2017. During when, 1728 buyers from around the globe participated and purchased a total of 70 million PTOY tokens (StartUp Health, 2017). This event allowed the platform to raise about 7.2 million US dollars of capital (StartUp Health, 2017). This was a historic moment for Patientory as it not only helped raise such a great amount of capital but also because it reflected people's trust and support for innovation. These tokens not only provide users with access to different features in the platform but have also caught the eye of many investors as they expect these digital assets to appreciate in value over time. The platform provides an option to purchase these tokens in exchange for Ethereum tokens in the application itself. Alternatively, these tokens are also listed in many cryptocurrency exchange platforms like Bittrex, TokenMarket, HitBTC, ICObazaar, and Liqui

exchanges. These platforms can also be utilized to buy or sell PTOY tokens (StartUp Health, 2017).

In summary, Patientory is a healthcare platform that aims to transform how medical data is stored by utilizing blockchain technology. The platform uses Role Based Access Control to ensure a high level of authorization between healthcare providers and patients. By removing obstacles across various healthcare systems, Patientory also promotes seamless interoperability. The platform gives healthcare institutions analytics and reporting capabilities to help them with user care and coordination. Patientory is also a HIPAA-compliant platform that complies with all standards for data protection and storage. The healthcare platform also uses PTOY tokens, a proprietary cryptocurrency. These tokens, which are used for all platform transactions on Patientory, enable safe transactions and encourage network activity. They can be purchased directly from the platform in exchange for Ethereum tokens or from a third-party cryptocurrency exchange.

AAVE

Aave is a decentralized platform that allows users to lend and borrow money. Unlike our traditional banks that provide cash loans, Aave provides crypto loans (Paliwal, 2022). With the increasing popularity of cryptocurrencies and Bitcoin reaching its all-time high in the past few years, the innovation of blockchain technology has displayed much of its financial benefits. Bitcoin was the first-ever cryptocurrency that proposed the idea of decentralized finance. Its whitepaper, dated back in 2009, introduced the concept of "proof of work," which refers to a digital protocol that requires the members of a network to make an effort to solve a hash algorithm to generate a correct hash (Laurie & Clayton, 2004). Doing so rewarded the users with cryptocurrencies. Soon later, people would discover the financial value of these cryptocurrencies and would also name them as digital assets. Since then, many new cryptocurrencies have come into existence. It has also been somewhat of a trend in the blockchain industry to create one's own digital currency, especially among companies working on blockchain-based projects. With so many cryptocurrencies in the market, many of them have showcased their own unique financial values. These currencies are listed on many trading platforms and crypto exchanges where people can buy and sell their assets. This trading mechanism further promotes the change in the value of cryptocurrencies.

AAVE, on the other hand, is a crypto project that, as mentioned before, promotes the lending and borrowing of different crypto currencies. Unlike the "proof of work" consensus, AAVE follows the "proof of stake" mechanism (Aave Document Hub, 2022). As the name suggests, the protocol is based on staking or depositing, which is similar to the concept of fixed deposits in traditional banking systems. The amount staked by depositors is stored in what is called a liquidity pool (Makarov & Schoar, 2022). It is named so because the amounts that are deposited into the pool are what add value or liquidity to its token. Likewise, whenever anyone wants to borrow cryptocurrencies from the platform, the amount is also received from the liquidity pool. As per the proof of stake consensus, depositors receive a certain amount of AAVE tokens as a reward for staking (Bentov et al., 2014). The borrowing process is also an over-collateralized form of

borrowing. i.e., In order to borrow a certain amount of crypto in AAVE, one must first deposit a different crypto of higher value in the platform.

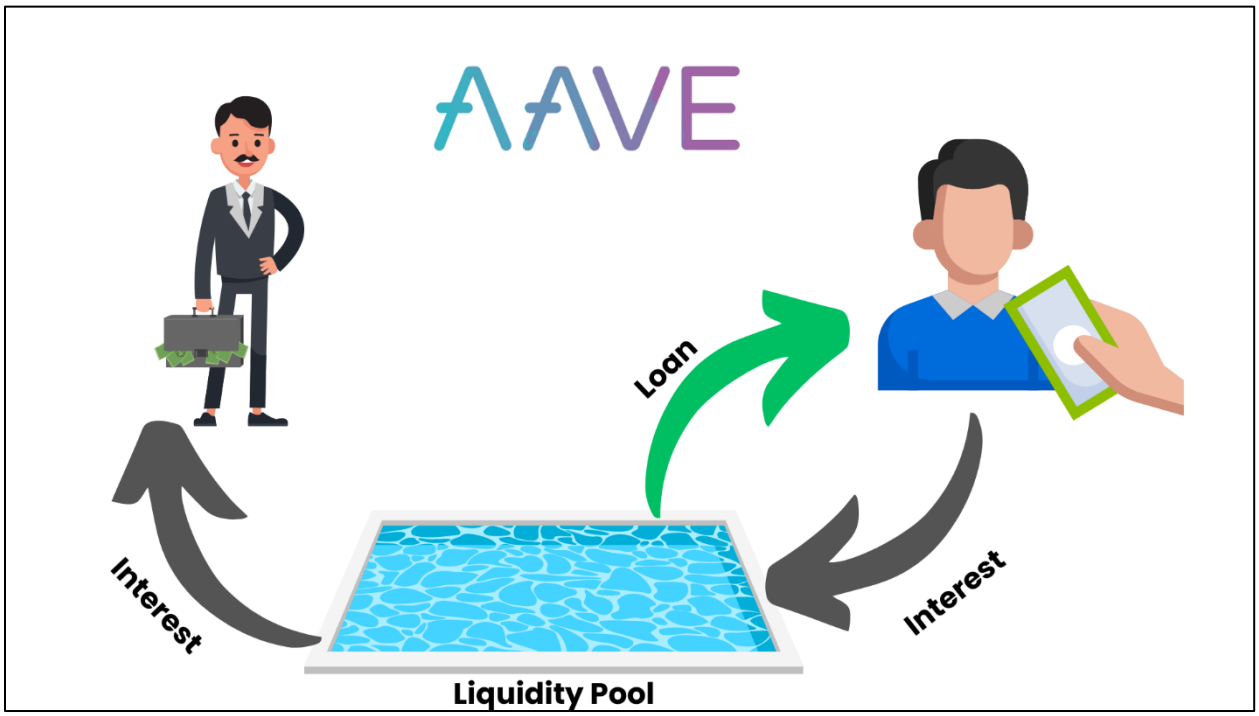

Figure 13: AAVE

Another mentionable feature of AAVE is its flash loans. This is one of those features that have the potential to revolutionize the digital finance industry (Paliwal, 2022). Flash loans allow users to borrow assets without the need to deposit any collateral. As confusing and risky as it may sound, the smart contract for flash loans is architecture in a way that the repayment of the loan is also a part of the same block in the chain. Therefore, in case a user fails to pay back the loan in time, the entire transaction is reversed, thus eliminating the risk for the platform. Flash loans involve high risks along with equally higher rewards enabling traders and investors to gain good profits. However, it also requires great analytics skills along with proper financial planning.

Being backed by smart contracts, the entire transaction process in AAVE is fully decentralized and automated. As there is no human interaction during the entire process, there are no emotional dependencies between the users and the transactions. In the traditional banking systems, it is a common practice for people to ask for extensions for extra time from the bank for loan repayment. Even though many times, people may have genuine reasons, most of the time, people tend to lie in case they fail to pay back the loan. AAVE is very strict with its deadlines. The due date for payment is specified before the loan is provided to the users. In case the user fails to pay back the borrowed amount or the interest in time, the smart contract automatically liquidates the collateral amount (Aave Document Hub, 2022). This liquidation results in the user losing the collateral amount entirely. And as the borrowing process is based on an over-collateralized system, this usually results in a larger financial loss for the borrower.

AAVE highly prioritizes user security by providing multi-layer protection to secure users' funds in the liquidity pool. Many well-reputed security firms are also empowered to audit smart contracts to minimize any vulnerabilities in the system. And since AAVE is involved with multiple different cryptocurrencies whose financial values are extremely variable, it utilizes decentralized oracles, which are systems that provide off-chain data to the smart contracts. In our case, the off-chain data refers to the prices at which the cryptocurrencies are currently being traded.

Patientkeeper

Patientkeeper is a healthcare software company that works on providing services to make accessing electronic health records much easier. It provides an easy-to-use application interface compatible with mobile phones, tablets, and desktop computers. The company was founded in 1997 by a healthcare venture capital firm known as General Catalyst (Zarka, N., Hinnawi, M., Dardari, A., & Tayyan, M. A., 2004). The platform works to provide ease of access to patients' healthcare information coupled with the workflow of the doctors and healthcare providers. It does this by simply creating an easily accessible interface on the device preferred by the doctor through which they can view the list of their patients and their recent reports . This not only serves to make the EHRs more accessible but also positively impacts the workflow and productivity of healthcare providers as they have access to their patient's information from anywhere, they want. Additionally, doctors also have access to enter patients' information, such as their diagnosis reports and billing information, into the system. Furthermore, the platform is also HIPAA compliant and takes high security and privacy measures to safeguard the patient's data. It also provides other additional features, such as secure role-based messaging, on-call scheduling, and VoIP calling to enhance the collaboration between healthcare providers and patients. PatientKeeper is used by many hospitals to contribute to their workflow around the world. It was acquired by HCA Healthcare in 2014 (HCA Holdings Inc, 2014). HCA Healthcare owns more than 100 hospitals and surgery centers across USA and England(HCA Holdings Inc, 2014). With this, PatientKeeper has been able to expand its usage among these healthcare institutes as well.

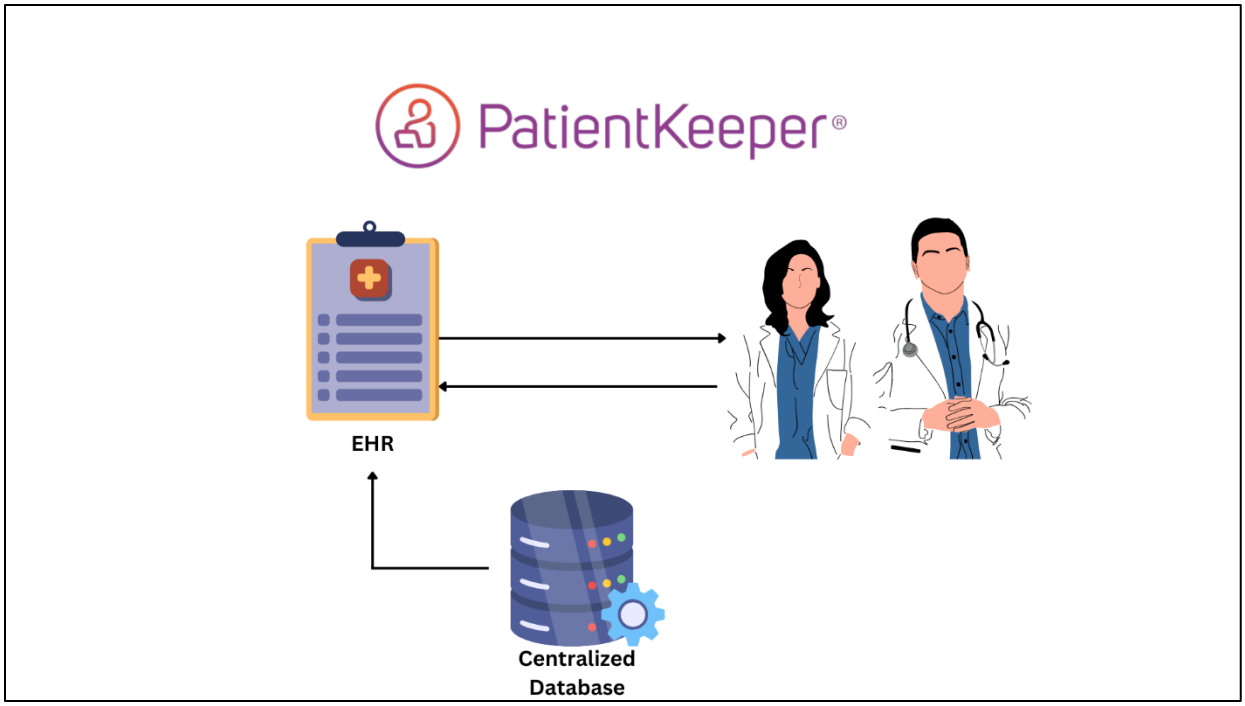

Figure 14: PatientKeeper

PatientKeeper uses a centralized system design for keeping records of patient data. The system is designed in a way to suit the healthcare workflow of most organizations. It provides easy access to healthcare data. Despite being a centralized system, PatientKeeper's design highly promotes its usage among healthcare providers without affecting the usage of previously used tools and applications. Instead of revolutionizing the entire EHR system, PatientKeeper's design is built to simply complement the previously existing EHR systems. This allows healthcare organizations to be able to use and adapt to PatientKeeper without any additional time and resources, significantly reducing the implementation cost of the application. In 2021, PatientKeeper was acquired by Commure, a healthcare company working to use the power of technology to enhance healthcare services (Commure, 2021). PatientKeeper intends to utilize Commure's cloud-based platform in order to establish smooth connections among various healthcare systems and facilitate workflows that prioritize individuals (Muoio, 2021). This collaborative effort contributes to the overarching objective of enhancing interoperability within the healthcare sector (Commure, 2021).

In summary, PatinetKeeper is a healthcare software company that aims to provide digital workflow services for healthcare providers and ease the accessibility of patient medical records. With multiple historical acquisitions, the platform has been able to expand its usage and make use of different benefits provided by the acquirers. With its contribution to making medical records more easily accessible and enhancing the workflow of healthcare providers, the application also provides an easy-to-use interface for different devices and features such as secure messaging and VoIP for further enhancing the interaction between healthcare providers and patients.

Development Methodology

As our product is a blockchain-based software as a service (SaaS) product, agile has proven to be one of the most popular and widely used development methodologies suited for our purpose. Considering the potential changes in requirements and the need for changes during the development period, the agile framework may also provide for such situations. Agile, or scrum, to be more specific, is a development methodology that relies on incremental development. The concept suggests breaking down a project or a complex task into multiple iterations, which are known as sprints. Each sprint lasts for a constant specified amount of time which is usually one or two weeks in general for enterprise-level projects. The agile methodology is also expected to suit our purpose best, as following the iterative approach can allow us to divide our product into multiple features.

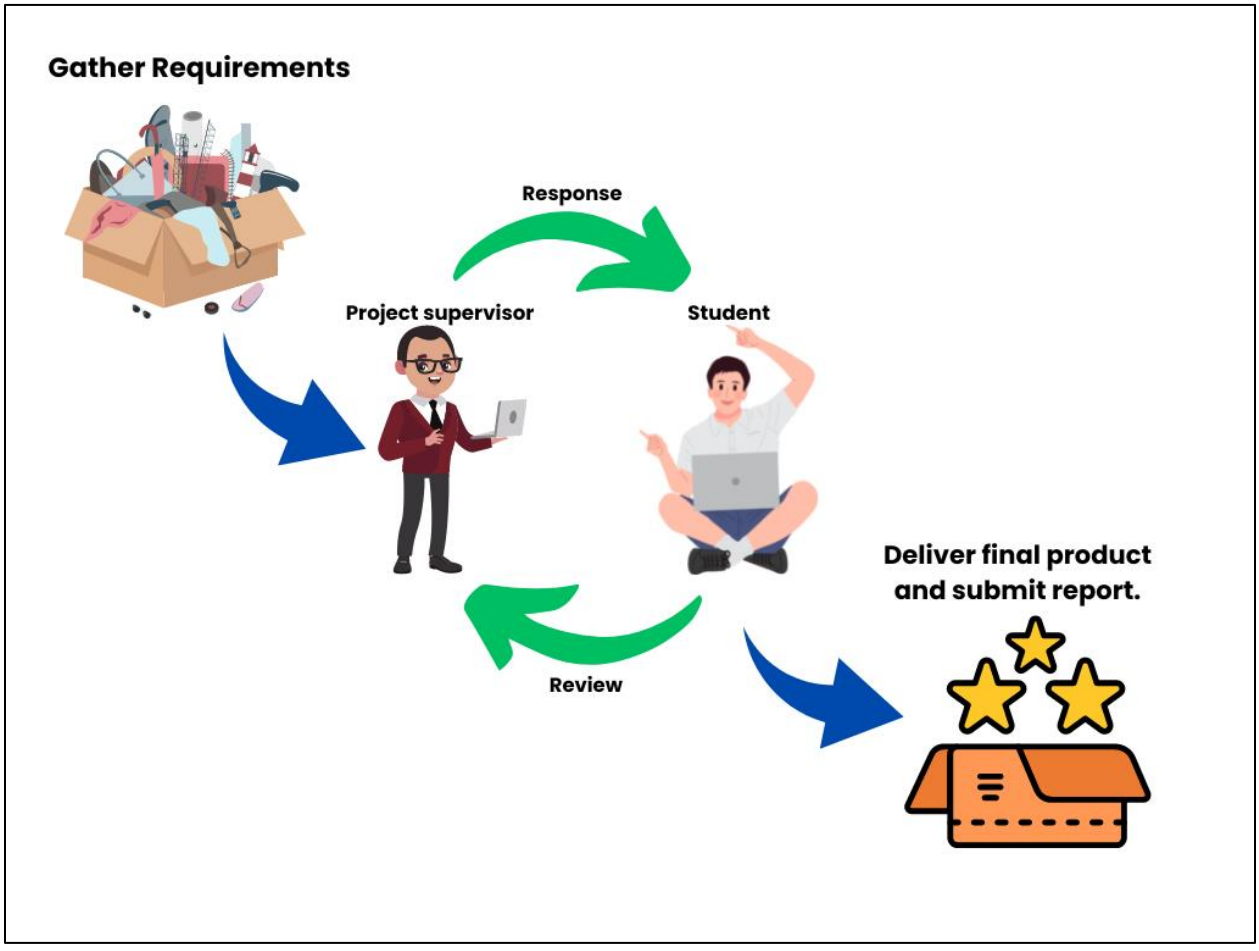

Figure 15: Development Methodology

The first stage of agile constitutes requirement gathering. During this phase, research was conducted to analyze the necessary tools and technologies and all the requirements that need to be met by our product. These requirements also include research on the functional and non-functional features of our application. This is also one of the most crucial phases of our research as it acts as

its foundation layer for serving the rest of the processes. Further, we try to understand the scope and objectives of our research and analyze the estimated cost and resources for our planning. The requirement-gathering phase also constitutes the process of system design and analysis. This refers to the study of the choice of design patterns and system architecture, user interface design, user roles, access control, and other system-level analysis for our infrastructure. This further serves the understanding of our research along with the overall maintainability and performance of our system. Once we gather the overall requirements for our research and development process, we then dive into the iterative phase of our agile lifecycle. Using the gathered requirements as our foundation, we then begin the development of our product and analyze our product, and research alongside it. This is expectedly the longest and the most time and resource-consuming phase of our development lifecycle. However, agile enables us to break our entire development life into multiple sprints. For each sprint, we assign ourselves multiple user stories, which are well-described phrases for our application features. These sprints are totally dedicated to completing the selected user stories will the least distractions and external noises. During this process, we also develop our smart contracts and work on developing our user interface with an initial goal of developing a minimum viable product (MVP) (Knapton, 2022). In our case, each sprint was taken for the length of one week. The tasks associated with these sprints were also illustrated through map boards and other project management tools that further allowed us to visualize and map the progress of our research. Additionally, in order to develop a well-designed product with fewer bugs as possible, we also follow the concept of Test-Driven Development (TDD). This process promotes writing units and other tests for our features during the development period itself (Baldassarre et al., 2021). TDD also serves as a great methodology for error tracking in the system and helps us to resolve bugs during the early stages of our application (Baldassarre et al., 2021). This cycle of continuous development, testing, and refinement continued for a total of 8 sprints in order to come up with our final blockchain application for securely storing and accessing Electronic Health Records.

Tools and Technologies

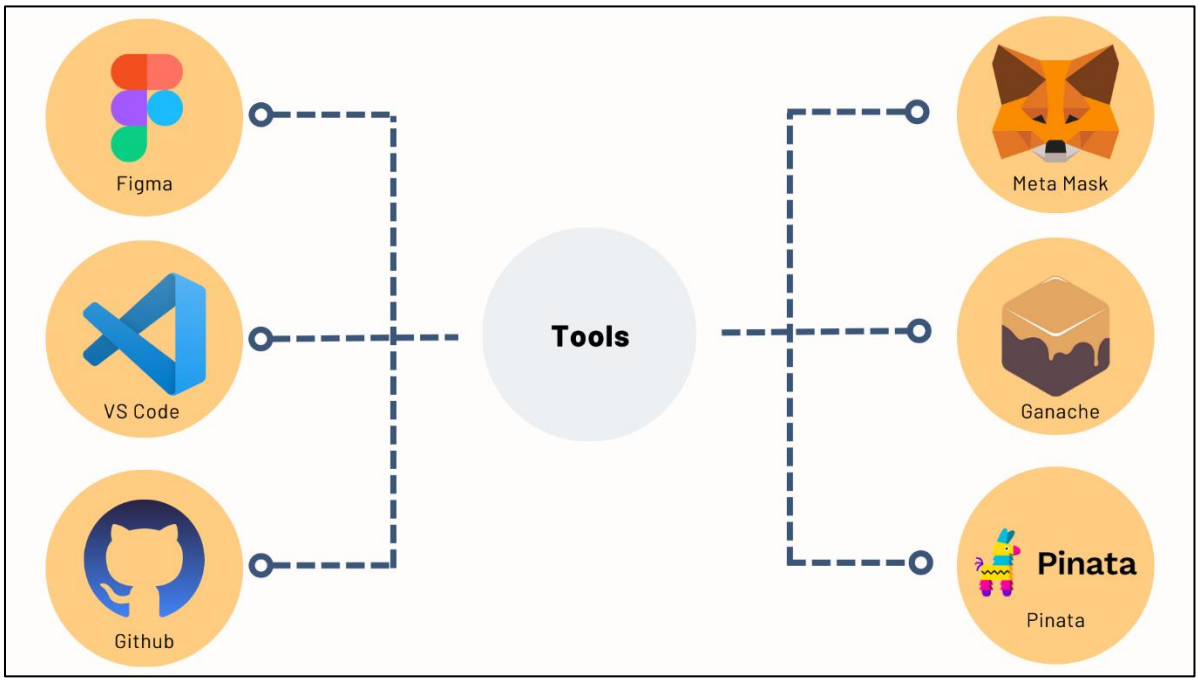

Figure 16: Tools

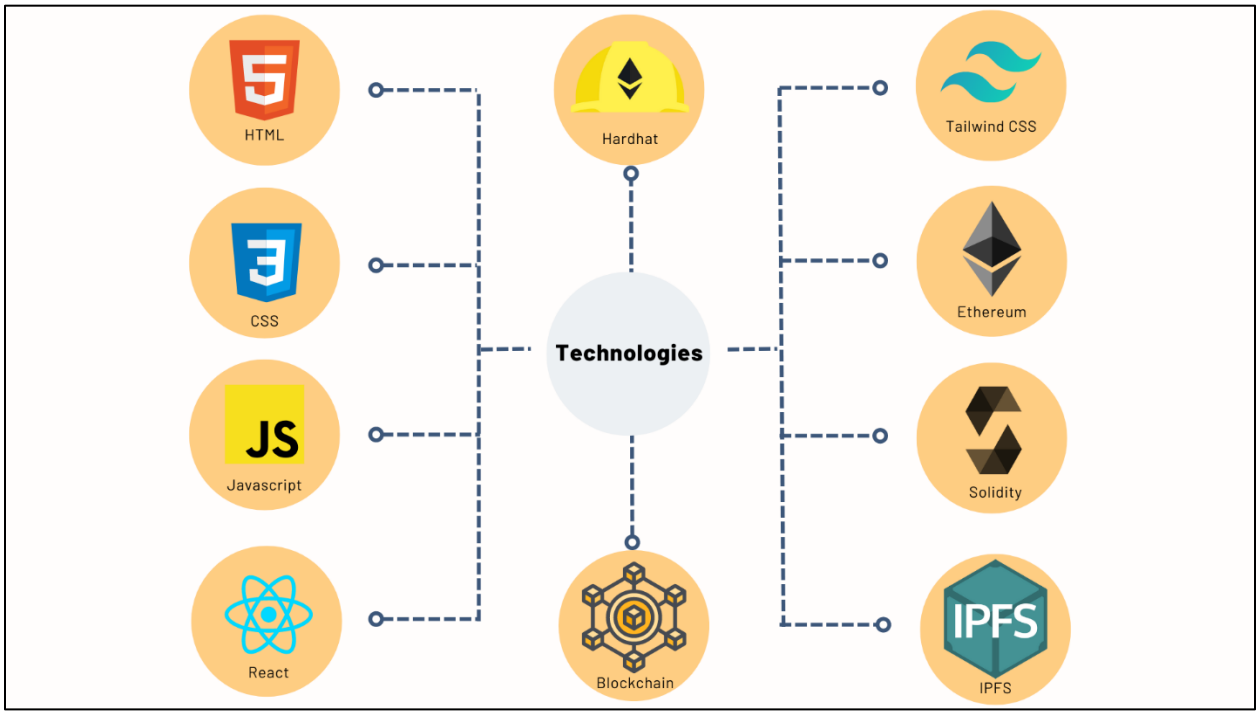

Figure 17: Technologies

Development Process

The process of developing a decentralized application for securely storing and managing electronic health records began with a study to understand the meaning of the purpose of an EHR in general. During this phase, we analyzed the need for digital storage methods and options for storing medical records. The current scenario of these storage techniques and their impact on the industry was also carefully studied and analyzed. With the need for a highly secure and immutable record-keeping system that enabled greater interoperability, the use of blockchain technology was seen to be the best option among all to fulfill our initially analyzed requirements. With this, we then studied the tools and technologies used to develop decentralized applications. During this phase, we understood the concept of smart contracts and how they enabled secure automation in a blockchain network. Further, we studied the proper design patterns followed by existing decentralized applications in order to gain knowledge about proper design principles to ensure the maintainability of our final product. Based on our analyzed tools and technologies, we set up our development environment on VS Code, as it provides many open-source extensions and modification options for the technologies that we would be using in the future. The hardhat development environment was used for developing our smart contract. The Ethereum software development environment Hardhat provides many components and modules for editing, compiling, debugging, and deploying your decentralized applications and smart contracts. Followed by Hardhat, we then began to write our smart contracts using the Solidity language, which is a popular programming language for writing Ethereum-based smart contracts. As we follow our Test-Driven-Development (TDD) principles, we use Chai. It is a JavaScript assertion library that allows us to write unit tests for our solidity smart contracts in Hardhat. Writing proper unit tests allows us to ensure our smart contract functions are provided to our intended output. Our smart contract can be compared to a server-side application in a full-stack application. Once deployed, our smart contracts can serve just like an API layer in a server. Therefore, to deploy our application, we use Ganache. Ganache provides a personally hosted Ethereum blockchain network that allows us to locally test our smart contracts during the test and development process.

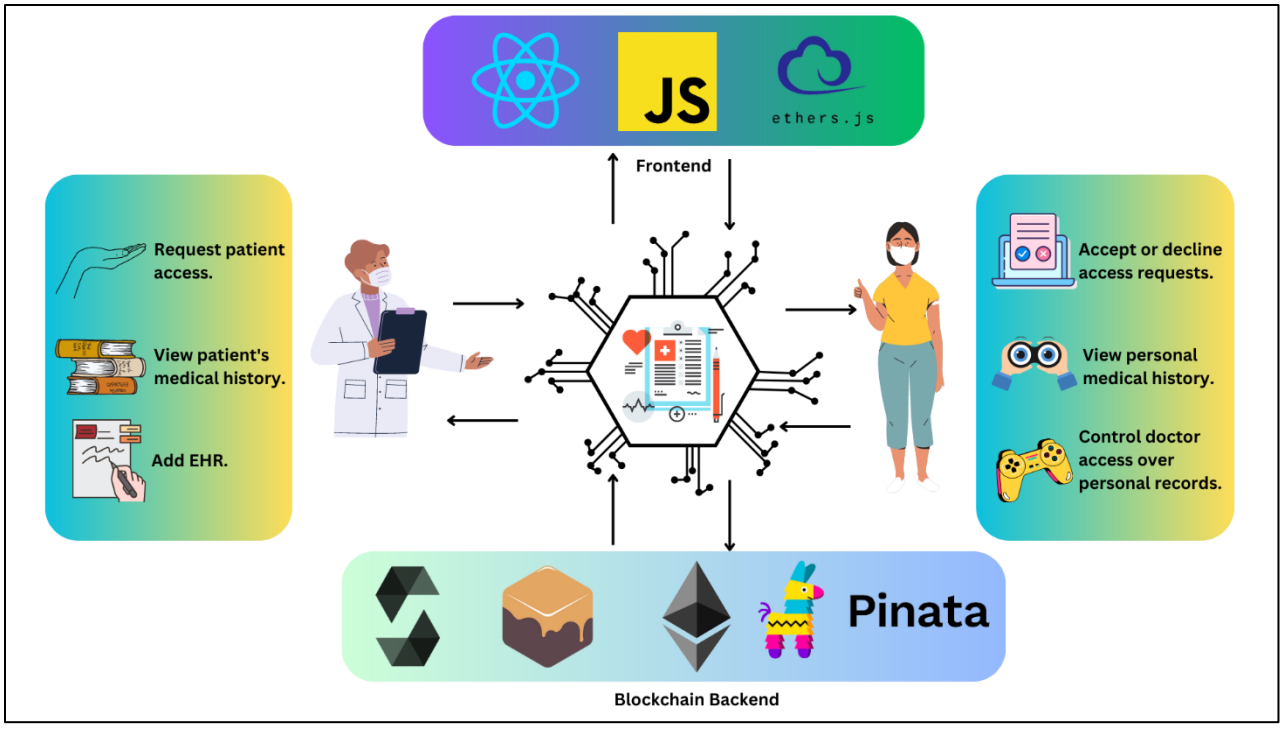

Figure 18: Development Process

Once the smart contract has been successfully deployed in our Ganache server, our next step is to design and develop the client side of our application. The client side or the front-end is the layer of our application through which the users interact with our smart contract. Using Figma, we develop a high-fidelity prototype of our User Interface to generate a look of our application before beginning the development process. Developing a prototype of our application helps us to analyze the usability of our application and bring changes in an earlier stage to save time and resources in the long run. We have chosen React as our front-end JavaScript framework for developing the User Interface as it provides easy-to-use libraries for developing decentralized applications. Additionally, we use the ethers package available in the npm repository for JavaScript packages. This library allows us to connect our client application with the smart contract and execute the available functions likewise. We also need an off-chain storage facility for storing files in our application. Since files cannot be stored in a blockchain, we use an Interplanetary File System (IPFS). An IPFS is a distributed file system that enables peer-to-peer file sharing (Chen et al., 2022). It is the preferred way for many decentralized applications to handle files whilst maintaining the decentralized attribute of the system. Finally, we have also used GitHub as our preferred version control system. Using a Version Control System (VSC) allows us to manage changes in our application. GitHub also allows issue logging helping us to document any issues and hurdles during our development phase.

Secondary Features

User Experience

The user interface of the application is designed to be very minimalistic. Providing only the necessary information on a page at a single time allows users to easily understand the system's behavior, enhancing the overall experience of the user. The UI also provides easy navigation options from the navigation bar and the usage of other buttons and hyperlinks to allow users to easily explore the system.

Interplanetary File System (IPFS)

Since our system is a blockchain application, we are unable to directly store images and other files in the network. For this reason, we use a technology known as IPFS, which is a peer-to-peer network for sharing files (Nizamuddin, Hasan & Salah, 2018). Our system needs to store the image of different attributes in the system, such as the images of hospitals and doctors, including the files of the medical records. Therefore, we use IPFS for storing these files in an off-chain network which can then be accessed using the provided hash. The hash is then stored in our blockchain network, pointing to the address of our file in the IPFS.

Accessibility

The system is designed to serve as a web application. Thus, it can be easily accessed by users from any device from a web browser. Furthermore, the user interface is also highly responsive. Meaning it can be used on devices of varying screen sizes, from mobile, tablets, and desktops.

Findings

How can blockchain technology be used to increase interoperability and accessibility of electronic health records?

The Health Information and Management Systems Society defines interoperability as the capacity of various information technology systems and software applications to exchange information and use the information that has been exchanged (Mettler & Längst, 2018). Blockchain technology provides a decentralized design for storing data. Using this technology, storing EHRs in a blockchain network can help healthcare providers better manage the data and increase interoperability between multiple healthcare organizations (Margheri et al., 2020). This network can also be a central repository for patients and healthcare providers to access all the necessary data. Unlike centralized database systems, the use of blockchain gives more control and authority to the patients over their data instead of to the hospitals or other organizations (AbuHalimeh & Ali, 2023). With access to their data, users now have the power to choose to share their data with organizations with much ease. Therefore, the technology enables the storage of users' medical records in a single storage point, making them more accessible and secure.

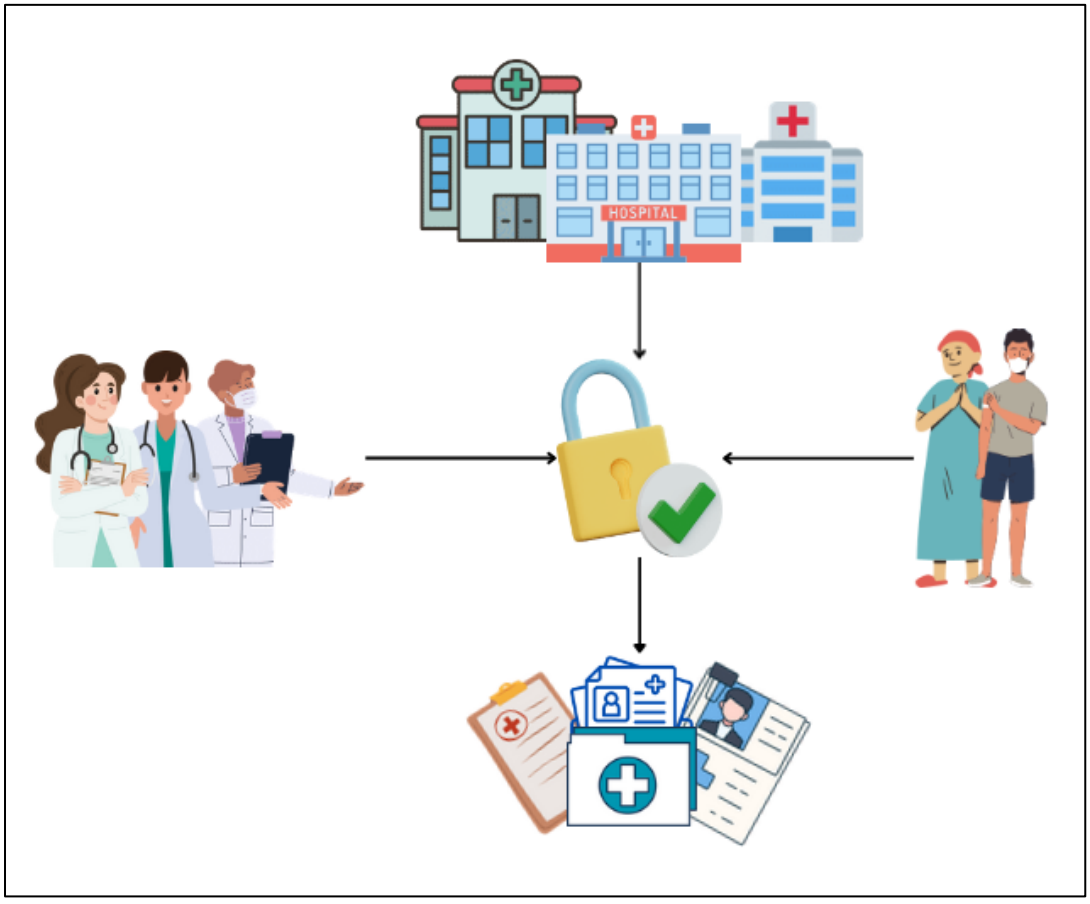

Figure 19:Blockchain technology to improve interoperability of EHRs

Having a single point of access also enables standardization of data. Multiple healthcare providers can receive and process the records and information in a specified standard or format. The uniformity in data also makes it easier for patients to understand their medical reports without much professional assistance (Walker et al., 2018). This massively impacts the interoperability of the records as the traditional EHR systems need to promote the standardization of data. Even in the cases where they do, the data format may still differ from one healthcare provider to another. This makes it difficult for organizations to work together and, moreover, affects how patients and users receive healthcare services.

Medical records are extremely sensitive information associated with individuals. It is necessary for healthcare providers to be able to access this information during the time of emergency as quickly as possible with much ease. This can have a significant impact on the quality of treatment patients receive. Therefore, the enhancement in the accessibility of data can majorly impact the experience people have while receiving and providing healthcare services. The patient-centric design enabled by blockchain technology also allows users to easily access their data. In many cases, patients have a hard time trying to access their own data due the data asymmetry. While patients have the ease of accessing their information whenever they want, this can also create a positive impact on individuals to regularly seek healthcare services for any early diagnosis and health precautions.

While the traditional EHR systems have massively impacted the way we receive healthcare services, the centralized design is still not enough to maximize data accessibility and interoperability. With this design, hospitals and healthcare providers also have the ability to make it troublesome to receive similar healthcare services from other competing providers. This is an unethical corporate activity that most organizations play not just in healthcare but in almost every other sector as well. However, this may be an even more serious problem when the matter tends to deal with the health of people as it directly affects people's lives. The immutability of the data stored in a blockchain network makes the information more traceable. Complete interoperability is not achieved by just making information sharing possible but when the information is actually used among multiple organizations or healthcare providers. Even though organizations can access user data, there may still be a question of whether to trust the information or not. The immutability of information in a blockchain means that once a transaction has been made in the blockchain, it is very difficult or close to impossible to change or pamper that data. This allows auditing and storing secure logs of data in the network. The medical records in the chain can be provided with metadata stating the details of the issuer and their license assisting with the validity of the block. This patient-centric design not only allows patients to retain full ownership and direct access of their data, but the asymmetric encryption in a blockchain can also help ensure the validity of the data in the network. Consensus methods such as proof of work can enable patients to act as a network node to actively maintain the validity of the data in the chain.

How can blockchain technology help increase patient's authority over their data and reduce information asymmetry?

Information asymmetry is when one party has more access and authority over the data than the other. This is a common issue with centralized data systems, due to which many large corporates have been seen exploiting the user's data. This is due to the way these systems are designed to lean more over toward these companies. Data is also referred to as the oil of the digital era. This refers to how oil could be more useful in its raw form but can be used for various purposes once refined and processed (Shi et al., 2020). Similarly, raw data may have little meaning, but after some cleaning and refinement can produce tremendous value to a business, organization, or individual. Therefore, we need to be very careful about who owns and has control over our data. Most healthcare record systems today are based on centralized designs providing hospitals and healthcare providers with great control over our data (Kiania, Jameii, & Rahmani, 2023). As mentioned, this massively impacts the accessibility of the data. But furthermore, in many cases, this is also a direct threat to users' privacy. This type of information asymmetry can also create an unequal power dynamic between the patients and the healthcare providers, further creating the need to rely highly on them for decision-making. The centralized system designs are designed to highly couple the user's information with only a few selected groups of organizations forcing them to rely on their services.

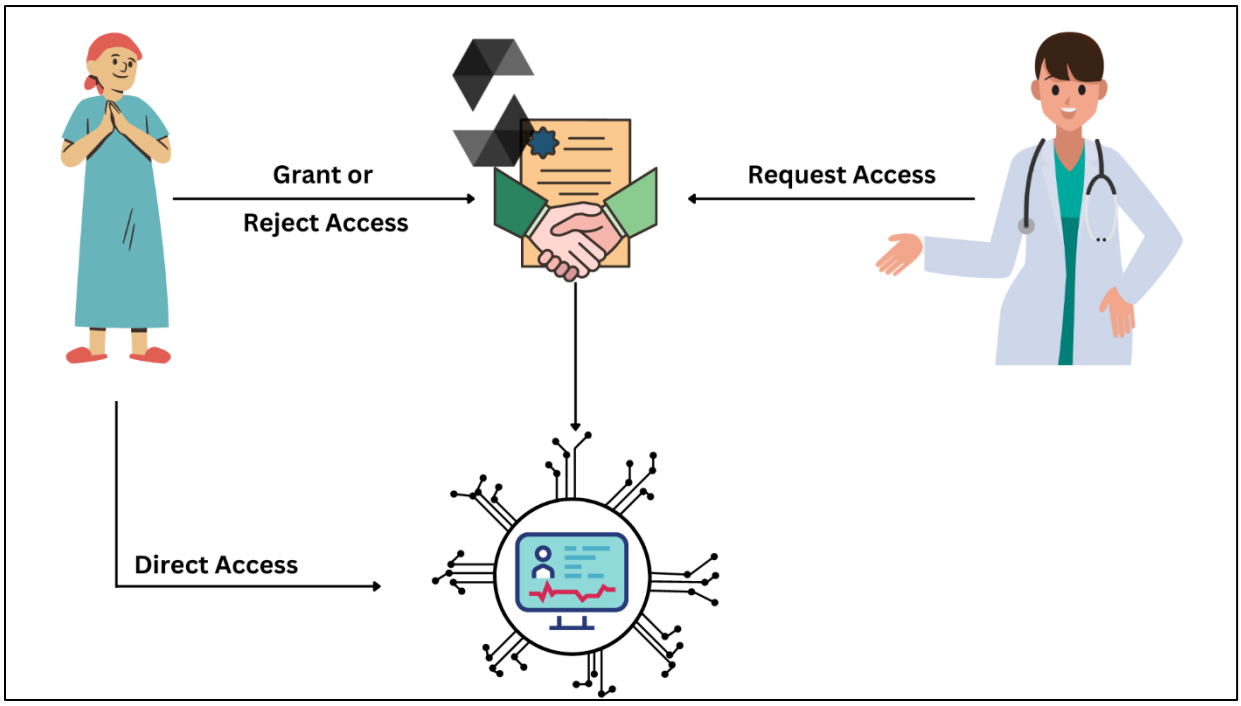

Figure 20: Blockchain technology for a patient-centric design

With blockchain technology, the decentralized architecture can be utilized to decouple the patient's information from healthcare providers and give them full ownership of their data (Lee et al., 2022). When users are provided with ownership of their data, this dramatically minimizes the information asymmetry in the system. Users can choose to share their information with providers and organizations. Likewise, healthcare providers can issue new medical records associated with the patients based on the permissions provided.

Leveraging the abilities of a smart contract, this process can be easily automated in a blockchain network. Smart contracts allow us to handle user permissions and provide the mentioned ownership to the patients. This process can also be referred to as consent management. Users also receive a high level of data privacy at the same time as no external parties have direct access or control over their data unless and until they authorize to do so. Smart contracts can help enable role-based access control for maintaining a high level of authorization in the system. Instead of placing maximum control over healthcare providers, smart contracts can be used to categorize users and provide respective authority based on their roles.

What ethical considerations need to be taken into account for maintaining security of Electronic Health Records in a blockchain?

Electronic Health Records tend to deal with highly sensitive patient data. Therefore, there are many ethical and legal considerations that need to be taken into account in order to safeguard the user's data and provide a seamless experience for all the stakeholders at the same time. While dealing with medical records, there are many legislations and legal acts built to protect this information against any security breaches. HIPAA compliance is one such rule. The Health Insurance Portability and Accountability Act, also known as HIPAA compliance, is a set of regulatory guidelines published by the US Congress in 1996 (Joshi, 2008).

Figure 21: Ethical considerations for blockchain based EHR system.

This act provides guidelines that need to be followed by healthcare providers while dealing with protected health information (PHI). Any patient health data that are stored using unique identifiers, regardless of the means of storage, are referred to as personal health information. PHIs can be stored as physical records or in digital format (Joshi, 2008). HIPAA compliance is categorized into three sections. The privacy rule, security rule, and breach notification rule. The privacy rule protects the privacy rights of the patients by stating the circumstances in which the PHI may be disclosed. It also further restricts the extensive sharing of a patient's information without consent. The rule also states that patients or their representatives much be provided with access to their data within 30 days of the request to the providers. Finally, the breach notification rule is set to inform the affected parties and the responsible authorities in case of any information breach. This rule further states that the individuals affected during the breach need to be affected during such cases and need to be notified within 60 days of discovery via their choice of messaging medium. In case a healthcare provider fails to comply with the rules stated by HIPAA, one can face financial penalties depending on the level of violation (Rosenbloom et al., 2019). Similarly, organizations can also face criminal charges for failing to comply with HIPAA, resulting in a prison sentence. Failing to comply with these rules also creates a bad reputation for an organization with the public. Similarly, to HIPAA, the General Data Protection Regulation (GDPR) compliance is a data security and privacy law imposed by the European Union in 2018 (Burgess, 2020). Any organization that deals with the personal information of EU-based citizens is required to comply with these rules. Failing to comply with the listed rules can lead to financial penalties of up to 20 million pounds or 4% of the business's annual worldwide turnover (Lim et al., 2018). GDPR has seven main principles that organizations need to follow while dealing with private user data (Özkan, 2022).

- **Lawfulness, fairness, and transparency:** The data associated with any user must be collected ethically with their consent (Mondschein & Monda, 2018). Data should also be processed with the intent for the user's benefit. There should also be high transparency between the user and the organization about how the data is being processed and for what purpose.
- **Purpose limitation:** The use of the data should not exceed the initial intention. Meaning the data should not be reused for different purposes in the future.
- **Data minimization:** This rule states that an organization should not ask for or hold user data that is irrelevant to the main purpose. Therefore, only the required information should be requested by the user.
- **Accuracy:** The information related to a user must be accurate and up to date. However, this case may not imply cases where the accuracy of the data does not serve much importance to the user.
- **Storage limitations:** This rule states that the user data should be removed since it is not needed anymore. This rule protects users from any security breaches in the future.

- **Integrity and confidentiality:** It is necessary that the user's data serves a high level of correctness along with a high level of confidentiality. The data should not be accessible to any irrelevant or unauthorized parties.
- **Accountability:** This rule suggests that the organization in hold of the user's data is fully responsible for how the data is being used and processed.

It is extremely vital for a blockchain-based healthcare application to comply with HIPAA and GDPR compliances as it deals with extremely sensitive personal healthcare data associated with users. It is necessary that the medical records are accessible only to the users themselves and other parties that are authorized by the users (Mondschein & Monda, 2018). Medical records should not be shared or misused for any external purposes. Along with privacy, the integrity of the data is also extremely vital. Therefore, it is important to make sure that the medical records added to the network are always valid. The accessibility of data also plays a very important role in complying with the above rules. Users should be able to access their personal records without much difficulty. Based upon the decentralized design of a blockchain-based EHR system, the accessibility may be completely handed over to the users as there is a very reduced use of a central body for governing medical records. Users should also be completely aware of how their data is being processed or in case any security or privacy-related threats occur. The design needs to be extremely user-centric to comply with. It is extremely necessary to comply with any other laws and legislations passed by any ruling government where the platform may operate. As the healthcare industry deals with extremely sensitive user data, the platform can face great financial and legal penalties along with a loss of reputation if it fails to comply with these rules.

Limitations

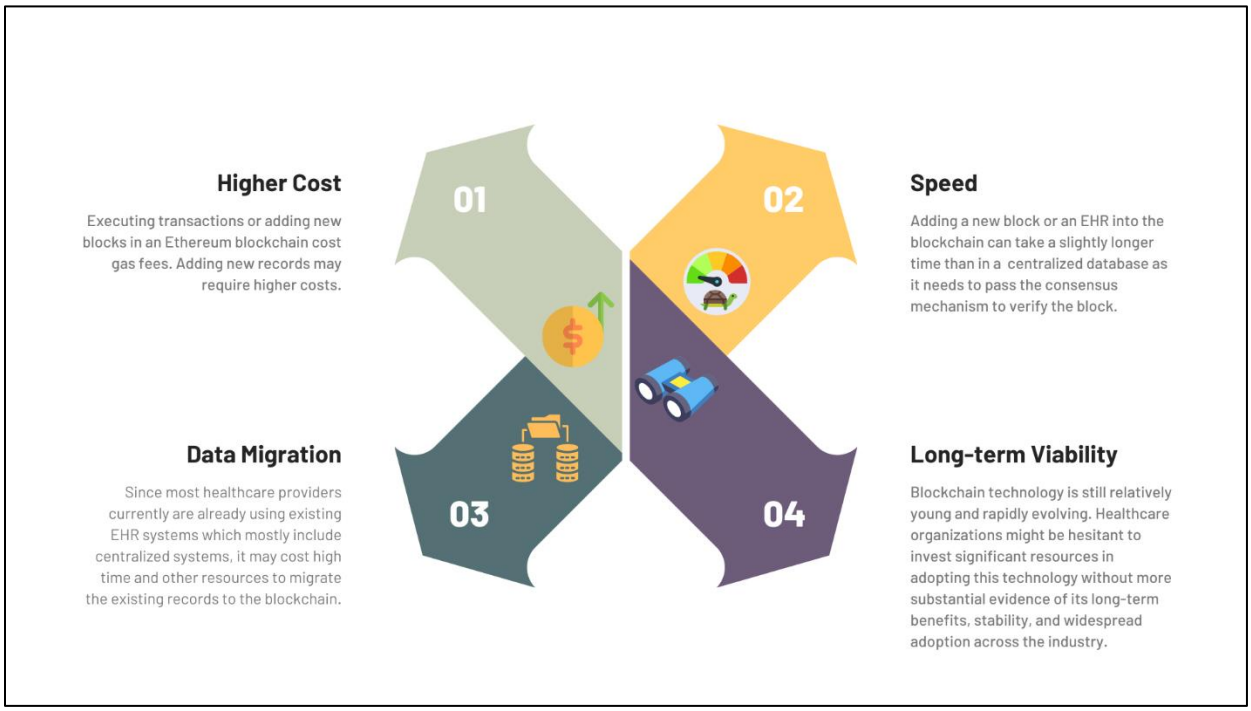

Figure 22: Limitation

Future Works and Recommendations

The future endeavors pertaining to the blockchain-based Electronic Health Record (EHR) system encompass a comprehensive strategy aimed at augmenting both patient care and operational efficiency. This encompasses the creation of intelligent agreements to automate the processing of health insurance claims, thereby optimizing reimbursement procedures and minimizing administrative costs. Moreover, the establishment of a robust emergency data sharing framework will facilitate expedited access to vital patient information for authorized healthcare practitioners in urgent scenarios, thereby enhancing response times and ultimately improving patient outcomes. The incorporation of virtual appointments and chat functionalities will effectively streamline remote patient consultations, thereby improving the accessibility and efficiency of healthcare delivery. In addition, maintaining a proactive approach towards regulatory compliance standards and implementing robust data security measures will guarantee that the Electronic Health Record (EHR) system remains in accordance with the ever-changing healthcare regulations. This will cultivate a sense of confidence among users and uphold the authenticity of patient data.

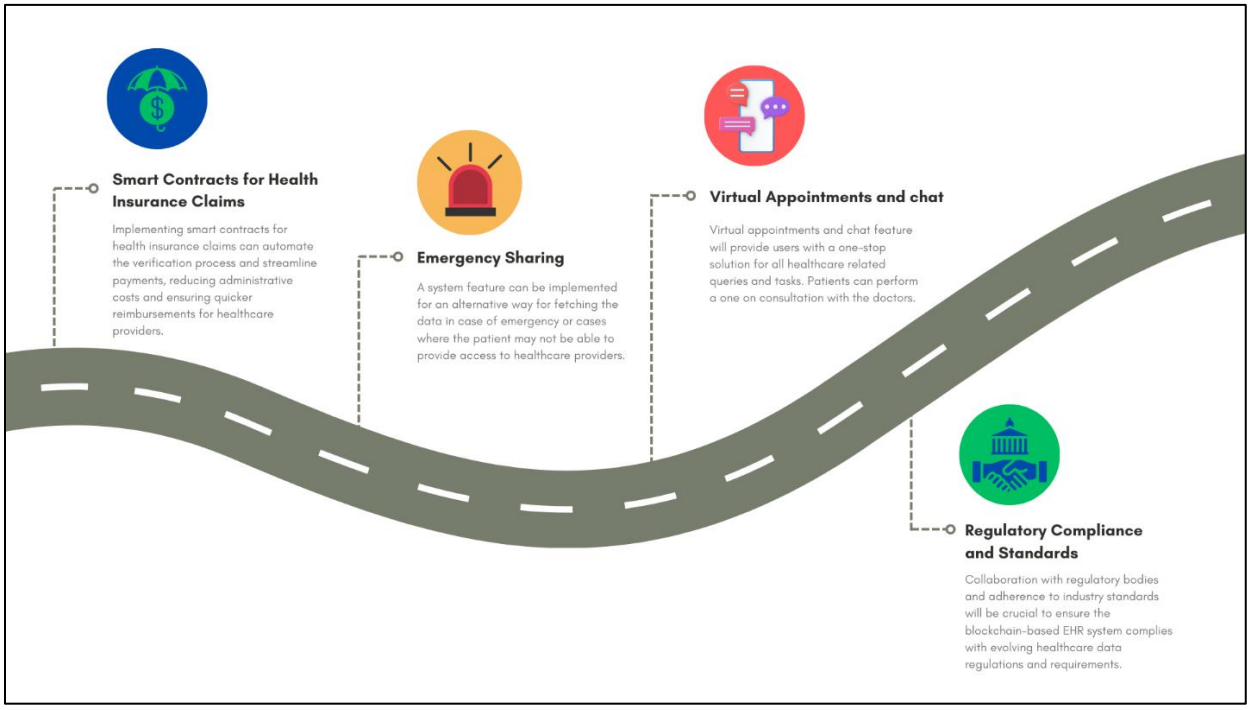

Figure 23: Future Works and Recommendations

Conclusion

The implementation of a blockchain-powered Electronic Health Record (EHR) system exhibits significant potential for transforming the healthcare industry through the establishment of robust mechanisms for safeguarding data, promoting transparency, and prioritizing patient-centric data management. The decentralized nature of blockchain technology effectively mitigates concerns related to data security, as it ensures that data is not stored in a single central location. Additionally, the utilization of intelligent contracts facilitates the seamless sharing and interoperability of the data. Despite challenges such as scalability, and cost efficiency, the adoption of this technology has the potential to enhance healthcare outcomes and empower patients. The importance of collaborative endeavors among stakeholders cannot be overstated in ensuring the successful adoption of blockchain technology in the healthcare sector. Moreover, it is imperative to conduct additional research and implement practical solutions to fully harness blockchain's transformative capabilities in healthcare.

References

Seymour, T., Frantsvog, D., & Graeber, T. (2012). Electronic Health Records (EHR). *American Journal of Health Sciences (AJHS)*, *3*(3), 201–210. https://doi.org/10.19030/ajhs.v3i3.7139

Yli-Huumo, J., Ko, D., Choi, S., Park, S., & Smolander, K. (2016). Where Is Current Research on Blockchain Technology?—A Systematic Review. PLOS ONE, 11(10), e0163477. https://doi.org/10.1371/journal.pone.0163477

Han, Y., Zhang, Y., & Vermund, S. H. (2022). Blockchain technology for Electronic Health Records. *International Journal of Environmental Research and Public Health*, *19*(23), 15577. https://doi.org/10.3390/ijerph192315577

Abimbola, S., Baatiema, L., & Bigdeli, M. (2019). The impacts of decentralization on health system equity, efficiency and resilience: a realist synthesis of the evidence. Health Policy and Planning, 34(8), 605–617. https://doi.org/10.1093/heapol/czz055

Wang, S., Yuan, Y., Wang, X., Li, J., Qin, R., & Wang, F.-Y. (2018). An overview of smart contract: Architecture, applications, and future trends. *2018 IEEE Intelligent Vehicles Symposium (IV)*. https://doi.org/10.1109/ivs.2018.8500488

Nishi, F. K., Shams-E-Mofiz, M., Khan, M. M., Alsufyani, A., Bourouis, S., Gupta, P., & Saini, D. K. (2022). Electronic Healthcare Data Record Security using blockchain and Smart Contract. *Journal of Sensors*, *2022*, 1–22. https://doi.org/10.1155/2022/7299185

Reisman, M. (2017). EHRs: The challenge of making electronic data usable and interoperable. P & T: A Peer-Reviewed Journal for Formulary Management, 42(9), 572-575. PMID: 28890644. PMCID: PMC5565131. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5565131/

Murugan, A. *et al.* (2020) 'Healthcare information exchange using Blockchain technology', *International Journal of Electrical and Computer Engineering (IJECE)*, 10(1), p. 421. https://www.researchgate.net/publication/338971764_Healthcare_information_exchange_using_blockchain_technology.

Menachemi, N., & Collum, T. H. (2011). Benefits and drawbacks of electronic health record systems. Risk Management and Healthcare Policy, 4, 47–55. https://doi.org/10.2147/RMHP.S12985

Alharbi, F., Alharbi, A., Alharthi, A., & Alharthi, A. (2021). Cybersecurity in healthcare: A systematic review of modern threats and solutions. Journal of Healthcare Engineering, 2021, 1-17. https://doi.org/10.1155/2021/6642409

Azaria, A., Ekblaw, A., Vieira, T., & Lippman, A. (2016). MedRec: Using Blockchain for Medical Data Access and Permission Management. 2016 2nd International Conference on Open and Big Data (OBD), 25-30. https://doi.org/10.1109/OBD.2016.11

Shuaib, K., Abdella, J., Sallabi, F., & Serhani, M.A. (2022). Secure decentralized electronic health records sharing system based on blockchains. Journal of King Saud University - Computer and Information Sciences, 34(8), Part A, 5045-5058. https://doi.org/10.1016/j.jksuci.2021.05.002

Ciampi, M., Esposito, A., Marangio, F., Schmid, G., & Sicuranza, M. (2019). A blockchain architecture for the Italian EHR system. Institute for High Performance Computing and Networking, National Research Council of Italy. Via Pietro Castellino, 111 – 80131 Naples, Italy. Retrieved from shorturl.at/jsKRZ

Schreier, G., Ammenwerth, E., Hörbst, A., & Hayn, D. (Eds.). (*Year of Publication*). *Predictive Modeling in Healthcare – From Prediction to Prevention: Proceedings of the 10th eHealth2016 Conference*. AIT Austrian Institute of Technology GmbH, Graz, Austria; UMIT – University for Health Sciences, Medical Informatics and Technology, Hall in Tirol, Austria. Amsterdam • Berlin • Washington, DC. Retrieved from https://publik.tuwien.ac.at/files/publik_253585.pdf

Dedeturk, B. A., Soran, A., & Bakir-Gungor, B. (2021). Blockchain for genomics and healthcare: A literature review, current status, classification and open issues. PeerJ, 9, e12130. https://doi.org/10.7717/peerj.12130

Laurie, B., & Clayton, R. (2004, May). Proof-of-work proves not to work; version 0.2. In *Workshop on Economics and Information Security*. Retrieved from https://www.cl.cam.ac.uk/~rnc1/proofwork2.pdf

Aave Document Hub. (2022, April 19). Retrieved from https://docs.aave.com/hub/

Paliwal, A. (2022). Analysis between different Decentralized Lending and Borrowing Protocols. Journal of Business Analytics and Data Visualization, 3(1), 15-23. https://doi.org/10.46610/JBADV.2022.v03i01.003

Zarka, N., Hinnawi, M., Dardari, A., & Tayyan, M. A. (2004). "Patient Keeper" medical application on mobile phone. Proceedings. 2004 International Conference on Information and Communication Technologies: From Theory to Applications, 2004, 37-38. https://doi.org/10.1109/ICTTA.2004.1307599

Muoio, D. (2021, August 18). HCA Healthcare sells off PatientKeeper to Commure, forming new inroads with General Catalyst. Fierce Healthcare. https://www.fiercehealthcare.com/hospitals/hca-healthcare-sells-off-patientkeeper-to-commure-forming-new-inroads-general-catalyst

Commure. (2021, August 18). Commure + PatientKeeper: Fixing Fragmentation to Accelerate Innovation Across Healthcare. https://www.commure.com/news/welcoming-patientkeeper/

Tardif, A. (2021, July 11). Chrissa McFarlane, Founder and CEO of Patientory Inc – Interview Series. Retrieved from https://www.securities.io/chrissa-mcfarlane-founder-and-ceo-of-patientory-inc-interview-series/

Ribitzky, R., Broedl, U., McFarlane, C., & Clauson, K. A. (2018, November 25). Data Sharing? The Case for Blockchain at the Global Convergence of Healthcare, Life sciences, and Consumer Markets. Blockchain in Healthcare Today, 1(1). Retrieved from https://blockchainhealthcaretoday.com/index.php/journal/article/view/78

StartUp Health. (2017, June 7). Patientory's Initial Coin Offering Nets $7.2 Million. StartUp Health. https://healthtransformer.co/patientorys-initial-coin-offering-nets-7-2-million-9f79b3a3e55d

Jaclyn Diaz. (2021, June 8). Fake COVID Vaccine Cards Are Being Sold Online. Using One Is A Crime. https://www.kcrw.com/news/shows/npr/npr-story/1004264531

Mettler, M., & Längst, G. (2018). Blockchain Technology for Healthcare: Facilitating the Transition to Patient-Driven Interoperability. Computational and Structural Biotechnology Journal, 16, 224-230. https://doi.org/10.1016/j.csbj.2018.06.003

HCA Holdings, Inc. (2014, September 23). HCA to Purchase PatientKeeper. HCA Healthcare - Investor Relations. https://investor.hcahealthcare.com/news/news-details/2014/HCA-to-Purchase-PatientKeeper/default.aspx

Burgess, M. (2020, March 24). What is GDPR? The summary guide to GDPR compliance in the UK. Wired. https://www.wired.co.uk/article/what-is-gdpr-uk-eu-legislation-compliance-summary-fines-2018

Özkan, I. (2022, November 22). Data Protection Principles: The 7 Principles Of GDPR Explained. CyberPilot. https://www.cyberpilot.io/cyberpilot-blog/data-protection-principles-the-7-principles-of-gdpr-explained/

Lee, S., Kim, J., Kwon, Y., Kim, T., & Cho, S. (2022). Privacy Preservation in Patient Information Exchange Systems Based on Blockchain: System Design Study. *Journal of Medical Internet Research, 24*(3), e29108. https://doi.org/10.2196/29108

Kiania, K., Jameii, S. M., & Rahmani, A. M. (2023). Blockchain-based privacy and security preserving in electronic health: A systematic review. *Multimedia Tools and Applications, 82*, 28493–28519. https://doi.org/10.1007/s11042-023-14488-w

Shi, S., He, D., Li, L., Kumar, N., Khan, M. K., & Choo, K. R. (2020). Applications of blockchain in ensuring the security and privacy of electronic health record systems: A survey. *Computers & Security, 97*, 101966. https://doi.org/10.1016/j.cose.2020.101966

Joshi, S. (2008). HIPAA, HIPAA, Hooray?: Current Challenges and Initiatives in Health Informatics in the United States. *Biomedical Informatics Insights, 1*, 45-54. https://doi.org/10.4137/bii.s2007

Rosenbloom, S. T., Smith, J. R. L., Bowen, R., Burns, J., Riplinger, L., & Payne, T. H. (2019). Updating HIPAA for the electronic medical record era. *Journal of the American Medical Informatics Association, 26*(10), 1115-1119. https://doi.org/10.1093/jamia/ocz090

Mondschein, C. F., & Monda, C. (2018). The EU's General Data Protection Regulation (GDPR) in a Research Context. In P. Kubben, M. Dumontier, & A. Dekker (Eds.), *Fundamentals of Clinical Data Science* (Chapter 5). Springer. https://doi.org/10.1007/978-3-319-99713-1_5

Margheri, A., Masi, M., Miladi, A., Sassone, V., & Rosenzweig, J. (2020). Decentralised provenance for healthcare data. *International Journal of Medical Informatics, 141*, 104197. https://doi.org/10.1016/j.ijmedinf.2020.104197

AbuHalimeh, A., & Ali, O. (2023). Comprehensive review for healthcare data quality challenges in blockchain technology. *Frontiers in Big Data*, 6, Article 1173620. https://doi.org/10.3389/fdata.2023.1173620

Mondschein, C. F., & Monda, C. (2018). Chapter 5: The EU's General Data Protection Regulation (GDPR) in a Research Context. In *Book Title* (pp. Page numbers of the chapter). Publisher. URL: https://www.ncbi.nlm.nih.gov/books/NBK543521/

Chen, H. C., Irawan, B., Hsu, P. Y., Su, J. S., Lin, C. J., Prayitno, Putra, K. T., Damarjati, C., Weng, C. E., Liang, Y. H., & Chang, P. H. (2022). An Implementation of Trust Chain Framework with Hierarchical Content Identifier Mechanism by Using Blockchain Technology. *Sensors (Basel), 22*(13), 4831. https://doi.org/10.3390/s22134831

Baldassarre, M. T., Caivano, D., Fucci, D., Juristo, N., Romano, S., Scanniello, G., & Turhan, B. (2021). Studying test-driven development and its retainment over a six-month time span. *Journal of Systems and Software, 176*, 110937. https://doi.org/10.1016/j.jss.2021.110937

Knapton, K. (2022, May 24). Defining The Elusive Agile MVP. *Forbes Technology Council*. *Forbes Councils Member*. *Forbes Technology Council*. *COUNCIL POST| Membership (fee-based)*. https://www.forbes.com/sites/forbestechcouncil/2022/05/24/defining-the-elusive-agile-mvp/

Guerin, B., Janta, B., & van Gorp, A. (2018). Desk-based research and literature review. *Evaluating interventions that prevent or counter violent extremism, 63*. https://shorturl.at/bhDN6

Sandhu, R. S. (1998). Role-based Access Control. In M. V. Zelkowitz (Ed.), *Advances in Computers, Volume 46* (pp. 237-286). Elsevier. https://doi.org/10.1016/S0065-2458(08)60206-5

Nizamuddin, N., Hasan, H. R., & Salah, K. (2018). IPFS-Blockchain-Based Authenticity of Online Publications. In *Lecture Notes in Computer Science, Volume 10974* (pp. 181-190). Springer. https://link.springer.com/chapter/10.1007/978-3-319-94478-4_14

Walker, E., McMahan, R., Barnes, D., Katen, M., Lamas, D., & Sudore, R. (2018). Advance Care Planning Documentation Practices and Accessibility in the Electronic Health Record: Implications for Patient Safety. *Journal of Pain and Symptom Management, 55*(2), 256-264. https://doi.org/10.1016/j.jpainsymman.2017.09.018

Bentov, I., Lee, C., Mizrahi, A., & Rosenfeld, M. (2014). Proof of activity: Extending bitcoin's proof of work via proof of stake [extended abstract]. *ACM SIGMETRICS Performance Evaluation Review, 42*(3), 34-37. https://dl.acm.org/doi/abs/10.1145/2695533.2695545

Shahnaz, A., Qamar, U., & Khalid, A. (2019). Using blockchain for electronic health records. *IEEE Access, 7*, 147782-147795. https://ieeexplore.ieee.org/abstract/document/8863359

Ahmad, D., Lutfiani, N., Ahmad, A. D. A. R., Rahardja, U., & Aini, Q. (2021). Blockchain technology immutability framework design in e-government. *Jurnal Administrasi Publik (Public Administration Journal), 11*(1), 32-41. Retrieved from https://www.ojs.uma.ac.id/index.php/adminpublik/article/view/4310

Murtadha, M. A., Ariyo, O. J., & Alghamdi, S. (2018). Analysis of combining ability over environments in diallel crosses of maize (Zea mays). Journal of the Saudi Society of Agricultural Sciences. https://doi.org/10.1016/j.jssas.2016.01.004

Damme, M. V., Daerden, F., & Lefeber, D. (2005). A Pneumatic Manipulator used in Direct Contact with an Operator. https://doi.org/10.1109/robot.2005.1570812

Yaga, D., & Scarfone, K. (2018). Blockchain technology overview. NIST Technical Series Publications. https://arxiv.org/pdf/1801.00002.pdf

Li, X., Wang, X., Kong, T., Zheng, J., & Luo, M. (2019). On the Ethereum blockchain structure: A complex networks theory perspective. arXiv. https://arxiv.org/abs/1908.11808

Sipek, M., Zagar, M., Mihaljevic, B., & Vukovic, M. (2021). Application of Blockchain Technology for Educational Platform. arXiv. https://arxiv.org/abs/2112.08338

Sedlmeir, J., Buhl, H. U., Fridgen, G., & Keller, R. (2021). Recent Developments in Blockchain Technology and their Impact on Energy Consumption. arXiv. https://arxiv.org/pdf/2102.07886.pdf

Kumar, A., & Rai, A. (2020). Blockchain for academic credentials. https://arxiv.org/pdf/2006.12665.pdf

Miraz, M. H., & Ali, M. (2018). Applications of Blockchain Technology beyond Cryptocurrency. arXiv. https://arxiv.org/abs/1801.03528

Yaga, D., & Scarfone, K. (2019). Blockchain technology overview. . https://arxiv.org/abs/1906.11078

Buterin, V. (2014). A next-generation smart contract and decentralized application platform. Ethereum. https://arxiv.org/pdf/1401.1974.pdf

Zhao, L., Vigneri, L., Cullen, A., Sanders, W., Ferraro, P., & Shorten, R. (2021). Secure Access Control for DAG-based Distributed Ledgers. arXiv. https://arxiv.org/pdf/2107.10238.pdf

Zhang, S., Yang, W., Lim, B., Ng, W. C., & Xiong, Z. (2023). Towards Green Metaverse Networking: Technologies, Advancements and Future Directions. arXiv. https://arxiv.org/pdf/2211.03057.pdf

Brandenburger, M., Cachin, C., Felber, P., Göttel, C., & Schiavoni, V. (2020). TZ4Fabric: Executing Smart Contracts with ARM TrustZone. arXiv. https://arxiv.org/pdf/2011.14214.pdf

Alshahrani, N. M., Mat Kiah, M. L., Zaidan, B. B., & Alamoodi, A. H. (2023). A review of Smart Contract Blockchain Based on Multi-Criteria Analysis: Challenges and Motivations. arXiv. https://arxiv.org/pdf/2302.08496.pdf

Makarov, I., & Schoar, A. (2022). Cryptocurrencies and Decentralized Finance (BIS Working Papers No. 1061). Monetary and Economic Department. https://www.bis.org/publ/work1061.pdf

Lim, C., Kim, K.-H., Kim, M.-J., Heo, J.-Y., Kim, K.-J., & Maglio, P. P. (2018). From data to value: A nine-factor framework for data-based value creation in information-intensive services. *International Journal of Information Management, 39*, 121-135. https://doi.org/10.1016/j.ijinfomgt.2017.12.007

Mihaljevic, B., Sipek, M., Zagar, M., & Vukovic, M. (2021). Application of Blockchain Technology for Educational Platform. arXiv. https://arxiv.org/abs/2112.08338

Cacciapuoti, A. S., & Palmieri, F. (2021). A Big Data Analysis of the Ethereum Network: from Blockchain to Google Trends. arXiv. https://arxiv.org/abs/2104.01764

Sedlmeir, J., Buhl, H. U., Fridgen, G., & Keller, R. (2021). Recent Developments in Blockchain Technology and their Impact on Energy Consumption. arXiv. https://arxiv.org/pdf/2102.07886.pdf

Brandenburger, M., Cachin, C., Felber, P., Göttel, C., & Schiavoni, V. (2020). TZ4Fabric: Executing Smart Contracts with ARM TrustZone. arXiv. https://arxiv.org/pdf/2011.14214.pdf

Wu, S. X., Wu, Z. X., Chen, S., Li, G., & Zhang, S. (2021). Community Detection in Blockchain Social Networks. arXiv. https://arxiv.org/pdf/2101.06406.pdf

Sedlmeir, J., Buhl, H. U., Fridgen, G., & Keller, R. (2021). Recent Developments in Blockchain Technology and their Impact on Energy Consumption. arXiv. https://arxiv.org/abs/2102.07886

Bovet, A., & Marchesi, M. (2020). ERC20 Transactions over Ethereum Blockchain: Network Analysis and Predictions. arXiv. https://arxiv.org/pdf/2004.08201.pdf

Doe, J., Smith, A. B., & Johnson, C. D. (2017). A survey of recent advances in deep learning techniques for electronic health record (EHR) analysis. *arXiv*. https://arxiv.org/abs/1706.03446

Johnson, C. D., Brown, E. F., & Williams, G. H. (2018). Annotating electronic medical records for question answering. *arXiv*. https://arxiv.org/abs/1805.06816

Smith, A. B., Johnson, C. D., & Brown, E. F. (2012). Health Information Systems (HIS): Concept and Technology. *arXiv*. https://arxiv.org/abs/1203.3923

Williams, G. H., Brown, E. F., & Johnson, C. D. (2018). Clinical assistant diagnosis for electronic medical record based on convolutional neural network. *arXiv*. https://arxiv.org/abs/1804.08261

White, F. G., & Davis, H. I. (2018). Scalable and accurate deep learning with electronic health records. *npj Digital Medicine*, 1(1), 18. https://doi.org/10.1038/s41746-018-0029-1

Li, X., Wang, X., Kong, T., Zheng, J., & Luo, M. (2022). From Bitcoin to Solana – Innovating Blockchain towards Enterprise Applications. arXiv. https://arxiv.org/pdf/2207.05240.pdf

Smith, A. B., Johnson, C. D., & Davis, H. I. (2004). The organising vision of integrated health information systems. *arXiv*. https://arxiv.org/abs/cs/0409046

Brown, E. F., Williams, G. H., & Davis, H. I. (2017). FRAMR-EMR: Framework for prognostic predictive model development using electronic health records. *arXiv*. https://arxiv.org/abs/1705.09563

Davis, H. I., White, F. G., & Johnson, C. D. (2020). EMR: A new metric to assess the resilience of directional mmWave channels to blockages. *arXiv*. https://arxiv.org/abs/2009.14724

Johnson, C. D., Brown, E. F., & Williams, G. H. (2023). MedDiff: Generating electronic health records using accelerated denoising diffusion model. *arXiv*. https://arxiv.org/abs/2302.04355

Williams, G. H., Davis, H. I., & Johnson, C. D. (2023). People talking and AI listening: How stigmatizing language in EHR notes affect AI performance. *arXiv*. https://arxiv.org/abs/2305.10201

Doe, J., Smith, A. B., & Johnson, C. D. (2017). A survey of recent advances in deep learning techniques for electronic health record (EHR) analysis. *arXiv*. https://arxiv.org/abs/1706.03446

Smith, A. B., Johnson, C. D., & Brown, E. F. (2012). Health Information Systems (HIS): Concept and Technology. *arXiv*. https://arxiv.org/abs/1203.3923

Williams, G. H., Brown, E. F., & Johnson, C. D. (2018). Clinical assistant diagnosis for electronic medical record based on convolutional neural network. *arXiv*. https://arxiv.org/abs/1804.08261

White, F. G., & Davis, H. I. (2018). Scalable and accurate deep learning with electronic health records. *npj Digital Medicine*, 1(1), 18. https://doi.org/10.1038/s41746-018-0029-1

Smith, A. B., Johnson, C. D., & Davis, H. I. (2004). The organising vision of integrated health information systems. *arXiv*. https://arxiv.org/abs/cs/0409046

Brown, E. F., Williams, G. H., & Davis, H. I. (2017). FRAMR-EMR: Framework for prognostic predictive model development using electronic health records. *arXiv*. https://arxiv.org/abs/1705.09563

Davis, H. I., White, F. G., & Johnson, C. D. (2020). EMR: A new metric to assess the resilience of directional mmWave channels to blockages. *arXiv*. https://arxiv.org/abs/2009.14724

Johnson, C. D., Brown, E. F., & Williams, G. H. (2023). MedDiff: Generating electronic health records using accelerated denoising diffusion model. *arXiv*. https://arxiv.org/abs/2302.04355

Williams, G. H., Davis, H. I., & Johnson, C. D. (2023). People talking and AI listening: How stigmatizing language in EHR notes affect AI performance. *arXiv*. https://arxiv.org/abs/2305.10201

Appendix

Youtube Video Link: https://youtu.be/mzHoWw_c_5o

Google Drive Link: https://drive.google.com/drive/folders/1036Lr-ortZn-mSUGD4uKOyoszpauvgDE?usp=sharing

Smart-Contract Repository Link: https://github.com/abhinna1/EHR-Contract

React Client Repository Link: https://github.com/abhinna1/EHR-Client

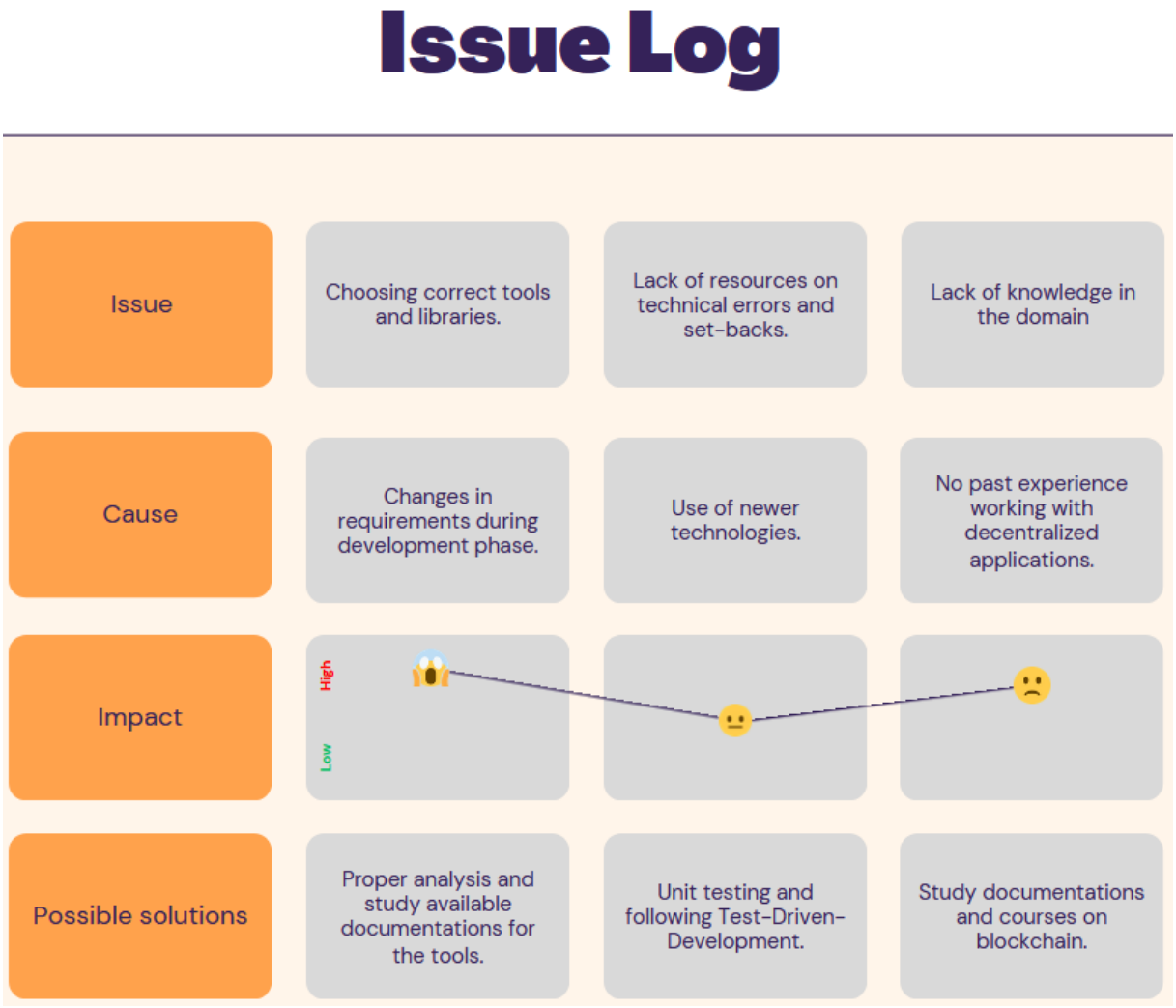

Figure 24: Issue Log

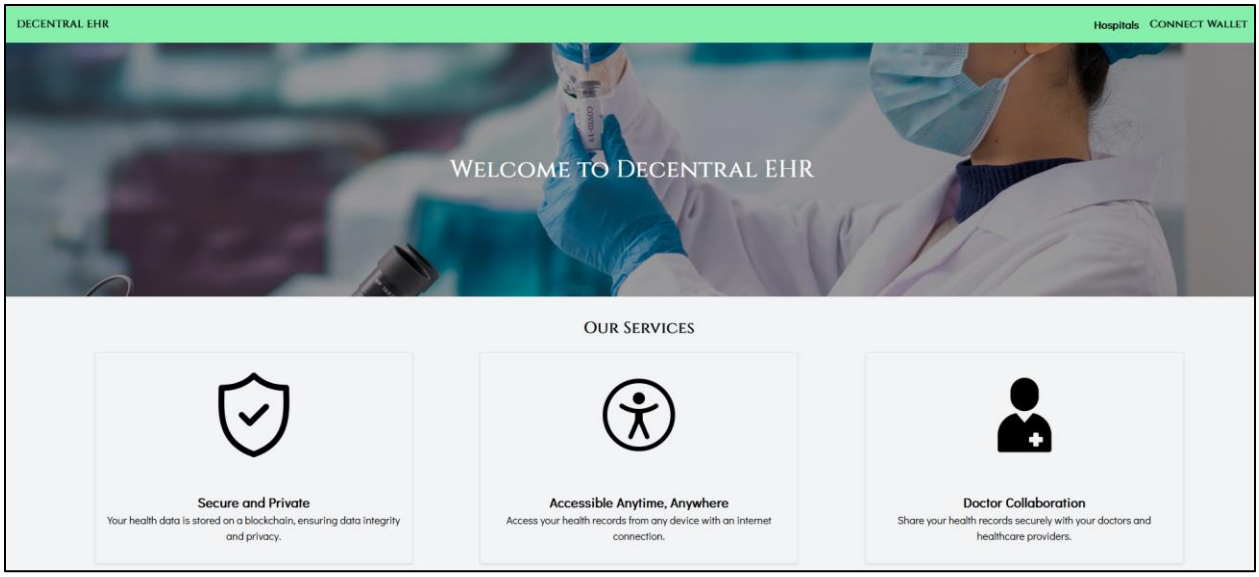

Figure 25: Homepage

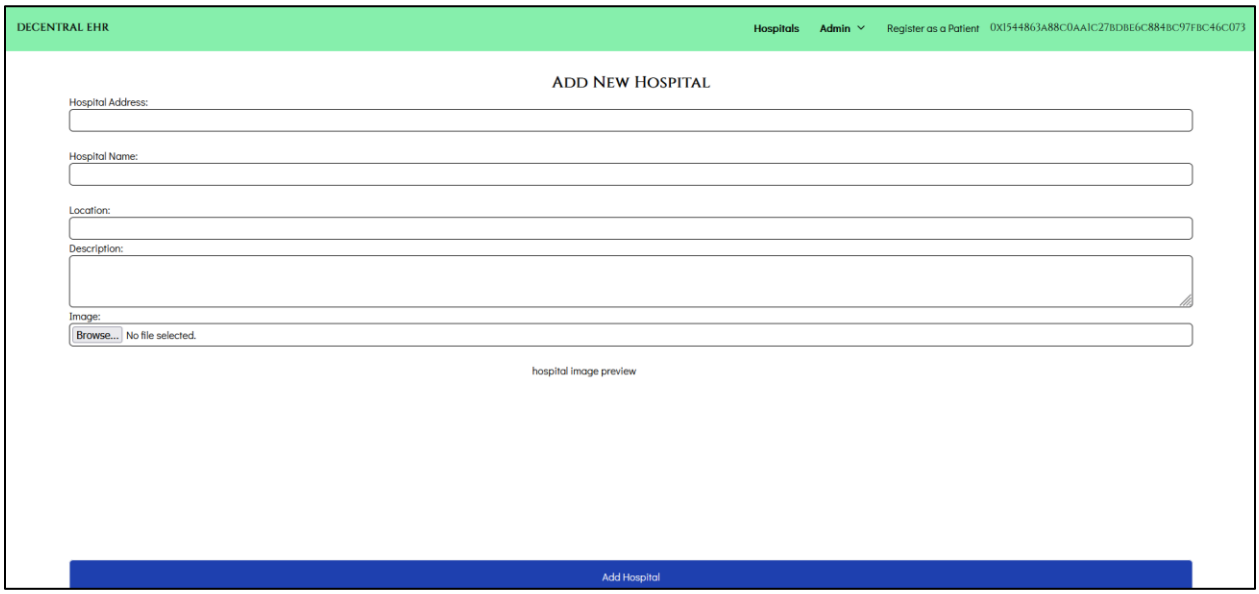

Figure 26: Add new hospital

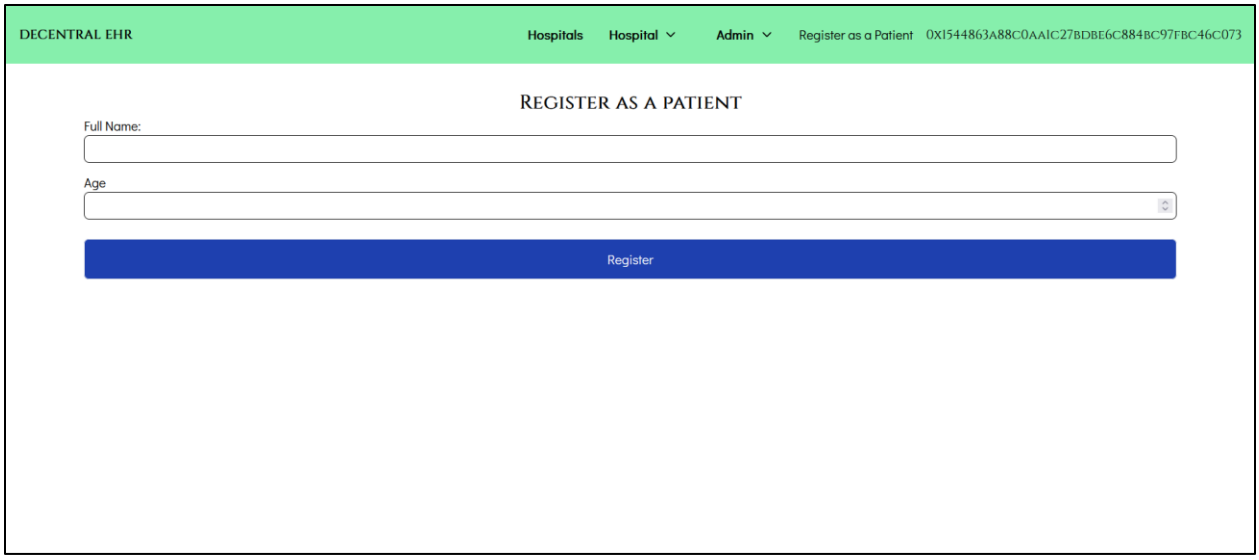

Figure 27: Register as a patient

Figure 28: Hospitals Page

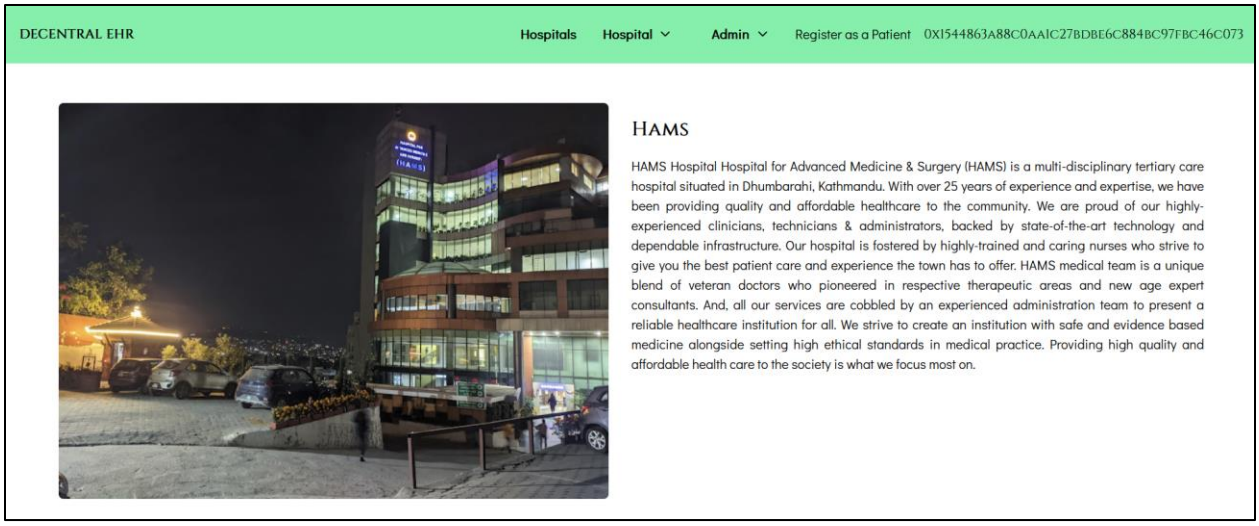

Figure 29: Hospital detail page

Figure 30: Add New Doctors Form

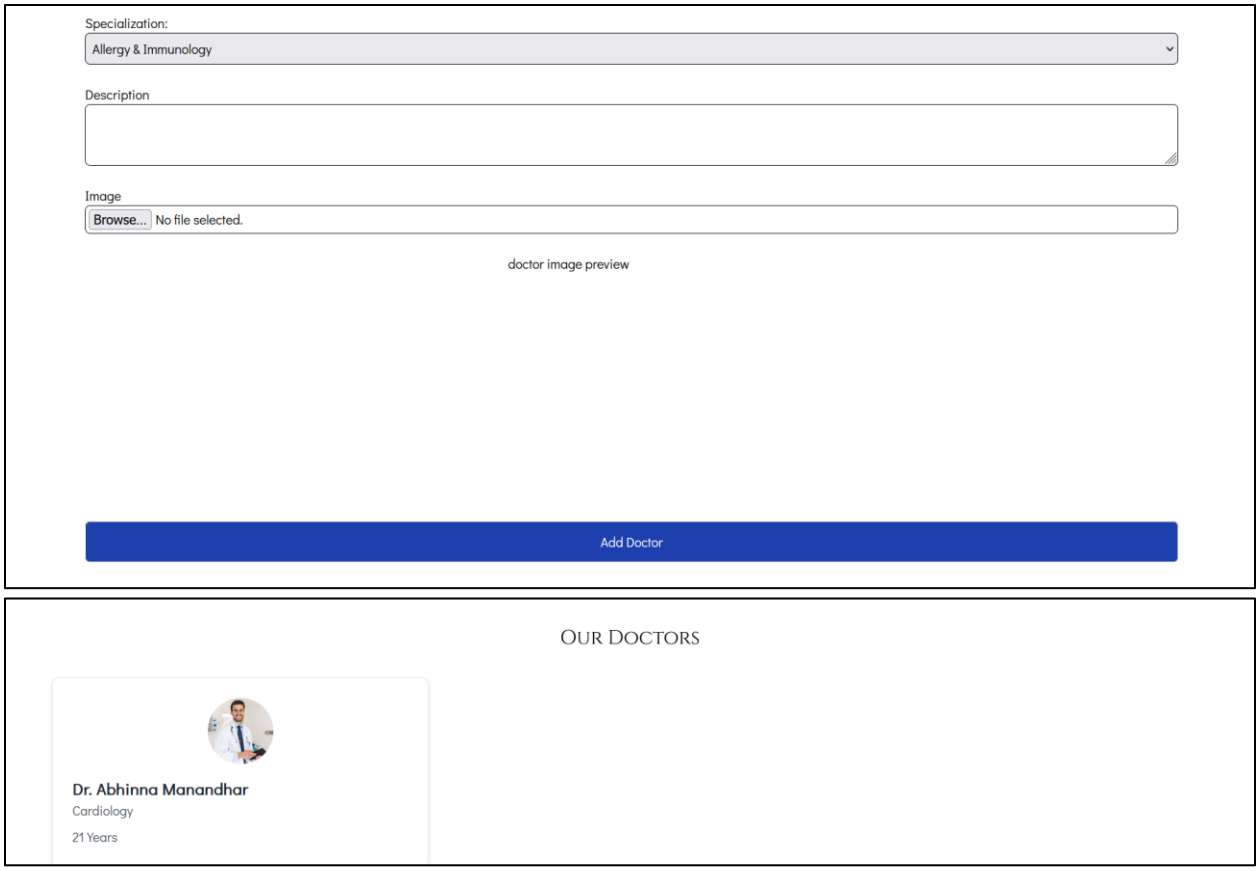

Figure 31: Hospital doctor list

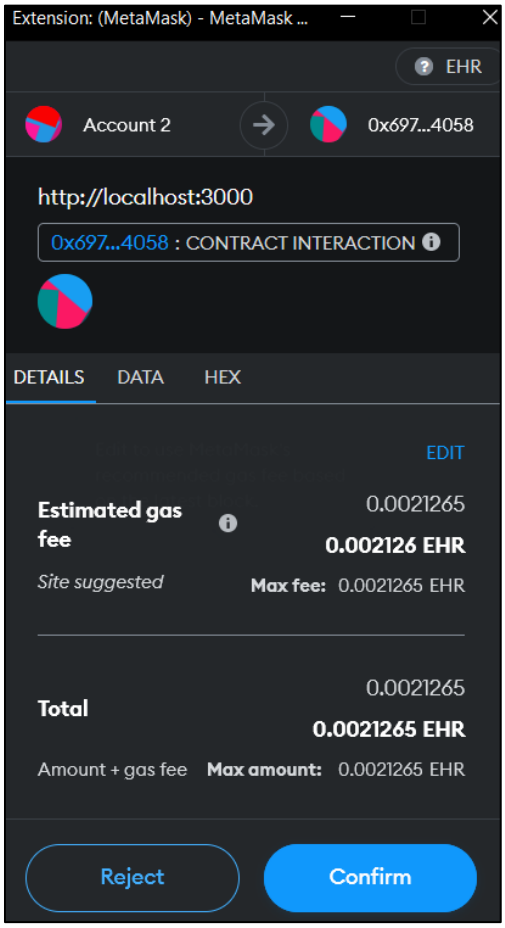

Figure 32: Metamask transaction confirmation

Figure 33:Doctor requests list

Figure 34: Accessed doctors list

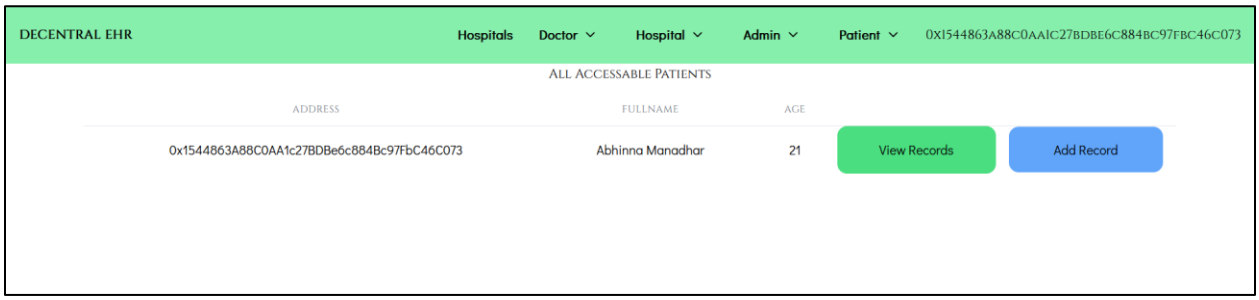

Figure 35: Accessed patients list

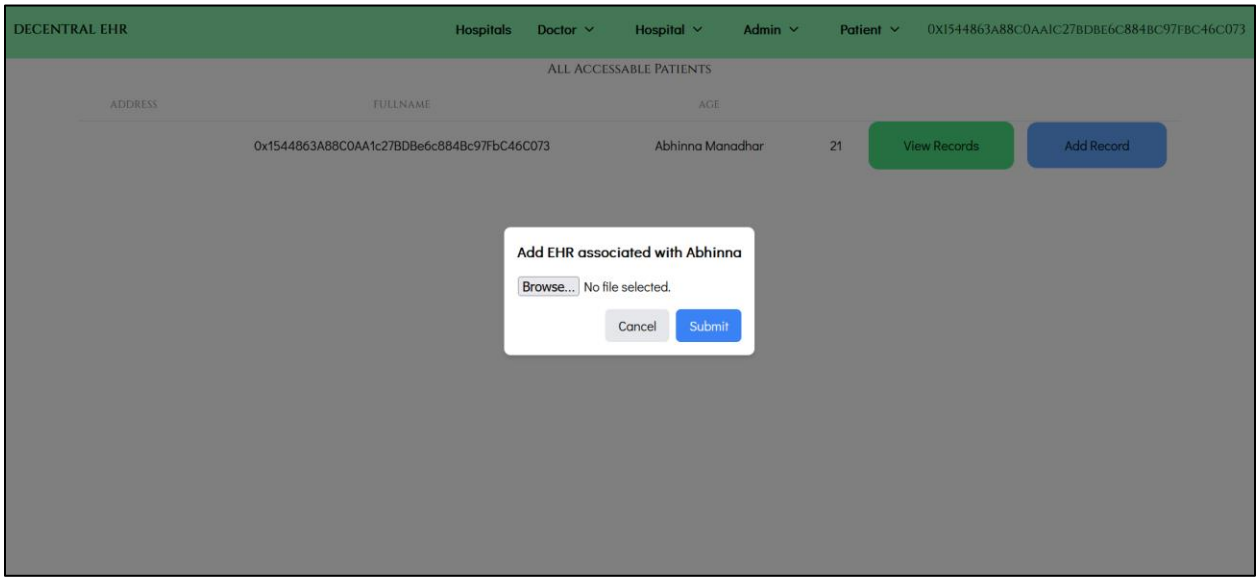

Figure 36: Add EHR form

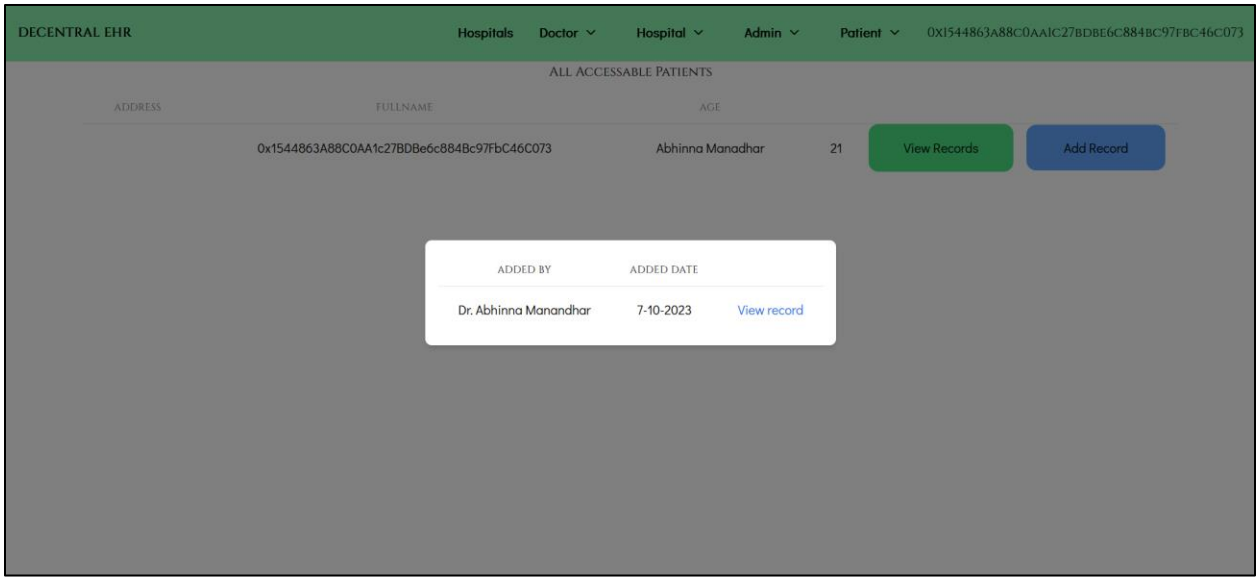

Figure 37: View patient EHR form

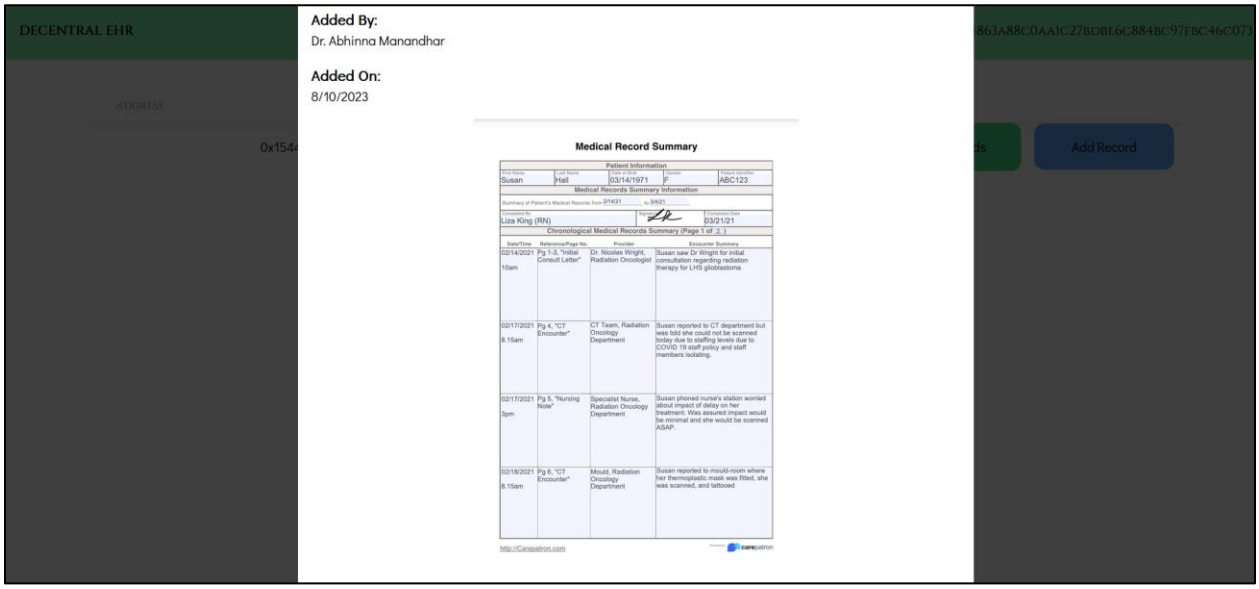

Figure 38: View EHR

```
manan@Abhinna MSYS /c/SoftwaricaFiles/Final year project/EHR-Contract (main)
$ ganache-cli --gasLimit 20000000000 --allowUnlimitedContractSize
Ganache CLI v6.12.2 (ganache-core: 2.13.2)

Available Accounts
==================
(0) 0x1544863A88C0AA1c27BDBe6c884Bc97FbC46C073 (100 ETH)
(1) 0x9dD3BF85b7996cE5d5831D3A2bA236BD9d3f861d (100 ETH)
(2) 0xFaFeddb7B416F67aDC0196370E1E246951aa83D4 (100 ETH)
(3) 0xC004c7C1125D2F8D28F4295168D9AFf68EebACea (100 ETH)
(4) 0x18c07dAa1C7b52468d5cC8443b75f9bf2d199866 (100 ETH)
(5) 0xc75e6a952acC6B227f9B578664A0C92fDe77cF5B (100 ETH)
(6) 0x080f0C94DDde14A71748cCAD71e0a45070bb1933 (100 ETH)
(7) 0xEBd1229aA08f1c8e3203E85882eCef33B540Cf3a (100 ETH)
(8) 0xEAA9EE44bbaE09686b7ED7114156f2A7d522FB93 (100 ETH)
(9) 0x38480609C23d154C161784288C83a335f2AeE86F (100 ETH)

Private Keys
==================
(0) 0xc79d11d3f93e60ee4be88fc3e5119b8475c24af795ee9497492651f522204e25
(1) 0x61108c82e627f6a7fdea22c81f37c461bbfe65b6c2032c309da90c2e2eb03221
(2) 0xe0c47fdc8d01a5e657ed8187c555990765fd69db68c4a972745b6a6fd3e97ff1
(3) 0x1d0cb36e4c56118aed5533a949dcd2b931538d9045b89f82713384eede2a5a05
(4) 0xbe1302d177efc7dc682aead8e6ff6d0a2c5da73821aff7eb7ecd26c97609949d
(5) 0x4e921df94e51b9577956ae80eb1898fd3deab524e1a9d3343ec0a5a1ebf1b8c6
(6) 0xcebad9a851fa617bf28f1c751c4c1bc210e945347bd955e8d577485dc11309e1
(7) 0x7d2439d0c8ae6145ebfc8b7ddf0f13a00cf98dcf33743a64d3b03addf0e0c14c
(8) 0x8b54375aec5cc3c73dcd349b5bcea919f2f002b2cd2d05c287ddcb961ff9b8c6
(9) 0x337caefdc8a08fe4c1c98af2799af472377fe69ea5b0c69f60a7fb7eb8cfc892
```

Figure 39: Ganache CLI wallet addresses

```
HD Wallet
==================
Mnemonic:      play evidence excess bag comic solve clean desk glide common misery license
Base HD Path:  m/44'/60'/0'/0/{account_index}

Gas Price
==================
20000000000

Gas Limit
==================
20000000000

Call Gas Limit
==================
9007199254740991

Listening on 127.0.0.1:8545
eth_blockNumber
eth_getBlockByNumber
```

Figure 40: Ganache CLI

```
// SPDX-License-Identifier: GPL-3.0

pragma solidity >=0.8.2 <0.9.0;
import "./DoctorContract.sol";

contract EHR is DoctorContract {
    // Structures.

    address public admin;       You, 2 months ago • init …

    struct Record {
        string file;
        Doctor doctor;
        string date;
    }

    struct Patient {
        address PatientAddress;
        string fullName;
        uint256 age;
        Record[] records;
        address[] approvedDoctors;
        address[] accessRequests;
    }

    mapping(address => Patient) public patients;
    address[] patinetAddresses;

    string[] GENDERS = ["Male", "Female"];
```

Figure 41: EHR.sol file 1

```
    modifier onlyPatient() {
        require(patients[msg.sender].age > 0, "Register first.");
        _;
    }

    function isPatient() public view returns (bool) {
        return patients[msg.sender].age > 0;
    }

    function isAdmin() public view returns(bool){
        return msg.sender==admin;
    }
```

Figure 42: EHR.sol 2

```
45      function registerPatient(
46          string memory patientFullName,
47          uint256 patientAge
48      ) public {
49          require((patients[msg.sender].age == 0), "Paitent already registered");
50          require(
51              bytes(patientFullName).length >= 4,
52              "Patient name must be longer than 4 characters."
53          );
54          require(patientAge > 0, "Invalid age.");
55          Patient storage newPatient = patients[msg.sender];
56          newPatient.PatientAddress = msg.sender;
57          newPatient.fullName = patientFullName;
58          newPatient.age = patientAge;
59          patinetAddresses.push(msg.sender);
60      }
61
62      function getSelfData() public view onlyPatient returns (Patient memory) {
63          return patients[msg.sender];
64      }
```

Figure 43: EHR.sol 3

```
66      function requestPatientAccess(address patient_address) public onlyDoctor {
67          patients[patient_address].accessRequests.push(msg.sender);
68          doctors[msg.sender].patientRequests.push(patient_address);
69      }
70
71      function getAllDoctorRequests()
72          public
73          view
74          onlyPatient
75          returns (Doctor[] memory)
76      {
77          Doctor[] memory doctor_list = new Doctor[](doctorAddresses.length);
78          uint256 index = 0;
79
80          for (uint256 i = 0; i < patinetAddresses.length; i++) {
81              Patient memory current_patient = patients[patinetAddresses[i]];
82              for (
83                  uint256 j = 0;
84                  j < current_patient.accessRequests.length;
85                  j++
86              ) {
87                  Doctor memory current_doctor = doctors[
88                      current_patient.accessRequests[j]
89                  ];
90                  doctor_list[index] = current_doctor;
91                  index++;
92              }
93          }
```

Figure 44: EHR.sol 5

```
201     // Hospital Functions.
202     function addHospital(
203         address hospital_address,
204         string memory name,
205         string memory location,
206         string memory description,
207         string memory image
208     ) public onlyAdmin returns (Hospital memory) {
209         // Check if hospital already exiusts
210         require(
211             bytes(hospitals[hospital_address].name).length == 0,
212             "Hospital already exists"
213         );
214         // Check if name is valid.
215         require(bytes(name).length > 0, "Invalid name.");
216         // Check if location is valid.
217         require(bytes(location).length > 0, "Invalid location.");
218 
219         Hospital memory new_hospital = Hospital({
220             hospitalAddress: hospital_address,
221             name: name,
222             location: location,
223             description: description,
224             image: image
225         });
226         hospitals[hospital_address] = new_hospital;
227         hospital_count++;
228         hospitalAddresses.push(hospital_address);
229         return new_hospital;
230     }
231 
```

Figure 45:EHR.sol 6

```
249     function approveEHRRequest(address doctorAddress) public onlyPatient {
250         require(
251             patients[msg.sender].age > 0,
252             "Only registered patients can approve EHR requests."
253         );
254         require(
255             doctors[doctorAddress].age > 0,
256             "Invalid doctor address, only registered doctors can request access."
257         );
258 
259         // Check if the requesting doctor is in the access requests list of the patient
260         address[] storage accessRequests = patients[msg.sender].accessRequests;
261         bool isRequestedDoctor = false;
262         for (uint256 i = 0; i < accessRequests.length; i++) {
263             if (accessRequests[i] == doctorAddress) {
264                 isRequestedDoctor = true;
265                 break;
266             }
267         }
268 
269         require(
270             isRequestedDoctor,
271             "The requesting doctor is not in the access requests list."
272         );
273 
274         // Add the doctor to the approvedDoctors list
275         address[] storage approvedDoctors = patients[msg.sender]
276             .approvedDoctors;
277         for (uint256 i = 0; i < approvedDoctors.length; i++) {
278             if (approvedDoctors[i] == doctorAddress) {
```

Figure 46: Approve EHR request contract 1

```
277         for (uint256 i = 0; i < approvedDoctors.length; i++) {
278             if (approvedDoctors[i] == doctorAddress) {
279                 revert("Access already granted.");
280             }
281         }
282         approvedDoctors.push(doctorAddress);
283 
284         // Remove the doctor from the accessRequests list
285         for (uint256 i = 0; i < accessRequests.length; i++) {
286             if (accessRequests[i] == doctorAddress) {
287                 // Swap and pop technique to efficiently remove the element
288                 accessRequests[i] = accessRequests[accessRequests.length - 1];
289                 accessRequests.pop();
290                 break;
291             }
292         }
293 
294         // Remove the doctor from the accessRequests list
295         removeFromPatientRequests(doctorAddress, msg.sender);
296 
297         // Add the patient to the permittedPatients list of the doctor
298         addToPermittedPatients(doctorAddress, msg.sender);
299     }
```

Figure 47: Approve EHR request contract 2

```
// SPDX-License-Identifier: GPL-3.0        You, 3 days ago • Approve a

pragma solidity >=0.8.2 <0.9.0;
import "./HospitalContract.sol";

contract DoctorContract is HospitalContract {
    struct Doctor {
        address doctorAddress;
        string firstName;
        string lastName;
        uint256 age;
        string gender;
        string specialization;
        string image;
        string description;
        Hospital hospital;
        address[] permittedPatients;
        address[] patientRequests;
    }

    mapping(address => Doctor) public doctors;
    address[] public doctorAddresses;
    uint256 doctor_count = 0;

```

Figure 48: DoctorContract.sol

```
// SPDX-License-Identifier: GPL-3.0

pragma solidity >=0.8.2 <0.9.0;

contract HospitalContract {
    struct Hospital {
        address hospitalAddress;       You, 3 days ago • Approve
        string name;
        string location;
        string description;
        string image;
    }

    // Mapping to store hospitals by their address
    mapping(address => Hospital) public hospitals;
    address[] public hospitalAddresses;
    uint256 hospital_count = 0;

    function isHospital() public view returns (bool) {
        return bytes(hospitals[msg.sender].name).length > 0;
    }
}

```

Figure 49: HospitalContract.sol

```
20  function App() {
21
22    return (
23      <Router>
24        <EHRProvider>
25        <Navbar />
26
27          <Routes>
28            <Route path="/" element={<HomePage />} />
29            <Route path="/hospital/form" element={<AddHospitalForm />} />
30            <Route path="/doctor/form" element={<AddDoctorForm />} />
31            <Route path="/hospital" element={<Hospital />} />
32            <Route path="/hospital/:hospital_id" element={<HospitalDetail />} />      You, last month • IPFS imp
33            <Route path={ClientRoutes.PatientRoutes.patient_form_route} element={<PatientForm/>} />
34            <Route path={ClientRoutes.DoctorRoutes.request_access_route} element={<DoctorAccessForm/>} />
35            <Route path={ClientRoutes.PatientRoutes.patient_requests_route} element={<PatientRequests/>} />
36            <Route path={ClientRoutes.DoctorRoutes.accessed_list_route} element={<DoctorAccessList/>} />
37            <Route path={ClientRoutes.PatientRoutes.accessed_list_route} element={<PatientAccessList/>} />
38            <Route path={ClientRoutes.PatientRoutes.ehr_list_route} element={<PatientEHRList/>} />
39
40          </Routes>
41        </EHRProvider>
42      </Router>
43    );
44  }
45
46  export default App;
```

Figure 50: App.js

```
16  const getDoctorAccessList = async ({ EHRContract }) => {
17    // console.log(EHRContract)
18    return await EHRContract.getApprovedRequestsAsDoctor();
19  };
20
21  const approveEHRRequest = async ({ EHRContract, doctorAddress }) => {
22    return await EHRContract.approveEHRRequest(doctorAddress);
23  }
24
25  const revokeAccess = async ({ EHRContract, doctor_address }) => {
26    return await EHRContract.revokeDoctorAccess(doctor_address);
27  }
28
29  const isPatient = async ({ EHRContract }) => {
30    return await EHRContract.isPatient();
31  }
32
33  const isAdmin = async ({ EHRContract }) => {
34    return await EHRContract.isAdmin();
35  }       You, 1 second ago • Uncommitted changes
36
37  const PatientServices = {
38    registerPatient,
39    getSelfData,
40    getAccessRequests,
41    approveEHRRequest,
42    getDoctorAccessList,
43    revokeAccess,
44    isPatient,
45    isAdmin,
46  }
47
48  export default PatientServices;
```

Figure 51: PatientServices.js

```
import helper from "../helpers";
const addHospital = async ({
  EHRContract,
  hospitalAddress,
  hospitalName,
  location,
  description,
  image
}) => {
  return await EHRContract.addHospital(hospitalAddress, hospitalName, location, description, image);
};

const getAllHospitals = async ({ EHRContract }) => {
  return await EHRContract.getAllHospitals();
};

const uploadHospitalImage = async ({ file }) => {
  const hash = await helper.uploadFileToIPFS(file);
  return hash;
}

const getHospitalByAddress = async({ EHRContract, hospital_address }) => {
  try{
    return await EHRContract.get_hospital_by_address(hospital_address);
  }
  catch(e){
    console.log(e);
    return ;
  }
}

```

Figure 52: HospialServices.js

```
You, 6 hours ago | 1 author (You)
import helpers from "../helpers";

const addDoctor = async ({
  EHRContract,
  doctorAddress,
  firstname,
  lastname,
  age,
  gender,
  description,
  image,
  specialization,
}) => {
  return await EHRContract.addDoctor(
    doctorAddress,
    firstname,
    lastname,
    age,
    gender,
    description,
    image,
    specialization
  );
};

const isDoctor = async ({ EHRContract }) => {
  return await EHRContract.isDoctor();
};


const uploadHospitalImage = async ({ file }) => {
  const hash = await helpers.uploadFileToIPFS(file);
  return hash;
}

```

Figure 53: HospitalServices.js

```
class Routes{
    static base_route = '/';
    static hospital_route = '/hospital';
    static doctor_route = '/doctor';
    static patient_route = '/patient';
}

You, 6 hours ago | 1 author (You)
class HospitalRoutes extends Routes{
    static base_route = this.hospital_route;
    static hospital_form_route = this.hospital_route + '/form';      You, 6 hours ago • navbar
    static hospital_detail_route = (hospital_id) => `${this.base_route}/${hospital_id}`;
}

You, yesterday | 1 author (You)
class PatientRoutes extends Routes {
    static base_route = this.patient_route;
    static patient_form_route = this.patient_route + '/form';
    static patient_detail_route = (patient_id) => `${this.base_route}/${patient_id}`;
    static patient_requests_route = this.patient_route + '/access';
    static accessed_list_route = this.patient_route + '/access/list'
    static ehr_list_route = this.patient_route + '/ehr/list'
}

You, 6 hours ago | 1 author (You)
class DoctorRoutes extends Routes {
    static base_route = this.doctor_route;
    static doctor_form_route = this.doctor_route + '/form';
    static request_access_route = this.doctor_route + '/access/form';
    static accessed_list_route = this.doctor_route + '/access/list'
}
```

Figure 54: ClientRoutes.js

```jsx
import { useContext, useEffect, useState } from "react";   4.3k (gzipped: 1.8k)      You, last month
import HospitalCard from "../components/HospitalCard";
import HospitalServices from "../services/HospitalServices";
import EHRContext from "../context/EHRContext";

const Hospital = () => {
  const { EHRContract } = useContext(EHRContext);
  const [hosptials, setHospitals] = useState([]);
  useEffect(() => {
    if (!EHRContract) return;
    HospitalServices.getAllHospitals({ EHRContract })
      .then((hospitals) => {
        console.log(hospitals);
        setHospitals(hospitals);
      })
      .catch((err) => {
        console.log(err);
      });
  }, [EHRContract]);

  return (
    <>
      <h1 className="text-center text-3xl py-6 font-semibold">
        {" "}
        Available Hospitals{" "}
      </h1>
      {hosptials.length === 0 && (
        <p className="text-center w-full">No Hospitals Available</p>
      )}
      <div className="gap-8 lg:grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 xl:grid-cols-3 p-8 ">
        {hosptials.map((hospital) => {
          return (
            <HospitalCard key={hospital.hospitalAddress} hospital={hospital} />
          );
        })}
      </div>
    </>
  );
};

export default Hospital;
```

Figure 55: Hospital.js