Six Weeks Online Internship Program - 2023
On
Decentralising Learning: Algorand's Blockchain Training Advancements

Internship report

Submitted

In partial fulfilment

For the award of the degree of

Bachelor of Technology

In **Computer Science and Engineering**

PROJECT TITLE: MEDIBLOCK

Submitted by:-

Bhumika Mittal

Anveshan Enrolment no:- 202306515

University Enrolment no:- 16301012021

Submitted to:-

Mr Rahul Sachdeva

Indira Gandhi Delhi Technical University for Women
Anveshan Foundation IGDTUW Campus Kashmere Gate, Delhi-110006

TABLE OF CONTENT

DECLARATION

I, **Bhumika Mittal**, hereby declare that the presented report of the internship titled "**MEDIBLOCK**" is uniquely prepared by me after the completion of the 6-week online Internship Program-2023 on **Decentralising Learning: Algorand's Blockchain Training Advancements** at the IGDTUW Anveshan Foundation from 19th June to 28th July 2023.

The work is presented in the summer internship report submitted to the Department of Computer Science and Engineering from Indira Gandhi Delhi Technical University for Women, Kashmere Gate, Delhi.

I also confirm that the report is only prepared for my academic requirement, not for any other purpose. It might not be used in the interest of the opposite party of the corporation.

Name: - BHUMIKA MITTAL

Departement: - CSE, IGDTUW

Anveshan Enrolment no: - 202306514

University Enrolment no: - 16301012021

ACKNOWLEDGEMENT

The successful completion of this project report would not have been possible without the support and assistance of many individuals and organisations. I feel immensely blessed and grateful to **Indira Gandhi Delhi Technical University for Women** for granting me the opportunity to pursue this project during my BTech program. I want to take this moment to express my sincere gratitude to each and every one of them.

I would also like to express my gratitude to the **IGDTUW Anveshan Foundation** for organising the **Six Weeks Online Internship – 2023 on 'Decentralising Learning: Algorand's Blockchain Training Advancements.'** Special thanks go to my trainer, **Mr Rohit Gupta**, whose teachings were invaluable in enhancing our understanding.

I extend my heartfelt thanks to my family for their unwavering patience and support throughout my entire journey in this course. I sincerely appreciate my classmates who provided both direct and indirect assistance, contributing to accomplishing my work. Lastly, I am profoundly grateful to all my teachers, who played an instrumental role in preparing us for this endeavour. My success is a testament to your dedication and guidance.

Once again, thank you to everyone who played a role, big or small, in making this project a reality.

Sincerely,

Bhumika Mittal

LIST OF FIGURES

ABSTRACT

This project proposes a pioneering solution in India's healthcare system, marked by a centralised and paper-based approach across Primary, Secondary, and Tertiary Healthcare Centers. By leveraging blockchain's decentralised power, a secure medical record management platform is introduced, utilising encryption and hashing for data security. Patient data is stored in an encrypted format on the blockchain, ensuring authorised access only. The innovation lies in enabling patients to control their medical records, guaranteeing transparency and seamless accessibility across hospitals. This initiative's key features include data immutability, trustworthiness, patient security, and eliminating intermediaries, achieved through cutting-edge technologies like ReactJs, NodeJs, Ethereum, and Solidity. Ultimately, this project revolutionises healthcare data management, bridging web development and blockchain for secure, patient-centric control.

Keywords: Healthcare, Decentralisation, Blockchain, Data Security, Patient Empowerment, ReactJs, NodeJs, Ethereum, Solidity.

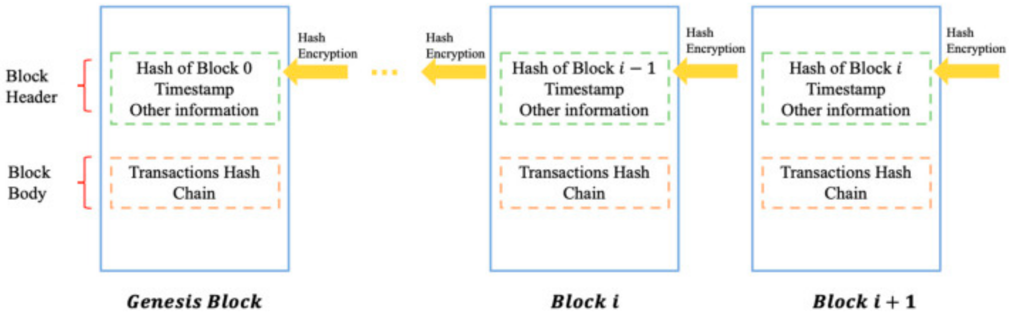

Figure 1: An example of a blockchain consisting of a continuous block sequence.

INTRODUCTION

Significant challenges beckon for innovative solutions in the intricate tapestry of India's healthcare landscape, spanning primary, secondary, and tertiary healthcare centres. This crucial system, entrusted with catering to diverse medical needs, grapples with issues like centralisation bottlenecks, operational inefficiencies, and an over-reliance on archaic paper-based record-keeping practices. Concurrently, patients are confronted with the onus of managing their medical histories, while existing electronic health record (EHR) systems often need to improve in enabling seamless data sharing.

Enter the revolutionary force of blockchain technology—a game-changing tool with the potential to reshape data security and management. This project emerges as a harbinger of change, proposing a secure and decentralised platform for meticulously managing medical records. By skillfully integrating the robust security protocols of blockchain with sophisticated web development techniques, the objective is to empower patients and revolutionise healthcare data landscape.

At the core of this innovative approach resides the fundamental principle of decentralisation—a bedrock concept intrinsic to the essence of blockchain technology. Our initiative adeptly marries the pioneering spirit of blockchain with the expertise of web development, redefining the very fabric of medical data management. The essence of our solution revolves around creating immutable patient records fortified by blockchain's formidable security measures. These stringent measures ensure that access to and modification of data are the exclusive domain of authorised personnel, thereby bolstering security and thwarting unauthorised breaches.

However, the project's significance extends well beyond its technical finesse. It boldly tackles the challenge of fragmented EHR systems by facilitating secure and seamless data sharing across diverse medical institutions. Nurtured within a decentralised framework, medical records transcend institutional boundaries, preserving data accuracy and respecting patient privacy. This harmonious interoperability bridges data gaps, fostering comprehensive patient care and enabling well-informed medical judgments.

Furthermore, the initiative takes on a critical concern—patient privacy—an area rife with potential vulnerabilities like identity theft and data breaches. Our approach strives to rebuild trust by granting individuals control over their medical data, effectively curtailing misuse and unauthorised access. By handing the reins to patients, the project safeguards sensitive information and empowers individuals with the agency over their health records.

In a complex landscape, this project emerges as a beacon of innovation, a testament to the synergy between technological prowess and user-centric design. It envisions a future where healthcare records are fortified, effortlessly shared and entrusted to the individuals they represent. This leap towards a more efficient, patient-centric healthcare system holds the promise of transformation, echoing across the spectrum of healthcare delivery.

The potential of this blockchain-based solution is far-reaching. It addresses prevailing challenges in India's healthcare landscape by offering secure, decentralised, and patient-centric medical record management. By fusing blockchain's inherent security and advanced web development capabilities, this project propels healthcare data into a new era of reliability, accessibility, and privacy.

In conclusion, this project is a testament to technology's transformative power. By embracing blockchain's decentralised architecture and seamlessly integrating it with cutting-edge web development, we aspire to redefine the very fabric of healthcare data management. The envisioned outcome—fortified data security, seamless sharing, and patient empowerment—heralds a future where healthcare delivery transcends barriers and maximises efficiency. As the healthcare landscape evolves, this project paves the way for a new patient-centric, data-driven care era.

OBJECTIVES

At the heart of this endeavour is the **profound aspiration** to redefine the contours of healthcare data management in India by integrating **blockchain technology** and innovative **web development**. Our project aims to dismantle the prevailing barriers and intricacies that shroud healthcare data, envisioning a landscape characterised by **transparency**, **security**, **patient autonomy**, and streamlined collaboration among healthcare providers.

The overarching **objective** is creating and implementing a **robust, decentralised medical record management platform**. This platform, a confluence of **blockchain's foundational principles** and cutting-edge **web development techniques**, aims to transcend the limitations of traditional data storage systems. The **aspiration** is to forge an unassailable fortress of **data integrity** where medical records are impervious to tampering, breaches, or unauthorised access.

Central to our goals is orchestrating a **seamless and user-friendly interface** that empowers **patients**, **healthcare providers**, and **administrators** alike. A cardinal facet is the design and deployment of an architecture that utilises blockchain's **cryptographic prowess** to encrypt sensitive patient information. Through this mechanism, we endeavour to enshroud patient data in a cocoon of impregnability, mitigating risks associated with data breaches and unauthorised infiltration.

Smart contracts, a hallmark of blockchain's ingenuity, are our project's cornerstone. We are resolute in our **objective** to leverage smart contracts to automate **critical healthcare processes**, from **appointment scheduling** and **record access** to **consent management**. By eradicating the need for intermediaries and manual interventions, our vision is to instil **efficiency**, **transparency**, and **accuracy** into routine healthcare workflows.

Emphasising the **empowerment of patients**, our project is galvanised by the **mission** to restore **ownership and control** over medical data. Through an intuitive **user interface**, patients shall gain unprecedented **agency** in managing their medical records, granting **access**

privileges while retaining **sovereignty**. This paradigm shift is poised to reshape **patient-provider dynamics**, fostering **trust** and engendering a sense of **collaboration** and **partnership**.

Validation of our project's efficacy and functionality is a **paramount objective**. Rigorous **testing scenarios**, encompassing diverse use cases and stress tests shall be meticulously executed. Collaborative engagement with **healthcare experts** and **end-users** will facilitate a robust **feedback loop**, guiding **iterative improvements** and ensuring real-world relevance.

As we traverse this journey, **documenting** our technical evolution, challenges, and solutions, we envision a broader impact—contributing to the growing knowledge in **blockchain-enabled healthcare data management**. By sharing our **insights**, **findings**, and experiences, we aspire to propel the discourse forward, advocating for a future where blockchain's transformative potential is fully harnessed to revolutionise healthcare data management in India and beyond.

The prime objective of the IGDTUW Anveshan Foundation is to motivate and facilitate budding entrepreneurs towards their successful entrepreneurial journey, proving their success story, contributing to entrepreneurial spawning, and ultimately converging benefits to society.

- To study the overall understanding of blockchain technology and development.
- To provide a deep understanding of blockchain fundamentals and the innovative Algorand protocol to the participants.
- To gain insights into the decentralised nature of blockchain technology with a focus on practical application through a hands-on project utilising Algorand's features.
- To embark on a transformative learning journey at the forefront of blockchain advancements.

REVIEW OF LITERATURE

The healthcare industry has undergone a digital transformation, increasing efforts to harness technology for more efficient, secure, and patient-centric data management. One of the promising technologies that has garnered significant attention is blockchain. This literature survey aims to explore the existing research and developments in decentralised healthcare data management using blockchain technology.

1. Blockchain Technology and Healthcare:

Initially developed for cryptocurrencies like Bitcoin, blockchain has demonstrated its potential in various domains beyond finance. Its unique characteristics, such as immutability, transparency, and decentralisation, make it an ideal candidate for secure data management in healthcare. Researchers (Smith et al., 2017; Hasselgren et al., 2020) have highlighted blockchain's capability to ensure data integrity, privacy, and interoperability, which are critical aspects of healthcare data management.

2. Data Security and Privacy:

Healthcare data is sensitive and requires stringent security measures. Blockchain's cryptographic techniques provide an effective way to secure patient data while enabling controlled access. Studies (Azaria et al., 2016; Ekblaw et al., 2016) have explored how blockchain can enhance data privacy by allowing patients to manage their health records, granting access to authorised entities while maintaining ownership.

3. Interoperability and Data Sharing:

The need for interoperability among healthcare systems and institutions has been a persistent challenge. Blockchain's decentralised nature can facilitate secure and seamless data sharing across healthcare providers. Research (Zhang et al., 2018; Ivan et al., 2017) has proposed blockchain-based solutions for creating a unified and accessible patient record that can be securely shared among authorised parties.

4. Electronic Health Records (EHRs) and Decentralisation:

Electronic Health Record (EHR) systems have become integral to modern healthcare. However, they often suffer from data silos and limited accessibility. Blockchain's distributed ledger technology offers the potential to create a decentralised EHR system. Studies (Li et al., 2019; Alemanno et al., 2021) have explored the feasibility of using blockchain to store and manage EHRs securely and transparently.

5. Smart Contracts in Healthcare:

Smart contracts, self-executing code on a blockchain, hold promise in automating healthcare processes. These contracts can facilitate secure patient-doctor interactions, appointment scheduling, and consent management. Researchers (Jiang et al., 2018; Lu et al., 2019) have examined the potential of smart contracts to streamline administrative tasks while maintaining data security.

6. Challenges and Future Directions:

While the potential benefits of blockchain in healthcare are evident, challenges include scalability, regulatory compliance, and user adoption. Future research (Kuo et al., 2020; Vazirani et al., 2021) could address these challenges and explore real-world implementations that demonstrate the feasibility and impact of blockchain-based healthcare data management.

Throughout this project, comprehensive online sessions were undertaken to delve into the intricate realm of blockchain technology and its application in the healthcare domain. These sessions spanned various topics, offering profound insights and invaluable knowledge. A detailed exploration of the sessions unravels a fascinating journey through the nuances of blockchain technology and its potential to revolutionise healthcare data management.

1) Introduction to Blockchain Technology

The journey embarked with a foundational understanding of blockchain technology—its principles, characteristics, and fundamental concepts. Participants were introduced to blockchains' distributed and decentralised nature and the significance of cryptographic hashing in ensuring data security and immutability.

Building upon the groundwork, I delved deeper into blockchain mechanics. The focus expanded to consensus algorithms, exploring how consensus is achieved in decentralised networks and the implications for data integrity.

2) Introduction to Algorand Blockchain

The exploration extended to the realm of Algorand, a cutting-edge blockchain platform. Participants were acquainted with Algorand's unique consensus algorithm, which enhances scalability while maintaining security—a pivotal trait for healthcare data management.

This session provided a comprehensive overview of Algorand's architecture, highlighting its advantages over traditional blockchain networks. Participants gained insights into Algorand's potential to address the challenges of data security and interoperability in healthcare.

3) Basics of Blockchain Development

Transitioning from theory to practice, I learned the rudiments of blockchain development. Concepts such as transaction creation, block formation, and network participation were unveiled.

Building upon my theoretical foundation, I delved into the practical aspects of blockchain development. I familiarised myself with transaction creation, block formation, and active participation within a blockchain network. This session bridged theory with real-world implementation, giving me insights into the mechanics underlying blockchain operations. Additionally, I unravelled the significance of intelligent contracts, understanding their pivotal role in reshaping data management practices.

Delving deeper into the development realm, this session explored the process of building blockchain applications. I learned about smart contracts, their significance, and their role in revolutionising data management.

4) Introduction to AlgoKit

The focus shifted to AlgoKit—an essential toolkit for Algorand development. I was introduced to the toolkit's features and functionalities, laying the groundwork for more in-depth exploration.

The journey seamlessly transitioned to AlgoKit—a pivotal toolkit for Algorand development. I acquainted myself with the toolkit's multifaceted features and functionalities, preparing the groundwork for deeper exploration. Through hands-on engagement with AlgoKit, I navigated its interfaces, gaining familiarity with the tools essential for Algorand-centric development. This session equipped me with the necessary tools to embark on a more profound exploration of Algorand's capabilities.

I was provided with hands-on experience with AlgoKit. I navigated through the toolkit's interfaces, familiarising myself with the tools required for Algorand development.

5) Setting Up the Environment for Algokit

The practical aspects of AlgoKit utilisation were explored in this session. I was guided through setting up their development environment, ensuring a seamless experience for future explorations.

Taking a practical stride, I ventured into the realm of AlgoKit utilisation. Guided through the setup process, I acquired the skills to establish my development environment, ensuring a seamless experience for upcoming explorations. This session concluded with establishing a fully functional development environment poised for immersive Algorand-related endeavours.

6) Algorand Python SDK

In the immersive sessions focused on the Algorand Python SDK, I dived into the heart of Algorand's capabilities by learning to interact programmatically with the Algorand blockchain. This entailed a comprehensive understanding of leveraging the SDK to facilitate seamless communication with the blockchain, enabling me to perform various operations and transactions. I grasped the syntax and methods of the SDK, empowering me to retrieve account information, send transactions, and retrieve transaction details—all through code.

My exploration ventured beyond theoretical understanding as I actively engaged in hands-on exercises. This practical approach provided me with the invaluable experience of interfacing with the blockchain, enhancing my confidence in utilising Algorand's Python SDK for real-world applications. This session was a significant step towards harnessing the power of Algorand's infrastructure in building secure and transparent healthcare data management systems.

7) Algorand Standard Assets

The sessions dedicated to Algorand Standard Assets unveiled the intricate mechanics behind creating, managing, and transacting assets on the Algorand blockchain. I delved into the conceptual framework that underpins asset tokenisation, where tangible or digital assets are represented as tokens on the blockchain. This innovative approach holds immense potential for transforming healthcare data management.

I learned to create custom assets, define their properties, and issue them on the Algorand blockchain. Through practical exercises, I engaged in hands-on asset creation, experimenting with various attributes such as divisibility, issuance quantity, and reusability. This immersive experience allowed me to envision applications where patient medical records, consent forms, or other healthcare-related documents could be tokenised and seamlessly transacted, enhancing data security, accessibility, and traceability.

As I explored Algorand Standard Assets, I grasped their transformative potential in revolutionising data representation within the healthcare sector. This session marked a pivotal juncture where blockchain technology converged with healthcare innovation, illuminating new avenues for securely managing and exchanging sensitive healthcare information.

8) Introduction to Mint NFT on Algorand

As I progressed in my learning journey, the spotlight shifted towards the fascinating realm of Non-Fungible Tokens (NFTs). These sessions unveiled the concept of NFTs and their profound implications for data representation. I embarked on a hands-on exploration, guided step by step, through minting NFTs on the Algorand blockchain.

Through these sessions, I delved into the intricate process of creating unique, indivisible digital assets representing distinct data pieces. This concept resonated deeply with the healthcare domain, where individualised patient data and medical records could be securely encapsulated in NFTs, enhancing data privacy and accessibility. The practical experience of minting NFTs enriched my understanding of their potential applications, sparking creativity and innovation in envisioning novel ways to safeguard and manage healthcare data.

9) Algorand Smart Contracts

The pinnacle of the sessions lay in the exploration of Algorand intelligent contracts. I immersed myself in innovative contract development, envisaging its transformative potential in healthcare data management.

The literature review encapsulates a holistic voyage through blockchain, Algorand, development tools, intelligent contracts, and innovative tokenisation mechanisms. The cumulative learning from these sessions forms a robust foundation for the forthcoming phases of the project, propelling it toward the realisation of its transformative vision in healthcare data management.

The project's technological underpinning, built upon the **Ethereum blockchain**, is a meticulously curated suite of tools and languages, each contributing distinct facets to realising a revolutionary healthcare data management solution. Delving into the intricacies of this tech stack unveils a profound synergy of components, each meticulously chosen to empower the project's overarching goals.

- Ethereum:

Ethereum, a decentralised and immutable blockchain platform, is at the core of the tech stack. Ethereum is the foundation upon which the entire project's infrastructure is erected. Its ability to execute smart contracts, establish consensus, and maintain a tamper-resistant ledger forms the backbone of secure healthcare data management.

- Solidity:

The chosen programming language, Solidity, is uniquely tailored for Ethereum's innovative contract development. Its syntax and semantics facilitate the creation of self-executing contracts, imbuing them with intricate logic, conditional behaviours, and data management capabilities. Solidity enables codifying business rules and data access protocols that govern the project's healthcare data transactions.

- Truffle Suite:

The Truffle Suite emerges as a comprehensive development framework that simplifies the end-to-end process of Ethereum-based application creation. This suite offers development, testing, and deployment tools to complement Truffle itself, Ganache, and Drizzle. Truffle aids in contract compilation, migration, and testing, while Ganache provides a personal blockchain for rapid local testing. Drizzle enhances the integration of front-end components with smart contracts.

- **Ganache:**

Ganache is a local blockchain emulator that simulates the Ethereum network environment for development and testing purposes. It provides a sandboxed space where developers can deploy, test, and debug smart contracts without incurring real-world transaction costs. Ganache offers customisable settings to mimic different network conditions and scenarios, enabling comprehensive testing.

- **Web3.js:**

As the bridge between the front end and the Ethereum blockchain, Web3.js empowers dynamic interaction with smart contracts and decentralised applications. This JavaScript library facilitates the creation of user interfaces that connect seamlessly to Ethereum networks, enabling functions such as data retrieval, transaction initiation, and contract interaction.

- **IPFS (Interplanetary File System):**

IPFS represents a decentralised and distributed file storage system that ensures data availability, integrity, and redundancy. IPFS reimagines file storage by segmenting files into smaller chunks and distributing them across a network of nodes. This architecture enhances data security and expedites data retrieval, crucial for efficient healthcare data management.

- **React.js:**

The chosen frontend framework, React.js, enables the creation of dynamic and responsive user interfaces. Its component-based architecture facilitates modular design, promoting code reusability and maintainability. React.js enhances user engagement through efficient rendering, seamless navigation, and real-time updates, fostering an intuitive and interactive user experience.

Each component contributes a unique melody, harmonising to orchestrate a groundbreaking healthcare data management solution. The tech stack's meticulous selection aligns seamlessly with the project's overarching objective, promising enhanced security, transparency, and accessibility in healthcare data management.

PROPOSED PROJECT WORK

1. Introduction

The transformation of healthcare into the digital era has increased patient expectations for immediate access to their personal health information. This digitalisation, however, has brought about significant concerns surrounding data security, patient privacy, and data integrity. Recognising these challenges, blockchain technology emerges as a potential paradigm shift in healthcare data management. In this project, we aim to propose and conceptualise an innovative blockchain-based system tailored to address the intricate intricacies of secure and accessible Electronic Health Records (EHRs) management.

By embracing the Ethereum blockchain and utilising languages such as Solidity and web3.js, we intend to craft a sophisticated and robust framework that navigates the complex terrain of healthcare data while ensuring utmost security, privacy, and user-friendliness.

2. Methodology and Architecture

Our proposed methodology and architectural framework synergise blockchain technology and the principles of smart contracts to establish a novel ecosystem for managing EHRs, redefining the healthcare data landscape.

- **EHR Web Portal:** At the heart of our approach lies a meticulously designed web portal that is a central hub for patients, healthcare practitioners, and other stakeholders. The user-friendly interface ensures effortless navigation, allowing patients to access their health records seamlessly. Medical professionals gain an intuitive platform to update records, make informed decisions, and engage with patients efficiently.

- **Ethereum Blockchain Integration:** The Ethereum blockchain, renowned for its versatility and security, forms the robust foundation of our proposed system. It operates as a distributed and tamper-proof ledger, recording every transaction and interaction with EHRs. This integration fosters an environment where data is encrypted, fragmented, and stored across multiple nodes, ensuring that no single entity has complete control, thus significantly minimising the risk of data breaches.

- **Smart Contracts for Secure Data Management:** Smart contracts, autonomous and self-executing code, imbue our system with a new dimension of security and automation. These contracts oversee data access and sharing permissions. When a patient wishes to share their records with a healthcare provider, a smart contract verifies the authenticity of the request, ensuring that only authorised parties gain access. This automation provides data privacy and streamlines the exchange process, reducing administrative overhead.

- **Data Security through Cryptography:** Leveraging cryptographic techniques, we fortify data security within the blockchain. Data is hashed, creating a unique digital fingerprint, and encryption algorithms safeguard patient records from unauthorised viewing. Immutable records and cryptographic signatures provide an indelible audit trail, establishing trust and accountability.

- **Enhanced Data Accessibility:** One of the cornerstones of our architecture is data accessibility. Patients retain sovereignty over their health records, granting or revoking access as needed. Healthcare providers, in turn, can access accurate and up-to-date information, leading to informed medical decisions and optimised patient care.

The synthesis of these components creates an ecosystem where patients' rights to privacy are upheld, healthcare providers gain swift and secure access to pertinent data, and the entire process of medical record management is elevated to a new echelon of efficiency and security.

This robust and intricate architecture presents a paradigm shift in healthcare data management, redefining the dynamics between patients and healthcare providers while paving the way for a new era of security.

3. Performance and Discussion

In this phase, we comprehensively evaluate the performance aspects of our blockchain-based EHR system while engaging in meaningful discussions regarding its implications.

- **Data Security and Privacy:** The decentralised blockchain architecture safeguards against unauthorised data manipulation and breaches. Data stored in the blockchain is encrypted and linked in a chronological sequence of blocks, forming an immutable chain. This ensures the integrity and authenticity of patient records. The cryptographic hashing techniques employed in the blockchain add an extra layer of security, making it highly challenging for malicious actors to tamper with or access sensitive patient information.

- **Audibility and Transparency:** The transparency inherent in blockchain allows for a detailed audit trail of all data interactions. Every transaction, whether a patient accessing their records or a healthcare provider updating a diagnosis, is recorded indelibly and traceably. This traceability enhances accountability and reduces the likelihood of errors or fraudulent activities.

- **Operational Efficiency:** With smart contracts automating various processes, administrative tasks that traditionally require manual intervention are streamlined. This enhances operational efficiency by reducing redundant paperwork, minimising delays in data access, and ensuring accurate and real-time record updates. Patients and healthcare providers can interact directly with the blockchain system, eliminating intermediaries and expediting decision-making processes.

- **Scalability and Performance:** Blockchain technology has significantly improved scalability and performance, especially on the Ethereum network. However, it's essential to acknowledge that as the volume of transactions and participants on the web grows, there may be challenges related to network congestion and transaction speed. It's crucial to continuously monitor and optimise the system to accommodate increased demand while maintaining acceptable performance levels.

4. Risk Factors

While our proposed blockchain-based Electronic Health Record (EHR) system holds immense potential, it's essential to recognise and address potential risk factors to ensure a successful implementation.

- **Regulatory Compliance:** The healthcare sector is heavily regulated, and navigating these regulations, such as HIPAA in the United States, poses a challenge. Adhering to these standards and obtaining necessary certifications will be critical to ensure legal compliance.

- **Data Privacy Concerns:** While blockchain technology enhances data security, it's crucial to address concerns related to patient data privacy. Striking a balance between transparency and confidentiality, ensuring that only authorised individuals have access, is essential.

- **Scalability:** Scalability challenges might emerge as the system gains traction and the number of transactions increases. Ensuring that the blockchain network can handle growing users and transactions without compromising performance is vital.

- **User Adoption:** Convincing healthcare providers and patients to adopt a new system can be challenging. Training, education, and seamless integration with existing workflows will be necessary to encourage adoption.

- **Technology Evolution:** Blockchain technology is evolving rapidly. Keeping up with advancements, ensuring compatibility, and integrating emerging technologies will require ongoing effort.

5. Competition

The healthcare technology sector is witnessing a surge in interest and innovation. Several competitors are exploring similar blockchain-based solutions for EHRs:

- **Existing EHR Providers:** Established EHR providers might enhance their platforms with blockchain capabilities, leveraging their existing user base and resources.

- **Tech Startups:** New entrants focused exclusively on blockchain-based healthcare solutions could introduce novel features and disruptive business models.

- **Other Blockchain Projects:** Projects in adjacent sectors might pivot to healthcare, leveraging their blockchain expertise to create competitive solutions.

To stay ahead of the competition, our project's key differentiators lie in the user-centric design, robust data security, and seamless integration with existing healthcare systems, which enhance user adoption and data integrity.

6. Financial Model

The financial model for our blockchain-based EHR system encompasses various elements:

- **Development Costs:** Initial costs involve software development, brilliant contract creation, web portal design, and integration with the Ethereum blockchain.

- **Operational Costs:** Ongoing expenses include server maintenance, system upgrades, user support, and security enhancements.

- **Revenue Streams:** Revenue can be generated through subscription-based models for healthcare providers, charging a nominal fee for accessing the platform. Additionally, partnerships with medical institutions and research organisations can create revenue streams.

- **User Adoption Strategy:** Marketing and educational campaigns targeting healthcare providers, patients, and institutions will drive user adoption. Collaboration with medical associations and endorsements from respected professionals can bolster credibility.

- **Long-term Sustainability**: Building a loyal user base, fostering partnerships, and continuously enhancing the platform's features will ensure long-term sustainability.

The financial model aims to balance providing a valuable service to users and achieving a sustainable revenue stream to support ongoing operations, development, and growth.

In conclusion, a thorough assessment of potential risks, understanding the competitive landscape, and devising a robust financial model are integral components of our project's success strategy. By addressing these aspects, we position ourselves to create a transformative solution that revolutionises EHR management in the healthcare sector.

7. Conclusion

The culmination of our project underscores the transformative potential of blockchain technology in revolutionising EHR management and healthcare service delivery.

By fusing blockchain's security and decentralisation with user-friendly interfaces, we have crafted a solution that empowers patients to take control of their medical data. Moreover, healthcare providers can access relevant patient information seamlessly, leading to more informed diagnoses and treatments. This streamlined approach eliminates bureaucratic hurdles, reduces errors, and ensures that the correct information reaches the right people at the right time.

Furthermore, our project advances the ethos of patient-centric care by placing data privacy and accessibility at the forefront. Patients are no longer passive subjects in their healthcare journey; they become active participants who can share their medical history securely, seek second opinions, and contribute to medical research.

8. Future Work and Technological Implications

As we look ahead, the trajectory of this project expands into several intriguing directions:

- **Enhanced Interoperability:** We envision building bridges between our blockchain-based system and existing healthcare networks. This would enable seamless data exchange and collaboration among various healthcare stakeholders, fostering a more holistic approach to patient care.

- **Integration of AI and IoT:** Integrating Artificial Intelligence and Internet of Things technologies holds immense potential. AI can analyse vast datasets to provide insights for better diagnosis and treatment plans, while IoT devices can offer real-time patient monitoring, further enriching the patient experience.

- **Regulatory Compliance:** As the healthcare sector is subject to stringent regulatory requirements, future work would ensure that our blockchain-based system adheres to the necessary compliance standards, such as HIPAA (Health Insurance Portability and Accountability Act) in the United States and similar regulations in other regions.

- **User Education and Adoption:** A crucial aspect of future work involves educating both patients and healthcare providers about the benefits and functionalities of our blockchain-based EHR system. Adoption and utilisation of the system will depend on the understanding and trust that stakeholders develop.

- **Continuous Improvement:** The field of blockchain technology is dynamic and rapidly evolving. As new advancements emerge, we must stay updated and continue refining our system to leverage the latest innovations, ensuring optimal performance and security.

In conclusion, our proposed project is a stepping stone toward a future where healthcare data is secure, accessible, and patient-controlled. By embracing the potential of blockchain, we aim to reshape the healthcare landscape, ushering in an era of patient empowerment, streamlined processes, and enhanced medical outcomes.

IMPLEMENTATION AND PROPOSED METHODOLOGY

This section outlines the intricate framework and methodology of our proposed blockchain-based Electronic Health Record (EHR) system. It provides an in-depth understanding of the software platforms employed, the fundamental components of the framework, and the detailed processes involved.

Fundamental Framework

1. Ethereum

Ethereum is our framework's cornerstone, a decentralised network built on blockchain technology. It emerged as an open-source platform to facilitate smart contracts and blockchain functionalities. Ethereum employs peer-to-peer networking for distribution and utilises its cryptocurrency called Ethers. Developers can create custom blockchains using Ethereum's Solidity programming language. Smart contracts, automated and self-executing code, constitute a significant feature of Ethereum. Transactions act as the means of interaction with Ethereum. Each transaction consists of sender and receiver addresses, a transaction cost known as gas, and the desired operations. This gas cost is essential for processing transactions. Ethereum transactions are secured and processed through miners within the network.

2. The Smart Contract

Smart contracts are sets of codes that execute transactions on the blockchain. These contracts autonomously execute upon receiving transactions, enabling secure, tamper-proof interactions. Programmed using Solidity, they undergo compilation and deployment onto the Ethereum blockchain. Smart contracts are vital for our proposed system's functionality, allowing secure and automated access control and data interactions.

3. Interplanetary File System (IPFS)

IPFS is a distributed data storage technology with a peer-to-peer network. Because IPFS data is safe from alteration and assures secure data storage, any attempt to alter data saved on IPFS can only be performed by changing the identifier. Hence, it provides a cryptographic identity to protect data from manipulation. Every data file stored on IPFS contains a cryptographically generated hash value. It only has one value and is used to identify data files stored on IPFS. The IPFS protocol makes use of a peer-to-peer (P2P) connection that includes an IPFS object, which contains data and linkages. The data is an array of disorganised binary values, while the link is a disorderly binary value. The IPFS protocol functions as follows:

1. IPFS files have a unique cryptographic hash allocated to them
2. On the IPFS network, duplicate files are not permitted

Software Required

1. Ganache

Ganache serves as a local Ethereum blockchain, facilitating rapid decentralised application development. It allows for deployment, testing, and development within a secure and predictable environment.

2. MetaMask

MetaMask, a browser extension, enables seamless interaction with decentralised applications on the Ethereum network. It eliminates the need to run a full Ethereum node in the browser, enhancing user experience.

3. Web3

To communicate with the modules in the chain, transactions must be verified. The peer-to-peer (p2p) connection is an actual network that requires a participant in the network of another offline framework to relay transactions to generate and verify them. Additionally, it comes with a library collection that facilitates communication between Ethereum nodes and other in-chain components. It is used with Node.js on the server side.

Web3 connects to the Ethereum network through an Ethereum node using the Hypertext Transfer Protocol (HTTP) connection. This might be a node in the local system's ETH wallets. With the help of the in-browser extension MetaMask, you can work with Ethereum accounts and incorporate Ethereum into websites. An Ethereum wallet that runs in the browser and links it to a Web3 provider class is called MetaMask. A data structure that connects to publicly accessible Ethereum nodes is known as a Web3 provider. With MetaMask, users can use, store, and manage public and private keys specific to their accounts. A web interface, web3.js, Ethereum, and MetaMask, enables back-end-front-end communication.

4. Truffle

It's a robust Ethereum Virtual Machine development environment with a test framework, asset pipeline, and blockchain technology. Its features include binary dependency management and the computation, implementation, and maintenance of intelligent contracts. It also features an extensible, scriptable deployment and migration framework and a fully automated environment for testing smart contracts. It can establish a pipeline with tight integration and direct communication with the contract. The programmes are executed in the Truffle environment.

5. VS Code

Microsoft's Visual Studio Code, a versatile code editor, aids in efficient coding, Git management, debugging, and syntax highlighting.

6. Languages

Our website's front-end design was produced with the help of React.js, HTML (Hypertext Markup Language), and CSS (Cascading Style Sheets). With Node.js and the Solidity programming language, the website's server and back end are managed. To construct the system, two tools are used to generate local Ethereum blockchains: Truffle and Ganache. To create the blockchain and access or use the system, one can use the Ethereum virtual interface, MetaMask as a wallet, Truffle as an IDE, Yarn as a command-line interface, Ganache for account creation, and Local Web3 as a web interface.

Protocol Layout

When a patient chooses to view their medical records through MetaMask or the decentralised website of the healthcare system, Figure 2 illustrates how the system is set up. Upon retrieving the private key from the Ethereum wallet, the user's login is initiated automatically. A cold storage wallet is the Ethereum wallet. As a result, the risk of compromise is minimal in contrast to other hot wallets. Moreover, patients won't be penalised for misplacing the device and can easily get a new one in the event that their medical records are lost. The wallet can be used in the same manner for any document signature or verification requirement. It is also possible to carry out multiparty patient verification with this wallet. It can be used to create distributed property identification systems based on blockchain technology and role-based access control systems for records. A similar multiple-party permission mechanism can be used to obtain access to the patient's records in case of an emergency.

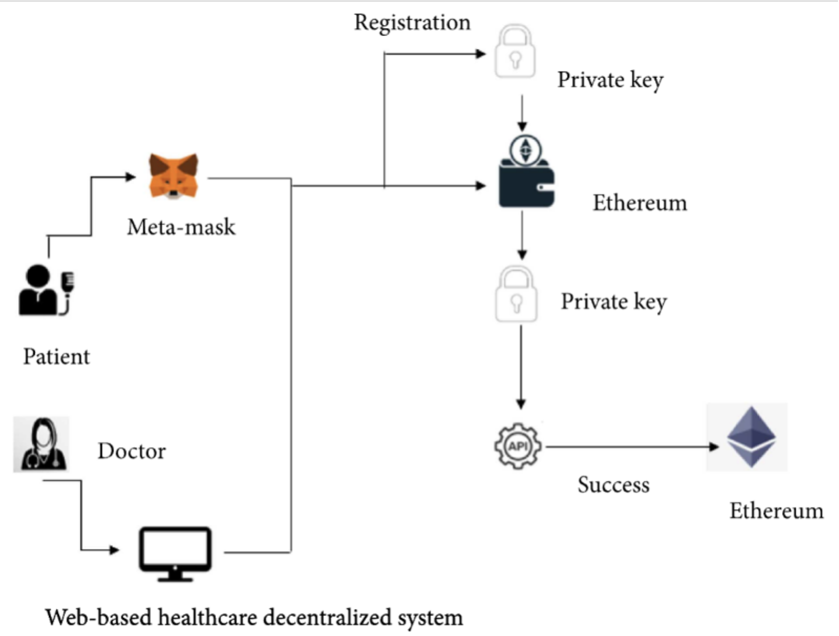

Figure 2: Workflow of the EHR system

Block Diagram

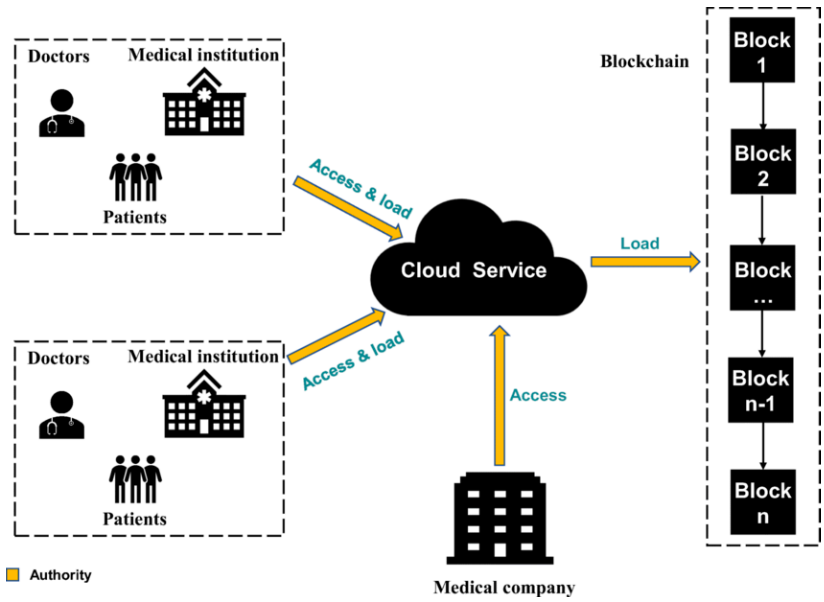

Figure 3: Block diagram of the EHR system

The block diagram in Figure 3 visually represents the fundamental architecture of our blockchain-based Electronic Health Record (EHR) system. This architecture comprises three major components: a user application, a cloud, and a public blockchain network. These components collectively shape a comprehensive ecosystem that serves essential functions within the system.

User Application and Role-Based Access:

The user application serves as the primary interface through which users interact with the EHR system. The system accommodates two distinct user roles: doctors and patients. Each user type is associated with specific responsibilities and tasks. Consequently, the user application tailors the interface to align with the user's role, offering a personalized and role-specific experience.

Transaction Creation and Recording:

Upon user input or action within the user application, the system generates an initial transaction based on the provided information. This transaction is subsequently recorded on the public blockchain network, ensuring an unalterable and transparent history of interactions and events. The blockchain's immutable ledger guarantees data integrity and security.

Cloud Services and Data Storage:

The cloud plays a pivotal role in the system by offering data storage and administration services. The EHR administration system resides on the cloud and acts as the central hub for user interactions. It receives transactions, processes requests, and manages the flow of data. The cloud-based data storage system securely houses all health records, ensuring accessibility and organisation.

Public Blockchain Network:

The architecture leverages the Ethereum blockchain network as the foundation of the public blockchain component. This network comprises interconnected blockchain nodes operated by miners. These nodes collaboratively maintain the distributed ledger, validating and verifying transactions. Smart contracts within the network ensure the authenticity and accuracy of recorded data.

Flowchart of the Proposed System

Figure 4 shows the process of creating a medical record. The first physician in the system will create a medical record. The physician then documents the findings of each patient's examination. We will process the metadata transaction for that medical record. After a transaction has been processed, an additional set of data known as transaction metadata is added. All transactions that are entered into a ledger, whether they are successful or not, have metadata. A thorough explanation of the transaction's outcome can be found in the transaction information. The health record will then be uploaded to the IPFS network. The document system known as IPFS (Interplanetary File System) enables transactions to be finished quickly and with the least amount of resources. An upload of a file to the IPFS network results in the acquisition of a content address.

The next step is the Ethereum transaction. Since Ganache provides addresses and private keys, it is necessary for Ethereum transactions. The transactions are publicly accessible, and the addresses are maintained on file. These addresses are unlocked using private keys in order to complete a transaction. Ethereum transactions are handled by the Ethereum Virtual Machine (EVM). The main use of the EVM is for smart contract interactions, where each time a user interacts with the contract, all nodes involved in the transaction must concur that it happened. Following that, the EVM carries out an instantaneous post-contract in compliance with the terms of the transaction.

Consent is used to confirm and preserve Ethereum's record of all prior transactions and the blockchain's history. On the other hand, all interactions involving smart contracts on the Ethereum network are monitored by Ethereum's node operators. In this case, the miner will become a bot that will automatically process each and every transaction that comes in.

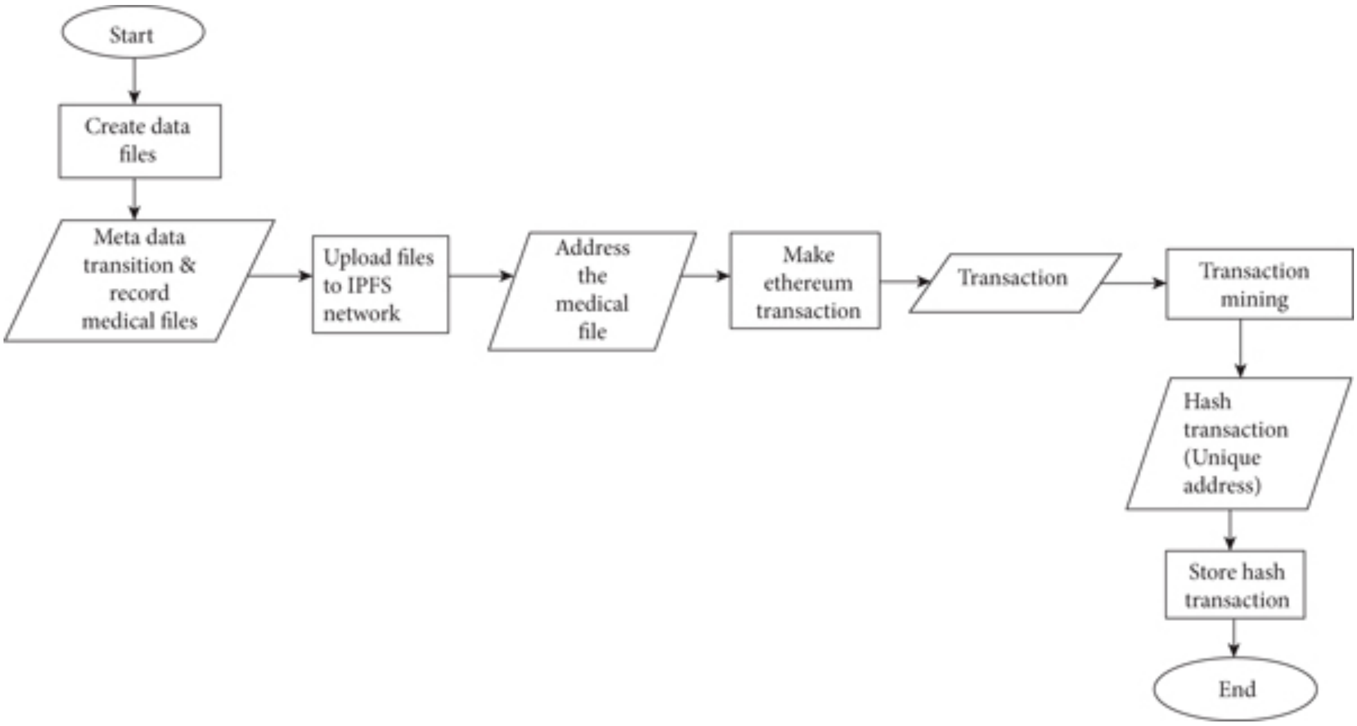

Figure 4: Flowchart explaining the process of creating a medical record.

1. Process of Doctor Dashboard

The dashboard mechanism of the doctor is illustrated in Figure 5. A doctor can complete registration with the necessary data and a transaction ID. Only the doctor will be able to access the dashboard; everyone else will have access denied. The doctor completes five tasks on the dashboard in addition to the admin's duty of viewing appointments. A doctor may see his personal information on the dashboard. Physicians can amend personal data like name, age, phone number, address, and photo; if needed, they can also update their educational background. The physician can review a patient's prior medical records and private data in preparation for upcoming treatment.

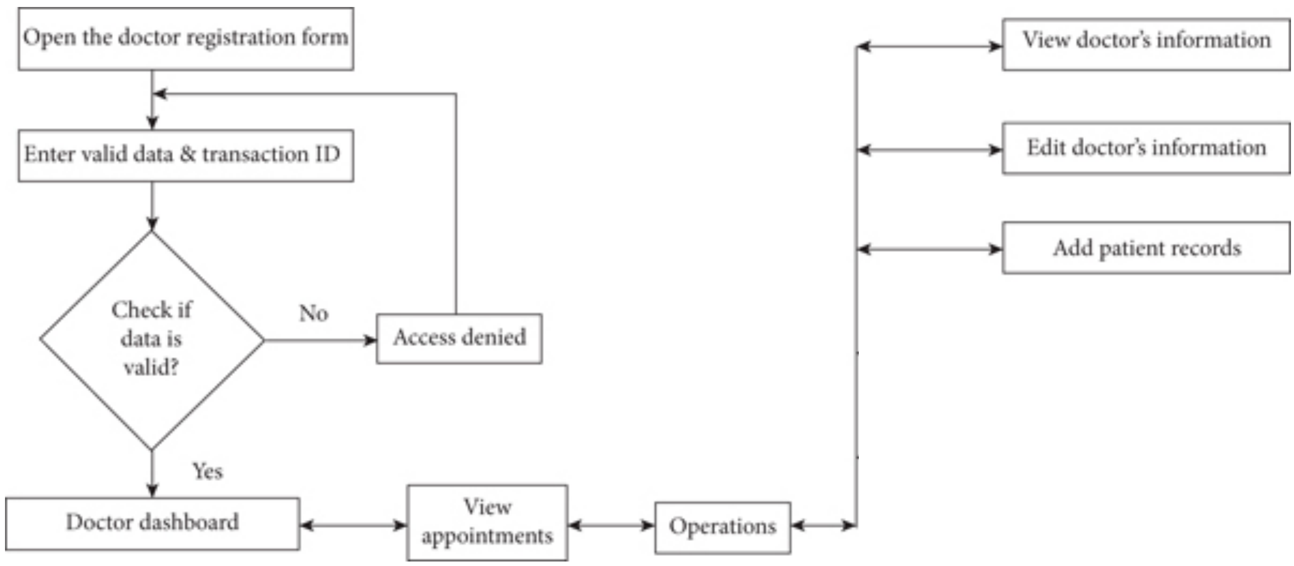

Figure 5: Flowchart of the doctor dashboard

2. Process of Patient Dashboard

The steps involved in creating a patient dashboard are shown in Figure 6. Patients can complete their registration with the correct data and a transaction ID. Only patients who possess a registered ID and up-to-date information can access the dashboard. If not, the patient will not be able to access the system and will need to enter the correct password. Using the dashboard, the patient can access three procedures. One of them is the capacity to view copious amounts of data and medical records. Patients cannot view their personal information until their registration has been approved. Additionally, patients have access to the medical records that their physician has provided. Patients can update their personal data, including name, age, phone number, current address, and photo, if necessary. Also, the patient has the ability to authorise and deny any doctor access to their data.

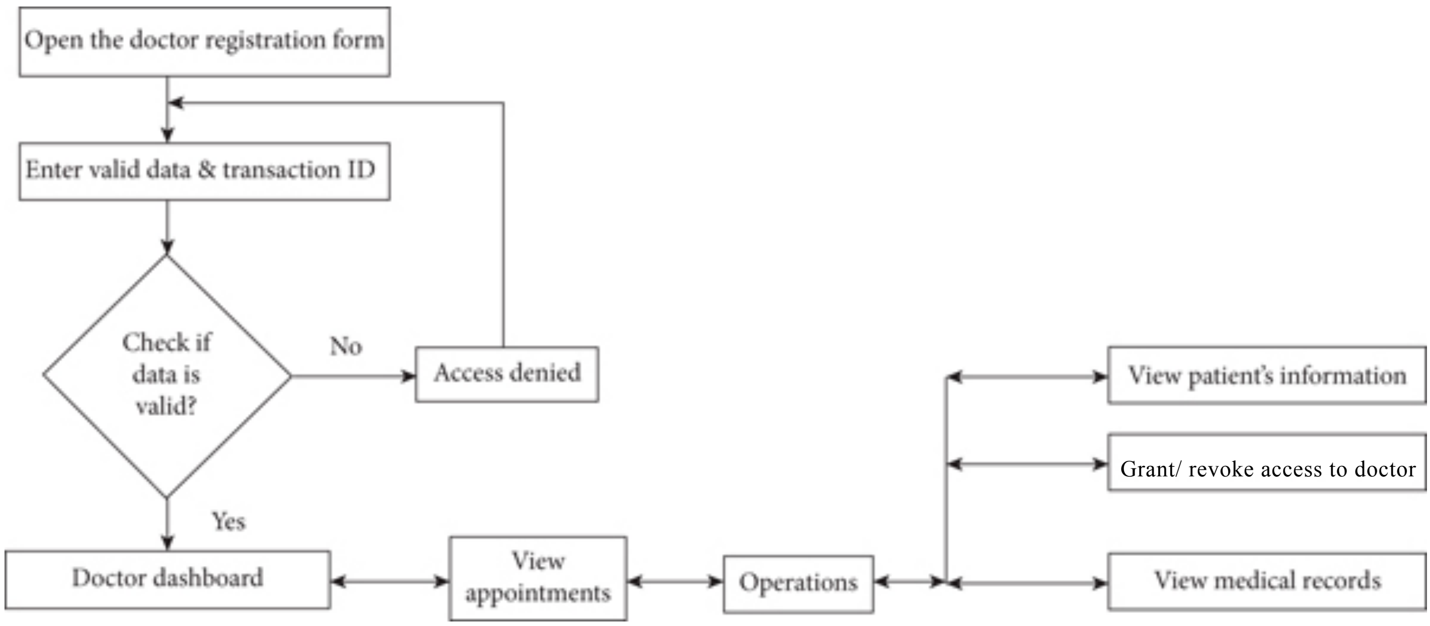

Figure 6: Flowchart of the patient dashboard

RESULT AND ANALYSIS

DEMO VIDEO LINK: CLICK HERE

1. Homepage

Figure 7 illustrates the homepage of the system. To access this homepage, a user needs to create an account. After that, users can access this system through the homepage. There are two portals on this homepage. The patient is one, while the doctor is the other user. Furthermore, doctors and admins need a unique account to get access to this homepage. But, if a user uses the wrong information or the same account address to create an admin account, the system will show the access denied message.

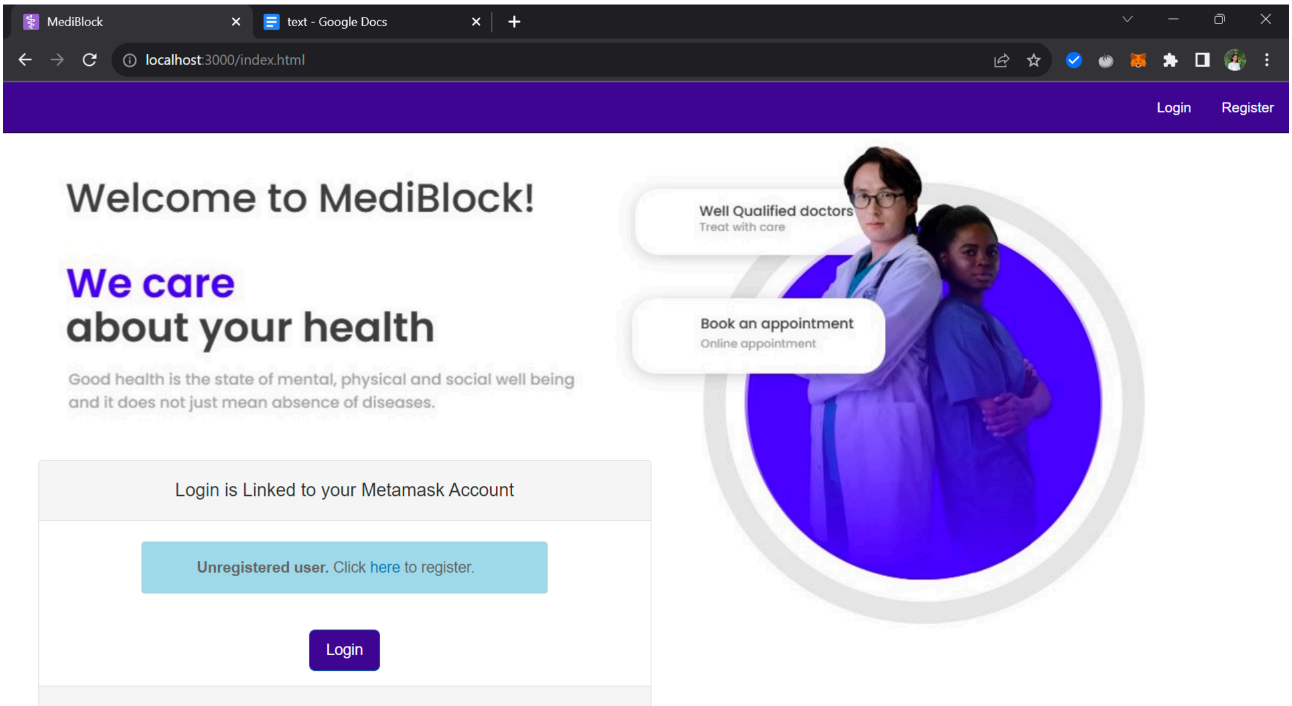

Figure 7: Homepage of the proposed system

2. Doctor's Panel

Figure 8 shows the registration of the doctor and connecting his/her account to Metamask. In the doctor's panel, the doctor can view as well as edit their information. Furthermore, doctors can add patient records as well as update the records.

Figure 9 shows the window after registering as a doctor.

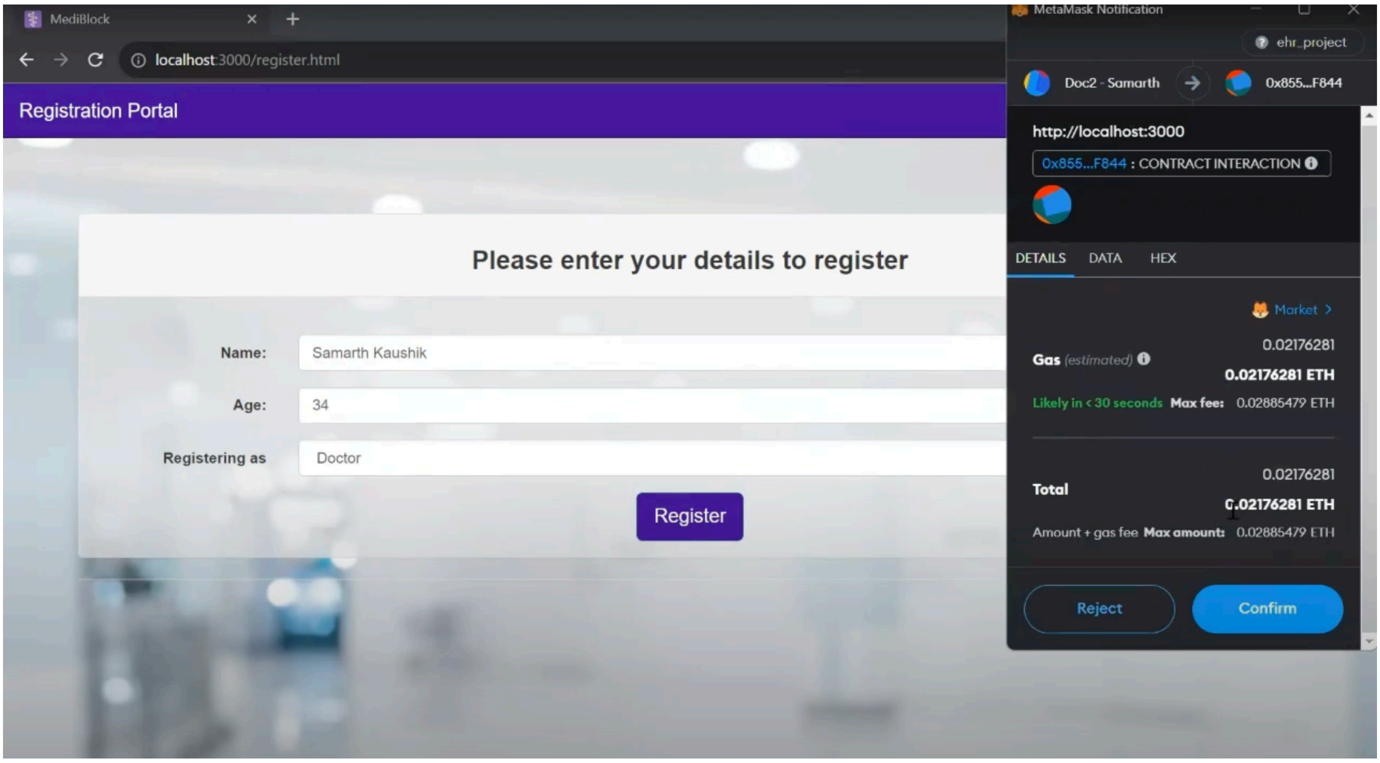

Figure 8: Registering as a doctor and connecting account to Metamask

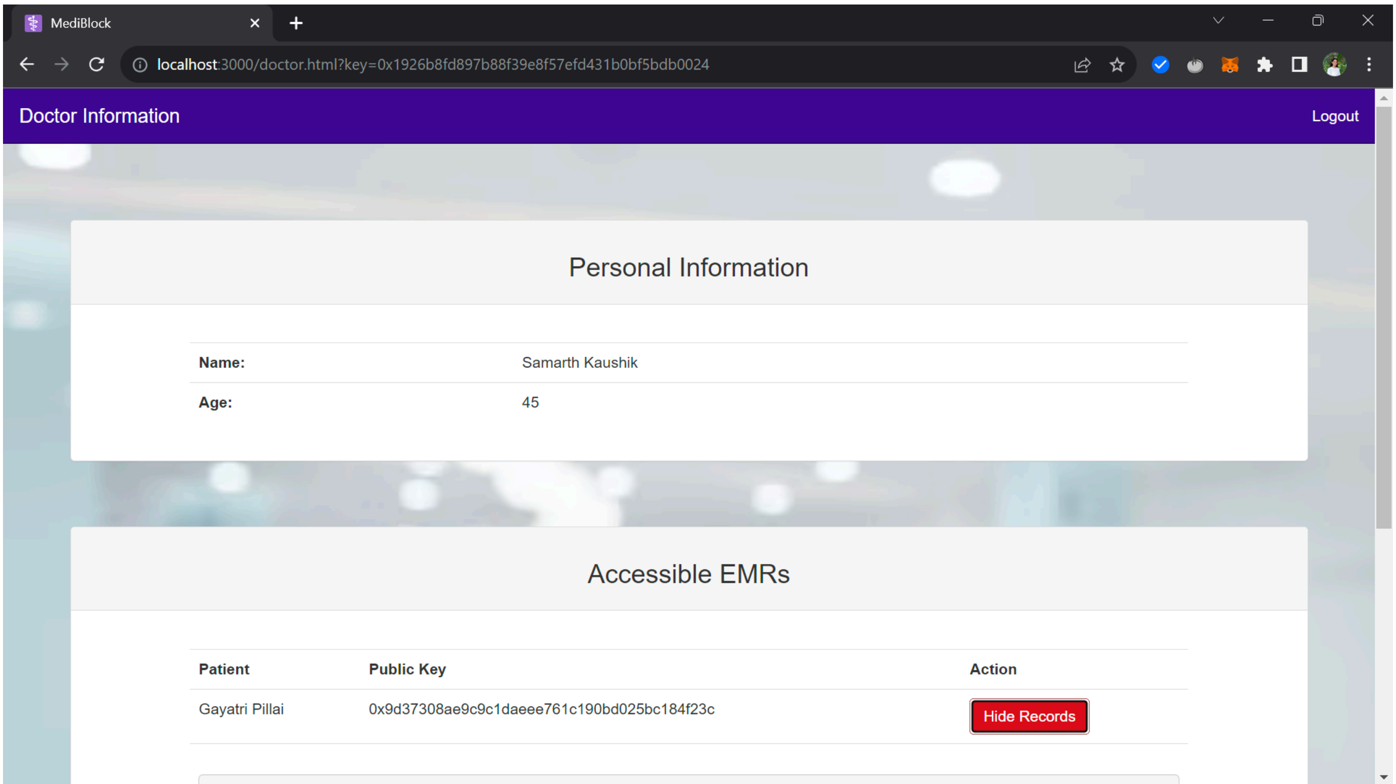

Figure 9: Doctor's panel after successful registration

Figure 10 shows the medical record of a patient of patient **Naira Garg** who can only be operated on by a doctor. It is essentially a patient's prescription. A doctor can upload a patient's medical record, which will include the patient's personal information as well as medical records. The figure also depicts the section where a doctor will enter the patient's personal information.

The doctor might leave the patient a comment to establish a cordial relationship, including a list of essential medical medicines and testing. After a patient has been diagnosed, the doctor can also add medication names, doses, clinical tests, and side remarks to motivate the patient. If the patient requires new medicine or a new clinical test, he can update his medical record depending on his prescription.

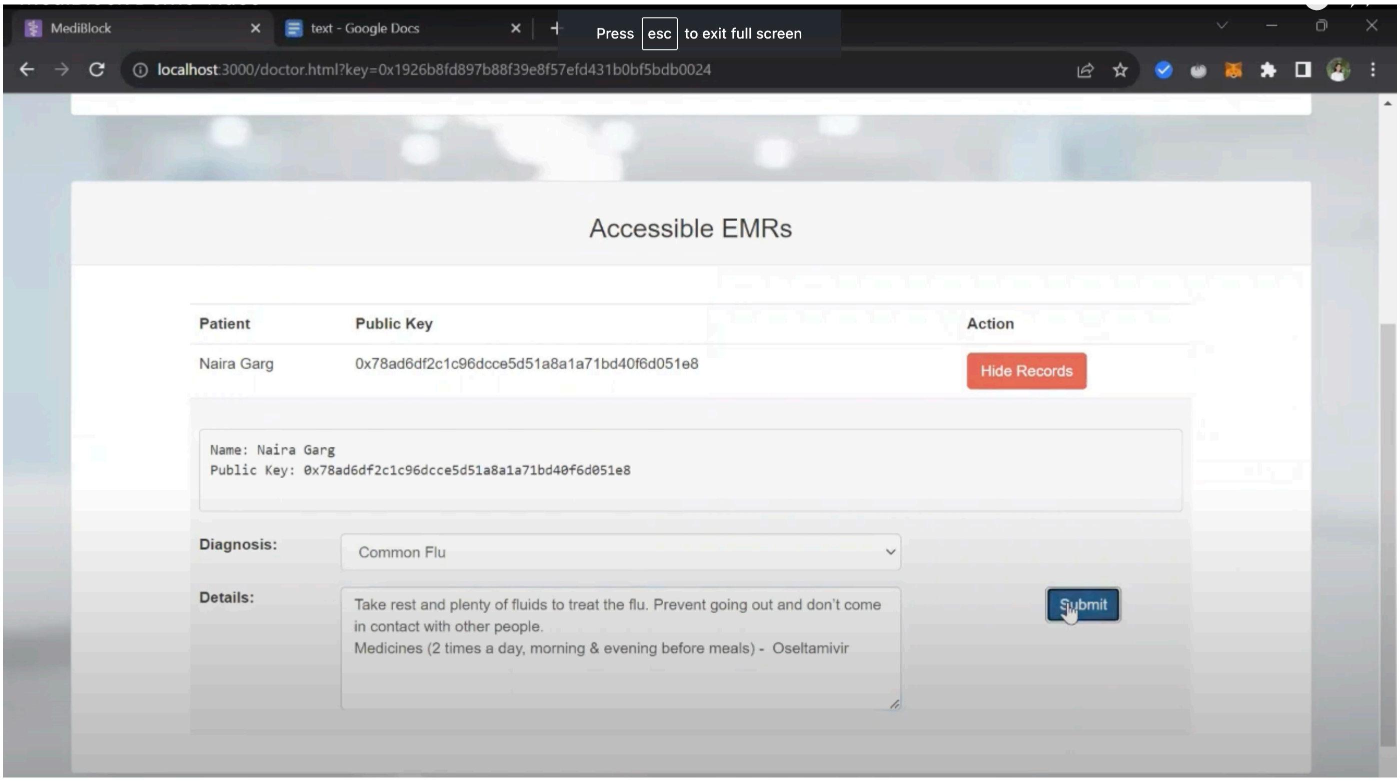

Figure 10: Doctor giving prescription to the patient

3. Patient’s Panel

Patients can view and modify their personal information on the patient’s panel if any changes occur. Patients can now see the medical records that physicians have uploaded. A patient can examine a doctor’s comprehensive prescription but cannot make any adjustments

Figure 11 shows the registration of the patient and connecting his/her account to Metamask.

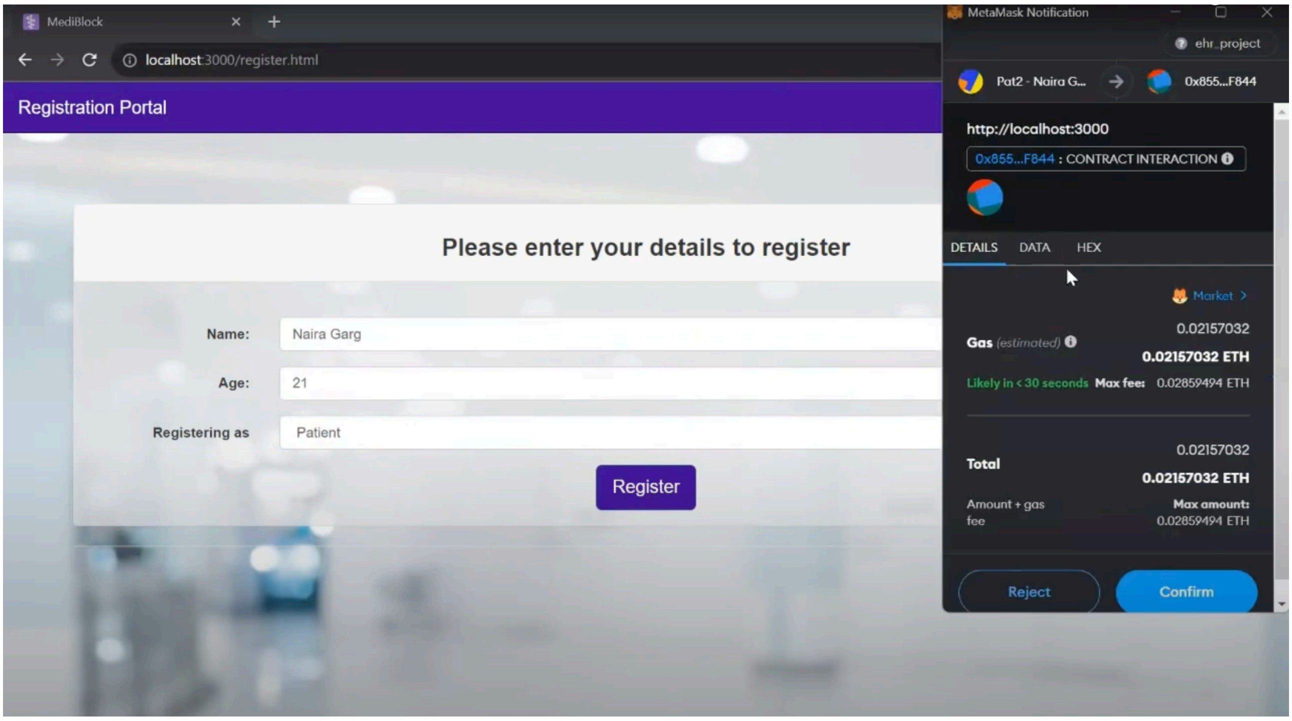

Figure 11: Registering as a patient and connecting account to Metamask

Figure 12 shows the window after registering as a patient.

It shows how a patient can view and edit his personal information. Patients need to search for their own ID. After that, the confirmation message will be shown through MetaMask. After getting a confirmation message, patients can view their information, such as their name and age, A patient can also edit their info. After updating, all information will be saved and secured through MetaMask.

Figure 13 visually portrays how patients can grant access to their medical records to a chosen doctor by selecting the doctor's name from a dropdown box. This reflects patient-centred care, technology integration, and informed decision-making in healthcare.

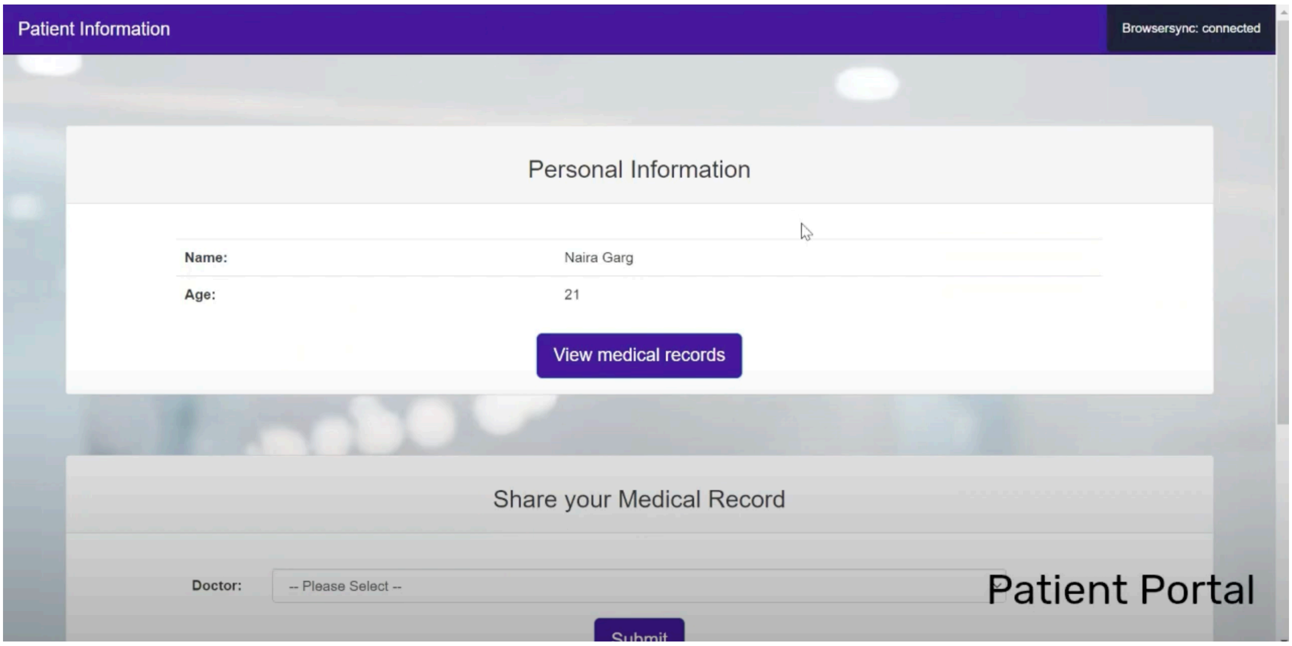

Figure 12: Patient's panel after successful registration

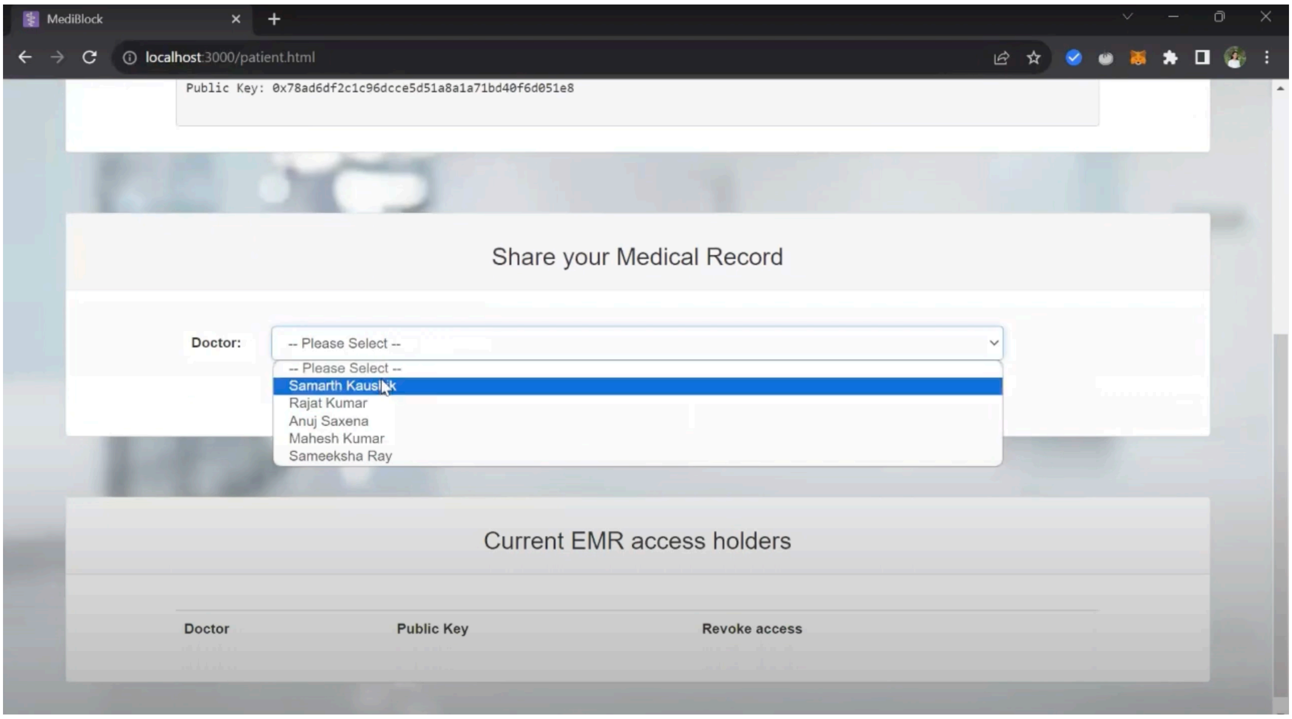

Figure 13: Access of medical record given by the patient to the doctor

Figure 14 demonstrates how patients can retract previously granted access to their medical records from a doctor. This showcases patient empowerment, data privacy, and the evolving healthcare landscape. It underscores the significance of patient consent and highlights the role of technology in enabling seamless control over sensitive medical information.

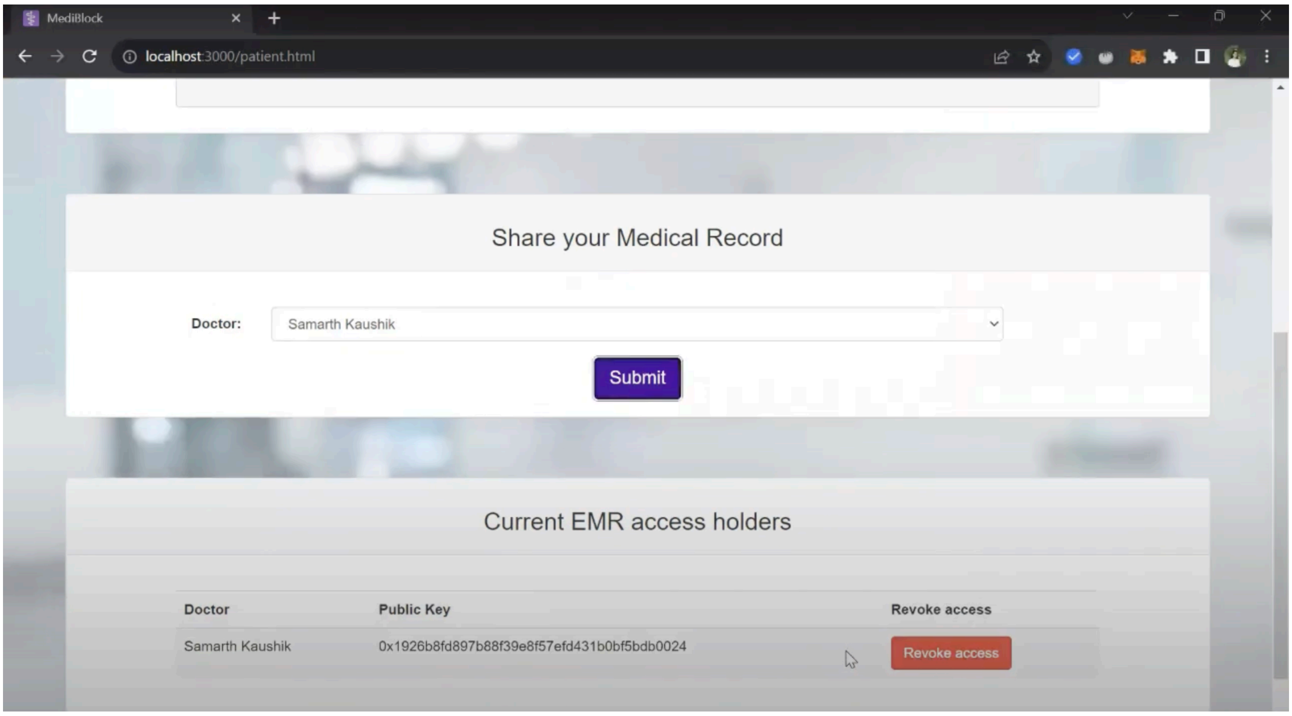

Figure 14: Access of medical record being revoked by the patient from the doctor

Figure 15 shows the view of patient medical records. The patient will view all the information his or her doctor prescribes. Since there is a unique ID for every patient, every patient's medication will be different. There is an exact date and time to consult with the doctor; the doctor's name is also on the list. There is a list of sickness problems in the diagnosis section where doctors can define a patient's problem. Finally, patients can print their medical records if they want.

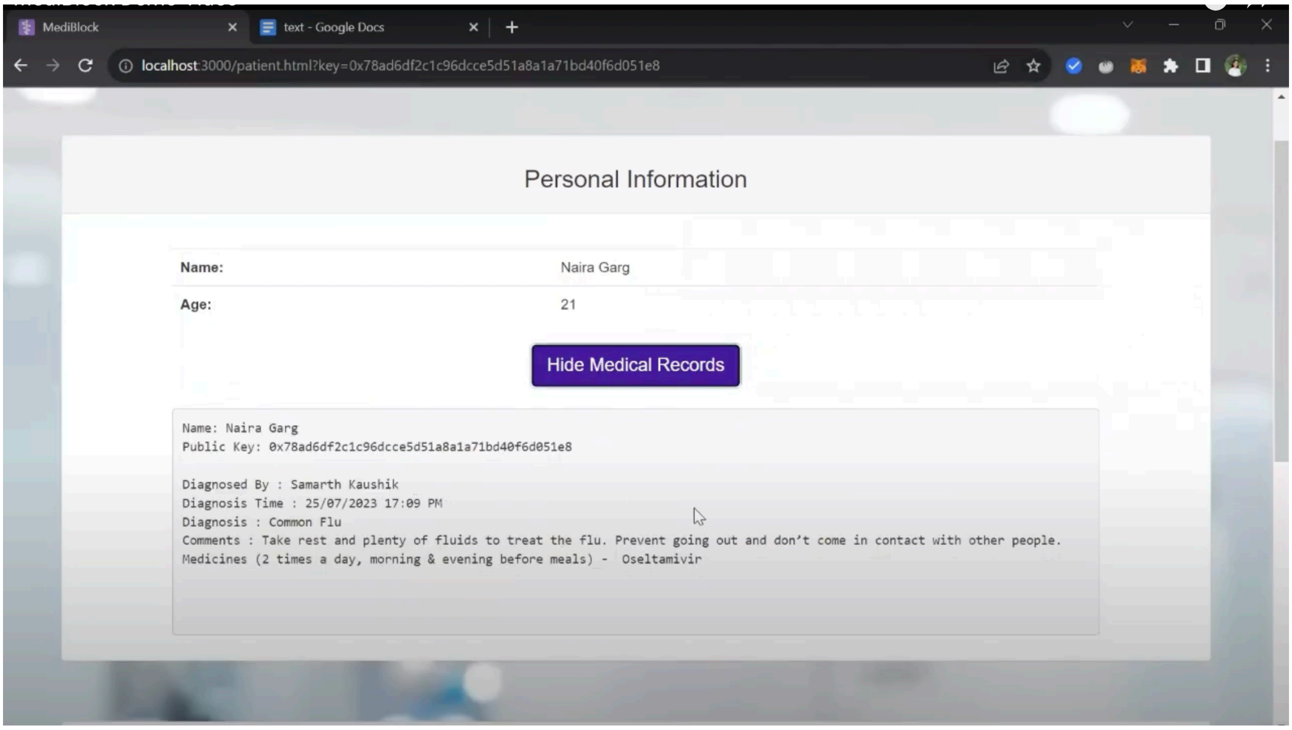

Figure 15: View of patient medical records prescribed by the doctor

4. Process to Get Access to the Proposed System (Back-End Part)

4.1. Deploying Transaction Using Ethereum Blockchain

Figure 16 illustrates the personal Ethereum blockchain, represented by Ganache. It has been used for testing and deployment of the system. Ganache offers some virtual accounts with 100 ETH for testing and local development. It provides a comparable capability to Ganache when deployed on the Ethereum main net. Some of the fake transactions are executed by virtual accounts on the application, together with transaction hashes and the contract address to which they were deployed. Each transaction's currency value is also displayed in a column.

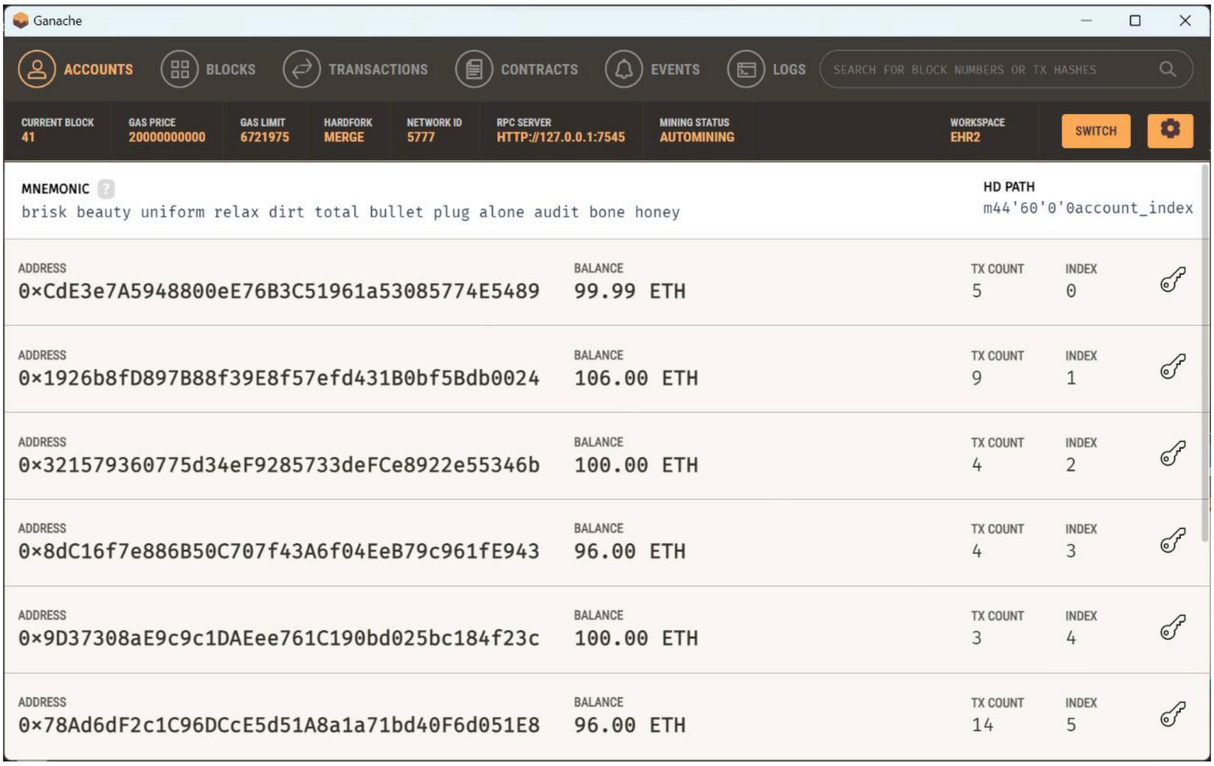

Figure 16: Deployment of the transaction log.

The first step on the back end is to download and install Ganache Ethereum from the Truffle Suite. The Truffle Suite is a world-class development environment for decentralised applications (apps) and smart contracts on the blockchain. After completing the Ganache installation, you must establish a new workspace named the ehr_project. Then, in the project directory, add Truffle from truffle-config.js. After that, this will show secure unique addresses, the index, balance, and corresponding private keys to create accounts for accessing the system. The Ganache will keep the accounts' privacy and serial numbers safe.

4.2. Connection to the Server Using Smart Contract

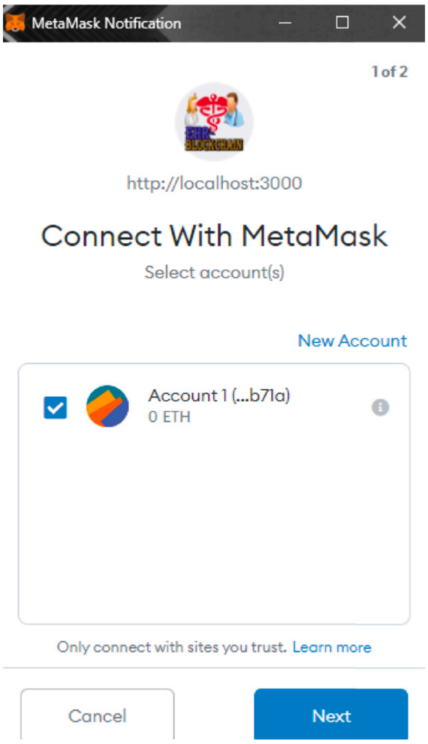

Figure: 17 Metamask wallet connection

Figure 17 depicts the system's connection via smart contracts. MetaMask is utilised as a smart contract to connect to the system. Contracts are created that provide metadata about record titles, access, and data integrity. Cryptographically signed instructions for controlling these characteristics are included in this system's blockchain transactions. Only legal transactions ensuring data alternation are used by the contract's state-transition functionalities to carry out policies. As long as a medical record can be stored electronically, these laws can be built to enforce any set of rules controlling it.

This smart contract, which is run on blockchain technology, might include all of the conditions, such as handling various permits and data access. A number of stakeholders are involved in this scheme, each performing different tasks. This will make it easier for doctors and patients to communicate. Smart contracts include data authorisation rules. It can also assist in tracing all activities associated with a unique ID from the point of origin to the point of submission. Different situations have been created and explained, as well as all of the functions and procedures that are included in the smart contracts.

The function can be supervised and approved directly through the smart contract. There is no need for a centralised authority to do so, which significantly decreases the administration's cost. All medical record data are saved in local database storage to enhance efficiency.

4.3. Truffle migration and deployment to compile and execute the contracts

Figure 18 illustrates a snapshot of the implementation in which a Truffle migrates and deploys on a blockchain with a smart contract and the execution on the Ethereum network. Truffle migrations allow you to upload smart contracts to the Ethereum blockchain (local), set up the essential procedures for integrating transactions with other transactions, and provide contracts with initial data.

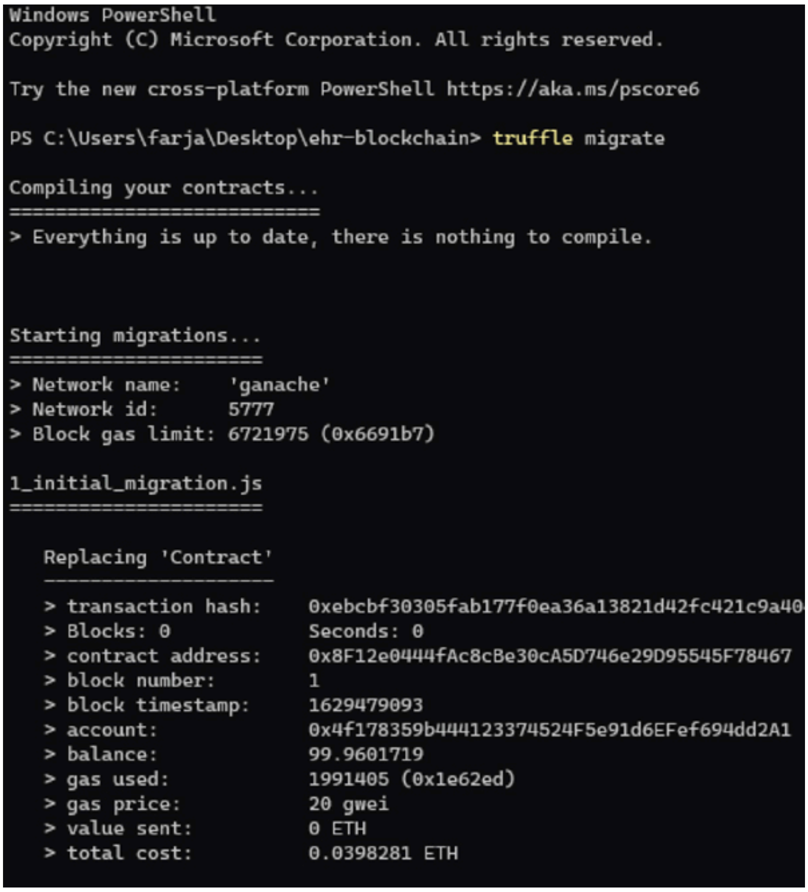

Figure: 18 Truffle migration and deployment and smart contract execution

Users also need to require the new contract and add a deployer statement inside the function if they want to deploy another contract from the same migration file. Two different smart contracts will be deployed as a result of this migration. Moreover, this process executes the smart contract for the system.

CONCLUSION

In conclusion, developing and deploying a blockchain-based Electronic Health Record (EHR) system represents a technological advancement and a transformative paradigm shift in healthcare management. This report has delved into intricate details, presenting a comprehensive blueprint of the proposed system, its underlying methodologies, and the potential it holds to revolutionise the healthcare landscape.

The cornerstone of the proposed framework lies in its utilisation of the Ethereum blockchain network, an innovative and decentralised platform that brings unparalleled security, data immutability, and transparency to health data management. Through the implementation of smart contracts and robust cryptographic algorithms, the system ensures that patient records are not only impervious to unauthorised access but also tamper-proof, thereby safeguarding patient privacy and trust.

The user-centric design, as highlighted through the role-based user application, serves as a testament to the system's commitment to optimising user experience. Doctors and patients alike are granted access to tailored interfaces, enabling seamless interaction with health records, appointments, and vital medical information. Such an approach empowers patients to take an active role in their healthcare journey and fosters efficient communication and collaboration between healthcare providers and their patients.

The integration of cloud services within the architecture further elevates the system's capabilities. The EHR administration system, hosted on the cloud, orchestrates the flow of transactions, ensuring accurate and swift processing while providing a centralised point for system management. Simultaneously, the cloud-based data storage component ensures the safekeeping of extensive medical records, accessible at the click of a button, thereby minimising administrative bottlenecks and enhancing patient care.

The proposed system's intrinsic potential extends far beyond its immediate impact. It has the capability to foster a broader cultural change in the healthcare domain, where data security, interoperability, and patient-centricity take centre stage. The benefits are multifaceted – from reduced administrative overhead and streamlined data sharing between healthcare institutions to improved patient outcomes and an empowered patient population.

In this dynamic era of technology-driven innovation, the blockchain-based EHR system stands as a beacon of hope and progress. It offers a tangible solution to the longstanding challenges faced by the healthcare sector, transcending geographical boundaries and bureaucratic barriers. As this report has expounded upon the intricate details of the system's architecture, protocol layout, and process flow, it is clear that the proposed framework is not merely a theoretical concept but a viable and promising avenue for reshaping healthcare management.

In conclusion, the blockchain-based EHR system has the potential to be a transformative force, ushering in a new era of healthcare data management characterised by security, transparency, and patient empowerment. As technology continues its rapid evolution, embracing forward-thinking solutions such as the one outlined in this report holds the key to unlocking a brighter, more efficient, and patient-centric future for healthcare worldwide.

FUTURE SCOPE

The proposed blockchain-based Electronic Health Record (EHR) system lays a strong foundation for revolutionising healthcare data management. However, its potential extends well beyond its current implementation, opening up a realm of exciting future possibilities and avenues for exploration. The following outlines the potential future scope of the system: Absolutely, incorporating these features will significantly enhance the capabilities and usability of your blockchain-based Electronic Health Record (EHR) system. Let's explore how each of these features can contribute to the system's functionality and value:

1. Chatbot Integration:

Integrating a chatbot within the EHR system can provide users with real-time assistance and support. Patients and doctors can interact with the chatbot to get answers to common queries, schedule appointments, receive medication reminders, and even access basic medical information. The chatbot can also assist with guiding users through the system's various features and providing a more user-friendly experience.

2. Mobile App Integration:

Developing a dedicated mobile app for the EHR system allows users to access their health records and manage appointments on the go. The app can provide secure authentication, push notifications for appointment reminders and updates, and a user-friendly interface optimized for mobile devices. This extends the accessibility of the system and enhances user engagement.

3. Patient-Uploaded Medical Records:

Allowing patients to upload their previous medical records directly into the system ensures a comprehensive and accurate health history. This feature empowers patients to take ownership of their health data and contributes to a more holistic view of their medical journey. Robust data validation and privacy measures will be essential to ensure the integrity and security of the uploaded records.

4. Cross-Hospital Data Sharing:

Enabling cross-hospital data sharing fosters collaboration among different healthcare providers. With patient consent, authorised doctors from different hospitals can access relevant medical records, enabling better-informed treatment decisions. Smart contracts can facilitate secure and auditable data sharing while maintaining patient privacy and data security.

Each of these features requires careful consideration, design, and implementation to ensure they seamlessly integrate into the existing EHR system. As you expand your system with these features, keep in mind the following aspects:

- **User Experience:** Prioritise a user-centric design to ensure that the added features are intuitive and easy to use for both patients and healthcare professionals.

- **Security and Privacy:** Implement robust security measures, encryption, and access controls to safeguard sensitive patient data. Compliance with data protection regulations is crucial.

- **Scalability:** Design the system architecture to accommodate increased user activity and data volume, especially as more patients and doctors adopt the system.

- **Testing and Feedback:** Thoroughly test the new features in a controlled environment before deploying them to the live system. Gather feedback from users to make iterative improvements.

By incorporating these features, your blockchain-based EHR system can offer a comprehensive, user-friendly, and secure platform that addresses the evolving needs of patients and healthcare providers while contributing to the advancement of healthcare data management.

ANNEXURE

GITHUB REPOSITORY LINK: CLICK HERE

a) Annexure 1

1) SMART CONTRACTS

```solidity
pragma solidity ^0.5.1;

contract Agent {
    struct patient {
        string name;
        uint age;
        address[] doctorAccessList;
        uint[] diagnosis;
        string record;
    }

    struct doctor {
        string name;
        uint age;
        address[] patientAccessList;
    }

    uint creditPool;

    address[] public patientList;
    address[] public doctorList;

    mapping(address => patient) patientInfo;
    mapping(address => doctor) doctorInfo;
    mapping(address => address) Empty;
    // might not be necessary
    mapping(address => string) patientRecords;

    function add_agent(
        string memory _name,
        uint _age,
        uint _designation,
        string memory _hash
    ) public returns (string memory) {
        address addr = msg.sender;
```

```solidity

        if (_designation == 0) {
            patient memory p;
            p.name = _name;
            p.age = _age;
            p.record = _hash;
            patientInfo[msg.sender] = p;
            patientList.push(addr) - 1;
            return _name;
        } else if (_designation == 1) {
            doctorInfo[addr].name = _name;
            doctorInfo[addr].age = _age;
            doctorList.push(addr) - 1;
            return _name;
        } else {
            revert();
        }
    }

    function get_patient(
        address addr
    )
        public
        view
        returns (string memory, uint, uint[] memory, address, string memory)
    {
        // if(keccak256(patientInfo[addr].name) == keccak256(""))revert();
        return (
            patientInfo[addr].name,
            patientInfo[addr].age,
            patientInfo[addr].diagnosis,
            Empty[addr],
            patientInfo[addr].record
        );
    }
```

```solidity

    function get_doctor(
        address addr
    ) public view returns (string memory, uint) {
        return (doctorInfo[addr].name, doctorInfo[addr].age);
    }

    function get_patient_doctor_name(
        address paddr,
        address daddr
    ) public view returns (string memory, string memory) {
        return (patientInfo[paddr].name, doctorInfo[daddr].name);
    }

    function permit_access(address addr) public payable {
        require(msg.value == 2 ether);

        creditPool += 2;

        doctorInfo[addr].patientAccessList.push(msg.sender) - 1;
        patientInfo[msg.sender].doctorAccessList.push(addr) - 1;
    }

    //must be called by doctor
    function insurance_claim(
        address paddr,
        uint _diagnosis,
        string memory _hash
    ) public {
        bool patientFound = false;
        for (
            uint i = 0;
            i < doctorInfo[msg.sender].patientAccessList.length;
            i++
        ) {
            if (doctorInfo[msg.sender].patientAccessList[i] == paddr) {
```

```solidity
                creditPool -= 2;
                patientFound = true;
            }
        }
        if (patientFound == true) {
            set_hash(paddr, _hash);
            remove_patient(paddr, msg.sender);
        } else {
            revert();
        }

        bool DiagnosisFound = false;
        for (uint j = 0; j < patientInfo[paddr].diagnosis.length; j++) {
            if (patientInfo[paddr].diagnosis[j] == _diagnosis)
                DiagnosisFound = true;
        }
    }

    function remove_element_in_array(
        address[] storage Array,
        address addr
    ) internal returns (uint) {
        bool check = false;
        uint del_index = 0;
        for (uint i = 0; i < Array.length; i++) {
            if (Array[i] == addr) {
                check = true;
                del_index = i;
            }
        }
        if (!check) revert();
        else {
            if (Array.length == 1) {
                delete Array[del_index];
            } else {
                Array[del_index] = Array[Array.length - 1];
```

```
    function remove_patient(address paddr, address daddr) public {
        remove_element_in_array(doctorInfo[daddr].patientAccessList, paddr);
        remove_element_in_array(patientInfo[paddr].doctorAccessList, daddr);
    }

    function get_accessed_doctorlist_for_patient(
        address addr
    ) public view returns (address[] memory) {
        address[] storage doctoraddr = patientInfo[addr].doctorAccessList;
        return doctoraddr;
    }

    function get_accessed_patientlist_for_doctor(
        address addr
    ) public view returns (address[] memory) {
        return doctorInfo[addr].patientAccessList;
    }

    function revoke_access(address daddr) public payable {
        remove_patient(msg.sender, daddr);
        msg.sender.transfer(2 ether);
        creditPool -= 2;
    }

    function get_patient_list() public view returns (address[] memory) {
        return patientList;
    }

    function get_doctor_list() public view returns (address[] memory) {
        return doctorList;
    }

    function get_hash(address paddr) public view returns (string memory) {
        return patientInfo[paddr].record;
    }

    function set_hash(address paddr, string memory _hash) internal {
        patientInfo[paddr].record = _hash;
    }
}

```

Migrations.sol 1 ×

final_work > app > contracts > Migrations.sol

```
pragma solidity ^0.5.1;

contract Migrations {
  address public owner;
  uint public last_completed_migration;

  modifier restricted() {
    if (msg.sender == owner) _;
  }

  constructor() public {
    owner = msg.sender;
  }

  function setCompleted(uint completed) public restricted {
    last_completed_migration = completed;
  }

  function upgrade(address new_address) public restricted {
    Migrations upgraded = Migrations(new_address);
    upgraded.setCompleted(last_completed_migration);
  }
}

```

2) JAVASCRIPT CODE (Only the backend frontend integration part is shown)

app.js

final_work > app > src > js > app.js > ...

```javascript

// if (typeof web3 !== 'undefined') {
//     web3 = new Web3(web3.currentProvider);
// } else {
//     // set the provider you want from Web3.providers
// web3 = new Web3(new Web3.providers.HttpProvider("http://127.0.0.1:8545"));
// }

var web3;

var agentContractAddress = '0x855DB599286649e4ffb6965821D4c0D8Ad64F844';

function connect() {
    web3 = new Web3(window.ethereum)
    window.ethereum.enable().catch(error => {
        // User denied account access
        console.log(error);
    })
    abi = JSON.parse('[{"constant":true,"inputs":[{"name":"addr","type":"addr
    AgentContract = web3.eth.contract(abi);
    contractInstance = AgentContract.at(agentContractAddress);
    web3.eth.defaultAccount = web3.currentProvider.selectedAddress;
    console.log("Web3 Connected:" + web3.eth.defaultAccount);
    return web3.currentProvider.selectedAddress;
}

window.addEventListener('load', async () => {
    // New web3 provider
    connect();
    console.log("Externally Loaded!");
});
```

3) CSS CODE (only the initial code is shown)

bootstrap.css 4 ×

final_work > app > src > css > bootstrap.css > ...

```css
/*!
 * Bootstrap v3.3.7 (http://getbootstrap.com)
 * Copyright 2011-2016 Twitter, Inc.
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)
 */
/*! normalize.css v3.0.3 | MIT License | github.com/necolas/normalize.css */
html {
  font-family: sans-serif;
  -webkit-text-size-adjust: 100%;
  -ms-text-size-adjust: 100%;
}

body {
  margin: 0;
}

article,
aside,
details,
figcaption,
figure,
footer,
header,
hgroup,
main,
menu,
nav,
section,
summary {
  display: block;
}

audio,
canvas,
progress,
```

bootstrap.css 4 ×

final_work > app > src > css > bootstrap.css > ...

```css
  background-color: transparent;
}

:active,
:hover {
  outline: 0;
}

bbr[title] {
  border-bottom: 1px dotted;
}

,
trong {
  font-weight: bold;
}

fn {
  font-style: italic;
}

1 {
  margin: .67em 0;
  font-size: 2em;
}

ark {
  color: #000;
  background: #ff0;
}

mall {
  font-size: 80%;
}
```

```css
}

svg:not(:root) {
  overflow: hidden;
}

figure {
  margin: 1em 40px;
}

hr {
  height: 0;
  -webkit-box-sizing: content-box;
  -moz-box-sizing: content-box;
  box-sizing: content-box;
}

pre {
  overflow: auto;
}

code,
kbd,
pre,
samp {
  font-family: monospace, monospace;
  font-size: 1em;
}

button,
input,
optgroup,
select,
textarea {
  margin: 0;
```

```css
html input[type="button"],
input[type="reset"],
input[type="submit"] {
  -webkit-appearance: button;
  cursor: pointer;
}

button[disabled],
html input[disabled] {
  cursor: default;
}

button::-moz-focus-inner,
input::-moz-focus-inner {
  padding: 0;
  border: 0;
}

input {
  line-height: normal;
}

input[type="checkbox"],
input[type="radio"] {
  -webkit-box-sizing: border-box;
  -moz-box-sizing: border-box;
  box-sizing: border-box;
  padding: 0;
}

input[type="number"]::-webkit-inner-spin-button,
input[type="number"]::-webkit-outer-spin-button {
  height: auto;
}
```

```css
fieldset {
  padding: .35em .625em .75em;
  margin: 0 2px;
  border: 1px solid #c0c0c0;
}

legend {
  padding: 0;
  border: 0;
}

textarea {
  overflow: auto;
}

optgroup {
  font-weight: bold;
}

table {
  border-spacing: 0;
  border-collapse: collapse;
}

td,
th {
  padding: 0;
}

/*! Source: https://github.com/h5bp/html5-boilerplate/blob/master/src/css/ma
@media print {
  *,
  *:before,
```

```css
  }

  h2,
  h3 {
    page-break-after: avoid;
  }

  .navbar {
    display: none;
  }

  .btn>.caret,
  .dropup>.btn>.caret {
    border-top-color: #000 !important;
  }

  .label {
    border: 1px solid #000;
  }

  .table {
    border-collapse: collapse !important;
  }

  .table td,
  .table th {
    background-color: #fff !important;
  }

  .table-bordered th,
  .table-bordered td {
    border: 1px solid #ddd !important;
  }
}
```

4) HTML CODE (only the initial code is shown)

- **HOMEPAGE**

index.html

final_work > app > src > index.html > html > head > meta

```html
<!DOCTYPE html>
<html lang="en">

<head>

    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="">
    <meta name="author" content="">

    <title>MediBlock</title>

    <!-- Bootstrap Core CSS -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <link rel="icon" type="image/x-icon" href="favicon.ico" />

    <script src="js/bundle.js"></script>


    <!-- Custom CSS -->
    <style>
        body {
            background-image: url("homepagecrop.jpg");
            /* background-size: cover; */
            background-repeat: no-repeat;

        }

        .panel-heading {
            margin-bottom: 5px;
        }

        .container {
            margin-top: 325px;
```

```html
            margin-left: 20px;
            max-width: 50%;
            max-height: 1%;
        }

        .navbar {
            margin-bottom: 70px;
            background-color: rgb(59, 13, 144);
        }

        a.navbar-brand {
            font-size: 100%;
        }

        .aa {
            align-items: center;
            font-size: 15px;
            margin-left: 40px;
        }
    </style>

</head>

<body>

    <!-- Navigation -->
    <nav class="navbar navbar-inverse navbar-static-top" role="navigation">
        <div class="container-fluid">
            <!-- Brand and toggle get grouped for better mobile display -->
            <div class="navbar-header">
                <button type="button" class="navbar-toggle" data-toggle="coll
                    data-target="#bs-example-navbar-collapse-1">
                    <span class="sr-only">Toggle navigation</span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
```

```html
                <a class="navbar-brand" href="#"></a>
            </div>
            <!-- Collect the nav links, forms, and other content for toggling
            <div class="collapse navbar-collapse" id="bs-example-navbar-colla
                <ul class="nav navbar-nav navbar-right">
                    <li class="active">
                        <a style="font-size: 100%; background-color:  rgb(5
                    </li>
                    <li>
                        <a style="font-size: 100%;background-color: rgb(59,
                            href="./register.html">Register</a>
                    </li>
                </ul>
            </div>
            <!-- /.navbar-collapse -->
        </div>
        <!-- /.container -->
    </nav>

    <div>
        <div id="includedContent"></div>
    </div>


    <!-- Page Content -->
    <div class="container">
        <div class="panel panel-default">
            <div class="panel-heading">
                <h4 class="text-center">Login is Linked to your Metamask Acco
            </div>
            <div class="panel-body">
                <div class="row">
                    <div style="background-color: lightblue; color: rgb(1
                        class="alert alert-warning col-sm-8 col-sm-offset-2">
                        <p class="aa"> <strong>Unregistered user. </strong>Cl
```

```html

            </div>
            <div class="panel-footer">
            </div>
        </div>


    </div>
    <!-- /.container -->

    <!-- jQuery Version 1.11.1 -->
    <script src="js/jquery.js"></script>

    <!-- Bootstrap Core JavaScript -->
    <script src="js/bootstrap.min.js"></script>
    <script src="/js/web3.min.js"></script>
    <script src="js/app.js"></script>
    <script>
        $(function () {
            $("#includedContent").load("anim.html");
        });

        connect();
        function login() {
            $(".alert-warning").hide();

            publicKey = web3.currentProvider.selectedAddress;
            console.log(publicKey);
            contractInstance.get_patient_list(function (error, result) {
                if (!error) {
                    var PatientList = result;
                    for (var i = 0; i < PatientList.length; i++) {
                        if (publicKey.toLowerCase() == PatientList[i]) {
```

- **DOCTOR'S PORTAL**

```html
<!DOCTYPE html>
<html lang="en">

<head>

    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="">
    <meta name="author" content="">

    <title>MediBlock</title>

    <!-- Bootstrap Core CSS -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <link rel="icon" type="image/x-icon" href="favicon.ico" />
    <!-- <script src="js/bundle.js"></script> -->
    <!-- Custom CSS -->
    <style>
        body {
            background-image: url("hos_bg2.avif");
            background-size: cover;
            background-repeat: no-repeat;

            /* padding-top: 70px; */
            /* Required padding for .navbar-fixed-top. Remove if using .navb
        }

        .panel {
            margin-bottom: 60px;
        }

        .navbar {
            margin-bottom: 70px;
            background-color: rgba(59, 13, 144, 1);
```

```html
        .navbar {
            margin-bottom: 70px;
            background-color: rgba(59, 13, 144, 1);
        }

        .panel-heading {
            margin-bottom: 20px;
        }

        .nav-pills>li>a {
            padding: 0;
            padding-right: 10px;
        }

        .nav-pills>li>a:hover {
            background-color: initial;
        }

        .nav-pills>li.active>a {
            color: #23527c;
            background-color: initial;
        }

        .nav-pills>li.active>a:hover,
        .nav-pills>li.active>a:focus {
            color: #23527c;
            background-color: inherit;
        }
    </style>


</head>

<body>
```

```html
<body>

    <!-- Navigation -->
    <nav class="navbar navbar-inverse navbar-static-top" role="navigation">
        <div class="container-fluid">
            <!-- Brand and toggle get grouped for better mobile display -->
            <div class="navbar-header">
                <button type="button" class="navbar-toggle" data-toggle="colla
                    data-target="#bs-example-navbar-collapse-1">
                    <span class="sr-only">Toggle navigation</span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                <a style="color: white;" class="navbar-brand" href="#">Doct
            </div>
            <!-- Collect the nav links, forms, and other content for toggling
            <div class="collapse navbar-collapse" id="bs-example-navbar-colla
                <ul class="nav navbar-nav navbar-right">
                    <li>
                        <a style="color: white;" href="./index.html">Logout
                    </li>
                </ul>
            </div>
            <!-- /.navbar-collapse -->
        </div>
        <!-- /.container -->
    </nav>

    <div class="container">
        <div class="panel panel-default">
            <div class="panel-heading">
                <h3 class="text-center">Personal Information</h3>
            </div>
            <div class="panel-body">
```

```html
            <div class="panel-body">
                <div class="row">
                    <div class="col-sm-offset-1 col-sm-10">
                        <table class="table">
                            <tr>
                                <th>Name:</th>
                                <td id="name"></td>
                            </tr>
                            <tr>
                                <th>Age:</th>
                                <td id="age"></td>
                            </tr>
                        </table>
                    </div>
                </div>

            </div>
        </div>


        <div class="panel panel-default">
            <div class="panel-heading">
                <h3 class="text-center">Accessible EMRs</h3>
            </div>
            <div class="panel-body">
                <div class="row">
                    <div class="alert alert-danger col-sm-8 col-sm-offset-2">
                        <strong>Notice!</strong> Could not access records. Acc
                        or patient.
                    </div>
                </div>
                <div class="row">
                    <div class="col-sm-offset-1 col-sm-10">
                        <table id="viewPatient" class="table table-hover">
                            <tr>
```

- **PATIENT'S PORTAL**

```html
<!DOCTYPE html>
<html lang="en">

<head>

    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="">
    <meta name="author" content="">

    <title>MediBlock</title>

    <!-- Bootstrap Core CSS -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <link rel="icon" type="image/x-icon" href="favicon.ico" />
    <!-- <script src="js/bundle.js"></script> -->

    <!-- Custom CSS -->
    <style>
        body {
            background-image: url("hos_bg2.avif");
            background-size: cover;
            background-repeat: no-repeat;

            /* padding-top: 70px; */
            /* Required padding for .navbar-fixed-top. Remove if using .navba
        }

        .navbar {
            margin-bottom: 70px;
            background-color: rgba(59, 13, 144, 1);
        }

        .panel-heading {
```

```css
            margin-bottom: 20px;
        }

        .panel {
            margin-bottom: 60px;
        }

        /* .publicKeyDoctor{
        display: none;
    } */
        .well>h3 {
            margin: 10px auto;
        }

        .checkbox label:after {
            content: '';
            display: table;
            clear: both;
        }

        .checkbox .cr {
            position: relative;
            display: inline-block;
            border: 1px solid #a9a9a9;
            border-radius: .25em;
            width: 1.3em;
            height: 1.3em;

            margin-right: .5em;
        }

        .checkbox .cr .cr-icon {
            position: absolute;
            font-size: .8em;
            line-height: 0;
```

```html
            line-height: 0;
            top: 50%;
            left: 15%;
        }

        .checkbox label input[type="checkbox"] {
            display: none;
        }

        .checkbox label input[type="checkbox"]+.cr>.cr-icon {
            opacity: 0;
        }

        .checkbox label input[type="checkbox"]:checked+.cr>.cr-icon {
            opacity: 1;
        }
    </style>

</head>

<body>

    <!-- Navigation -->
    <nav class="navbar navbar-inverse navbar-static-top" role="navigation">
        <div class="container-fluid">
            <!-- Brand and toggle get grouped for better mobile display -->
            <div class="navbar-header">
                <button type="button" class="navbar-toggle" data-toggle="coll
                    data-target="#bs-example-navbar-collapse-1">
                    <span class="sr-only">Toggle navigation</span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                <a style="color: white;" class="navbar-brand" href="#">Pati
```

```html
            <!-- Collect the nav links, forms, and other content for toggling
            <div class="collapse navbar-collapse" id="bs-example-navbar-colla
                <ul class="nav navbar-nav navbar-right">
                    <li>
                        <a style="color: white;" href="./index.html">Logout
                    </li>
                </ul>
            </div>
            <!-- /.navbar-collapse -->
        </div>
        <!-- /.container -->
    </nav>

    <div class="container">
        <div class="panel panel-default">
            <div class="panel-heading">
                <h3 class="text-center">Personal Information</h3>
            </div>
            <div class="panel-body">
                <div class="row">
                    <div class="col-sm-offset-1 col-sm-10">
                        <table class="table">
                            <tr>
                                <th>Name:</th>
                                <td id="name"></td>
                            </tr>
                            <tr>
                                <th>Age:</th>
                                <td id="age"></td>
                            </tr>
                        </table>

                        <div class="text-center">
                            <!-- <h5>Your records are stored here: http://loca
                            </h5> -->
```

REFERENCES

https://www.youtube.com/watch?v=gyMwXuJrbJQ

https://www.youtube.com/watch?v=Wn_Kb3MR_cU

https://www.youtube.com/watch?v=fghqq3-P3x0

https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0266916

https://www.hindawi.com/journals/js/2022/7299185/

https://github.com/IBM/Medical-Blockchain

https://ieeexplore.ieee.org/document/8433583

https://www.researchgate.net/publication/336773543_Single-molecule_imaging_of_the_biophysics_of_molecular_interactions_with_precision_and_control_in_cell-like_conditions_and_without_tethers

https://www.researchgate.net/publication/344065489_Current_topics_in_emergency_trauma_care_part_2_resuscitation_in_trauma

https://www.mdpi.com/2079-9292/9/10/1609