# Table of Contents

# Abstract

Storing large volumes of medical data using centralized, cloud-based architectures presents several challenges, including data ownership disputes, limited interoperability with other platforms, vulnerability to security a acks, and privacy concerns. These issues undermine the reliability, scalability, and security of current Electronic Health Record (EHR) systems, leaving pa ents and healthcare providers at risk of data breaches and unauthorized access.

This project introduces Medical-Chain, a decentralized framework designed to securely store and share sensitive medical data. It leverages the Interplanetary File System (IPFS) for distributed, encrypted storage of health records, ensuring enhanced security and privacy. Access to these records is managed through Smart Contracts on the Ethereum blockchain, giving pa ents full control over who can view and use their data. This approach eliminates reliance on centralized intermediaries, ensuring enhanced privacy, security, and data ownership.

Additionally, the system features a personalized dashboard for real-me monitoring of network performance, including key security metrics such as network hash rate and resistance to a acks. The solution is validated through performance benchmarking across metrics like latency, throughput, scalability, and usability. By comparing these results with existing frameworks, Medical Chain proves to be a scalable, secure, and efficient alternative for managing EHRs in a decentralized manner.

# List of Figures

# CHAPTER 1

# INTRODUCTION

## 1.1 Motivation:

The project addresses the critical issue of **secure and efficient sharing of Electronic Health Records (EHRs)**, a challenge due to the sensitive nature of medical data and the increasing need for privacy-preserving solutions. Traditional centralized systems are prone to data breaches, unauthorized access, and inefficiencies, which compromise patient confidentiality and trust.

This project leverages **blockchain technology and smart contracts** to create a decentralized system for managing and sharing medical records. Blockchain ensures that data integrity, privacy, and security are maintained while enabling authorized users to access records seamlessly. Additionally, by integrating decentralized storage solutions like IPFS and Pinata, the project overcomes the limitations of centralized data storage, ensuring that medical records are tamper-proof and easily retrievable.

Healthcare data breaches have been a growing concern, with reports indicating that over 50 million patient records were exposed in 2021 alone. According to a study by the Ponemon Institute, the average cost of a healthcare data breach reached $10.1 million in 2022, making it imperative to adopt innovative solutions like blockchain to address these vulnerabilities.

This project not only aims to mitigate these risks but also empowers patients with control over their medical data, fostering transparency and trust within the healthcare ecosystem.

## 1.2 What is Blockchain?

Blockchain is a decentralized, distributed digital ledger technology that records transactions across many computers in such a way that the data is immutable and secure. It is the underlying technology behind cryptocurrencies like Bitcoin, but its applications extend beyond digital currencies.

Each block has **three** essential parts:
**Data**: The actual information (such as transaction details) being recorded.
**Hash**: A unique identifier for the block, like a digital fingerprint.
**Previous Hash**: The hash of the previous block in the chain, linking them together.

## 1.3 What is IPFS?

The Inter Planetary File System (IPFS) is a decentralized, peer-to-peer file-sharing protocol that aims to make the web faster, safer, and more open. Unlike traditional systems that rely on central servers, IPFS stores files by splitting them into smaller pieces and distributing them across a network of nodes. Each file is identified by a unique cryptographic hash based on its content, making retrieval more secure and efficient. This content-based addressing allows files to be accessed from multiple nodes, enhancing redundancy and availability. IPFS is especially useful for sharing large datasets, preserving digital content, and building decentralized applications (dApps).

### 1.4 Key Features of Blockchain-Based Medical

1. **Decentralized Data Storage**: Medical records are stored across a distributed network (e.g., IPFS/Pinata), eliminating the need for a central database and reducing the risk of a single point of failure.

2. **Preservation**: Sensitive patient data is encrypted before storage, ensuring that only authorized users with decryption keys can access it. Patients retain full control over who can view their data.

3. **Role-Based Access Control**: Smart contracts enforce permissions based on user roles (Admin, Doctor, Patient), ensuring that only authorized individuals can perform specific actions.

4. **Secure File Sharing**: Encrypted files are shared seamlessly among stakeholders while maintaining security through end-to-end encryption and private key mechanisms.

5. **Cost-Effective and Efficient**: By eliminating intermediaries and automating processes with smart contracts, the system reduces operational costs and speeds up workflows.

6. **Data Integrity and Immutability:** Blockchain ensures that medical records cannot be altered or tampered with, providing a transparent and trustworthy system for data sharing.

### 1.5 Objectives of the project

1. **Decentralized Data Management:** Develop a blockchain-based system for securely storing and sharing Electronic Health Records (EHRs) using decentralized storage solutions like IPFS, ensuring data integrity and availability without relying on a central authority.
2. **Privacy Preservation:** Implement robust encryption techniques to ensure that sensitive medical data remains secure and accessible only to authorized users, maintaining patient confidentiality**.**
3. **Efficient Data Sharing**: Facilitate seamless and secure sharing of medical records between stakeholders, optimizing the process for both speed and reliability while maintaining security standards.
4. **Tamper-Proof Record Keeping**: Leverage blockchain's immutability to create a transparent, auditable ledger of all interactions, ensuring accountability and trust among stakeholders.
5. **Efficient Data Sharing**: Facilitate seamless and secure sharing of medical records between stakeholders, optimizing the process for both speed and reliability while maintaining security standards.
6. **Scalability and Interoperability**: Develop a system architecture capable of supporting a growing number of users and integrating with existing healthcare standards and platforms.

### 1.6 Organisation of the report

**Chapter 1**: Introduction provides an overview of the project, highlighting the motivation, objectives, and significance of using blockchain technology for secure and efficient medical data sharing.

**Chapter 2: Literature Survey** provides an overview of the theoretical background of blockchain technology and its application in medical data management, along with a discussion on related work in the domain.

**Chapter 3: Requirements to Develop the Project** outlines the prerequisites for developing the blockchain-based medical data sharing system, including the functional and non-functional requirements, architectural framework, and flow of data processing.

**Chapter 4: Proposed Solution and System Architecture** presents the design and architecture of the system, including an explanation of the key components such as blockchain, smart contracts, and decentralized storage, and their interaction.

**Chapter 5: System Analysis** elaborates on the detailed analysis of the proposed solution, focusing on aspects such as role-based access control, data encryption, and secure file sharing.

**Chapter 6: Tools and Technologies Used** describes the tools, frameworks, and programming environments employed in the project, such as Ethereum, IPFS, React, and Solidity.

**Chapter 7: Results** discusses the evaluation outcomes, including performance metrics such as system scalability, data security, and efficiency of file retrieval and sharing.

**Chapter 8: Applications** explores the practical use cases and scenarios where the blockchain-based medical data sharing system can be deployed, such as in hospitals, research collaborations, and health data exchanges.

**Chapter 9: Future Work** highlights potential enhancements and avenues for further research, such as incorporating AI, improving scalability, and integrating with other healthcare systems.

**Chapter 10: Conclusion** summarizes the findings, draws conclusions, and emphasizes the significance of the project in addressing the challenges of secure and efficient medical data sharing.

# CHAPTER 2
# LITERATURE SURVEY

This chapter reviews existing research on blockchain technology and its application in secure medical data sharing, highlighting the current challenges and gaps in achieving privacy preservation, data integrity, and efficient data management.

## 1. Implementing Blockchain for Secure EHR Management

**Overview:** This article discusses a case study on implementing blockchain technology to enhance the security and accessibility of electronic health records.

**Key Article**: A. S. Khan, B. A. Patel, and C. R. Greene, "Implementing blockchain for secure management of electronic health records: A case study," International Journal of Medical Informatics, vol. 150, pp. 104-112, 2021.

**Summary:** The authors describe a blockchain-based EHR system implemented in a healthcare facility, detailing how it achieves data integrity, enhanced access control, and security. The blockchain network structure supports data immutability and is paired with cryptographic protocols to manage permissions. Improved data accuracy, reduction in data tampering, and increased trust among patients and practitioners are notable achievements reported. The article presents quantitative performance metrics like system latency, scalability, and throughput, along with user feedback demonstrating significant improvements in the system's reliability and efficiency.

## 2. A Blockchain-Based Framework for Health Data Sharing

**Overview:** This article presents a framework designed for secure and efficient sharing of health data among stakeholders using blockchain technology.

**Key Article:** J. M. Smith and L. R. Johnson, "A blockchain-based framework for health data sharing: Improving interoperability and security," Journal of Biomedical Informatics, vol. 120, pp. 103-115, 2020.

**Summary:** The authors propose a blockchain-based framework to allow healthcare providers to share patient data securely while adhering to privacy standards such as HIPAA. The framework leverages smart contracts to automate permission control and update logs, ensuring transparency and traceability in data sharing. The article details a pilot test, showing successful data exchange between multiple healthcare institutions with improved access and data management. Metrics such as system response time, scalability under load, and patient satisfaction are provided, indicating the framework's capacity to handle large volumes of requests efficiently.

## 3. Privacy-Enhancing Technologies in Health Data Sharing

**Overview:** This article explores the integration of privacy-enhancing technologies with blockchain to secure health data sharing processes.

**Key Article:** K. T. Lee and D. H. Kim, "Integrating privacy-enhancing technologies with blockchain for secure health data sharing," Health Informatics Journal, vol. 26, no. 3, pp. 215-230, 2021.

**Summary:** The study reviews privacy-preserving mechanisms like homomorphic encryption, zero-knowledge proofs, and secure multi-party computation alongside blockchain to safeguard sensitive health data. The authors analyze the performance of these methods, focusing on their effectiveness in protecting data privacy without compromising accessibility for authorized users. Detailed tests on computational load, encryption time, and data retrieval speed are presented, demonstrating that these technologies reduce the risk of data breaches and unauthorized access. Additionally, case studies illustrate how these techniques maintain compliance with legal standards while supporting seamless data access for medical providers.

**4. Smart Contracts for Automated Consent Management in Healthcare**

**Overview**: This article focuses on the use of smart contracts for managing patient consent in healthcare applications.

**Key Article:** M. R. T. Arora, E. N. Williams, and T. D. Wilson, "Using smart contracts for automated consent management in healthcare," Journal of Healthcare Engineering, vol. 2021, Article ID 985743, 2021.

**Summary:** This article presents a smart contract-based system to automate patient consent management, reducing administrative processes while empowering patients. It describes a permission-based blockchain structure that allows patients to grant or revoke access to their data autonomously. Results from a clinical pilot project suggest reduced consent processing time, improved patient satisfaction, and lower instances of unauthorized access. Detailed analysis of smart contract execution speed, gas costs, and error handling mechanisms is included, highlighting the practicality and robustness of the system in a healthcare context.

**5. Decentralized Storage Solutions for Healthcare Data Management**

**Overview**: This article discusses the use of decentralized storage solutions, particularly IPFS, for managing healthcare data securely.

**Key Article**: N. J. Patel and A. K. Roy, "Decentralized storage solutions for healthcare data management: A practical approach using IPFS," Health Information Science and Systems, vol. 9, no. 1, pp. 67-78, 2021.

**Summary**: The authors describe a healthcare data management system using IPFS for decentralized storage to improve data resilience and accessibility. Their approach

minimizes dependence on centralized databases, reducing data breach risks and enhancing data availability. Blockchain integration ensures data integrity, with IPFS handling large data sets effectively. They provide quantitative evaluations on storage efficiency, access latency, and data retrieval accuracy. Real-world applications in clinics showed an increase in patient data availability and protection against unauthorized access, indicating IPFS's viability for healthcare data storage.

### 6. A Blockchain-Based Framework for Health Data Sharing

**Overview:** This article presents a framework designed for secure and efficient sharing of health data among stakeholders using blockchain technology.

**Key Article:** J. M. Smith and L. R. Johnson, "A blockchain-based framework for health data sharing: Improving interoperability and security," Journal of Biomedical Informatics, vol. 120, pp. 103-115, 2020.

**Summary:** The proposed framework enables secure, interoperable data sharing among healthcare providers, addressing challenges in data privacy, accessibility, and integration across multiple platforms. Using a permissioned blockchain, it combines cryptographic signatures with smart contracts to regulate data access based on each provider's authorization. Smart contracts manage permissions dynamically, supporting real-time updates to records while keeping a transparent and immutable log of modifications. A pilot test conducted with multiple healthcare institutions yielded promising results, showing efficient data retrieval and access control, faster patient record exchange, and seamless interoperability across systems. Detailed performance metrics such as transaction speed, data access latency, and compliance with privacy standards like HIPAA demonstrate the framework's reliability.

## Issues with Traditional Approaches to Medical Data Sharing:

**Data Privacy Concerns**: Traditional healthcare systems often require centralized storage and management of sensitive patient data. This centralization raises significant privacy concerns, especially with the increasing incidents of data breaches and unauthorized access, putting patient confidentiality at risk. Strict data protection regulations (e.g., HIPAA) require that health data be handled with high security, but traditional systems often struggle to meet these standards effectively.

**Lack of Data Integrity**: Centralized medical data systems are vulnerable to unauthorized alterations or tampering. Without a transparent and immutable audit trail, ensuring the integrity of medical records becomes difficult, which can lead to mismanagement and mistrust in the system.

**Security Threats**: Centralized storage models expose medical data to high-security risks, including cyberattacks, hacking, and data breaches. These attacks can compromise

the confidentiality, integrity, and availability of sensitive medical records, threatening patient safety.

**Scalability Challenges**: Traditional healthcare systems are often unable to handle large volumes of patient data efficiently. As the volume of medical records grows, the centralized infrastructure faces limitations in terms of storage, computational capacity, and data processing, leading to performance bottlenecks.

## Blockchain as Solution:

Blockchain technology addresses the limitations of traditional medical data management systems by ensuring data privacy, security, and scalability. By decentralizing data storage and using cryptographic techniques, blockchain keeps patient data secure and private, with only authorized users able to access it.

**Data Privacy Preservation**: Blockchain enables the storage of encrypted medical records across a distributed network, ensuring that no single entity has access to all data. Patients have control over their records, and only authorized users can access them, which helps to meet privacy regulations like HIPAA.

**Immutable Data Integrity**: Blockchain guarantees that once a record is added to the ledger, it cannot be altered, ensuring the integrity of medical data. Every transaction is logged on the blockchain, creating a transparent, auditable record that helps to prevent fraud and unauthorized changes.

**Enhanced Security**: The decentralized nature of blockchain reduces the risk of a single point of failure. Since data is not stored in one central location, cyberattacks targeting a single server or database will not affect the entire system. Additionally, blockchain uses encryption and cryptographic hashing to secure data, ensuring that patient records remain confidential and protected from malicious actors.

**Scalable Infrastructure**: Blockchain's distributed architecture allows it to scale efficiently as the volume of data grows. By decentralizing data storage, the system avoids the bottlenecks associated with centralized servers, and each node in the network can independently process transactions, improving system performance.

# CHAPTER 3

# REQUIREMENTS TO DEVELOP THE PROJECT

This chapter outlines the system's **functional requirements**, detailing core operations such as secure storage, access control, and data sharing of medical records. It also describes **non-functional requirements**, emphasizing system performance, scalability, security, and privacy preservation. Additionally, the chapter highlights the **software tools and technologies** required for implementing the blockchain-based medical data sharing system, and identifies key **constraints**, such as resource limitations, network connectivity, and regulatory compliance (e.g., HIPAA). It also presents the **assumptions** made during the development of the project, such as the availability of reliable network infrastructure and participation from key stakeholders.

## 3.1 Functional Requirements:

### 3.1.1 Frontend:

The user interface will be developed using modern web development frameworks like **React**. These frameworks will provide a responsive and user-friendly interface for patients, doctors, and admins to interact with the system. The frontend will integrate **MetaMask** or similar wallet extensions for user authentication, enabling secure interactions with the Ethereum blockchain. The UI will display medical records and enable actions like record viewing, sharing, and updating with real-time access.

### 3.1.2 Backend:

The backend will leverage **blockchain technology**, primarily using **Web3.js** or **Ethers.js** to interact with the Ethereum network. **Solidity** smart contracts will manage access control, ensuring that only authorized users (doctors, patients, admins) can interact with the system and modify records. **IPFS** will be utilized for decentralized storage of medical records, ensuring data privacy and integrity. Development tools like **Hardhat** will be employed for smart contract development, testing, and deployment, ensuring seamless interaction between the frontend and backend.

## 3.2 Non-Functional Requirements:

#### 3.2.1 Performance:

The system should efficiently manage and share medical records across authorized users (doctors, patients, admins) within a reasonable time frame. This ensures prompt access to patient data, facilitating timely decision-making by healthcare professionals. The system must also maintain high performance during high volumes of data requests and interactions, ensuring smooth user experience.

### 3.2.2 Scalability:

The architecture must support the seamless addition of new clients (patients, doctors, healthcare providers) to the network without significant degradation in performance. As more participants join the decentralized medical data sharing system, it should handle the increased volume of requests, data storage, and transaction processing efficiently, ensuring continuous and smooth operation.

### 3.2.3 Security:

Given the sensitive nature of medical records, the system must implement robust security protocols to protect against unauthorized access, data breaches, and tampering. This includes encryption of medical data stored on decentralized storage (IPFS), secure communication channels between clients and the server, and strong access control measures through smart contracts. Blockchain technology will ensure data integrity and prevent unauthorized modifications.

### 3.2.4 Maintainability:

The codebase should be modular, well-documented, and follow best practices to facilitate easy updates and integration of new features or enhancements, such as incorporating additional access control mechanisms or improving the encryption algorithms. This will ensure that the system can adapt to new privacy regulations, evolving healthcare practices, or emerging security threats without requiring major overhauls.

### 3.2.5 Reliability:

The system must be highly reliable, consistently providing accurate access to medical records while ensuring minimal downtime. It should gracefully handle failures, such as network disruptions or server downtime, and ensure that patient data remains accessible and secure. The decentralized nature of the system, leveraging blockchain and IPFS, ensures that there is no single point of failure, maintaining continuous and reliable service.

## 3.3 Software Requirements:

### 3.3.1 Operating System:

The project is compatible with various operating systems, including Linux, Windows, and macOS. For server-side operations, Linux-based systems (e.g., Ubuntu) are preferred, as they provide a stable environment for Ethereum and blockchain-related development.

### 3.3.2 Programming Languages:

**Solidity:** Used to write smart contracts for managing access control and permissions in the decentralized medical record storage system on the Ethereum blockchain.
**JavaScript (React):** The primary language for implementing the frontend interface, enabling patient and healthcare provider interactions with the system.
**Python:** Used for backend development and machine learning model training, particularly for the federated learning framework for malware detection.
**Node.js:** Used for setting up and managing server-side logic, such as interacting with IPFS for decentralized storage and handling API requests.

### 3.3.3 Libraries and Frameworks:

**Ethers.js:** Libraries used to interact with the Ethereum blockchain, enabling the integration of smart contracts with the frontend and backend.
**React.js:** Used to develop the frontend interface, providing a user-friendly experience for both patients and healthcare providers.
**IPFS:** A decentralized file storage protocol used to store medical records securely and efficiently.

**Pinata:** A service for managing IPFS files, ensuring smooth upload and retrieval of encrypted medical records.

## 3.4 Constraints:

**Decentralized Storage Challenges:** Storing encrypted medical records on IPFS requires efficient file management to minimize latency while ensuring that records are securely stored and easily retrievable.

**Blockchain Network Limitations:** Ethereum transactions can be costly and time-consuming due to gas fees and network congestion. The project must optimize smart contract interactions to reduce costs and improve transaction efficiency.

**Data Privacy and Security:** Ensuring compliance with healthcare data protection regulations like HIPAA and GDPR is critical. The encryption, storage, and sharing of medical records must follow strict guidelines.

**Resource Constraints:** Computational resources for encryption, decryption, and blockchain interactions may be limited for users with less powerful devices, necessitating lightweight solutions.

**Scalability Issues:** As the number of patients and healthcare providers grows, the system must efficiently handle increased data storage and transaction volume without performance degradation.

**User Adoption and Technical Barriers:** The decentralized nature of blockchain may pose challenges for users unfamiliar with blockchain wallets (e.g., MetaMask). Adequate training and intuitive design are essential to drive user adoption.

## 3.5 Assumptions:

**Consistent Data Quality:** It is assumed that all medical records uploaded to the system are accurate, consistent, and free of errors to ensure smooth storage and retrieval.

**Secure User Authentication:** Patients and healthcare providers are expected to use secure wallet extensions (e.g., MetaMask) to authenticate and interact with the system.

**Uniform Smart Contract Usage:** It is assumed that all participants will use a standardized smart contract architecture for consistent interaction and access control.

**Reliable Network Infrastructure:** The Ethereum blockchain and IPFS networks are assumed to be accessible and reliable, ensuring uninterrupted operation.

**Non-Malicious Users:** The system assumes that participants, including patients and healthcare providers, will act in good faith and follow ethical practices when interacting with the platform.

**Stable Environment for Testing and Deployment:** It is assumed that the development and testing environments mirror real-world deployment conditions to ensure smooth transition and scalability.

# CHAPTER 4
# PROPOSED SOLUTION AND SYSTEM ARCHITECTURE

This chapter presents the proposed solution for implementing a blockchain-based system for secure and privacy-preserving electronic health records (EHR) management. It details the system's architectural framework and provides a comprehensive explanation of its workflow to achieve the project objectives.

## 4.1 Proposed Solution

Our system is designed with three main user roles: Patients, Doctors, and Admins. Patients have complete control over their medical records, allowing them to grant or revoke access for doctors as needed. Doctors can only access and update medical records with the patient's permission, ensuring that privacy is maintained. Admins facilitate user onboarding by verifying and registering new patients and doctors. The project workflow includes the use of smart contracts to manage permissions and access control. Once a patient grants access, this permission is securely recorded on the blockchain, ensuring transparency and accountability. Medical data is stored in IPFS (InterPlanetary File System), a decentralized storage solution that enhances security and data availability while reducing dependency on centralized servers. By integrating blockchain, smart contracts, and IPFS, this project delivers a robust, efficient, and secure system for EHR management, empowering patients and fostering trust in the healthcare data-sharing process. This approach addresses the limitations of current systems, ensuring data integrity, privacy, and accessibility, making it a pioneering solution in healthcare data management

## 4.2 System Architecture

The proposed system, Medical-Chain, is a decentralized platform for managing EHRs.It incorporates the following key technologies: Interplanetary File System (IPFS): Used for distributed, encrypted storage of medical records, ensuring scalability and resilience against data loss. Ethereum Smart Contracts: Smart contracts are used to manage data access permissions, ensuring that only authorized parties can view or modify records. Patients retain full control over who can access their data, without relying on third-party intermediaries. Decentralized Application (DApp): The system provides a user-friendly interface for patients and healthcare providers, allowing them to interact with the blockchain and manage records efficiently. By eliminating the reliance on a single centralized server, this architecture reduces the risk of breaches and enhances patient privacy.

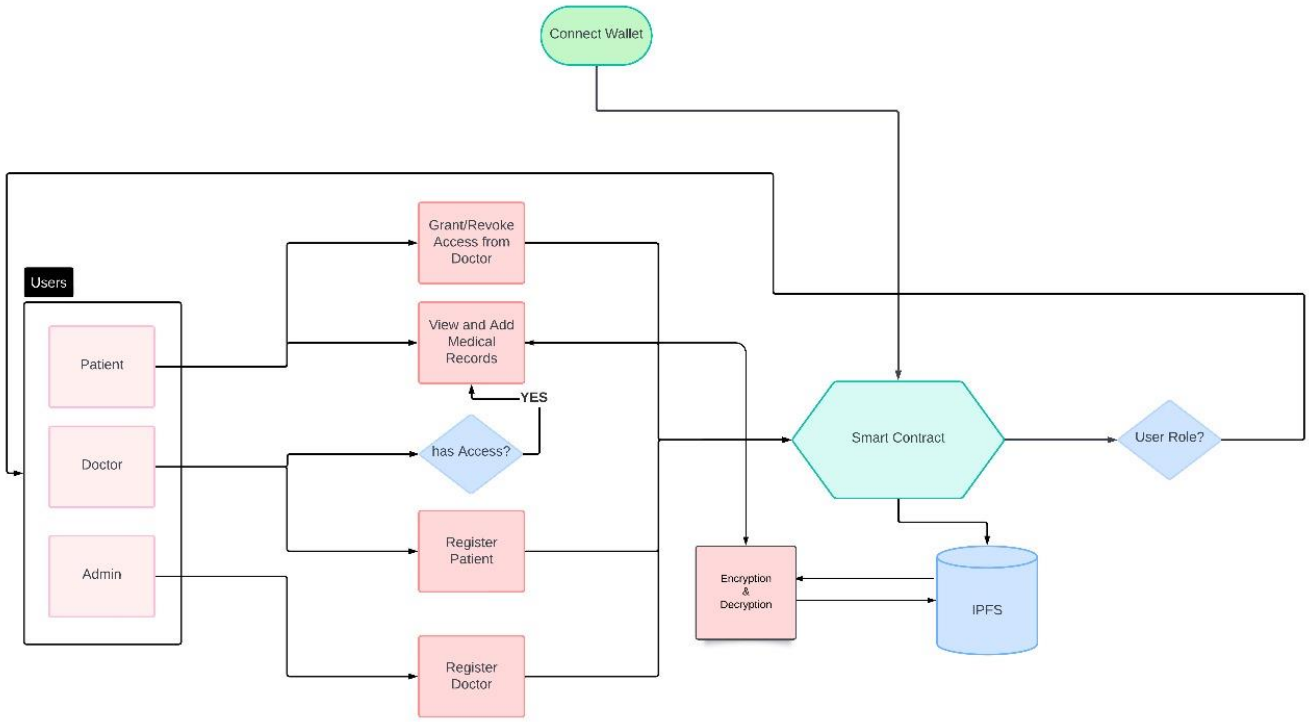


Fig. 4.2 System Architecture

**User Roles:**

**Patient:** Can view and add medical records, as well as grant or revoke access to doctors.

**Doctor:** Can view medical records if access is granted by the patient.

**Admin:** Responsible for registering patients and doctors in the system.

**Authentication:**

Users connect their wallets (e.g., MetaMask) for authentication and interaction with the blockchain.

**Smart Contract:**

Acts as the central controller managing user roles, access permissions, and data flow. Handles patient and doctor registration, access control, and interactions with IPFS.

**Access Control:**

Access permissions are verified through the smart contract.

Patients can grant or revoke access to their medical records for specific doctors.

**Data Storage:**

Medical records are encrypted and stored on **IPFS** for decentralized and secure storage. The system ensures data privacy by encrypting sensitive information before storing it.

**Encryption & Decryption:**

Encryption is applied to medical records before uploading to IPFS. Authorized users decrypt the records upon retrieval.

## 4.3 Detailed Workflow

**1. User Roles and Initial Access**:

Users in the system are categorized into three roles: Patient, Doctor, and Admin. Admins have the responsibility of registering new patients and doctors into the system, verifying their identities, and providing them access to the platform. This ensures that only verified individuals have access to sensitive medical data.

**2. User Authentication and Wallet Connection:**

Each user must connect their digital wallet to authenticate and access the system. This wallet connection is crucial for interacting with the blockchain. - The wallet also links each user's identity to their specific role (Patient, Doctor, or Admin) within the smart contract, helping determine their permissions and actions within the system.

**3. Smart Contract Interaction for Access Control**:

A smart contract acts as a central authority on the blockchain to manage permissions and access control for EHR data. - The smart contract handles: - Granting and Revoking Access: Patients can use the smart contract to grant or revoke access to their records for specific doctors. Recording Access Events: Each time access is granted or revoked, the event is logged on the blockchain, ensuring transparency and accountability. This log is immutable and provides a record of all interactions with patient data.

**4. EHR Data Storage and Retrieval with IPFS**:

Medical records are stored on IPFS (Inter Planetary File System), a decentralized storage network that provides secure, efficient, and distributed storage. When a record is created, it is saved on IPFS, generating a unique hash or CID (Content Identifier). This CID is then stored on the blockchain via the smart contract instead of the actual data. This approach ensures that sensitive data is kept off the blockchain, reducing storage costs and improving scalability while still maintaining secure, verifiable access through the blockchain.

**5. Patient Actions:**

A patient can select a doctor and grant access to their records by interacting with the smart contract. This triggers a blockchain transaction that records the patient's consent and assigns the doctor permission to view the medical records linked to that patient. If a patient wants to remove access, they can revoke the doctor's permission through the smart contract. The smart contract updates the permissions and logs this change on the blockchain, instantly removing the doctor's access to the records.

**6. Doctor Actions**:

When a doctor has been granted access, they can retrieve the medical records by fetching the IPFS hash from the blockchain. The hash allows them to access the file from IPFS, provided they have permission. Add New Medical Records Doctors can also add new

records to a patient's file (if permission is granted). These records are uploaded to IPFS, generating a new hash, which is then stored on the blockchain, updating the patient's medical history.

**7. Admin Actions:**

Admins are responsible for onboarding and verifying new users (patients and doctors). This ensures that only legitimate, verified users have access to the system. While the system is decentralized, admins can still monitor access logs for compliance and audit purposes, ensuring that all actions align with legal and organizational standards.

**8. Transparency and Auditability:**

All interactions with patient data such as granting, revoking, viewing, and adding records are recorded on the blockchain, creating a permanent, transparent record. Patients, doctors, and admins can all verify access logs, which builds trust in the system. Patients are assured that their data is only accessed by authorized personnel, while doctors can confidently access data without the risk of unauthorized alterations.

**9. Security and Privacy Measures:**

Data Encryption: Even though records are stored on IPFS, they can be encrypted before uploading to ensure only authorized users can view the contents. Both IPFS and blockchain are decentralized, reducing the risk of data loss or tampering and making the system resilient to central points of failure. Since access permissions are stored on the blockchain, they cannot be altered without triggering a new transaction. This immutability ensures that permissions are transparently tracked over time.

**10. System Workflow Summary:**

Patients register and connect their wallets, enabling them to control access to their medical records. Doctors, once granted access by a patient, can view and add to medical records as needed. Admins manage user onboarding and maintain oversight, ensuring the system operates smoothly.

## 4.4 Workflow of Encryption and Secure Sharing of Medical Data

## Encryption and Decryption of Medical Records

### 1. Data Encryption Using AES

- **Step 1**: **Data Upload**
  - The user (patient or healthcare provider) uploads a medical record (e.g., medical report, diagnosis, prescription) to the system.
- **Step 2**: **Generate AES Key**
  - A random secret key is generated for the AES encryption algorithm. This key will be used to encrypt the medical record. AES is chosen for its efficiency and security for encrypting large amounts of data.
- **Step 3**: **Generate SHA-256 Digest and Digital Signature**
  - The SHA-256 digest of the medical record is computed to create a unique fingerprint of the data.
  - The generated SHA-256 digest is then signed using the private MetaMask key to create a digital signature.
- **Step 4**: **Encrypt Medical Data**
  - The medical record is encrypted using the AES encryption algorithm, with the generated secret key. The encryption process ensures that only someone with the correct AES key can access the data in its original, readable form.
- **Step 5**: **Upload Encrypted Data to IPFS**
  - The encrypted medical record is uploaded to the **InterPlanetary File System (IPFS)**. IPFS is used to store the data in a decentralized manner, ensuring availability without relying on a central server.
- **Step 6**: **Store AES Key**
  - The AES key used for encrypting the data is stored separately. It will not be stored on IPFS, ensuring that the key is not exposed along with the data.

### 2. RSA Encryption for Secret Key Sharing

- **Step 7**: **Recipient Identification**
  - The system identifies the recipient (e.g., doctor, healthcare provider) who needs access to the encrypted medical record. The recipient's public RSA key is retrieved from the system.
- **Step 8**: **Encrypt AES Key with RSA**
  - The AES key is encrypted using the recipient's **public RSA key**. RSA is used because of its ability to securely encrypt small amounts of data (like the AES key) and ensure that only the corresponding private RSA key can decrypt it.
- **Step 8**: **Send Encrypted AES Key**

- The encrypted AES key is sent securely to the recipient. This ensures that only the intended recipient, who possesses the corresponding private RSA key, can decrypt the AES key.

**3. Decryption Process**

- **Step 10**: **Decrypt AES Key with Private RSA Key**
  - The recipient (e.g., doctor) uses their **private RSA key** to decrypt the AES key. The private key is only accessible to the recipient, ensuring that only they can unlock the AES key.
- **Step 11**: **Decrypt Medical Data Using AES**
  - Once the AES key is decrypted, the recipient uses it to decrypt the medical record stored on IPFS. This allows the healthcare provider to access the medical data in its original form.

**4. Data Access and Control**

- **Step 12**: **Grant or Revoke Access**
  - The system can grant or revoke access to the data by controlling who has access to the AES key, ensuring that data is only available to authorized users. Smart contracts can be used to automate the process of granting or revoking access based on user roles and permissions.

## Granting and Revoking Access in the System

**1. Granting Access**

- **Step 1**: **Recipient Identification**
  - The system identifies the user (e.g., a doctor or healthcare provider) who requires access to the encrypted medical records. The recipient's **public RSA key** is retrieved from the system.
- **Step 2**: **Encrypt AES Key Using Recipient's Public RSA Key**
  - The AES key, which was used to encrypt the medical data, is encrypted using the recipient's **public RSA key**.
  - This ensures that only the recipient can decrypt the AES key with their **private RSA key**, as only they possess the matching private key.
- **Step 3**: **Share the Encrypted AES Key**
  - The system sends the **encrypted AES key (ciphertext)** along with the encrypted medical record to the recipient.
  - This ensures that the recipient can decrypt the AES key and subsequently access the encrypted medical data.

**2. Revoking Access**

- **Step 1**: **Identify Users with Access**
  - The system retrieves the list of all users who currently have access to the data, i.e., those who have been provided with the AES key.
- **Step 2**: **Revoke Access**
  - Once the system identifies that a user's access needs to be revoked, the following occurs:
    - The **medical records** that were previously encrypted with the AES key are decrypted using the current AES key.
    - A **new AES key** (Secret Encryption Key or SEK) is generated to replace the old one.
- **Step 3**: **Regenerate SHA-256 and Digital Signature**
  - The **SHA-256 digest** of the previously decrypted medical data is computed.
  - The computed digest is then **signed** using the **private MetaMask key** to generate a new **digital signature**.
- **Step 4**: **Re-encrypt Medical Data with New AES Key**
  - The previously decrypted medical data is now re-encrypted using the newly generated **AES key (new SEK)**.
  - This ensures that the data is now protected with a new key, and the old AES key is no longer valid for decrypting the data.
- **Step 5**: **Re-encrypt the AES Key with the Recipient's Public RSA Key**
  - For the remaining users who still have access, their **public RSA keys** are used to re-encrypt the new AES key (new SEK).
  - This ensures that they can continue accessing the updated encrypted data, while the revoked user will not be able to decrypt it using the old AES key.
- **Step 6**: **Send New Encrypted AES Key to Authorized Users**
  - The system sends the newly encrypted AES key (ciphertext) to the remaining authorized users.
  - These users can now use their private RSA keys to decrypt the new AES key and access the medical data securely with the updated encryption.

# Algorithms

## Algorithm: Encryption with Signature and Encryption

1. Start
2. Generate SHA-256 Digest
    - Compute the SHA-256 digest of the medical data (record) to get the unique fingerprint of the file.
3. Sign the Digest
    - Use the private MetaMask key to sign the SHA-256 digest of the medical data.
    - This generates a digital signature that proves the authenticity of the data and can be used to verify its integrity during decryption.
4. Encrypt the Medical Data
    - Encrypt the medical data using the AES key (SEK).
5. Send Data
    - Upload the SHA-256 digest, digital signature to Blockchain and encrypted medical data to IPFS.
6. End

## Algorithm: Decryption with Signature Verification

1. Start
2. Fetch Data
    - Fetch the SHA-256 digest, digital signature, encrypted secret key (if the user is doctor and has access) from Blockchain and encrypted medical data from IPFS
3. Decrypt the AES Key (SEK)
    - Use the recipient's private RSA key to decrypt the AES key (SEK).
4. Verify the Digital Signature
    - Use the public MetaMask key to verify the digital signature against the SHA-256 digest of the medical data.
    - If the signature verification fails, reject the data and notify the user, as the integrity of the file cannot be guaranteed.
5. Verify the SHA-256 Digest
    - Recompute the SHA-256 digest of the received medical data.
    - Compare the recomputed digest with the SHA-256 digest sent along with the encrypted data.

- If the digests do not match, it means the data has been tampered with, and the decryption is rejected.

6. Decrypt the Medical Data
    - If the signature is valid and the digest matches, use the AES key (SEK) to decrypt the encrypted medical data.
7. Show the Decrypted Data
    - Display the decrypted medical data to the recipient, as it is now verified and authentic.
8. End

**Algorithm: Granting Access**

1. Start
2. Identify Recipient
    - Retrieve the public RSA key of the recipient (e.g., doctor or healthcare provider).
3. Encrypt AES Key (SEK)
    - Use the recipient's public RSA key to encrypt the AES key (SEK).
4. Share Encrypted AES Key and Medical Data
    - Send the encrypted AES key (ciphertext) and encrypted medical data to the recipient.
5. End

**Algorithm: Revoking Access**

1. Start
2. Identify Users with Access
    - Retrieve the list of authorized users who currently have access to the data.
3. Decrypt Medical Data
    - Decrypt the medical data using the AES key (SEK) that was originally used to encrypt the data.
4. Re-encrypt Medical Data
    - Re-encrypt the decrypted medical data using the same AES key (SEK), which will still be valid for all remaining users.
5. Re-encrypt AES Key (SEK)

- For each remaining authorized user, use their public RSA key to encrypt the AES key (SEK) again. This ensures the remaining users can still decrypt the AES key with their private RSA key.

6. Share New Encrypted AES Key
    - Send the new encrypted AES key (ciphertext) to the remaining authorized users.
7. End

# CHAPTER 5

# SYSTEM ANALYSIS

## 5.1 Existing System Analysis:

The existing systems for managing electronic health records (EHRs) typically rely on centralized databases maintained by hospitals, clinics, or third-party service providers. While these systems offer convenience, they face several limitations:

1. **Centralized Control**:
    - Data is stored in a central repository, making it vulnerable to breaches, unauthorized access, and single points of failure.
    - Patients often lack direct control over their records.
2. **Lack of Transparency**:
    - Users cannot trace who has accessed or modified their data.
    - Limited auditing mechanisms to ensure accountability.
3. **Privacy Concerns**:
    - Data breaches are common, exposing sensitive patient information to unauthorized entities.
    - Third-party data sharing often occurs without explicit user consent.
4. **Interoperability Issues**:
    - Inconsistent formats and standards among different healthcare providers create challenges for data exchange.
    - Limited integration across platforms for seamless sharing.
5. **High Costs**:
    - Maintaining and securing centralized systems can be expensive for healthcare providers.
    - Data redundancy is often required to ensure backups, further adding to costs.

## 5.2 Proposed System Analysis:

The proposed system addresses the limitations of the existing systems by leveraging blockchain technology to create a decentralized, secure, and transparent platform for managing EHRs.

1. **Decentralized Architecture**:
    - The system eliminates single points of failure by storing data across a distributed blockchain network.
    - Patients retain control over their data and manage permissions using smart contracts.

2. **Enhanced Security**:
    - Blockchain's cryptographic mechanisms ensure data integrity and prevent unauthorized modifications.
    - Immutable records guarantee that all actions are logged and traceable.
3. **Transparency and Accountability**:
    - Every transaction (e.g., access or modification) is recorded immutably on the blockchain.
    - Smart contracts automate access permissions, ensuring fair and rule-based data sharing.
4. **Privacy Preservation**:
    - Users can define access policies, granting or revoking permissions for specific healthcare providers.
    - No data is stored off-chain, ensuring full control and minimizing risks.
5. **Interoperability**:
    - The system uses standardized formats to ensure seamless data sharing across different platforms.
    - Healthcare providers can access real-time updates while maintaining compliance with regulations.
6. **Cost Efficiency**:
    - Reduces overhead costs associated with maintaining centralized servers.
    - Minimizes redundancy by securely storing data in a distributed network.
7. **User Empowerment**:
    - Patients are at the center of the system, with complete control over their records.
    - Enhances trust between users and healthcare providers through transparent interactions.

# CHAPTER 6

# TOOLS AND TECHNOLOGIES USED

This chapter outlines the tools, programming languages, and libraries employed to design, develop, and implement the **Medical Blockchain-Based System for Secure and Privacy-Preserving Data Sharing**, highlighting their significance in efficiently achieving the project objectives.

## 6.1 Programming Languages:

**JavaScript**: JavaScript is utilized for creating the frontend of the system, ensuring an interactive and responsive user experience.

**Solidity**: Solidity is used for developing and deploying smart contracts on the Ethereum blockchain to manage data access, permissions, and transactions securely.

**Python**: Python serves as a core programming language in the system for real time monitoring of network performance, including key security metrics such as network hash rate and resistance to attacks. The solution is validated through performance benchmarking across metrics like latency, throughput, scalability, and usability, and data processing.

## 6.2 Blockchain Framework:

**Hardhat**: A powerful Ethereum development framework used for deploying and testing smart contracts. It provided utilities such as network management, debugging, and script execution. Hardhat's plugins, such as Hardhat Toolbox, enhanced development efficiency by integrating testing libraries and blockchain interaction modules.

## 6.3 Libraries and Utilities:

**Ethers.js**: Facilitated seamless interaction with the Ethereum blockchain. It was used for reading and writing data to smart contracts, handling wallets, and connecting to Ethereum nodes. Ethers.js' lightweight nature and comprehensive documentation made it a preferred choice.

**Pinata API**: Integrated for decentralized file storage using IPFS (InterPlanetary File System). The Pinata API allowed secure upload and retrieval of files, making it ideal for storing sensitive medical records.

## 6.4 Frontend Framework:

**React.js**: The core framework used to build the frontend. React's component-based architecture enabled the creation of reusable UI elements, simplifying development and maintenance. Features like hooks allowed efficient state management and lifecycle handling.

**Vite**: A fast build tool and development server used in conjunction with React.js. Vite provided superior performance with its Hot Module Replacement (HMR) feature, which sped up the development process by instantly reflecting changes made in the code.

## 6.5 Version Control and Collaboration:

**Git**: Used for version control, enabling tracking of changes and collaborative development. Git ensured that code changes were managed effectively, and different versions of the project could be accessed as needed. Features like branching facilitated team collaboration and feature isolation during development.

## 6.6 Security and Encryption:

**Ethereum Cryptography Library:** Implemented for secure hashing and cryptographic operations. This ensured that data integrity and confidentiality were maintained throughout blockchain interactions.

**AES Encryption:** Advanced Encryption Standard (AES) was applied to encrypt sensitive data before uploading it to IPFS. This added an additional layer of security for medical records and ensured compliance with data protection standards.

**RSA Encryption:** RSA is employed to encrypt the secret key used in AES encryption while sharing it with authorized users. This ensured that only the intended recipient could decrypt the secret key, adding another layer of protection during the sharing process.

## 6.7 Backend Services:

**FastAPI:** Used as the backend framework to develop high-performance APIs for retrieving system metrics such as throughput and latency. Its asynchronous capabilities ensured efficient handling of requests, enabling real-time insights into the system's performance.

# CHAPTER 7

# RESULTS

## Main Page:

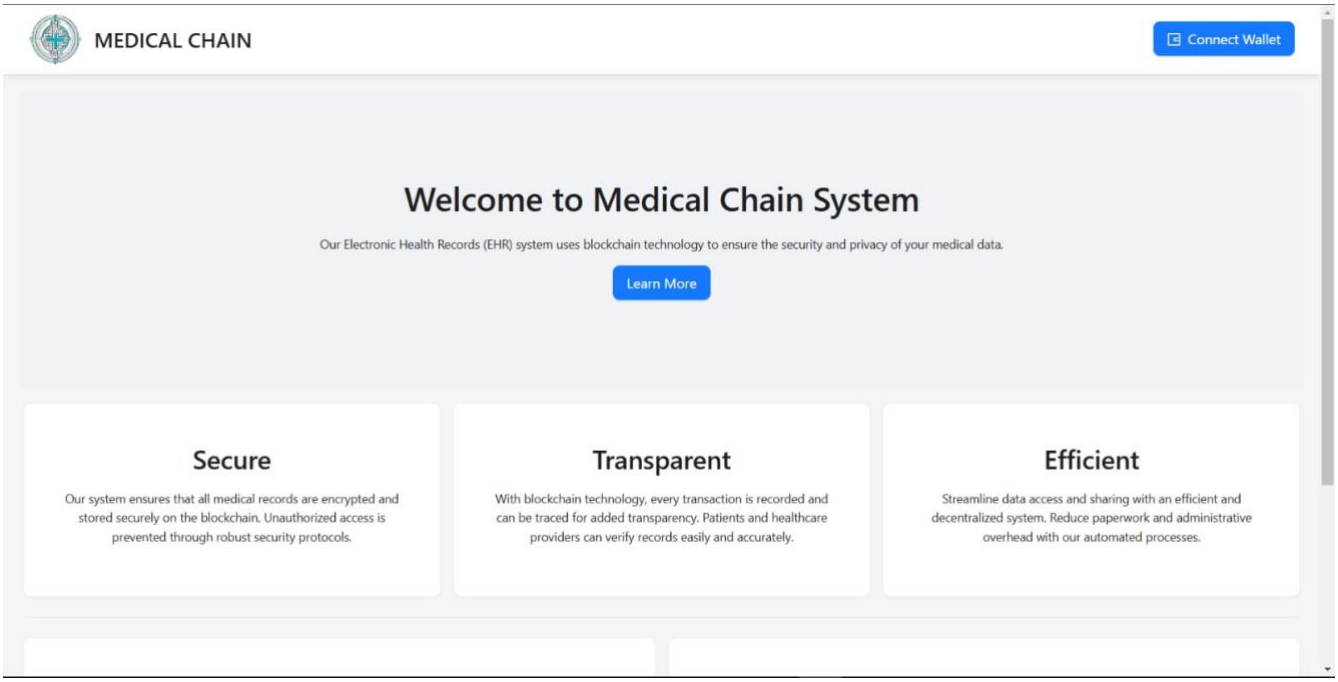


Fig 7.1: Main Page

## Admin Profile :

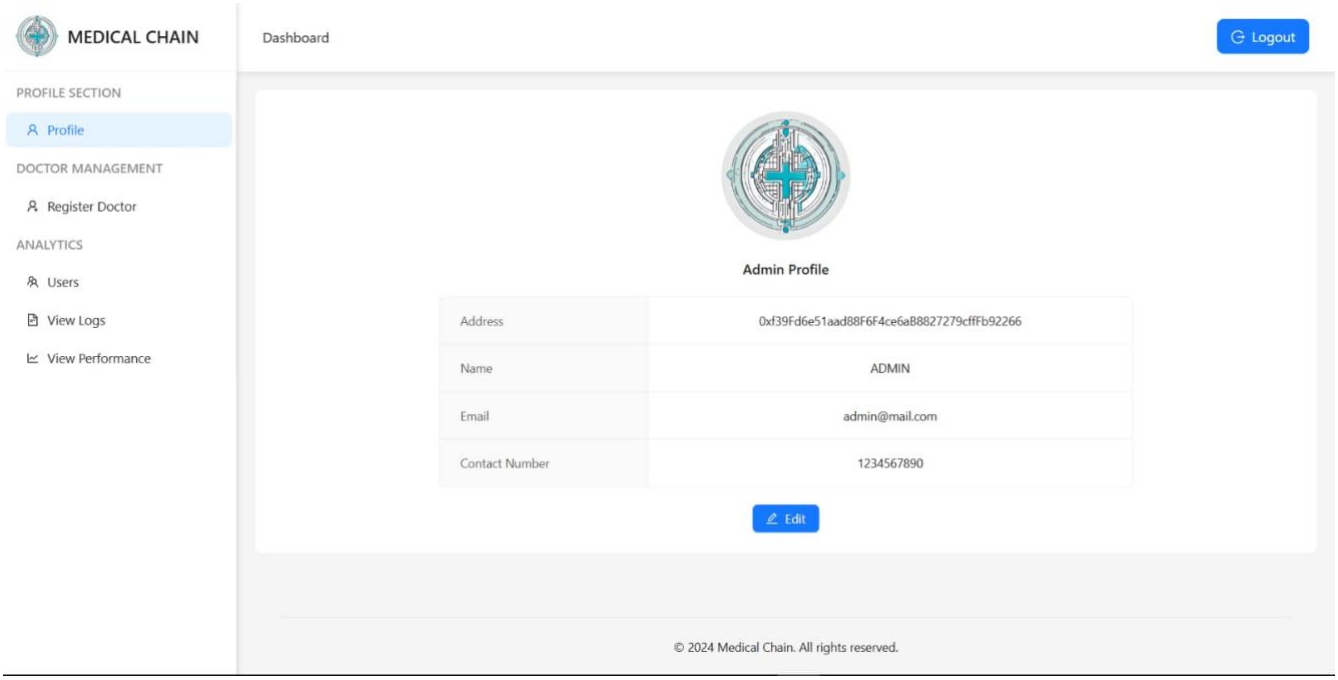


Fig 7.2: Admin Main Page

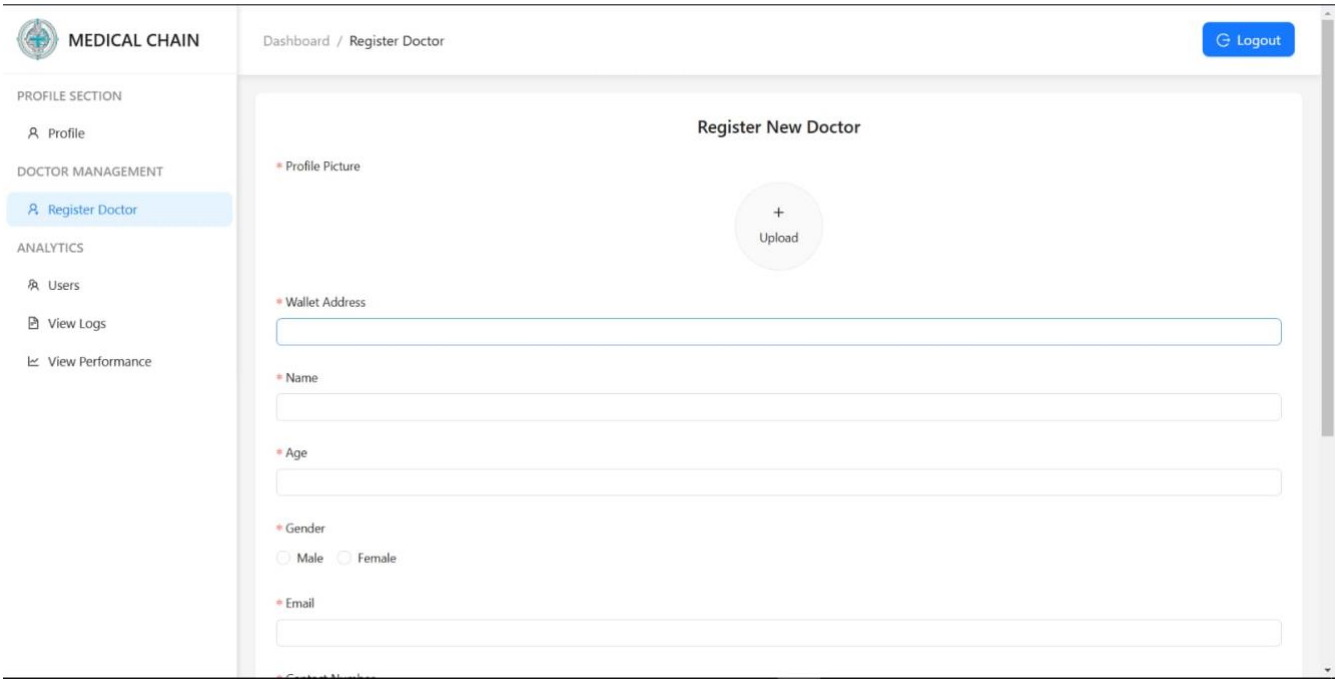


Fig 7.3: Doctor registration by admin

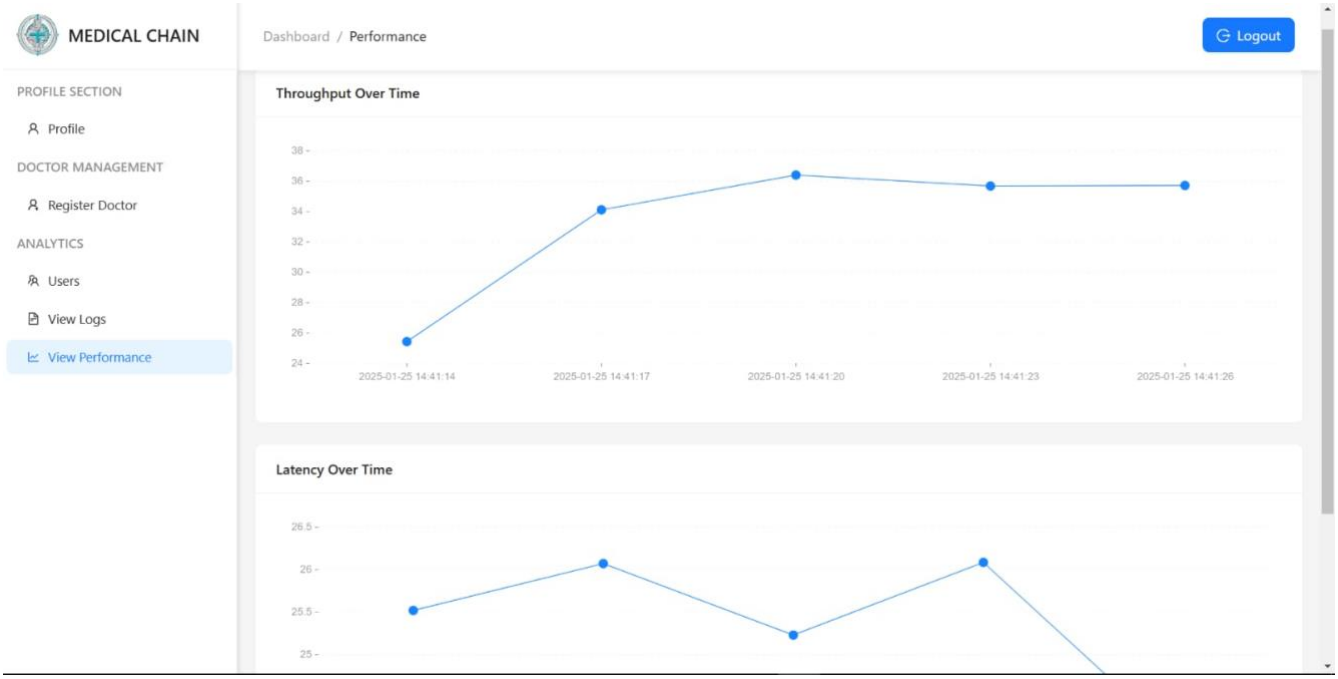


Fig 7.4: Performance and Throughput

# Doctor Profile:

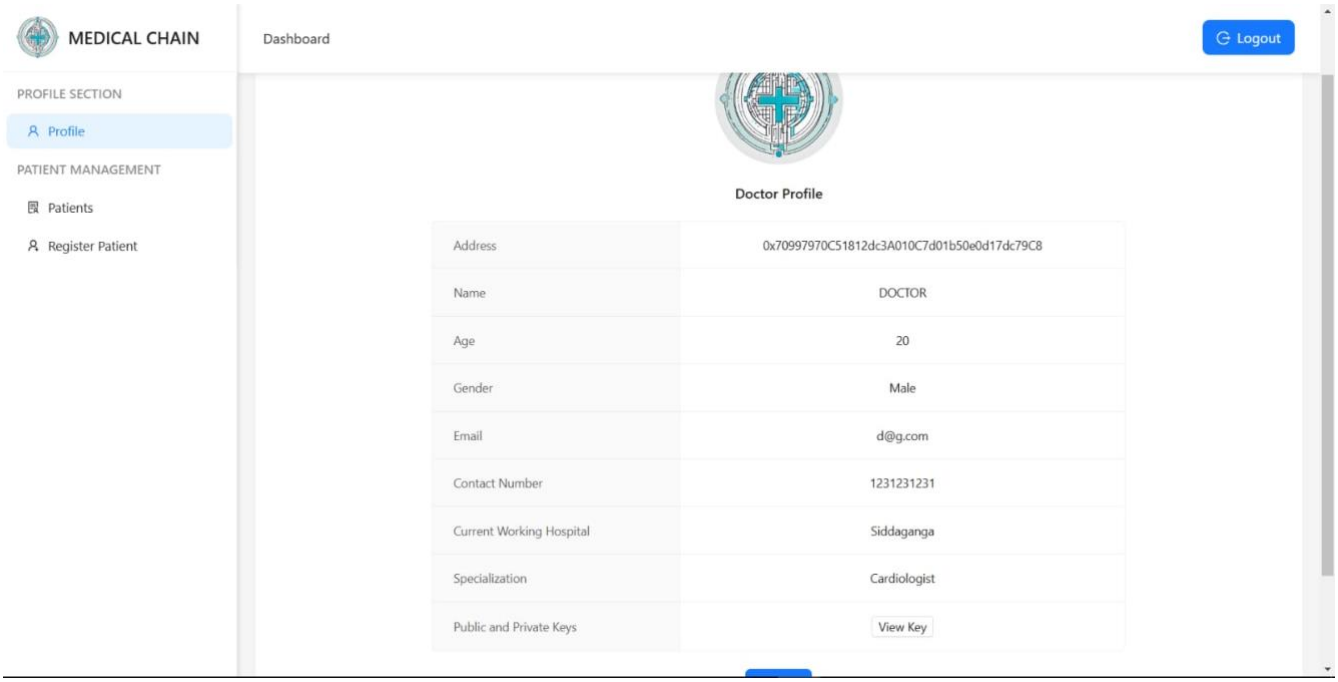


Fig 7.5: Doctor Main Page

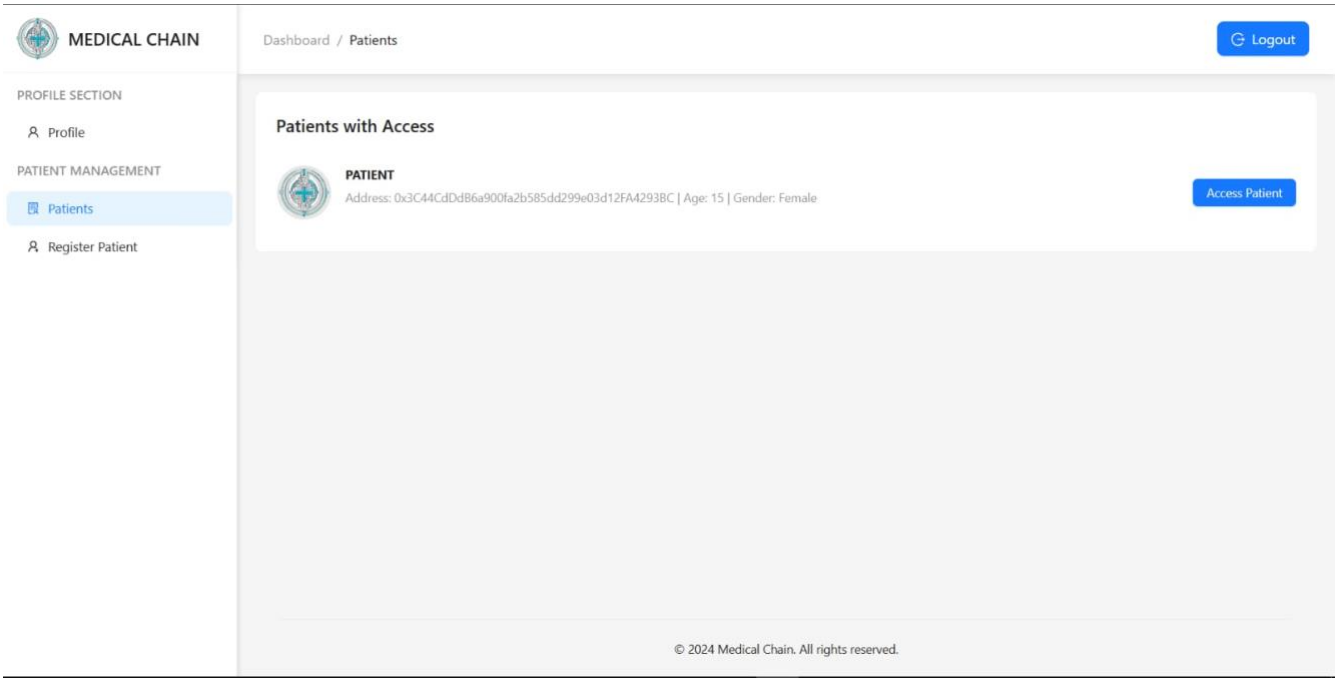


Fig 7.6: List of Patients who have granted access

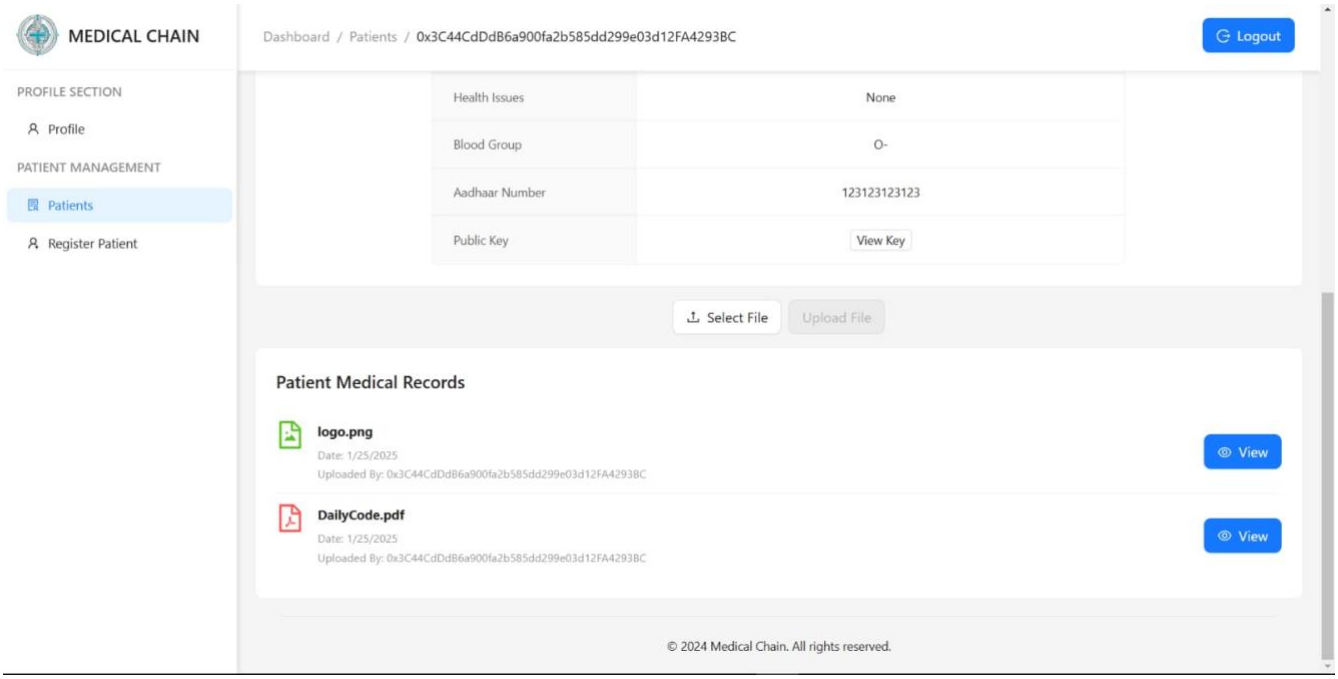


Fig 7.7: Doctor Viewing the Patient's records

# Patient Profile:

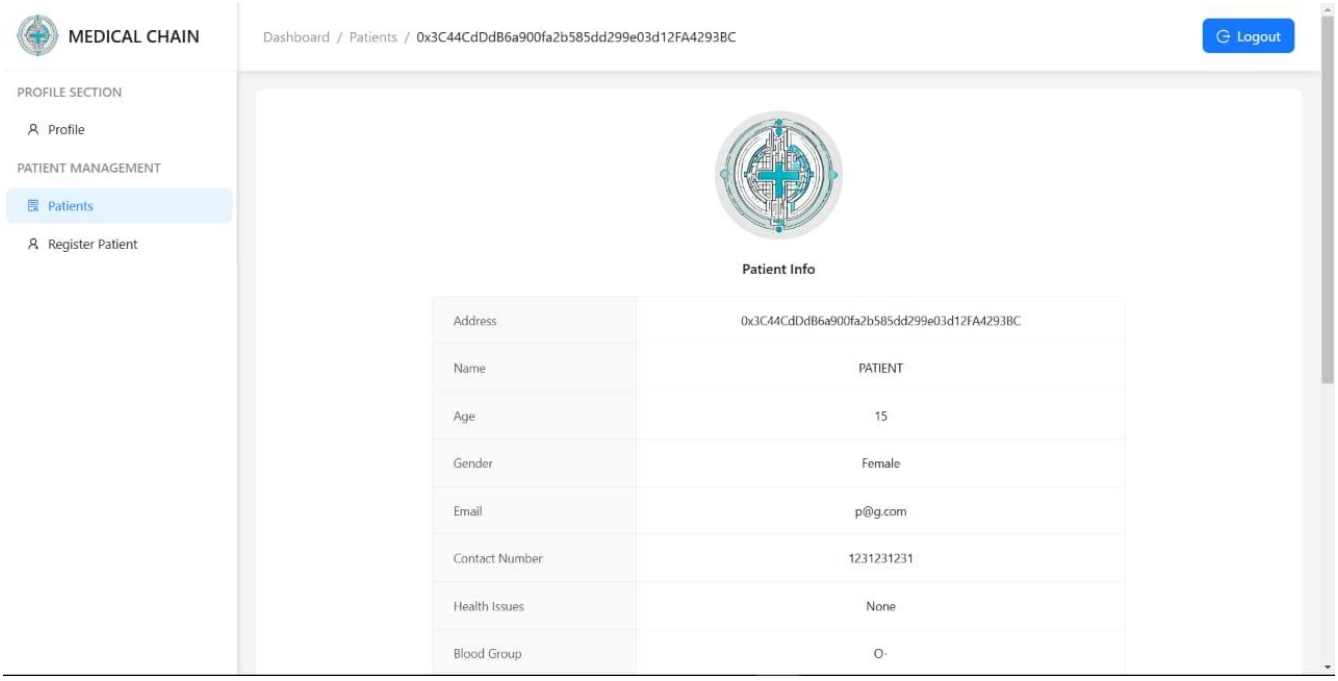


Fig 7.8: Patient Profile

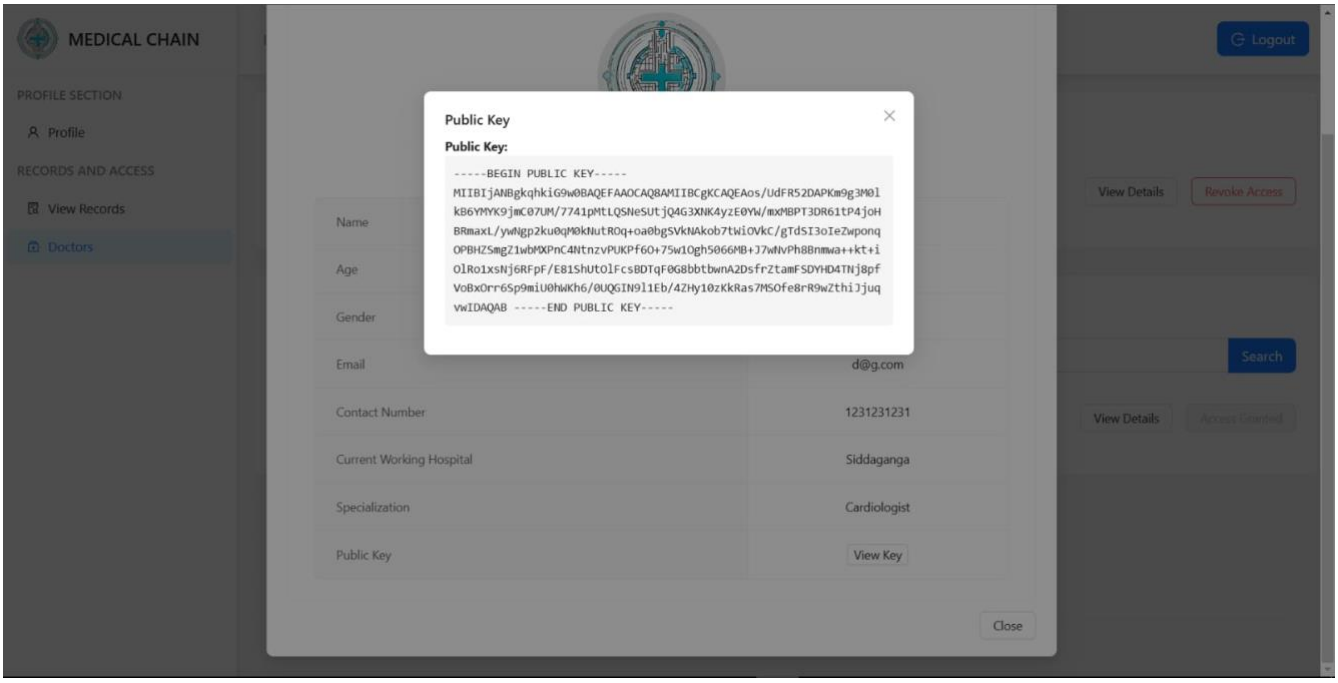


Fig 7.9: Public key

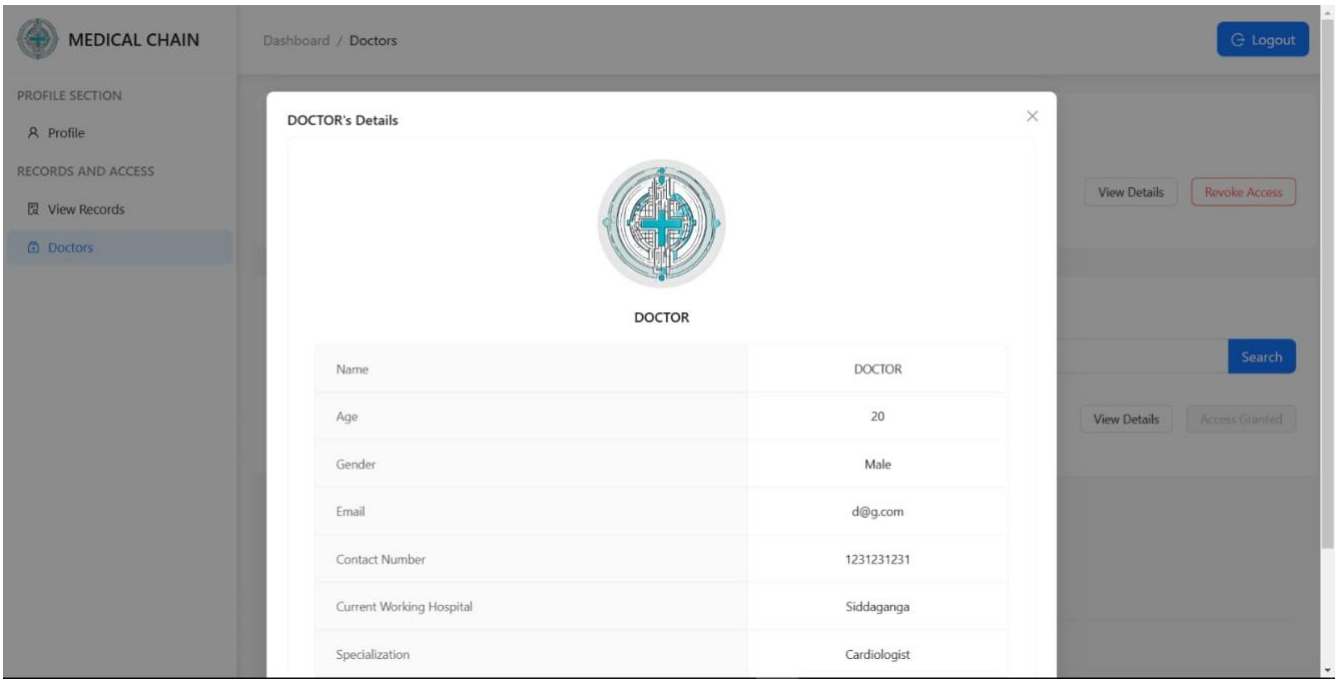


Fig 7.10: Doctor Info

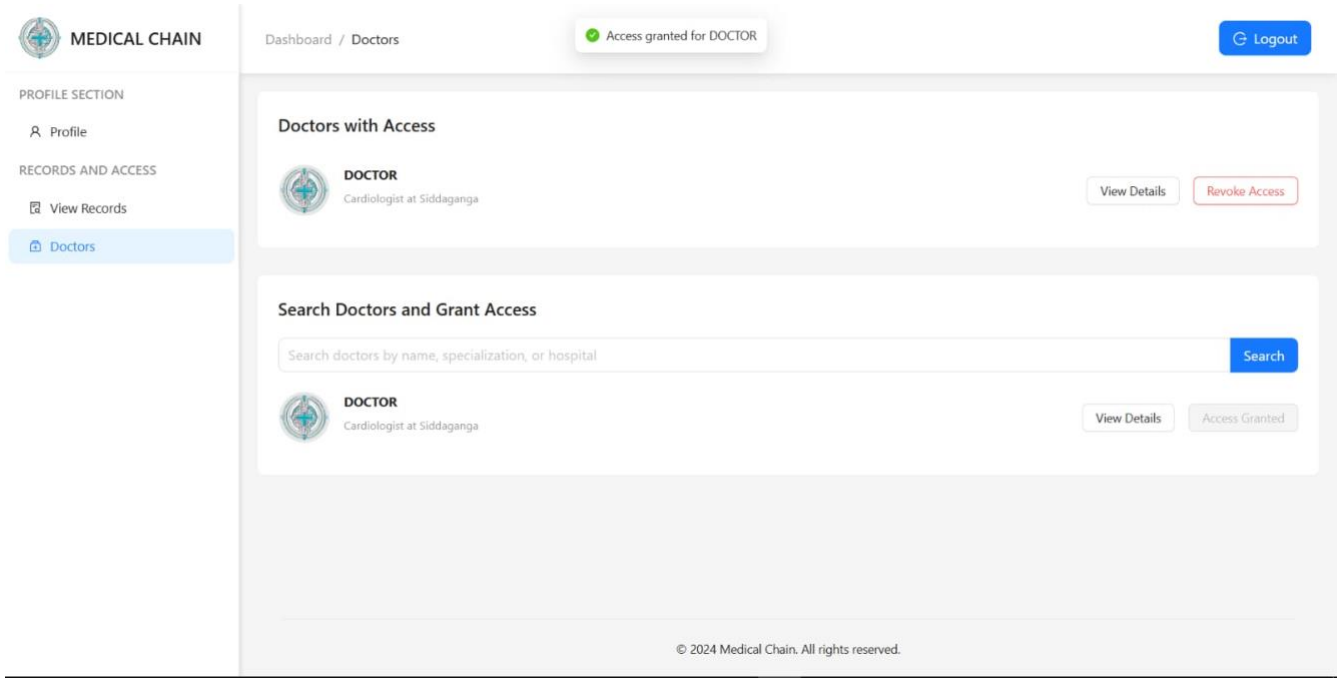


Fig 7.11: Granting and revoking access to doctor

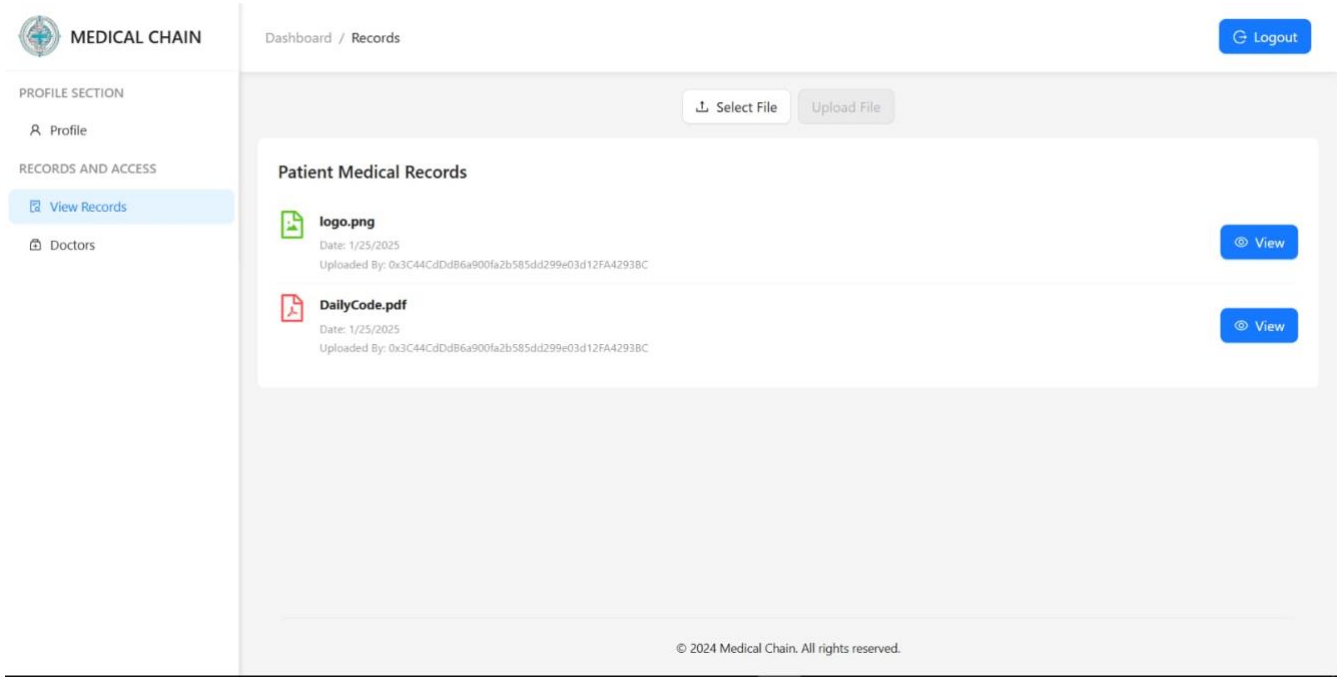


Fig 7.12: Uploaded patient records

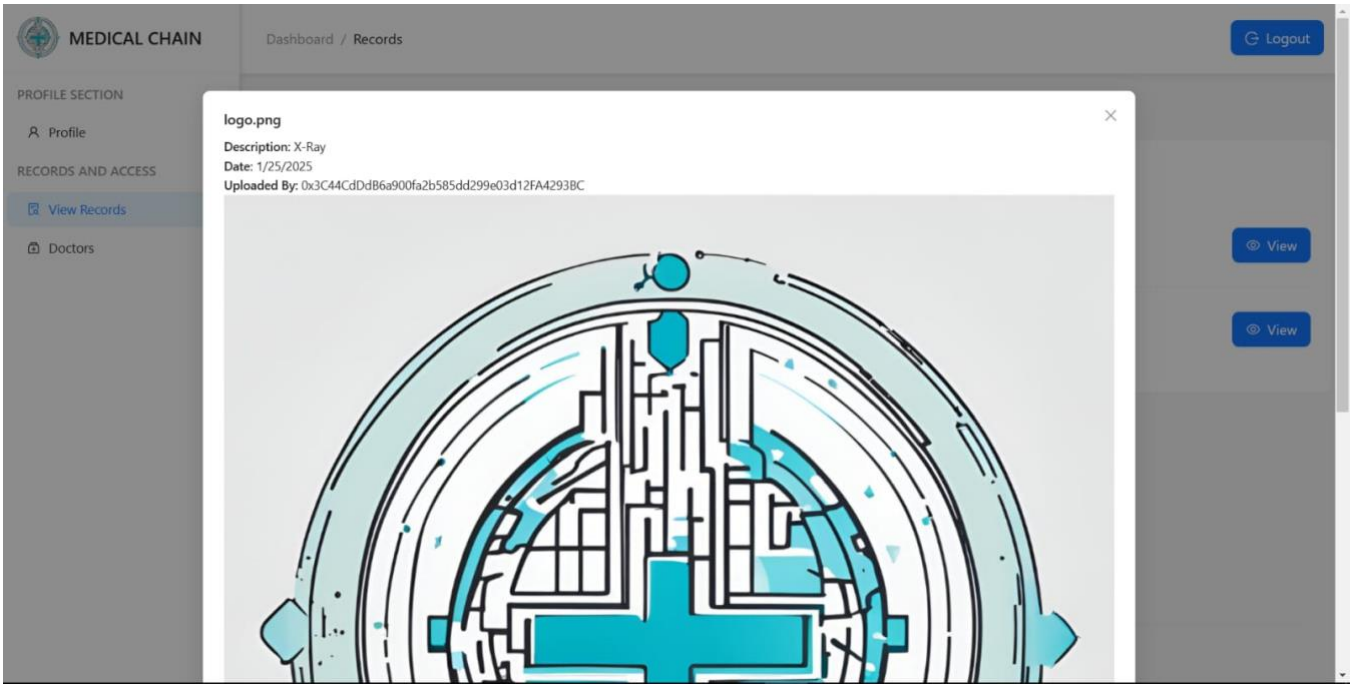


Fig 7.13: PNG record view

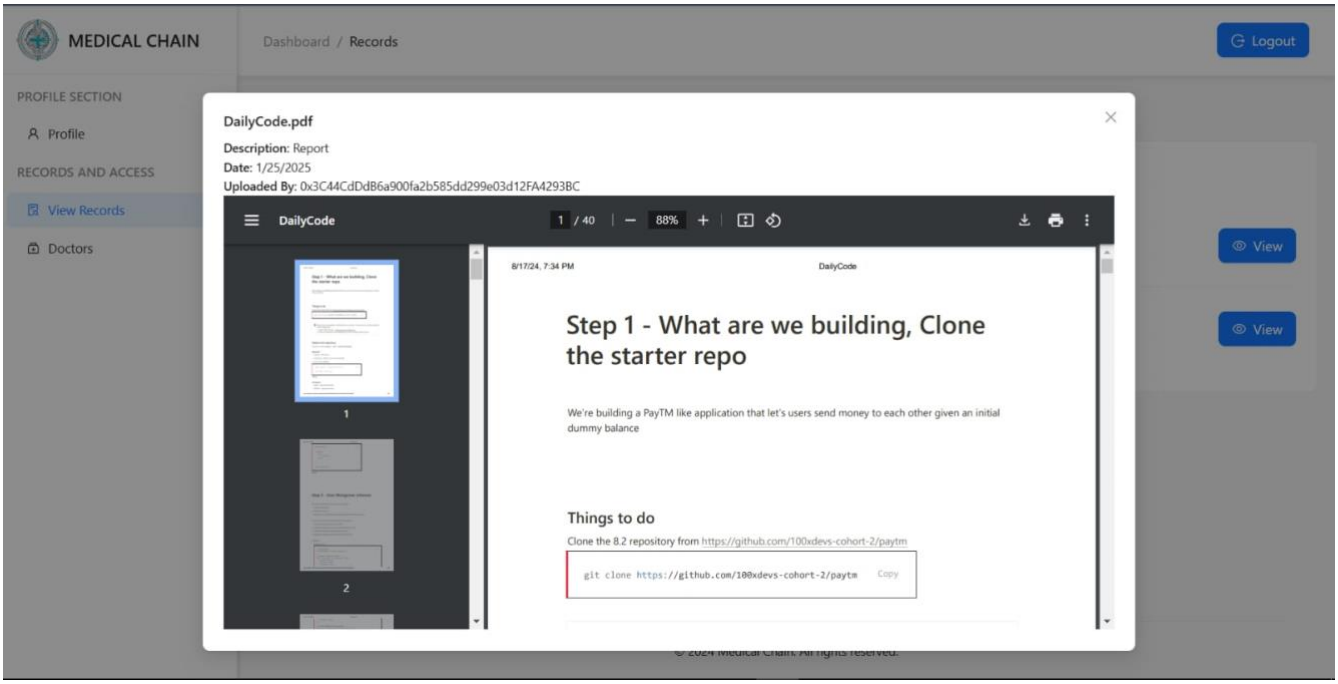


Fig 7.14: PDF record view

# CHAPTER 8

# APPLICATIONS

The blockchain-based Electronic Health Record (EHR) system can be applied in various healthcare settings to enhance data security, patient privacy, and interoperability. Here are some specific applications:

1. **Secure Medical Record Storage and Access**
    - The system ensures that medical records are securely stored on a decentralized blockchain network, reducing the risk of data breaches and unauthorized access. This application is crucial for hospitals, clinics, and healthcare providers looking to improve the security of patient information [1].
2. **Patient-Centric Data Management**
    - Patients can control who accesses their medical records, giving them more power over their personal health data. They can grant or revoke access permissions to healthcare providers, family members, or researchers, ensuring privacy and trust in the healthcare system [2].
3. **Interoperability Among Healthcare Providers**
    - Blockchain facilitates seamless sharing of medical records among different healthcare providers, improving collaboration and continuity of care. This interoperability ensures that a patient's medical history is readily available to any authorized provider, leading to better-informed treatment decisions [3].
4. **Medical Research and Data Analysis**
    - Researchers can access anonymized patient data for medical studies, clinical trials, and data analysis while maintaining patient privacy. The immutable nature of blockchain ensures the integrity of the data used in research, leading to more reliable outcomes [4].
5. **Real-Time Health Monitoring and Telemedicine**
    - Blockchain can be integrated with Internet of Things (IoT) devices for real-time health monitoring. Patients can share their health data securely with their doctors, enabling remote consultations and continuous health monitoring, which is particularly beneficial for managing chronic conditions [5].
6. **Insurance Claims Processing**

- The system can streamline insurance claims by providing verifiable and tamper-proof medical records. This application reduces fraud, speeds up the claims process, and ensures accurate and transparent transactions between patients, healthcare providers, and insurance companies [6].

7. **Regulatory Compliance and Auditability**

- Blockchain provides a transparent and auditable trail of all transactions related to medical records, helping healthcare providers comply with regulations such as HIPAA in the United States and GDPR in Europe. This feature ensures that all data handling practices meet legal requirements [7].

8. **Emergency Medical Access**

- In emergency situations, authorized healthcare providers can quickly access a patient's medical history through the blockchain network, ensuring timely and effective treatment. This application is critical for emergency responders and emergency room personnel [8].

9. **Personalized Medicine and Treatment Plans**

- The blockchain system can facilitate the creation of personalized medicine by allowing access to comprehensive patient data, including genetic information, lifestyle factors, and previous medical history. This data enables healthcare providers to develop tailored treatment plans that are more effective for individual patients [9].

**10.Supply Chain Management in Pharmaceuticals**

- Blockchain can enhance the transparency and traceability of pharmaceuticals within the supply chain. By recording every transaction on a blockchain, stakeholders can track the origin of medications, verify their authenticity, and ensure that drugs are handled properly, thereby reducing counterfeit products in the market [10].

**11.Patient Consent Management**

- The system can provide a secure method for managing patient consent for data sharing. Patients can digitally sign consent forms that are stored on the blockchain, ensuring that their consent is easily verifiable and auditable [11].

# CHAPTER 9

# FUTURE WORK

As part of our future enhancements, we plan to extend the capabilities of the current system by incorporating advanced technologies such as AI and machine learning (AI/ML) to enable **hierarchical data sharing**. Additionally, we aim to implement an **emergency access** mechanism for critical scenarios where a patient is unconscious or unable to provide consent. These enhancements will improve data accessibility, security, and usability while maintaining patient privacy and compliance with healthcare standards.

## Hierarchical Data Sharing

1. **Objective**:
    - To categorize uploaded records hierarchically based on their content.
    - To automate the process of assigning a hierarchy number or category to each record, enabling more precise access control.
2. **How It Works**:
    - When a new medical record is uploaded to the system, an AI/ML model will analyze its contents.
    - Based on predefined categories such as cardiology, ENT, orthopedics, etc., the system will automatically classify the record and assign it a hierarchical tag.
3. **Access Control Based on Hierarchy**:
    - When granting access to a healthcare provider (e.g., a doctor), the system will take into account their specialization (e.g., cardiologist, ENT specialist).
    - Doctors will only have access to records within their hierarchy level or specialization, ensuring that they see only the relevant files.

**Advantages of AI/ML Integration:**

- **Improved Accuracy**: AI/ML models can quickly and accurately analyze record contents, minimizing human error.
- **Enhanced Privacy**: By limiting access based on hierarchy and specialization, patient privacy is further safeguarded.
- **Automation**: Reduces manual intervention in assigning access levels or categories, making the system more efficient.
- **Scalability**: The system can easily handle large datasets and complex hierarchies as it grows.

**Implementation Roadmap:**

1. **Phase 1**: Develop and train an AI/ML model to classify records based on their content.
2. **Phase 2**: Integrate the classification module with the blockchain-based smart contract system to include hierarchical tagging.

3. **Phase 3**: Design and implement a mechanism to define and enforce access rules dynamically based on hierarchy tags and doctor specializations.
4. **Phase 4**: Test the system with real-world scenarios to ensure accuracy and security.

**Long-term Vision:**
This hierarchical data sharing approach will not only improve the usability and functionality of the current system but also pave the way for broader AI/ML adoption in decentralized healthcare systems. It will ensure that patient records are handled with the utmost precision, privacy, and relevance.

## Solution for Emergency Access to Patient Records

1. **Objective**:
   - To provide a mechanism for healthcare providers to access critical patient records during emergencies when the patient is unconscious or unable to provide explicit consent.
2. **How It Works**:
   - **Emergency Access Trigger**: In an emergency, authorized healthcare providers can request temporary access to the patient's medical records.
   - **Emergency Key Activation**:
     - The system generates a one-time **Emergency Access Key (EAK)** that is encrypted with the provider's public key and approved via the patient's pre-set consent conditions (e.g., an emergency contact or predefined smart contract rules).
   - **Audit Trail**:
     - Every emergency access request and its approval is logged immutably on the blockchain.
     - The system ensures that these records are flagged and available for review by the patient after the emergency is resolved.
3. **Safeguards**:
   - Emergency access is strictly time-limited and scope-limited, ensuring that only relevant records are accessible for a specific duration.
   - Misuse or unauthorized attempts are flagged, and strict penalties are enforced via the blockchain system's governance protocols.

**Advantages**:
- **Life-saving**: Ensures that critical patient information is available when needed most.
- **Transparency**: Logs every access attempt and ensures accountability.
- **Compliance**: Adheres to privacy standards by implementing time-limited and need-based access only.

**Implementation Roadmap**:

1. **Phase 1**: Define emergency access rules and conditions with patient consent.
2. **Phase 2**: Develop the smart contract module for generating and managing Emergency Access Keys (EAK).
3. **Phase 3**: Integrate with the auditing and notification system to ensure transparency.
4. **Phase 4**: Conduct rigorous testing under simulated emergency scenarios.

**Long-term Vision**:

The emergency access mechanism will complement the hierarchical data-sharing system by adding flexibility to handle unforeseen situations without compromising patient privacy. It ensures a balance between security and accessibility in life-critical scenarios, enhancing trust in decentralized healthcare solutions.

# CHAPTER 10

# CONCLUSION

The Medical Blockchain project signifies a transformative step in addressing the long-standing challenges of secure and efficient Electronic Health Record (EHR) management. By utilizing blockchain technology in conjunction with decentralized storage solutions like IPFS, the system ensures unparalleled levels of data integrity, security, and privacy. Unlike traditional centralized systems, this innovative approach eliminates single points of failure, making medical data resilient to breaches and tampering. At the heart of the system is patient empowerment, allowing individuals full control over their medical records and granting or revoking access as needed through smart contracts. This role-based access mechanism not only strengthens trust among stakeholders but also streamlines the process of data sharing in a secure and transparent manner. Furthermore, the integration of encryption techniques ensures that sensitive medical data remains protected while adhering to global standards such as HIPAA. The decentralized nature of the architecture enhances scalability and efficiency, enabling the system to accommodate an ever-growing number of users without compromising performance.

Beyond its technical achievements, the project lays a strong foundation for revolutionizing healthcare ecosystems globally. By facilitating seamless interoperability among healthcare providers, it enables improved care coordination and fosters better patient outcomes. Its applications extend to empowering medical research by offering access to anonymized data while ensuring privacy, paving the way for advancements in personalized medicine and predictive analytics. The system also holds promise for real-time health monitoring, secure insurance claims processing, and enhanced pharmaceutical supply chain management, demonstrating its far-reaching impact across the healthcare domain. Moreover, by reducing dependency on intermediaries, the system achieves cost-effectiveness while enhancing efficiency, making it a sustainable and scalable solution.

Looking to the future, this project has the potential to evolve further by integrating artificial intelligence for predictive diagnostics and early disease detection, incorporating IoT devices for continuous health monitoring, and exploring advanced privacy-preserving techniques like zero-knowledge proofs. It also provides opportunities to improve transaction costs through Layer 2 blockchain solutions and extend its reach for cross-border healthcare data sharing. In conclusion, the Medical Blockchain project not only addresses the pressing need for secure EHR management but also charts a visionary path for a decentralized and patient-centric healthcare ecosystem, setting the stage for innovations that will redefine the future of medical data management.

# BIBILIOGRAPHY

[1] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf.
[2] G. Zyskind, O. Nathan, and A. Pentland, "Decentralizing privacy: Using blockchain to protect personal data," in 2015 IEEE Security and Privacy Workshops, 2015, pp. 180-184. https://doi.org/10.1109/SPW.2015.27
[3] X. Yue, H. Wang, D. Jin, M. Li, and W. Jiang, "Healthcare data gateways: Found healthcare intelligence on blockchain with novel privacy risk control," Journal of Medical Systems, vol. 40, no. 10, pp. 1-8, 2016. https://doi.org/10.1007/s10916-016-0574-6
[4] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "MedRec: Using blockchain for medical data access and permission management," in 2016 2nd International Conference on Open and Big Data (OBD), 2016, pp. 25-30. https://doi.org/10.1109/OBD.2016.11
[5] Q. Xia, E. B. Sifah, A. Smahi, S. Amofa, and X. Zhang, "BBDS: Blockchain-based data sharing for electronic medical records in cloud environments," Information, vol. 8, no. 2, p. 44, 2017. https://doi.org/10.3390/info8020044
[6] P. Zhang, D. C. Schmidt, J. White, and G. Lenz, "Blockchain technology use cases in healthcare," in Advances in Computers, vol. 111, Elsevier, 2018, pp. 1-41. https://doi.org/10.1016/bs.adcom.2018.03.006
[7] F. Anwar and A. Shamim, "Barriers in adoption of health information technology in developing societies," International Journal of Advanced Computer Science and Applications, vol. 2, no. 8, pp. 40-45, 2011. https://doi.org/10.14569/IJACSA.2011.020806
[8] G. Wood, "Ethereum: A Secure Decentralized Generalized Transaction Ledger," Ethereum Project Yellow Paper, vol. 151, pp. 1-32, 2014.
[9] M. J. A. F. M. Almeida and R. M. L. Lima, "The Role of Blockchain in Personalized Medicine," *Computational and Structural Biotechnology Journal*, vol. 18, pp. 1298-1306, 2020.
[10] H. J. Kim and D. K. Kim, "Blockchain Technology for Pharmaceutical Supply Chain: A Review," *Sustainability*, vol. 12, no. 6, p. 2602, 2020.
[11] J. Wang, C. Guo, and Y. Zhang, "Managing Patient Consent in Blockchain-Based Health Information Exchange," in *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2021, pp. 789-794.
[12] T. S. B. Doan, "Population Health Management in the Era of Blockchain," *International Journal of Health Services*, vol. 51, no. 2, pp. 242-247, 2021.

# APPENDICES

## Appendix: Glossary of Terms

1. **Blockchain**: A decentralized ledger technology that records transactions securely and transparently.
2. **Smart Contracts**: Self-executing contracts with predefined rules and conditions stored on the blockchain.
3. **AES (Advanced Encryption Standard)**: A symmetric encryption algorithm used to securely encrypt data.
4. **RSA (Rivest-Shamir-Adleman)**: An asymmetric cryptographic algorithm used for secure key exchange.
5. **SHA-256 (Secure Hash Algorithm 256-bit)**: A cryptographic hash function that generates a fixed-size 256-bit digest.
6. **IPFS (InterPlanetary File System)**: A peer-to-peer network for storing and sharing files in a distributed manner.
7. **Metamask**: A browser extension used as a cryptocurrency wallet to interact with Ethereum blockchain applications.
8. **Decentralized Storage**: A method of storing data across multiple nodes in a network to ensure availability and security.
9. **Public Key Cryptography**: A cryptographic system that uses key pairs—public and private keys—for secure communication.
10. **Role-Based Access Control (RBAC)**: A method of restricting access to authorized users based on their roles.