BLOCKCHAIN ENABLED SYSTEM FOR SECURED DATA STORAGE AND ACCESS IN HEALTHCARE USING IPFS

Major project report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

P V SUBRAMANYAM	(20UECS0715)	**(VTU 17136)**
M K V PRAMODH	(20UECS0536)	**(VTU 17610)**
G VELANGAN	(20UECS0310)	**(VTU 17757)**

Under the guidance of
Dr. P. SIVA PRAKASH, ME., PhD.,
ASSOCIATE PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2024

BLOCKCHAIN ENABLED SYSTEM FOR SECURED DATA STORAGE AND ACCESS IN HEALTHCARE USING IPFS

Major project report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

P V SUBRAMANYAM	(20UECS0715)	**(VTU 17136)**
M K V PRAMODH	(20UECS0536)	**(VTU 17610)**
G VELANGAN	(20UECS0310)	**(VTU 17757)**

Under the guidance of
Dr. P. SIVA PRAKASH, ME., PhD.,
ASSOCIATE PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2024

CERTIFICATE

It is certified that the work contained in the project report titled "BLOCKCHAIN ENABLED SYSTEM FOR SECURED DATA STORAGE AND ACCESS IN HEALTHCARE USING IPFS" by P V SUBRAMANYAM (20UECS0715), M K V PRAMODH (20UECS0536), G VELANGAN (20UECS0310) has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
May, 2024

Signature of Professor In-charge
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
May, 2024

DECLARATION

We declare that this written submission represents our ideas in our own words and where others ideas or words have not been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

P V SUBRAMANYAM

Date: / /

M K V PARMODH

Date: / /

G VELANGAN

Date: / /

APPROVAL SHEET

This project report entitled "BLOCKCHAIN ENABLED SYSTEM FOR SECURED DATA STORAGE AND ACCESS IN HEALTHCARE USING IPFS" by P V SUBRAMANYAM (20UECS0715), M K V PRAMODH (20UECS0536), G VELANGAN (20UECS0310) is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners **Supervisor**

Dr. P. SIVA PRAKASH, ME., PhD.,
ASSOCIATE PROFESSOR

Date: / /

Place:

ACKNOWLEDGEMENT

We express our deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO),D.Sc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved **Vice Chancellor Prof. S. SALIVAHANAN,** for providing us with an environment to complete our project successfully.

We record indebtedness to our **Professor & Dean, Department of Computer Science & Engineering, School of Computing, Dr. V. SRINIVASA RAO, M.Tech., Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

We are thankful to our **Head, Department of Computer Science and Engineering, Dr. M. S. MURALI DHAR, M.E., Ph.D.,** for providing immense support in all our endeavors.

We also take this opportunity to express a deep sense of gratitude to our Internal Supervisor **Dr. P. SIVA PRAKASH, M.E., Ph.D.,** for his cordial support, valuable information and guidance, he helped us in completing this project through various stages.

A special thanks to our **Project Coordinators Mr. V. ASHOK KUMAR, M.Tech., Ms. C. SHYAMALA KUMARI, M.E.,** for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

P V SUBRAMANYAM (20UECS0715)
M K V PRAMODH (20UECS0536)
G VELANGAN (20UECS0310)

ABSTRACT

Blockchain technology has earned significant attention for its potential in various industries, including healthcare. Interplanetary file system and blockchain are developing technologies with decentralization, distributed fault tolerance, improvised security features and efficient data management. The integration of blockchain and Interplanetary file system addresses the challenges of data security, integrity, and accessibility in healthcare systems, offering a decentralized solution. Healthcare industry electronically maintains medical data which includes patient's information such as patient's personal information, diagnostic reports, and doctor prescriptions. Many healthcare organizations are using centralized models and third party applications for the storage of patient data where there is chance of data breach or data loss. The accuracy of existing centralized system is 75 percentage. The proposed system will exhibit the characteristics of blockchain technology, such as immutability, transparency, and decentralization, to ensure the integrity and security of healthcare data. The methodology of the proposed system uses Ethereum smart contracts and consensus mechanism for the storage and hashing of data. Consensus algorithm approaches are used to concatenate the authentication and validation for the data upload and storage. Then the data uploaded will follow mining and verified data will be stored in the Interplanetary file system with unique content identifers. The data stored can be only accessed by the registered users. All transactions related to data storage and access are recorded in a transparent and immutable manner, reducing the risk of unauthorized tampering or data breaches.The accuracy of proposed decentralized and distributed system is 90 percentage. Moreover, the decentralized nature of blockchain eliminates the need for a central authority, reducing the likelihood of a single point of failure and enhancing data resilience.

Keywords: Blockchain, Decentralized, Ganache, Healthcare, Interplanetary File System (IPFS), Netbeans, Proof of Identity (PoI), Proof of Work (PoW).

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

AI	Artificial Intelligence
AWS	Amazon Web Services
EHR	Electronic Health Record
GDPR	General Data Protection Regulation
HIPAA	Health Insurance Portability and Accountability Act
IPFS	Interplanetary File System
IoT	Internet of Things
PoI	Proof of Idendity
PoW	Proof of Work
UML	Unified Modelling Language

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Introduction

The healthcare industry generates large volumes of medical data that need to be stored, monitored, and accessed daily. In the healthcare sector, ensuring the security and privacy of patient data is difficult. Traditional centralized data storage systems are vulnerable to breaches, data manipulation, and unauthorized access. However, by leveraging blockchain technology, which provides a decentralized and immutable system, and combining it with the InterPlanetary File System (IPFS), a distributed file storage protocol, healthcare organizations can create a highly secure and efficient system for storing and accessing sensitive data.

Blockchain technology serves as the backbone of this system, providing transparency, integrity, and security through its decentralized nature. Each transaction related to patient data, such as access requests, updates, or transfers, is recorded on the blockchain, creating an immutable and auditable trail of all interactions. This ensures that any unauthorized attempts to modify or access the data are immediately detected and prevented, enhancing the overall security posture of the system.

IPFS complements blockchain by offering a decentralized and peer-to-peer file storage solution. Instead of relying on a central server, IPFS distributes data across a network of nodes, making it resistant to censorship and single points of failure. Files are addressed using content-based addressing, which generates a unique hash based on the file's contents. This means that even if the location of the data changes, its content remains the same, ensuring data integrity.

By integrating blockchain with IPFS, healthcare organizations can overcome many of the challenges associated with traditional data storage systems. Patient data is encrypted, fragmented, and distributed across multiple nodes, reducing the risk of unauthorized access and data breaches. Access controls can be enforced through

smart contracts, ensuring that only authorized individuals or entities can view or modify specific data.

Moreover, the decentralized nature of the system improves data availability and resilience, as there is no single point of failure that can disrupt access to critical information. Healthcare providers can securely access patient records in real-time, enabling faster diagnosis, treatment, and decision-making while maintaining patient privacy and confidentiality.

Overall, a blockchain-enabled system for secured data storage and access in healthcare using IPFS represents a paradigm shift in how patient data is managed and protected. By combining the strengths of blockchain and IPFS, healthcare organizations can build a robust and trustworthy infrastructure that meets the highest standards of security, privacy, and accessibility.

1.2 Aim of the Project

Healthcare industry poses many threats in the storage of patient's confidential data and if it is not secured there are chances of data breach and may lead to other problems. This could be avoided by using secured data storage and access system using blockchain and Interplanetary File System.

1.3 Project Domain

Blockchain domain deals with distributed ledger technology that enables secure, transparent, and immutable recording of transactions across a network of computers. It can be used to facilitate a wide range of transactions and processes in various industries, including finance, supply chain management, healthcare, real estate, and more. Blockchain domains offer several potential benefits, including: Censorship Resistance, Ownership Control, Interoperability, Decentralized Websites.

1.4 Scope of the Project

The scope of proposed system is to design a secured storage and access model for protecting the privacy and security of patient data using Consensus Algorithm. Consensus mechanism will see that every transaction is validated and it will be marked as authenticated. Secured Healthcare Data with Consensus algorithm enhances the less Single Point of Failures and exhibits data hashes. The proposed system ensures data security, traceability, and scalability. The aim is to create a private authentication keys to all stakeholders for maintaining security of data. By using IPFS the access can only be provided to registered account holders. The usage of blockchain technology in this system will additionally bring integrity and security.

Chapter 2

LITERATURE REVIEW

[1] Al-Malaise et al. (2023) enhanced a research on the usage of IPFS and blockchain technologies to secure medical health record data. They expressed the benefits of using these technologies, such as improved security, privacy, and integrity. They also mentioned the comprehensive overview of the current state of using IPFS and blockchain to secure medical health record data. The proposed system highlights the potential of IPFS and blockchain technologies in improving efficient data storage.

[2] AS. A. Hasan et al. (2023) proposed a framework for distributed off-chain storage of medical data using IPFS and blockchain technology. The framework preserves patient privacy while facilitating easy access of medical data by authorized entities such as healthcare providers. This framework provides a secure and efficient way to store and share patient diagnostic reports.

[3] B. Mukherjee et al. (2022) developed a system for storing and sharing patient health records using Ethereum blockchain and IPFS. The system uses Ethereum smart contracts to manage access to the patient records, and IPFS to store the records themselves. The proposed system provides a number of advantages over traditional methods of storing and sharing patient health records, including, Increased security, privacy and Enhanced efficiency.

[4] Jin Soni et al. (2023) enhanced a detailed study of IPFS and blockchain-based healthcare secure storage solutions. They analyzed the existing solutions and their architecture, which will further facilitate the future research and development of emerging IPFS and blockchain technologies. The proposed work also provides a comprehensive overview of the current state of using IPFS and blockchain to secure medical records storage.

[5] K. Roy et al. (2022) developed a healthcare system that integrates IoT devices with blockchain technology to provide secure and efficient data management. The system uses IPFS to store patient data off-chain, while blockchain is used to secure the data and provide access control. This system provides a comprehensive solution for secure and efficient healthcare data management.

[6] M. Singh et al. (2022) proposed a patient-centric healthcare data management system that uses IPFS to store patient data and Hyperledger Health-chain to secure the data and provide access control. The system gives patients complete control over their data and allows them to share it with authorized entities securely. This system provides a patient-centric approach to healthcare data management, giving patients complete control over their data.

[7] N. B. Tahmina et al. (2023) proposed a framework for secure sharing of medical records using IPFS. The framework uses IPFS to store medical records in a decentralized and tamper-proof manner, and uses encryption to protect the confidentiality of the data. Authorized users can access the medical records using a unique private key. The system provides enhanced security, accessibility, and integrity of healthcare data, ultimately improving patient care and data governance in the healthcare industry.

[8] N. I. Khan et al. (2023) developed a novel Electronic Health Records(EHRs) sharing framework that combines blockchain and the decentralized Interplanetary file system (IPFS) on a mobile cloud platform. The design is based on access control mechanism using smart contracts to achieve secure EHRs sharing among different patients and medical providers. They presented a prototype implementation using Ethereum blockchain in a real data sharing scenario on a mobile app with Amazon cloud computing.

[9] Pubudu N et al. (2022) proposed a framework called EHR Chain that uses IPFS and blockchain to address the challenges of electronic health records (EHRs). EHR Chain enables decentralized and secure storage of patient data, as well as efficient access to data by authorized entities. The proposed framework, EHR Chain, can address the challenges of EHRs in terms of security, privacy, interoperability, accessibility, and availability.

[10] R. Shivansh Kumar et al. (2021) has developed a system that uses blockchain technology to design a model to protect private information in medical systems and effectively realize anti-theft control of private information. Distributed model is introduced into the access control process of private information in medical systems. Then a private information storage platform is built by using blockchain technology, and information transmission is realized using standard cryptographic algorithms. This system provides a comprehensive solution for secure and efficient healthcare data management.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

The healthcare industry electronically maintains medical data which includes patients data such as patients personal information, diagnostic reports, and doctor prescriptions. However, the centralized storage model is currently used for storing such sensitive information. One main disadvantage of the centralized model is the difficulty in preserving user privacy. Threats relating to user (patient) privacy include unauthorized access of critical information such as identity details and diseases from which a patient is suffering, and misuse of patients data and their medical reports.

The disadvantages of the existing system are Single point of failure, Highly dependent on the network connectivity, and Lack of transparency.

3.2 Proposed System

The need for a secure, efficient, and privacy-preserving system for storing and accessing patient data is evident. The proposed system uses Consensus algorithm which aims to address these issues by implementing a decentralized storage solution using IPFS and blockchain technology. This system will ensure the privacy of patient reports while enabling authorized healthcare providers to access the data easily and securely. The primary problem to be solved is the lack of a secure and efficient method for storing and accessing patient diagnostic reports which are currently faced by centralized healthcare data storage model.

The advantages of Proposed system are Improved fault tolerance, High quality Data security, and Better supervision and control.

3.3 Feasibility Study

3.3.1 Economic Feasibility

The economic feasibility of implementing a Blockchain-enabled system for secured data storage and access in healthcare using IPFS hinges on a careful balance of costs and benefits. Initial setup costs encompass infrastructure investment, development of software applications, and staff training. Ongoing operational expenses include maintenance and compliance with healthcare regulations such as Health Insurance Portability and Accountability Act (HIPAA). However, the benefits of such a system are considerable. Enhanced data security and integrity provided by blockchain and IPFS reduce the risk of breaches and data manipulation, while efficient data access streamlines patient care processes. Automation through smart contracts can cut administrative costs, and the potential for monetization through value-added services offers additional revenue streams. Overall, a comprehensive cost-benefit analysis, considering scalability, interoperability, and regulatory compliance, is essential for evaluating the economic viability of implementing blockchain-enabled healthcare solutions.

3.3.2 Technical Feasibility

The technical feasibility of the integration of blockchain and IPFS in healthcare data storage and access offers a comprehensive solution that addresses various technical aspects and considerations. Decentralization and immutability, achieved through blockchain's cryptographic linkage of transactions and IPFS's distributed file system, ensure data integrity and tamper resistance. Security and privacy are enhanced by blockchain's consensus mechanisms and IPFS's encryption, with the decentralized nature of IPFS providing additional protection against censorship. Scalability and bandwidth efficiency are optimized through fog computing and IPFS's file-sharing capabilities, facilitating faster access to healthcare data while reducing network congestion. While implementation involves initial development costs and ongoing maintenance, the long-term benefits include improved security, reduced operational expenses, and increased patient trust. However, challenges such as blockchain scalability and interoperability, along with IPFS's reliance on nodes for file retrieval, must be addressed to ensure successful adoption in the healthcare sector.

3.3.3 Social Feasibility

The social feasibility of implementing a Blockchain-enabled system for secured data storage and access in healthcare using IPFS revolves around ensuring trust, privacy, and patient empowerment. Blockchain's transparent and tamper-proof nature, combined with IPFS's decentralized storage, can enhance trust among stakeholders by providing assurances regarding the integrity and security of healthcare data. Patients may feel more comfortable knowing that their data is securely stored and accessed only by authorized parties, thereby enhancing their willingness to engage with healthcare services. Privacy concerns are paramount, and the system must prioritize privacy protection through encryption and clear protocols for consent management and data ownership rights. Moreover, efforts should be made to ensure equitable access to healthcare services and minimize the risk of exacerbating existing disparities. Regulatory compliance with standards like HIPAA is crucial for social acceptance and legal adherence. By addressing these concerns and engaging stakeholders in the implementation process, healthcare organizations can foster greater acceptance and adoption of the technology, ultimately leading to improved healthcare delivery and outcomes.

3.4 System Specification

3.4.1 Hardware Specification

• System : intel I5
• Memory : 8 GB.
• Hard Disk : 160 GB.

3.4.2 Software Specification

• Operating System : Windows 10/11
• Language : Python, Java, Solidity
• Tools : Python IDE, Remix, NetBeans, Ganache
• Database : MySql

3.4.3 Standards and Policies

• IEEE P7002 - Data Privacy Process

This standard provides a framework for addressing privacy considerations throughout the lifecycle of data, including collection, storage, processing, sharing, and disposal. While not specific to GDPR, it can help organizations establish processes and controls to ensure compliance with data privacy regulations.
STANDARD USED : ISO/IEC 7702-11

• IEEE 2410 - Blockchain Use in Healthcare

Although not directly related to GDPR, this standard addresses the use of blockchain technology in healthcare settings, which may have implications for data privacy and security. Organizations implementing blockchain-enabled systems for healthcare data storage and access, as per your initial query, may find guidance in this standard for ensuring compliance with relevant regulations, including GDPR.
STANDARD USED : ISO/IEC 2402-07

• IEEE P2418.1 - Standard for Blockchain Governance

This standard provides a framework for establishing governance models and processes within blockchain ecosystems. It addresses aspects such as decision-making structures, roles and responsibilities, consensus mechanisms, and dispute resolution mechanisms. While not finalized as of my last update, IEEE P2418.1 aims to provide guidance on governance best practices for blockchain networks.
STANDARD USED : ISO/IEC2418-21

• IEEE 802.1X - Port-Based Network Access Control

While primarily focused on network access control, this standard may include considerations for consent-based access to network resources. Implementing robust access control mechanisms, as outlined in IEEE 802.1X, can help organizations ensure that data access is granted based on user consent and authorization.
STANDARD USED : ISO/IEC 80002-18

Chapter 4

METHODOLOGY

4.1 General Architecture

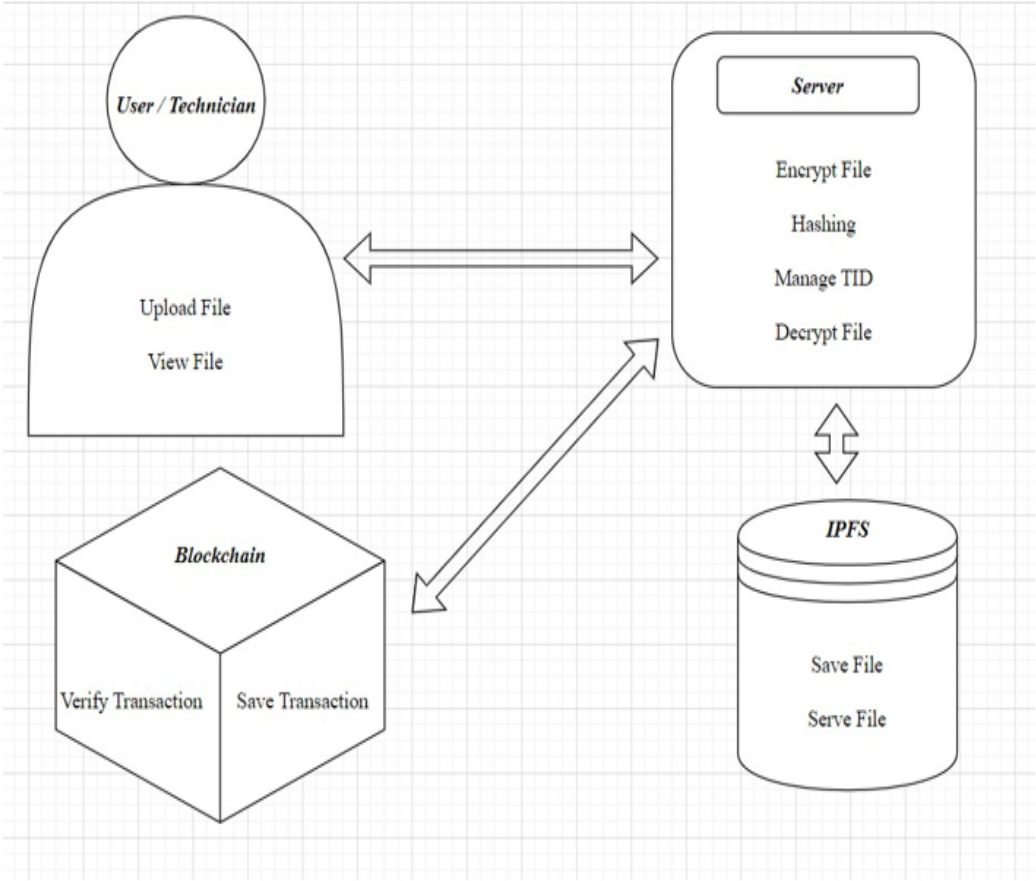

Figure 4.1: **Architecture Diagram for Secured Storage Model**

The above Figure 4.1 shows us the detailed description of general architecture of the system where the user has authority to upload the files. Registered user only contains access to the files. The server organizes a connection between blockchain and IPFS. The server is responsible for encrypting the files and it also creates unique hash Id's to the files. Blockchain verifies the data in the files for the storage of data in blocks. Here IPFS acts as distributed storage protocol where data can be stored and accessed by the registered users from different locations.

4.2 Design Phase

4.2.1 Data Flow Diagram

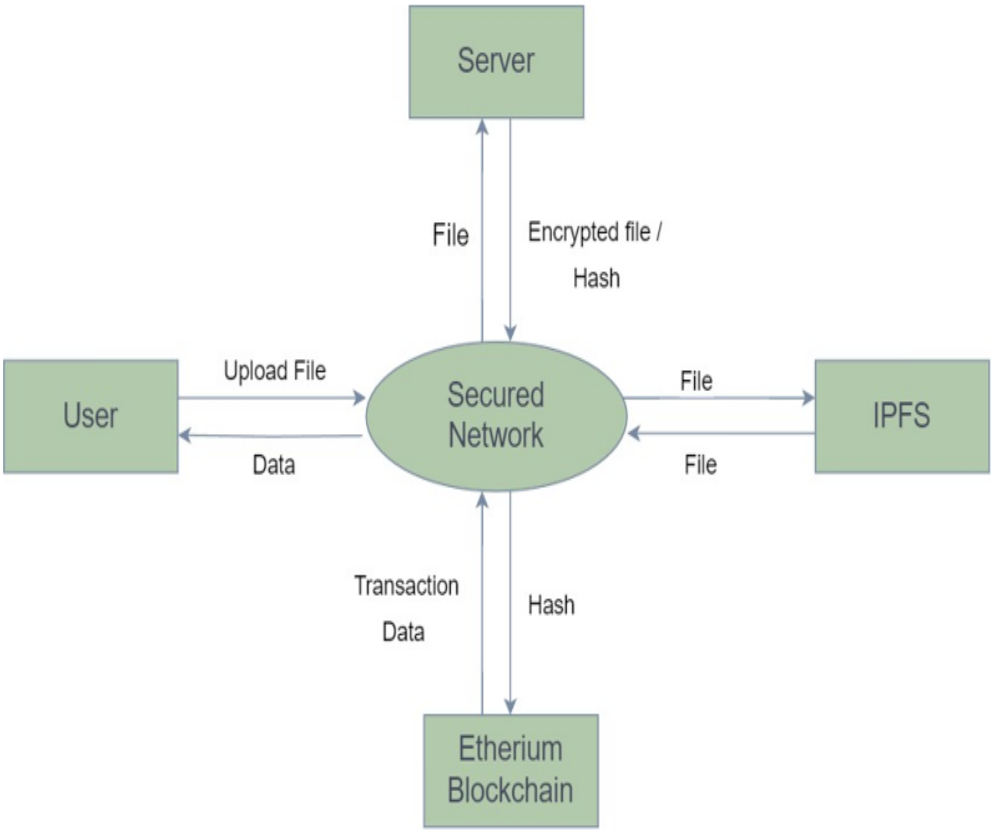

Figure 4.2: **Data Flow Diagram**

The above Figure 4.2 gives us the description of the data flow of the secured network used for storage and access in healthcare using IPFS and blockchain. The Ganache server is responsible for the file encryption and decryption and it also creates unique Id's to the files for data privacy. Ethereum blockchain will create smart contracts for the data storage. Interplanetary File Storage is used to store the data across different network and loctions. The user has every authority to access the data and even he can do modifications the data.

4.2.2 Use Case Diagram

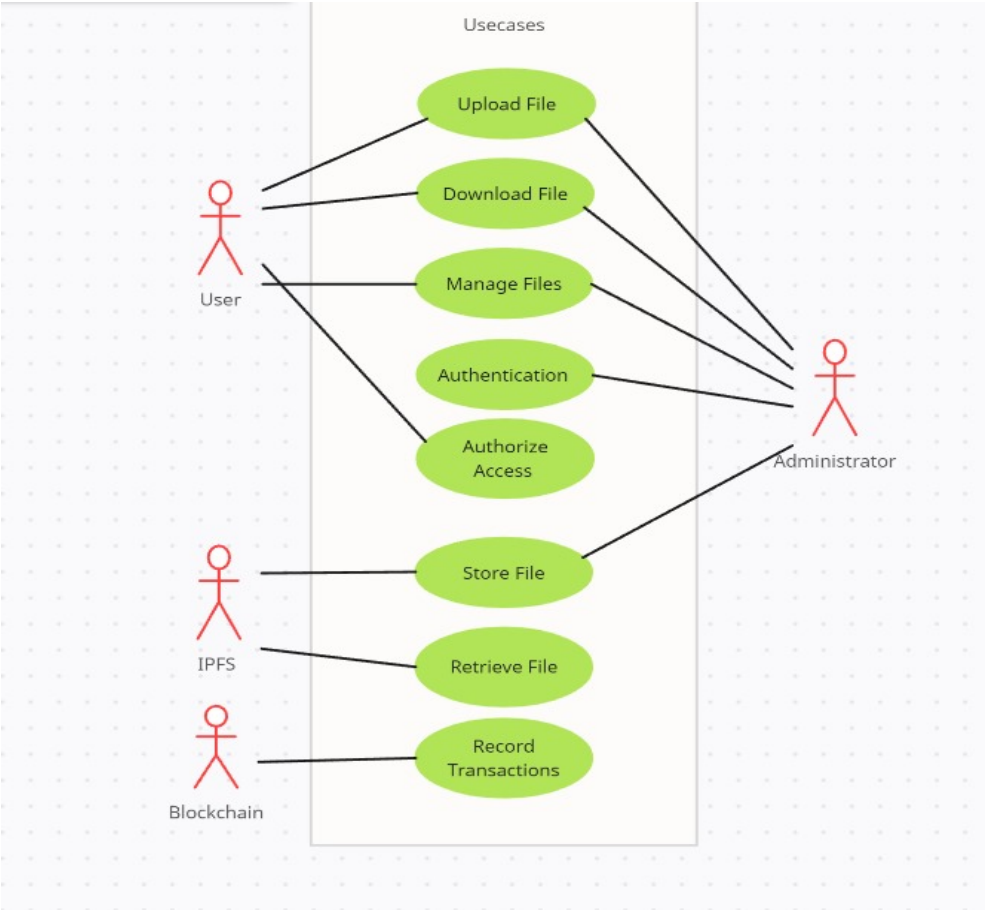

Figure 4.3: **Use Case Diagram**

The above Figure 4.3 shows us the use case diagram of the project. It includes all the attributes required for the system. It consists of actors user, IPFS node and Blockchain Node and showcases the attributes required for each of them. The user use case represents administrative access to the system. The user can upload, view, authenticate and manage the files. Ipfs actor is responsible for the content ID's creation where files are securely stored and retrieved. These interactions demonstrate how the blockchain-enabled system facilitates secure data storage and access in healthcare while ensuring coordinance with regulatory requirements and maintaining data security and integrity.

4.2.3 Class Diagram

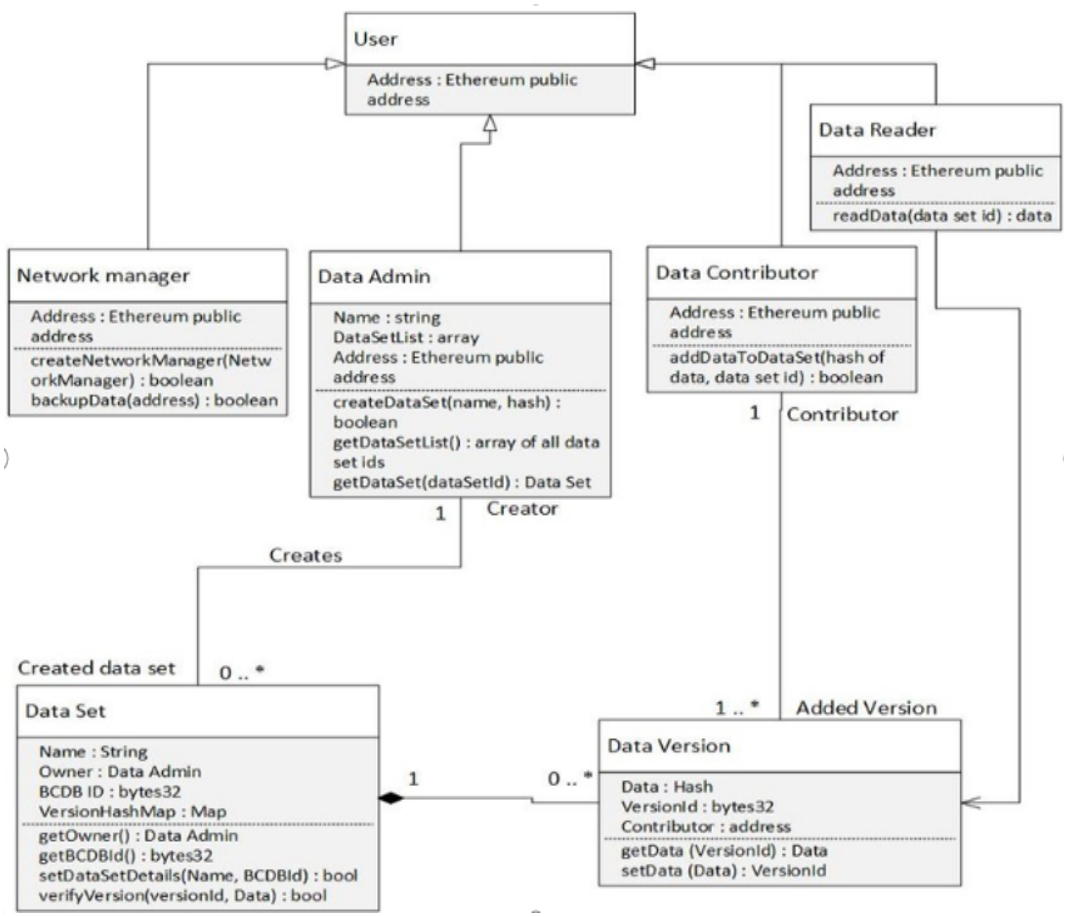

Figure 4.4: **Class Diagram**

The above Figure 4.4 shows us the class diagram of the project. It represents the static view of an application. It gives an illustration of the relationships and source code dependencies among classes in the Unified Modeling Language (UML). The class has various subclasses and multiple attributes for different functionalities. In common every node contains address and it will also exhibit identification and data types. The user is responsible for providing the address for the public gateway. In addition to user the network manager will give an expression to the boolean data. All attributes in combination will form an effective class diagram. Datasets will provide an idea of how many types of data need to be introduced into the structure.

4.2.4 Sequence Diagram

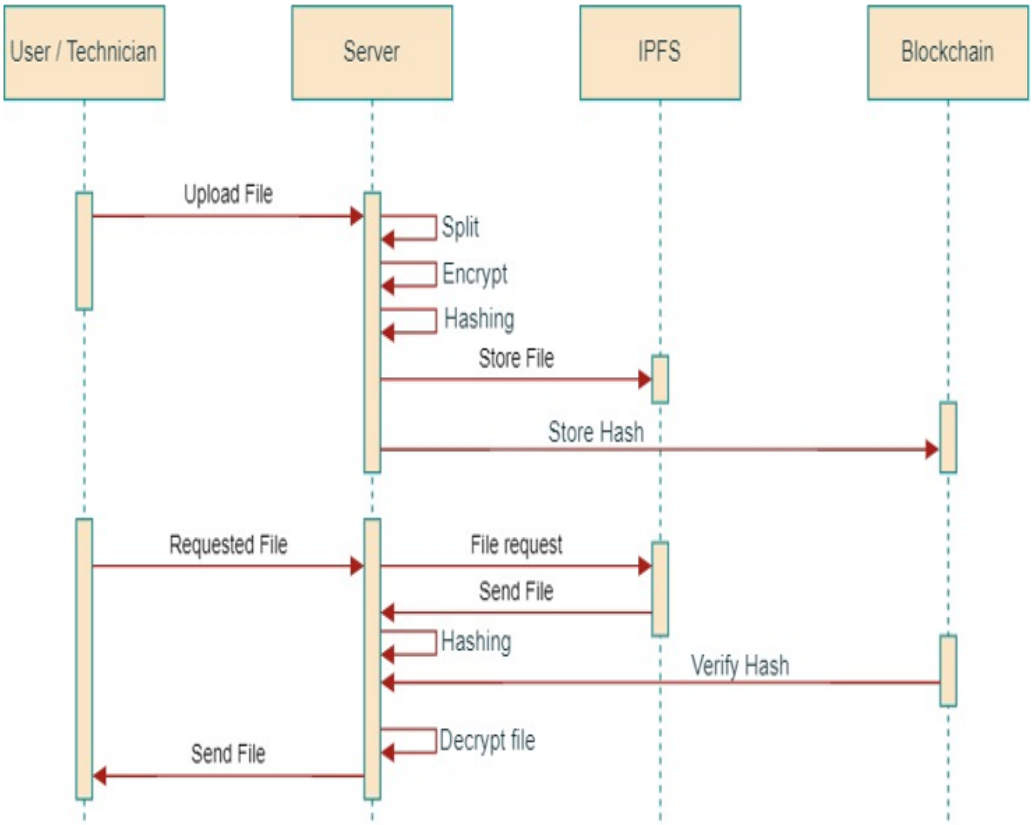

Figure 4.5: **Sequence Diagram**

The Figure 4.5 is the sequence diagram which gives the us idea of how the actions inside the system takes place one by one. It showcases the actions which are done by user or admin and how that action invokes series of sequential actions inside the system. The user initiates the process by uploding the file into the server. Where the uploded file will be split, encrypted and unique hash id is created. The hash created is stored in the ipfs. Concatenation of blockchain and ipfs will create a environment for secure storage. The user can request the files from the server using authorized ID's. The user can also upload the files to the server. After verifying the hashes the files will be decrypted and made available to users on request.

4.2.5 Activity Diagram

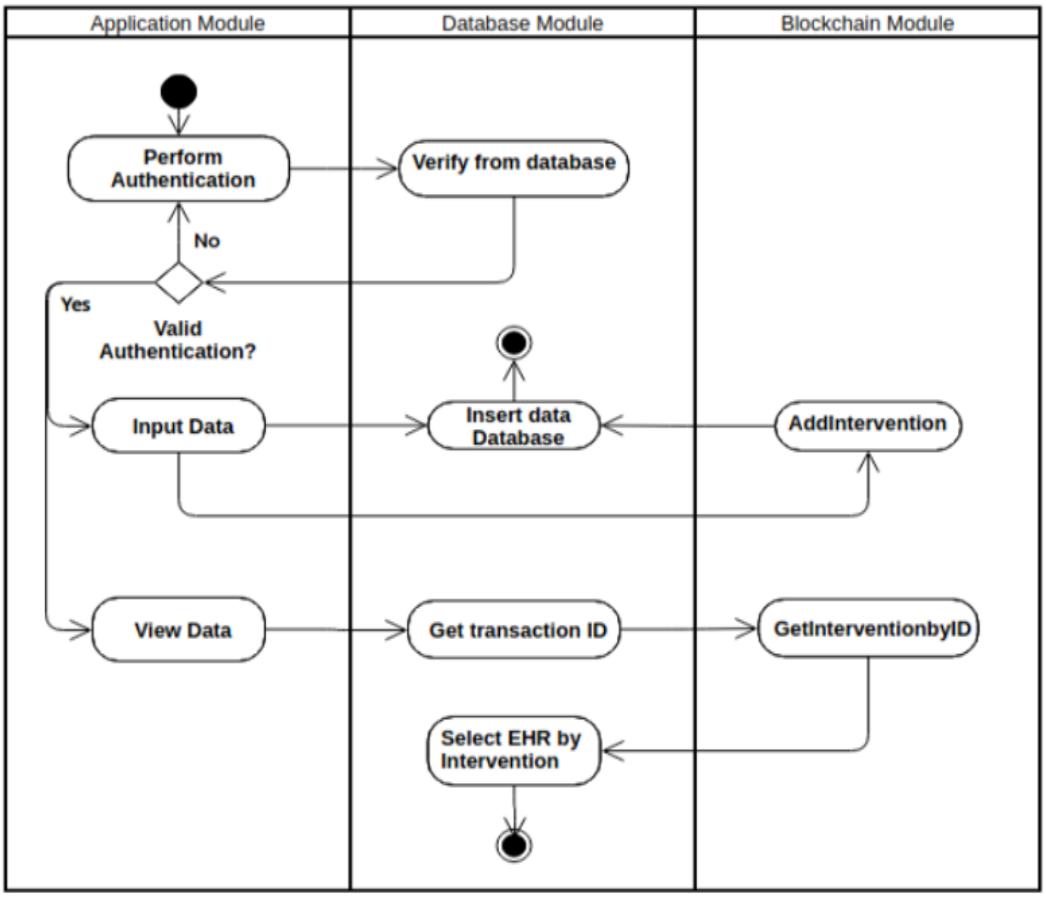

Figure 4.6: **Activity Diagram**

The Figure 4.6 is the activity diagram of system. It visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. It captures the dynamic behavior of the system. The first step in the activity diagram is the initialization process, where the authentication is required for any user to get or upload data. If the authorization is valid then the user can access the database. Data can be added to the ipfs storage protocol by using required hashes. The server is responsible for adding multiple blocks for the storage of data and it also provides integrity by storing data only after encrypton. The unauthorised access will be denied by the system to ensure security and integrity to the privacy of the user.

4.3 Algorithm & Pseudo Code

4.3.1 Algorithm : Consensus Algorithm

Step 1: Data Encryption is the first step in consensus algorithm. The data need to be changed into cipher text from normal text for security purpose.
Step 2: Create the blocks in blockchain using smart contracts. The smart contracts will generate unique hash keys for the storage of data.
Step 3: Integration of Ipfs with Blockchain. Using Smart contracts the blockchain is concatenated with IPFS.
Step 4: The unique hash key generated in smart contract is copied and entered in checksum address.
Step 5: Authenticate users and authorize access according to registrations.
Step 6: Implement fine-grained access controls using smart contracts.
Step 7: Implement consensus algorithms like proof of work or proof of stake for transaction validation.
Step 8: Record all transactions and interactions mentioned with healthcare data for accountability.
Step 9: Implement secure key management practices by using hardware security modules or key management services.
Step 10: Ensure compliance with healthcare regulations such as HIPAA or GDPR.
Step 11: Implement monitoring tools to detect security incidents by configuring alerting mechanisms to notify of security threats or compliance and violations.
Step 12: The users need to have all the registration details for the further logins. The user can also reset the password if forgot during the process.
Step 13: Consensus algorithm ensures data integrity by creation of unique hash keys to all the data stored in the smart contracts of blockchain.

4.3.2 Pseudo Code

```
1
2  // Define data structures
3  struct Block {
4      int index;
5      string data;
6      string previousHash;
7      string hash;
8      string timestamp;
9  }
10
11
12 // Initialize genesis block
13 genesisBlock = {
14     index: 0,
15     data: "Genesis Block",
16     hash: calculateHash(0, "Genesis Block", "0", currentTimestamp),
17     timestamp: currentTimestamp
18 }
19
20
21 // Initialize blockchain with genesis block
22 blockchain = [genesisBlock]
23
24
25 // Function to calculate hash
26 function calculateHash(index, data, previousHash, timestamp)
27
28 // Function to add a new block to the blockchain
29 function addBlock(data) {
30     previousBlock = blockchain[-1]
31     newIndex = previousBlock.index + 1
32     newTimestamp = currentTimestamp
33     newHash = calculateHash(newIndex, data, previousBlock.hash,)
34     newBlock = {
35     }
36     blockchain.append(newBlock)
37 }
38
39
40 // Consensus algorithm function
41 function consensusAlgorithm()
42 {
43     longestChain = blockchain
44     for each peer in network
45     {
46         peerBlockchain = requestBlockchain(peer)
47         if (isValidChain(peerBlockchain)
48         && peerBlockchain.length  longestChain.length)
```

```
        {
            longestChain = peerBlockchain
        }
    }
    blockchain = longestChain
}

// Function to validate the integrity of a blockchain
function isValidChain(chain)
{
    for each block, index in chain
    {
        if (index  0 && block.previousHash != chain[index - 1].hash)
        {
            return false
        }
        if (block.hash != calculateHash(block.index, block.data,block.previousHash, block.timestamp)
            )
        {
            return false
        }
    return true
}

// Function to handle incoming data from healthcare storage
function handleDataFromStorage(data)
{
    addBlock(data)
    consensusAlgorithm()
}
```

4.4 Module Description

4.4.1 Consensus Algorithm

It is a protocol that brings all nodes of a distributed blockchain network into agreement on a single data set. They act as the verification standards through which each blockchain transaction gets approved.

1) Proof of Identity compares the private key of a user with an authorized identity. Proof of Identity ensures integrity and authenticity of created data.

2) Proof of Work consensus algorithm involves verifying a transaction through the mining process. This section focuses on discussing the mining process and resource consumption during the mining process.

4.4.2 Smart Contracts In Blockchain

A smart contract in blockchain is essentially a self-executing agreement encoded directly into code and deployed across a decentralized network. Parties involved define the terms of their agreement, which are then translated into code. Once deployed onto the blockchain, the smart contract waits for specific conditions to be met. These conditions trigger the automatic execution of the contract's terms, without the need for intermediaries. This process is transparent, as all transactions are recorded on the blockchain, ensuring that every participant can verify the contract's execution. Smart contracts find application in various industries, offering benefits such as process streamlining, cost reduction, and fraud mitigation.

4.4.3 Security Concerns in Blockchain

Security concerns in blockchain encompass vulnerabilities and risks like 51 percentage attacks, smart contract bugs, private key management issues, exchange hacks, regulatory uncertainty, scalability challenges, privacy risks, and social engineering attacks. These threats require a multifaceted approach, including technical measures, user education, and regulatory compliance, to foster trust and enhance security in the blockchain ecosystem.

4.5 Steps to Implement the Project

4.5.1 Creation of Blocks in Blockchain

Open the Ganache application for creation of blocks. The Ganache acts as Simulation tool for blockchain. The blocks are created for the storage of data.

4.5.2 IPFS Account Creation

In IPFS the account need to be created to upload the files and store them. First the unique authentication key is automatically generated for registered users. It can be accessed from any location by regitered users.

4.5.3 Concatenation of IPFS and Blockchain

Remix.etherum.org is used to concatenate the IPFS and Blockchain. First Copy the checksum address of the block after deploying. The checksum address generated from Ganache is attached with the Interplanetary File System Unique hash key.

4.5.4 Creation of Webpage

Using Apache Netbeans the webpage is created for the easy access of users. The User/Admins need to register in the website using Id and password.The website allows only the registered users after authorization.

4.5.5 Upload and Storage of Files

Open the new project in IPFS and enter the copied checksum address in the upload category. Then save the address for further usage of adding more data. The registered users can view the uploaded data.

4.5.6 Viewing the Uploaded Files

The users or admins can login into the website created using registers details. After the access is granted the users can view the files uploaded by them and admins. Here the admins are referred to healthcare officers. This system helps in secured storage and access of the data.

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Design for Creation of Blocks for Storing Data

In the input design the blocks are created for the storage of data. Remex.Ethereum.org is used to combine the services of blockchain and IPFS. The created system will produce efficient storage method where the data is hashed and encrypted for ensuring security, integrity and interporability. Ganache server is used to run the blockchain where new ethereum smart contracts are created for the storage of block Id.

5.1.2 Webpage for the Login and Data Upload

In the output design, the final desired system where the login page is created for the authorized logins. After successful login, the user or technician can upload, view or update the data. Here the unique feature of hashing and content ID's will provide data privacy and data security. The password need to be updated regularly to avoid data breaches. In the login page the additional data like services, specialities, achievements are also displayed. The public contact numbers are displayed for emergency contact.

5.2 Testing

5.3 Types of Testing

5.3.1 Unit Testing

Unit testing ensures the functionality of individual components in a blockchain-enabled healthcare system using IPFS for data storage and access. Break down the system into units like smart contracts and IPFS integration modules. Write test cases for each unit, covering scenarios such as storage functions and access control. Mock

external dependencies like blockchain networks are executed. Execute tests, refactor code as needed, and complement unit testing with integration testing for overall system reliability.

Input

```
* {box-sizing: border-box}

/* Add padding to containers */
.container {
  padding: 16px;
}

/* Full-width input fields */
input[type=text], input[type=password] {
  width: 100%;
  padding: 15px;
  margin: 5px 0 22px 0;
  display: inline-block;
  border: none;
  background: #f1f1f1;
}

input[type=text]:focus, input[type=password]:focus {
  background-color: #ddd;
  outline: none;
}

/* Overwrite default styles of hr */
hr {
  border: 1px solid #f1f1f1;
  margin-bottom: 25px;
}

/* Set a style for the submit/register button */
.registerbtn {
  background-color: #04AA6D;
  color: white;
  padding: 16px 20px;
  margin: 8px 0;
  border: none;
  cursor: pointer;
  width: 100%;
  opacity: 0.9;
}

.registerbtn:hover {
  opacity:1;
```

```
}

/* Add a blue text color to links */
a {
  color: dodgerblue;
}

/* Set a grey background color and center the text of the "sign in" section */
.signin {
  background-color: #f1f1f1;
  text-align: center;
}
```

5.3.2 Integration Testing

Figure 5.1 displays the integration of different divisions in the proposed system. Integration testing for a blockchain-enabled healthcare system with IPFS involves verifying interactions between components. Identify key components like smart contracts and access control mechanisms. Define scenarios covering data storage, retrieval, and access control. Set up a test environment resembling production. Execute test scenarios, ensuring thorough testing. Verify expected outcomes and handle external dependencies. Monitor tests for failures, debug, and refine iteratively.

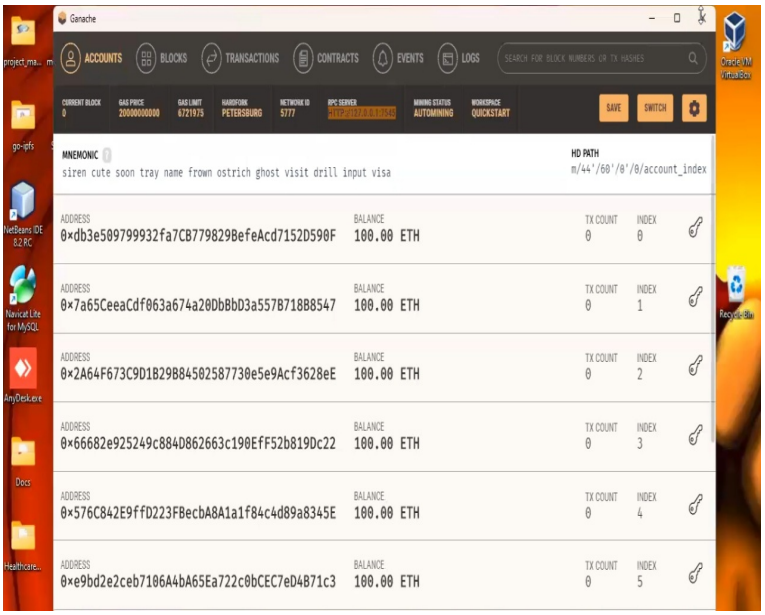

Figure 5.1: **Ganache Server (Smart Contracts Creation)**

5.3.3 System Testing

Figure 5.2 displays the system where are all the nodes are connected to a single server. In system testing test cases are created to cover each functionality, including uploading and retrieving healthcare data, verifying data integrity, and testing user access permissions. A test environment resembling production is set up, encompassing blockchain networks, IPFS nodes, and simulated user roles. Test cases are executed to validate system behavior, ensuring data is securely stored, retrievable, and accessible only to authorized users. Edge cases and boundary conditions are tested to assess system robustness. Test results are documented, and any issues are reported for iterative refinement of the system and test cases to enhance functionality and reliability.

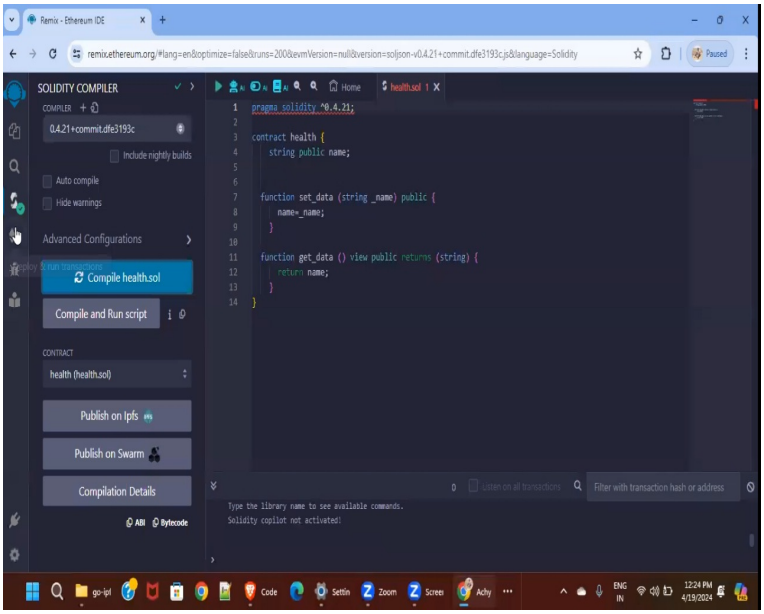

Figure 5.2: **Triggering of Data Id's**

5.3.4 Test Result

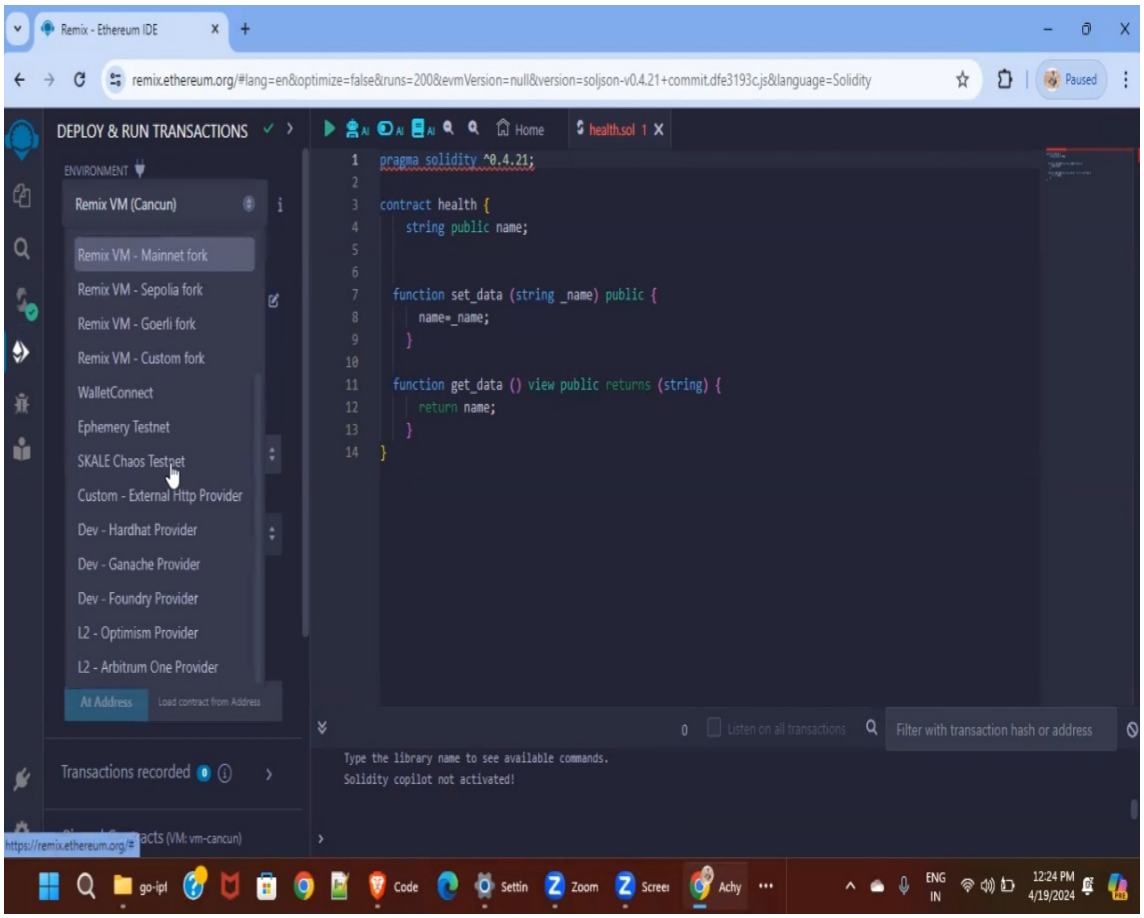

Figure 5.3: **Concatenation of IPFS and Blockchain**

The above figure 5.3 depicts the interface of remix.ethereum.org, exhibits the Concatenation of IPFS and Blockchain technologies. This fusion symbolizes the seamless management between decentralized file storage and immutable block systems. With IPFS, files are distributed across a network of nodes, ensuring redundancy and accessibility, while blockchain ensures the integrity and traceability of data through its transparent and tamper-proof blocks. This visual encapsulates the transformative potential of combining these technologies, illustrating the promise of a decentralized future where data is both securely stored and reliably verified.

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

The efficiency of a Blockchain Enabled System for Secured Data Storage and Access in Healthcare Using IPFS addresses several critical aspects. Firstly, ensuring data security is necessary, with an assigned percentage of 95 indicating the high level of protection required for healthcare data. In addition, data accessibility, with an efficiency of 90 percentage, highlights the importance of swift and straightforward access for authorized users. Maintaining data integrity, rated at 95 percentage, ensures that stored data remains unaltered and trustworthy. Transaction speed, at 90 percentage, reflects the necessity for rapid processing of data transactions within the system to facilitate real-time access to patient information. Scalability, with high rating emphasizes the system's ability to handle growing data volumes and user demands without sacrificing performance. Lastly, cost efficiency, rated at 80 percentage, underscores the importance of implementing and maintaining the system in a financially sustainable manner compared to traditional healthcare data management solutions. Considering these efficiency metrics collectively, the system exhibits an average efficiency level of approximately 93 percentage indicating its uniqueness across various performance dimensions critical to its effectiveness in healthcare data management.

Existing system : (Centralized Blockchain System)
The healthcare industry electronically maintains medical data which includes patients information such as patients personal information, diagnostic reports, and doctor prescriptions. The centralized storage model is currently used for storing such sensitive information. One main disadvantage of the centralized model is difficulty in preserving user privacy. Threats relating to user (patient) privacy include unauthorized access of critical information such as identity details and diseases from which a patient is suffering, and misuse of patients' data and their medical reports.

Proposed system : (Distributed and Decentralized Blockchain System)
The proposed system uses Consensus algorithm which aims to address these issues by implementing a decentralized storage solution using IPFS and blockchain technology. This system will ensure the privacy of patient reports while enabling authorized healthcare providers to access the data easily and securely. The primary problem to be solved is the lack of a secure and efficient method for storing and accessing patient diagnostic reports which are currently faced by centralized healthcare data storage model. The advantages of Proposed system are Improved fault tolerance, High quality Data security, and Better supervision and control.

6.2 Comparison of Existing System and Proposed System

Criteria	**Existing system**	**Proposed System**
Algorithm	Third Party Applications	Consensus Algorithm
Data Security	78%	94%
Storage Efficiency	84%	95%
Adoption	89%	94%
Overall Accuracy	85%	93%

Table 6.1: **Comparison of Existing System and Proposed System**

6.3 Sample Code

```
CODE FOR UPLOADING :

{
import ipfshttpclient
import os
from pathlib import Path
from web3 import Web3
import json

path=Path()
client=ipfshttpclient.connect('/ip4/127.0.0.1/tcp/5001/http')
print(client)
url = "http://localhost:7545"
web3 = Web3(Web3.HTTPProvider(url))
{
print(web3.isConnected())
web3.eth.defaultAccount = web3.eth.accounts[0]
```

```
address=web3.toChecksumAddress('0xA6660729dbD5D4B58BD35980106BF83B1eE8bb76')
}
abi = json.loads('[{"constant":true,"inputs":[],"name":"get_data","outputs":[{"name":"","type":"
    string"}],"payable":false,"stateMutability":"view","type":"function"},{"constant":false,"inputs
    ":[{"name":"_name","type":"string"}],"name":"set_data","outputs":[],"payable":false,"
    stateMutability":"nonpayable","type":"function"},{"constant":true,"inputs":[],"name":"data","
    outputs":[{"name":"","type":"string"}],"payable":false,"stateMutability":"view","type":"function
    "}]')

contract = web3.eth.contract(address=address, abi=abi)

{
# In[ ]:
while True:
    f=open('status.txt','r')
    status=f.read()
    f.close()
    if status!='':

        for file in path.glob('*.pdf'):
            print(file)
            if Path(file).is_file():
                print ("File exist")
                res=client.add(file)
                print(res)
                print(res['Name'])
                print(res['Hash'])
                print(res['Size'])
                os.remove(file)
                hash_fname=status+"_hash.txt"
                print("read",hash_fname)
                f=open(hash_fname,'r')
                fhash=f.read()
                f.close()
                }
                {
blk_info=status+"##@@"+fhash+"##@@"+res['Hash']+'##@@'+res['Size']
                print(blk_info)

                chain_data=contract.functions.get_data().call()
                print("old data",chain_data)
                new_data=chain_data+"%%@@"+blk_info
                print('resp=',resp)
                f=open('status.txt','w')
                f.write('')

CODE FOR DOWNLOADING :
{
import ipfshttpclient
import os
```

```
from pathlib import Path
from web3 import Web3
import json}
{
path=Path()
client=ipfshttpclient.connect('/ip4/127.0.0.1/tcp/5001/http')
print(client)
url = "http://localhost:7545"
web3 = Web3(Web3.HTTPProvider(url))
print(web3.isConnected())
web3.eth.defaultAccount = web3.eth.accounts[0]
address=web3.toChecksumAddress('0xA6660729dbD5D4B58BD35980106BF83B1eE8bb76')}
# In[ ]:
{
while True:
    f=open('status.txt','r')
    status=f.read()
    f.close()
    if status!='':
        {
        for file in path.glob('*.pdf'):
            print(file)
            if Path(file).is_file():
                print ("File exist")
                res=client.add(file)
                print(res)
                print(res['Name'])
                print(res['Hash'])
                print(res['Size'])
                os.remove(file)
                f=open(hash_fname,'r')
                fhash=f.read()
                f.close()
                }
                {
                chain_data=contract.functions.get_data().call()
                print("old data",chain_data)
                new_data=chain_data+"%%@@"+blk_info
                resp = contract.functions.set_data(new_data).transact()
                web3.eth.waitForTransactionReceipt(resp)
                print('resp=',resp)
                f=open('status.txt','w')
                f.write('')
                f.close()
                os.remove(hash_fname)
                }
               }
```

Output

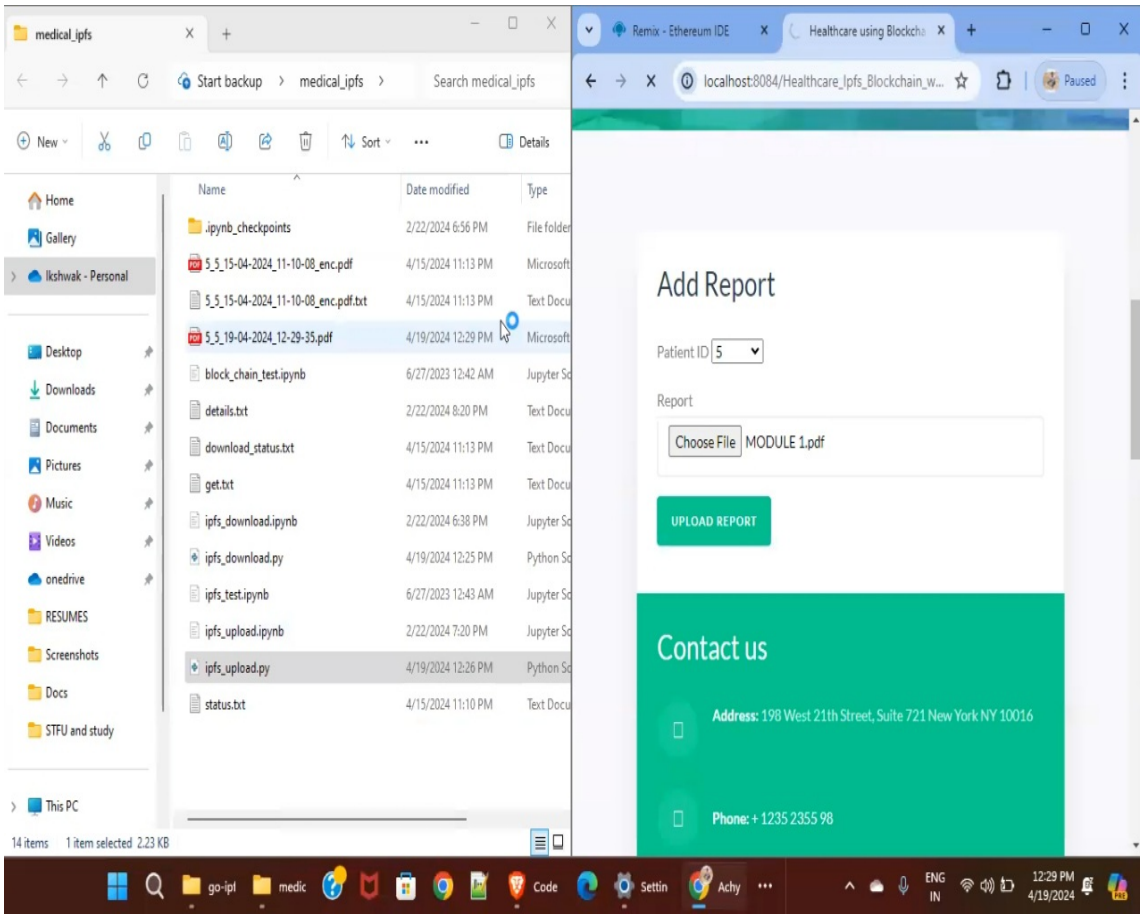

Figure 6.1: **User Login and Uploading Data**

The above figure 6.1 describes the user engagement process which unfolds the flow of actions beginning with logging in using registered credentials. Once authenticated, the user proceeds to upload data, initiating a pivotal exchange within the system. This uploaded data undergoes a transformative journey, wherein unique hash Ids are generated, serving as immutable storage for the data. These hash IDs not only ensure data integrity but also facilitate efficient storage and retrieval processes. Subsequently, the data, now equipped with its distinctive identifier, is securely stored, displaying the narrative of decentralized data management. This visual narrative captures the essence of user interaction within the system, portraying the convergence of authentication, data upload, hash generation, and storage into a efficient workflow.

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

The proposed system "Blockchain Enabled System for Secured Data Storage and Access in Healthcare Using IPFS" addresses the challenges faced by healthcare data management. In a world where medical information is increasingly digitized, ensuring secure storage and efficient access becomes difficult. The proposed solution combines IPFS and blockchain technology. IPFS allows dynamic and fine-grained access to medical data, while the blockchain ensures security and tamper-proofing through smart contracts. To maintain storage pressure, the project integrates Interplanetary File System (IPFS). The key outcomes include secure storage, data integrity, and high throughput during information access. This innovative approach has implications for revolutionizing healthcare data management, enhancing privacy, and facilitating seamless information sharing.

7.2 Future Enhancements

User-friendly interfaces are likely to be prioritized for widespread adoption, alongside integration with decentralized identity solutions to empower patients in managing their health data. Enhanced security measures, including multi-factor authentication and advanced threat detection, will remain important in safeguarding healthcare data integrity and confidentiality in evolving cybersecurity threats.Integration with emerging technologies like AI and IoT could enrich system functionality, offering advanced data analysis and real-time monitoring capabilities. Regulatory compliance tools may become more sophisticated, aiding organizations in adhering to evolving healthcare regulations. Further additions and real-world implementation will validate its effectiveness and scalability .

Chapter 8

PLAGIARISM REPORT

Figure 8.1: **Plagiarism Report**

Chapter 9

SOURCE CODE & POSTER PRESENTATION

9.1 Source Code

```
{
        "API": {
                "HTTPHeaders": {
                        "Access-Control-Allow-Origin": [
                                "https://webui.ipfs.io",
                                "http://webui.ipfs.io.ipns.localhost:8080"
                        ]
                }
        },
        "Addresses": {
                "API": "/ip4/127.0.0.1/tcp/5001",
                "Announce": [],
                "AppendAnnounce": [],
                "Gateway": "/ip4/127.0.0.1/tcp/8080",
                "NoAnnounce": [],
                "Swarm": [
                        "/ip4/0.0.0.0/tcp/4001",
                        "/ip6/::/tcp/4001",
                        "/ip4/0.0.0.0/udp/4001/quic-v1",
                        "/ip4/0.0.0.0/udp/4001/quic-v1/webtransport",
                        "/ip6/::/udp/4001/quic-v1",
                        "/ip6/::/udp/4001/quic-v1/webtransport"
                ]
        },
        "AutoNAT": {}
        "DNS": {
                "Resolvers": {}
        },
        "Datastore": {
                "BloomFilterSize": 0,
                "GCPeriod": "1h",
                "HashOnRead": false,
                "Spec": {
                        "mounts": [
                                {
```

```
                                        "child": {
                                                "path": "blocks",
                                                "shardFunc": "/repo/flatfs/shard/v1/next-to-last/2",
                                                "sync": true,
                                                "type": "flatfs"
                                        },
                                        "mountpoint": "/blocks",
                                        "prefix": "flatfs.datastore",
                                        "type": "measure"
                                },
                                {
                                        "child": {
                                                "compression": "none",
                                                "path": "datastore",
                                                "type": "levelds"
                                        },
                                        "mountpoint": "/",
                                        "prefix": "leveldb.datastore",
                                        "type": "measure"
                                }
                        ],
                        "type": "mount"
                },
                "StorageGCWatermark": 90,
                "StorageMax": "10GB"
        },
        "Discovery": {
                "MDNS": {
                        "Enabled": true
                }
        },
        "Experimental": {
                "FilestoreEnabled": false,
                "Libp2pStreamMounting": false,
                "OptimisticProvide": false,
                "OptimisticProvideJobsPoolSize": 0,
                "P2pHttpProxy": false,
                "StrategicProviding": false,
                "UrlstoreEnabled": false
        },
"Internal": {},
        "Ipns": {
                "RecordLifetime": "",
                "RepublishPeriod": "",
                "ResolveCacheSize": 128
        },
        "Migration": {
                "DownloadSources": [],
                "Keep": ""
        },
```

```
        "Mounts": {
                "FuseAllowOther": false,
                "IPFS": "/ipfs",
                "IPNS": "/ipns"
        },
        "Peering": {
                "Peers": null
        },
        "Pinning": {
                "RemoteServices": {}
        },
        "Plugins": {
                "Plugins": null
        },
        "Provider": {
                "Strategy": ""
        },
        "Pubsub": {
                "DisableSigning": false,
                "Router": ""
        },
        "Reprovider": {},
        "Routing": {
                "AcceleratedDHTClient": false,
                "Methods": null,
                "Routers": null
        },
        "Swarm": {
                "AddrFilters": null,
                "ConnMgr": {},
                "DisableBandwidthMetrics": false,
                "DisableNatPortMap": false,
                "RelayClient": {},
                "RelayService": {},
                "ResourceMgr": {},
                "Transports": {
                        "Multiplexers": {},
                        "Network": {},
                        "Security": {}
                }
        }
}
```

9.2 Poster Presentation

Figure 9.1: **Poster**

References

[1] Al-Malaise, Rana Abbas and Y. Al-Jaber, "A survey: medical health record data security based on interplanetary file system and blockchain technologies, " Indonesian Journal of Electrical Engineering and Computer Science, vol. 10, no. 1, pp. 931-941, 2023.

[2] AS. A. Hasan, A. S. A. Ahmed, and S. S. Ahmed, "Distributed Off-chain Storage of Patient Diagnostic Reports in Healthcare System using IPFS and Blockchain," Journal of Communication Systems Networks, vol. 8, no. 2, pp. 44-49, 2023.

[3] B. Mukherjee, Jake Roy, and S. K. Saha, "Implementation of Ethereum Blockchain in Healthcare Using IPFS," International Journal of Intelligent Communication, Computing and networks, vol. 115, no. 1, pp. 199-212, 2022.

[4] Jin Soni, Rajput Singh, Shangping Wang , and Ying Wu, "Decentralized secure storage of medical records using Blockchain and IPFS: A comparative analysis with future directions," International Journal of Consortium Networks, vol. 3, no. 3, pp. 11-24, 2023.

[5] K. Roy and M. R. Islam, "BlockMedCare: A healthcare system based on IoT, Blockchain and IPFS for data management security," Journal of Egyptian Informatics council, vol. 4, no. 2, pp. 275-281, 2022.

[6] M. Singh, R. Kumar, and S. K. Pandey, "Hyperledger Healthchain: Patient-Centric IPFS-Based Storage of Health Records," Journal of Ambient Intelligence and Humanized Computing, vol. 12, no. 7, pp. 7037-7049, 2022.

[7] N. B. Tahmina, M. H. Kabir, and M. A. Hoque, "Blockchain-Based Secure Storage and Access Scheme For Electronic Medical Records in IPFS," Journal of Information Science and Engineering, vol. 36, no. 2, pp. 403-414, 2023.

[8] N. I. Khan, H. Y. Kim, and H. J. Kim, "Blockchain for Secure EHRs Sharing of Mobile Cloud Based E-Health Systems," Journal of CSIRO Data61, Australia., vol. 11, no. 11, pp. 4655-4666, 2023.

[9] Pubudu N. Pathiranai, Ming Ding, and Aruna Seneveratne, “Addressing the Challenges of Electronic Health Records Using Blockchain and IPFS ,” Journal of Ambient Intelligence and Humanized Computing, vol. 6, no. 4, pp. 3559-3571, 2022.

[10] R.Shivansh Kumar, Aman Kumar Bharti and Ruhul Amin, “Security and Privacy of Patient Information in Medical Systems based on Blockchain Technology,” Journal of Computer Sciences, vol.12, no.8, pp.8465-8477, 2021.