

Blockchain technique into Cloud System for Secure Distributed storage management.

By

Sharmin Khatun

ID: 213-25-984

Department of Computer Science and Engineering (CSE)
Daffodil International University, Dhaka.

This Report Presented in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science and Engineering.

Supervised By

Professor Dr. Sheak Rashed Haider Noori

Professor & Associate Head

Department of CSE

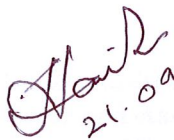
Faculty of Science and Information Technology
Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY
DHAKA, BANGLADESH
SEPTEMBER 2022

DECLARATION

I hereby declare that this research has been done by me under the supervision of **Professor Dr. Sheak Rashed Haider Noori, Associate Professor & Associate Head, Department of CSE, Daffodil International University**. I also declare that neither this research nor any part of this research has been submitted elsewhere for the award of any degree or diploma.


21.09.2022

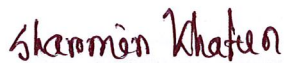
Supervised by:

Professor Dr. Sheak Rashed Haider Noori

Associate Professor & Associate Head

Department of Computer Science and Engineering

Daffodil International University



Submitted by:

Sharmin Khatun

ID: 213-25-984

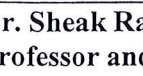
Department of Computer Science and Engineering

Daffodil International University

APPROVAL

This Project/internship titled “**Blockchain technique into Cloud System for Secure Distributed storage management.**” submitted by **Sharmin Khatun, ID No: 213-25-984** to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of M.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on September 21, 2022.

BOARD OF EXAMINERS



Dr. Sheak Rashed Haider Noori, PhD

Professor and Associate Head

Department of Computer Science and Engineering

Faculty of Science & Information Technology

Daffodil International University



Dr. Moushumi Zaman Bonny

Assistant Professor

Department of Computer Science and Engineering

Faculty of Science & Information Technology

Daffodil International University


Md. Sazzadur Ahamed

Assistant Professor

Department of Computer Science and Engineering

Faculty of Science & Information Technology

Daffodil International University


Md. Safaet Hossain

Associate Professor & Head

Department of Computer Science and Engineering

City University

ACKNOWLEDGEMENT

First I express my heartiest thanks and gratefulness to Almighty God for His divine blessing makes it possible to complete the final year thesis successfully.

Thankful and wish my significant obligation to **Professor Dr. Sheak Rashed Haider Noori**, Department of CSE, Daffodil International University, Dhaka. Profound information and unmistakable fascination of my administrator in the field of " **Blockchain technique into Cloud System for Secure Distributed storage management**" causes me to complete this proposal paper. His unending tolerance, academic direction, ceaseless consolation, consistent and lively supervision, useful analysis important exhortation, perusing numerous mediocre draft and redressing them at all stage have mode it conceivable to finish this proposal.

I would like to express my heartiest gratitude to honorable Professor and Head, Department of CSE, **Professor Dr. Touhid Bhuiyan** for his kind help, to finish my research and also to other faculty members and the staff of the CSE department of Daffodil International University.

I would like to thank our entire coursemate in Daffodil International University, who took part in this discussion while completing the course work.

Finally, I must acknowledge with due respect, the constant support and patience of my parents.

Abstract

A blockchain is an organized collection of data-storing nodes and linkages. This technology allows a public record of transactions, popularized by cryptocurrency. Blockchain relies on cryptographic solutions, as no practical cryptosystem has ever been able to be rigorously shown to have perfect or unconditional security. A decentralized, peer-to-peer network is used for the operation of algorithm-based security in blockchains. Within this network, exact protocols are used to build bonding across all entities, which prevents any one entity from dominating the underlying infrastructure. Blockchain technology improves cloud data storage and retrieval. Innovative and unlike databases, blockchain stores transaction records. Blockchain technology is a practical way to get around risk and computerized voting today. No one is aware of how to validate that document, therefore anyone who wants to update or change it may do so fast. Thus, this system proposes a blockchain-based online voting system for secure distributed storage management. From creating a genesis block, to utilize data sets and adding a node with hashing operation creates a unbreakable chain for voting system where any third party is unable to alter the placed votes or verifying the unauthorized voter from an unregistered source. Each chain in the Blockchain system in our proposed system connected to the system successfully, and each chain in the system was able to receive and transmit node information to other chains in the Blockchain system. In this system, the constructed chain was successful in obtaining all of the strings from the network. By obtaining the address of a single chain in the Blockchain network, each chain may thereafter gather data about the whole network and only accept votes from authorized users.

Table of Contents

Contents.....	Page
Board of examiners.....	I
Declaration.....	II
Acknowledgements.....	III
Abstract.....	IV
Chapter.....	Page
Chapter 1: Introduction.....	1-4
1.1 Introduction.....	1
1.2 Motivation.....	2
1.3 Rationale of the study.....	2
1.4 Objectives of this Research.....	3
1.5 Expected Output.....	3
1.6 Report Layout.....	3
Chapter 2: Literature Review.....	5-8
2.1 Chapter Introduction.....	5
2.2 Literature Review.....	5
2.3 Blockchain Technology.....	5
2.4 Blockchain-Based Cloud Computing.....	6
2.5 Challenges related to Blockchain.....	7
2.6 Hashing implementation in Blockchain.....	7
Chapter 3:Methodology.....	9-14
3.1 Blocks.....	9
3.2 The Genesis Block.....	9

3.3 Data Blocks.....	10
3.3.1 Previous Hash.....	11
3.3.2 Current Hash.....	11
3.3.3 Timestamp.....	12
3.3.4 ID Flag and Account Flag.....	12
3.3.5 Device ID and Account Number.....	12
3.3.6 Nonce.....	13
3.3.7 Message.....	13
3.4 The Consensus.....	14
Chapter 4: System Implementation.....	15-28
4.1 Introduction.....	15
4.2 Creating a Genesis Block.....	15
4.3 Express Js Framework.....	16
4.4 Blockchain functions.....	18
4.4.1 Block Creation.....	18
4.4.2 Returning Previous block.....	19
4.4.3 Proof of Work.....	20
4.4.4 Function for hashing.....	21
4.4.5 Placing Vote.....	21
4.4.6 Getting Node.....	22
4.4.7 Adding Node	22
4.4.8 Replacing longest chain.....	22
4.5 API Interaction.....	24
4.6 Workflow of Blockchain.....	25

4.6.1 Designing Blockchain.....	25
4.6.2 Initialization of chain.....	26
4.6.3 Connecting node.....	26
4.6.4 Placing vote.....	26
4.6.5 Mining block.....	26
4.6.6 Checking validity.....	26
4.6.7 Checking longest chain.....	27
4.6.8 Clearing data.....	27
4.6.9 Data formatting.....	27
4.6.10 Registration for Users.....	27
4.6.11 Data of Votes.....	28
Chapter 5: Result and Discussion.....	29-37
5. 1 Data-set.....	28
5.2 Performance.....	32
5.3 Distributed System test.....	32
5.4 Block mining.....	33
5.5 Longest chain replication.....	37
Chapter 6: Conclusion & Discussion.....	38-40
6. 1 Impact on Society.....	37
6.2 Ethical Aspects.....	37
6.3 Summary of the Study.....	37
6.4 Conclusion.....	38
6.5 Future Work.....	40

List of Figures

Figure 1: Architecture of Genesis Block.....	9
Figure 2: Architecture of all data blocks.....	10
Figure 3: A part of the connection of blocks through a chain.....	11
Figure 4:Chain of Blocks.....	13
Figure 5:Nodejs Implementation.....	17
Figure 6: Workflow of Blockchain.....	25
Figure 7: Network address given to chain.....	33
Figure 8: Parsed network addresses from other chains.....	33
Figure 9: New Block generation.....	34
Figure 10: All Other Data Blocks.....	35
Figure 11: Chain information in Node 127.0.0.1:5001.....	36

List of Tables

Table 1: API List.....	18
Table 2: Blockchain network chain address table.....	28
Table 3: Votes in a single chain of Blockchain network.....	30
Table 4: Proof of work complexity based on leading zeros of Target Hash.....	31
Table 5: Proof of work complexity based on Order of equation.....	32
Table 6: State change table.....	32

Chapter 1

Introduction

1.1 Introduction

A blockchain is an organized collection of data-storing nodes and linkages. This technology allows a public record of transactions, popularized by cryptocurrency. Many applications have developed since then. Blockchain is a distributed database of ordered entries connected by chains [1]. Only authorized individuals can access the information in these blocks. Blockchain is a distributed append-only public ledger system that was at first designed specifically for use with cryptocurrencies such as Bitcoin. Satoshi Nakamoto first presented the idea of a blockchain in 2008; since then, it has garnered great interest as an emergent peer-to-peer (P2P) technology for distributed computing and decentralized data exchange [1]. Blockchain technology is becoming more popular across various industries, even though it was first created to support cryptocurrencies like Bitcoin (such as secure contracts, financial transactions, health information sharing, sharing health information, and so on). Despite the fact that supporting cryptocurrency was its main objective, this has happened. Blockchain is well recognized as a computer algorithm created to provide decentralized communication in a subscriber network that is peer-to-peer. This type of network ensures that all transactions involving its participants are open and accessible to all parties. It is important to note that Wall Street and the financial market perceive bitcoin like a stock, despite the cryptocurrency's true purpose and value have not yet been established, and its future is uncertain [2]. As "data sharing" is something that many businesses, as well as individual users, will require in the near future, contemporary research on the topic has become essential in the context of the current circumstances. Blockchains are often run on a peer-to-peer network that is decentralized and in which all entities conform to the same protocols [3]. This prevents any single party from dominating the underlying architecture of the blockchain. In an ideal world, this open, decentralized, permissioned architecture stops anybody from imposing regulatory pressures on the blockchain or interfering with its operations in any other way. Distributed ledger, cryptography, consensus mechanism, and smart contracts makeup blockchain architecture [10]. BlockBench, a standard blockchain for comparing private blockchains. There are multiple frameworks of blockchain, each with its own benefits and drawbacks [10]. The most popular framework is Bitcoin, which was created by Satoshi Nakamoto in 2009. Bitcoin is based on a distributed database that is open to all nodes. This makes it difficult to hack, and it has the advantage of being anonymous.

Another popular framework is Ethereum, which was created by Vitalik Buterin. Ethereum uses a different approach to blockchain technology [9]. It allows for more than one blockchain to be connected, which makes it easier to create smart contracts and applications on the network. Other popular frameworks include Hyperledger Fabric and IBM Blockchain. It's important to choose the right framework for your needs because not all frameworks are suitable for every application. Blockchain relies on cryptographic solutions, as no practical cryptosystem has ever been able to be rigorously shown to have perfect or unconditional security. In general, no cryptosystem has ever been capable of being shown to have perfect or unconditional security [6]. "Computational security can only exist under certain assumptions," as stated by Stefan Wolf. These assumptions include the constraints of computer power and the difficulty of the underlying cryptographic problem to be addressed [6].

1.2 Motivation

Everyone wants to see fairness in the voting process. Voters frequently have questions or concerns about the voting methods, counting processes, and the declaration of the results after an election has taken place. The requirements and advances of the historical period in which the election systems were formed have led to their gradual evolution over the course of history. The development of new technologies presents opportunities for innovation in every industry similarly, the addition of digitalization techniques to voting systems is expected to reduce the likelihood of mistakes made by humans. On the other hand, in contrast to voting methods based on paper, electronic voting systems are susceptible to issues such as system failure, compromised network security, and compromised information security. People who are permitted to access the voting system from the inside or the outside might create security flaws, which is one of the most critical concerns with electronic voting systems.

1.3 Rationale of the study

In our daily life, so many fields blockchain can integrate and make our life easier, such as it can be used for education, fitness, sports, media, video, shopping, finance etc. Some of the basic businesses and organizations are not a good fit for centralized information management [2]. When there is an increase in the volume of traffic in centralized systems, you run the risk of encountering bureaucratic leadership, delays in work, remote control, the management of large-scale systems, and bottlenecks. Additionally, centralized systems may have a high dependence on network connectivity, fewer possibilities for data backup, and complex server

maintenance, which are all potential drawbacks. It is possible that this leakage will persuade user-based systems to switch from centralized systems to decentralized systems [1]. Nevertheless, maintaining the system's integrity while accommodating a wide variety of users is the most important challenge. A system is said to be distributed when its component parts are dispersed over multiple locations yet nonetheless work together toward a unified purpose. It shouldn't rely on centralized servers and shouldn't have a single point of failure either [5]. A well-known example of a distributed system is Facebook, as is the early form of email. They send and receive messages directly to other parties in order to connect with one another and coordinate their activities. A distributed system has the ability to manage a large number of IoT units, as well as devices, applications, and users. The primary benefits of utilizing this design include concurrent access to resources, scalability, and transparency [7]. One of the limitations of the decentralized system would be that it would not discourage businesses from taking an interest in it. And moreover, if we have a massive data exchange method, then it will be difficult work to do [4].

1.4 Objectives of this Research

The main objective of this research is to:

- Illustrate cloud computing with blockchain technology.
- Implement SHA256 hashing algorithm in secure distributed online voting system storage management.
- Illustrate and compare the existing online voting system with a blockchain technique-based voting system for secure distributed storage management.

1.5 Expected Output

Modern culture is increasingly embracing the practice of online voting. It has a significant chance of lowering administrative expenses and raising participation rates. Voters can cast their ballots from any location with an Internet connection, eliminating the need to print ballots or set up polling places. The article that comes next provides an overview of blockchain-based electronic voting systems. The primary objective of this analysis was to assess the state of blockchain-based voting research at the moment and illustrate online voting platforms and any associated challenges to forecast future advancements.

1.6 Report Layout

The layout of this report is described below: -

In chapter 1 I have covered the introduction to my thesis, rationale of the study for building this kind of system, the objectives, and goals of the Blockchain technique into cloud system for secure distributed storage management of an online voting system, what I have planned or the expected outcome of the application and the ultimate layout of this report.

In chapter 2 I have added some related works and some studies that helped me a lot in this application.

In chapter 3 I have talked about AES, Hash algorithm, Cloud section of the system, Cloud-Computing-Technology, Cloud-Computing-Technology Managed, and Cloud Storage Managed & Cloud Service Providers Store.

In chapter 4 I have specified the whole process of this system using some Proposed Schema, concepts, Secret Image Encryption, DWT-SVD-based Image Steganography, and Integrity Check Using the SHA-512 Hash Function.

In chapter 5 I included the specification that I have described in Introduction of Experiment, Experimental Results, Results of the Encryption-based AES Algorithm, and Test of the Proposed Method.

In chapter 6 I have added the conclusion and challenges details and analysis Scope of the Problem.

Chapter 2

Literature Review

2.1 Chapter Introduction

In this chapter, I will dialogue about the associated works, case studies, scope of the problem, and challenges. After solving the plan, I have commenced analyzing on some different related packages and case studies. Summarize of those are delivered on this bankruptcy.

2.2 Literature Review

A decentralized, peer-to-peer network is used for the operation of algorithm-based security in blockchains. Within this network, exact protocols are used to build bonding across all entities, which prevents any one entity from dominating the underlying infrastructure. Blockchain technology improves cloud data storage and retrieval. Innovative and unlike databases, blockchain stores transaction records. Satoshi Nakamoto introduced Blockchain in 2008 and implemented it into Bitcoin in 2009. The blockchain records transactions chronologically. A blockchain network's ledger is administered by all participating nodes, unlike a centralized directory where each node updates the ledger on a server. A decentralized database is one way to think of blockchain, which is a technology that is not that old but is still relatively new. As a result of the heavy reliance that blockchain systems place on cryptographic hash functions to preserve their data, it is exceedingly difficult to manipulate any data that is stored within the system [8].

2.3 Blockchain Technology

Blockchain technology has been known as a cryptocurrency platform since the advent of Bitcoin, the first and largest application. Blockchain contributes to the process of transforming a centralized, unreliable ledger maintained by a single third party into a decentralized, reliable form maintained by various validation nodes. Due to its decentralized nature and robustness of security, blockchain has great potential to be applied in various fields other than cryptocurrencies. A blockchain is an increasing list of records, called blocks, that are linked using encryption. Virtualization technology covers the IT architecture in the cloud computing system, making all of the system virtualized, including servers, storages, networks, applications, and so on. It achieved unified management, as well as the monitoring and control of all resources, allowing the system to be more flexible while also increasing its efficiency

[9]. Once the practical memory is brought under unified management, the file system, which is located at the upper data management layer, will be able to refer to the storage areas without making a distinction. This will make accessing data and managing files easier [9]. The methods through which the objective of data storage and administration can be achieved utilizing a type of data storage known as cloud storage are server cluster, grid computing, and distributed storage technologies [9]. Blockchain technology has fixed the weaknesses in the current voting process, improved accessibility and transparency, stopped fraudulent voting, reinforced data privacy, and verified the outcome [10]. The adoption of the electronic voting process in the blockchain is a major development. Electronic voting does, however, come with some serious hazards. For instance, if the system is compromised, it's likely that all votes will be manipulated and utilized improperly.

2.4 Blockchain-Based Cloud Computing

The International Data Corporation (IDC) defines cloud computing as an evolving structure and model of ICT (Information and Communication Technology) that can be used to construct or supply applications, platforms, infrastructure, and other public services [9]. Distributed computing, parallel computing, grid computing, virtualization, loading balance, and so on are all examples of the classic computer and network technologies that are further integrated and developed through cloud computing [9]. Each of these smaller programs is then sent into the cloud computing system, which is made up of numerous servers, before the results are finally sent back to the clients. The internet is used to complete this process [9]. Blockchain is extensively used important in sectors including data sharing, the Internet of Things (IoT), encrypted currencies, and supply chain financing. However, there are security concerns with various blockchain levels [11]. Hash functions are a crucial component of the blockchain, although they are not very effective. Consequently, this research suggests a new method based on PRCA for optimizing blockchain hash algorithms (Proactive Reconfigurable Computing Architecture). In order to improve the computational performance of hash functions, this paper implements a pipeline hash algorithm and optimizes the efficiency of communication equipment and network data transmission by combining the blockchain with a mimic computer [8].

2.5 Challenges related to Blockchain

In spite of the many benefits offered by blockchain technology, there are two problems that frequently arise with blockchain-enabled cloud solutions at the present time. The use of blockchain technology in cloud applications is typically met with a number of specialized technical obstacles. This presents the first category of challenges. The majority of problems may be traced back to the technical properties of blockchain, some of which are actually regarded as benefits. For instance, a setting with complete decentralization (such as a public blockchain) provides a powerful autonomous working mode; yet, the absence of control in this mode is, in many real-world contexts, regarded as a significant limitation. There are many factors, including legal considerations and governmental responsibilities, that make it impossible to completely abandon centralized forms of governance. Confidentiality is one of the attributes of cryptographic systems that is regarded as a "must have" characteristic of these kinds of systems. The objective is to transmit a message that can be understood by no one other than the person to whom it is addressed because no one else can decipher it. In light of this strategy, we need procedures that alter the message in such a way as to render it unintelligible [12]. One vulnerability of blockchain transactions is a replay attack. This occurs when someone sends you an incorrect bitcoin transaction that you then use to spend your own bitcoins. Because the data in the blockchain is tamper-proof, you would never be able to spend the bitcoins again without being detected. An impersonation attack occurs when someone tries to use your account without your permission. This might be done by stealing your password or by tricking you into entering it on a fake website [11].

2.6 Hashing implementation in Blockchain

A hashing function is applied to the data in order to establish a unique digital fingerprint that is almost identical to the fingerprint of any other data file. This fingerprint is practically impossible to differentiate from any other data file. The objective of hashing is not to hide information; rather, it is to offer verification that the material in question has not been altered in any manner. This is not to say that hashing cannot be used for this purpose. It is not possible to recover the original file back from the hash by reversing the hashing process and using the hash [3]. These functions are necessary for cryptography since one of their primary properties is that it is difficult to compute the inverse of the function. Because these concerns involve mathematical problems that are not completely solved or known, we have to proceed with extreme caution [12]. The authenticity of a message or file can be checked using a technique

known as hashing. The hash function by itself is open source and does not require any kind of key in order to function properly. When a message or file is processed by a hash function, the function generates an output string of a predetermined length. Among the many methods, the Secure Hash Algorithm 256 (SHA-256) is one of the most common ones. It possesses qualities such as When SHA-256 is applied to any file, regardless of how long it is, the result is a binary string that has 256 bits. Hashes for files that are really identical can be completely different in unanticipated ways. In point of fact, SHA-256 was developed so that the hashes of various files are effectively random and uniformly dispersed across the entire set of all possible 256 binary strings, of which there are 2^{256} , or around 10^{77} [3]. In the year 2000, a more secure cryptographic hash function known as SHA-256 was developed. In the year 2002, it became a FIPS standard. Any string or input data can be hashed using SHA256 using tools that generate hashes. There is a 264-1bit cap on the maximum size of a message, and 256 hash values are generated. Any file can provide a hash that is completely unique. The same amount of input will always result in the same amount of output. Because of this, the hash is also referred to as the "fingerprint" of the file. Even though a file's hash always returns the same result, it's possible for two files to share the same hash. This is what's known as a "collision." In actuality, though, there is a very slim chance that this will take place. Hash functions cannot be reversed once they have been used. It is not possible to restore a file using the hash of the file. The immutability of the blockchain is ensured by the hashing process, which uses a Merkle tree. It is improbable that there will be a single point of failure in the blockchain system because of its distributed nature and the fact that transaction data is backed up on all of the network nodes. A blockchain enables transactions to be verified and completed quickly without the need for a third party. Every block in the order makes a reference to the one before it. This is essentially the parent block's preceding block's cryptographic hash. The Genesis block, which is the very first block on the blockchain, has no parent block. The block header and block body, which both include a list of transactions, make up the majority of the blockchain design structure [3]. Version number, preceding block hash, Merkle root, date, difficulty target, and nonce are just a few of the data included in the block header. Also uploaded is a single digest produced by the Merkle tree using a safe hashing technique, such as SHA-256. The integrity of the certificate of origin is covered by this digest.

Chapter 3

Methodology

3.1 Blocks

Blocks are data structures within a database, where transaction data or any other type of data are stored permanently. In a blockchain, the blocks are chained together as a permanent record. A block holds the transactions including those not yet validated by the network. After the validation of the data, the block needs to be closed. After that, a new block is generated for new data to be created, stored, and validated. So, a block is a permanent store of records or data that, once written, cannot be changed, altered, or removed.

3.2 The Genesis Block

The very first block that is generated in a blockchain is called the genesis block. It mainly represents the beginning of a blockchain. This block can be considered as a dummy block or dummy head of the blockchain where the first data which has been recorded in the block. The other blocks are the continuation of the genesis block. These other blocks are often considered as the children of the genesis block. In our model or design, it doesn't include any recorded data, The overall architecture of the Genesis Block may be broken down into the following components:

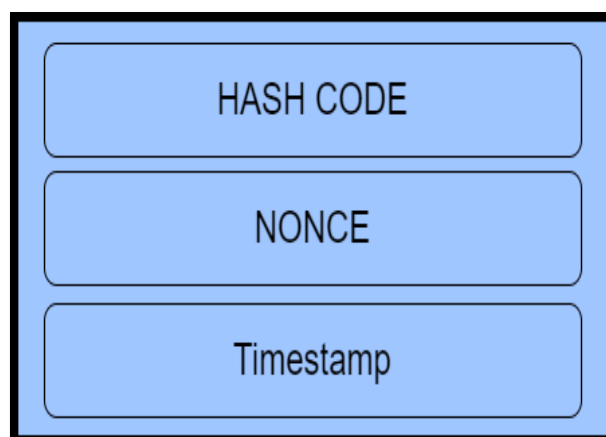


Figure 1: Genesis Block Workflow

3.3 Data Blocks

Our blockchain, with the exception of the first block, is composed entirely of blocks, which serve as the primary units for storing data. All of the transactions that take place are recorded in the corresponding data blocks and uploaded to the blockchain as they take place. Our blockchain is constructed using blocks, each of which performs the function of an information keeper. Blockchain is the term that was used to describe the subsequent linkage of blocks via chains. These blocks are the primary piece of virtual content that goes into the construction of the blockchain. The whole of the blocks that make up our blockchain may be seen as a connected sequence, with each block being linked to its predecessors by their respective hashes. Blocks, also known as nodes, are the core component of the blockchain that is responsible for the storage of information [55]. The most difficult part of our design is the construction of the blocks, which is necessary since various kinds of data is required. The following is how each of our blocks is laid out architecturally:

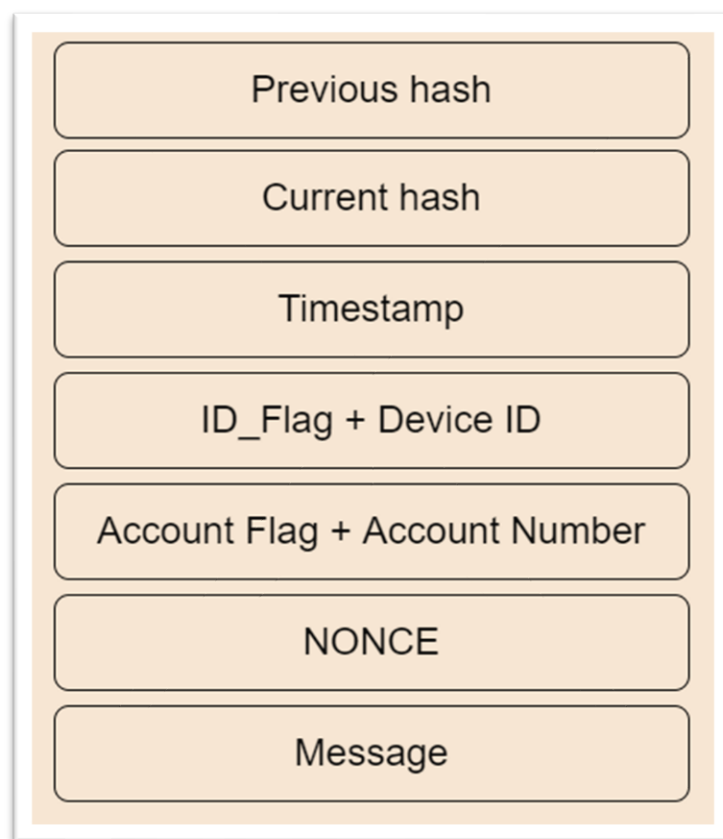


Figure 2: Architecture of all data blocks

3.3.1 Previous Hash Function

All of the blocks in our blockchain are related to one another by the use of the previous hash, which is SHA256 [58]. This ensures that the blockchain can only be retrieved in a single direction. This particular system made use of an immutable 64-bit SHA256 hash, which implies that a hash can never be altered by being rewritten [58]. In our system, each block has a reference to the one that came before it, and as a result, each block is connected to the genesis block. However, since each block in the chain is constantly produced and may change at any time, it is impossible to arrive at the genesis block by working one's way through the other blocks. Therefore, Blockchain is able to preserve its authenticity and is unchangeable.

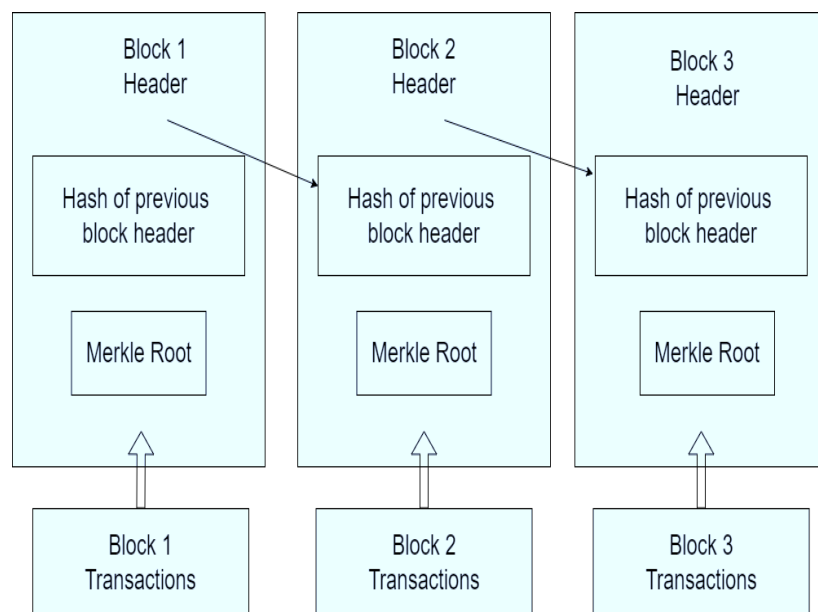


Figure 3: A part of the connection of blocks through a chain

3.3.2 Current Hash

Every single one of the blocks that make up our blockchain has a hash, which serves as the block's individual identifier [59]. These hashes cannot be changed in any way, and they serve as the fundamental building block of the blockchain since they not only link but also reflect the whole network. Each block makes a reference to the hash of the block that came before it. Now, the issue that has to be asked is how these hashes are created in a manner that ensures they are both non-deterministic and one-of-a-kind. These hashes are produced using the date, nonce, and flags as their inputs, respectively. Because each of these values is one of a kind, we only get one hash from the 264 numbers, and the odds of the two hashes being identical are one in $3.7 * 10^{11}$ [60].

3.3.3 Timestamp

When the block is mined, the timestamp is saved, and it displays the moment at which it was mined. As a result, each block in our system is one of a kind, it is not possible for it to undergo any further changes, and it is added to the blockchain along with the specified date. This timestamp is also used to generate a one-of-a-kind hash, which contributes to the increased transparency of the blockchain.

3.3.4 ID Flag and Account Flag

Because we are dealing with many different forms of information that we need to store and transact, each category has to be split in a way that is completely distinct so that we can exercise control over it inside the blockchain. In order to construct this model, we will mostly be relying on two different kinds of input, which are:

- a. Smart Devices
- b. Information Relating to Finances

To begin, the cause for selecting these two distinct kinds of data is due to the fact that they each represent an entirely separate field. In addition to that, the goal is to either generalize or simplify the model in the near future. In this particular instance, we will be using a one-directional array that is one in length and only stores the information listed above. In continuation with the previous point, these two different kinds of dossiers need to have a symbolic notation in order for us to be able to symbolize them and make them more comprehensible for the model's implementation and for any future analysis that may be performed. As a result, the ID Flag is such that each block refers to variable and the 16-bit integer variable have become more common.

3.3.5 Device ID and Account Number

The sequence the one preceding it. This is effectively the cryptographic hash of the block before the parent block. There is no parent block for the Genesis block, which is the very first block on the blockchain. The majority of the blockchain design structure is made up of the block header and block body, both of which include a list of transactions [3]. The block header contains information like as the version number, previous block hash, Merkle root, date, difficulty target, and nonce, to name a few. A single digest generated by the Merkle tree using

a secure hashing method, such as SHA-256, is also uploaded. This digest provides protection for the authenticity of the certificate of origin.. Therefore, it is essential that this information be preserved inside each block that is being hashed in order for us to be in a position to unambiguously identify each device as well as the kinds of devices that constitute each device.

3.3.6 Nonce

A nonce is a random number that is often used in the field of cryptography for the purposes of generating and controlling hashes. Nonce values in our system are most often random numbers, but they are also capable of taking on a pseudorandom value. In our system, not only can the timestamp but also other information be utilized to make encrypted communication more transparent, but also the notation. To a significant extent, we relied on the nonce for blockchain's proof-of-work in order to locate the needed hash in the mempool [14]. Therefore, in order to discover the appropriate hash up to a given limit, hashes more quickly, and this allows us to make our model more dynamic.

3.3.7 Message

In our system, the Message that is shown when each transaction is completed provides information about the current condition of the block, including whether or not the block was mined successfully and whether or not it had any problems. This not only makes the blockchain system more user-friendly but also assists the engineers in identifying problems and weaknesses in the system.

To summarize, the following is the final order for all of the blocks:

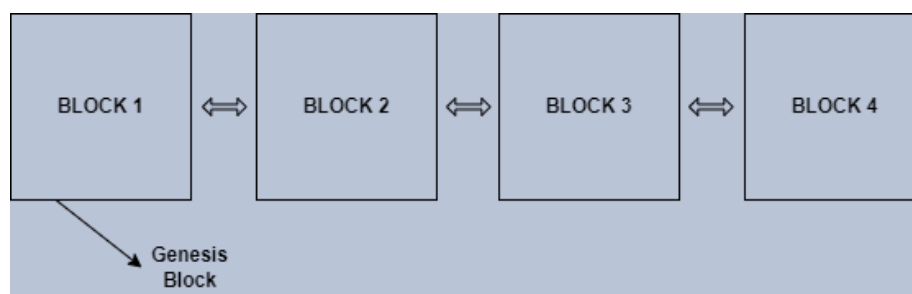


Figure 4: Chain of Blocks

3.4 The Consensus

The consensus is primarily an agreement that can be implemented as a method for a digitalized system that can tolerate faults [63]. One of the most significant issues with conventional transactions was the participation of a third party, which would take up both time and financial resources [64]. Blockchain was able to overcome this issue, but it still requires some type of algorithm that can monitor every activity or transaction that involves information. Because of this, the concept of consensus emerged. We designed a mechanism for reaching a consensus for our system and then put it into action so that we could have transparent and secure communication. The Proof-of-Work algorithm is the primary component of consensus in our system. This method allows a digitalized as well as a decentralized system to determine if the transaction is genuine or not.

Chapter 4

System Implementation

4.1 Introduction

The fundamental premise behind Blockchain is that it should be possible to add information to blocks and connect those blocks in such a manner that enables consumers of the data to monitor the data's continued authenticity. In reference to our system, the first block that is generated after the genesis block is produced is referred to as the genesis block. Depending on the settings that have been applied to it, the genesis block may either be an empty block or one that contains information.

4.2 Creating a Genesis Block

In our system, we have produced what is known as a "dummy" genesis block. This block does not have any information other than the hash, Nonce, and Timestamp associated with it. When a specific criterion has been met, information will begin to be added to the next block, and at that point, A trustworthy miner will mine the block and then add it to the Blockchain. The hash of the block that came before it will be included in the freshly mined block. The hash of the block that is being mined at the moment will be included in the subsequent block that is mined. In this manner, each block generates a chain that can be followed all the way back to the genesis block by including the hash of the block that came before it in the chain. After a predetermined amount of time, each link in our system compares itself to determine if it is the longest chain or not. This process continues as more information is gathered and more chains are generated, creating a network of chains. If so, it will be regarded as being the longest chain. If the chain is already the longest chain, nothing will happen; if not, the longest chain will take its place. In this way, each chain pits itself against the others to see which can grow the longest. This is done in order for the block to be mined. The successful miner will be given the option to mine the block and add it to the Blockchain, as well as the prize of winning the competition. There seems to be a correlation between the number of miners and the amount of time it takes to find a solution to the challenge required to mine a block. A validation check must be performed on each vote before it can be included in the block. If this is not addressed, the system will become worthless. As a result, a process to verify the results of the vote has been developed and put into place. In addition, when a shorter chain attempts to replicate the longest chain, it is also

required to check if the chain being copied is legitimate or has been tampered with. In such a case, the data on the shorter chains can start to get damaged and duplicated, rendering the system completely unusable. As a result, step 28 of the replication process in our system involves a process of validation, which is something that has already been completed in our model. For the purpose of recording a smart device or information in the Blockchain, the very first vote that is connected with the device or information will not have any sender identifier; instead, it will only contain an identification of the recipient. Therefore, information may be traced to its very first appearance on the Blockchain network, which can give crucial analytical information for a variety of applications, including study. In our Blockchain network, it is possible to transmit both physical and intangible elements, such as dollars, which are not tangible elopements smart devices, which are tangible. For example, a tangible element would be a smart gadget. Additionally, the system is able to produce a list of information that is connected to a certain identity. Because of the way that Blockchain technology works, it is possible to trace devices that are linked to a specific identity or id if the person has only one identity through which the entity conducts all of the votes. This is only the case if the entity uses Blockchain technology to keep track of votes. As a result, persons running into problems with their privacy might be doing themselves a disservice. As a result, the system will alter the user's identity whenever they cast a vote, making it impossible to attribute information to a specific person. The voting and verification processes are governed by a set of specific functions and restrictions that are included in the Blockchain concept. The JavaScript programming language was used to execute the specified Blockchain model after it was developed.

4.3 Express Js Framework

Express.js is a web application framework for Node.js that is both open-source and free to use. It makes it possible to develop and construct web apps in an efficient and straightforward manner. Node.js is an open-source, cross-platform JavaScript runtime environment that was developed specifically for back-end use. It runs on the V8 engine and allows JavaScript code to be executed outside of a web browser. Node.js was developed so that scalable network applications could be produced. Express.js is often used in the process of developing web servers for apps based on angular or react. Therefore, in order to host a blockchain on our local PC, we decided to utilize express js, and the initialization of it looks like this:

```

const express = require("express");
const mongoose = require('mongoose'); 1.1M (gzipped: 290k)
const path = require('path'); 211 (gzipped: 165)
const dotenv = require('dotenv').config() 1.4k (gzipped: 776)

const cors = require('cors'); 4.5k (gzipped: 1.9k)
const errorHandler = require('./middleware/error-handler');

const adminRoutes = require('./routes/admin');
const searchRoutes = require('./routes/search.js');

const app = express();
app.use(express.json());
app.use(express.urlencoded({ extended: true }));
app.use(cors());

app.use('/api/admin', adminRoutes);
app.use('/api/search', searchRoutes);

app.get('/', (req, res) => {
  res.send('<div style="width: 100%; height: 100vh; display: flex; flex-direction: column;
});

mongoose.connect(
  // `mongodb://${process.env.DB_USERNAME}:${process.env.DB_PASSWORD}@localhost:27017/${process.env.DB_NAME}`
  `mongodb://localhost:27017/${process.env.DB_NAME}`,
  {
    useNewUrlParser: true,
    useUnifiedTopology: true
  }
)
.then(() => {
  const port = process.env.PORT || 3000;
  app.listen(port, () => console.log(`Server is running at port:${port}`));
  console.log('Connected to mongoDB');
})
.catch(err => {
  console.error('Oops! Could not connect to mongoDB Cluster0', err);
})

```

Figure 5: Nodejs Implementation

Applications, by way of the creation of GET and POST APIs, In our system, the JavaScript code is used to immediately invoke Node.js, which then creates a server that may either host web pages or answer with a JSON response. Both of these options are available. For the sake of storing and updating the database, MongoDB is employed here. MongoDB is software that operates as a document-oriented database that is compatible with several platforms. MongoDB, which is a NoSQL database application, stores data in documents that are similar to JSON and may have optional schemas. The application programming interfaces (APIs) that were developed for use with the Blockchain are shown in the table that follows, along with a description of the functionality provided by each API.

API Name	Type	Description
Vote Block	GET	The online voting procedure of the Blockchain is started using the application programming interface. It gives back the information about the block that was voted on.
Get Chain	GET	The application programming interface gives back the blockchain's blocks.
Is valid	GET	Performs a check on the Blockchain and then delivers a response with information on the chain's legitimacy.
Place Vote	POST	Performs a check on the Blockchain and then delivers a response with information on the chain's legitimacy.
Connect node	POST	When a new chain is created, this message is broadcast to all participants in the Blockchain network.
Replace chain	GET	Check to see whether the current chain is already the longest chain, and if it isn't, replace it with the chain that is now the longest.

Table 1: API List

4.4 Blockchain functions

Multiple roles are essential to ensure that the Blockchain model can successfully carry out its intended purposes. Our model incorporates the features that are necessary for the Blockchain infrastructure to operate. The following are some of the functions that are included in the model.

4.4.1 Block Creation

After mining a block, devices must immediately begin to gather votes so that they may be included in the next block. It will be necessary to create a new block. In order to complete the transaction, the function checks that the proof of work and the preceding hash value are correct before adding the block to a chain that is part of the Blockchain network. Additionally, a timestamp is appended to the produced block

Algorithm:

1. Produce a Block dictionary complete with an index, timestamp, proof, and a prior hash, as well as a data key-value pair.
2. Empty data for storing next state data.
3. Add the Block element to the chain.
4. Send back the details for the Block.

Following the completion of the calculation for the proof, this function will construct a new block by using the proof, the hash of the prior block, and the date. Following is an example of the pseudo code for the new block.

```
FUNCTION create_block (proof, previous hash):  
    block = len(chain) + 1,  
    timestamp = string(current_datetime),  
    proof = proof,  
    previous_hash = previous hash,  
    votes = votes  
    votes = []  
    chain.append(block)  
    return block
```

4.4.2 Returning Previous block

In order to participate in the Blockchain system, one must get the data from the block that came before them. The previously mined block from the chain will be returned to you as a result of the functionality of this method.

Algorithm:

1. Retrieves the index of the currently active state.
2. Returns Block with the previous value of the index.

The value of the current state is retrieved by the system as the first step.

The current state stores information on the preceding block. The mechanism just returns the block address from the previous iteration. Below is some pseudo code for the function that may be used:

```
FUNCTION get_previous_block( ):  
    return (chain[-1])
```

4.4.3 Proof of Work

This is one of the most crucial uses for the Blockchain's underlying technology. This function will first compute the nonce value that will result in the block's hash being lower than the target hash, and then it will return that value. This will allow the block to be mined.

Algorithm:

1. Put a value of one in the new proof variable.
2. Make the validity of the evidence incorrect.
3. Determine whether or not the evidence is legitimate; if it is, return the previous value; otherwise, go to the next step.
4. Determine the hash value for the updated proof.
5. Compare the value of the hash to the value of the target hash.
6. If the current hash value is lower than the intended hash value, the proof validity should be set to the true state.
7. Add one to the newly calculated proof value.
8. Proceed to the next stage.

The value of the nonce is increased from 1 by the system such that the generated hash will have a value that is less than the value of the target hash. A system will create a hash for a certain nonce and a timestamp that is going to be lower than the hash that the destination system is expecting. The proof of work's pseudo code is presented in the following format:

```

FUNCTION proof_of_work(previous_proof):
    new_proof = 1
    check_proof = false
    while (check_proof is False):
        hash_operation = hashlib.sha256(str(new_proof**2 - previous
        proof**2).encode()).hexdigest()
        if (hash_operation[:4] = '0000'):
            check_proof = true
        else :
            new_proof += 1
    endif
endwhile
return new_proof

```

4.4.4 Function for hashing

A significant amount of hash values are used by the system. As a result, it requires a function that can return the hash value associated with a particular object. The following bogus code has been written in the following space:

```

FUNCTION hashing(block):
    encoded_block = json.dumps(block,sort keys=true).encode()
    return (hashlib.sha256(encoded_block).hexdigest())

```

4.4.5 Placing Vote

Checking the legitimacy of a vote and putting it to a serial, which will later transfer the data to a block so that it may be mined, are the two tasks that fall within the purview of this function. Following the completion of a vote and the verification that all other requirements have been met, this function will be executed in order to add a vote to the blockchain. After a vote has

occurred through passing the proof of work algorithm the sequence of miner and smart contract continues for mining. This continues until the next vote occurs. In the event that mining takes place, this function will be called to add a vote to a block, and as a result, the block will be added to the blockchain.

Algorithm:

1. Obtain information on voting.
2. Incorporate all of the material into a dictionary.
3. Include the information in the data Queue you have created.
4. The value of the preceding block's index should be increased by one.

When the voting information is received by the system, it is first converted into a JSON format, and then the information is placed to the votes queue.

```
FUNCTION add_vote(data, voter):  
    votes.append({  
        voter = voter,  
        data = data  
    })  
    previous_block = get_previous_block()  
    return (previous_block['index'] + 1)
```

4.4.6 Getting Node

The network's get Node function is an additional significant function that the network has. This node's role is to build a JSON list of all of the known chains that are part of the Blockchain network and then send that data to other chains so that those chains may discover the addresses of the chains that are part of the Blockchain network. Get Node and Add Node are required operations that must be carried out in order for the Blockchain system to operate in its entirety. If these features were missing, it would be necessary for each chain in the Blockchain to undergo manual configuration in order to link to each of the other chains. Even yet, it is still possible for new chains to be excluded from the Blockchain system.

4.4.7 Adding Node

This function's duty is to incorporate a newly discovered node into its existing list of known nodes. A new node indicates that the beginning of a new chain has been reached. This particular chain or node in the Blockchain network is unable to verify the blocks and chain since it is unaware of the existence of other nodes in the network. Additionally, it is unable to determine whether or not it is the chain with the longest length in the network.

4.4.8 Replacing longest chain

The replace chain operation is yet another essential component of the Blockchain algorithm. The goal of this function is to determine whether or not the currently active chain is the chain with the greatest length among the chains connected to the known nodes in the list.

Algorithm:

1. Set network value to nodes.
2. Make the value of the chain with the longest length null.
3. Make the chain length equal to the maximum length value.
4. Determine the length of the chain from the node.
5. If the length of the chain is longer than the longest chain, set the value of the longest chain to be equal to the length of the node's chain; otherwise, go to step 8.
6. Replicate the nodes chain, and then replace it with self-chain.
7. Back.
8. Proceed to next node.
9. Continue on to step 4

The procedure obtains a list of nodes from the queue of the target node, iterates over the nodes, and then compares the chain of the target node with the chain with the most nodes. It will replace the value with the chain that is currently the longest if it finds a longer chain. The pseudo-codes presented in the following format:


```

FUNCTION replace_chain(self ):
network = nodes
longest chain = None
max length = len( chain)
for (node in network):
response = requests.get(f 'http://node/get chain')
IF response.status code = 200:
length = response.json()['length']
chain = response.json()['chain']
If (length > max length AND is chain_is_valid(chain)):
max length = length
longest chain = chain
endif
endif
endfor
if longest_chain :
chain = longest_chain
return true
endif
return false

```

4.5 API Interaction

The Postman platform was used for the purpose of connecting with the APIs. Postman is a platform that may be used to improve the quality of API development while also making it simpler to do. Postman is capable of receiving data and displaying it in a formatted way that is tailored to the requirements of the situation. Additionally, API keys may be readily set in Postman so that they can be used whenever an API is called. When interacting with APIs, Postman is a very helpful tool. Postman was used throughout the testing process of the whole model to validate whether or not the system was functioning flawlessly. Continuous GET and POST commands were delivered in accordance with the specifications for the purpose of testing. For illustration purposes, the request for adding the vote would be considered a POST request while the vote is being submitted. Because if a user wants to cast a vote, he must first

satisfy the requirements for doing that action, which is why he must send the request to the server using the POST method and include all of the required information. POSTMAN formats all of the API calls in an easy-to-understand way and gives us the ability to make post requests of this sort. It performs these operations in a highly effective and efficient manner. If, on the other hand, we take into account how the blockchain operates, we will see that we are required to send various GET as well as POST requests. One example of this is that when the get chain function is called, it utilizes the GET method because its operation is such that the blockchain is requesting to get all of the blocks for the current chain, and the server responds to this request. As a result, POSTMAN engages in interaction with the blockchain. On a subsequent page, the technique for the whole system will be broken down in detail.

4.6 Workflow of Blockchain

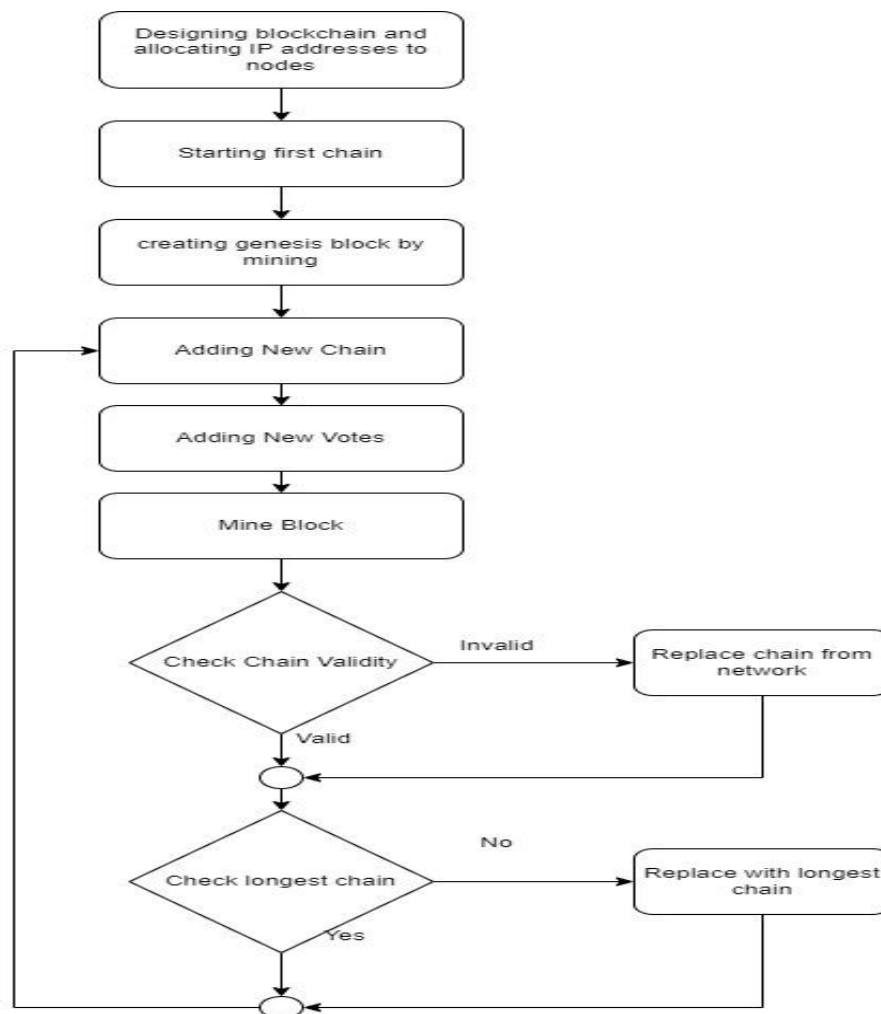


Figure 6: Workflow of Blockchain

4.6.1 Structuring Blockchain

During this stage, we will develop, analyze, and construct a Blockchain, as well as assign IP addresses to any nodes or chains that are created. The addresses 192.168.0.100:5001, 192.168.0.152:4000, and 192.168.0.112:3000 were assigned to the three devices that were used to construct the three chains.

4.6.2 Initialization of chain

We must use the Blockchain algorithm to mine the 1st block of the Blockchain. in order to start the Blockchain. This specific block is also known as the Genesis block. The validation process for all chains begins with this block. As the first chain in the address chain, the system was launched in the device with the address.

4.6.3 Connecting node

All of the nodes that had been patiently waiting in Blockchain's node queue up until this moment have now been added. In the process of putting up a Blockchain network, this particular step is really necessary. If this step is missed, the chain will not be able to find any other chains in the network, check itself, or decide whether or not it has the longest chain. This is true regardless of whether or not this phase is skipped.

4.6.4 Placing vote

A duplicate copy of each vote that has been validated for correctness may be found in the data queue. At this point in the procedure, the Blockchain will refer to the block that will be placed at a later time.

4.6.5 Mining block

To begin mining the block, each chain must first be solved. Only then will it be allowed to mine the block. After the proof of work has been successfully completed, the block's nonce value will be returned by the function if the block's hash value is less than the desired hash value. Only if the block's hash value is less than the intended hash value will this happen. During the block's mining process, the hash value will be utilized.

4.6.6 Checking validity

The chain has to do a validity check to ensure that the data is intact. In the event that the validity is found to be invalid, the chain is required to replace its value with the data that has been verified by the Blockchain network. In this scenario, there is a possibility that the network contains numerous invalid chains. It is necessary for the chain to replicate data from just the legitimate chains; in this example, the chain with the largest number of nodes that has the identical values is regarded to be the valid chain. The chain needs to determine which of those chains is the longest and then repeat the data from that chain.

4.6.7 Checking longest chain

The Blockchain needs to determine whether or not it is now the chain with the greatest length. If it is not the longest chain, it needs to find the longest chain and replace the current chain with the longest chain if it is not the longest chain. This ensures that the chain may continue to compete effectively with its peers inside the network.

4.6.8 Clearing data

During this stage of the process, the chain will purge the whole data queue in preparation for adding new data to the subsequent block. The information that was waiting in the data queue has already been included in the block that was mined before. Consequently, emptying the queue will not result in any data being lost in any way.

4.6.9 Data formatting

For its communication needs, the model employs an API-based method. JSON data packets are what are used to carry out the communication. Our approach is capable of storing a variety of data forms.

4.6.10 Registration for Users

The database maintained by the government stores the NID numbers of each user. The government is able to trace individuals' ownership of the NID using this method. In order to relate people to one another inside our system, we made use of their NID numbers. The JSON packet solely includes information on the recipient. This packet does not include any information about the sender.

4.6.11 Data of Votes

There are three fields included in the JSON packet that represents votes. To start, there is the voter. In order for the vote to be validated, the system must first determine whether or not the sender is the actual voter who is enrolled in the system. In addition to this, the system must determine whether or not the sender's NID is presently held by the individual. If all of these requirements are met, then and only then is the person who sent the vote entitled to cast it. Following that, the JSON packet will include the voter id inside it. This ID is one of a kind for each individual. Additionally, this information is capable of being categorized. The first thing that the system must do is check to see whether the individual has been entered into the system by making use of any registration package. The vote will not count if the individual is not registered in the system where it will be cast.

Chapter 5

Result and Discussion

The initial iteration of the Blockchain technology could support three separate chains operating independently. Every one of them has been started on a single device utilizing a variety of different socket addresses. For the purpose of picking the appropriate ports, three ports from the range of 1024 to 49151 that were registered were selected. 5001, 5002, and 5003 are the three ports that are available. The IP address 172.0.0.1 has been designated to be used only by the local server and is known as the dedicated IP address.

Chain Name	IP Address	Port Address
Chain 1	172.0.0.1	5001
Chain 2	172.0.0.1	5002
Chain 3	172.0.0.1	5003

Table 2: Blockchain network chain address table.

5.1 Dataset

For the sake of testing, the data for the votes have been produced in a random fashion. The voter address was produced at random to reflect the reality of the situation, which is that the username will be a 128-bit HASH address. The data consists of information about the device, the extent of which might vary in length based on the kind of data. It is possible to store different forms of value in the data information. During the course of our test, we made use of a setup that automated the addition of votes from various users to the Blockchain. A grand total of six thousand votes have been produced for the purpose of being deposited in the Blockchain. The total number of blocks in the first minute of play. Every user system contributed a vote at a random frequency ranging from three to seven seconds, depending on how long it had been since the last vote. After ten seconds have passed since the completion of the mining of the prior block and its addition to the blockchain, the mining of the next block will begin. . Table 2 displays a tabular depiction of the first 20 blocks produced in a chain of the Blockchain network. This representation comprises the block's index, the Nonce or proof's value, the block's generation timestamp, and the total number of votes that the block contains. In order to guarantee that the result could be duplicated in a setting that was conducive to doing so, the

votes were provided by automated users who added each vote within a predetermined time frame. After conducting tests to determine the ideal amount of votes to include in each block, the mining interval was selected. Figure 1 is where we may see a representation of the number of blocks that are produced in a specific time period. Within around twenty minutes, the approximately six thousand new votes that were generated were added. Because the amount of time needed to mine each block increases exponentially based on the number of leading zeros in target Hash, increasing the number of leading zeros in target Hash will result in a significant increase in the amount of time needed to deposit the same number of votes. This will result in a significant increase in the amount of time required. Our system's best settings were determined to be four leading zeros, and the power of the equation used to generate the hash was determined to be two; this resulted in the formation of a curve.

Index	Proof	Timestamp	Number of Votes
1	1	12:37:39.409978	0
2	533	12:37:56.748273	5
3	45293	12:38:13.207833	8
4	21391	12:38:29.520651	9
5	8018	12:38:45.662530	8
6	48191	12:39:02.195232	8
7	19865	12:39:18.505137	10
8	95063	12:39:35.292175	11
9	15457	12:39:51.544803	12
10	15479	12:40:07.758151	12
11	7889	12:40:23.936847	15
12	72474	12:40:40.578504	14

13	126616	12:40:57.611848	13
14	64161	12:41:14.394433	11
15	144125	12:41:31.556901	13
16	2492	12:41:47.694196	13
17	22592	12:42:03.987885	14
18	107780	12:42:20.845995	16
19	47346	12:42:37.354394	15
20	46891	12:42:53.895067	14

Table 3: Votes in a single chain of Blockchain network.

There are several possible states for the Blockchain system. It's possible for each state to have a unique set of attributes, as well as its own unique chain count. Table 3 provides a depiction of the various stages of the Blockchain system. The Blockchain system consists of three unique chains, and each chain was launched at a different time interval for the purpose of conducting a study of the Blockchain system's performance. Within the framework of the system, every chain has the ability to both recognize other active chains within the network and interact with other chains. As a result, was able to compile data on the operational chains that were already existing in the Blockchain and compare the data included inside the blocks in order to verify it.

Number of leading zeros	Time required (Seconds)
4	1.0849964618682861
5	5.25169825553894
6	20.6606292724609
7	1473.9516570568085

Table 4: Proof of work complexity based on leading zeros of Target Hash.

Another way to look at the amount of time needed for calculation is to consider the power of the numbers in the equation that must be solved in order to get a valid hash for which the current hash value is less than the goal hash value. In the course of our investigation, the Blockchain system prototype 45 utilized an equation that incorporated a power of two. Table 4 contains the equation that is used to generate the hash value.

New proof is equal to the hash value. Hash2 is equivalent to the prior proof. Hash2

Now, depending on the order of the equation, the amount of time needed for computation will change. Table 5 provides a depiction of the amount of time necessary for various power values. The amount of time necessary to compute the hash value varies very little depending on the sequence in which the problem is solved. As a result, one may get the conclusion that the order of the equation does not have a significant role in the amount of processing time necessary to obtain a hash value.

Power value	Time required (Seconds)
1	1.2212324142456055
2	1.145613670349121
3	1.0227384567260742
4	1.0944087505340576
5	1.2442378997802734
6	1.275045394897461
7	1.3591926097869873
8	1.3885324001312256
9	1.3314905166625977
10	1.3451454639434814

Table 5: Proof of work complexity based on Order of equation.

5.2 Performance

Our model's proof-of-work technique was based on establishing a target hash of a value range that had four leading zeros. This provided a vast value pool from which to build a hash that was acceptable. The amount of processing power and time needed to construct a target hash grows significantly with each new zero that is added. To have a clear vision of what is expected from each chain in the network and how the network should function, a model of the different states of the Blockchain system needs to have a better grasp of its various states. Using the information provided by the model, one may produce a visual depiction of an efficient procedure. The table 6 contains a reflection of the information of nodes that are in various stages of the system.

State	Time	Known chains in system
Initialization of system with chain 1	T0	Chain 1 (127.0.0.1:5001)
Initiation of chain 2	T1	Chain 1 (127.0.0.1:5001), Chain 2 (127.0.0.1:5002)
Initiation of chain 3	T2	Chain 1 (127.0.0.1:5001), Chain 2 (127.0.0.1:5002), Chain 3 (127.0.0.1:5003)

Table 6: State change table

5.3 Distributed System test

Every chain that was a part of the Blockchain system was able to successfully connect to the Blockchain system, as well as receive and transfer information about nodes to and from the other chains that were a part of the Blockchain system. During the course of our experiment, we established the connection 127.0.0.1:5001 and 127.0.0.1:5002. The addresses of chains 127.0.0.1:5001, 127.0.0.1:5002, and 127.0.0.1:5003 were contained in the chain 127.0.0.1:5002 directory. Following this, the address of 127.0.0.1:5002 was added to the chain that began with 127.0.0.1:501. After that, we were successful in retrieving all of the chains from the network when we received the list of known addresses from 127.0.0.1:5001. Because of this, as long as a chain in the Blockchain network has the address of another chain in the Blockchain network, it is able to gather information from the whole Blockchain network.

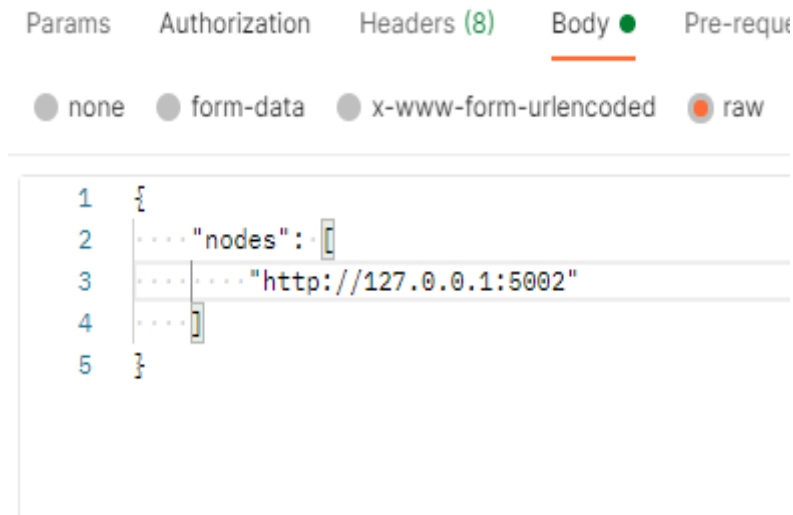


Figure 7: Network address given to chain

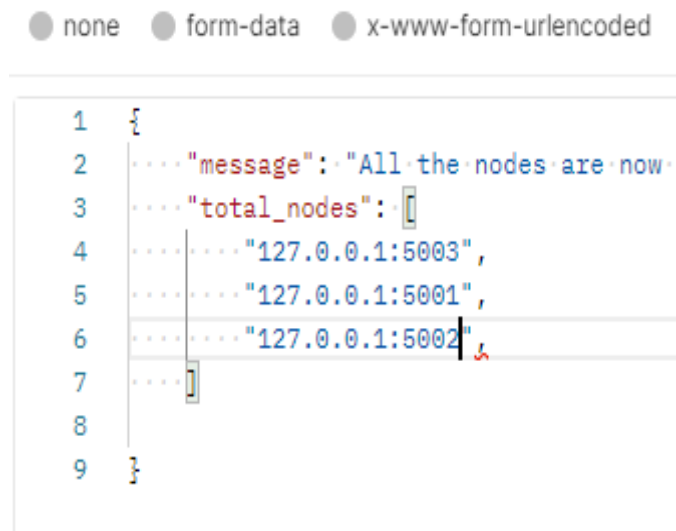
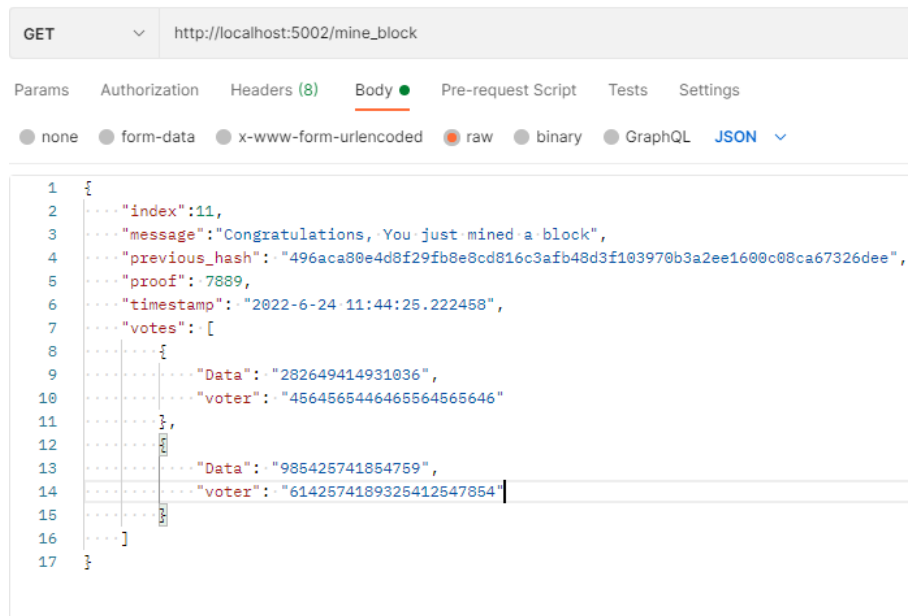


Figure 8: Parsed network addresses from other chains.

5.4 Particular Block mining

We provided a rather simple computation as the proof of work, where the target hash value is any object whose hash value has the same first four MSB bits as the number "0000." Our system was able to solve the proof of work in less than one second on each and every attempt. In contrast, the objective hash value in a real-world scenario will be incredibly low, necessitating a longer processing time on the part of the mining equipment to solve the issue. In addition, an equation with a complexity of n^3 was required to solve the proof of work. As a result,

the degree of difficulty might be deemed to be rather easy. The structure of the data is displayed down below:



```

GET http://localhost:5002/mine_block

Params Authorization Headers (8) Body ● Pre-request Script Tests Settings
● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL JSON ▼

1 {
2   ... "index": 11,
3   ... "message": "Congratulations, You just mined a block",
4   ... "previous_hash": "496aca80e4d8f29fb8e8cd816c3afb48d3f103970b3a2ee1600c08ca67326dee",
5   ... "proof": 7889,
6   ... "timestamp": "2022-6-24 11:44:25.222458",
7   ... "votes": [
8     {
9       ... "Data": "282649414931036",
10      ... "voter": "45645654464656564565646"
11    },
12    {
13      ... "Data": "985425741854759",
14      ... "voter": "6142574189325412547854"
15    }
16  ]
17 }
  
```

Figure 9: New Block generation

When it comes to making new blocks, there are two possible options. The first option is to produce a new block after a predetermined number of votes have been added to the waiting list. The second method requires the user to wait for a predetermined length of time before producing a new block. The second methodology was used in the development of our system. After a delay of ten seconds following the conclusion of the mining of the prior block, the mining of the next block can begin. All of the votes that were in the voting queue prior to the creation of block 49 being started were added to the block. Additionally, the voting queue has been purged to make room for fresh votes. The chains are updated to include the newly created blocks. Following the completion of a vote, the data blocks will be generated. The following is the appearance of each individual block:

```

1  GET http://localhost:5002/get_chain
2
3  Params Authorization Headers (8) Body ● Pre-request Script Tests Settings
4  none form-data x-www-form-urlencoded raw binary GraphQL JSON
5
6  [
7    {
8      "index": 6,
9      "message": "Congratulations, You just mined a block",
10     "previous_hash": "f6fc84c9f21c24907d6bee6eec38cabab5fa9a7be8c4a7827fe9e5",
11     "proof": 54784,
12     "timestamp": "2022-7-24 02:15:12.241578",
13     "votes": [
14       {
15         "Data": "992877612067658",
16         "voter": "3526943229607174433781"
17       },
18       {
19         "Data": "697174303462700",
20         "voter": "1250749323599237998844"
21       },
22       {
23         "Data": "299332570052936",
24         "voter": "6656496599546180472509"
25       },
26       {
27         "Data": "799695535619532",
28         "voter": "793721808781566421572"
29       },
30       {
31         "Data": "118972425699752",
32         "voter": "6656496599546180472509"
33       },
34       {
35         "Data": "415400667366522",
36         "voter": "6425879274245987484725"
37       }
38     ]
39   }
40 ]

```

Figure 10: Relevant Data Blocks

5.6 Longest chain replication

The first chain was the starting point for our system. Chain 2 then executed the largest chain replication method and copied the value from the longest chain, which in our network is chain 1. This occurred after chain 1 had mined a few blocks and after chain 2 had been started in the network. Following the addition of a small number of additional blocks to the chain, chain 3 was then initiated. Data from the chain with the longest history might also be copied to Chain 3. As a result of synchronization, each chain always included the same data at any given time period. Even though Node 2 was started after Node 1, we can see in figure 5.8 that the information included in Node 2 with the address 127.0.0.1:5002 is identical to the information contained in Node 1 with the address 127.0.0.1:5001 in the chain. The data from the chain with

the longest length was successfully replicated to Node 2. New devices can be registered with our system if necessary. The information that has been recently entered can be kept track of by our system. In addition to that, our system is capable of effectively verifying votes.

```

45  {
46    "Data": "992877612067658",
47    "voter": "3526943229607174433781"
48  },
49  {
50    "Data": "697174303462700",
51    "voter": "1250749323599237998844"
52  },
53  {
54    "Data": "299332570052936",
55    "voter": "6656496599546180472509"
56  },
57  {
58    "Data": "799695535619532",
59    "voter": "793721808781566421572"
60  },
61  {
62    "Data": "118972425699752",
63    "voter": "6656496599546180472509"
64  },
65  {
66    "Data": "415400667366522",
67    "voter": "6425879274245987484725"
68  },
69  ]
70  },
71  ],
72  "length": "18"
73  }

```

Figure 11: Node Chain Information 127.0.0.1:5001

Chapter 6

Conclusion & Discussion

6.1 Impact on Society

Elections are still mostly run offline, on paper, despite the digitalization of many significant parts of modern life. E-voting has been viewed as a hopeful and (ultimately) inevitable development that might hasten, simplify, and lower the cost of elections. It may also result in higher voter turnout and the growth of stronger democracies. However, a number of abnormalities and unethical attempts are constantly a big problem with electronic voting. By authorizing users and permitting the precise individual to participate in a vote, blockchain-based E-voting thus has the potential to revolutionize the entire view of the voting system. An isolated network that either uses blockchain technology to give citizens access to an open voting record or continues to rely on centralized institutions to oversee elections. E-voting would necessitate significant advancements in security mechanisms, according to many experts. Debatable issues include whether blockchain will be a revolutionary or merely incremental development and what its potential effects on democracy would be. As a result, this endeavor has a tremendous impact on society.

6.2 Ethical Aspects

Although electronic voting (or "e-voting") is widely used in place of traditional election methods, there is still a serious issue with the results' level of trust. E-voting systems are highly susceptible to manipulation problems, such as modifications to election results brought on by hacking or by the creator of the electoral system. Data sources from one party have the right to be stored and managed in a network because of centralized systems. By dispersing data across a system network, the trust problem brought on by a centralized data distribution system can be resolved. Therefore, if upholding an ethical code of conduct is a priority, this distributed electronic voting system may be the best way to prevent unethical behavior and voting fraud.

6.3 Summary of the Study

Our main goal was to develop a system that could save user data across many categories and link that data to national identity (NID). All blockchain models have already been developed and built for operation with regard to a single type of data or information. With

this framework, presenting a concept where the blockchain technology could be used for storing various types of information and effectively managing the votes while concurrently withstanding a common quality of processing capacity, effectiveness, and reliability of algorithm. Our system's implementation successfully showed that our concept is workable, entirely implementable, and performs better than existing Blockchain systems, which have low throughput, a lot of idle time, and other drawbacks. and weak protection in specific situations when computational capacity is constrained.

6.4 Conclusion

The suggested model is successful in challenging the prevalent perception of Blockchain technology, which holds that systems are computationally expensive and involve significant delays and overhead in data transfer. This type of blockchain technology has been shown to be unsuitable for everyday computing devices like smartphones, tablets, and notebook-style devices, which place a premium on performance metrics like battery efficiency and computability. In comparison to fundamental computational devices with frequent computational capacity, this system can operate with lower heightened computability time. The suggested approach can protect information in a setting where new attack mechanisms are constantly being developed against information security. The study and development of quantum computing has given rise to a new type of security breach, but the technology to protect blockchain systems from these kinds of problems is also constantly evolving. Our research on developing a system based on the proposed model allowed us to analyze the scalability of such a system in terms of real-world usage where the system needs to withstand attacks and provide standardized quality of service, in addition to giving us the chance to compare the system to currently used client-server-based models. In the present world, it is challenging to create a model that complies with standards and the notion of a system for providing information storage capabilities without downtime. When creating new full-scale systems that can operate in the actual world, the structure of the model and the analysis of the already-implemented system are taken into account.

6.5 Future Work

The model that has been built is a simplified version of Blockchain that consists of only the fundamental capabilities required for operation and analysis. In order to completely implement a system that is capable of managing the votes, there must to be a smart contract that takes into account every conceivable scenario. Additionally, our system is capable of managing numerous different types of information. The onboarding of new users and the setting of default values for attributes inside the system both require more investigation and study. In addition, during the course of some period of time, the system can get clogged with properties that have been deserted. It's possible that storing such information will prove to be problematic for the system. It's possible that a system that may delete information from the system will be required for either reasons related to security or criminal activity. As a result, the system ought to be altered in order to provide room for such functions. In the future, it is possible that the model will require revisions to the smart contract in order to accommodate newly enacted laws and guidelines. In addition, the system must to include a feature to update the smart contract so that it can manage the many types of data that will be presented in the foreseeable future. The system has to be modified so that it can do this function. In general, the functioning of the system was a smashing success. However, there is a need for further investigation and analysis to be carried out in order to enhance the system and locate any weaknesses it may have.

Reference:

- [1] F. Constantinos Patsakis, "A systematic literature review of blockchain-based applications: Current status, classification and open issues, *Telematics and Informatics*", *I. Fran Casino, Thomas K. Dasaklis, Constantinos Patsakis*, vol. 36, 2019, no. 0736-5853, pp. 55-81, 2022.
- [2] T. John, "Blockchain: A Misunderstood Digital Revolution. Things You Need to Know about Blockchain", *SSRN Electronic Journal*, no. 102139, pp. 1-25, 2019.
- [3] P. 3. Paul J. Taylor, Tooska Dargahi, Kim-Kwang Raymond Choo, "A systematic literature review of blockchain cyber security, *Digital Communications and Networks*", vol. 6, no. 2352-8648, pp. 147-156, 2022.
- [4] B. Nascimento, S., Kritikos, "4. European Parliament, Directorate-General for Parliamentary Research Services", *How blockchain technology could change our lives: in-depth analysis*, 2018.
- [5] W. N., Lawrence, *Cybersecurity, Data Privacy and Blockchain*, no. 42979-022-01020, pp. 3, 127, 2022.
- [6] "Unconditional Security of Cryptosystem", *A Review and Outlook. Trends in Applied Sciences Research*, pp. 554-562, 2011.
- [7] E. Rasslan, Mohamed, "Security Issues in Distributed Computing System Models. Security Solutions for Hyperconnectivity and the Internet of Things, *Advances in Information Security, Privacy, and Ethics*", vol. 36, no. 104018978-1-5225-0741-3009, pp. 211-259, 2016.
- [8] U. Javaid, Nadeem, "Enhanced Data Sharing Model", *Using Blockchain and Incentive Mechanism.*, 2019.
- [9] "Unconditional Security of Cryptosystem: A Review and Outlook. Trends in Applied Sciences Research", vol. 6, no. 1017311, pp. 554-562, 2011.
- [10] P. Feng Hao, " A Smart Contract for Boardroom Voting with Maximum Voter Privacy Available", pp. <https://eprint.iacr.org/2017/110.pdf>, 2017.
- [11] D. Kishor Datta, "A survey of blockchain from a security perspective", *Journal of Banking and Financial Technology*, pp. 3. 10.1007/s42786-018-00002-6, 2019.
- [12] V. Orel, "A handbook of germanic etymology", *BRILL LEIDEN*, 2003.

- [13] L. James & Adedokun, Emmanuel & Loveth, karngong, "Crypto Hash Algorithm-Based Blockchain Technology for Managing Decentralized Ledger Database in Oil and Gas Industry", 2019.
- [14] Z. Peng & Zhang, Wenzhen & Li, Huiyang, "Focus on Blockchain: A Comprehensive Survey on Academic and Application", vol. 8, no. 187182-187201, pp. 10.1109/ACCESS.2020.3030491, 2020.
- [15] W. Mandalenakis, M., Hein, "The impact of blockchain technology on business models – a taxonomy and archetypal patterns", vol. 30, no. 10100712525-019-00386-3, pp. 285–305, 2020.
- [16] "A Data Processing View of Blockchain Systems. IEEE votes on Knowledge and Data Engineering", *Untangling Blockchain*, pp. 10.1109/TKDE.2017.2781227., 2017.
- [17] "A Low Storage Room Requirement Framework for Distributed Ledger in Blockchain", pp. 1-1. 10.1109/ACCESS.2018.2814624., 2018.
- [18] "A survey of blockchain from a security perspective. Journal of Banking and Financial Technology", pp. 3. 10.1007/s42786-018-00002-6, 2019.
- [19] "Crypto Hash Algorithm-Based Blockchain Technology for Managing Decentralized Ledger Database in Oil and Gas Industry", 2019.
- [20] 2022. Available: <<http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf?>>.
- [21] *Enhanced Data Sharing Model By Using Blockchain and Incentive Mechanism.*, 2019.
- [22] "Blockchain Techniques for Secure Storage of Data in Cloud Environment. Turkish Journal of Computer and Mathematics Education", pp. 12. 1515-1522. 10.17762/turcomat.v12i11.6074., 2021.
- [23] "A Study on the Optimization of Blockchain Hashing Algorithm Based on PRCA", *Security and Communication Networks*, vol. 2020, no. 8876317, pp. 12, 2020. <https://doi.org/10.1155/2020/8876317>, 2020.
- [24] "A Systematic Literature Review of Blockchain Technology: Security Properties, Applications and Challenges. Journal of Internet Technology", pp. 22. 789-801. 10.53106/16079264202107220400797, 2021.

ORIGINALITY REPORT

13%	9%	2%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.bracu.ac.bd:8080 Internet Source	8%
2	Submitted to University of East Anglia Student Paper	1%
3	Cai, Ze Hua, Shun Yan Wang, Shang Yin Long, and Hai Tao Zhou. "A Data Storage and Management Scheme in Cloud Storage Model", Applied Mechanics and Materials, 2013. Publication	1%
4	Submitted to University of Essex Student Paper	<1%
5	Dipankar Dasgupta, John M. Shrein, Kishor Datta Gupta. "A survey of blockchain from security perspective", Journal of Banking and Financial Technology, 2019 Publication	<1%
6	Submitted to Turun yliopisto Student Paper	<1%