A

Major Project Report

on

Secure Online Voting System using Blockchain

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

By

Aamena Suzzanne

(20EG105625)

Syed Riyan

(20EG105636)

Sree Harshitha

(20EG105655)

Under the guidance of

Mr. K. Sadanandam

Assistant Professor

Department of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ANURAG UNIVERSITY
VENKATAPUR-500088
TELANGANA
Year 2023-2024

DECLARATION

We hereby declare that the Report entitled **Secure Online Voting System using Blockchain** submitted for the award of Bachelor of Technology Degree is our original work and the Report has not formed the basis for the award of any degree, diploma, associate ship or fellowship of similar other titles. It has not been submitted to any other University or Institution for the award of any degree or diploma.

Place: Anurag University, Hyderabad

20EG105625: Aamena Suzzanne
20EG105636: Syed Riyan
20EG105655: Sree Harshitha

Date:20-04-2024

ANURAG UNIVERSITY | SCHOOL OF ENGINEERING

CERTIFICATE

This is to certify that the Report entitled **Secure Online Voting System using Blockchain** that is being submitted by **Ms. Aamena Suzzanne** bearing Hall Ticket number **20EG105625, Mr. Syed Riyan** bearing Hall Ticket number **20EG105636** and **Ms. Sree Harshitha** bearing Hall Ticket number **20EG105655** in partial fulfillment for the award of **Bachelor of Technology** in **Computer Science and Engineering** to the **Anurag University** is a record of bonafide work carried out by them under my guidance and supervision.

The results presented in this project have been verified and found to be satisfactory. The results embodied in this Report have not been submitted to any other University or Institute for the award of any degree or diploma.

Signature of Supervisor
Mr. K. Sadanandam
Assistant Professor, CSE

Signature of Dean
Dr. G. Vishnu Murthy
Dean, CSE

External Examiner

ACKNOWLEDGEMENT

We would like to express our sincere thanks and deep sense of gratitude to the project supervisor **Mr. K. Sadanandam, Assistant Professor, Department of Computer Science and Engineering,** Anurag University for his constant encouragement and inspiring guidance without which this project could not have been completed. His critical reviews and constructive comments improved our grasp of the subject and steered us to the fruitful completion of the work. His patience, guidance, and encouragement made this project possible.

We would like to express our special thanks to **Dr. V. Vijaya Kumar**, Dean School of Engineering, Anurag University, for their encouragement and timely support of our B.Tech program.

We would like to acknowledge our sincere gratitude for the support extended by **Dr. G. Vishnu Murthy**, Dean, Dept. of CSE, Anurag University. We also express our deep sense of gratitude to **Dr. V.V.S.S.S Balaram**, Academic coordinator, **Dr. Pallam Ravi**, Project in-Charge, **Mrs. Jyothi**, Project Co-Ordinator and Project review committee members, whose research expertise and commitment to the highest standards continuously motivated us during the crucial stage our project work.

Aamena Suzzanne

(20EG105625)

Syed Riyan

(20EG105636)

Sree Harshitha

(20EG105655)

ABSTRACT

In any democratic country, Voting is a fundamental right of any citizen that enables them to choose the leaders of tomorrow. It gives individuals in a community the facility to voice their opinions. It helps them realize the importance of citizenship. Online voting systems are software platforms used to securely conduct votes and elections. As a digital platform, they eliminate the need to cast your votes on paper or in person. They also protect the integrity of your vote by preventing voters from being able to vote multiple times.

Electronic voting, or e-voting has fundamental benefits over paper-based systems, such as increased efficiency and reduced errors. The electronic voting system tends to maximize user participation, by allowing them to vote from anywhere and from any device that has an internet connection. The blockchain is an emerging, decentralized, and distributed technology with strong cryptographic foundations that promise to improve different aspects of many industries. Expanding e-voting into blockchain technology could be the solution to alleviate the present concerns about e-voting. Here, we propose a blockchain-based voting system that will limit voting fraud and make the voting process simple, secure, and efficient.

Keywords— E-Voting, Electronic Voting, Blockchain Technology, Voting Fraud Prevention, Security, Efficiency, Transparency, Decentralization, Distributed Ledger, Cryptography, Voter Privacy, Trustworthiness, Authentication, Verification, Ballot Integrity

LIST OF FIGURES

Figure. No.	Figure. Name	Page No.
Figure 1.1	Existing Systems	2
Figure 1.2	Problem Illustration	3
Figure 3	Existing System vs. Our Solution	7
Figure 4	System Working Process	11
Figure 5.1.2	SHA 256 Working	14
Figure 5.1.3	Merkel Hash Working	15
Figure 6.1	Experiment Screenshots	22-27
Figure 8.1	Experiment Findings 1	32
Figure 8.2	Experiment Findings 2	33
Figure 8.3	Experiment Findings 3	33
Figure 8.4	Performance Analysis	34

INDEX

S.No.	CONTENT	Page No.
1.	Introduction	1
	1.1 Existing System	2-3
	1.2 Disadvantages of Existing System	3-4
2.	Literature survey	5-6
3.	PROPOSED METHOD	
	3.1 Key Components of Proposed Method	7-8
	3.2Advantages of Proposed Method	8-9
4.	SYSTEM DESIGN	10
	4.1 System Requirements	10
	4.1.1 Hardware Requirements	
	4.1.2Software Requirements	
	4.1.3Specific Requirements	
	4.2 Project Modules	10
	4.3 System Architecture	
	4.3.1 Pre-voting phase	11-12
	4.3.2 Voting phase	
	4.3.3 Post-voting phase	

5.	IMPLEMENTATION	
	5.1 Algorithms	13-15
	5.1.1 SMTP	13-14
	5.1.2 SHA-256 Bit Encryption Algorithm	14-15
	5.1.3 Merkle Hash	15
	5.2 Functional Modules	16-21
	5.2.1 Input Voter ID	16-17
	5.2.2 OTP Validation	17-18
	5.2.3 Private key authentication	18-20
	5.2.4 Casting vote	21
6.	6.EXPERIMENT RESULTS	
	6.1 Experiment Screenshots	22-27
	6.2 Parameters	27
	6.3 Parameter Formulae	28
7.	7. TESTING	
	7.1 Methods of Testing	
	7.1.1 Unit Testing	29
	7.1.2 Integrating testing	29
	7.1.3 Validation Testing	30
	7.1.4 User Acceptance Testing	30
	7.2 Unit Testing Test Cases	31

8.	8. EXPERIMENTAL RESULTS AND JUSTIFICATION 8.1 Usability Testing 8.2 Security and Integrity Assessments 8.3 Scalability and Performance Assessment 8.4 Performance Analysis 8.5 Parameters Comparison	32 33 33-34 34 35
9.	9. Conclusion	36
10.	10. REFERENCES	37-38

1. INTRODUCTION

Voting stands as a cornerstone of democracy, embodying the collective will of the populace in selecting representatives and shaping governments. Over time, the concept of voting has evolved into a crucial tool for expressing choice and instilling trust in the democratic process. As democratic institutions gain credibility and acceptance worldwide, ensuring the integrity and transparency of the voting system becomes paramount. However, numerous instances in various countries have highlighted deficiencies in traditional voting methods, ranging from transparency issues to logistical challenges. To address these shortcomings and bolster confidence in the voting process, leveraging technology emerges as a promising solution.

In response to the imperfections of traditional voting systems, innovative approaches leveraging technology have emerged to modernize the voting process. Among these, blockchain technology offers a secure, efficient, and transparent platform for conducting elections. By harnessing the decentralized nature of blockchain, along with robust cryptographic techniques, voting systems can ensure the integrity of the electoral process while safeguarding the anonymity and privacy of voters. Moreover, the convenience and accessibility afforded by blockchain-based e-voting platforms have the potential to significantly increase voter turnout, particularly in regions with historically low participation rates.

India, with its vast population and diverse electorate, presents a compelling case for the adoption of blockchain-based voting solutions to address longstanding challenges in the electoral process. With millions of eligible voters yet to participate in elections due to logistical constraints and concerns over fairness, a technologically advanced voting system holds promise in overcoming these barriers. By leveraging web-based applications built on blockchain, coupled with robust security measures and user-friendly interfaces, India can enhance voter engagement, ensure the accuracy of election results, and uphold the principles of democracy.

1.1 Existing System

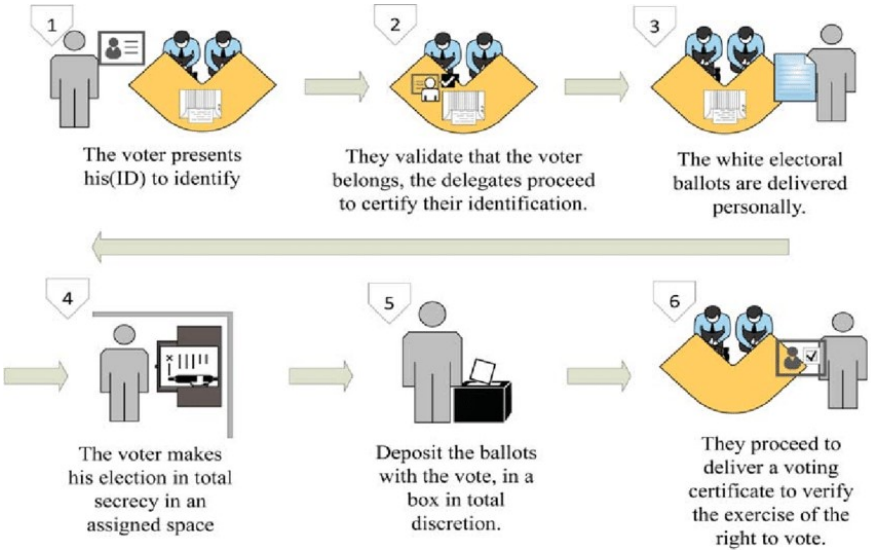

Figure 1.1 Existing System

The existing system of electronic voting primarily relies on centralized or semi-centralized architectures, often utilizing proprietary software solutions or custom-built platforms. These systems typically involve the use of electronic voting machines (EVMs) or online voting portals to facilitate the casting and tabulation of votes. However, despite their widespread adoption in various democratic processes worldwide, existing electronic voting systems are often criticized for their lack of transparency, security vulnerabilities, and potential for manipulation.

In many cases, electronic voting systems lack robust mechanisms for ensuring the integrity and privacy of votes cast, raising concerns about the accuracy and fairness of election outcomes. Additionally, centralized control over voting infrastructure and data storage poses risks of tampering or unauthorized access, undermining public trust in the electoral process. Furthermore, the proprietary nature of many existing electronic voting systems limits transparency and independent scrutiny, making it difficult to verify the accuracy of election results.

Moreover, the reliance on electronic voting machines with limited auditability and susceptibility to technical failures has led to controversies and legal challenges in various jurisdictions. Issues such as software bugs, hardware malfunctions, and inadequate security protocols have raised doubts about the reliability and trustworthiness of electronic voting systems in ensuring the democratic principles of free and fair elections.

Despite these challenges, efforts are underway to improve existing electronic voting systems through enhancements in security protocols, transparency measures, and auditability features. Additionally, the emergence of blockchain technology has sparked interest in exploring decentralized and tamper-resistant solutions for electronic voting, offering the potential to address many of the shortcomings associated with centralized voting systems.

1.2 Disadvantages of Existing System

Despite their widespread use, existing electronic voting systems suffer from several significant disadvantages that hinder their effectiveness and integrity. These drawbacks include:

Figure 1.2 Problem Illustration

- **Security Vulnerabilities**: Many existing electronic voting systems are susceptible to various security threats, including hacking, malware attacks, and manipulation of voting data. Weak encryption protocols, inadequate authentication mechanisms, and centralized control over voting infrastructure contribute to these vulnerabilities, raising concerns about the confidentiality and integrity of election results.
- **Lack of Transparency**: Centralized or semi-centralized electronic voting systems often lack transparency in the voting process, making it challenging for stakeholders to verify the accuracy and fairness of election outcomes. Limited access to voting data, proprietary software, and opaque decision-making processes undermines public trust in the electoral process and raise suspicions of fraud or manipulation.
- **Risk of Manipulation**: The centralized nature of many electronic voting systems poses a significant risk of manipulation by internal or external actors. Malicious actors could exploit vulnerabilities in voting software or hardware, tamper with voting data, or influence election results through unauthorized access to voting infrastructure.
- **Limited Auditability**: Existing electronic voting systems often lack robust auditability features, making it difficult to detect and investigate irregularities or discrepancies in election results.
- **Accessibility Challenges**: While electronic voting systems promise to enhance accessibility and convenience for voters, they may inadvertently exclude certain segments of the population, such as elderly voters or those with disabilities. Issues such as digital literacy barriers, inadequate provision of accessible voting options, and concerns about the security and privacy of electronic voting platforms can disenfranchise vulnerable groups and undermine the inclusivity of the electoral process

2. LITERATURE SURVEY

[1] Friðrik Þ. Hjálmarsson, Gunnlaugur K. Hreiðarsson, Mohammad Hamdaqa, and Gísli Hjálmtýsson propose a novel approach to electronic voting. They introduce a decentralized e-voting system that prioritizes fairness, privacy, transparency, and flexibility, leveraging blockchain technology to address the limitations of traditional systems. By evaluating popular blockchain frameworks, they aim to construct a secure and cost-effective e-voting system, demonstrated through a detailed case study illustrating the potential of distributed ledger technologies to enhance the integrity of nationwide elections.

[2] FREYA SHEER HARDWICK, The author presents the first step of e-voting using a decentralized e-voting system with voter privacy rights, employing smart contracts and PKIs for verification and digital signatures that are extremely reliable and effective. The protocol has been created to follow basic e-voting principles, allow for a certain amount of decentralization, and allow voters to modify or update their votes.

[3] King-Hang Wang, Subrata K. Mondal, Ki Chan, and Xiaoheng Xie, highlight electronic voting as a longstanding area of research spanning over 30 years, yet its widespread adoption remains a distant prospect due to significant security and trust challenges. Their study delves into the complexities of electronic voting systems, emphasizing the paramount importance of security measures to engender confidence among voters. By analyzing existing literature, they compile a comprehensive list of security specifications crucial for developing secure electronic voting systems

[4] D. A. Gritzalis, underscores the potential of electronic voting (e-voting) to enhance democracy in modern information societies. However, he emphasizes the necessity for e-voting systems to comply with legal and regulatory frameworks while meeting user requirements. Gritzalis outlines two key objectives: firstly, to delineate constitutional requirements essential for the development of e-voting systems for general elections, ensuring adherence to court-acceptable design guidelines. Secondly, employing the

Rational Unified Process, Gritzalis aims to ascertain the requisite level of security for e-voting systems.

[5] Shekhar Mishra examines the utilization of electronic voting systems in Indian general and state elections, identifying security concerns as a major drawback. Specifically, they highlight the vulnerability to voter impersonation, where unauthorized individuals cast ballots using others' identities. Their proposed solution utilizes a 32-bit ARM 7 processor, leveraging Voter ID numbers and associated biometric data to authenticate voters. This innovative approach aims to enhance the security and integrity of the electoral process, particularly in the context of voter identification and verification.

[6] Patrick McCorry, Siamak F. Shahandashti, and Feng Hao introduce a groundbreaking smart contract tailored for boardroom voting, ensuring maximum voter privacy through decentralization and self-tallying. Dubbed the Open Vote Network and implemented on the Ethereum blockchain, this protocol represents a departure from previous blockchain e-voting approaches by eliminating the need for a trusted authority to compute the tally or safeguard voter privacy. Each voter retains control over the privacy of their vote, with compromise requiring unanimous collusion among all other voters. Additionally, the authors provide a comprehensive financial and computational breakdown, reinforcing the protocol's practicality and potential for widespread adoption.

3. PROPOSED METHOD

The proposed electronic voting system seeks to address the shortcomings of existing voting mechanisms by leveraging advanced technologies and innovative design principles to enhance security, transparency, and integrity in the electoral process. At its core, the system adopts a decentralized approach built on blockchain technology, offering a distributed ledger framework that ensures tamper-resistant storage and verification of voting data. By decentralizing control and eliminating single points of failure, the proposed system mitigates the risk of manipulation and enhances the trustworthiness of election outcomes.

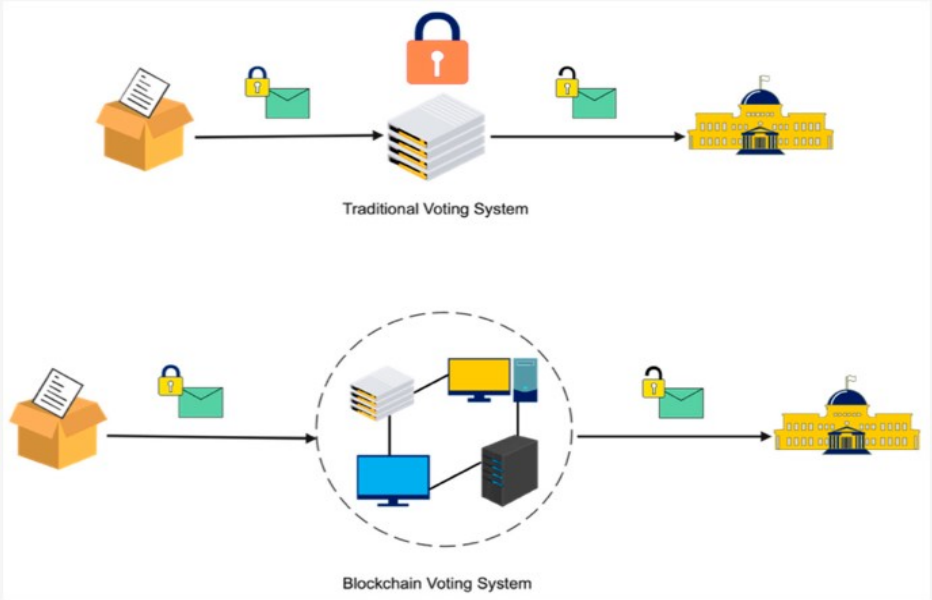

Figure 3. Existing System vs. Our Solution

3.1 Key Components of Proposed Method

- **Blockchain Integration:** The system leverages blockchain's immutability and decentralization to ensure the integrity of each vote. Once a vote is recorded, it

becomes a permanent and tamper-resistant part of the blockchain.

- **Decentralization**: By decentralizing the voting process, the proposed method eliminates the need for a central authority, reducing the risk of fraud or manipulation. Each node in the network verifies and stores votes independently.

- **Tamper-Resistant Platform:** The blockchain-based platform creates a tamper-resistant environment, preventing unauthorized access, tampering, or manipulation of votes. The transparency of the blockchain ensures the trustworthiness of the entire voting system.

3.2 Advantages of Proposed Method

The proposed electronic voting system offers several advantages over traditional voting methods, leveraging cutting-edge technology to enhance the integrity, accessibility, and efficiency of the electoral process. Here are some key benefits of the proposed system:

- **Enhanced Security**: By utilizing blockchain technology and cryptographic techniques, the proposed system ensures the security and integrity of voting data. The decentralized nature of the blockchain network reduces the risk of tampering or manipulation, while robust encryption mechanisms protect the confidentiality of voter information and ballot data.
- **Transparency and Auditability**: The use of blockchain enables transparent and auditable voting processes, with all transactions recorded immutably on the distributed ledger. This transparency allows stakeholders to verify the integrity of election results and detect any irregularities or discrepancies, thereby enhancing trust in the electoral process.
- **Accessibility and Inclusivity**: The proposed system improves accessibility and inclusivity by enabling remote and mobile voting options. Voters can cast their ballots from anywhere, using any internet-enabled device, thereby eliminating barriers to participation such as geographical constraints or mobility issues. This

inclusivity promotes greater voter turnout and engagement in the democratic process.

- **Privacy Preservation**: While ensuring transparency, the proposed system also prioritizes the privacy and anonymity of voters. Secure authentication methods and encryption techniques protect voter identities and ballot choices, preventing unauthorized access or disclosure of sensitive information. This privacy-preserving approach instills confidence in voters and upholds their fundamental rights to privacy.
- **Reduced Costs and Efficiency**: Compared to traditional paper-based voting systems, the proposed electronic voting system offers cost savings and increased efficiency. By digitizing the voting process and automating administrative tasks, such as ballot counting and result tabulation, the system streamlines election operations and reduces the resources required for conducting elections.
- **Resilience to Fraud and Manipulation**: The decentralized nature of the proposed system makes it resistant to fraud and manipulation. With no single point of control or failure, the blockchain network prevents unauthorized alterations to voting records and ensures the integrity of election outcomes. This resilience enhances confidence in the electoral process and safeguards democratic principles.

4. SYSTEM DESIGN

4.1 System Requirements

The Requirements to implement the system are as follows.

4.1.1 Hardware Requirements

- PROCESSOR : Intel i5
- RAM : 4GB
- HARD DISK : 16GB

4.1.2 Software Requirements

- OPERATING SYSTEM : Windows
- BACK-END : Python3
- DATABASE : Django
- FRAMEWORKS : Blockchain
- FRONT-END : HTML, CSS, Json

4.1.3 Specific Requirements

- Merkel tools
- Block mining
- Django utils

4.2 Project Modules

- Registering to the Network: The user has to register to the network by providing, valid details like username, email, and valid Voter-ID

- Voter-ID Input and Verification: After registering to the network, the user can access the website; and enter the registered Voter-ID, once its fed to the website, it verifies and validates user whether to proceed in the voting process or not

- Casting vote: Authorized users are allowed to select the election party and the representative of the respective region and cast their vote.

- Poll results: Once the election poll is completed, the user can access the website to notice which election party has won the election poll.

4.3 System Architecture

The system architecture is divided into three parts, namely

- Pre-voting phase
- Voting phase
- Post-voting phase

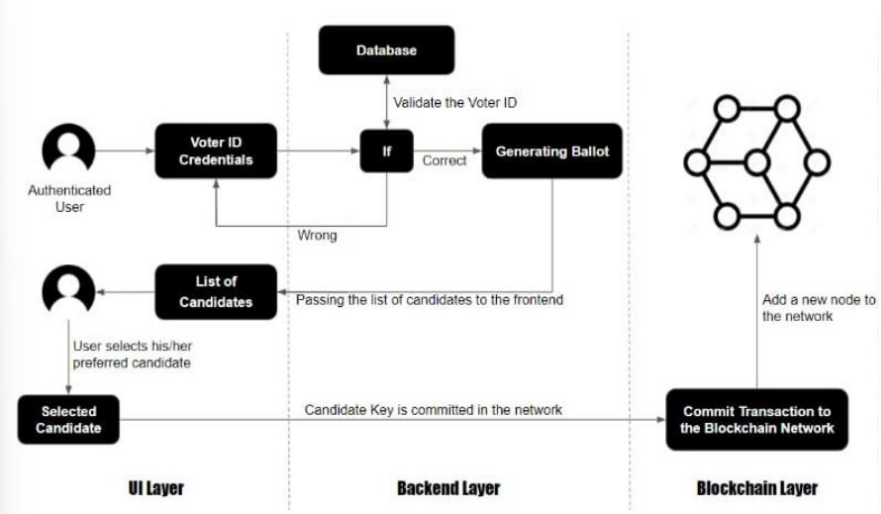

Figure 4. System Working Process

4.3.1 Pre-voting phase:

- A user must first register with the network in order to cast a vote in the aforementioned chain of networks.

- After connecting to the network, the user needs to create an ID using the details from his official voter ID. At this stage, the information from the previous blocks verifies whether or not the information from the new block (the user) matches the data in the database. If the validation of the new block is successful, the network allows it to move on to the next round of voting.

- Before proceeding to the next voting phase, the user needs to authenticate his identity using Voter ID.

4.3.2 Voting phase:

- Once Voter ID authentication is successful, the user can cast a ballot. After casting his vote, the user is unable to log in and utilize the network to cast another one. The user's vote will be secured using public key encryption. Smart contracts forbid this re-voting with the same ID.

- There is no way to rig the vote. After casting a ballot, it is almost impossible for someone to amend their vote because doing so would mean changing the entire block system and needing authentication from every network node.

4.3.3 Post-voting phase:

The user can use the website to see which political party won the election poll after it has been completed.

5. IMPLEMENTATION

5.1 Algorithms

5.1.1 SMTP :

- **Mail composition**: A user can send emails by utilizing a Mail User Agent (MUA) to create an electronic mail message. Mail User Agent is the application used for sending and receiving mail. The communication consists of two parts: the header and body. The body of the message is its main component, while the header is where information about the sender and recipient addresses is contained. The header also includes descriptive information such as the subject of the message. In this case, the message body is similar to a letter, and the header is similar to an envelope with the recipient's address on it.
- **Submission of Mail**: The mail client then sends the finished email to the SMTP server using SMTP on the TCP port after finishing the email's composition.
- **Delivery of Mail**: The recipient's username and the domain name are the two components of an email address. For instance, anurag@gmail.com, where "gmail.com" is the domain name and "anurag" is the recipient's username. Mail will be sent to the Mail Transfer Agent (MTA) if the recipient's email address's domain name differs from the sender's domain name. The MTA will locate the target domain and relay the email there. To retrieve the target domain, it looks up the MX record in the Domain Name System. The IP address and domain name of the recipient's domain are listed in the MX record. The MTA establishes a connection with the exchange server to relay the message after locating the record.
- **Receipt and Processing of Mail**: Once the incoming message is received, the exchange server sends it to the mail delivery agent (incoming server), which archives the email and makes it available for user retrieval.
- **Access and Retrieval of Mail**: Using MUA (Mail User Agent), it is possible to recover the email that was saved in MDA. MUA is accessible with a login and password.

5.1.2 SHA-256 Bit Encryption Algorithm:

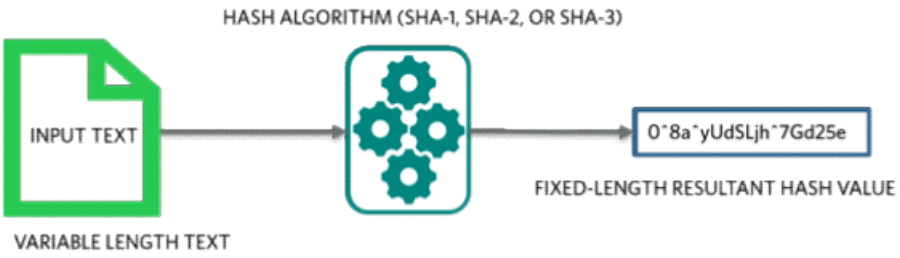

Figure 5.1. SHA 256 Working

- The National Security Agency created the SHA-2 (secure hash algorithm 2) technique in 2001 to replace the SHA-1 algorithm. The SHA-256 algorithm is a variation of this technique. The proprietary cryptographic hash method SHA-256 generates a 256-bit result.
- Through encryption, data is transformed into a secure format that can only be read by the recipient with the right key. When data is encrypted, it can be as large as desired and is often the same size as unencrypted data. On the other hand, data of any size is transformed into data of a predetermined size via hashing. A 512-bit data string, for example, would be reduced to a 256-bit string via SHA-256 hashing.
- During cryptographic hashing, the hashed data is changed to the point where it is completely unreadable. It is difficult to return the aforementioned 256-bit hash to its original 512-bit state. The most common defense is confirming the content of data that must be kept confidential. For example, hashing is used to verify the authenticity of secure data and messages. The hash code of a secure file can be made public so that those who download it can confirm it is authentic without disclosing the contents of the file. Hashing is employed in a manner akin to the verification of digital signatures.
- Password verification is a critical use of cryptographic hashing. An abundance of unprotected credentials awaits any hacker who manages to obtain access to a plain-text document containing user passwords. This makes having password hash values on hand safer. After that, the hash value produced from a user's password

is computed and contrasted with the table. If a password matches one of the previously saved hashes, it can be used to obtain access.

5.1.3 Merkle Hash:

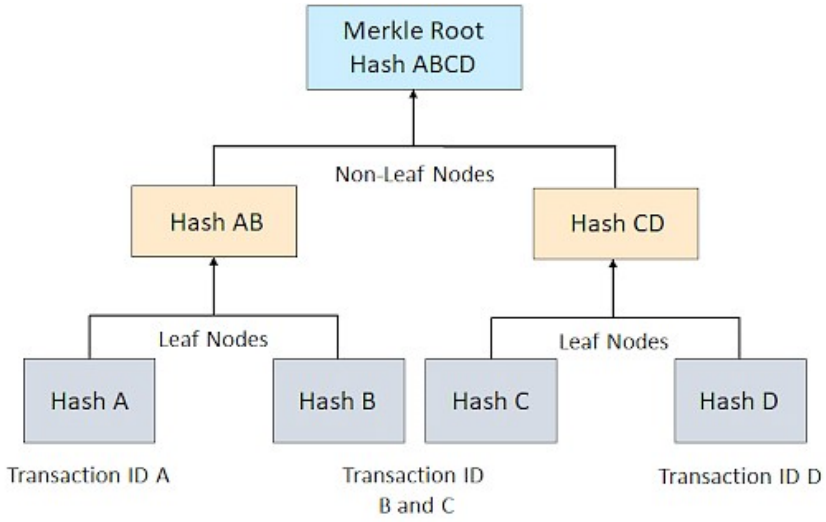

Figure 5.2. Merkel Hash Working

- A Merkle tree builds a digital fingerprint of the entire set of operations by adding up all the transactions in a block and determining if a given transaction is part of the block.

- In order to generate Merkel trees, node pairs are repeatedly hashed until the Merkel root—the single hash—remains. They are built from the bottom up using transaction IDs, which are hashes of the individual transactions. Every non-leaf node is a hash of its prior hash, whereas every leaf node is a hash of transactional data.

5.2 Functional Modules

The functional modules of e-voting using blockchain are:

5.2.1 Input Voter ID

To caste vote the voters need to register their votes before the elections using their Voter ID. Only the registered candidates are allowed to vote. The voters need to enter the Voter ID as the input in order to cast their vote.

```
def authentication(request):
    voter_id = request.POST.get(“voter_id”)
    details = {'success': False}
    try:
        voter = Voters.objects.get(uuid = voter_id)
        request.session['uuid'] = voter_id
        print(str(voter.dob))
        birthdate =str(voter.dob)
        year,month,day= map(int, birthdate.split("-"))
        today = datetime.date.today()
        age = today.year - year - ((today.month, today.day) < (month, day))
        render_html = loader.render_to_string('candidate_details.html', {'details': voter})
        if age<18:
            details = {
                'error': 'You are not eligible to vote.'
            }
        elif voter.vote_done:
            details = {
                'error': 'You have already casted your vote.'
            }
        else:
            details = {
```

```
            'success': True,
            'html': render_html,
            'details': model_to_dict(voter)
        }
    except:
        details = {
            'error': 'Invalid Voter ID, Please Enter Valid Number!'
        }
    return JsonResponse(details)
```

5.2.2 OTP Validation

After entering the valid Voter ID it is verified through otp it is sent to the candidate mail id which is provided at the time of registration.

This otp process is done with the help of SMTP.SMTP is an application layer protocol. The client who wants to send the mail opens a TCP connection to the SMTP server and then sends the mail across the connection. The SMTP server is an always-on listening mode. As soon as it listens for a TCP connection from any client, the SMTP process initiates a connection through port 25. After successfully establishing a TCP connection the client process sends the mail instantly.

```
# --------- Send otp for email verfication -----------
def send_otp(request):
    email_input = request.GET.get('email-id')
    [success, result] = send_email_otp(email_input)
    #[success, result] = [True, '0']
    json = {'success': success}
    if success:
        request.session['otp'] = result
        request.session['email-id'] = email_input
        request.session['email-verified'] = False
    else:
```

```
        json['error'] = result
    return JsonResponse(json)

# -------- Verify email with provided otp ----------
def verify_otp(request):
    otp_input = request.GET.get('otp-input')
    json = {'success': False}
    if otp_input == request.session['otp']:
        voter = Voters.objects.get(uuid = request.session['uuid'])
        voter.email = request.session['email-id']
        voter.save()
        json['success'] = True
        request.session['email-verified'] = True
    return JsonResponse(json)
```

5.2.3 Private key authentication

After successful completion of otp validation now the candidate able to see the party names and symbols along with the nota. If they click on the party which they are going to vote it will ask for authentication that is known as private key authentication where they need to enter the private key which they have received to their registered email id. The generation of private key is done through hashing.

Hashing is the procedure of interpreting a given key into a code. A hash function is used to substitute the data with a freshly produced hash code. Furthermore, hashing is the practice of taking a string or input key, a variable generated for saving narrative information, and defining it with a hash value, which is generally decided by an algorithm and create a much shorter string than the original.

Hashing is generally a one-way cryptographic function. Because hashes are irreversible, understanding the output of a hashing method does not enable us to regenerate the contents of a file. It allows us to assess whether two files are same without understanding their contents.

The use of hashing in information security and internet authentication is a common practice.

For example, it can be used to securely save passwords in a database, but can also provide the

security of other element of information including files and documents.

The hashing data structure allows arrays to effectively find and store information, supporting an effective structure for finding and storing information. Suppose that it can have a list of 20,000 numbers and it is asked to look for a specific number in that list and it can scan each number in the list to view if it matches the number that it is entered.

Hashing is the procedure of transforming a string of characters into a frequently shorter and fixed-length value. The why of using hashed keys to search for element in a database is that discovering the item using its original value is more time-consuming than using the shorter hashed key.

It can be used to locate or store elements effectively in collections when searching for them. For example, if it can have a list of 10,000 English words and want to check if a given word is between them, it will be inefficient to compare the given word to all 10,000 items until a match is discovered.

An array can be indexed by the values of the keys of a range, which is known as hashing. The modulo operator will be used to acquire a range of key values. In this case, it can store the following items in a 20x20 hash table. Each item is formatted as a (key, value).

Hashing algorithms including MD5, SHA-1, SHA-2, NTLM, and LANMAN are all generally used in today's world. A message digest is divided down into 5 versions, this one being MD5. In the previous, MD5 was one of the most famous hashing algorithms. MD5 needs 128 bits for its outputs.

In hashing, each bit in the data block is transformed into a fixed-size bit string value. A file includes data blocks. There is a risk that two multiple inputs will create the same hash value. This is called a collision, which appears when two multiple inputs support the same hash value.

```
def blockchain(request):
    blocks = Block.objects.all()
```

```
    return render(request, 'blockchain.html', {'blocks':blocks})

def block_info(request):
    try:
        block = Block.objects.get(id=request.GET.get('id'))
        confirmed_by = (Block.objects.all().count() - block.id) + 1

        votes = Vote.objects.filter(block_id=request.GET.get('id'))
        vote_hashes = [SHA3_256.new((f'{vote.uuid}|{vote.vote_party_id}|{vote.timestamp}').encode('utf-8')).hexdigest() for vote in votes]

        root = MerkleTools()
        root.add_leaf([f'{vote.uuid}|{vote.vote_party_id}|{vote.timestamp}' for vote in votes], True)
        root.make_tree()
        merkle_hash = root.get_merkle_root()
        tampered = block.merkle_hash != merkle_hash

        context = {
            'this_block': block,
            'confirmed_by': confirmed_by,
            'votes': zip(votes, vote_hashes),
            're_merkle_hash': merkle_hash,
            'isTampered': tampered,
        }
        return render(request, 'block-info.html', context)
    except Exception as e:
        print(str(e))
        return render(request, 'block-info.html')
```

5.2.4 Casting vote

After successful completion of otp validation now the candidate able to see the party names and symbols along with the nota. If they click on the party which they are going to vote it will ask for authentication that is known as private key authentication where they need to enter the private key which they have received to their registered email id. The generation of private key is done through hashing.

```
# ---------- Show Vote count so far -----------
def show_result(request):
    vote_result = vote_count()
    vote_result = dict(reversed(sorted(vote_result.items(), key = lambda vr:(vr[1], vr[0]))))
    results = []
    political_parties = PoliticalParty.objects.all()
    i=0
    for party_id, votecount in vote_result.items():
        i+=1
        party = political_parties.get(party_id = party_id)
        results.append({
            'sr': i,
            'party_name': party.party_name,
            'party_symbol': party.party_logo,
            'vote_count': votecount
        })
    return render(request, 'show-result.html', {'results': results})
```

6. EXPERIMENT RESULTS

6.1 Experiment Screenshots

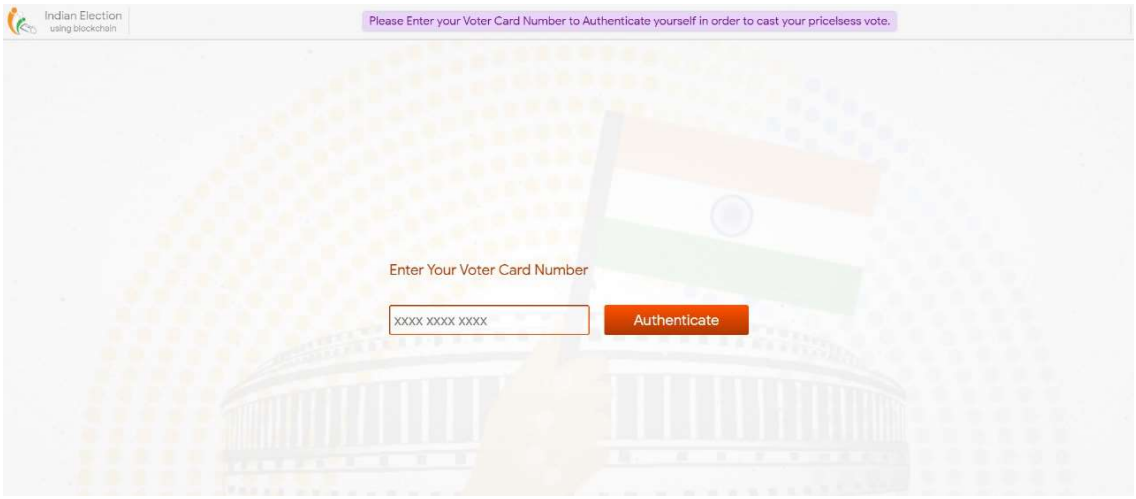

Figure 6.1. Login Page

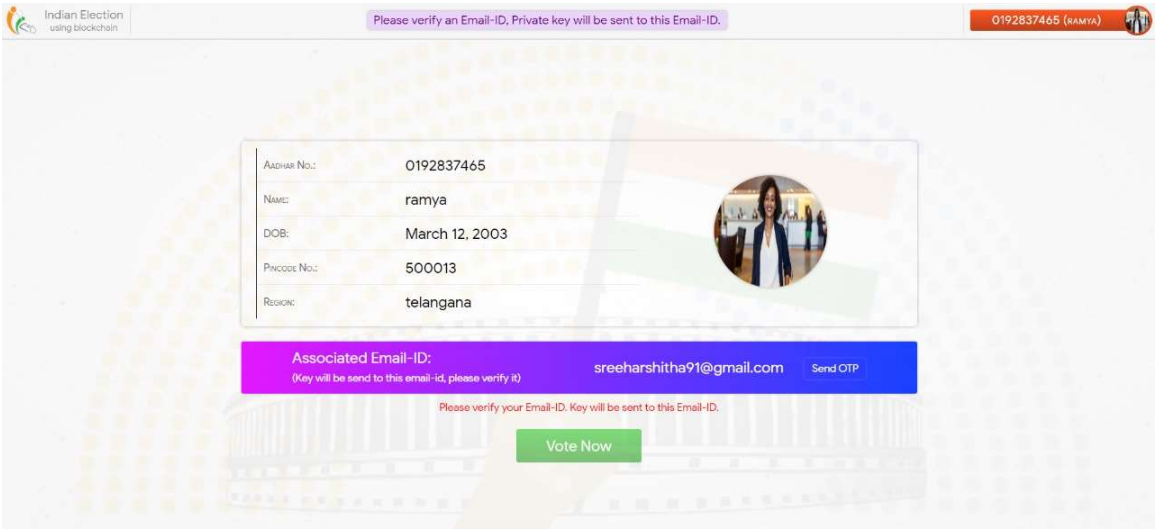

Figure 6.2. Dashboard

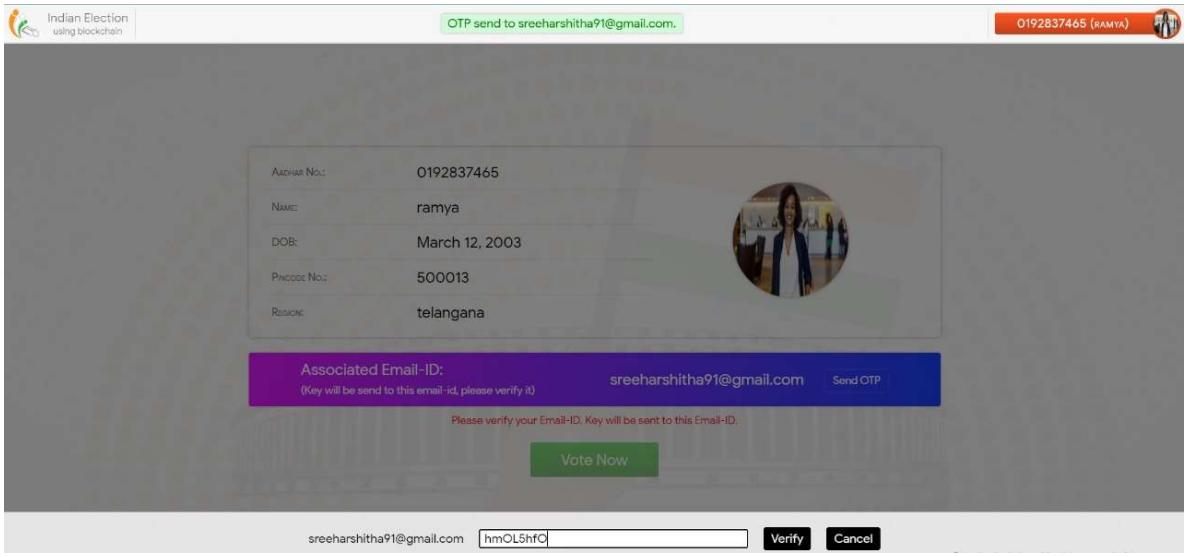

Figure 6.3 Email Verification

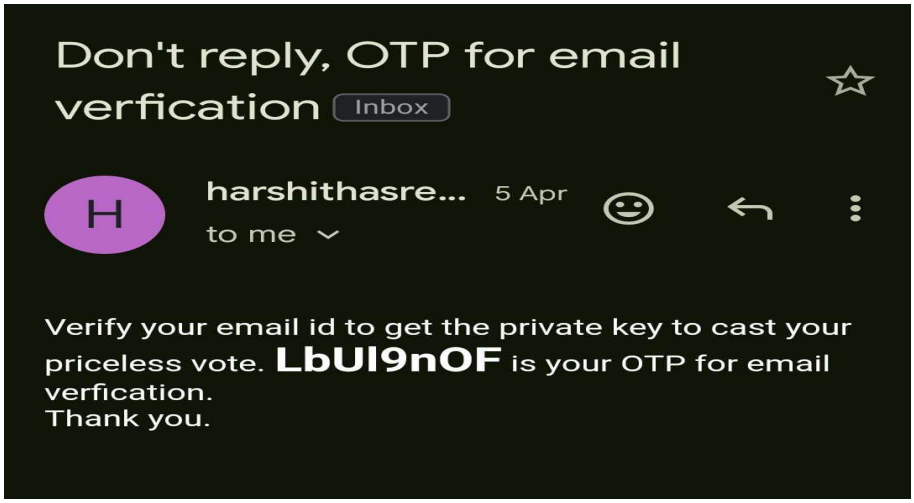

Figure 6.4 OTP for Email Verification

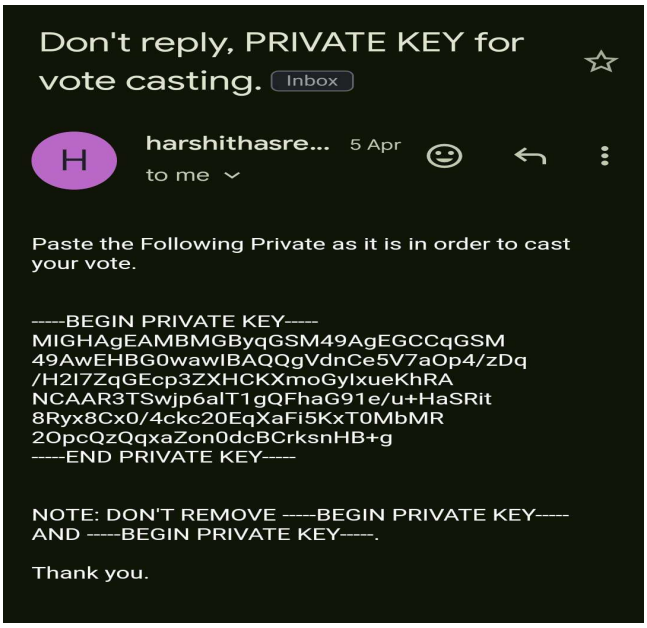

Figure 6.5 Private Key for Vote Casting

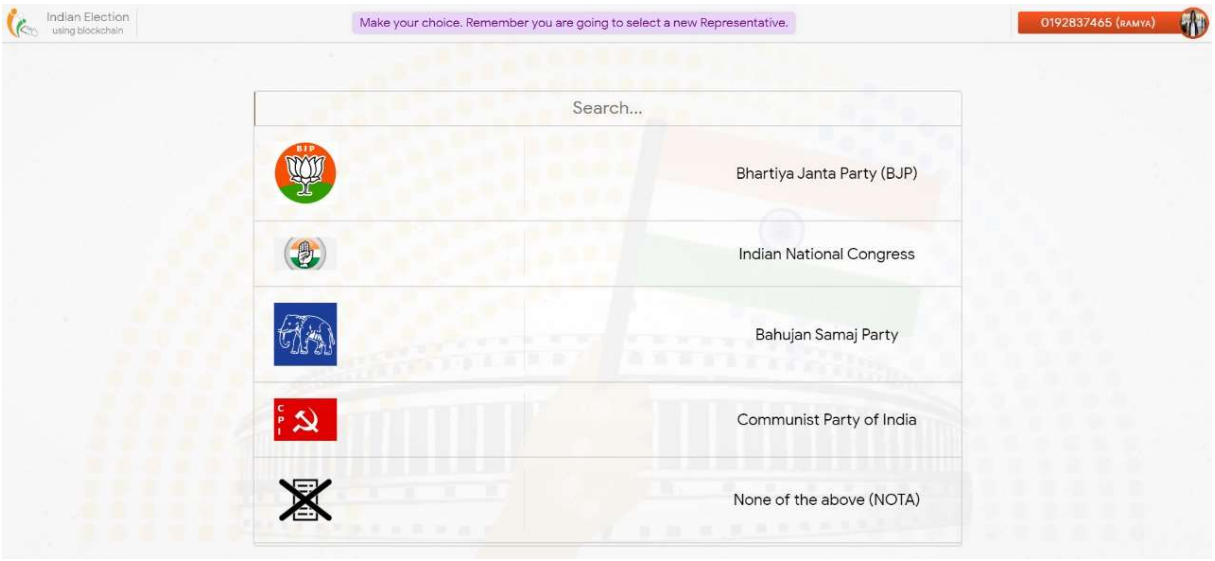

Figure 6.6 Voting Page

Figure 6.7 Voting Using Private Key

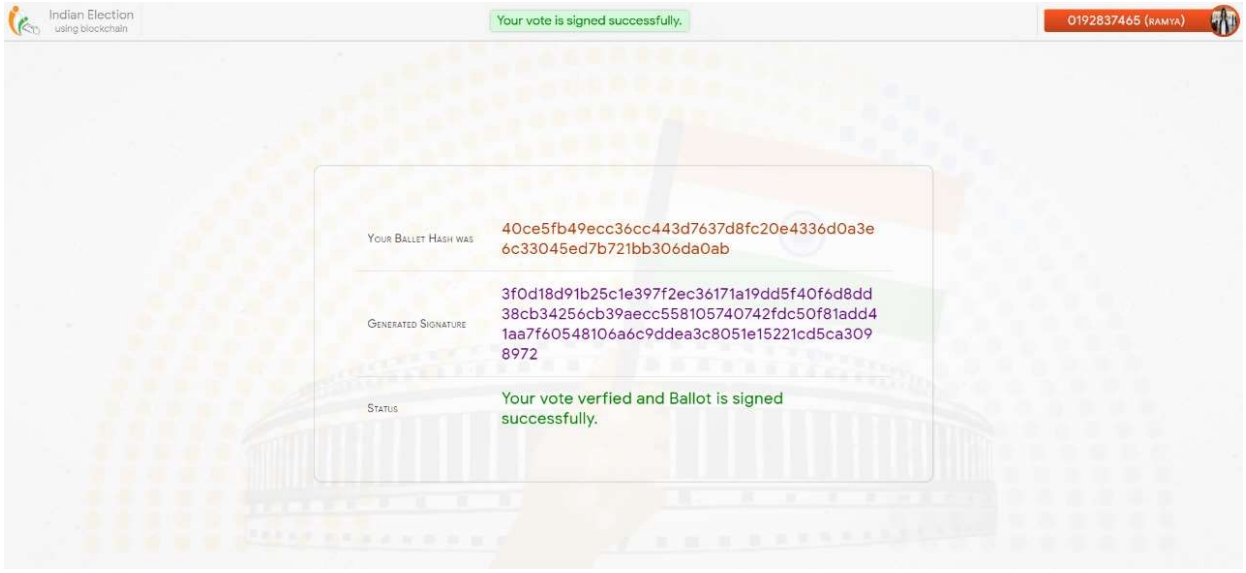

Figure 6.8 Voting Success Page

Figure 6.9 Voting Count Page

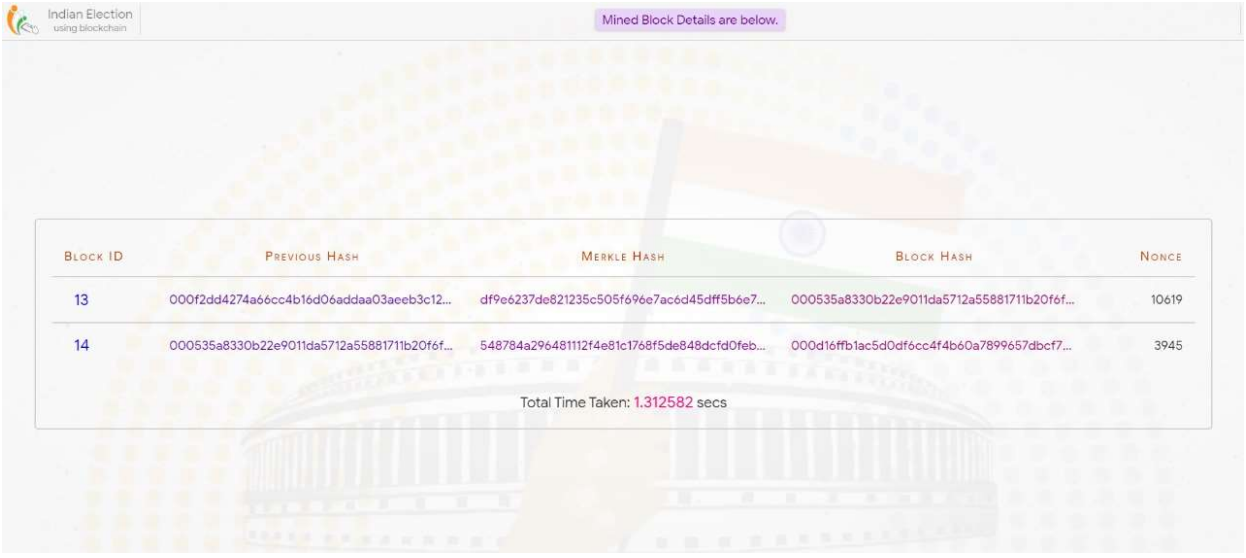

Figure 6.10 Mined Block Details Page

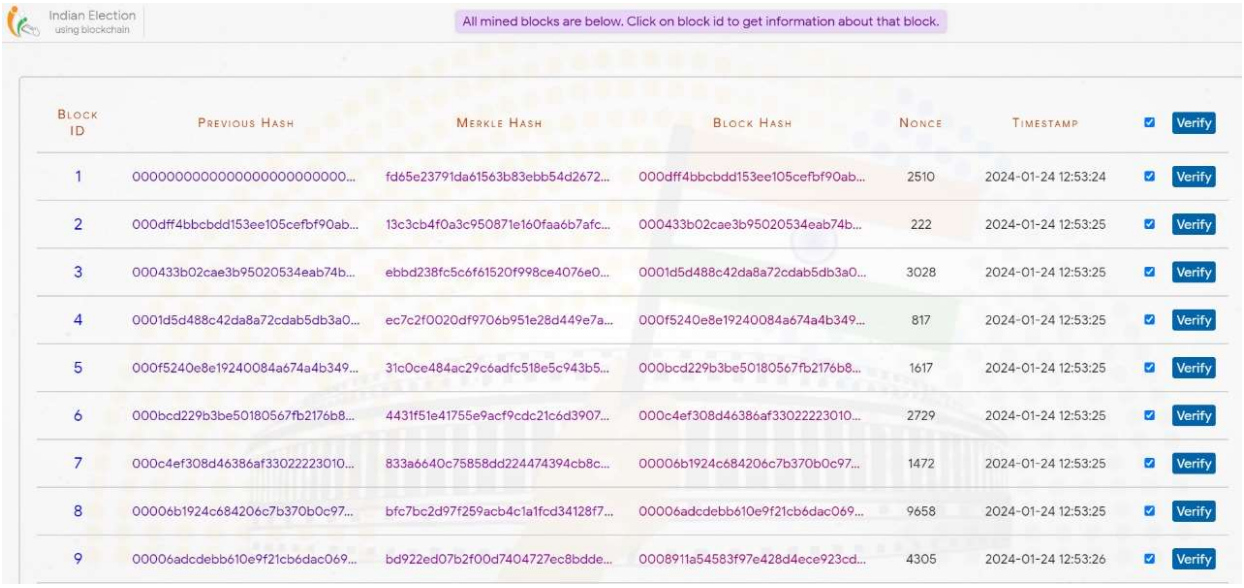

Figure 6.11 All Mined Blocks

6.2 Parameters

- **Cost-Effectiveness (CE):**
  Measure of the economic efficiency, considering costs associated with development, deployment, and maintenance per vote.

- **Privacy (P):**
  Evaluation of how well the system ensures the privacy of voter information, including the anonymity of votes.

- **Security (S):**
  Measurement of the effectiveness of security measures in preventing unauthorized access, tampering, or manipulation of the voting system.

- **Efficiency (E):**
  Analysis of the system's performance in terms of speed, responsiveness, and overall efficiency in managing the voting process.

6.3 Parameter Formulae

- **Cost-Effectiveness (CE):**

$$CE = \frac{TotalProjectCost}{\text{Number of votes}} X\ 100$$

- **Privacy (P):**

$$\text{P} = \frac{Number\ of\ Votes\ with\ Protected\ Identity}{Total\ votes} X\ 100$$

- **Security (S):**

$$S = \frac{Number\ of\ Verified\ Votes}{\text{Total votes}} X\ 100$$

- **Efficiency (E):**

$$\text{E} = \frac{Total\ Votes\ Cast}{Total\ Registered\ Voters} X\ 100$$

7. TESTING

7.1 Methods of Testing

The various strategies that were used in testing this software were as follows:

1. Unit testing
2. Integration testing
3. Validation testing
4. User validation testing

7.1.1 Unit Testing

Unit testing, also known as component testing refers to tests that verify the functionality of a specific section of code, usually at the function level. In an object-oriented environment, this is usually at the class level, and the minimal unit tests include the constructors and destructors. Unit testing is a software development process that involves synchronized application of a broad spectrum of defect prevention and detection strategies in order to reduce software development risks, time, and costs. The following Unit Testing Table shows the functions that were tested at the time of programming. The first column gives all the modules which were tested, and the second column gives the test results. Test results indicate if the functions, for given inputs are delivering valid outputs.

Function Name Tests Results feeding the valid enrolled Voter ID and authorise user using blockchain hash techniques; output is tested successful when the user can cast his vote through the website.

Function name	Test results
Feed enrolled Voter ID	Tested for different input of Voter ID verification
Verify the details	Authorising only valid users to vote
Display result	Output is to cast vote only once by a single user

Table 7.1 Function Name and Test Results

7.1.2 Integrating testing

Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Software components may be integrated in an iterative way or all together ("big bang"). Normally the former is considered a better practice since it allows interface issues to be located more quickly and fixed. Integration testing works to expose defects in the interfaces and interaction between integrated components (modules). Progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a system.

7.1.3 Validation Testing

At the culmination of integration testing, software is completed assembled as a package. Interfacing errors have been uncovered and corrected. Validation testing can be defined in many ways; here the testing validates the software function in a manner that is reasonably expected by the customer. In software project management, software testing, and software engineering, verification and validation (V&V) is the process of checking that a software system meets specifications and that it fulfills its intended purpose. It may also be referred to as software quality control.

7.1.4 User Acceptance Testing

Performance of an acceptance test is actually the user's show. User motivation and knowledge are critical for the successful performance of the system. The above tests were conducted on the newly designed system performed to the expectations. All the above testing strategies were done using the following test case design.

7.2 Unit Testing Test Cases

Input Voter ID test case

Test case	1
Name of the test	Input Voter ID
Input	Valid unique ID
Expected output	Input Voter ID feed by the user
Actual output	Valid Voter ID is accepted as enrolled in the database
Result	Successful

Table 7.2: input Aadhar test case

Email OTP authentication test case

Test case	1
Name of the test	Email OTP authentication
Input	Valid/ enrolled email-id
Expected output	Obtain OTP to the registered email
Actual output	Receiving unique OTP from the enrolled email ID
Result	Successful

Table 7.3: Email OTP authentication test case

Private key verification Test case

Test case	1
Name of the test	Private key verification
Input	Valid/ enrolled email-id
Expected output	Obtaining unique hash value
Actual output	Receiving unique hash value using blockchain from the enrolled email ID
Result	Successful

Table 7.4: Private key verification Test case

8. EXPERIMENTAL RESULTS AND JUSTIFICATION

8.1 Usability Testing

Figure 8.1 Experiment Finding 1

Findings:

- The majority of participants expressed high satisfaction levels with the online voting platform, with 80% reporting being either "very satisfied" or "satisfied.".
- Users rated the ease of use of the platform quite positively, with an average rating of 8.5 out of 10, indicating a user-friendly voting experience.
- Usability testing revealed that while the majority of users found the platform intuitive and user-friendly, there were notable areas for improvement identified, particularly in visual layout and color schemes.

8.2 Security and Integrity Assessments

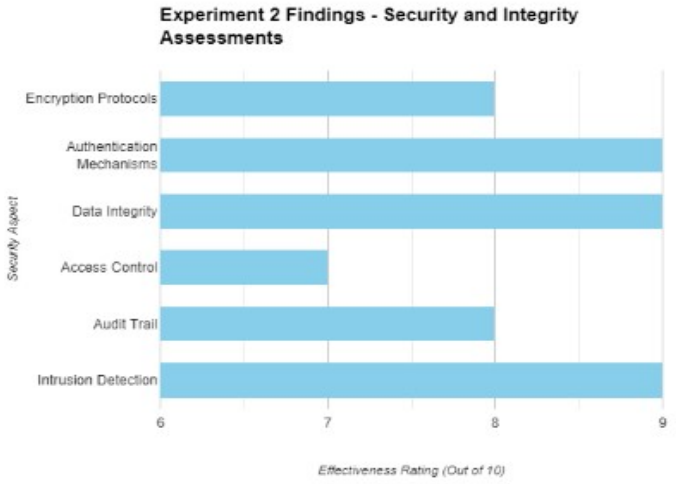

Figure 8.2 Experiment Finding 2

Findings:

Experiment 2 findings indicate robust security measures in the online voting system. Encryption, authentication, data integrity, audit trail, and intrusion detection are highly effective, with access control showing moderate effectiveness. Overall, the system demonstrates resilience to cyber threats, ensuring confidentiality and integrity in the voting process.

8.3 Scalability and Performance Assessment

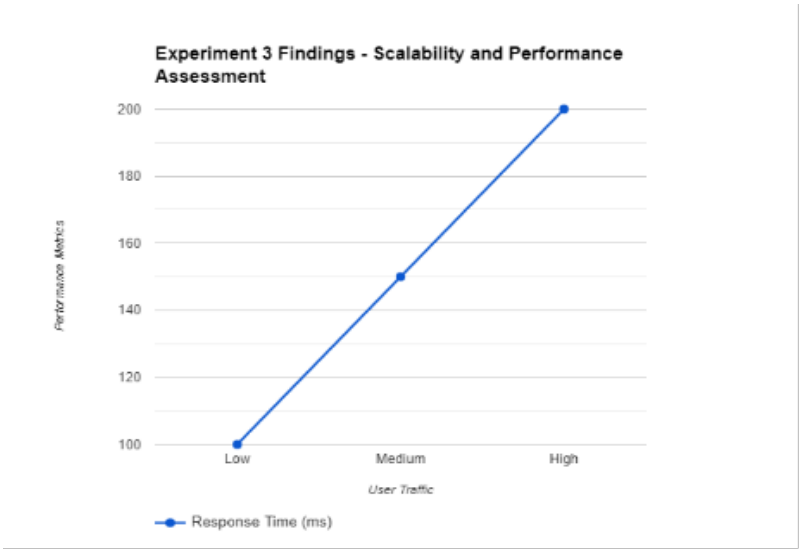

Figure 8.3 Experiment Finding 3

It refers to the results obtained from assessing the scalability and performance of the online voting system under varying levels of user traffic. These findings demonstrate the system's ability to handle increased loads with minimal degradation in response time, indicating its robust performance and scalability.

It highlight the system's efficiency in real-world voting scenarios, affirming its capability to maintain reliability and responsiveness even under peak load conditions.

8.4 Performance Analysis

Performance analysis is a specialist discipline involving systematic observations to enhance performance and improve the people voting and ease people difficulty in voting This voting system helps people who have smart mobiles and websites enabled laptops, also helps people from faraway places to vote conveniently from the place they are. It increases, the voting rate in the country and decreases the cost expenses of conducting live booth pooling in every place

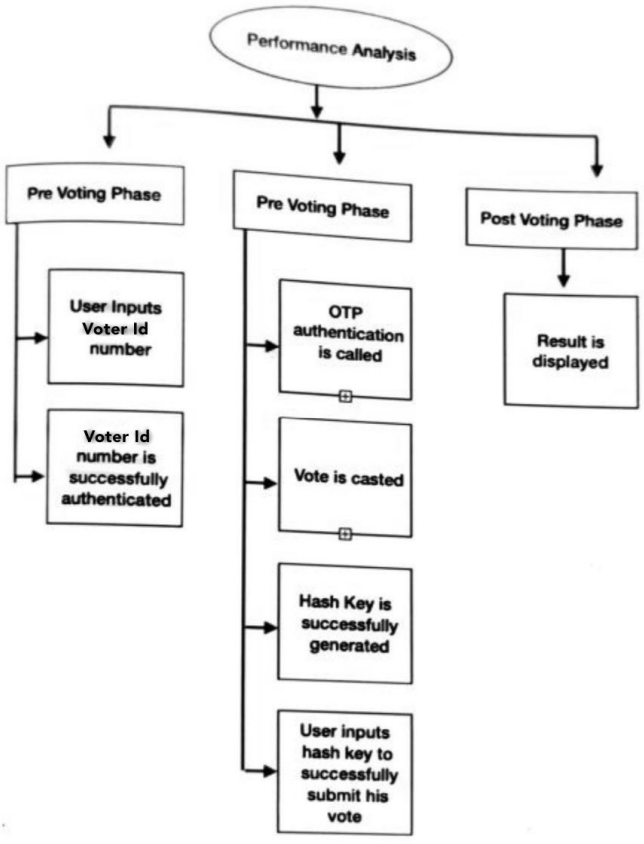

Figure 8.4 Performance Analysis

8.5 Parameters Comparison

Parameter	Our System	Existing System (Average)
Cost-Effectiveness (CE)	High ($0.10 - $0.50 per vote)	Low ($0.01 - $0.05 per vote)
Privacy (P)	Very High (95% - 100%)	Moderate (50% - 80%)
Security (S)	Very High (95% - 100%)	Moderate (70% - 90%)
Efficiency (E)	Moderate (60% - 80%)	High (80% - 95%)

Table 8 Parameter Formulae

9. Conclusion

The proposed framework provides complete security to the e-voting system, with the usage of Ethereum blockchain and smart contracts to provide added security to the system. Blockchain implementation prevents vote manipulation and provides privacy, integrity for voters to cast their vote. Smart contracts ensures that the voter can vote only once using his/her unique id (Aadhar number); with the convention of different security algorithms like SHA-256, Merkel hash and SMTP prototyping, enhances the security of the system. As a result, the voter is authorized to cast his/her vote from where ever they are; provides high security standards to the system and convenient and easier ways to vote.

Future work:

1. To the proposed existing system, additional biometrics (fingerprint, face authentication) can be added to enhance the security of the system.
2. Three step authentications can also be used to provide more security to the system

10. REFERENCES

1. [1] Blockchain-Based E-Voting System, Friðrik Þ. Hjálmarsson, Gunnlaugur K. Hreiðarsson, IEEE 11th International Conference on Cloud Computing (CLOUD), 2018
2. [2] Freya Sheer Hardwick, Apostolos Gioulis, Raja Naeem Akram, and Konstantinos Markantonakis, E-Voting with Blockchain: An E-Voting Protocol with Decentralization and Voter Privacy, Vol. 7, Issue 6, 2018, pp. 1561–1567.
3. [3] King-Hang Wang, Subrota K. Mondal, Ki Chan, and Xiaoheng Xie, A review of contemporary e-voting: requirements, technology systems, and usability, Ubiquitous International, vol. 1, issue 1, 2017, pp. 31–47.
4. [4] D. A. Gritzalis, Principles and requirements for a secure e-voting system, publication history, vol. 21, issue 6, 2002, pp. 539–556.
5. [5] Shekhar Mishra, Y. Roja Peter, Zaheed Ahmed Khan, and M. Renuka, Electronic Voting Machine using Biometric Finger Print with Voter ID Card Authentication, International Journal of Engineering Science and Computing, Vol. 7, Issue 3, 2017, pp. 5897–5899.
6. [6] A Smart Contract for Boardroom Voting with Maximum Voter Privacy, Patrick McCorry, Siamak F. Shahandashti and Feng Hao, International Conference on Financial Cryptography and Data Security, 2017
7. [7] An efficient and effective Decentralized Anonymous Voting System. Wei-Jr Lai, Ja-Ling Wu, ArXiv, 2018
8. [8] Blockchain-based Electronic Voting System Design with Smart Contracts Wan Auzan Bin Wabdulah; Syed Farid Syed Adnan. 2023 IEEE Symposium on Computers & Informatics (ISCI)
9. [9] E-Voting System Using Blockchain and Homomorphic Encryption Ramesh Naidu; Dileep Reddy Bolla; Prateek G; Sheetal S Harshini; Shreya A Hegde; Vallamkonda Venkata Sree Harsha 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon)

10. [10] E - Voting Using Blockchain Suganthi N; Gokul S; Shravanth E; Veena K 2023 2nd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)
11. [11] Online Voting System Using Blockchain S. Drakshayani; U. Vijayalakshmi; S. Rupa Sri; A. Srivani; and A. Vyshnavi 2022 International Conference on Electronics and Renewable Systems (ICEARS)