SIDDAGANGA INSTITUTE OF TECHNOLOGY, TUMAKURU- 3

(An Autonomous institution affiliated to Visvesvaraya Technological University- Belagavi, Approved by AICTE, Accredited by NAAC with 'A++' Grade, Awarded Diamond College Rating by QS I-GAUGE & ISO 9001:2015 certified)

MINI PROJECT REPORT

ON

"Secure Ballot: A Blockchain Based Voting Solution"

submitted in the partial fulfilment of the requirements for VI semester,
Bachelor of Engineering in Computer Science and Engineering

By

Dileep Kumar D	**1SI21CS039**
Karthik Gowda S	**1SI21CS050**
Nachiket	**1SI21CS065**

Under the guidance of
Dr. M B Nirmala Ph.D
Associate Professor.

Department of Computer Science and Engineering
(Program Accredited by NBA)

Academic Year: 2023-24

Siddaganga Institute of Technology, Tumakuru-3

(An Autonomous institution affiliated to Visvesvaraya Technological University- Belagavi, Approved by AICTE, Accredited by NAAC with 'A++' Grade, Awarded Diamond College Rating by QS I-GAUGE & ISO 9001:2015 certified)

Department of Computer Science and Engineering

(Program Accredited by NBA)

CERTIFICATE

This is to certify that the mini project entitled "**Secure Ballot: A Blockchain Based Voting Solution**" is a bonafide work carried out by **Dileep Kumar D (1SI21CS039) , Karthik Gowda S(1SI21CS050) , Nachiket(1SI21CS065)** of VI semester **Computer Science and Engineering**, **SIDDAGANGA INSTITUTE OF TECHNOLOGY** during the academic year 2022-2023.

Signature of the Guide
Dr. M B Nirmala Ph.D
Associate Professor.

Signature of the Convener
Prof. Thejaswini S M.Tech
Assistant Professor

Signature of the HOD

Dr. N R Sunitha
Prof. and Head, Dept. of CSE

Name of the Examiners: **Signature with Date**

1. **Prof.**
2. **Prof.**

ACKNOWLEDGEMENT

With great reverence, we express our sincere gratitude and salutations to his holiness **Dr. Sree Sree Shivakumara Swamigalu,** Founder of Sree Siddaganga Education society and **Sree Sree Siddalinga Swamigalu,** President of Sree Siddaganga Education society for their blessings.

First and foremost, we wish to express our deep sincere feelings of gratitude to our institution, **Siddaganga Institute of Technology,** for providing us an opportunity for completing our project successfully.

We are grateful to **Dr. M. N. Channabasappa,** Director and **Dr. Shivakumaraiah,** CEO, Siddaganga Institute of Technology-Tumakuru, for their kind cooperation and encouragement.

We express our kind thanks to **Dr. S V Dinesh,** Principal, Siddaganga Institute of Technology, Tumakuru for his encouragement towards students attitude.

We express our heartfelt thanks to **Dr. N R Sunitha,** Professor and Head, Department of Computer Science and Engineering, Siddaganga Institute of Technology, Tumakuru for her suggestions and advice.

We express our gratitude and humble thanks to our project guide,**Dr.M B Nirmala** Ph.D. **,** Professor, Department of Computer Science and Engineering, Siddaganga Institute of Technology, Tumakuru for guiding us to complete our Mini-project successfully.

We thank our mini project convener, **Mrs. Thejaswini S**, Assistant Professor, Department of Computer Science and Engineering, for allowing us to take up this idea as our Mini Project.

We are conscious of the fact that we have received cooperation in many ways from the **Teaching, Technical and supporting staff** of the Department of Computer Science and Engineering and we are grateful to all for their cooperation.

We express heartfelt gratitude to our **Parents and Friends** for their constant moral support and encouragement throughout this work.

ABSTRACT

In the pursuit of enhancing the integrity, transparency, and security of electoral processes, traditional voting systems have been scrutinized for their susceptibility to fraud, tampering, and lack of trust. "Secure Ballot: A Blockchain-Based Voting System" addresses these concerns by leveraging blockchain technology to create a decentralized, immutable, and transparent voting platform. This system ensures that each vote is securely recorded and verifiable, eliminating the possibility of manipulation and fostering public trust in the electoral process. By utilizing cryptographic techniques and distributed ledger technology, the Secure Ballot system provides a robust framework for secure voter authentication, vote casting, and real-time results auditing.

The proposed blockchain-based voting system offers significant advantages over conventional methods, including enhanced security, reduced administrative costs, and increased voter participation through remote voting capabilities. The system employs smart contracts to automate and enforce election rules, ensuring adherence to predefined protocols without the need for intermediaries. Additionally, it guarantees voter privacy while maintaining the transparency of the overall process, allowing stakeholders to independently verify the election results. This paper explores the technical architecture, implementation challenges, and potential impacts of the Secure Ballot system, presenting it as a viable solution for modernizing electoral practices and strengthening democratic processes.

Table of Contents

CHAPTER 1

INTRODUCTION

The integrity of democratic processes is a cornerstone of any functional society. Traditional voting systems, whether paper-based or electronic, have faced numerous challenges that threaten the reliability and transparency of elections. These systems are often plagued by issues such as ballot tampering, voter fraud, lack of voter privacy, and inefficiencies in vote counting and result verification. As the demand for more secure and transparent voting systems grows, the advent of blockchain technology offers a promising solution to address these concerns comprehensively.

Blockchain technology, initially developed as the underlying structure for cryptocurrencies like Bitcoin, has since evolved to demonstrate potential across various domains, including supply chain management, healthcare, finance, and now, electoral processes. At its core, blockchain is a decentralized, distributed ledger that records transactions in a secure, transparent, and immutable manner. Each transaction, or block, is cryptographically linked to the previous one, forming a chain that is virtually tamper-proof. This inherent security and transparency make blockchain an ideal candidate for developing a secure voting system.

The Secure Ballot system leverages blockchain's unique properties to create a robust, transparent, and secure voting platform. One of the primary advantages of using blockchain in voting systems is the elimination of a central authority. In traditional voting systems, central authorities are responsible for managing voter rolls, securing ballots, and counting votes, which introduces potential points of failure and avenues for corruption. By decentralizing these responsibilities, blockchain reduces the risk of single points of failure and makes the system more resilient to attacks.

In addition to decentralization, the immutability of blockchain ensures that once a vote is cast, it cannot be altered or deleted. This feature addresses the issue of ballot tampering, a significant concern in both paper-based and electronic voting systems. Furthermore, the transparency of blockchain allows all stakeholders to verify the results independently. Every transaction, including each cast vote, is recorded on a public ledger that can be audited by anyone, enhancing trust in the electoral process.

Voter privacy is another critical aspect that the Secure Ballot system aims to address. Traditional and electronic voting systems often struggle to balance transparency with voter anonymity. Blockchain, however, can employ advanced cryptographic techniques to ensure that while the votes are publicly verifiable, the identity of the voters remains confidential. This is achieved through the use of zero-knowledge proofs and cryptographic key pairs, ensuring that the system maintains the integrity of the vote without compromising voter privacy.

Moreover, the implementation of smart contracts within the blockchain framework allows for the automation of election rules and processes. Smart contracts are self-executing contracts with the terms of the agreement directly written into code. In the context of voting, smart contracts can enforce election rules, validate voter eligibility, and tally votes automatically. This reduces the need for human intervention and minimizes the risk of human error or bias, further ensuring the integrity of the election process.

The potential benefits of a blockchain-based voting system extend beyond security and transparency. The adoption of such a system can also lead to increased voter participation by facilitating remote voting. Traditional voting methods often require voters to be physically present at polling stations, which can be a significant barrier for those with disabilities, those living abroad, or those unable to leave work or home. Secure Ballot enables voters to cast their votes from anywhere in the world securely, thus enhancing accessibility and potentially increasing voter turnout.

Despite its numerous advantages, the implementation of a blockchain-based voting system is not without challenges. Technical complexities, scalability issues, and the need for widespread digital literacy are significant hurdles that must be addressed. Additionally, legal and regulatory frameworks will need to adapt to accommodate and govern the use of blockchain in electoral processes. These challenges, however, are not insurmountable and provide opportunities for further research and innovation.

In conclusion, the Secure Ballot system presents a transformative approach to modernizing electoral processes. By harnessing the power of blockchain technology, it aims to create a secure, transparent, and efficient voting system that can enhance the integrity of elections and restore public trust in democratic institutions. The following sections of this paper will delve deeper into the technical architecture of the Secure Ballot system, explore the implementation challenges, and discuss the broader implications of adopting blockchain-based voting systems in contemporary society.

CHAPTER 2

LITRETURE SURVEY

[1]. BLOCKCHAIN BASED E-VOTING SYSTEM

AUTHORS: Kamal Desai, Disha Gosar, Rudresh Pachorkar - April 2023

OVERVIEW:

This research paper describes the design and implementation of a blockchain-based e-voting system. The proposed system uses a distributed ledger to record votes securely and transparently, making it difficult for anyone to manipulate or alter the results. The system safeguards voter anonymity and prevents double voting. The system uses Ethereum, a popular blockchain platform, and smart contracts to automate the voting process. Smart contracts handle vote counting and tallying, which eliminates the need for centralised authority. By requiring a majority of network nodes to agree on the validity of each transaction, a consensus algorithm ensures the system's security.
Advanced Cryptography is used to ensure data integrity and block chaining. Another distinguishing feature is the mode of network communication. To communicate between network nodes, a client method is used. There is no need for trust in this person because there is no third party to facilitate communication between clients.

[2]. Secure Voting Website Using Ethereum and Smart Contract

AUTHOR: Abhay Singh , Ankush Ganesh , Rutuja Rajendra Patil , Sumit Kumar , Ruchi Rani and Sanjeev Kumar Pippal [April 2023]

OVERVIEW:

The research focuses on developing a robust and user-friendly voting system by leveraging the advantages of decentralized technology. The proposed model employs Ethereum as the underlying blockchain platform through an innovative and iterative approach. Smart contracts are used to record and validate votes, while AI-based facial recognition technology is integrated to verify the identity of voters. Rigorous testing and analysis are conducted to validate the effectiveness and reliability of the proposed blockchain-based voting model.
The issue of maintaining a secure voting system is a prevalent problem in many countries. Current voting systems, such as the EVM in India, face significant problems, including decreased security and privacy concerns. The proposed model presents solutions to these problems through a blockchain-based voting system that utilizes smart contracts to ensure voter privacy, increase voting motivation, and enhance overall election security.
Smart Contract: Smart contracts are a simple logical bit of programs stored on each block of blockchain that run when some conditions are proper. They follow “if/when . . . then . . . ” statements written on the blockchain and help automate the verification of parties’ agreement.
Wallet: Wallet is a collection of user identities. It allows users to store and manage their Bitcoin, Ether, and other cryptocurrencies**.**

[3]. A Research Paper on E-Voting Using Blockchain Technology

AUTHORS: TANIKELLA SAI CHARAN, SRINANDA PENTAPATI, Mrs. R. PREMA - Mar 2022

OVERVIEW:

The system employs cryptographic methods to ensure voter anonymity and data integrity, and uses smart contracts to manage voter registration, vote casting, and result tallying automatically. Miners, including distributed network participants, election authorities, and academic institutions, validate and record votes on the blockchain, ensuring decentralization and tamper-proof records. Unique features include real-time vote tracking, immutable and transparent vote recording, and instant result publication. The project addresses significant challenges in traditional voting systems, such as susceptibility to fraud, inefficiency, and lack of transparency. By eliminating the need for a central authority, it enhances trust in the electoral process. The system promises a modernized, secure, and efficient voting experience, making it a significant advancement in the field of electronic voting.
The admin creates and deploys a voting system on a blockchain network (EVM), then sets up an election instance with necessary details. Voters register on the blockchain network, and their details are sent to the admin for verification. The admin checks and approves valid registrations, making the users eligible to vote. Approved voters cast their votes for their preferred candidates on the voting page.

[4]. Going from bad to worse: from Internet voting to blockchain voting

AUTHORS Sunoo Park ,Michael Specter, Neha Narula1 and Ronald L. Rivest [2020]

OVERVIEW:

This research paper critically examines the transition of voting technologies from traditional internet-based systems to the emerging realm of blockchain, focusing on the risks and challenges associated with integrating blockchain into voting systems. One of the primary concerns explored is security. Despite blockchain's decentralized structure and encryption, it introduces new vulnerabilities such as risks from smart contract bugs, potential network attacks, and the threat of a 51% attack, where a controlling faction could compromise the voting process. The paper delves into these vulnerabilities, along with broader security risks and ethical considerations inherent in blockchain-based voting solutions. It addresses how internet voting, initially lauded for its promise of enhanced accessibility and efficiency, has encountered substantial scrutiny due to issues like susceptibility to hacking, privacy infringements, and concerns about the integrity of election outcomes. The authors argue that while blockchain technology offers potential improvements in transparency and decentralization, its implementation in voting systems also presents significant technical and regulatory challenges. They emphasize the importance of understanding these complexities to inform ongoing discussions among policymakers, technologists, and the public regarding the future integration of blockchain in electoral processes.

[5]. Blockchain based voting system using Meta Mask and Ganache

AUTHOR: Hariram Srikanth, Department of Information Technology, St. Joseph's College of Engineering, Chennai, India, [2024]

OVERVIEW:

This project aims to develop a secure and transparent voting system using blockchain technology, Meta Mask for user authentication, and Ganache for local blockchain simulation. By leveraging Ethereum smart contracts, the system ensures that votes are recorded as immutable transactions on a decentralized ledger, guaranteeing tamper-proof record-keeping and providing unparalleled transparency in electoral processes. Meta Mask plays a crucial role as the user interface, facilitating seamless and anonymous voting experiences while maintaining robust security measures. Meanwhile, Ganache serves as a dependable testing environment, supporting the development and scalability of the voting system with reliable blockchain simulation capabilities. Key features include real-time result tabulation to expedite election outcomes, robust resistance mechanisms against fraud or manipulation, and strict adherence to electoral regulations to uphold legal standards and voter trust. The project actively tackles challenges such as optimizing user interface design for accessibility and usability, fortifying network security against potential cyber threats, and navigating complex regulatory landscapes to ensure compliance and acceptance. By demonstrating the tangible benefits of blockchain in enhancing the integrity, transparency, and efficiency of electoral processes, this initiative aims to set a precedent for secure digital voting systems on a global scale.

[6]. Blockchain-based electronic voting systems: A case study in Morocco

AUTHORS: Tarik Chafiq, Rida Azmi

OVERVIEW:

This research examines the feasibility of implementing blockchain-based electronic voting systems in Morocco to enhance electoral transparency and integrity. The study employs a methodology that combines Distributed Permission Ledger Technology (DPLT) and the Solana blockchain, resulting in a multilayered system. The main findings highlight the effectiveness of blockchain technology in mitigating electoral fraud and manipulation when implemented with precision, underscoring the importance of meticulous design and execution. These findings contribute significantly to discussions surrounding the modernization of electoral processes in the digital age and support the hypothesis that blockchain can address vulnerabilities in traditional voting methods. Moreover, the study marks a significant step toward modernizing elections, preserving democratic principles, and reinforcing the role of technology in addressing persistent electoral challenges, ultimately enhancing accessibility, security, and transparency in elections and strengthening democracy in the digital era.

The Solana blockchain assumes a pivotal role as the cornerstone of our proposed electronic voting system. Solana is renowned for enabling smart contracts, non-fungible tokens (NFTs), and decentralized applications and facilitating high-speed, secure, and scalable transactions .With the capacity to handle up to 50,000 transactions per second (TPS) and block times of 400 milliseconds, Solana offers the speed and scalability necessary for a nationwide voting system. It aims to demonstrate the feasibility of a blockchain embodying scalability, security, and decentralization. Under typical conditions, the system can sustain up to 710,000 TPS on a standard gigabit network and reach 28.4 million TPS on a 40-gigabit ethernet, positioning it as one of the world's preeminent decentralized computer networks. The second layer continuously maintains a hash of the most recent transaction, enhancing transparency, security, and system integrity.

[7]. B Vote: Block chain Based Voting System

AUTHORS: Shubham Chutke , Shreyas Parkar, Omkar Swargam [2024]

OVERVIEW:

This paper addresses the pressing challenges of traditional electoral systems, including vote rigging, electronic voting machine hacking, and election manipulation, which have eroded public trust. It proposes an electronic voting (e-voting) model leveraging blockchain technology to mitigate these issues. The model utilizes popular blockchain frameworks that offer blockchain as a service, ensuring participant confidentiality while allowing for public scrutiny. The paper conducts a thorough analysis, highlighting blockchain's capacity to enhance security and transparency in e-voting systems. Distributed ledger technology, known for its cryptographic foundations and transparency, is identified as pivotal in revolutionizing various sectors, including electoral processes. The proposed e-voting system emphasizes the advantages of blockchain's immutability and consensus mechanisms, supported by technologies such as Meta Mask, Solidity, Ethereum, and decentralized applications (Dapps) deployed on testnets like Hardhat. By ensuring data integrity and decentralization, the system aims to bolster the resilience of e-voting systems and uphold the integrity of election results. Overall, this paper contributes valuable insights into blockchain technology's potential to address electoral challenges, promising improved transparency, security, and reliability in modern e-voting systems.

[8]. E-voting systems using blockchain: a systematic review and future research direction

AUTHORS: Dhiraj Amrutkar , Gaurav Dongare [2021]

OVERVIEW:

Blockchain technology represents a groundbreaking advancement in the professional world, characterized by its decentralized, digitized, and consensus-driven approach to secure information storage. Its pervasive influence continues to drive revolutionary changes across various industries. In recent years, the exponential growth of blockchain has prompted scholars and experts to explore its diverse applications, including its potential in e-voting systems. This article presents a systematic review of emerging blockchain-based e-voting systems, highlighting their strengths and identifying critical research gaps. The review underscores the need for enhanced frameworks tailored specifically for voting systems, addressing concerns such as scalability, security, and regulatory compliance. Key technologies like Ethereum and cloud computing play pivotal roles in these advancements, facilitating secure and transparent ballot processes. The findings emphasize ongoing challenges and opportunities in leveraging blockchain for e-voting, urging further research to refine and optimize these systems for widespread adoption. Overall, this paper contributes to the evolving discourse on blockchain's transformative impact on electoral processes, advocating for continued innovation and development in this dynamic field.

[9]. E-voting systems using blockchain: a systematic review and future research direct

Authors: Uzma Jafar, Mohd Juzaiddin Ab Aziz [2021]

OVERVIEW:

Online voting is a trend that is gaining momentum in modern society. It has great potential to decrease organizational costs and increase voter turnout. It eliminates the need to print ballot papers or open polling stations—voters can vote from wherever there is an Internet connection. Despite these benefits, online voting solutions are viewed with a great deal of caution because they introduce new threats. A single vulnerability can lead to large-scale manipulations of votes. Electronic voting systems must be legitimate, accurate, safe, and convenient when used for elections. Nonetheless, adoption may be limited by potential problems associated with electronic voting systems. Blockchain technology came into the ground to overcome these issues and offers decentralized nodes for electronic voting and is used to produce electronic voting systems mainly because of their end-to-end verification advantages. This technology is a beautiful replacement for traditional electronic voting solutions with distributed, non-repudiation, and security protection characteristics. The following article gives an overview of electronic voting systems based on blockchain technology. The main goal of this analysis was to examine the current status of blockchain-based voting research and online voting systems and any related difficulties to predict future developments. This study provides a conceptual description of the intended blockchain-based electronic voting application and an introduction to the fundamental structure and characteristics of the blockchain in connection to electronic voting. As a consequence of this study, it was discovered that blockchain systems may help solve some of the issues that now plague election systems. On the other hand, the most often mentioned issues in blockchain applications are privacy protection and transaction speed. For a sustainable blockchain-based electronic voting system, the security of remote participation must be viable, and for scalability, transaction speed must be addressed.

[10]. Cloud Based Voting System Using Blockchain Technology

Authors: Jai Kaushik, Harshul Dudeja [2024]

OVERVIEW:

This study explores the potential of a cloud-based voting system leveraging blockchain technology to ensure the legitimacy and security of election processes. Using a cloud-based platform, voters can cast ballots securely stored on a blockchain network, with each block containing a cryptographic hash of the previous block and a timestamp, ensuring the chain's resilience and tamper-resistance. Cloud computing plays a crucial role by providing the necessary infrastructure to support the scalability and accessibility of the voting system. It allows for real-time data processing, storage, and retrieval, ensuring that voters can cast their ballots from anywhere with internet access. The research highlights the significant advantages of blockchain, such as decentralization, transparency, and immutability, which enhance the security and reliability of electronic voting systems. However, it also addresses the challenges and threats associated with blockchain-based voting systems, including scalability issues, potential reliance on dubious mechanisms, and the need for resistance to coercion. The study underscores the necessity for substantial and in-depth investigations to overcome these challenges and ensure the **security, reliability, and integrity of the voting process. As technology advances and the world rapidly moves toward cloud-based digitization, the need for robust cloud security is becoming increasingly critical. This research demonstrates the immense potential of blockchain technology for online voting,** suggesting that a secure and transparent voting system can be created by utilizing blockchain's advantages, thereby fostering faith and confidence in the democratic **process.**

[11]. online voting systems using blockchain

Authors: Saurab Chauhan, Dev Rabadia, Sohang Patel, Bhavesh Suthar [2024]

OVERVIEW:

In an era marked by technological advancements and a growing demand for secure and transparent electoral processes, the integration of blockchain technology into online voting systems has emerged as a promising solution. This research paper presents a comprehensive exploration of the design, implementation, and implications of an online voting system built upon blockchain technology. Through an in-depth analysis of existing electronic voting challenges and the potential of blockchain, this paper demonstrates how the decentralized, immutable, and transparent nature of blockchain addresses critical concerns such as security, voter privacy, and trust in electoral outcomes. The paper delves into the core architecture of the proposed system, highlighting the role of smart contracts in automating voting processes while ensuring authenticity and verifiability. Security and transparency are examined in detail, showcasing the cryptographic measures that safeguard voter information and prevent fraudulent activities. The challenges of voter authentication, scalability, and accessibility are discussed, along with potential solutions to overcome these obstacles. Drawing on case studies of real-world implementations, the paper offers insights into the successes, challenges, and lessons learned from adopting blockchain-based online voting systems. Legal and ethical considerations are also explored, emphasizing the need for aligning technological innovations with legal frameworks and ethical standards. Finally, the research paper contemplates the future of blockchain-powered online voting, envisioning how emerging technologies such as biometrics, artificial intelligence, and zero-knowledge proofs could further enhance the security and inclusivity of electoral processes. Overall, this paper underscores the transformative potential of blockchain in revolutionizing online voting, fostering a more resilient and democratic electoral landscape.

[12]. Blockchain based voting system for Jordan parliament elections

Authors: Mohammad Malkawi , Muneer Bani Yassein, Asmaa Bataineh [2020]

OVERVIEW:

Covid-19 pandemic has stressed more than any-time before the necessity for conducting election processes in an electronic manner, where voters can cast their votes remotely with complete security, privacy, and trust. The different voting schema in different countries makes it very difficult to utilize a one fits all system. This paper presents a blockchain based voting system (BBVS) applied to the Parliamentary elections system in the country of Jordan. The proposed system is a private and centralized blockchain implemented in a simulated environment. The proposed BBVS system implements a hierarchical voting process, where a voter casts votes at two levels, one for a group, and the second for distinct members within the group. This paper provides a novel blockchain based e-Voting system, which proves to be transparent and yet secure. This paper utilizes synthetic voter benchmarks to measure the performance, accuracy and integrity of the election process. This research introduced and implemented new algorithms and methods to maintain acceptable performance both at the time of creating the blockchain(s) for voters and candidates as well as at the time of casting votes by voters.

CHAPTER 3

PROBLEM STATEMENT

Elections globally confront Challenges with voting integrity,transparency and accessibility. Current System (paper-based or electronic) are susceptible to tampering ,Casting doubts on accuracy. Transparency gap prevent voters from verifying election results.Centralized voter database raise privacy concerns due to security vulnerabilities. Integration of the new technology like Blockchain rise regulatory issue for legal compliance and public trust.

OBJECTIVES

- **Add a new Candidate:** This function helps to add new candidate to the election,which can be done only by the admin before the election starts.
- **Add a new Voter:** This function helps to add a voter,which can be done only once by the admin before the election
- **Display the Candidate details:**This function helps to show candidate details
- **Display the results of the Election:** This function helps to display the result of the winner of Election
- **Start and End Election:** This function helps to start and end the election only by admin

CHAPTER 4

System Requirement Specification

Functional Requirements:

Voter Registration: The system will provide a secure and user-friendly interface for voter registration. Voters will be able to register using government-issued identification, and their details will be verified against existing databases to ensure eligibility.

Voter Authentication: The system will authenticate voters based on their age. The voters below 18 years are not eligible to vote

Ballot Creation and Distribution: The system will allow election administrators to create and distribute digital ballots to registered voters. Each ballot will be uniquely associated with a specific voter to prevent duplication.

Vote Casting: Voters will be able to cast their votes securely and anonymously. The system will provide a straightforward interface for voters to select their preferred candidates or options.

Vote Encryption: Each vote will be encrypted using advanced cryptographic techniques before being recorded on the blockchain to ensure that the content of the vote remains confidential.
Blockchain Ledger: The system will record each vote on a decentralized blockchain ledger. This ledger will be immutable and publicly verifiable to ensure transparency and prevent tampering.

Smart Contracts: Smart contracts will be used to automate the election process, including voter eligibility checks, vote validation, and result tallying. These contracts will enforce election rules and protocols without manual intervention.

Real-Time Auditing and Monitoring: The system will provide real-time auditing and monitoring capabilities. Election stakeholders will be able to verify the integrity of the voting process and the votes recorded on the blockchain.

Result Tallying and Reporting: The system will automatically tally votes once the voting period ends. It will generate comprehensive reports on the election results, including total votes for each candidate or option.

Non-Functional Requirements

Security: The system will employ robust security measures to protect against unauthorized access, data breaches, and cyber-attacks. This includes encryption of data at rest and in transit, secure coding practices, and regular security audits.

Scalability: The system will be scalable to accommodate a large number of voters and transactions. It will handle high volumes of concurrent users without performance degradation, especially during peak voting times.

Performance: The system will provide fast response times for all operations, including voter registration, authentication, vote casting, and result tallying. The performance metrics will be within acceptable limits to ensure a smooth user experience.

Reliability and Availability: The system will be highly reliable and available, with minimal downtime. It will have a robust disaster recovery plan and failover mechanisms to ensure continuous operation during elections.

Usability: The system will be user-friendly and accessible to voters of all technical skill levels. The interface will be intuitive, and instructions will be clear and easy to follow. Accessibility features will be included for voters with disabilities.

Compliance: The system will comply with all relevant legal and regulatory requirements related to elections and data protection. This includes adherence to local, national, and international laws governing electoral processes and privacy.

Interoperability: The system will be interoperable with existing government databases and systems for voter registration and verification. It will also support integration with other systems, such as national ID databases and biometric systems.

Maintainability: The system will be designed for ease of maintenance, with modular components that can be updated or replaced without significant disruption. Documentation will be comprehensive to assist in troubleshooting and updates.

Transparency: The system will ensure transparency in the voting process. All actions and transactions will be traceable and verifiable by authorized stakeholders, enhancing trust in the electoral process.

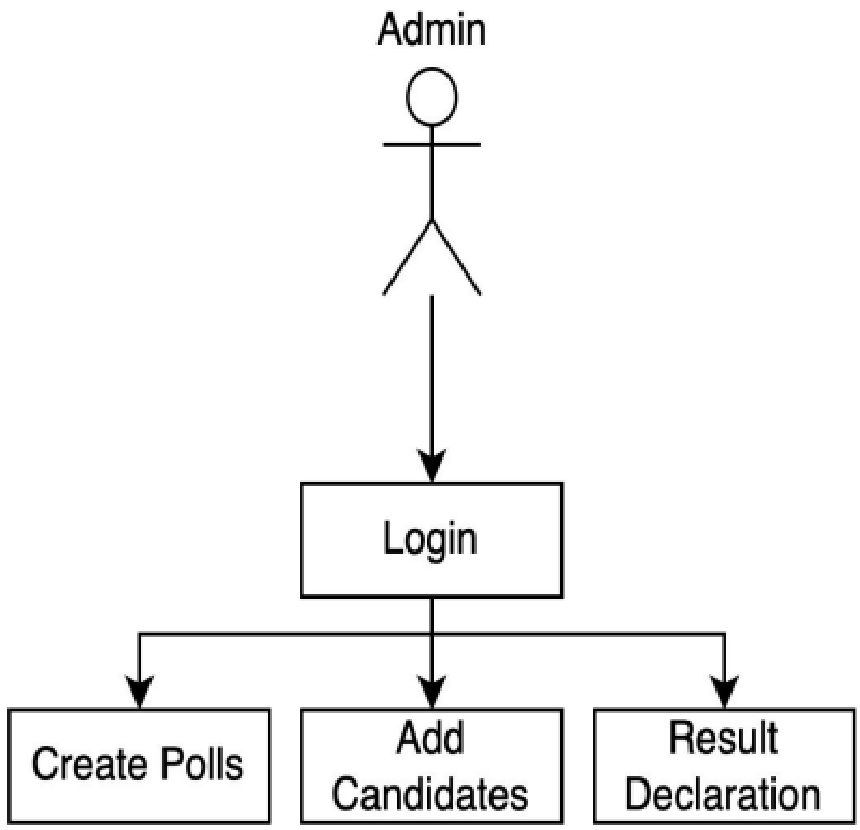
Auditability: The system will support comprehensive auditing capabilities. It will maintain detailed logs of all activities and transactions, which can be reviewed by auditors to ensure the integrity of the election process.

Data Privacy: The system will protect voter data privacy, ensuring that personal information is securely stored and processed in accordance with data protection laws. Only authorized personnel will have access to sensitive voter information.

CHAPTER 4

SYSTEM DESIGN

❖ Admin module



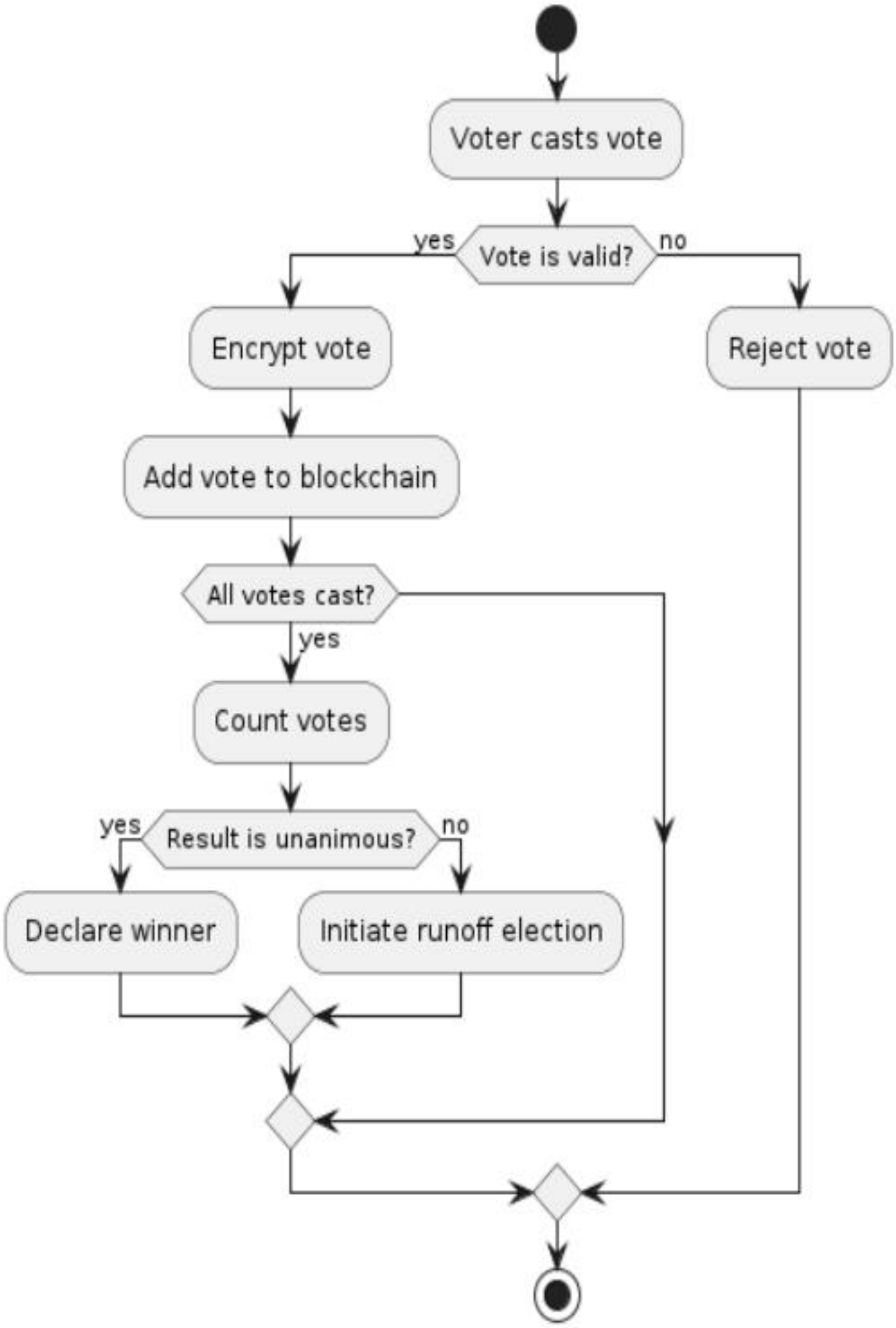
Home: the entire election process from start to finish, including setting dates and monitoring progress. Includes options to start, end, and reset elections.

Candidates: Add, edit, and manage the list of candidates participating in the election. Ensure all candidate information is accurate and up-to-date.

Voter List: Manage voter registration and approval process. After admin approval, registered voters are added to the official voter list.

Result: View and publish the final election results. This tab displays detailed vote counts and winning candidates.

❖ flowchart of the proposed blockchain voting system

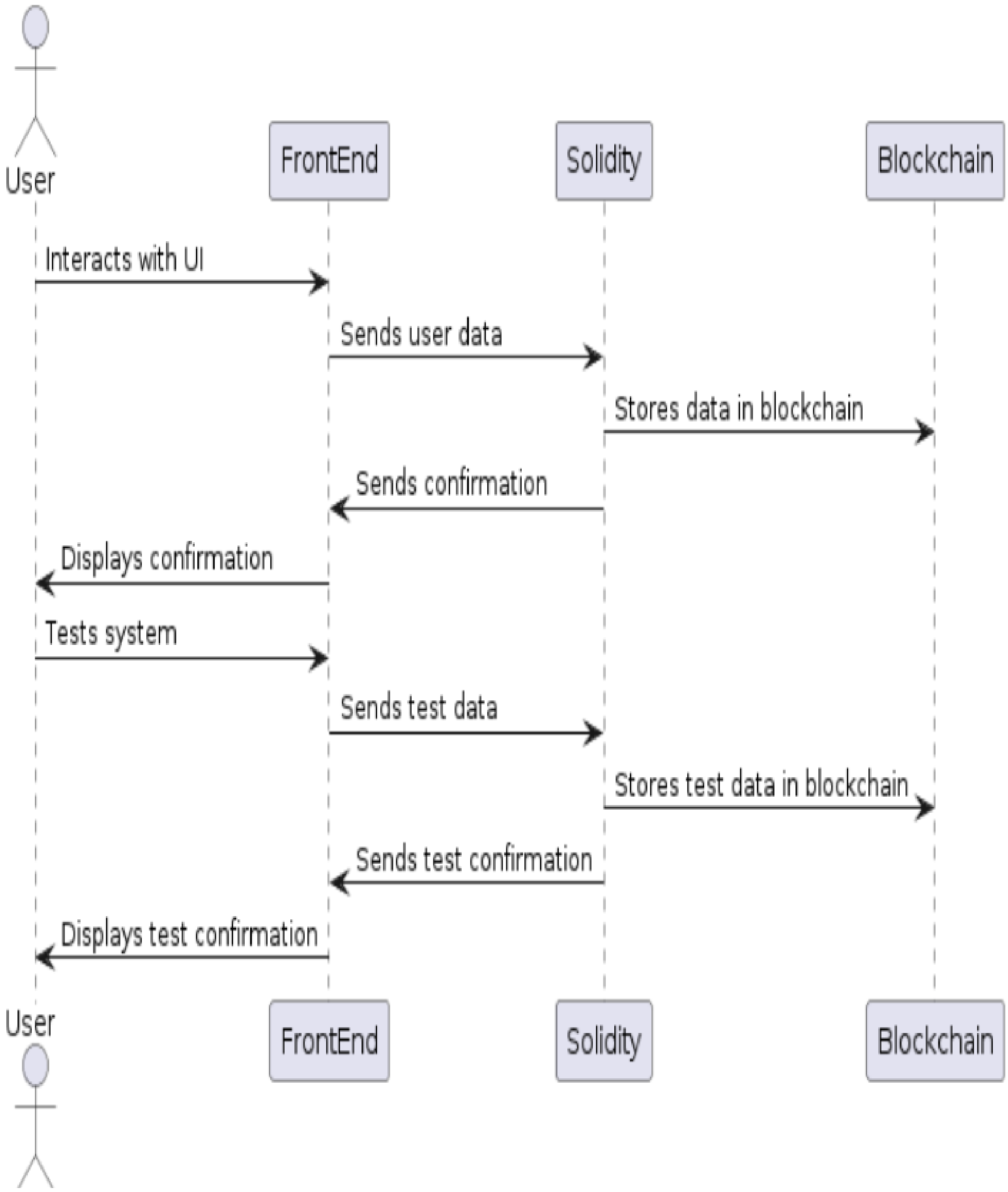


It explains the process flow of every step involved via graphical representation. The voter's identity is verified through a unique identifier provided during registration. During the voting process, the voter submits their vote through an interface linked to the blockchain network. The vote is then verified through smart contracts and stored on the blockchain in a secure and immutable manner. The voter can also track their vote through the blockchain to ensure accuracy and transparency. Vote casting involves the voter accessing the voting application and selecting their preferred candidate. Once the vote is cast, it is recorded on the blockchain and cannot be altered. Subsequent vote counting involves tallying the votes recorded on the blockchain to determine the election results. The ballots can be counted manually or using an automated system, depending on the size of the election and the resources available. Depending on the development of the previous step, the winner is declared, or the system initiates a runoff election.

A sequence diagram is a tool to visualize the interactions between different objects in a system over time. It begins with the voter registering for the election, and their information is stored securely on the Firebase Cloud. When voters attempt to vote, their identity is verified through the blockchain using the stored registration data. Once the voter selects a candidate, the vote is added as a transaction on the blockchain. The transaction includes information about the voter, the candidate they voted for, and the voting time. A sequence diagram helps demonstrate a blockchain voting system's interactions and processes. It can help to identify potential bottlenecks or areas for improvement while also showcasing the transparency and security features of the system.

The data flow in a blockchain voting system starts with the voter registration process, where the voters. The vote is encrypted for security and privacy. The blockchain then verifies the vote's validity and adds it to the ledger. The results are then tallied and published on a public website for transparency. This ensures the voting process's security and accuracy while allowing easy access to information and results. By utilizing these technologies, the voting process becomes more efficient and transparent, which helps to increase trust in the system.

- **Votig System Sequence Diagram**



Users interact with the election system through a web-based front end, which provides an intuitive interface for managing and participating in the election process. The front end is designed to be user-friendly, allowing administrators to effortlessly set election dates, add candidates, manage voter registrations, and view results.

When an administrator starts an election, the front end sends transaction requests to the Solidity smart contracts deployed on the blockchain. These contracts handle all the critical functions of the election process, ensuring that all actions are recorded immutably and transparently. For instance, setting the election parameters, such as start and end dates, is securely stored on the blockchain, preventing any tampering or unauthorized changes.

Candidates can be added through the front end, which communicates with the Solidity contracts to store candidate information on the blockchain. This ensures that candidate details are immutable and publicly verifiable. Administrators can update or manage candidate information through similar interactions with the blockchain.

The voter registration process is also managed through the front end. When a voter registers, their information is sent to the blockchain, where the Solidity contracts handle the verification and approval process. Once approved by the administrator, the voter is added to the official voter list stored on the blockchain. This list can be accessed and managed through the front end, ensuring that only authorized voters can participate.

During the election, voters cast their votes via the front end, which securely submits their choices to the blockchain. The Solidity contracts tally the votes in real-time, ensuring that each vote is counted accurately and securely. The results can be viewed on the front end, where they are directly fetched from the blockchain, providing a transparent and tamper-proof display of the election outcome.

In case the election needs to be reset, the administrator can use the front end to initiate a reset process. This action triggers the Solidity contracts to clear all election-related data on the blockchain, ensuring that the system is ready for a new election cycle. This process guarantees a secure and clean slate for future elections, maintaining the integrity and trustworthiness of the system.

The integration of the front end with Solidity blockchain technology not only enhances security and transparency but also simplifies the user experience. By leveraging the immutable nature of blockchain, all election data, from candidate details to voter registrations and final results, is securely recorded and easily accessible. This eliminates the risk of data manipulation and ensures that all stakeholders, including administrators, candidates, and voters, have confidence in the election's integrity. Moreover, the front end's intuitive design ensures that users, regardless of their technical expertise, can interact with the system seamlessly. The combination of a user-friendly interface and robust blockchain backend creates a reliable and efficient election management system that can be trusted to deliver accurate and fair outcomes.

Software Requirements

1. **Ethereum Blockchain Platform:**

Description: Ethereum is a decentralized platform that enables the creation and execution of smart contracts and dApps (decentralized applications). It utilizes blockchain technology to provide a secure and transparent environment for transactions and data storage.

Version: Latest stable release recommended.

Purpose: Ethereum will serve as the underlying blockchain platform for recording and validating votes securely using smart contracts.

2. MetaMask:

Description: MetaMask is a cryptocurrency wallet and browser extension that allows users to interact with the Ethereum blockchain. It provides a secure way to manage digital assets and interact with decentralized applications directly from the web browser.

Version: Latest version compatible with the chosen Ethereum network.

Purpose: MetaMask will enable voters to securely authenticate and interact with the voting dApp through their web browsers, ensuring a user-friendly voting experience.

3. React.js:

Description: React.js is a JavaScript library for building user interfaces, particularly single-page applications. It allows developers to create reusable UI components and manage application state efficiently.

Version: Latest stable release (e.g., React 17.x).

Purpose: React.js will be used for frontend development, providing a responsive and interactive user interface for the voting system. It facilitates modular development and seamless integration with other JavaScript libraries.

4. Ganache:

Description: Ganache is a personal blockchain for Ethereum development purposes. It provides a local Ethereum blockchain environment that developers can use for testing smart contracts, deploying dApps, and running automated tests.

Version: Latest stable release.

Purpose: Ganache will serve as the development blockchain environment, allowing developers to deploy and test smart contracts locally before deploying them to the main Ethereum network.

5. Truffle Suite:

Description: Truffle Suite is a development framework for Ethereum dApps. It includes tools like Truffle (for smart contract compilation, deployment, and testing), Drizzle (for frontend development with React), and Ganache CLI (for command-line blockchain management).

Version: Latest stable release.

Purpose: Truffle Suite provides a comprehensive set of tools for developing, testing, and deploying the voting system smart contracts and frontend components efficiently.

6. Solidity:

Description: Solidity is a high-level, statically-typed programming language designed for writing smart contracts on the Ethereum blockchain. It is used to define the logic and behavior of smart contracts, including voting rules and data handling.

Version: Latest stable release.

Purpose: Solidity will be used for smart contract development, defining the rules and procedures for voter registration, ballot creation, vote casting, and tallying within the voting system.

7. Bootstrap

Description: Bootstrap is a popular front-end framework for developing responsive and mobile-first websites and web applications. It provides pre-designed CSS and JavaScript components that help streamline UI design and development.

Version: Latest stable release (e.g., Bootstrap 5.x).

Purpose: Bootstrap will be used for designing the user interface of the voting system, ensuring that the voting dApp is visually appealing, responsive across different devices, and easy to navigate for voters.

CHAPTER 6

System Analysis

System Analysis: Existing System

The existing electoral systems in many countries rely heavily on traditional paper-based or electronic voting methods. These systems often face significant challenges related to security, transparency, and efficiency, leading to concerns about the integrity of election outcomes. Paper ballots are susceptible to tampering, loss, and errors during counting, while electronic voting systems may suffer from vulnerabilities in software or hardware that could be exploited by malicious actors.

Security Concerns: Paper-based voting systems are vulnerable to physical tampering, such as ballot stuffing or misplacement of ballots. Even with secure storage and transport measures, ensuring the integrity of each vote cast remains a challenge. Electronic voting systems, on the other hand, face cybersecurity risks, including hacking attempts, malware infections, and denial-of-service attacks. These vulnerabilities can compromise the confidentiality and accuracy of votes.

Transparency and Trust: Transparency in the electoral process is crucial for fostering public trust. However, both paper-based and electronic voting systems often lack mechanisms for real-time auditing and verification of results. Paper ballots require extensive manual counting and auditing processes, which can be time-consuming and error-prone. Electronic voting systems may not provide sufficient transparency due to proprietary software or lack of access to source code.

Accessibility and Voter Participation: While electronic voting systems aim to improve accessibility for voters, including those with disabilities or those residing abroad, implementation challenges and concerns about system reliability often limit their effectiveness. Paper-based systems may also pose accessibility challenges for voters who cannot physically access polling stations.

Administration and Cost: Managing and administering elections using traditional methods involve significant logistical challenges and costs. This includes printing and distributing ballots, training election officials, securing polling stations, and transporting ballot boxes. Electronic voting systems, while potentially reducing some administrative burdens, may require substantial upfront investments in technology infrastructure and ongoing maintenance costs.

System Analysis: Proposed System

The proposed Secure Ballot system aims to address the limitations of existing electoral systems by leveraging blockchain technology to enhance security, transparency, and efficiency in the voting process.

Enhanced Security: Utilizing blockchain ensures that each vote is securely encrypted and recorded on a decentralized ledger. The cryptographic nature of blockchain technology makes it virtually impossible to alter or manipulate votes without detection. By distributing data across a network of nodes, the Secure Ballot system mitigates the risk of single points of failure and unauthorized access.

Transparency and Verifiability: Blockchain's transparent and immutable nature allows for real-time auditing and verification of votes by stakeholders, including voters, election officials, and auditors. Each transaction (vote) recorded on the blockchain can be traced back to its origin, ensuring that votes are counted accurately and transparently. Smart contracts enforce predefined election rules, automating processes such as voter eligibility verification and vote tallying.

Accessibility and Convenience: The Secure Ballot system facilitates remote and secure voting, enabling eligible voters to cast their ballots from anywhere with an internet connection. This enhances accessibility for voters with disabilities, those living abroad, or individuals unable to physically attend polling stations. MetaMask integration provides a user-friendly interface for interacting with the blockchain securely.

Cost Efficiency: By reducing reliance on paper-based materials and streamlining administrative processes through automation, the Secure Ballot system offers potential cost savings compared to traditional voting methods. While initial setup costs may include developing the voting dApp and integrating with Ethereum and MetaMask, long-term operational costs can be minimized through blockchain's decentralized nature and automated processes.

User Experience: Leveraging React.js and Bootstrap, the Secure Ballot system offers a modern and intuitive user interface. Voters can easily navigate the voting process, verify their identity securely using MetaMask, and cast their votes with confidence. Real-time updates and notifications ensure voters are informed throughout the election period, enhancing overall user experience and satisfaction.

Legal and Regulatory Compliance: The Secure Ballot system will comply with relevant legal and regulatory frameworks governing elections and data protection. This includes ensuring voter anonymity while maintaining auditability and transparency of the voting process. Compliance with international standards for electoral integrity will be a priority during system design and implementation.

High level design:

1. Introduction

A blockchain-based e-voting system leverages the immutable and decentralized nature of blockchain technology to ensure a secure, transparent, and tamper-proof voting process. The system allows voters to cast their votes electronically, ensuring that their votes are recorded accurately and can be verified independently by any participant in the blockchain network.

2. Components of the E-Voting System

User Interface (UI):

Voter Interface: Allows voters to register, verify their identity, view candidates, and cast their votes.
Admin Interface: Allows administrators to manage elections, verify voters, add candidates, and monitor the election process.

Smart Contracts:

Defines the rules of the election, including voter registration, candidate management, vote casting, and vote tallying.
Ensures that all operations are executed as per the defined rules and are recorded on the blockchain.
Blockchain Network:
A decentralized network where all transactions (votes) are recorded. Each node in the network holds a copy of the blockchain, ensuring data integrity and transparency.

Web3.js:

A JavaScript library that enables interaction with the Ethereum blockchain. It allows the frontend application to communicate with the smart contracts deployed on the blockchain.

Ethereum Virtual Machine (EVM):

Executes the smart contracts and ensures that the transactions are processed correctly.

Cryptographic Algorithms:

Used to secure the identities of voters and ensure that the votes are anonymous and tamper-proof.

Voting Application Backend:

Handles communication between the frontend application and the blockchain network, ensuring that data is properly formatted and transactions are executed correctly.

3. Workflow of the E-Voting System

Voter Registration:

Voters register on the platform by providing their name and Aadhar number.
The admin verifies the voter's identity and updates the smart contract to reflect the voter's registered status.

Election Setup:

The admin creates an election by deploying a smart contract that includes the list of candidates and the rules for the election. The smart contract is deployed on the Ethereum blockchain.

Vote Casting:
Registered voters log into the system and cast their votes by selecting a candidate.
The vote is sent to the smart contract, which records the vote on the blockchain.

Vote Tallying:
Once the election period ends, the admin triggers the vote tallying process.
The smart contract counts the votes and publishes the results on the blockchain.

Result Announcement:
The results are displayed on the frontend application, ensuring that the process is transparent and verifiable by any participant in the blockchain network.

4. Security Considerations

Identity Verification:
Ensures that only eligible voters can participate in the election by verifying their identity using Aadhar numbers.

Anonymity:
Ensures that votes are anonymous and cannot be traced back to the voter.

Data Integrity:
Ensures that votes cannot be tampered with once they are recorded on the blockchain.

Transparency:
Ensures that the entire voting process is transparent and verifiable by any participant in the blockchain network.

Detailed Steps for Voter Registration
- Voter enters their name and Aadhar number in the registration form.
- The registration form sends the data to the registerVoter function in the DistributedVoting smart contract.
- The registerVoter function checks if the voter is already registered.
- If not, it adds the voter's details to the voterList and sets their registered status to false.
- The admin verifies the voter and updates their registered status to true.

Detailed Steps for Vote Casting
- The voter selects a candidate and submits their vote.
- The vote is sent to the castVote function in the DistributedVoting smart contract.
- The castVote function checks if the voter is registered and hasn't voted yet.
- If the conditions are met, it records the vote and updates the voter's status to voted.
- The vote is recorded on the blockchain, ensuring it cannot be tampered with.

Detailed Steps for Vote Tallying

- The admin triggers the vote tallying process by calling the endElection function in the DistributedVoting smart contract.
- The endElection function changes the state of the election to ended.
- The smart contract counts the votes for each candidate and stores the results on the blockchain.
- The results are published on the frontend application, ensuring transparency and verifiability.

Security Measures

Identity Verification:
Ensures that only eligible voters can register and vote by verifying their identity using Aadhar numbers.
Anonymity:
Ensures that votes are anonymous and cannot be traced back to the voter, protecting voter privacy.
Data Integrity:
Ensures that once a vote is recorded on the blockchain, it cannot be altered or deleted, maintaining data integrity.
Transparency:
Ensures that the entire voting process is transparent and verifiable by any participant in the blockchain network, building trust in the system.

System Implementation for Blockchain-Based E-Voting System

1. Introduction

This section describes the system implementation of a blockchain-based e-voting system. The implementation details include the algorithms used, the modules implemented, and their respective pseudo code. The system aims to ensure secure, transparent, and tamper-proof electronic voting.

2. Algorithms Used

Voter Registration Algorithm:
Ensures only eligible voters are registered and recorded on the blockchain.

Vote Casting Algorithm:
Allows registered voters to cast their votes securely.

Vote Tallying Algorithm:
Counts the votes and publishes the results in a transparent manner.

3. Modules Implemented

User Interface (UI) Module:
Voter Registration
Vote Casting
Viewing Election Results
Smart Contract Module:
Voter Management
Candidate Management
Vote Casting and Tallying
Blockchain Interaction Module:
Connecting to Ethereum Blockchain
Deploying and Interacting with Smart Contracts

4. Pseudo Code for Modules

4.1 User Interface (UI) Module

Voter Registration Interface:

pseudo
Copy code

```
function VoterRegistrationInterface()
    display "Enter Name:"
    input name
    display "Enter Aadhar Number:"
    input aadhar

    call registerVoter(name, aadhar)
    if registration successful
```

```
    display "Registration Successful"
  else
    display "Registration Failed"
end function
```

description:

The `VoterRegistrationInterface` function is designed to manage the voter registration process in an election system. When invoked, it first prompts the user to enter their name and captures this input in the variable `name`. It then asks the user to input their Aadhar number, storing this information in the variable `aadhar`. With both pieces of information collected, the function calls the `registerVoter(name,adhar,age)` function to attempt to register the voter using the provided details. If the registration is successful, the function displays a message saying "Registration Successful"; otherwise, it displays "Registration Failed". This function ensures a seamless and interactive experience for users, guiding them through the registration process and providing immediate feedback on their registration status.

Vote Casting Interface:

pseudo
Copy code

```
function VoteCastingInterface()
  display list of candidates

  display "Select Candidate:"
  input candidate

  call castVote(candidate)
  if vote successful
    display "Vote Cast Successfully"
  else
    display "Vote Casting Failed"
end function
```

Viewing Election Results Interface:

Description:

The Vote Casting Interface guides voters through the election process by first displaying a list of candidates, allowing them to select their preferred candidate. Upon selection, the interface calls the `castVote` function to record the voter's choice. If successful, it confirms with "Vote Cast Successfully"; otherwise, it notifies "Vote Casting Failed". Meanwhile, the **Viewing Election Results Interface** enables stakeholders to access and review finalized election outcomes. It presents comprehensive results, including vote counts and percentages, ensuring transparency and accountability in the electoral process. These interfaces leverage blockchain technology to secure and validate votes, providing a robust framework for democratic participation and result verification.

pseudo
Copy code

```
function ViewResultsInterface()
  call getResults()
```

```
    display results
end function
```

Description:

The **ViewResultsInterface** function retrieves election results by calling `getResults()` and displays them on the interface. This function provides stakeholders with immediate access to comprehensive election outcomes, including detailed vote counts and any relevant statistics. Utilizing blockchain technology ensures the integrity and transparency of the results, fostering trust and accountability in the electoral process.

4.2 Smart Contract Module

Smart Contract for Voter Management:

```
pseudo
Copy code
contract DistributedVoting
    struct Voter
        address voterAddress
        string voterName
        string voterAadhar
        bool voterExists
        bool registered
        bool voted
    mapping(address => Voter) voters
    Voter[] voterList

    function registerVoter(name, aadhar)
        if voters[msg.sender].voterExists == false
            Voter newVoter = Voter(msg.sender, name, aadhar, true, false, false)
            voters[msg.sender] = newVoter
            voterList.push(newVoter)
        end if
    end function

    function verifyVoter(voterAddress)
        if onlyAdmin and voters[voterAddress].voterExists == true
            voters[voterAddress].registered = true
        end if
    end function

    function getAllVoters() returns (Voter[] memory)
        return voterList
    end function
end contract
```

Description:

The DistributedVoting smart contract manages voter registration and verification in an election system. It defines a `Voter` structure storing essential details like address, name, Aadhar number, and status flags for existence, registration, and voting eligibility. The contract includes functions for registering new voters, verifying voter eligibility by administrators, and retrieving a list of all registered voters. Utilizing mappings and arrays, it ensures efficient and secure management of voter data on the blockchain, enhancing transparency and integrity throughout the electoral process.

Smart Contract for Candidate Management:

```
pseudo
Copy code
contract DistributedVoting
    struct Candidate
        string candidateName
        bool exists
        int voteCount

    mapping(string => Candidate) candidates
    Candidate[] candidateList

    function addCandidate(name)
        if candidates[name].exists == false
            Candidate newCandidate = Candidate(name, true, 0)
            candidates[name] = newCandidate
            candidateList.push(newCandidate)
        end if
    end function

    function getAllCandidates() returns (Candidate[] memory)
        return candidateList
    end function
end contract
```

Description:

The DistributedVoting smart contract manages candidates in an election system through a defined `Candidate` structure, containing attributes such as the candidate's name, existence status, and vote count. Using a mapping named `candidates` and an array `candidateList`, it provides methods for adding new candidates and retrieving a list of all registered candidates. The `addCandidate` function checks if a candidate with the given name already exists; if not, it creates a new `Candidate` instance with initial values, marks them as existing, and appends them to `candidateList`. This ensures each candidate is uniquely identified and tracked within the contract. The `getAllCandidates` function retrieves the entire list of candidates stored in `candidateList`, facilitating transparency and accessibility of candidate information throughout the election process on the blockchain.

Smart Contract for Vote Casting and Tallying:

```
pseudo
Copy code
contract DistributedVoting
    enum State { NotStarted, Running, Ended }
    State state

    function castVote(candidateName)
        if voters[msg.sender].registered == true and voters[msg.sender].voted == false and state ==
State.Running
            candidates[candidateName].voteCount += 1
            voters[msg.sender].voted = true
```

```
        end if
    end function

    function startElection()
        if onlyAdmin and state == State.NotStarted
            state = State.Running
        end if
    end function

    function endElection()
        if onlyAdmin and state == State.Running
            state = State.Ended
        end if
    end function

    function getResults() returns (Candidate[] memory)
        return candidateList
    end function
end contract
```

4.3 Blockchain Interaction Module

Connecting to Ethereum Blockchain:

pseudo

Copy code

```
function connectToBlockchain()
    web3 = new Web3(window.ethereum)
    networkId = web3.eth.net.getId()
    contractData = DistributedVoting.networks[networkId]

    if contractData exists
        contract = new web3.eth.Contract(DistributedVoting.abi, contractData.address)
    else
        display "Contract not deployed"
    end if
end function
```

Description:

The DistributedVoting smart contract manages the entire voting process and election lifecycle on the blockchain. It includes functionalities for casting votes, starting and ending elections, and retrieving election results. The contract defines an `enum State` to track the current phase of the election, allowing operations only when the election is running and voters are registered and have not yet voted. Methods like `castVote` increment the vote count for specified candidates, mark voters as having cast their vote, and enforce election state constraints. For administrative control, `startElection` and `endElection` functions transition the election state between "NotStarted", "Running", and "Ended" phases, ensuring proper election management. The `getResults` function retrieves the current vote tallies from `candidateList`, providing transparent and verifiable election outcomes stored securely on the blockchain.

Interacting with Smart Contracts:

```
pseudo
Copy code
function registerVoter(name, aadhar)
   contract.methods.registerVoter(name, aadhar).send({ from: userAccount })
end function

function castVote(candidateName)
   contract.methods.castVote(candidateName).send({ from: userAccount })
end function

function getResults()
   contract.methods.getResults().call().then(results => display results)
end function
```

Description:

These functions facilitate interaction with a deployed smart contract on a blockchain using Web3.js. The `registerVoter` function allows users to register for voting by calling `registerVoter(name, aadhar)` and sending a transaction from `userAccount`. Similarly, `castVote(candidateName)` enables voters to cast their votes, while `getResults()` retrieves and displays election outcomes. These interactions ensure transparency and integrity in managing voter registration, vote casting, and result retrieval, leveraging blockchain's decentralized and immutable nature to maintain the integrity of electoral processes.

5. Detailed Steps for Each Module

5.1 Voter Registration

User Input:
Voter enters their name and Aadhar number.

Smart Contract Call:
The input data is sent to the registerVoter function in the DistributedVoting smart contract.

Voter Storage:
The smart contract stores the voter information in the voters mapping and voterList array.

Admin Verification:
The admin verifies the voter and updates their registered status.

5.2 Vote Casting

User Input:
Voter selects a candidate from the list.

Smart Contract Call:
The selected candidate's name is sent to the castVote function in the DistributedVoting smart contract.

Vote Recording:

The smart contract increments the vote count for the selected candidate and updates the voter's status to voted.

5.3 Vote Tallying

Election End:

The admin calls the endElection function to end the election.

Result Calculation:

The smart contract counts the votes for each candidate.

Result Publication:

The results are retrieved using the getResults function and displayed to the users.

6. Security Measures

Identity Verification:

Uses Aadhar number for voter identity verification.

Anonymity:

Ensures that votes are cast anonymously.

Data Integrity:

Ensures that once a vote is cast, it cannot be altered.

Transparency:

Allows anyone to verify the election process and results.

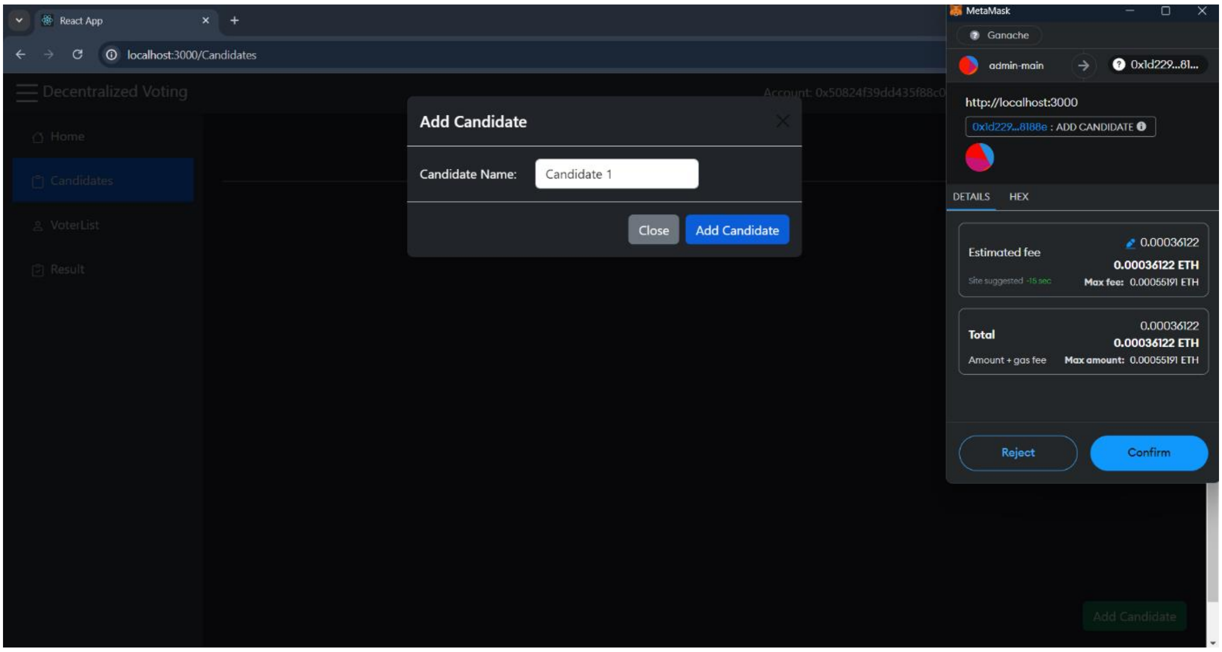
Results Based on Objectives

1. **Add a New Candidate**
    - **Result:** A new candidate is successfully added to the election.
    - **Process:** The administrator enters candidate details (name, party affiliation, etc.) into the election database.
    - **Outcome:** The candidate is officially registered and eligible to participate in the election once approved by the administrator.
2. **Add a New Voter**
    - **Result:** A new voter is successfully registered for the election.
    - **Process:** The administrator verifies voter eligibility and enters voter details (name, ID, etc.) into the election registration system.
    - **Outcome:** The voter receives confirmation of registration and is granted access to participate in voting during the election period.
3. **Display Candidate Details**
    - **Result:** The details of all candidates participating in the election are displayed.
    - **Process:** Users (administrators, voters, etc.) access the election system and navigate to the candidate details section.
    - **Outcome:** Comprehensive information about each candidate, including their profile, manifesto, and other relevant details, is visible and accessible.
4. **Display the Results of the Election**
    - **Result:** The election results, specifically the winner, are displayed.
    - **Process:** After the voting period ends, the system compiles and counts all votes cast.
    - **Outcome:** The system declares the candidate with the highest number of votes as the winner and displays the detailed election results, including vote counts for all candidates.
5. **Start and End Election**
    - **Result:** The election is started or ended as per the administrator's action.
    - **Process:** The administrator initiates the election process by enabling voting and sets a closing date for voting.
    - **Outcome:** Once voting concludes, the administrator ends the election, and the system ceases to accept further votes. Results are then finalized and made available for viewing.

CHAPTER 7

SNAPSHOTS

I. Add Candidate



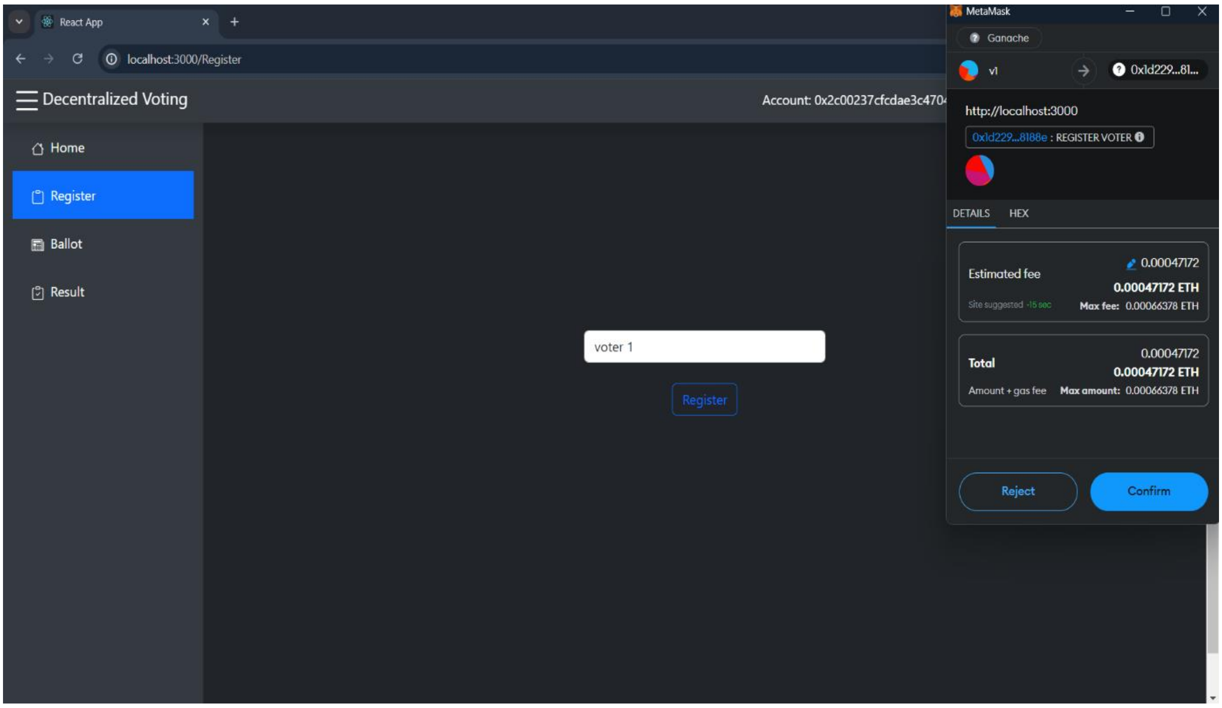
- **Transaction Confirmation**: When you interact with a decentralized application (DApp) that requires transactions (like adding a candidate), MetaMask will prompt you to confirm the transaction. This confirmation typically involves specifying gas fees and confirming the action.

- **Transaction Fee (Gas Fee)**: MetaMask deducts a gas fee from your account for each transaction on the Ethereum blockchain. This fee varies based on network congestion and the complexity of the transaction.

- **Confirmation and Block Confirmation**: After confirming the transaction in MetaMask, it may take a few moments to several minutes for the transaction to be included in a block and confirmed on the blockchain.

- **Checking Transaction Status**: You can check the status of your transaction by clicking on the transaction in MetaMask (under the "Activity" tab). This will show you details such as transaction hash, status (pending or confirmed), and block confirmation.

II. Add Voter



- **Transaction Confirmation**: When you register a voter or perform any action that involves a transaction on the blockchain, MetaMask prompts you to confirm the transaction. This confirmation includes details such as the gas fee (transaction fee) and requires your approval to proceed.

- **Gas Fee Deduction**: MetaMask deducts a gas fee from your account to cover the cost of processing the transaction on the blockchain network. The amount of this fee varies depending on factors like network congestion and the complexity of the transaction.

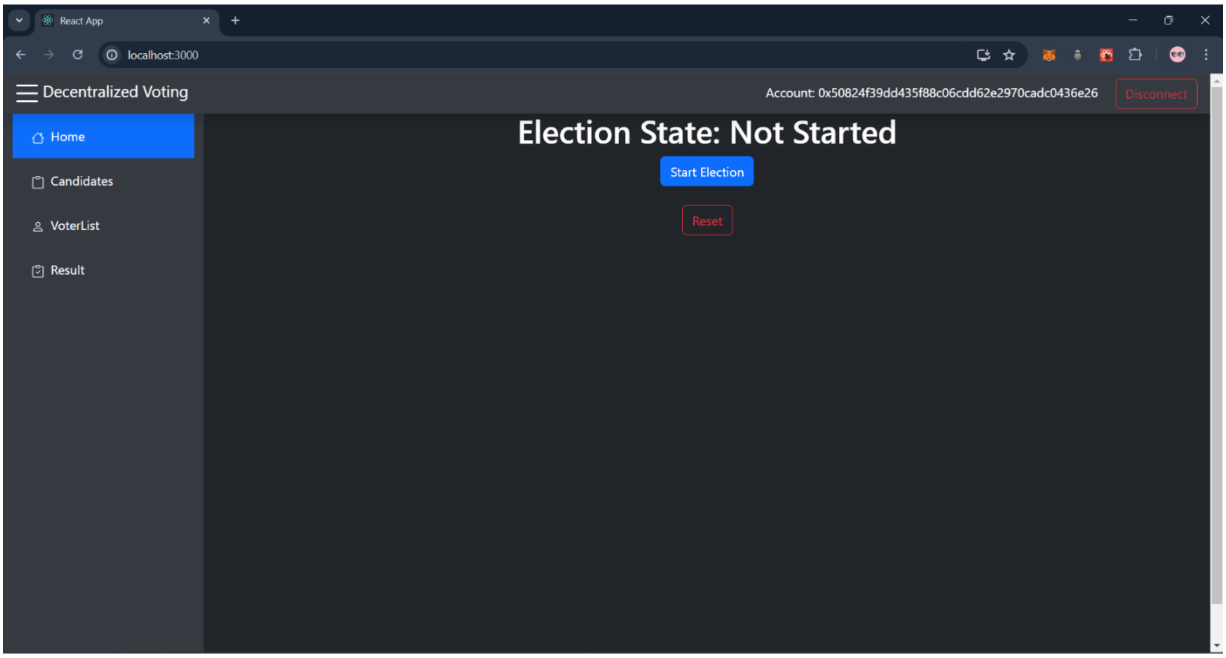
- **Transaction Processing**: Once you confirm the transaction in MetaMask, it is broadcasted to the blockchain network. The transaction then undergoes validation and inclusion in a block by miners.

- **Monitoring Transaction**: You can monitor the progress of your transaction by checking MetaMask's "Activity" tab. Here, you can view details such as the transaction hash (unique identifier), status (pending or confirmed), and the number of block confirmations.

- **Transaction Status**: If the transaction is pending for an extended period, you might consider increasing the gas fee to expedite its processing. Alternatively, if there are issues or errors with the transaction, MetaMask will provide relevant error messages that can help diagnose and resolve the problem.

- **Transaction Completion**: Once the transaction is confirmed (usually within seconds to minutes), the voter registration process is completed on the blockchain, ensuring that the voter's information is securely recorded and verifiable.

III. Start Election

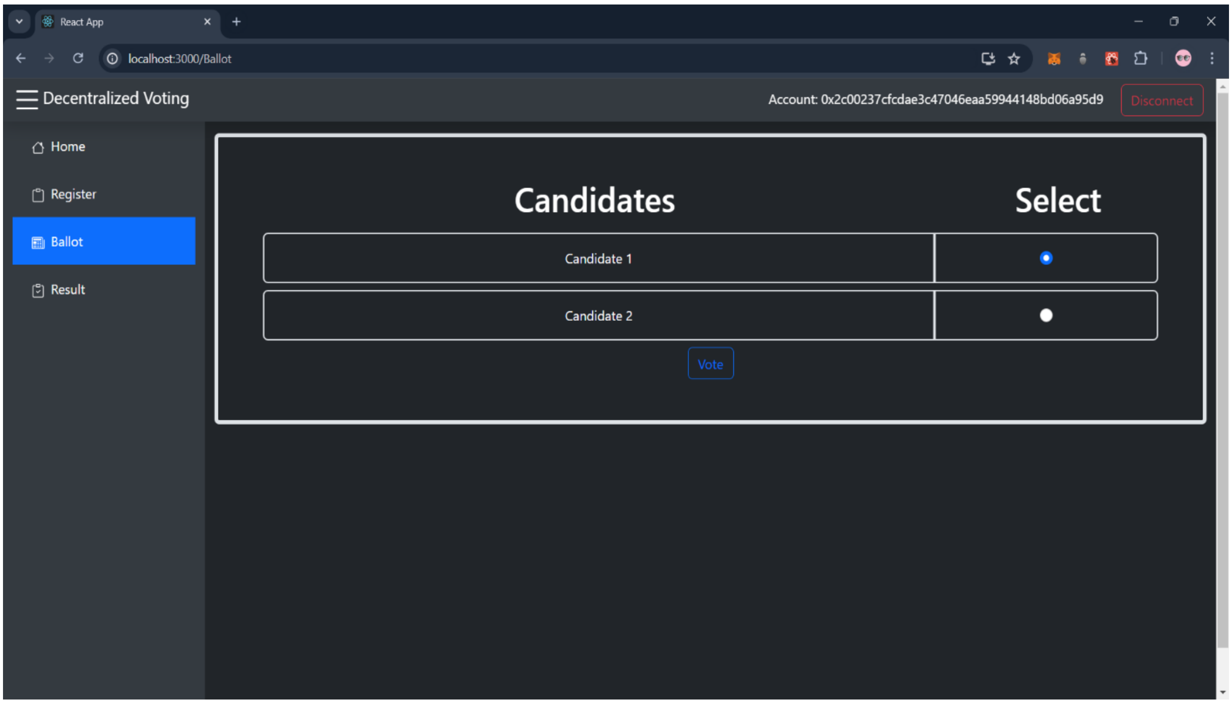


- **Starting the Election**: The administrator initiates the election by clicking "Start Election", prompting a MetaMask transaction for transitioning the smart contract state from "NotStarted" to "Running" upon confirmation.

- **Ending the Election**: Upon the voting period's end, clicking "End Election" triggers a MetaMask transaction to update the smart contract, transitioning the state from "Running" to "Ended" and finalizing the election results.

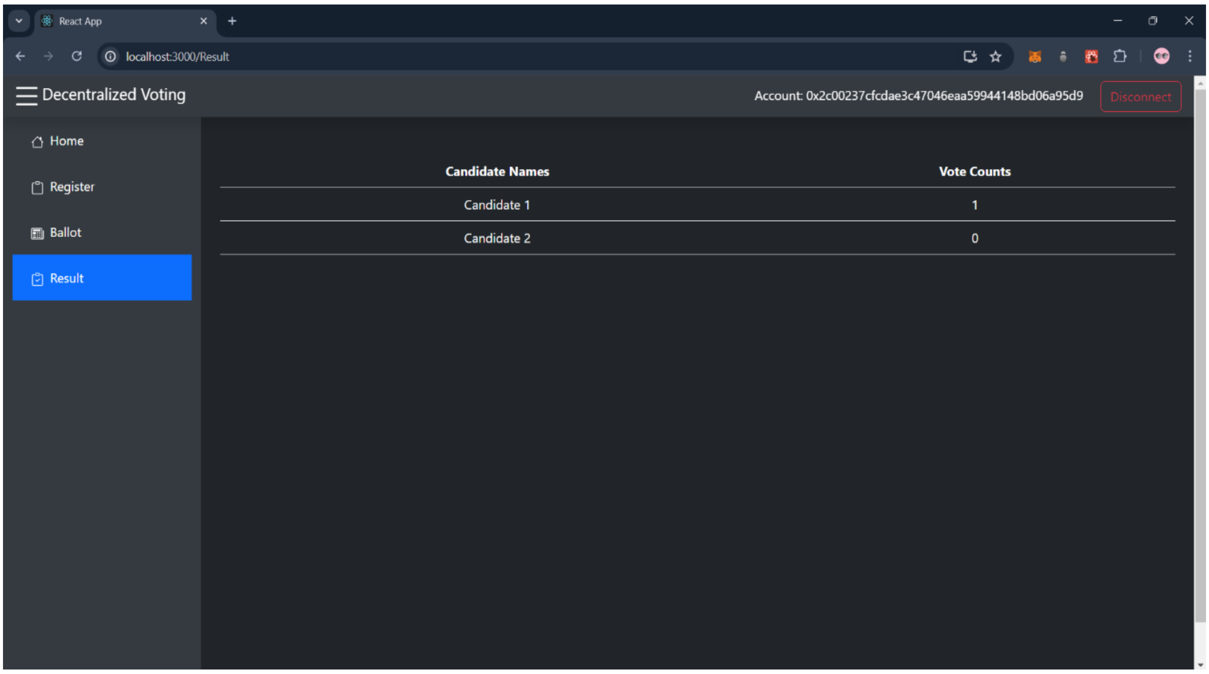
- **Resetting Election Data**: Clicking the "Reset" button initiates a MetaMask transaction to clear election-related data, such as voter statuses and votes, ensuring a clean slate for future electoral processes.

IV. Cast the Vote

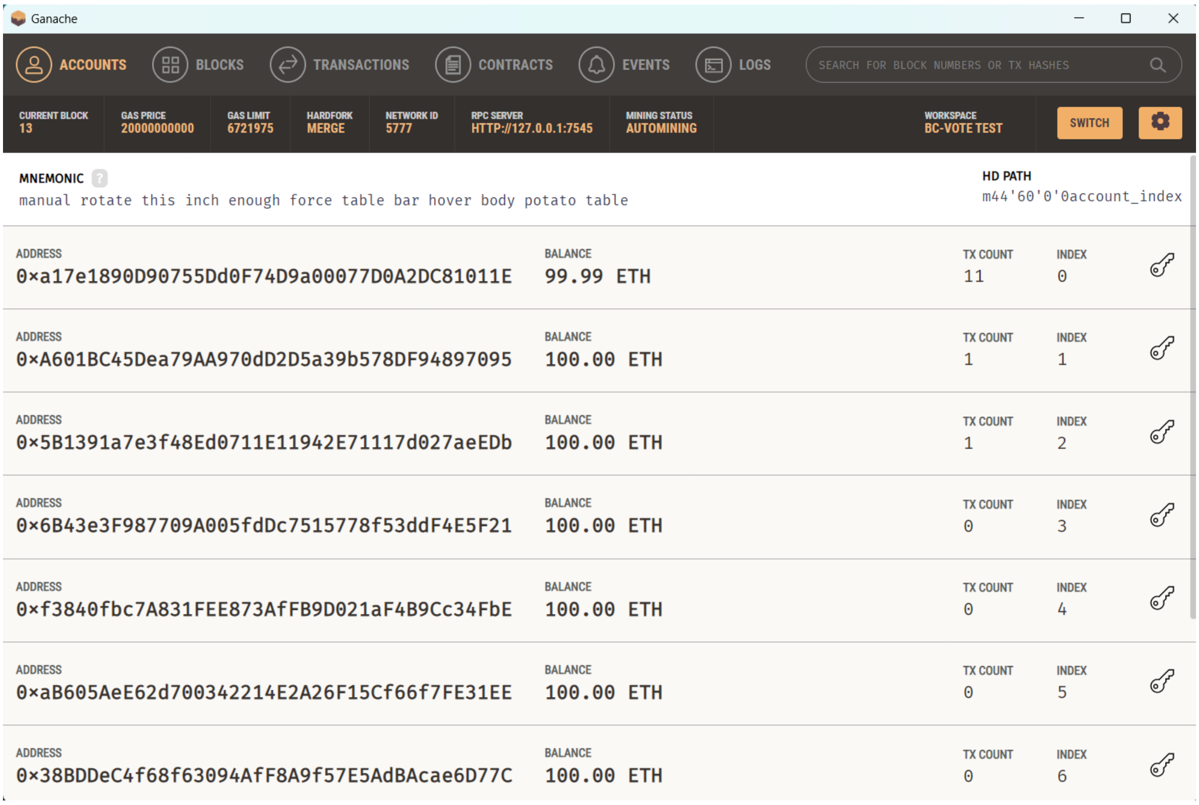


Vote Casting Transaction: When a voter casts their vote by clicking on their preferred candidate, MetaMask prompts a transaction. This transaction deducts a gas fee from the voter's account to process and record the vote securely on the blockchain. This ensures transparency and integrity in the voting process, leveraging blockchain technology for decentralized and verifiable elections.

V. Result



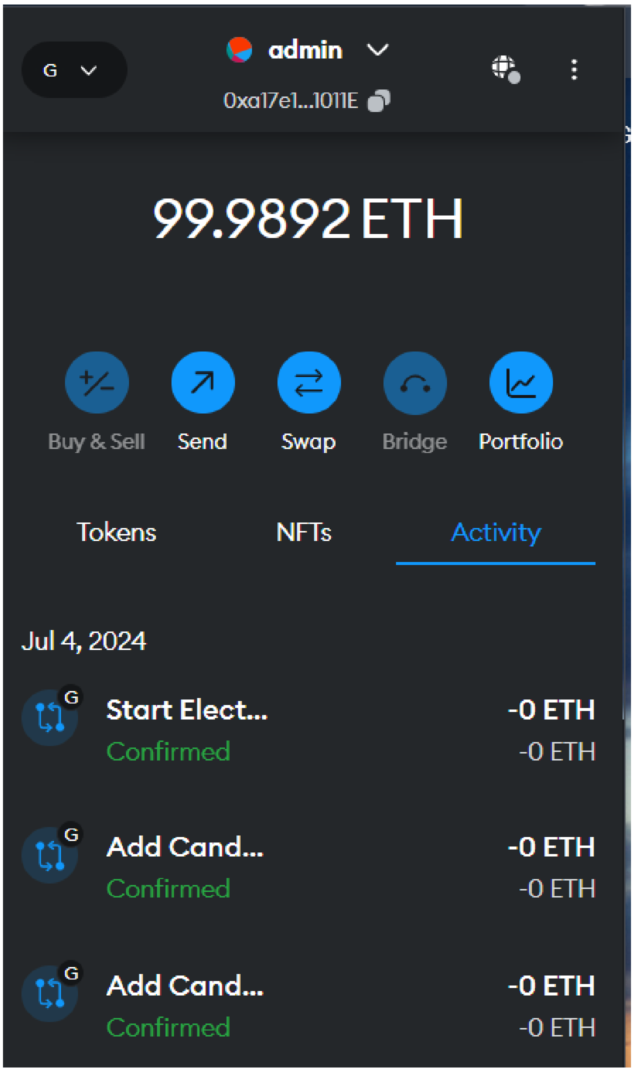
VI. Ganache



The image depicts a transaction recorded on the Ethereum blockchain, showing a deduction of 99 ETH from the account balance. This transaction is visible on the blockchain explorer, confirming the transfer of funds from one address to another. The deduction reflects the gas fee paid for processing the transaction, ensuring the security and validation of the transaction on the decentralized network. Such transactions are essential for executing smart contracts, transferring cryptocurrency, or participating in decentralized applicatins (DApps) securely and transparently.

VII. MetaMask

The MetaMask interface displays a transaction with a confirmed status on the Ethereum blockchain. It includes details such as the sender's address, recipient's address, transaction amount, and the confirmed status indicator. This indicator signifies that the transaction has been successfully processed and validated by the blockchain network. Additionally, MetaMask provides information on the block confirmation number, indicating how many blocks have included the transaction, thereby confirming its validity and permanence on the blockchain. This status ensures transparency and reliability in cryptocurrency transactions and interactions with decentralized applications (DApps) facilitated through MetaMask.



CHAPTER 8

CONCLUSION

A nation with a less voting percentage will fight to develop as choosing a right front-runner for the nation is very essential .our future system is designed to provide a secure data and dependable voting amongst the people of the equality . By assuming blockchain in the distributed database on voting system one can reduce the double dealing source of database management .This project aims to voting effect using blockchain procedure from every place of election

CHAPTER 9

REFERENCES

1. "Blockchain for Secure and Efficient Voting Systems"** by Nikos Fotiou, Vassilis Tsetsos, and George C. Polyzos (2018)

2. "A Survey on Blockchain Voting Systems"** by Kaiwen Zhang, Ximeng Zheng, and Xiangao Jiang (2019)

3. "Blockchain Voting and Its Effects on Election Transparency and Voter Confidence"** by Maria Soledad Martinez and Richard S. Warr (2020)

4. "Secure and Transparent Voting System Based on Blockchain Technology"** by Minhaj Ahmad Khan, Osman Ghazali, and Farhan Aadil (2019)

5. "An Overview of Blockchain Voting for Government Decision Making"** by Florian Dold, Dominik Schiener, and Walid Al-Saqaf (2017)

6. "Towards Secure and Verifiable Blockchain-Based Remote Voting"** by Alexei Zamyatin, Mike Hamburg, and Pedro Moreno-Sanchez (2020)

7. "Blockchain Voting Systems: A Survey and Taxonomy from the Governance Perspective"** by Massimo Bartoletti, Livio Pompianu, and Amrit Kumar (2019)

8. "Blockchain Voting Systems: Enhancing Transparency and Trust in Digital Democracies"** by Peter Y.A. Ryan and Filip Zagorski (2018)

9. "A Secure E-Voting System Based on Blockchain Technology"** by Chao Wang, Kuo-Ming Chao, and Cheng-Fu Chou (2019)

10. "Blockchain Technology in Voting Systems: Decentralization, Transparency, and Security"** by David Mazières and Joan Feigenbaum (2018)

11. "Blockchain-Based Voting Systems: A Case Study and Proposal"** by Aljosha Judmayer, Nicholas Stifter, and Alexei Zamyatin (2018)

12. "Blockchain Voting: A Simple Strategy towards Resolving Election Issues"** by Oyewole Ademola, Selvachandran G., and Adigun M. Oyebode (2021)

13. "Secure E-Voting Using Blockchain Technology"** by Anjana Gupta, Nidhi Kushwaha, and A. K. Misra (2020)

14. "Design and Implementation of Blockchain-Based E-Voting System"** by Ashiq Anjum, Muhammad Salman Razzaq, and Sungyoung Lee (2019)