## Blockchain Based Voting System

A PROJECT REPORT

***Submitted by***

### 21BCS9085 Jatin Verma

***in partial fulfillment for the award of the degree of***

### BACHELOR OF ENGINEERING

##### IN

COMPUTER ENGINEERING



Chandigarh University

NOVEMBER 2024



### BONAFIDE CERTIFICATE

Certified that this project report “Blockchain Based Voting System” is the Bonafide work of “Jatin Verma” who carried out the project work under our supervision.

**SIGNATURE SIGNATURE**

**HEAD OF THE DEPARTMENT AMANPREET KAUR**

##### AIT-CSE

Submitted for the project viva-voce examination held on

**INTERNAL EXAMINER EXTERNAL EXAMINER**

## TABLES OF CONTENTS

List of Images i

List of Tables iii

[Abstract iv](#_TOC_250004)

Chapter 1 : Introduction 1-5

: Problem Identification 1

: Post-Operative Rehabilitation 1

: The Role of ADIP-CI Software 3

: Challenges in Post-Operative Rehabilitation management 5

[: Existing System……………………………………………………..](#_TOC_250003)

:Problem Formulation………………………………………………..

[: Proposed System………………………………………………….](#_TOC_250002)

: Objectives………………………………………………………..

Chapter 2 : Literature Survey 6

:Post-Operative Rehebilitation challenges 12

: Unleasing the Potential of ADIP-CI Software 18

: A Complex and vital Endeavor 19

:The Perils of Traditional Method 20

: The Visionary Potential of ADIP-CI Software 20

2.9:Literature Review Summary Table

Chapter 3 : Design Flow / Methodology 21

* 1. : Implementation 28

Chapter 4 : Result Analysis 35

[Chapter 5 : Conclusion & Future Scope 42-46](#_TOC_250001)

: Conclusion42

: Discussion 43

[: Future Scope 45](#_TOC_250000)

### LIST OF FIGURES

1. System Architecture 8
2. Level 0 data flow diagram 12
3. Level 1 data flow diagram 12
4. Level 2 data flow diagram 13
5. ER Diagram 13
6. Use Case Diagram 14

### ABSTRACT

Voting using blockchain is a secure, transparent and tamper-proof way of conducting online voting. It is a decentralized application built on the Ethereum blockchain network, which allows participants to cast their votes and view the voting results without the need for intermediaries. In this system, votes are recorded on the blockchain, making it impossible for anyone to manipulate or alter the results. The use of smart contracts ensures that the voting process is automated, transparent, and secure. The use of blockchain technology and the implementation of a decentralized system provide a reliable and cost-effective solution for conducting trustworthy and fair elections

# CHAPTER 1

## INTRODUCTION

### Introduction to Blockchain

Blockchain is a distributed digital ledger technology that allows participants in a network to share and validate transactions in a secure and transparent manner without the need for intermediaries. The technology is designed to be decentralized, meaning that the data is stored on a network of computers instead of a central database. This makes it difficult to hack or manipulate the data, ensuring the integrity and security of the system.

The blockchain technology gained popularity with the emergence of Bitcoin, which was the first decentralized cryptocurrency. However, the technology has since been applied to various industries, including finance, supply chain management, healthcare, and voting, among others.

Blockchain works by creating blocks of data that are linked together in a chain, hence the name blockchain. Each block contains a unique code, known as a hash, that is generated based on the contents of the block. This hash is then used to link the block to the previous one, forming a chain of blocks.

Once a block is added to the blockchain, it cannot be altered or deleted without the consensus of the network participants. This makes the technology immutable, ensuring that the data stored on the blockchain is tamper-proof and transparent.

Overall, blockchain technology has the potential to revolutionize the way we store and share data, making it more secure, transparent, and accessible.

* 1. Decentralized Voting Using Blockchain

A decentralized voting system built on the Ethereum blockchain has the potential to revolutionize the way we conduct elections. By leveraging the security, transparency, and immutability of blockchain technology, decentralized voting systems can eliminate many of the challenges and risks associated with traditional voting systems.

In a decentralized voting system, each voter has a unique digital identity, and their vote is recorded on the blockchain, ensuring that the vote is tamper-proof and cannot be altered. Decentralized voting systems also eliminate the need for intermediaries, such as government agencies, to oversee the election process, making it more efficient and less susceptible to corruption or manipulation.

Furthermore, decentralized voting systems can increase voter participation by allowing voters to cast their ballots from anywhere in the world, as long as they have an internet connection. This can lead to a more democratic and inclusive electoral process, with greater voter engagement and higher turnout. Overall, a decentralized voting system using the Ethereum blockchain has the potential to bring significant benefits to the electoral process, making it more secure, transparent, and accessible to everyone.

The rapid advancement of digital technologies has significantly transformed the way societies function, particularly in the realm of governance and elections. Traditional voting systems, while fundamental to democratic processes, are often plagued by various challenges such as voter fraud, lack of transparency, and inefficient handling of data. In this context, blockchain technology emerges as a promising solution to modernize voting systems and address the inherent flaws in conventional methods.

Blockchain, often associated with cryptocurrencies like Bitcoin, is a decentralized and secure technology that has found applications far beyond financial transactions. By using blockchain for voting, we can ensure transparency, prevent tampering, and safeguard voter privacy. Blockchain-based voting offers a unique solution that guarantees the integrity of votes, reduces the risk of manipulation, and improves voter accessibility.

The potential benefits of implementing blockchain in voting are vast. It can create a voting system that is not only more secure and transparent but also more inclusive, providing a method for citizens to vote from anywhere in the world. This can lead to higher voter engagement and ensure the results are verifiable and tamper-proof.

This project explores the use of blockchain technology to develop a decentralized voting platform, with a focus on leveraging the Ethereum blockchain to facilitate secure and transparent elections. Through the integration of smart contracts and cryptographic techniques, this system promises to address the critical issues that have plagued traditional voting methods, offering a more efficient and trustworthy electoral process.

By the end of this project, the goal is to present a functional blockchain-based voting system that can be implemented for secure, transparent, and decentralized elections,

demonstrating the transformative power of blockchain in governance and democracy.

In addition to increased accessibility and security, blockchain-based voting systems can provide enhanced data integrity and auditability. Since all votes are recorded on a public ledger that is immutable, each voter can verify that their vote has been counted as intended. This transparency builds trust in the election process, addressing one of the key challenges faced by traditional voting systems.

Furthermore, by reducing the need for physical infrastructure and intermediaries, blockchain can lower the overall cost of conducting elections, making it a more efficient solution.

The implementation of blockchain also enhances accountability in elections. The transparency of the system allows stakeholders, such as election monitors and independent auditors, to review the results in real time, ensuring that the voting process is free from manipulation. Moreover, the use of smart contracts enables the automation of election rules, such as vote tallying and result announcements, further reducing the chances of human error or bias.

One of the most crucial aspects of this project is the ability to maintain voter anonymity while ensuring the accuracy and verifiability of votes. Blockchain technology allows for cryptographic techniques that secure each vote while ensuring that voter identities remain confidential, thus addressing concerns over privacy and potential voter intimidation.

This project explores the use of blockchain technology to develop a decentralized voting platform, with a focus on leveraging the Ethereum blockchain to facilitate secure and transparent elections. Through the integration of smart contracts and cryptographic techniques, this system promises to address the critical issues that have plagued traditional voting methods, offering a more efficient and trustworthy electoral process.

By the end of this project, the goal is to present a functional blockchain-based voting system that can be implemented for secure, transparent, and decentralized elections, demonstrating the transformative power of blockchain in governance and democracy. The implementation of such a system has the potential to drastically improve election processes worldwide, eliminating fraud, enhancing participation, and ultimately strengthening democratic practices.

# CHAPTER 2

## LITERATURE SURVEY

### Literature survey on Online Voting System Using Blockchain

**Authors:** Vaibhav Anasune, Pradeep Choudhari, Madhura Kelapure, Pranali Shirke and

Prasad Halgaonkar

Highly advanced security methods are necessary to introduce effective online voting system in the whole world. The aspect of security and transparency is a threat from global election with the conventional system. General elections still use a centralized system where one organization that manages it. Some of the problems that can occur in traditional electoral systems are with an organization that has full control over the database and system, it is possible to manipulate with the database. This paper presents a survey on some previous voting system that is used by different countries and organizations.

### A Systematic Literature Review and Meta-Analysis on Scalable Blockchain-Based Electronic Voting Systems

**Authors:** Uzma Jafar, Mohd Juzaiddin Ab Aziz, Zarina Shukur and Hafiz Adnan Hussain

Electronic voting systems must find solutions to various issues with authentication, data privacy and integrity, transparency, and verifiability. On the other hand, Blockchain technology offers an innovative solution to many of these problems. The scalability of Blockchain has arisen as a fundamental barrier to realizing the promise of this technology, especially in electronic

voting. This study seeks to highlight the solutions regarding scalable Blockchain-based electronic voting systems and the issues linked with them while also attempting to foresee future developments. A systematic literature review (SLR) was used to complete the task, leading to the selection of 76 articles in the English language from 1 January 2017 to 31 March 2022 from the famous databases. This SLR was conducted to identify well-known proposals, their implementations, verification methods, various cryptographic solutions in previous research to evaluate cost and time. It also identifies performance parameters, the primary advantages and obstacles presented by different systems, and the most common approaches for Blockchain scalability. In addition, it outlines several possible research avenues for developing a scalable electronic voting system based on Blockchain technology. This research helps future research before proposing or developing any solutions to keep in mind all the voting requirements, merits, and demerits of the proposed solutions and provides further guidelines for scalable voting solutions.

### A Survey of Blockchain Based on E-voting Systems

**Authors:** Yousif Osman Abuidris, Rajesh Kumar and Wang Wenyong

Blockchain technology as a decentralized and distributed public ledger in a P2P network has recently gained much attention. In this technology, a linked block structure is applied, and a trusted consensus mechanism is established to synchronize data modifications, making it possible to develop a tamper-proof digital platform for data storage and sharing. We think that blockchain could be used in various interactive online systems, such as the Internet of Things, supply chain systems, voting systems, etc. The scope of this survey is to shed light on some recent contributions of the security and privacy issues associated with e- voting based on blockchain. At the end of this paper, we provided a comparison for the security and privacy requirements of the existing e-voting systems based on blockchain.

### Survey on Voting System using Blockchain Technology

**Authors:** Mayur Shirsath, Mohit Zade, Riteshkumar Talke, Praful Wake and Maya Shelke

The use of information technology has in some ways revolutionized in many sectors. E-voting is said to be a symbol of modern democracy. While research on the topic is still emerging, it has mostly focused on the technical and legal issues instead of taking advantage of this technology and implementing it for good cause. Usefulness of e-voting will perform best when compared with the existing framework. The word Vote means to choose a candidate from a given list of candidates who will lead the organization or the group .The main goal of voting is to practice voting in such a way that every person votes to elect their leader. Most countries in the world, India is no exception, had trouble voting. Voting is still carried out in countries in physical mode. This physical mode process is not safe as it can be manipulated by members of voting commitment. There are many issues such as voting stations being too far and improper voting tools. The proposed flagship internet-based online voting system supported by blockchain technology solves this very problem. Blockchain technology uses encryption and hashing techniques with which it makes voting secure. In this case, each vote is considered as a unique transaction. A private blockchain is created using a peer to peer network where we store voting transactions. This application is programmed in such a way so that the details of voting are abstract from the user. Users will be given enough time for voting with the system running. The main purpose of this paper is to come up with a new unique solution, which does not require any technical skills. Since voting is in online mode, increased voter turnout is likely. In this project, the concept of developing an electronic voting system using blockchain technology is implemented.

* 1. **A Survey on Smart Electronic Voting System Using Blockchain Technology Authors:** Naina Nagesh Dhepe and Dr. Pathan Mohd Shafi

India is the world’s largest democracy with a population of more than 1 billion;

India has an electorate of more than 668 million and covers 543 parliamentary constituencies. Voting is the bridge between the governed and government. The last few years have brought a renewed focus on to the technology used in the voting process. The current voting system has many security holes, and it is difficult to prove even simple security properties about them. A voting system that can be proven correct has many concerns. There are some reasons for a government to use electronic systems are to increase elections activities and to reduce the elections expenses. Still there is some scope of work in electronic voting system because there is no way of identification by the electronic voting system whether the user is authentic or not and securing electronic voting machine from miscreants. The proposed system is to develop a compatible voting machine with high security by using Block-chain technology in order to increase security and transparency between the users.

# CHAPTER 3

## EXISTING SYSTEM

### 3.1 Brief Explanation of existing system

The existing voting system typically involves voters physically visiting a designated polling place to cast their vote on paper ballots. These ballots are then manually counted and recorded. Some countries also have electronic voting systems in place, which allow voters to cast their votes electronically through machines or the internet. However, electronic voting systems have faced criticism due to security concerns and potential vulnerabilities. Existing voting systems vary widely in their design and implementation, but many share similar challenges, such as issues related to security, transparency, voter accessibility, and integrity of the voting process. Traditional voting systems can be broadly categorized into two main types: paper-based voting and electronic voting (e-voting). Each of these systems has its own set of strengths and weaknesses.

* 1. Paper-Based Voting Systems

In paper-based voting, voters cast their votes on paper ballots, which are then counted manually by election officials. This method has been used for centuries and remains the most common form of voting in many countries, especially in local elections. The advantages of paper-based voting include:

* + - **Familiarity**: Many people are comfortable with the paper-based process because it has been used for generations.
    - **Tangible Record**: Paper ballots provide a physical record of votes, which can be useful in recounts or investigations.
    - **Low Initial Cost**: Setting up a paper-based election may require fewer upfront investments compared to digital or electronic systems.

However, paper-based voting systems suffer from several significant drawbacks:

* + - **Voter Fraud**: Paper ballots are vulnerable to fraud, such as vote tampering, ballot stuffing, and manipulation by election officials.
    - **Lack of Transparency**: The counting process is not always visible to the public, leading to questions about the integrity of the results.
    - **Slow Results**: The manual counting of votes can take a significant amount of time, delaying the announcement of results.
    - **Limited Accessibility**: Voters are required to be physically present at polling stations, making it difficult for those with disabilities or who live in remote areas to vote.
  1. Electronic Voting Systems (E-Voting)

Electronic voting (e-voting) systems are designed to replace paper ballots with digital technology, streamlining the voting process and making it more efficient. E- voting systems can be divided into two categories:

1. **Direct Recording Electronic (DRE) Systems**: These systems allow voters to cast their votes on electronic voting machines, which are typically located at polling stations. The votes are stored electronically, and the machine records and transmits the results after the polls close.
   * Advantages:
     + Faster vote counting and result reporting.
     + Reduced chances of human error in counting.
     + Some systems offer features like accessibility for disabled voters.
   * Disadvantages:
     + **Vulnerability to Hacking**: E-voting machines are susceptible to cyberattacks and hacking, which can lead to manipulation of the results.
     + **Lack of Transparency**: Voters may not have a way to verify that their vote was accurately recorded or counted.
     + **Reliability**: Technical issues, such as machine malfunction or system failures, can disrupt the voting process.
     + **Security Concerns**: Unauthorized access to the machines or databases storing votes can compromise voter privacy and election integrity.
2. **Internet Voting**: This method allows voters to cast their ballots online using a secure internet platform. While this approach offers greater accessibility, it also presents several challenges:
   * Advantages:
     + Voters can vote from anywhere with an internet connection, increasing accessibility.
     + Faster vote counting and result reporting.
     + Lower cost for election administration, as there is no need for physical polling stations or paper ballots.
   * Disadvantages:
     + **Security Risks**: Online voting systems are vulnerable to cyberattacks, such as hacking, data breaches, and DDoS attacks. There are concerns about the privacy and integrity of votes.
     + **Voter Authentication**: Ensuring that only eligible voters participate and preventing fraudulent voting can be challenging in an online environment.
     + **Digital Divide**: Not all voters have access to the necessary technology or internet connectivity, creating inequities in the voting process.
   1. Challenges with Existing Voting Systems

Despite advancements in technology, both paper-based and electronic voting systems continue to face several common challenges:

* + - **Fraud and Manipulation**: Traditional voting systems, whether paper-based or electronic, are vulnerable to fraud and manipulation, from vote tampering to the manipulation of results.
    - **Lack of Transparency**: In many cases, voters do not have visibility into the election process, from casting votes to tallying results. This lack of transparency can erode public trust in the system.
    - **Security Concerns**: Both paper-based and electronic systems are susceptible to security breaches, such as hacking, which can compromise the integrity of the election.
    - **Voter Privacy**: Ensuring that voters’ identities remain anonymous while their votes are counted accurately is a critical concern in all voting systems. Traditional systems often fail to guarantee this level of privacy.
    - **Accessibility**: Many traditional systems require voters to be physically present at polling stations, limiting participation for people with disabilities, those who live in remote areas, or those with time constraints.
    - **High Costs**: Managing and securing large-scale elections, whether through paper ballots or electronic machines, can be costly, particularly in countries with large electorates.

Potential Improvements and Future of Voting Systems

Blockchain technology presents a compelling solution to many of the challenges facing existing voting systems. By ensuring transparency, security, and immutability of voting data, blockchain can provide a more trustworthy, efficient,

and accessible voting system. Furthermore, blockchain’s decentralized nature reduces the risk of fraud, as no central authority can manipulate or alter the results. This innovation holds the potential to address the security flaws and accessibility barriers found in both paper-based and electronic voting systems, creating a more secure, democratic, and inclusive electoral process.

The shift to blockchain-based voting systems, while still in its early stages, could revolutionize how elections are conducted globally. Through the use of smart contracts, cryptographic techniques, and decentralized networks, blockchain can ensure that each vote is securely recorded and counted, making elections more transparent and reliable for all participants.

### Disadvantages of existing system

1. **Lack of transparency**: In most voting systems, it's difficult for voters to know whether their vote was counted correctly, and for observers to ensure that the vote counting process is fair.
2. **Vulnerability to fraud**: Both paper ballots and electronic voting machines can be vulnerable to tampering, hacking and other types of fraud. This can be especially problematic when there is no paper trail or other way to audit the results.
3. **Slow results**: Counting paper ballots can be a time-consuming and labor-intensive process, which can delay the announcement of election results.
4. **Cost**: Running a traditional voting system can be expensive, requiring the hiring of poll workers, the purchase of voting machines or paper ballots, and the rental of polling places.
5. **Centralization**: Many traditional voting systems are centralized, meaning that they are controlled by a small number of authorities. This can create the potential for abuse of power or manipulation of the voting process.
6. **Limited Accessibility**: Some voting systems require voters to travel to specific polling places, which can be difficult or impossible for people with disabilities, limited mobility, or other challenges. This can result in voter disenfranchisement.

# CHAPTER 4

## PROPOSED SYSTEM

### Brief explanation of proposed system

The proposed decentralized voting system using Ethereum blockchain aims to provide a transparent and tamper-proof solution for conducting elections. By leveraging smart contracts on the Ethereum network, the system enables secure and anonymous voting, while ensuring the integrity and immutability of the voting data. This would increase voter trust in the election process and reduce the risk of fraud or manipulation.

### Advantages of Proposed System

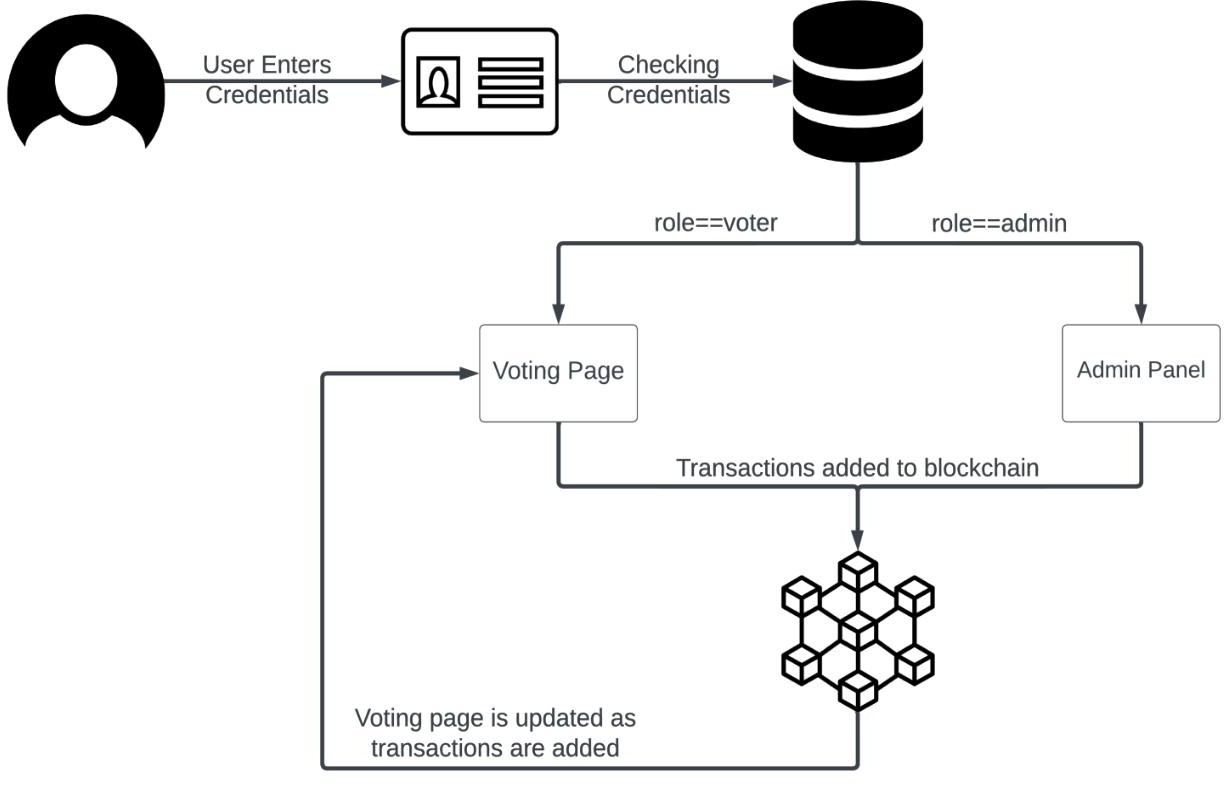
* + - Decentralization ensures that no party controls the voting process.
    - Transparency throughout the voting process.
    - It is tamper proof.
    - Voters can vote from any part of the world.
    - This method of voting is cost effective.
    - The results are provided in real time.

### Objectives of the Proposed Research

1. **Security:** The proposed system aims to provide a secure platform for conducting elections, eliminating the possibility of tampering with votes, and ensuring that the election results are transparent and verifiable.
2. **Transparency:** The proposed system aims to provide complete transparency to the voters, allowing them to view the entire voting process, including the vote counting and results
3. **Accessibility:** The proposed system aims to make the voting process more accessible to all eligible voters by eliminating the need for physical presence at a polling station, thus increasing voter turnout.
4. **Efficiency:** The system aims to increase the efficiency of the voting process by reducing the time and resources required to conduct elections. Since the system is automated and eliminates the need for intermediaries, it can significantly reduce the cost and time associated with traditional voting methods.
5. **Trust:** The proposed system aims to increase trust in the voting process by providing a transparent and tamper-proof mechanism for recording and tallying votes.

# CHAPTER 5

## DESIGN FLOW / METHODOLOGY



**Figure 1** System Architecture

User enters the credentials (voter id & password) and they are matched with the database. If the match is found user is either redirected to admin page or voter page as per their role corresponding to the credentials in the database. Once the admin is logged in he/she can start the voting process by adding candidates and defining dates. Voter can vote once the voting process has been started. Once the voter has voted the transaction is recorded to the blockchain and the voting page is updated with real-time votes.

The design flow for a blockchain-based decentralized voting system follows a systematic approach to ensure that each step of the process is designed, developed, and implemented in a secure, transparent, and efficient manner. The methodology ensures that the solution can be successfully deployed, tested, and used by the target audience while addressing the challenges associated with traditional voting systems. Below is an outline of the design flow and methodology for this blockchain-based voting system:

* + 1. Step 1: Requirement Gathering and Analysis

Before any development begins, it is crucial to gather and analyze the requirements of the system. This step involves:

* + - * Identifying Stakeholders: This includes election administrators, voters, and any other entities involved in the election process (e.g., third-party observers).
      * Understanding the Election Process: Analyzing the specific requirements of the voting system, including voter registration, casting votes, result tallying, and voter privacy.
      * Security Considerations: Identifying potential security risks, including voter fraud, data tampering, and hacking.
      * Legal and Regulatory Compliance: Ensuring that the voting system adheres to relevant laws and regulations, including voter anonymity and data protection.
    1. Step 2: System Architecture Design

Once the requirements are gathered, the next step is designing the system architecture. This includes:

* + - * Blockchain Platform Selection: Choosing a blockchain platform, such as Ethereum, which is well-suited for creating decentralized applications (DApps) using smart contracts.
      * System Components: Identifying and designing the major components of the system, including:
        + Frontend (User Interface): The interface where voters interact with the system (e.g., web portal or mobile app).
        + Backend (Server-side): The logic that processes the user's actions and communicates with the blockchain.
        + Blockchain Layer: The decentralized ledger where all votes will be recorded immutably.
        + Security Layer: A layer to ensure encryption, voter anonymity, and authentication.
    1. Step 3: Blockchain and Smart Contract Development

Blockchain and smart contracts form the backbone of this voting system. The main tasks in this phase include:

* + - * Smart Contract Design: Smart contracts will define the voting rules, voter eligibility, and the actual voting process. These contracts will be written in Solidity for the Ethereum blockchain.
      * Voting Contract Functions:
        + Voter Registration: Smart contracts handle user registration and assign each voter a unique digital identity.
        + Vote Casting: The smart contract ensures that votes are cast securely and recorded on the blockchain. Once a vote is cast, it cannot be altered, ensuring immutability.
        + Result Calculation: The smart contract will also tally the votes automatically and present the results in real-time.
      * Blockchain Interaction: The backend will interact with the Ethereum blockchain using Web3.js to send transactions to the network, trigger smart contracts, and retrieve data (e.g., vote counts).
    1. Step 4: Frontend Development

Frontend development focuses on creating the user interface where voters can interact with the system. Key elements of the frontend include:

* + - * Voter Registration: An easy-to-use interface where voters can sign up, undergo identity verification, and receive their unique digital identity.
      * Voting Interface: A simple interface for voters to cast their votes securely and anonymously.
      * Vote Verification: A feature that allows voters to verify that their vote has been successfully recorded on the blockchain.
      * Real-time Results: A live dashboard showing the results as they are tallied by the smart contract.

Tools such as React or Vue.js can be used for creating a dynamic and responsive web application.

* + 1. Step 5: Security Implementation

Security is a critical aspect of any voting system. In this phase, the following security measures are implemented:

* + - * Encryption: Voter data and votes are encrypted to ensure privacy. Public key infrastructure (PKI) is used for encrypting and securing communication.
      * Voter Authentication: Multi-factor authentication (MFA) or biometric verification will be used to ensure that only eligible voters participate in the election.
      * Anonymity: Blockchain’s inherent design ensures that while votes are recorded and counted, the identity of the voter remains anonymous. However, voter authentication is still necessary to prevent fraud.
    1. Step 6: Backend Development and Integration

The backend is responsible for managing voter authentication, registration, and the interaction with the blockchain network. Key tasks in backend development include:

* + - * Voter Authentication: Handling the registration process, validating the voter’s identity, and issuing a unique digital identity using smart contracts.
      * Data Flow Management: Ensuring secure and efficient data flow between the frontend, backend, and blockchain layers.
      * Interfacing with Blockchain: Using libraries like Web3.js to communicate with the Ethereum blockchain, trigger smart contracts, and fetch voting data.

Node.js, along with Express.js, can be used to build the API that facilitates communication between the frontend and the blockchain.

* + 1. Step 7: Testing and Quality Assurance

Testing ensures that the system functions as intended and is free from security vulnerabilities. This step includes:

* + - * Unit Testing: Testing individual components of the smart contracts to verify that they execute the intended logic.
      * Integration Testing: Verifying that the system components (frontend, backend, and blockchain) work together seamlessly.
      * Security Audits: Performing a thorough security audit of the smart contracts and system infrastructure to identify potential vulnerabilities.
      * Usability Testing: Ensuring the system is user-friendly and accessible to a wide audience.
    1. Step 8: Deployment and Monitoring

Once the system is thoroughly tested, the final step is deployment. This includes:

* + - * Blockchain Deployment: Deploying the smart contracts on the Ethereum network (or a private blockchain network).
      * Cloud Hosting: Hosting the frontend and backend on a cloud platform, ensuring scalability and availability.
      * Monitoring: Continuously monitoring the system for any security breaches, technical failures, or performance issues.
    1. Step 9: Training and Support

To ensure smooth adoption, election administrators will be trained on how to use the platform. Support will also be provided for any issues faced by voters or administrators during the election process.

Conclusion

The methodology for designing and developing a blockchain-based voting system follows a structured approach that prioritizes security, transparency, and efficiency. By using blockchain technology, the system ensures tamper-proof results, eliminates fraud, and enhances voter trust and participation. This methodology, when followed carefully, can lead to the successful implementation of a decentralized voting system that can improve the overall electoral process.

### Modules

* + 1. **Voter -** The voter module is designed for individuals who are eligible to participate in the voting process. It provides functionalities related to the voting experience and ensures the integrity and security of the votes. The main features of the voter module include:
       1. Voters can securely authenticate themselves to access the voting system using their unique credentials.
       2. Voters can access information about the candidates running for various positions, such as their names, parties, and other relevant details.
       3. Voters can verify the status of their votes and ensure that their choices are accurately recorded in the blockchain.
    2. **Admin -** The admin module is designed for administrators or election officials responsible for managing and overseeing the voting system. It provides functionalities to configure and monitor the voting process. The main features of the admin module include:
       1. Admins can set up the system parameters, such as defining the start and end dates of the voting period, candidate registration, and other administrative settings.
       2. Admin can manually verify the candidate and can start the voting process.

###### Requirement Gathering and Analysis

This is the initial phase where the scope of the voting system is defined. The project team collects detailed requirements through stakeholder interviews, workshops, and surveys.

* **Defining the Election Scope**: Determine the types of elections (local, national, organizational) that the system will support and whether there are any specific regulatory or legal constraints (e.g., voter eligibility, election fraud prevention).
* **Technology Stack Selection**: Evaluate and decide the blockchain platform (e.g., Ethereum, Hyperledger) based on the specific needs of the project, such as scalability, transaction speed, and support for smart contracts.
* **System Stakeholder Needs**: Gather input from voters, administrators, and regulatory bodies to ensure all user groups’ needs are addressed. This includes:
  + Voters needing easy access and strong anonymity.
  + Election officials requiring a secure, fraud-resistant, and efficient tool for managing elections.
  + Transparency needs for observers and regulators.

###### Step 2: System Architecture Design

The next step involves creating a high-level architecture that specifies how all components of the system will work together. The architecture is designed to be modular, scalable, and secure.

* + - * **Decentralization**: At the core of the architecture is the use of blockchain to ensure data is not controlled by a central authority. This ensures no single entity can manipulate voting records.
      * **Layered Architecture**: The system can be divided into several layers:
        + **Frontend Layer**: The user-facing interface, responsible for interacting with voters.
        + **Backend Layer**: Handles user requests, authentication, and interfaces with the blockchain.
        + **Blockchain Layer**: Where the actual voting process and data are stored and validated.
        + **Security Layer**: A dedicated security layer that ensures data encryption, voter privacy, and transaction integrity.
      * **Scalability and Performance**: The system should be designed to handle a large number of voters, especially during national elections. The architecture should be scalable to avoid bottlenecks during peak voting times.

###### Step 3: Blockchain and Smart Contract Development

This step is crucial as it implements the foundation of the voting system— secure and immutable data recording through blockchain and smart contracts.

###### Smart Contract Development:

* + - * + **Voting Contract**: This is the contract responsible for managing the entire election process, from voter registration to vote tallying. It ensures the integrity of each vote cast and calculates the final result.
        + **Eligibility Checks**: A smart contract that enforces the rules of voter eligibility. It ensures that only those who have passed authentication (e.g., ID verification) can vote.
        + **Vote Privacy**: Smart contracts will not store personal voter information but will instead store encrypted, anonymized voting records.
        + **Result Generation**: Once the voting period ends, the smart contract automatically tallies the votes and generates the results in real-time.
      * **Blockchain Interaction**: The Ethereum blockchain is used as it supports smart contract execution and has a robust community and tools like Truffle and Ganache for local testing and deployment.
        + **Public vs. Private Blockchain**: The system may use a private blockchain (if designed for a specific organization or country) or a public blockchain (if aiming for global access and transparency).

###### Step 4: Frontend Development

The user interface is one of the most important parts of the system, as it ensures that voters can easily participate in the election process.

* + - * **Voter Registration**: A simple, user-friendly interface for registering voters, which includes features like:
        + **User Authentication**: Integration with authentication mechanisms such as multi-factor authentication (MFA) or biometric verification to ensure the validity of the voter.
        + **Identity Verification**: Secure collection of documents (e.g., passport, ID) to verify eligibility.
      * **Voting Interface**: A minimal, clean, and accessible interface that allows voters to cast their vote anonymously. The system should include features like:
        + **Language Preferences**: Voters may have the option to choose their preferred language for ease of use.
        + **Vote Confirmation**: Once the voter has cast their vote, they should receive a confirmation message indicating that the vote has been successfully recorded on the blockchain.
      * **Real-time Results Dashboard**: A real-time view of voting results that shows how many votes have been cast, who is leading, etc., without revealing individual voter data.

###### Step 5: Security Implementation

Security is critical to the success of a decentralized voting system, especially in terms of data integrity and voter anonymity.

* + - * **Data Encryption**: All personal information and votes are encrypted using asymmetric encryption techniques, ensuring that only authorized parties (e.g., election administrators) can access sensitive data.
      * **Decentralized Authentication**: Multi-factor authentication or digital signatures are implemented to verify the identity of voters.
      * **Secure Communication**: All data transmitted between the frontend, backend, and blockchain is encrypted using SSL/TLS protocols to prevent man-in-the-middle (MITM) attacks.
      * **Anonymity and Privacy**: Voter identities are not stored on the blockchain. Instead, encrypted keys are used to ensure that votes are anonymous but still verifiable.

###### Step 6: Backend Development and Integration

The backend connects the frontend with the blockchain network and handles all non-transactional logic. It manages:

* + - * **Voter Management**: Handles the registration of voters, validation of their identities, and assignment of digital IDs using the blockchain.
      * **Blockchain Interaction**: The backend interacts with the blockchain via Web3.js, which facilitates interaction with Ethereum smart contracts.
      * **Database Management**: If additional data (such as metadata or records) is required, it is stored in a separate, off-chain database (e.g., a SQL or NoSQL database) with strict security protocols.

###### Step 7: Testing and Quality Assurance

The testing phase ensures that the voting system is secure, user-friendly, and performs as expected:

* + - * **Unit Testing**: Testing individual components, such as smart contracts and frontend components, to ensure that each part works correctly.
      * **Integration Testing**: Testing how well the system’s various components work together. For example, ensuring that when a voter submits their vote, it is correctly stored on the blockchain.
      * **Security Audits**: Conducting regular security audits to detect vulnerabilities, such as potential hacking attempts, data leakage, or smart contract bugs.
      * **User Acceptance Testing**: Allowing a small group of test users (e.g., election officials) to interact with the system in a simulated election environment to identify usability issues.

###### Step 8: Deployment and Monitoring

* + - * **Blockchain Deployment**: Deploying the smart contracts on the Ethereum network. If it’s a private blockchain, deployment is done on a private Ethereum network.
      * **Cloud Hosting**: Hosting the frontend and backend on cloud platforms like AWS or Azure ensures the system can scale and remain available throughout the election process.
      * **Monitoring**: Real-time monitoring tools track the health and performance of the system, including transaction speeds, blockchain status, and any potential vulnerabilities.

###### Step 9: Training and Support

* + - * **Training**: Providing training for election administrators and voters. This could involve:
        + **Admin Training**: How to monitor the election, view results, and manage voter registration.
        + **Voter Training**: Explaining the process of registration and voting, how to verify votes, and how to use the platform.
      * **Ongoing Support**: Offering technical support during the election period to address any issues or concerns from users.

# CHAPTER 6

## SYSTEM DESIGN

### Requirement Analysis

In order to effectively design and develop a system, it is important to understand and document the requirements of the system. The process of gathering and documenting the requirements of a system is known as requirement analysis. It helps to identify the goals of the system, the stakeholders and the constraints within which the system will be developed. The requirements serve as a blueprint for the development of the system and provide a reference point for testing and validation.

### Hardware Requirements

* + - * Processor – 2 GHz or more
      * RAM – 4 GB or more
      * Disk Space – 100 GB or more

### Software Requirements

* + - * Node.js (version – 18.14.0)
      * Web3.js (version – 1.8.2)
      * Truffle (version – 5.7.6)
      * Solidity (version – 0.5.16)
      * Ganache (version – 7.7.3)
      * Metamask
      * Python (version – 3.9)
      * FastAPI
      * MySQL Database (port – 3306)

The system design for a blockchain-based voting system requires careful planning and consideration to ensure it meets the essential criteria of security, transparency, and accessibility. Traditional voting systems have often been limited by issues of transparency, susceptibility to tampering, and difficulties in voter authentication. A blockchain-based voting platform offers solutions to these problems by utilizing the immutability, decentralization, and encryption features inherent in blockchain technology. This essay explores the key components and system design principles that would make a blockchain-based voting system effective, secure, and user- friendly.

1. System Architecture

The system architecture of the blockchain-based voting system is built on a multi- tier structure that includes a **user interface layer**, **backend layer**, and the **blockchain layer**. Each layer plays a vital role in delivering the voting functionalities while maintaining security and transparency.

* + **User Interface Layer**: This layer is responsible for the visual and interactive aspects of the voting system. Built with front-end frameworks like HTML, CSS, and JavaScript (or React for more dynamic experiences), it provides voters with an interface to register, authenticate, cast votes, and view results. The user interface also includes accessibility features, allowing a broad range of voters to interact with the system easily.
  + **Backend Layer**: This layer connects the user interface with the blockchain network. It handles all data processing, user authentication, and business logic needed to ensure secure and smooth operations. Node.js is a suitable choice for the backend in this design due to its asynchronous handling of data, which is beneficial for real-time systems. The backend also connects to databases for storing non-sensitive data (like voter registration status) and integrates with the blockchain for vote transactions.
  + **Blockchain Layer**: This layer is the core of the voting system, handling the secure recording of votes and ensuring that all data is immutable and transparent. A private Ethereum blockchain is deployed, enabling all votes to be stored on-chain. Smart contracts written in Solidity are used to create the voting protocols, enforce voting rules, and automate the counting and tallying of votes. This decentralized ledger is accessible to all stakeholders, allowing public verification without revealing individual voter details.

1. Key System Components
   * **Smart Contracts**: Smart contracts are central to the voting process. They are self-executing codes on the blockchain that define the rules of the election, ensuring each voter casts only one vote. Smart contracts also allow for automatic vote counting and results tallying, reducing the chance of human error and manipulation.
   * **Voter Authentication Module**: To prevent fraud, an authentication module is crucial for verifying voter identity. This module leverages multi-factor authentication (MFA) or biometric verification to ensure each voter is legitimate. Once verified, voters receive a unique digital ID linked to their blockchain wallet, enabling anonymous vote casting without exposing individual identities.
   * **Vote Casting Module**: This module provides an interface and mechanisms for casting votes securely. Upon voter authentication, each vote is recorded as a transaction on the blockchain, encrypted to ensure privacy. Since blockchain is immutable, each vote is stored permanently, guaranteeing no changes or deletions post-casting.
   * **Real-Time Results Dashboard**: The results dashboard displays ongoing vote counts and election results. Due to the transparency of blockchain, real- time tallies are visible to authorized users, allowing them to monitor the election's progress. However, personal data remains secure due to the cryptographic nature of blockchain, which prevents unauthorized access to individual voting patterns.
   * **Admin Control Panel**: Election administrators use this control panel to monitor the system, manage voter registrations, and oversee the voting process. Although the panel grants administrative access, it restricts permissions, ensuring that administrators cannot modify or delete vote records. This design maintains the integrity and impartiality of the system.
2. Data Security and Privacy

Data security and voter privacy are paramount in a voting system. The blockchain- based voting system employs multiple layers of encryption, with each vote transaction being encrypted before being recorded on the blockchain. By using public/private key pairs, voters can cast their votes without revealing personal data, thus ensuring anonymity. The use of decentralized storage also means that no single entity has control over the data, significantly enhancing security by preventing unauthorized modifications.

1. Testing and Validation

Testing is essential to ensure that the blockchain voting system functions correctly, securely, and reliably. Smart contract testing is particularly critical, as once deployed, these contracts are immutable and cannot be altered. Tests include checking the correct implementation of voting rules, ensuring accurate counting, and validating that no voter can cast more than one vote. Usability tests are also conducted to ensure that the system is accessible and user-friendly, even for those unfamiliar with blockchain technology.

1. Deployment and Maintenance

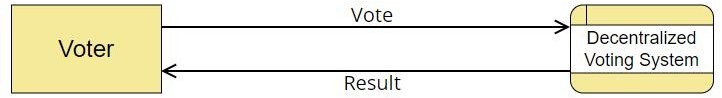
Deployment involves setting up the system on a private blockchain network with tools like Truffle and Ganache. MetaMask is integrated as a blockchain wallet to facilitate user authentication and interaction with the Ethereum blockchain. Post- deployment, routine audits and monitoring are necessary to ensure that the system remains secure, particularly as new security vulnerabilities and blockchain improvements emerge.

1. Challenges and Future Enhancements

A blockchain-based voting system faces certain challenges, including scalability, potential network delays, and user adoption barriers due to the relative complexity of blockchain. Solutions such as optimizing the blockchain protocol for faster transactions, user education on system use, and simplifying the user interface can help overcome these issues. Future enhancements may also include exploring interoperability with public blockchains for additional security, implementing more advanced cryptographic techniques, and integrating artificial intelligence for real- time monitoring of system health and voter turnout trends.

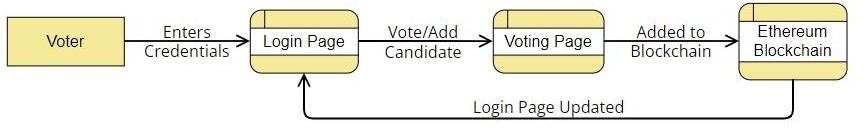
### Data Flow Diagram

* + - **Level 0 data flow diagram**



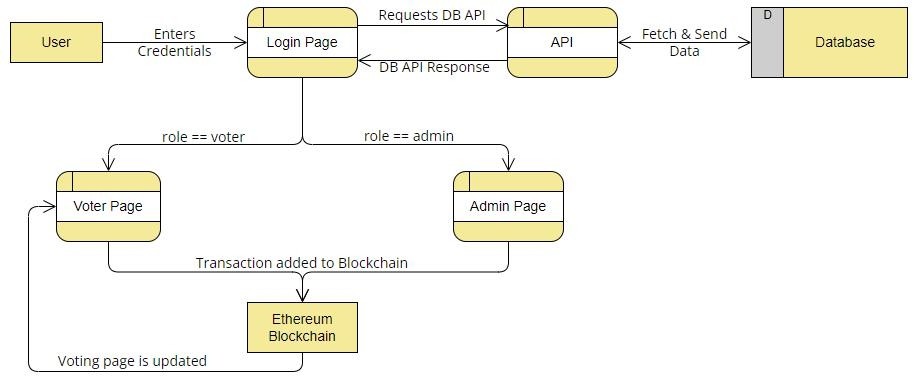
**Figure 2** Level 0 Data Flow Diagram

### Level 1 data flow diagram



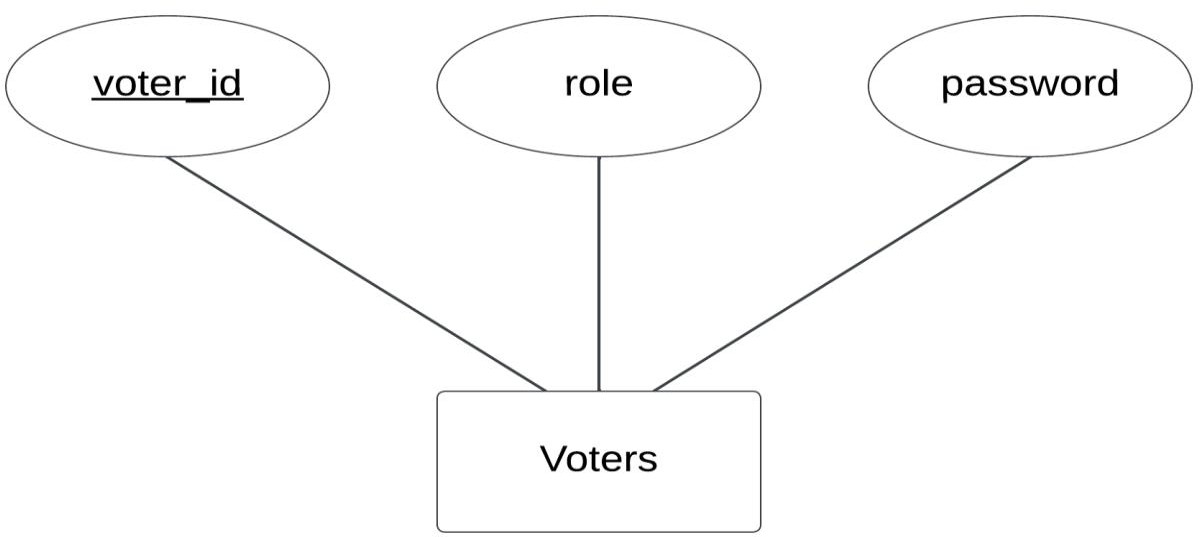
**Figure 3** Level 1 Data Flow Diagram

* Level 2 data flow diagram



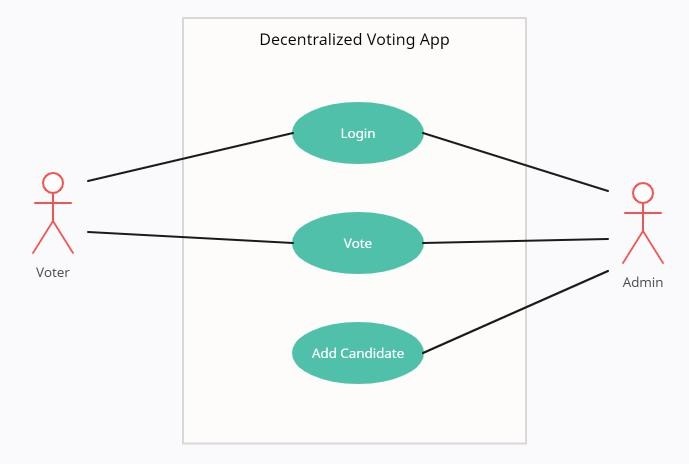
**Figure 4** Level 2 Data Flow Diagram

* 1. ER Diagram



**Figure 5** ER Diagram

6.5 Use Case Diagram



**Figure 6** Use Case Diagram

# CHAPTER 7

## CODE

#### Migrations.sol

pragma solidity ^0.5.15; contract Migrations {

address public owner;

uint public last\_completed\_migration;

modifier restricted() {

require(msg.sender == owner, "Access restricted to owner"); \_;

}

constructor() public { owner = msg.sender;

}

function setCompleted(uint completed) public restricted { last\_completed\_migration = completed;

}

function upgrade(address new\_address) public restricted { Migrations upgraded = Migrations(new\_address); upgraded.setCompleted(last\_completed\_migration)

;

}

}

#### Voting.sol

pragma solidity ^0.5.15; contract Voting {

struct Candidate {

uint id; string name; string party;

uint voteCount;

}

mapping (uint => Candidate) public candidates; mapping (address => bool) public voters;

uint public countCandidates; uint256 public votingEnd; uint256 public votingStart;

function addCandidate(string memory name, string memory party) public returns(uint) {

countCandidates ++; candidates[countCandidates] =

Candidate(countCandidates, name, party, 0);

return countCandidates;

}

function vote(uint candidateID) public { require((votingStart <= now) && (votingEnd > now));

require(candidateID > 0 && candidateID <= countCandidates);

//daha önce oy kullanmamıs olmalı require(!voters[msg.sender]);

voters[msg.sender] = true;

candidates[candidateID].voteCount ++;

}

function checkVote() public view returns(bool){ return voters[msg.sender];

}

function getCountCandidates() public view returns(uint) { return countCandidates;

}

function getCandidate(uint candidateID) public view returns (uint,string memory, string memory,uint)

{

return

(candidateID,candidates[candidateID].name,candidates[candidateID].party

,

candidates[candidateID].voteCount);

}

function setDates(uint256 \_startDate, uint256 \_endDate) public{ require((votingEnd == 0) && (votingStart == 0) && (\_startDate +

1000000 > now) && (\_endDate > \_startDate)); votingEnd = \_endDate;

votingStart = \_startDate;

}

function getDates() public view returns (uint256,uint256) { return (votingStart,votingEnd);

}

}

#### App.js

const Web3 = require('web3');

const contract = require('@truffle/contract');

const votingArtifacts = require('../../build/contracts/Voting.json'); var VotingContract = contract(votingArtifacts)

window.App = { eventStart: function() {

window.ethereum.request({ method: 'eth\_requestAccounts' }); VotingContract.setProvider(window.ethereum) VotingContract.defaults({from:

window.ethereum.selectedAddress,gas:6654755})

// Load account data App.account =

window.ethereum.selectedAddress; $

("#accountAddress").html("Your Account:

" +

window.ethereum.selectedAddress); VotingContract.deployed().then(function(instance){ instance.getCountCandidates().then(function(countCandidat es){

$(document).ready(function(){

$('#addCandidate').click(function() { var nameCandidate = $('#name').val(); var partyCandidate = $

('#party').val();

instance.addCandidate(nameCandidate,partyCandidate).th e n(function(result){ })

});

$('#addDate').click(function(){ var startDate =

Date.parse(document.getElementById("startDate").value)/1000;

var endDate

 Date.parse(document.getElementById("endDate").value)/1000;

instance.setDates(startDate,endDate).then(function(rsl

t){

console.log("tarihler verildi");

});

});

instance.getDates().then(function(result){

var startDate = new Date(result[0]\*1000); var endDate = new Date(result[1]\*1000);

$("#dates").text( startDate.toDateString(("#DD#/#MM#/#YYYY#")) + " - " + endDate.toDateString("#DD#/#MM#/#YYYY#"));

}).catch(function(err){ console.error("ERROR! " + err.message)

});

});

for (var i = 0; i < countCandidates; i++ )

{ instance.getCandidate(i+1).then(functio n(data){

var id = data[0]; var name = data[1]; var party = data[2];

var voteCount = data[3];

var viewCandidates = `<tr><td> <input class="form-check-input" type="radio" name="candidate" value="${id}" id=${id}>` + name + "</td><td>" + party + "</td><td>" + voteCount + "</td></tr>"

$("#boxCandidate").append(viewCandidates)

})

}

window.countCandidates = countCandidates });

instance.checkVote().then(function (voted) { console.log(voted); if(!voted){

$("#voteButton").attr("disabled", false);

}

});

}).catch(function(err){ console.error("ERROR! " + err.message)

})

},

vote: function() { var candidateID = $

("input[name='candidate']:checked").val(); if (!

candidateID) {

$("#msg").html("<p>Please vote for a candidate.</p>") return

}

VotingContract.deployed().then(function(instance){

instance.vote(parseInt(candidateID)).then(function(result){ $ ("#voteButton").attr("disabled", true); $ ("#msg").html("<p>Voted</p>");

window.location.reload(1);

})

}).catch(function(err){ console.error("ERROR! " + err.message)

})

}

}

window.addEventListener("load", function() { if (typeof web3 !== "undefined") {

console.warn("Using web3 detected from external source like Metamask")

window.eth = new Web3(window.ethereum)

} else {

console.warn("No web3 detected. Falling back to http://localhost:9545. You should remove this fallback when you deploy

live, as it's inherently insecure. Consider switching to Metamask for

deployment. More info here: [http://truffleframework.com/tutorials/truffle-and-metamask"](http://truffleframework.com/tutorials/truffle-and-metamask))

window.eth = new Web3(new Web3.providers.HttpProvider("http://127.0.0.1:9545"))

}

window.App.eventStart()

})

#### Login.js

const loginForm = document.getElementById('loginForm'); loginForm.addEventListener('submit',

(event) => { event.preventDefault();

const voter\_id = document.getElementById('voter- id').value; const password = document.getElementById('password').value; const token = voter\_id;

const headers = { 'method': "GET",

'Authorization': `Bearer ${token}`,

};

fetch(`http://127.0.0.1:8000/login?voter\_id=$

{voter\_id}&password=${pas sword}`, { headers })

.then(response => { if (response.ok) {

return response.json();

} else {

throw new Error('Login failed');

}

})

.then(data => {

if (data.role === 'admin') { console.log(data.role)

localStorage.setItem('jwtTokenAdmin', data.token);

window.location.replace(`http://127.0.0.1:8080/admin.html?Authoriz ation=Bearer ${localStorage.getItem('jwtTokenAdmin')}`);

} else if (data.role === 'user'){ localStorage.setItem('jwtTokenVoter', data.token);

window.location.replace(`http://127.0.0.1:8080/index.html?Authoriz ation=Bearer ${localStorage.getItem('jwtTokenVoter')}`);

}

})

.catch(error => { console.error('Login failed:',

error.message); });

});

#### Main.py

 Import required modules import dotenv import os

import mysql.connector

from fastapi import FastAPI, HTTPException, status, Request from fastapi.middleware.cors import CORSMiddleware

from fastapi.encoders import jsonable\_encoder from mysql.connector import errorcode import jwt

 Loading the environment variables dotenv.load\_dotenv()

 Initialize the todoapi app app = FastAPI()

 Define the allowed origins for CORS origins = [

"http:// localhost:8080", "http://127.0.0.1:80 80",

]

 Add CORS middleware app.add\_middleware(

CORSMiddleware, allow\_origins=origins, allow\_credentials=True, allow\_methods=["\*"], allow\_headers=["\*"],

)

 Connect to the MySQL database try:

cnx = mysql.connector.connect( user=os.environ['MYSQL\_USER'], password=os.environ['MYSQL\_PASSWORD'], host=os.environ['MYSQL\_HOST'], database=os.environ['MYSQL\_DB'],

)

cursor = cnx.cursor()

except mysql.connector.Error as err:

if err.errno == errorcode.ER\_ACCESS\_DENIED\_ERROR: print("Something is wrong with your user name or password")

elif err.errno == errorcode.ER\_BAD\_DB\_ERROR: print("Database does not exist")

else:

print(err)

 Define the authentication middleware async def authenticate(request: Request):

try:

api\_key = request.headers.get('authorization').replace("Bearer

", "")

cursor.execute("SELECT \* FROM voters WHERE

voter\_id = %s", (api\_key,))

if api\_key not in [row[0] for row in cursor.fetchall()]:

raise HTTPException(

status\_code=status.HTTP\_401\_UNAUTHORIZED

, detail="Forbidden"

)

except:

raise HTTPException( status\_code=status.HTTP\_401\_UNAUTHORIZED, detail="Forbidden"

)

 Define the POST endpoint for login @app.get("/login")

async def login(request: Request, voter\_id: str, password: str): await authenticate(request) role = await get\_role(voter\_id, password)

# Assuming authentication is successful, generate a token token = jwt.encode({'password': password, 'voter\_id': voter\_id,

'role': role}, os.environ['SECRET\_KEY'], algorithm='HS256') return {'token': token, 'role': role}

 Replace 'admin' with the actual role based on authentication async def get\_role(voter\_id, password):

try:

cursor.execute("SELECT role FROM voters WHERE voter\_id

= %s AND password = %s", (voter\_id, password,))

role = cursor.fetchone() if role:

return role[0] else:

raise HTTPException( status\_code=status.HTTP\_401\_UNAUTHORIZED, detail="Invalid voter id or password" except mysql.connector.Error as err:

print(err)

raise HTTPException( status\_code=status.HTTP\_500\_INTERNAL\_SERVER\_ERROR, detail="Database error"

)

#### Package.json

{

"name": "decentralized-voting", "version": "1.0.0",

"description": "",

"main": "index.js", "scripts": {

"test": "echo \"Error: no test specified\" && exit 1"

},

"author": "",

"license": "ISC", "dependencies": {

"jsonwebtoken": "^9.0.0", "@truffle/contract": "^4.6.18", "browserify": "^17.0.0",

"dotenv": "^16.0.3",

"express": "^4.18.2",

"web3": "^1.9.0"

}

}

# CHAPTER 8

## OUTPUT

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. It includes a set of techniques and methods to identify defects, bugs, performance issues and providing a reliable and quality product. The goal is to identify issues as early as possible and improve the overall quality of the system.

### Types of Testing

* + 1. **Unit Testing**

Unit testing is a type of testing that is used to evaluate the individual units or components of a software system. This type of testing helps ensure that each unit or component of the system is working correctly and is able to perform its intended function.

### Integration Testing

Integration testing is a type of testing that is used to evaluate how well the different units or components of a software system work together. This type of testing helps to identify and resolve issues related to compatibility, performance, and data flow between the different units or components.

### Functional Testing

Functional testing is a type of testing that is used to evaluate how well a system or software performs the specific functions or tasks that it is designed to perform. It is done by testing the system or software with various inputs and verifying that the outputs are correct. This type of

testing ensures that the system or software is working as intended and is able to perform the functions it was designed to perform.

### White Box Testing

White box testing, also known as structural testing or glass-box testing, is a type of testing that examines the internal structure and implementation of a software system. It involves testing the code itself and checking that it is functioning correctly and adhering to coding standards. This type of testing helps to identify and resolve issues related to logic, control flow, and data structures within the system.

### 8.1.4 Black Box Testing

Black box testing, also known as functional testing, is a type of testing that examines the external behavior and interfaces of a software system. It involves testing the system from the user's perspective, without looking at the internal structure or implementation, and checking that it is functioning correctly and meeting the requirements. This type of testing helps to identify and resolve issues related to usability, compatibility, and performance.

* 1. **Test Results**
     1. **Test Case 1**

|  |  |
| --- | --- |
| **Test Case No.** | 1 |
| **Test Type** | Unit Test |
| **Name of Test** | Checking JWT Authorization |
| **Test Case Description** | The objective of this test case is to check jwt authorization. |
| **Input** | Login and Password |
| **Expected Output** | User should not be able to login without proper authorization. |
| **Actual Output** | User cannot access voting or admin page without authorization. |
| **Result** | Pass |
| **Comments** | Working properly. |

* + 1. **Test Case 2**

|  |  |
| --- | --- |
| **Test Case No.** | 2 |
| **Test Type** | Functional Test |
| **Name of Test** | Verify user login |
| **Test Case Description** | The objective of this test case is to verify that user can login to the  voting portal. |
| **Input** | Voter\_id and password |
| **Expected Output** | User must be able to login if credentials match the database, else  unauthorized error is shown. |
| **Actual Output** | User is able to login with correct credentials only. |
| **Result** | Pass |
| **Comments** | Working properly. |

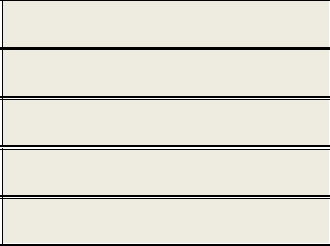
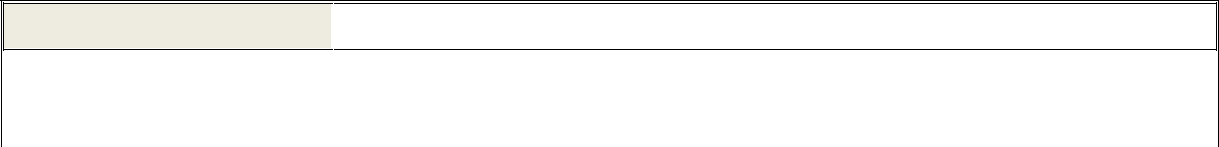
* + 1. **Test Case 3**

|  |  |
| --- | --- |
| **Test Case No.** | 3 |
| **Test Type** | Unit Test |
| **Name of Test** | Verify candidate registration |
| **Test Case Description** | The objective of this test case is to verify that candidate can be  registered by admin. |
| **Input** | Candidate name and party. |
| **Expected Output** | Registration transaction should be successful. |
| **Actual Output** | Registration transaction is successful. |
| **Result** | Pass |
| **Comments** | Working properly. |

* + 1. **Test Case 4**

|  |  |
| --- | --- |
| **Test Case No.** | 4 |
| **Test Type** | Unit Test |
| **Name of Test** | Verify date registration |
| **Test Case Description** | The objective of this test case is to verify that date of voting can  be specified by admin. |
| **Input** | Starting and ending date |
| **Expected Output** | Date transaction should be successful. |
| **Actual Output** | Date transaction is successful. |
| **Result** | Pass |
| **Comments** | Working properly. |

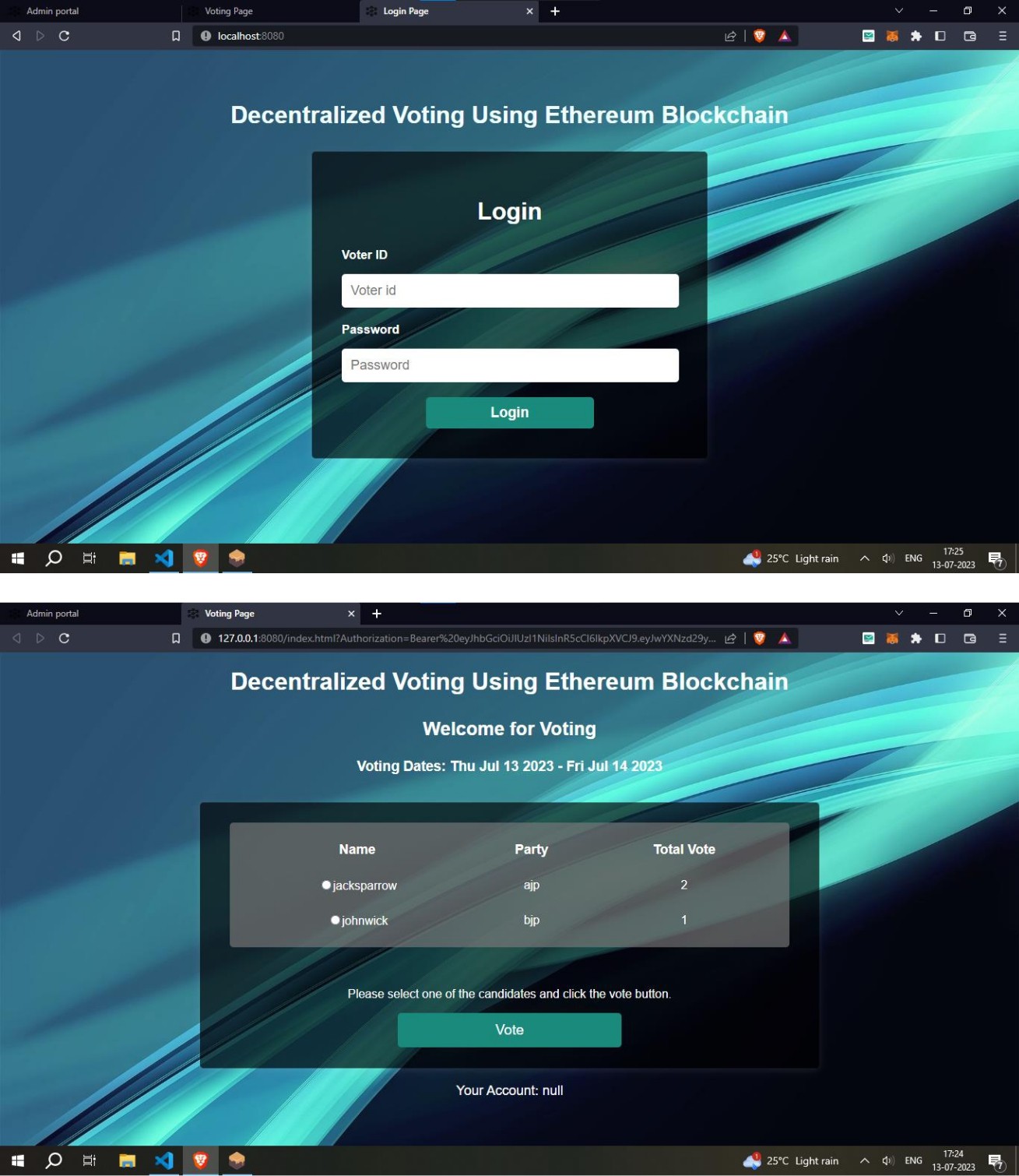
* + 1. **Test Case 5**



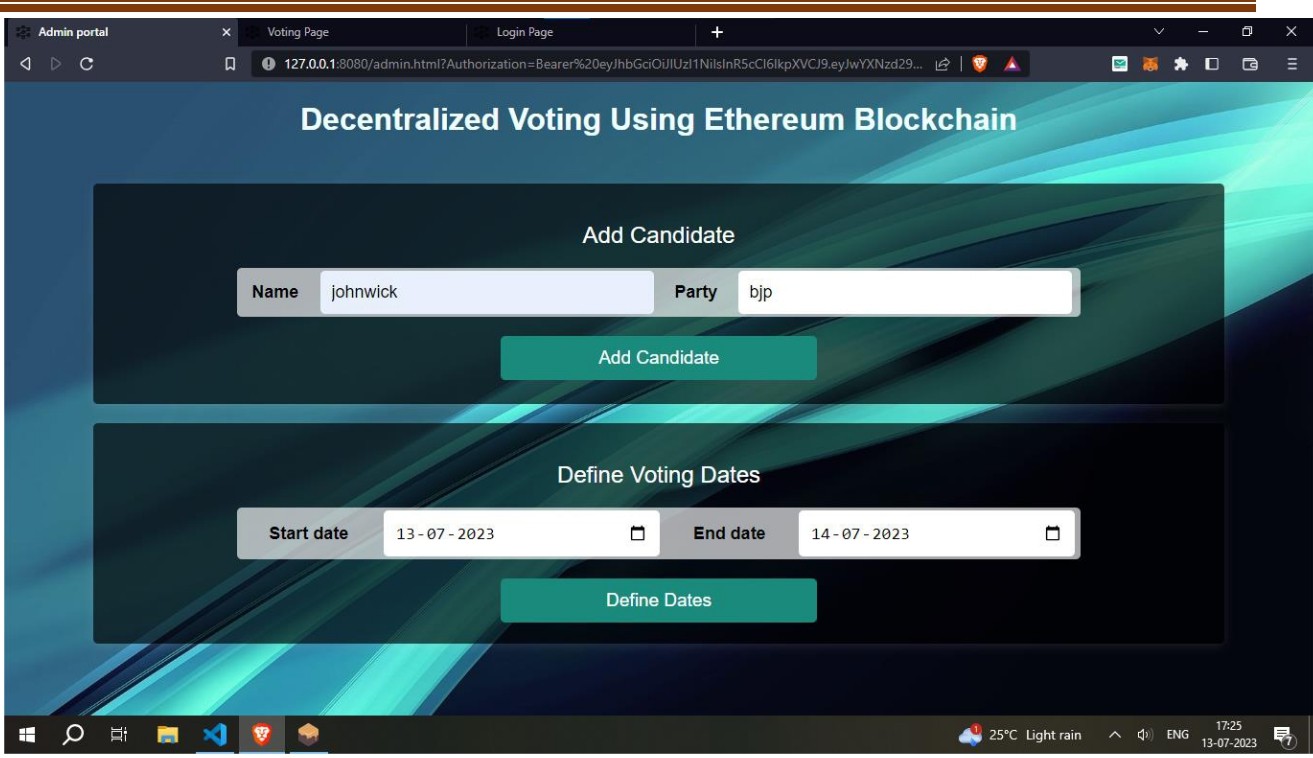
**Test Case No.**

5

|  |  |
| --- | --- |
| **Test Type**  **Name of Test** | Functional Test |
| Verify voting |
| **Test Case Description** | The objective of this test case is to verify that voter is able to cast their vote. |
| **Input**  **Expected Output Actual Output Result Comments** | Select a candidate and click “Vote” button. |
| Vote transaction should be successful. |
| Vote transaction is successful. |
| Pass |
| Working properly. |







##### CHAPTER – 5 CONCLUSION & FUTURE SCOPE

The blockchain-based voting system provides a revolutionary approach to modernize electoral processes by addressing key challenges faced by traditional voting systems. Through the use of decentralized technology, the system ensures transparency, security, and immutability of votes, which helps in eliminating issues such as vote tampering, fraud, and lack of voter trust. By incorporating advanced features such as secure voter registration, encrypted vote casting, real-time result tallying, and a transparent audit trail, the system enhances the overall integrity and accessibility of elections.

Moreover, the system allows for greater voter participation by enabling remote voting, which can be particularly beneficial in regions with limited access to polling stations. Voters can securely cast their votes from anywhere in the world, which can lead to increased turnout and a more inclusive electoral process. With the integration of blockchain, this voting system eliminates the need for intermediaries, reducing the potential for manipulation by centralized authorities and ensuring a fair election outcome.

The development of this blockchain-based voting system serves as a significant step towards building a more democratic, efficient, and transparent electoral infrastructure. The use of Ethereum’s blockchain and smart contracts guarantees a tamper-proof and verifiable record of every vote, fostering trust in the electoral process. This system not only benefits voters and election administrators but also contributes to the integrity and fairness of elections on a global scale.

Future Scope

While the blockchain-based voting system represents a major advancement, there are several areas for future growth and improvement to further enhance its effectiveness and reach. Below are key areas where this system can evolve:

* + - 1. **Scalability for Larger Elections:** One of the future challenges for blockchain voting systems is handling a large number of transactions in real- time during national or global elections. As blockchain networks, especially public ones like Ethereum, can face scalability issues (e.g., transaction fees and delays during high traffic), future versions of the system may incorporate more scalable blockchain solutions such as Layer 2 solutions (e.g., Optimistic Rollups) or alternative blockchain platforms designed for high throughput.
      2. **Cross-Blockchain Interoperability:** In the future, the ability for different blockchain networks to communicate and share data could become crucial. As more countries or organizations adopt blockchain for elections, there may be a need for cross-chain interoperability, which would allow voters to cast votes on different blockchain networks seamlessly, increasing the system's flexibility and reach.
      3. **AI Integration for Fraud Detection:** While blockchain guarantees the integrity of votes, integrating Artificial Intelligence (AI) could enhance security further by detecting fraudulent activities such as identity theft or anomalies in voting patterns. AI algorithms could analyze trends in voter data and flag suspicious activities for investigation, offering an additional layer of protection against electoral fraud.
      4. **Incorporation of Biometric Authentication:** To further enhance voter security, biometric authentication (e.g., facial recognition or fingerprint scanning) could be integrated into the registration and voting process. This would ensure that only eligible voters can participate and prevent identity theft, adding another layer of trust and reliability to the system.
      5. **Integration with Digital Identity Systems:** Blockchain-based voting could be integrated with national digital identity systems, which would provide a seamless and secure mechanism for voter registration and authentication. This could lead to faster and more efficient voter verification processes while ensuring that only legitimate, verified individuals can vote.
      6. **Voting Accessibility for Disabled and Remote Voters:** The future scope of this system includes improving accessibility for people with disabilities. Features such as voice recognition, text-to-speech, and other assistive

technologies could be incorporated into the platform to make the voting process more inclusive. Additionally, outreach programs and better interfaces can be developed to help remote voters in underserved regions access the platform.

* + - 1. **Privacy Enhancements:** Privacy concerns, especially related to voter anonymity, will remain a priority. Future versions of the system could utilize advanced cryptographic techniques, such as zero-knowledge proofs (ZKPs), which allow data to be verified without revealing the data itself, thereby enhancing both security and privacy.
      2. **Global Adoption and Regulation:** For blockchain-based voting to become widely adopted, clear legal and regulatory frameworks will be necessary. Future efforts should focus on working with governments and international organizations to establish regulatory standards for blockchain voting systems. This will ensure uniformity in its implementation and foster global trust in the system.
      3. **Blockchain for Multi-Level Elections:** Expanding the system to handle multi-level elections, such as local, regional, and national elections, with the ability to support multiple types of voting methods (e.g., proportional representation, ranked-choice voting) could make the platform even more versatile and adaptable to different political systems.
      4. **Environmental Sustainability:** As blockchain technology grows, so does the energy consumption of blockchain networks, particularly in proof-of- work (PoW) systems. The future scope of this project may include transitioning to energy-efficient consensus algorithms like proof-of-stake (PoS) to ensure that the environmental impact of using blockchain for voting remains minimal.

**Post-Vote Transparency and Audit Trails:** As blockchain technology inherently supports transparency, one significant area for future improvement is enhancing post-vote transparency. The system could be further developed to provide an immutable audit trail for votes after they are cast. This would allow any voter or stakeholder to verify, at any point in time, that their vote was accurately cast, counted, and recorded on the blockchain. It would also make election results tamper-proof, ensuring confidence in the outcome.

* + - 1. **Enhanced Voter Education and Adoption:** For the adoption of blockchain- based voting to be widespread, comprehensive voter education campaigns are essential. Governments and organizations may need to invest in educating the public about how blockchain voting works, its benefits, and how to use the system securely. This would help demystify the technology and increase its trustworthiness among voters.
      2. **Integration with Smart Cities and E-Government Services:** As cities and governments become increasingly digital, blockchain voting could be integrated into the broader context of e-governance. A blockchain voting platform could eventually become part of a larger smart city infrastructure, where citizens could participate in various decision-making processes—from local elections to community-based initiatives—securely and transparently.
      3. **Global Standardization and Compliance:** Blockchain voting systems, while promising, need to meet international standards and ensure compliance with electoral laws worldwide. Efforts to create a unified set of rules and guidelines will be crucial for enabling cross-border use of blockchain-based voting systems, particularly for expatriates and overseas citizens. Collaboration with international organizations like the United Nations and the European Union could help foster this global compliance.
      4. **Evolution of Governance Models:** The future development of blockchain voting may also include the evolution of governance models within decentralized voting platforms. Future blockchain-based systems might allow for decentralized autonomous organizations (DAOs) to facilitate voting, meaning that even election governance could be automated and decentralized, removing the need for central authorities altogether. This would ensure fairness, neutrality, and autonomy in voting processes.
      5. **Multi-Layered Security Frameworks:** As digital and cyber threats continue to evolve, multi-layered security frameworks will become essential for blockchain-based voting systems. The future scope might include the integration of advanced techniques such as biometric identification, multi- factor authentication (MFA), encryption algorithms, and cryptographic protocols like homomorphic encryption to provide an extra layer of security and prevent unauthorized access to the system.
      6. **Support for Various Voting Systems and Election Types:** Blockchain voting systems should evolve to support different electoral systems globally. This might include ranked-choice voting, proportional representation, and other systems used in different countries. A flexible blockchain

infrastructure could allow seamless integration of various voting methods to accommodate different types of elections.

* + - 1. **Blockchain Interoperability with Other Sectors:** As blockchain is increasingly used across various sectors, future blockchain-based voting systems could interact with other industries like healthcare, finance, and identity management. For example, integrating blockchain voting with secure medical records could enable citizens to verify their identities without a central database, offering a higher level of security for voting and participation in other services.
      2. **Proof-of-Existence for Historical Records:** Future blockchain voting systems could include a "proof-of-existence" feature for historical votes, where citizens can digitally prove that they participated in an election or referendum. This could be used for applications beyond voting, such as ensuring eligibility for certain government services or entitlements.
      3. **Integration with Social Media Platforms:** Blockchain-based voting could also integrate with social media platforms to encourage more voter engagement and participation. Blockchain tokens or rewards could be used to incentivize users to vote, further driving democratic participation. Social media platforms, when properly integrated with blockchain, can serve as a bridge between traditional forms of communication and the decentralized election process.

Potential Challenges and Areas for Future Development

While the future scope for blockchain-based voting systems is exciting, there are also several challenges that need to be addressed for successful widespread adoption:

* + - * + **Usability and Accessibility:** One of the main barriers to adopting new technology is ensuring that the platform is user-friendly and accessible to everyone, including those who are not tech-savvy. Efforts must be made to create intuitive interfaces and mobile apps that provide a smooth experience for all users, including the elderly or people with disabilities.
        + **Legal and Regulatory Barriers:** As blockchain voting systems are new, regulatory frameworks around the world are yet to be fully developed. Ensuring that these systems comply with existing electoral laws and legal

processes in various countries will be a crucial step for wider implementation.

* + - * + **Ensuring Consensus and Trust in the Network:** Since blockchain-based voting systems are decentralized, ensuring that there is consensus among all nodes (participants) in the network is vital for maintaining trust in the system. Furthermore, if a large portion of the global population is to trust blockchain for elections, there must be confidence that the system cannot be manipulated by bad actors.
        + **Energy Efficiency and Sustainability:** Many blockchain platforms still rely on energy-intensive proof-of-work (PoW) consensus mechanisms, which can be costly and environmentally damaging. Future blockchain voting systems could transition to proof-of-stake (PoS) or other energy-efficient algorithms to address these concerns and make the system more sustainable in the long run

Long-Term Vision

In the long run, blockchain voting systems could become an essential part of the global democratic framework. With increasing trust in decentralized technologies and advancements in blockchain scalability, security, and integration, blockchain- based voting could become the standard for free and fair elections worldwide.

Additionally, the development of user-friendly interfaces and increased voter education would pave the way for the mass adoption of this technology.

This system could eventually allow people to vote in any election, anywhere in the world, using just their mobile phones or computers, ensuring that every eligible individual has access to a transparent and secure voting process. The future of elections may very well be decentralized, with blockchain technology leading the way.

##### REFERENCES

1. Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System.

Retrieved from <https://bitcoin.org/bitcoin.pdf>

1. Atzori, M. (2017). Blockchain-based decentralized voting systems. *Journal of Computer Science and Technology*, 32(5), 987-996.
2. Buterin, V. (2013). A Next-Generation Smart Contract and Decentralized Application Platform. Ethereum Whitepaper. Retrieved from <https://ethereum.org/en/whitepaper/>
3. ​

Chaudhry, A., & Zubair, M. (2024). Decentralized Voting System Using

|  |  |
| --- | --- |
| Blockchain Technology. *SSRN*. Retrieved |  |
| from <https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4483688> | |

1. Kshetri, N. (2017). Blockchain's roles in meeting key supply chain management challenges. *International Journal of Information Management*, 37(6), 181–194.
2. Wüst, K., & Gervais, A. (2018). Do you Need a Blockchain? (Or a Smart Contract?). In *Proceedings of the 2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, 45-52.
3. Srivastava, S., & Vohra, R. (2020). Blockchain Technology for Election System: A Conceptual Model and Its Implementation. *Journal of Information Security and Applications*, 52, 102545.
4. Jaatun, M. G., & Tundal, H. (2020). Blockchain-Based Voting Systems: Challenges and Prospects. In *Proceedings of the International Conference on Distributed Computing and Internet Technology (ICDCIT)*. Springer.
5. Chainlink. (2020). Decentralized Oracles and Blockchain Voting. Retrieved from <https://chain.link/solutions/decentralized-oracles>

|  |  |  |
| --- | --- | --- |
| [10] Election Commission of India. (2022). Remote Voting Chain - PoC - | |  |
| Blockchain Technology. Retrieved |  | |
| from <https://blockchain.gov.in/Home/CaseStudy?CaseStudy=remotevotingchain> | | |