JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

STRATEGIC INTEGRATION OF BLOCKCHAIN TECHNOLOGY TO ESTABLISH A ROBUST AND SECURE E-VOTING SYSTEM

Ayush Abhigyan
Department of Computing Technologies
SRM Institute of Science and Technology, Kattankulathur Chengalpattu, India

Soham Ghosh
Department of Computing Technologies
SRM Institute of Science and Technology, Kattankulathur Chengalpattu, India

Mrs. R. Brindha
Department of Computing Technologies
SRM Institute of Science and Technology, Kattankulathur Chengalpattu, India

***Abstract*— Voting stands as a foundational pillar of democracy, granting citizens the essential right to select leaders and express their voices. Online voting systems, supplanting traditional paper ballots, not only enhance efficiency and minimize errors but also facilitate convenient remote voting while guarding against fraudulent practices, thereby upholding the sanctity of the electoral process. Embracing blockchain technology in electronic voting represents a promising frontier. The decentralized and immutable nature of blockchain holds the potential to alleviate existing concerns in e-voting, including issues related to tampering, hacking, and data security breaches. By leveraging blockchain, we can establish a transparent and trustless system where every vote is securely recorded and verifiable, ensuring the integrity of the electoral process. Our proposal fervently advocates for the implementation of a blockchain-based voting system, poised to combat fraud, streamline procedures, and fortify the overall security and efficiency of the voting process, ultimately strengthening the foundation of democracy and empowering citizens to participate confidently in shaping their collective future.**

Keywords—Blockchain, decentralized, immutable, security, democracy, voting

I. INTRODUCTION

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

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

II. LITERATURE SURVEY

A. *ACB-Vote: Efficient, Flexible, and Privacy-Preserving Blockchain-Based Score Voting With Anonymously Convertible Ballots Wenyi Xue, Yang Yang, Yingjiu Li, Hwee Hwa Pang, Robert H. Deng IEEE Transactions on Information Forensics and Security, VOL. 18, 2023*

The Blockchain-based electronic voting systems, particularly those employing score voting mechanisms, have encountered significant challenges such as high range proof overheads, inefficiency, and insufficient voter anonymity. In response to these issues, this literature survey explores the proposed technique known as ACB-Vote, presented by Wenyi Xue, Yang Yang, Yingjiu Li, Hwee Hwa Pang, and Robert H. Deng in their work published in IEEE Transactions on Information Forensics and Security, Volume 18, 2023. ACB-Vote introduces efficiently convertible ballots as a solution to the identified challenges. The system achieves voting anonymity through the use of BBS+ signatures and signature of knowledge. Leveraging convertibly linkable signatures (CLS), ACB-Vote addresses multiple voting concerns, eliminates range proof overheads, facilitates batch ballot verification, and supports flexible tallying methods. The existing literature on blockchain-based score voting systems reveals a lack of efficiency and robust voter anonymity. ACB-Vote aims to bridge this gap by presenting a secure, flexible, and privacy-preserving score voting system. Notably, the proposed system demonstrates competitive efficiency when compared to previous systems within blockchain environments.

B. E-Voting Meets Blockchain: A Survey Maria-Victoria Vladucu, Ziqian Dong, Jorge Medina, Roberto Rojas-Cessa IEEE Access, VOL. 11, 2023

Electronic voting systems (EVS) have witnessed widespread global adoption for their capacity to facilitate remote voting and streamline the vote counting process. However, concerns persist regarding the security and integrity of these systems, as they are susceptible to manipulation and interference. In response to these challenges, this survey explores the intersection of E-Voting and Blockchain technologies. Blockchain technology, renowned for its immutability, emerges as a promising solution to enhance the robustness of EVS. By leveraging the secure and decentralized ledger inherent in blockchain, votes can be stored in an unalterable manner. This approach aims to mitigate the risk of vote tampering and ensure the legitimacy of elections, fostering trust in the electoral process. Despite the immense promise of blockchain-based EVS, this survey highlights several challenges that require further attention and development. The need to ensure voter anonymity, scalability to accommodate large electorates, and accessibility for all voters represents significant hurdles. Additionally, the potential for cyberattacks and the necessity for robust governance frameworks demand continuous focus from the research community.

C. Conceptual Architecture of a Blockchain Solution for E-Voting in Elections at the University Level Simona-Vasilica Oprea, Adela Bâra, Anca-Ioana Andreescu, Marian Pompiliu Cristescu IEEE Access, VOL. 11, 2023

Electronic voting (e-voting) systems play a pivotal role in modernizing electoral processes, and their security and efficiency are of paramount importance. This paper presents a conceptual architecture designed to secure and streamline e-voting systems, focusing specifically on university-level elections. The proposed conceptual architecture leverages encrypted functions, introduces a two-stage voting and validation process, and incorporates blockchain tables. Through these elements, the system aims to enhance data integrity and safeguard voter privacy. The use of encrypted functions adds an additional layer of security, while the two-stage process ensures a robust validation mechanism. The integration of blockchain tables contributes to the immutability of the system, addressing concerns related to data tampering. Despite the advancements in blockchain-based e-voting systems, challenges persist, particularly in the university-level context. The research highlights the need to address issues related to voter anonymity, scalability, and accessibility. Ensuring voter anonymity is crucial for fostering trust, scalability is essential for accommodating diverse electorates, and accessibility is key to inclusivity. The research community is called upon to address these challenges to further enhance the effectiveness and trustworthiness of blockchain-based e-voting systems.

III. ARCHITECTURE AND ANALYSIS

The architectural diagram acts as a blueprint that displays the system's general design and helps stakeholders understand the system's structure and behavior. It provides a visual representation of the many components, their relationships, and how data flows through the system. By presenting this image, we want to provide a clear and simple portrayal of the system's architecture, enabling better comprehension and communication among project team members and stakeholders.

A. SYSTEM ARCHITECTURE

A system architecture diagram is a visual representation of the structure, components, and interactions of a system. It provides an overview of how different pieces of a system are structured and how they operate together to provide the intended functionality. The diagram often represents the high-level structure of the system, including its primary components, their relationships, and the flow of data or information between them. A system architecture diagram helps stakeholders, including developers, designers, and project managers, comprehend the system's design and visualize its many components. It acts as a communication tool to express the system's structure, interfaces, and behaviour to different stakeholders participating in the project.

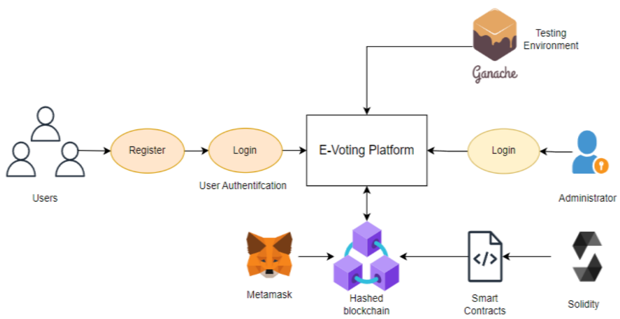

Fig 1: Architecture Diagram

B. Modules

- Frontend
  - Login Page
  - User Module
  - Admin Module
- Backend
  - Database Integration
  - Setting up Truffle suite environment for Smart Contract Development
  - Defining code logic using Solidity
- Using Ganache for setting up testing accounts locally

- Integrating MetaMask for smooth transaction handling
- Testing and Validation

C. Functional Requirements

- User Registration
- Voting Interface
- Result Display
- Smart Contract Implementation
- Authentication and Authorization
- Integration with Ethereum

D. Non Functional Requirements

- Performance
- Security
- Usability
- Reliability
- Scalability
- Compliance

IV. Design and Implementation

The design phase establishes the groundwork for our project, producing a blueprint for its structure and outlining the architecture and relationships between its components. During this phase, significant decisions are made about the choice of technologies and tools, such as the Node.js environment for backend development, MySQL Workbench for database design, Truffle suite with Ganache for testing environment, and FastAPI for constructing APIs. Additionally, the solidity-based blockchain code implementation and integration of Metamask for transaction management are discussed. The design process also covers the assessment of database structure for user and admin validation, ensuring effective data storage and retrieval.

Implementation is the phase where the designed system is developed and put into action. In this phase, the intended system is brought to life through coding and deployment. This involves constructing the backend logic using Node.js, designing and configuring the database using MySQL Workbench, and setting up the testing environment using Truffle suite with Ganache. The blockchain code is implemented using Solidity, and MetaMask interaction is facilitated for seamless transaction management. The FastAPI framework is leveraged to construct APIs for interacting with the system. Furthermore, performance criteria are defined to measure the system's efficiency, and thorough testing and validation are undertaken to assure proper functioning and reliable operation of the system.

A. Frontend Module

The front-end module plays a vital role in designing the user experience of our voting system. It covers the design and development of user interfaces designed to fulfill the demands of both voters and administrators.

a) Login Page

The login page is a critical entrance point into our voting system, providing secure access for both users and administrators. It has input areas for the voter ID and password, enabling individuals to safely input their credentials. Upon submission, the system executes a database query to retrieve the matching user data from the MySQL database, leveraging the voter ID as the main identifier. Subsequently, the recovered user data is compared with the entered login credentials to validate their accuracy, specifically checking the password against the saved password hash. If the credentials match, the user is granted access to the system and forwarded to their particular dashboard based on their role. The login page surely offers a secure and seamless authentication process, ensuring the integrity and confidentiality of user data within the voting system.

b) User Module

For the user module, a user-centric approach is employed to design interfaces that are intuitive, informative, and conducive to seamless interaction. A dynamic dashboard acts as the centerpiece, giving users with extensive insights about participating parties and candidates, thereby helping them to make well-informed voting decisions. Additionally, a faster registration process is introduced, allowing users to register accounts safely and conveniently. The voting interface is created with simplicity and clarity in mind, ensuring that users can cast their ballots with ease and confidence. Finally, a dedicated results component allows consumers real-time access to election outcomes, fostering transparency and trust in the electoral process.

c) Admin Module

In contrast, the admin module is intended to address the special demands of administrators entrusted with operating the voting system. Here, the emphasis is on offering comprehensive tools and functions to streamline administrative processes and enable effective governance of the voting process. An appealing dashboard acts as the command center, allowing administrators access to crucial functionality such as candidate administration and election setup. Admins are enabled with the ability to add, update, and delete candidate profiles, providing smooth management of applicant information. Moreover, the election setup functionality enables administrators to select critical criteria such as election start and end dates, ensuring that the voting process complies to predefined deadlines and norms.

B. Backend Module

The Backend modules are the essential building parts of our voting system. These modules cover key components like as the integration of databases, the development of smart contracts, the establishment of a local testing environment, and the management of transactions.

a) Database Integration

In the design of our voting system, the database acts as a core component, responsible for storing and maintaining critical data relating to users, candidates, elections, and vote records. To provide robust and effective data handling, we integrate a MySQL database into our backend infrastructure. MySQL is a widely-used relational database management system known for its dependability, performance, and scalability—making it a great fit for our project's requirements.

The first stage in integrating the database includes defining a complete schema that reflects the structure of our voting system and accommodates the numerous entities and relationships within it. We methodically create the database structure using MySQL Workbench, a powerful visual tool that simplifies database design and modeling. The schema comprises tables for users, candidates, elections, votes, and other related entities, along with the necessary relationships and constraints to guarantee data integrity. For this, we construct a Server Connection and name it as 'voter_db' and fill out the relevant details like Connection Method, Hostname, Port, Username and Password. Then after successful connection formation with the MySQL Server, we start the generate the schema.

voters			
Field	Type	NULL	Key
voter_id	varchar(36)	No	Primary
role	enum('admin','user')	No	
password	varchar(255)	No	

Fig. 2 Database schema

Once the structure is finalized, we proceed to populate the database with initial data, including user ID and passwords. This data population procedure is vital for ensuring that the voting system performs well upon deployment. Additionally, we integrate powerful data management techniques to handle CRUD (Create, Read, Update, Delete) actions safely and quickly. This includes implementing stored procedures, triggers, and indexes to optimize data retrieval and manipulation.

The MySQL database is seamlessly integrated with the backend logic of our voting system, facilitating efficient communication and data interchange between frontend and backend components. Through standardized SQL queries and prepared statements, the backend interacts with the database to retrieve, update, and modify data necessary for the application's functionality. This integration ensures reliable and real-time execution of user operations such as registration, voting, and result retrieval, thereby enhancing overall user experience.

Ensuring data security is paramount in our voting system, and we employ robust measures to safeguard sensitive information stored in the database. Critical data, including passwords, is encrypted, access control methods are implemented to prevent unauthorized access, and database activity is regularly monitored for potential security breaches. Additionally, robust backup mechanisms are in place to mitigate data loss and ensure data availability in case of system failures or disasters.

In anticipation of managing a potentially large user base and dataset, our database design incorporates scalability and performance optimization strategies. This includes partitioning tables, optimizing query performance through indexing and query optimization techniques, and implementing caching mechanisms to alleviate database load. These measures ensure that the voting system can seamlessly scale with growing user demands and maintain optimal performance under varying load conditions.

b) Setting up Truffle Suite environment for Smart Contract Development

In the area of blockchain technology, the development of smart contracts constitutes the backbone of decentralized apps. To help this key part of our project, we leverage the Truffle Suite—an industry-standard development platform for Ethereum-based apps. This suite contains a comprehensive set of tools and services meant to expedite the smart contract development process. Initially, we set up the Truffle environment, setting it to smoothly interface with our chosen blockchain network. Through Truffle's easy command-line interface (CLI), we initiate new projects, build smart contracts written in Solidity, and deploy them onto the blockchain network. Truffle's built-in testing framework enables us to develop automated tests to validate the reliability and correctness of our smart contracts. Additionally, Truffle's support for migrations streamlines the process of deploying contracts across different environments, supporting seamless transitions between development, testing, and production settings. To set up the Truffle environment in the local machine, it is essential to install Truffle globally using the 'npm install -g truffle' command.

c) Defining code logic using Solidity

Writing Solidity code for smart contracts is a vital component of designing decentralized apps, including our voting mechanism. Solidity is a programming language specifically intended for Ethereum-based smart contracts, enabling developers to specify the logic and behavior of decentralized apps in a secure and deterministic manner. In our project, we leverage Solidity to develop two crucial smart contracts: Migrations.sol and Voting.sol.

The Migrations.sol contract is a standard contract provided by the Truffle framework, functioning as a migration script for installing other contracts onto the Ethereum blockchain. This contract defines a simple migration protocol that tracks the deployment state of contracts and guarantees that migrations are completed in the correct order. In our implementation, the Migrations.sol contract typically has a single function, entitled migrate, which is responsible for installing other contracts or executing other necessary initialization duties throughout the migrating process. By adhering to the norms specified by the Migrations.sol contract, we ensure that our smart contracts are deployed and activated correctly on the Ethereum network.

The Voting.sol contract, on the other hand, constitutes the basic logic of our voting system, describing the rules and procedures for conducting fair and transparent elections. This smart contract comprises several critical functions and data structures that support various phases of the voting process, including voter registration, candidate nomination, ballot casting, and result tabulation. The Voting.sol contract normally provides functions to:

- Register voters and check their eligibility to participate in elections.
- Nominate candidates and manage the candidate list for each election.
- Cast ballots and record votes securely and confidentially.
- Calculate and publish election results based on the cast ballots.
- Ensuring that each voter can only cast one ballot per election.

By methodically constructing Solidity code for the Migrations.sol and Voting.sol contracts, we specify the logic and behavior of our voting system in a manner that is secure, transparent, and resistant to manipulation.

C. *Using Ganache for setting up testing accounts locally*

Testing is a key component of software development, allowing us to find and solve flaws before deployment to production settings. In the context of blockchain development, Ganache acts as a valuable tool for setting up local testing environments, enabling developers to replicate a blockchain network on their local PCs. By employing Ganache, we develop a lightweight, flexible blockchain network that functions purely locally, removing the need for costly interactions with public test networks. This local blockchain network simulates the capabilities of a real blockchain, allowing us to deploy and interact with smart contracts, create test accounts, and simulate transactions—all within a controlled environment. Ganache's user-friendly interface enables access into blockchain status and transaction history, aiding debugging and troubleshooting efforts. Additionally, Ganache includes capabilities such as deterministic blockchain behavior and snapshot functionality, enabling accurate and reproducible testing scenarios.

To establish a workspace in Ganache and configure the RPC servers, network ID, and other parameters, we follow these steps:

a) To begin, we download and install Ganache from the official website or via package managers like npm or Chocolatey, acquiring a personal Ethereum blockchain for smart contract development and testing.

b) Next, Ganache is launched to access its user-friendly interface, providing setup options and local blockchain network information.

c) A new workspace is created by clicking the “New Workspace” button or selecting “New Workspace” from the “File” menu, generating a new instance of Ganache with default settings.

d) Customizing workspace parameters such as the number of accounts, initial balance per account, gas limit, network ID, etc., according to our preferences is essential.

e) We ensure the RPC server is enabled and configured correctly to facilitate interfacing with the local blockchain network from external programs.

f) A unique network ID is assigned to the Ganache workspace to differentiate it from other Ethereum networks.

g) Additional variables like block duration, gas price, and chain ID are selected based on specific development requirements.

h) The workspace configuration is saved to preserve changes, and a descriptive name is assigned for future identification.

i) Finally, we start the Ganache workspace to generate the local blockchain network, accessing extensive network information within the Ganache interface.

By following these steps, we establish a workspace in Ganache and configure the RPC servers, network ID, and other settings to create a local blockchain environment for Ethereum development and testing. Ganache provides a convenient and user-friendly framework for designing and deploying smart contracts, enabling rapid iteration and efficient development processes.

.

D. *Integrating MetaMask for smooth transaction handling*

MetaMask plays a vital role in ensuring seamless interaction between users and the Ethereum blockchain, particularly in the context of decentralized apps. As a browser extension and mobile app, MetaMask provides users with a straightforward interface for maintaining Ethereum accounts, communicating with smart contracts, and executing transactions. In our voting system, we enhance the user experience by streamlining transaction handling and ensuring security through MetaMask. With MetaMask, users can securely sign transactions, verify transaction data, and manage their Ethereum accounts—all inside the familiar constraints of their web browser or mobile device. By employing MetaMask's powerful infrastructure, we ensure that transactions within our voting system are conducted securely and swiftly, improving user confidence and trust. Additionally, MetaMask's support for different networks and bespoke RPC endpoints enables easy interaction with other blockchain networks, giving flexibility and scalability. Hence, it is vital to know how to build a MetaMask account and then combine the Ganache network with it.

a) We begin by installing the MetaMask browser extension on our preferred web browser (e.g., Google Chrome, Mozilla Firefox). MetaMask can be found on the Chrome Web Store or the Firefox Add-ons page.

b) Next, we create a new MetaMask account by following the prompts after opening the extension. This involves setting up a password and consenting to the terms of service.

c) Upon creating a new MetaMask account, we receive a seed phrase consisting of 12 words. We securely store this seed phrase as it serves as a backup for accessing our MetaMask account if needed.

d) We then set a name for our MetaMask account to quickly identify it and ensure that MetaMask is linked to the Ethereum Mainnet by selecting “Ethereum Mainnet” from the network dropdown menu.

To integrate the Ganache network into MetaMask, we first ensure that Ganache is launched and connected to the local blockchain network. Then, in the Ganache interface, we locate and copy the RPC server endpoint URL. Next, we access the MetaMask settings in our web browser and add a custom RPC network, pasting the Ganache RPC server endpoint URL. After configuring the network details and saving the settings, we switch to the Ganache network in MetaMask. To import accounts from Ganache into MetaMask, we copy the private key of the selected account in the Ganache interface and import it into MetaMask following the prompts.

By integrating the Ganache network into MetaMask, we are provided with a smooth transition from development to deployment, guaranteeing that smart contracts behave as intended in a local context before being published to the Ethereum mainnet or test networks. This connection helps developers to test transaction handling, contract interactions, and account management smoothly, promoting confidence in the dependability and functionality of their Ethereum applications. With MetaMask providing a user-friendly interface for communicating with Ganache, developers can iterate rapidly and efficiently, speeding the development cycle and ultimately generating strong and dependable decentralized applications.

E. Testing and Validation

Being pivotal stages in the development lifecycle, testing and validation ensures the efficacy, reliability, and user satisfaction of the voting application. Developers establish a local testing environment using Node.js, facilitating comprehensive end-to-end testing to evaluate core functionalities such as user registration, voting, and result presentation. Performance testing is conducted to assess the application's behavior under varying load conditions, identifying and rectifying performance bottlenecks for optimized responsiveness. User acceptance testing gathers valuable feedback on usability and functionality, guiding iterative refinements to enhance user experience. Integration testing validates seamless connectivity between frontend and backend components, minimizing risks of errors or inconsistencies. Rigorous deployment preparation ensures swift resolution of any detected issues, streamlining the transition to production servers for general use. This iterative process, combining rigorous testing, user feedback, and meticulous deployment readiness, aims to deliver a high-quality application that meets user needs and expectations.

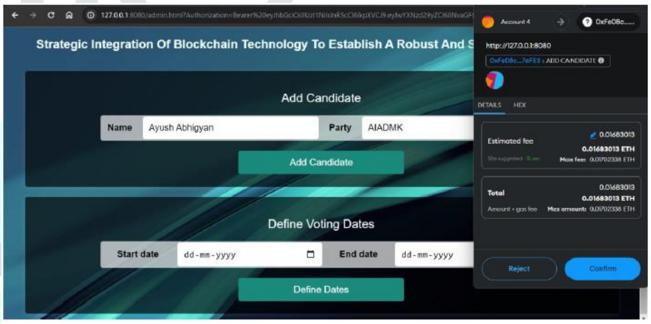

Fig 3 Adding Candidate through MetaMask transaction

V. Results and Discussion

The results of this study are significant and promising, as they offer insight on the efficacy and potential of decentralized voting methods in modern democratic processes. The project's major findings underline the necessity of adopting blockchain technology to enhance the integrity, security, and transparency of election systems. By adopting a decentralized approach to voting, the project attempted to overcome longstanding difficulties such as voter fraud, tampering, and lack of transparency in traditional voting systems. Through thorough design and implementation, the project guaranteed that each part of the voting process, from user authentication to ballot casting and tallying, was done securely and transparently.

One of the project's major decisions was the employment of blockchain technology, primarily Ethereum, to offer a tamper-proof and unchangeable ledger for recording votes. By employing smart contracts, the initiative ensured that votes were safely recorded and counted, without the possibility of manipulation or illegal access. Additionally, the integration of Metamask provides users with a seamless and user-friendly interface for interacting with the voting platform, significantly improving accessibility and acceptance.

It is necessary to examine its performance across multiple dimensions and explore its impact on democratic processes. The following points provide a detailed examination of the system's performance, user comments, election outcomes, obstacles faced, and prospective routes for future enhancements and study. Let us explore each issue in detail:

a) System Performance Evaluation:
The section begins by analyzing the performance of the decentralized voting system in terms of efficiency, security, and user experience. Metrics such as transaction processing speed, latency, and system uptime are monitored to determine the overall performance of the system. Additionally, security measures like as encryption, tamper resistance, and resistance to assaults are considered to protect the integrity and confidentiality of the voting process.

b) User Feedback and Satisfaction:
User feedback and satisfaction play a significant part in determining the effectiveness of the decentralized voting system. Surveys, interviews, and user testing sessions are undertaken to get feedback from users regarding their experience with the voting platform. Factors such as simplicity of use, accessibility, and contentment with the voting process are assessed to discover areas for improvement and promote user satisfaction.

c) Impact on Election Outcomes:
The impact of the decentralized voting system on election outcomes and democratic processes is addressed in detail. By analyzing voting patterns, turnout rates, and election results, the efficiency of the system in promoting fair and transparent elections is evaluated. Additionally, any observed anomalies or abnormalities in the voting process are probed to ensure the integrity and accuracy of election outcomes.

d) Challenges and Limitations:
The section also tackles the issues and limitations discovered during the implementation and deployment of the decentralized voting system. Technical concerns, regulatory limits, and user acceptance barriers are examined, along with potential tactics for addressing these challenges in future generations of the system.

e) Future Enhancements and Research Directions:
Finally, the results and discussions section describes potential future enhancements and research objectives for the decentralized voting system. Areas including as scalability, interoperability with other blockchain networks, and interaction with upcoming technologies are researched to further improve the system's functionality and answer increasing needs and expectations.

The use cases for decentralized voting systems extend beyond traditional electoral processes. In addition to aiding national and municipal elections, similar systems can be applied to numerous fields where secure and transparent decision-making is vital. For instance, decentralized voting systems can be applied in organizational governance, shareholder voting, and community decision-making procedures. By democratizing decision-making and preserving the integrity of votes, decentralized voting systems have the potential to generate greater confidence, participation, and accountability in both public and private sector environments.

Therefore, this study constitutes a successful initiative in demonstrating the viability and potential of decentralized voting systems in current democratic processes. Through thorough study and review, it has produced vital insights into the performance, effect, and future orientations of such systems. By tackling fundamental problems and highlighting the benefits of blockchain technology in voting systems, this study lays the framework for further breakthroughs in democratic governance and decision-making.

A. Performance Analysis

1. Test Case 1:
This is a unit test focused on checking JWT authorization, the objective was to verify that users cannot access the voting or admin page without proper authorization. The test confirmed that users are indeed unable to log in without the required authorization, indicating successful implementation of JWT-based authentication. This ensures that unauthorized access attempts are effectively prevented, enhancing the security of the voting system.

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.

Fig. 4 Test Case 1

2. Test Case 2:
Moving on to Test Case 2, a functional test aimed at verifying user login functionality, the goal was to ensure that users can successfully log

in to the voting portal using valid credentials. The test results demonstrated that users can log in only when their credentials match those stored in the database, as expected. This confirms that the login mechanism functions correctly, allowing authorized users to access the voting platform while maintaining security against unauthorized access attempts.

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.

Fig. 5 Test Case 2

3. Test Case 3:
This test focused on a unit test to verify candidate registration by the admin. The test aimed to confirm the successful registration of candidates by the admin, ensuring that candidate data is accurately recorded in the system. The test results indicated that the registration transactions were indeed successful, validating the functionality for registering candidates. This ensures that the voting system can effectively manage candidate information, facilitating a smooth election process.

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.

Fig. 6 Test Case 3

4. Test Case 4:
Another unit test was conducted to verify the ability of the admin to specify the date of voting. The test objective was to ensure that the admin can set the starting and ending dates for voting, enabling effective scheduling of the voting period. The test results showed that the date transactions were successful, confirming that the admin can specify the voting dates without any issues. This functionality ensures proper organization and timing of the voting process, contributing to its efficiency and reliability.

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.

Fig. 7 Test Case 4

5. Test Case 5:
Finally, Test Case 5 focused on a functional test to verify the voting process from the user's perspective. The test aimed to confirm that voters can cast their votes successfully for their chosen candidates. The test results revealed that the vote transactions were indeed successful, indicating that voters can participate in the voting process without encountering any issues. This ensures that the voting system operates smoothly, enabling users to exercise their voting rights effectively and securely.

Test Case No.	5
Test Type	Functional Test
Name of Test	Verify voting
Test Case Description	The objective of this test case is to verify that voter is able to cast their vote.
Input	Select a candidate and click "Vote" button.
Expected Output	Vote transaction should be successful.
Actual Output	Vote transaction is successful.
Result	Pass
Comments	Working properly.

Fig. 8 Test Case 5

In conclusion, the rigorous testing and validation undertaken on our voting system project have been crucial in assuring its operation, dependability, and adherence to criteria. Through a combination of unit testing, integration testing, functional testing, white box testing, and black box testing, we have systematically assessed the system from numerous viewpoints, detecting and fixing any flaws or faults along the way. The complete testing results, published in tabular style, provide vital insights into the performance and resilience of the voting system, confirming its preparedness for deployment and use in real-world circumstances. Moving ahead, continued testing efforts and periodic evaluations will be needed to maintain and enhance the system's quality and efficacy throughout time

B. Comparison Between existing Models

1. Transparency and Security:

• Proposed System: The decentralized voting system utilizing Ethereum blockchain ensures transparency and security through the utilization of smart contracts, providing a tamper-proof platform for elections. By leveraging blockchain technology, it offers visibility across the entire voting process, assuring fairness and integrity in the outcome.

• Existing System: In contrast, traditional voting systems lack transparency, making it difficult to verify the correctness and fairness of the counting process. Paper-based ballots and electronic voting machines are vulnerable to manipulation and hacking, posing significant security risks and undermining trust in the electoral process.

2. Accessibility and Global Reach:

• Proposed System: The decentralized voting system enables global accessibility, allowing voters to cast their ballots from anywhere in the world. By eliminating the need for physical infrastructure, it enhances accessibility and inclusivity, reducing barriers to participation and voter disenfranchisement.

• Existing System: Traditional voting systems often require voters to visit specific polling places, limiting accessibility for individuals with disabilities or mobility challenges. This centralized approach impedes voter turnout and restricts democratic participation, particularly in remote or underserved communities.

3. Efficiency and Cost-Effectiveness:

• Proposed System: Leveraging blockchain technology, the proposed system offers cost-effectiveness by eliminating the expenses associated with physical infrastructure, such as polling venues and voting machines. Additionally, it enhances efficiency by providing real-time access to election outcomes, streamlining the voting process, and reducing time and resource requirements.

• Existing System: Traditional voting systems incur significant costs in recruiting poll workers, acquiring and maintaining voting machines, and renting polling venues. Manual counting of paper ballots leads to delays in announcing results, contributing to inefficiencies and increased operational expenses.

In comparing the existing and proposed voting systems, it is evident that the decentralized approach offers significant advantages in terms of transparency, security, accessibility, efficiency, and cost-effectiveness. By leveraging blockchain technology and smart contracts, the proposed system addresses key shortcomings of traditional techniques, ultimately enhancing democratic values and citizen trust in the electoral process.

VI. Conclusion and Future Scope

A. Conclusion

Decentralized Voting with Ethereum Blockchain sits at the forefront of upgrading electoral systems by giving a robust and transparent alternative for secure elections. Through the deployment of blockchain technology, it not only ensures the integrity of votes but also provides a tamper-proof platform that safeguards against fraudulent actions and manipulation. The unchangeable nature of blockchain assures that once a vote is recorded, it cannot be altered or interfered with, hence improving trust and confidence in the democratic process.

Furthermore, Ethereum Blockchain's decentralized architecture decentralizes authority, eliminating the need for centralized authorities or middlemen, which are typically prone to corruption or manipulation. This decentralized method provides a more open and equitable voting environment, where every eligible participant has an equal opportunity to express their views and contribute to the decision-making process.

Moreover, the continued enhancements and advancements in decentralized voting systems, such as improved user experience, scalability, and integration with other cutting-edge technologies like biometrics or artificial intelligence, further solidify its potential to revolutionize the democratic process. By focusing user-centric design and accessibility, decentralized voting platforms can attract a broader audience and encourage greater involvement in the election process.

Additionally, the incorporation of smart contracts with Ethereum Blockchain-based voting systems offers automation and efficiency, speeding the voting process and lowering administrative overhead. Smart contracts provide transparent and auditable elections by automating vote counting, tallying, and result verification, thereby lowering the risk of human error or manipulation.

In conclusion, Decentralized Voting using Ethereum Blockchain marks a big step towards building a more democratic and responsible society. By using the power of blockchain technology, it empowers citizens to engage in a reliable and efficient voting system, thereby strengthening the foundations of democracy and assuring the participation of various voices in decision-making processes. As decentralized voting systems continue to expand and mature, they have the potential to transform the future of elections and governance, enabling more openness, integrity, and inclusivity in democratic countries.

B. Future Scope:

As decentralized voting systems continue to evolve, future iterations have great potential for boosting the efficiency, security, and inclusivity of the political process. By leveraging developing technologies and implementing creative features, decentralized voting systems can further promote democratic norms and empower citizens to participate in decision-making processes. Here are some potential future enhancements:

1) Real-time Vote Counting:

- Implementing real-time vote counting capabilities using blockchain technology can provide instant and accurate updates on election results.

- This feature enhances transparency and reduces the time required for tallying votes, enabling stakeholders to access timely information on the outcome of elections.

2) Secure Voter Identification Mechanisms:

- Introducing safe voter identification mechanisms, such as biometric authentication or digital identity verification, helps alleviate problems linked to voter impersonation and fraud.

- By employing biometric data or cryptographic approaches, decentralized voting systems can ensure that only eligible voters are able to cast their ballots, enhancing the integrity of the election process.

3) Advanced Data Analytics for Voter Insights:

- Integrating advanced data analytics tools into decentralized voting systems enables the generation of valuable insights into voter behavior, preferences, and demographics.

- By analyzing voting patterns and trends, election authorities can gain a deeper understanding of voter sentiment and tailor outreach efforts to better engage with diverse constituencies.

4) Integration with Emerging Technologies:

- Embracing emerging technologies like artificial intelligence (AI) and machine learning (ML) can further enhance the functionality and efficiency of decentralized voting systems.

- AI-powered algorithms can assist in spotting abnormalities or irregularities in voting patterns, while ML models can optimize resource allocation and decision-making processes within electoral management organizations.

5) Enhanced Accessibility Features:

- Implementing accessibility elements, such as multi-language support, audio ballots, and assistive technology for voters with impairments, guarantees that the voting process is accessible to all individuals.

- By stressing inclusivity and usability, decentralized voting systems can remove barriers to participation and create more civic involvement among various communities.

References

[1] W. Xue, Y. Yang, Y. Li, H. H. Pang, and R. H. Deng, "ACB-Vote: Efficient, Flexible, and Privacy-Preserving Blockchain-Based Score Voting With Anonymously Convertible Ballots," in IEEE Transactions on Information Forensics and Security, vol. 18, pp. 1-13, 2023, Art no. 5022113, doi: 10.1109/TIM.2023.3298426.

[2] M.-V. Vladucu, Z. Dong, J. Medina, and R. Rojas-Cessa, "E-Voting Meets Blockchain: A Survey," in IEEE Access, vol. 11, pp. 1-13, 2023, doi: 10.1109/ACCESS.2023.3297492.

[3] S.-V. Oprea, A. Bâra, A.-I. Andreescu, and M. P. Cristescu, "Conceptual Architecture of a Blockchain Solution for E-Voting in Elections at the University Level," in IEEE Access, vol. 11, pp. 1-13, 2023, doi: 10.1109/ACCESS.2023.3297492.

[4] S. Chaudhary et al., "Blockchain-Based Secure Voting Mechanism Underlying 5G Network: A Smart Contract Approach," in IEEE Access, vol. 11, pp. 76537-76550, 2023, doi: 10.1109/ACCESS.2023.3297492.

[5] S. Wang, X. Qu, Q. Hu, X. Wang, and X. Cheng, "An Uncertainty- and Collusion-Proof Voting Consensus Mechanism in Blockchain," in IEEE/ACM Transactions on Networking, vol. 31, no. 5, pp. 2376-2388, Oct. 2023, doi: 10.1109/TNET.2023.3249206.

[6] J. Huang, D. He, Y. Chen, M. K. Khan, and M. Luo, "A Blockchain-Based Self-Tallying Voting Protocol With Maximum Voter Privacy," in IEEE Transactions on Network Science and Engineering, vol. 9, no. 5, pp. 3808-3820, 1 Sept.-Oct. 2022, doi: 10.1109/TNSE.2022.3190909.

[7] M. S. Farooq, U. Iftikhar, and A. Khelifi, "A Framework to Make Voting System Transparent Using Blockchain Technology," in IEEE Access, vol. 10, pp. 59959-59969, 2022, doi: 10.1109/ACCESS.2022.3180168

[8] P. Y. A. Ryan, S. Schneider and V. Teague, "End-to-End Verifiability in Voting Systems, from Theory to Practice," in IEEE Security & Privacy, vol. 13, no. 3, pp. 59-62, May-June 2015, doi: 10.1109/MSP.2015.54.

[9] "IEEE Draft Framework for Use of Distributed Ledger Technology in Security of Electronic Voting (e-Voting) Systems," in IEEE P2418.11/D8, May 2023 , vol., no., pp.1-60, 2 June 2023.

[10] E. Zaghloul, T. Li and J. Ren, "d-BAME: Distributed Blockchain-Based Anonymous Mobile Electronic Voting," in IEEE Internet of Things Journal, vol. 8, no. 22, pp. 16585-16597, 15 Nov.15, 2021, doi: 10.1109/JIOT.2021.3074877.

[11] B. Shahzad and J. Crowcroft, "Trustworthy Electronic Voting Using Adjusted Blockchain Technology," in IEEE Access, vol. 7, pp. 24477-24488, 2019, doi: 10.1109/ACCESS.2019.2895670.

[12] D. Xu, W. Shi, W. Zhai and Z. Tian, "Multi-Candidate Voting Model Based on Blockchain," in IEEE/CAA Journal of Automatica Sinica, vol. 8, no. 12, pp. 1891-1900, December 2021, doi: 10.1109/JAS.2021.1004207.

[13] S. Panja, S. Bag, F. Hao and B. Roy, "A Smart Contract System for Decentralized Borda Count Voting," in IEEE Transactions on Engineering

Management, vol. 67, no. 4, pp. 1323-1339, Nov. 2020, doi: 10.1109/TEM.2020.2986371.

[14] C. Killer, B. Rodrigues, E. J. Scheid, M. F. Franco and B. Stiller, "Blockchain-Based Voting Considered Harmful?," in IEEE Transactions on Network and Service Management, vol. 19, no. 3, pp. 3603-3618, Sept. 2022, doi: 10.1109/TNSM.2022.3181028.

[15] S. Venugopalan, I. Stančíková and I. Homoliak, "Always on Voting: A Framework for Repetitive Voting on the Blockchain," in IEEE Transactions on Emerging Topics in Computing, vol. 11, no. 4, pp. 1082-1092, Oct.-Dec. 2023, doi: 10.1109/TETC.2023.3315748.