International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 03 | March - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Secured Voting System Using Blockchain

Prof. Aditi Warange

Krupa Panchal , Amruta Shinde , Pallavi Nesarkar, Nisarg Panchal

Bachelor's Degree in Computer Science Engineering AIML & Bharat College of Engineering

Abstract - Human rights have frequently been compromised, particularly in the realm of voting systems. Many digital voting platforms face criticism due to a lack of transparency, making it difficult for authorities to earn public trust. Both traditional and modern voting mechanisms have vulnerabilities that can be exploited, leading to potential injustices or irregularities during elections. Our proposed framework addresses these challenges by implementing a scalable blockchain infrastructure with adaptable consensus algorithms. The security protocol integrated into the voting system enhances the protection of transactions. Smart contracts establish a reliable link between users and the network, ensuring secure transaction execution. Additionally, the security of blockchain-based voting has been analyzed, highlighting cryptographic hashing for transaction encryption and strategies to mitigate the risks of a 51% attack on the blockchain.

***Key Words*:** Secure online voting by using blockchain system, Scalable consensus mechanisms .

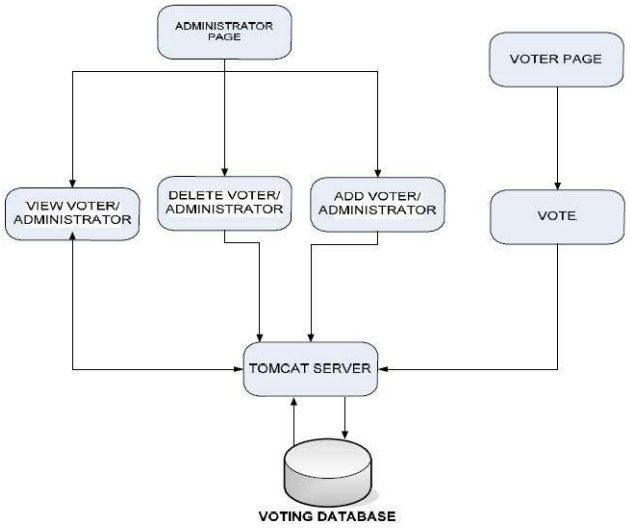
1. INTRODUCTION

Voting is fundamentally a method of expressing choice, typically conducted through ballots or other electoral procedures. Electronic voting introduces a digital approach, allowing votes to be cast, collected, counted, and securely stored using an electronic platform. This project aims to transform the University of Westminster Student Union's traditional paper-based election system into an efficient and modern electronic voting system.The existing voting system faces challenges, particularly low voter participation, due to its lack of accessibility and convenience. Many students find the current process inconvenient, which discourages them from participating in elections. To address this issue, the proposed system will enable students to cast their votes online using any internet-enabled device, making the process more accessible and user-friendly.This project will analyze the student union's current voting method and develop a structured model that integrates seamlessly into an online voting platform. Various election mechanisms will be implemented to ensure flexibility and adaptability to different voting scenarios.A key focus of this system will be its security measures, ensuring the integrity and confidentiality of the voting process. Security protocols will be in place from the moment a voter logs into the system, through the voting process, until they securely exit the platform. Additionally, the system will incorporate strict access controls to prevent multiple votes from being cast by a single user, ensuring fairness and transparency in the election process. With robust security, user authentication, and encryption techniques, the proposed system will provide a safe, reliable, and efficient voting experience for all students.

1.1 Project Plans:

The Secure Voting System is a web-based platform designed to facilitate a transparent, efficient, and secure election process. It operates within a structured framework where both voters and administrators interact with a Tomcat Server, which processes all requests and securely manages election data within a dedicated voting database. The system is built around two key user roles: administrators, who oversee and manage the election process, and voters, who can securely submit their ballots. The Administrator Panel provides comprehensive management tools, enabling administrators to oversee both voter and administrator accounts. Administrators have the ability to add, remove, and update user records while also monitoring election results. These tasks are processed through the Tomcat Server, which communicates with the voting database to ensure data integrity and seamless updates. Election outcomes are presented in a structured and graphical format for clear analysis and reporting. On the Voter Interface, eligible users can securely submit their votes through an intuitive and protected platform. When a voter submits a ballot, the system validates their eligibility, timestamps the vote, and securely records it in the database. To maintain the integrity of the election, multiple voting attempts from the same user are strictly prevented, anonymity is preserved, and real-time data synchronization ensures accuracy in vote counting. The Tomcat Server functions as the core processing unit of the system, managing user interactions, data transactions, and security protocols. It ensures efficient handling of administrative tasks such as user management, result retrieval, and data protection. The voting database is fortified against unauthorized access and external threats, ensuring confidentiality and security. The system follows a structured development lifecycle, beginning with research and planning, where security requirements and functional needs are analyzed. The design phase focuses on structuring the database and creating user-friendly interfaces for administrators and voters. During the development phase, key features such as authentication, voting functionality, and administrative controls are implemented.

Finally, rigorous testing is conducted to identify vulnerabilities, enhance security, and optimize system performance before deployment. By leveraging modern security measures and an efficient architecture, this system ensures a reliable and user-friendly voting experience, reinforcing transparency and trust in the electoral process.



2. Review of Literature:

The Secured Voting System Using Blockchain is designed to provide a secure and efficient online voting platform with robust authentication, ensuring only authorized voters can access the system. A secure database will store votes and user information, while advanced security measures will protect against compromise and external attacks. The system will automatically tally votes accurately and transparently. A backend administration section will facilitate election management, allowing administrators to add, update, and remove voters, candidates, and sub-administrators. Voting results will be displayed graphically for better analysis, and voters will have access to candidate biographies before casting their votes. Each vote will be timestamped for tracking, and administrators can generate comprehensive election reports. Additionally, built-in mechanisms will prevent multiple voting, ensuring fairness and integrity in the election process.

2.1 Existing Systems:

Traditional voting systems primarily rely on paper-based or electronic voting machines (EVMs), which are widely used in elections worldwide. While these methods have been in practice for decades, they suffer from multiple drawbacks, including the risk of electoral fraud, vote manipulation, and logistical inefficiencies. Paper-based voting requires extensive resources for printing, distributing, and counting ballots, leading to increased costs and human errors. Furthermore, manual vote counting is time-consuming and susceptible to tampering or misinterpretation. Electronic voting machines (EVMs) offer an alternative but still pose security concerns such as unauthorized access, tampering, and lack of transparency.

2.2 Literature Surveyof Similar Ideas:

- Zhang, Y., & Wen, J. (2019). A Blockchain-Based Voting System for the US Presidential Election.
- Benaloh, J., Rivest, R., Ryan, P. Y., et al. (2013). End-to-End Verifiability.
- Adida, B. (2008). Helios: Web-based Open-Audit Voting.
- Chaum, D. (1981). Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms.
- Cramer, R., Gennaro, R., & Schoenmakers, B. (1997). A Secure and Optimally Efficient Multi-Authority Election Scheme.
- Mercuri, R. (2002). Electronic Voting Systems: A Failure of Security.
- Rivest, R. L., & Wack, J. P. (2006). On the Notion of ‘Software Independence’ in Voting Systems.
- Feldman, A. J., Halderman, J. A., & Felten, E. W. (2007). Security Analysis of the Diebold AccuVote-TS Voting Machine.
- Kshetri, N., & Voas, J. (2018). Blockchain-Enabled E-Voting.

3. Proposed System

The proposed secure voting system operates using a Tomcat server and a voting database to manage voter authentication and election processes. The administrator page allows admins to view, add, or delete voters and administrators, ensuring proper user management. The voter page enables users to cast their votes securely, with the system handling authentication and vote storage. The Tomcat server acts as an intermediary, processing requests and updating the voting database in real time. This system ensures efficient election management, secure voting, and centralized control.

3.1 Analysis/Framework/Algorithm:

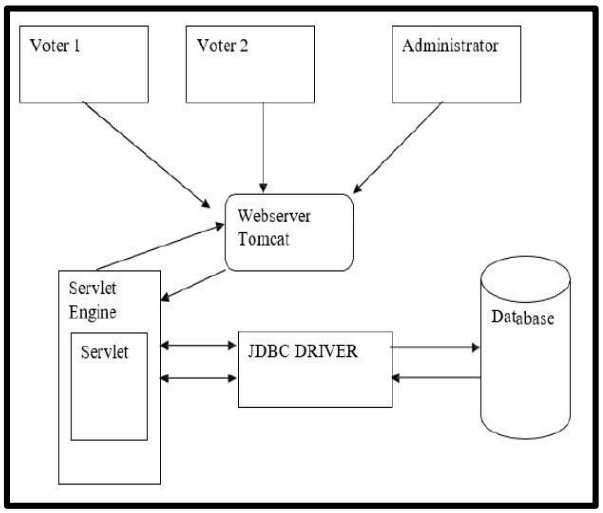
This chapter’s goal is to describe the way in which the online voting system is to be built. In order to build an efficient and flexible system, the appropriate system development methodology has to be chosen to suit the system to be created. The waterfall design methodology is been utilized to design and develop the online voting system. In order for any form of computer systems to be built in an efficient and user-friendly manner, a highly structured and well-engineered design has to be created. The design of a software orientated system has to follows certain steps in achieving its end product. The design of a system enables organizations and companies to map out a strategic plan which the system developers would have to follow. The design of a system is very important in the construction of any web-based application, and it prevents the occurrence of mistakes and errors during the implementation phase which can be highly costly to the organization funding the specify project.

The development of an online voting system will involve two key user roles: voters, who will participate in elections, and administrators, who will manage and oversee system operations. Since voting is a highly sensitive process, ensuring top-tier security is imperative to prevent any manipulation, fraud, or unauthorized access. The primary goal of this system is to maintain election integrity by preventing any form of vote tampering. To achieve this, a robust authentication mechanism must be implemented to verify user identities and safeguard election data. In addition to security, the system's usability is crucial. A well-designed voting system should be intuitive, efficient, and free from unnecessary complexity, allowing users to cast their votes with ease. A seamless interface ensures accessibility for all users, encouraging voter participation. The development process must also consider how data is handled—how information is entered, processed, and displayed within the system. To achieve an efficient and structured design, various system modeling tools will be utilized to map the development process from inception to implementation. Use Case Diagrams will provide a visual representation of how different users interact with the system, while Data Flow Diagrams (DFDs) will illustrate the complete architectural structure and flow of information. These design approaches will serve as a blueprint for developers, ensuring the system is built with precision, consistency, and efficiency. Furthermore, the system will include multiple security layers, such as encryption for vote protection, secure login procedures, and preventive measures against unauthorized voting attempts. By combining advanced security protocols with user-friendly functionality, the online voting system will provide a reliable, transparent, and fraud-proof platform, promoting trust and efficiency in the electoral process.

3.2 System Architecture:

The system's architecture will incorporate various client-side and server-side technologies to ensure seamless operation and security. The frontend of the system will serve as the user interface within a web browser, enabling users to send HTTP requests and receive HTTP responses from the server. To build this interface, HTML will be used to structure and display content, ensuring an intuitive and user-friendly experience. On the other hand, the backend will manage all server-side operations, processing incoming requests from the frontend. A Tomcat server engine will be utilized to load servlets and JSPs, which will handle dynamic content generation. The processed data will be sent back to the client in HTML format, allowing users to view relevant voting information in real time. For data storage, the system will employ a MySQL database to manage election-related information, including user credentials, votes, and election records. To facilitate efficient data retrieval and interaction between the database and the server, a JDBC API driver will act as a middleware layer, ensuring smooth data transactions.

Given the sensitive nature of an online voting system, implementing robust security measures is a top priority to protect user data and maintain system integrity. To enhance security, the system will incorporate three key protective measures. First, an account lockout mechanism will be implemented, recording failed login attempts. If a user exceeds a predefined number of incorrect password entries, the system will temporarily lock their account to prevent brute-force attacks. Second, password encryption will be enforced, ensuring that all stored passwords are secured using a strong encryption algorithm. This prevents unauthorized access even if the database is compromised. Lastly, rather than displaying stored passwords when users forget them, the system will implement a secure password reset mechanism. This approach ensures that users can regain access to their accounts without exposing their stored credentials. By integrating these technologies and security protocols, the proposed system will deliver a secure, efficient, and user-friendly online voting platform, ensuring a fair and transparent electoral process.



(fig. 2 System Architecture)

3.3 Data Model:

In order to develop the online voting system, a database system has to be in place to be used to store all the data retrieved from the users of the system. The database system to be created will also play a major part for enforcing and strengthening the security of the voting system. Authentication of the system's users will rely on the details of the users which would be stored in the database system. MySQL database server has been selected as the database of choice, due to the sheer fact that it is open source which cuts the cost of having to buy database software. MySQL has a very large storage capacity which will be essential for storing the large amount of data to be inputted.

Entity Name	Description
Administrator	The administrator table will be used to store all the details of the administrators utilizing the system. Each administrator will have a unique username. The attributes utilized in this entity are shown in figure.
Candidates	The candidates table will be used to store all the details of the elections candidates. Each candidate will have a unique username. The attributes utilized in this entity are shown in figure
Users	The user's tables will be used to store the encrypted passwords of the voters and administrators. A field within the table called "type" will also be used to differentiate voters from administrators within the table. The tries field will be used to store the number of login attempts by each user. The attributes utilized in this entity are shown in figure
Voters	The voters table will be used to store the details of each voter in the system. Due to the high security measures to be taken when developing the system, the voters table will also contain fields with the records of each candidate voted for by the voter, this design has to be done to prevent the possibility of a voter voting more than one. Through this means if there is any need for suspicion of vote rigging by the elections organizer the database table can prove that each voter voted only once. A field called "voted will also be used to record when each voter has cast their vote by incrementing to 1. A timestamp field will also be added to record the exact time each voter cast their vote. The attributes utilized in this entity are shown in figure

Fig 3: The Entity Relationship Diagram

3.4 Methodology:

A secure voting system using blockchain ensures transparency, privacy, and security by leveraging decentralized and immutable technology. Voter registration involves identity verification, and once verified,voters receive unique tokens stored on the blockchain. Votes are cast via encrypted channels and recorded on the blockchain, ensuring they can't be altered. Smart contracts manage the voting process, and a consensus mechanism validates votes. Blockchain's public ledger provides transparency while keeping votes private through encryption. Multi-factor authentication, digital signatures, and advanced encryption methods ensure security. Post-election, the results are immutable and auditable, providing integrity and transparency throughout the process.Finally, post-election integrity is assured by the transparent nature of blockchain. Once the election concludes, the results are immutable, and any changes or tampering are easily detectable. The entire voting process is recorded on the blockchain, creating a verifiable audit trail. In the event of a dispute, authorities can use this audit trail to trace discrepancies and resolve the issue fairly.

4. Result and Discussion:

In summary, blockchain technology provides a secure, transparent, and private environment for online voting by leveraging its inherent features like decentralization, immutability, and cryptographic security.

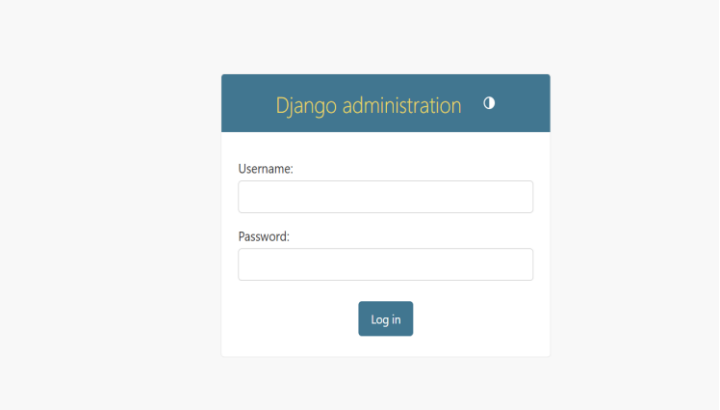
4.1 Proposed System Result:

The proposed system aims to correct these problems by introducing a more secure and efficient way of voting. This could involve using a voter ID card or other form of identification to verify each person's identity before they are allowed to vote.

UI of Homepage:

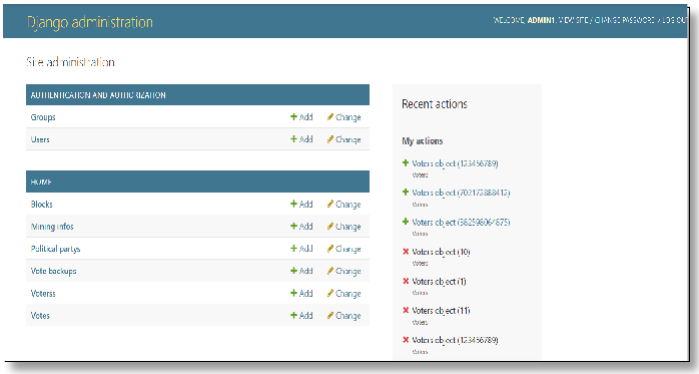
A blockchain-based voting system with a secure and transparent approach. Users authenticate using their Aadhaar number for identity verification. Frontend developed using HTML, CSS, and Python for a smooth user experience.

Django Administration:



Django Administrator login console, the admin have to put there username & password to login into the admin portal. we are using Django backend framework, where Admin can Add voters Details & political Parties.

Django Administration Backend :



Django Servers as the backend for stroing Users votes & details calmly during the enrollment process. It supports multiple relational databases like PostgreSQL, MySQL, SQLite, and Oracle. Django automatically creates tables based on model definitions.Supports connection pooling to improve database performance.

5. Conclusion:

The primary objective of this project was to develop a secure online voting system that would replace the traditional paper-based voting method with a digital alternative. The goal was to enable voters to cast their ballots remotely from any location via the internet, ensuring greater accessibility and convenience. Extensive research was conducted on existing online voting systems to analyze their features and assess how they influence voter participation in elections.Additionally, various server-side technologies were explored to determine the most suitable programming language for building the system. A thorough examination of potential security risks was carried out, identifying vulnerabilities that could compromise the integrity of the voting process. Countermeasures were then devised to strengthen the system's security and prevent potential threats. To ensure a structured and efficient development process, multiple software development methodologies were reviewed.

After careful evaluation, the waterfall methodology was selected as the most appropriate approach for this project.During the design and development phase, significant effort was dedicated to creating a user-friendly and efficient system aligned with the initial project proposal. This phase provided a well-defined framework for system implementation, ensuring a structured approach to development.

References:

[1] Jayson Falkner, Ben Galbraith, Romin Irani, Casey Kochmer, Sathya Narayana Panduranga, Krishnaraj Perrumal, John Timney, Meeraj Moidoo Kunnumpurath, (2001), Beginning JSP Web Development,Wrox. [2] Peter denHaan, Lance Lavandowska, Sathya Narayana Panduranga, Krishnarag Perrumal, (2004), Beginnign JSP 2 From Novice to Professional, Apress. [3] Aneesha Bakharia, (2001),PythonServerPages,Prima Tech. [4] Bruce W.Perry, (2004), Python Servlet & JSP Cookbook, O'Reilly[5] Simson Garfinkel, Gene Spafford,(1997), Web Security & Commerce, O'Reilly.

Acknowledgment

We express our gratitude to our project guide Prof.Aditi Warange, Assistant Professor, Department of Computer Science Engineering AIML for her valuable suggestions, cooperation, and support in the working of this paper.