International Journal of Multidisciplinary Engineering in Current Research - IJMEC
Volume 10, Issue 6, June-2025, http://ijmec.com/, ISSN: 2456-4265

Anomalynet: An Anomaly Detection Network For Video Surveillance

Dr P Sumalatha, Mallela Spandana, M Sri Navya

[1]Associate Professor, Department Of Ece, Bhoj Reddy Engineering College For Women, India.
[2,3,4]B. Tech Students, Department Of Ece, Bhoj Reddy Engineering College For Women, India.

ABSTRACT

The "Secured Digital Voting System" is a robust and advanced web-based solution aimed at providing a secure, transparent, and tamper-proof voting experience. Built using Python (Flask) for the backend and MongoDB as a NoSQL database, this platform ensures the safe storage of voting records. The system is designed to be scalable, user-friendly, and resilient against fraudulent activities, guaranteeing that each vote is authentic, verifiable, and immutable.

Multi-factor authentication (MFA) is implemented to verify the identity of eligible voters, utilizing OTP-based authentication, Aadhaar/ID verification (when applicable), and optional biometric verification. A unique encrypted token is assigned to each voter after successful authentication to prevent duplicate voting, and this token is securely stored in the database.

To protect sensitive voter data, the platform integrates state-of-the-art encryption (AES-256) and hashing techniques (SHA-256, bcrypt), ensuring data integrity and security. Digital signatures and timestamps are used for vote verification, while blockchain technology can be optionally integrated to create an immutable, decentralized ledger of votes, enhancing transparency and trust.

The frontend, built with HTML, CSS, and JavaScript, offers a user-friendly interface for easy voting. Real-time election results are enabled through WebSockets and AJAX while maintaining confidentiality. Role-based access control (RBAC) ensures that only authorized personnel can manage elections and monitor results.

1. INTRODUCTION

The digital transformation of various sectors has led to the introduction of online systems that aim to increase efficiency, accessibility, and security. One such area that has seen significant advancements is the voting process. Traditional voting systems, although widely trusted, face numerous challenges such as fraud, voter manipulation, and lack of transparency. These issues call for the development of a modern, secure, and transparent online voting platform that can effectively address these concerns while ensuring voter privacy and data integrity.

The "Secured Digital Voting System" is an innovative web-based platform designed to revolutionize the way elections are conducted. It utilizes cutting-edge technologies like Flask for backend development, MongoDB for secure data storage, and advanced encryption techniques to ensure the integrity of the voting process. The system prioritizes security, scalability, and transparency by leveraging multi-factor authentication (MFA), blockchain technology, and real-time updates of election results.

In this system, voters are authenticated through multiple layers of verification, including OTP, Aadhaar/ID verification, and optional biometric authentication, ensuring only eligible voters can cast their ballots. Each vote is protected by encryption, digital signatures, and timestamps, ensuring its authenticity and immutability. The addition of blockchain technology further strengthens the system by creating an immutable record of votes that is resistant to tampering.

With its user-friendly interface, real-time election results, and role-based access control, the Secured Digital Voting System offers a secure, reliable, and transparent solution to modern voting challenges, making it a promising tool for the future of democratic elections.

Existing System

Traditional voting systems, particularly paper-based voting and electronic voting machines (EVMs), have been widely used to conduct elections. Paper-based voting systems are common in many parts of the world, where voters cast their ballots on paper, and results are manually counted. While electronic voting systems have been introduced to streamline this process, they still rely on physical machines to collect and tally votes. Despite their advancements, these existing systems are often vulnerable to a variety of issues such as fraud, human error, and inefficiencies in the electoral process.

In many countries, the system of voting is centralized and heavily reliant on government-controlled infrastructure, making it susceptible to manipulation, delays, and discrepancies in vote counting. Additionally, existing systems typically do not offer transparency in terms of how votes are processed or counted, leading to a lack of trust among the public. Moreover, the security of voter information is often inadequate, leaving sensitive personal details exposed to potential threats.

Proposed system

The "Secured Digital Voting System" is an advanced web-based platform designed to overcome the limitations of traditional voting methods by providing a secure, transparent, and efficient voting experience. This system integrates cutting-edge technologies to ensure that the electoral process is tamper-proof, scalable, and accessible for all eligible voters.

2.REQUIREMENT ANALYSIS

Functional Requirement

Administrator Module

The Administrator module allows authorized personnel to manage and oversee the entire digital voting process.

Functionalities include:

- **Manage Elections:** Create and configure new elections, including selecting candidates, voting timelines, and election rules.
- **User Access Management:** Monitor and control voter registration, verify voter eligibility, and manage access rights.
- **Monitor Election Progress:** View real-time updates on voter participation, vote counts, and election status.
- **View Election Results:** Access detailed election results, including vote breakdowns and summaries.
- **Generate Reports:** Create comprehensive reports on election outcomes, voter participation, and security logs.
- **Role-Based Access Control (RBAC):** Manage different levels of access for system administrators, election managers, and other personnel.

Voter Module

The Voter module allows eligible users to securely cast their votes and interact with the system.

Functionalities include:

- **Registration and Authentication:**
  - Register voters and authenticate them using multi-factor methods (OTP, Aadhaar/ID verification, optional biometric).
  - Ensure secure, one-time voter authentication.
- **Cast Vote:**
  - Provide an easy-to-use interface for voters to select and submit their vote for chosen candidates.

- Ensure votes are cast securely and stored in an encrypted format.
- **View Confirmation:**
  - Display a confirmation of the voter's choice along with a unique, encrypted token for verification.
- **Track Voting Status:**
  - Enable voters to view their voting status (e.g., whether they have successfully voted).

Real-Time Monitoring Module

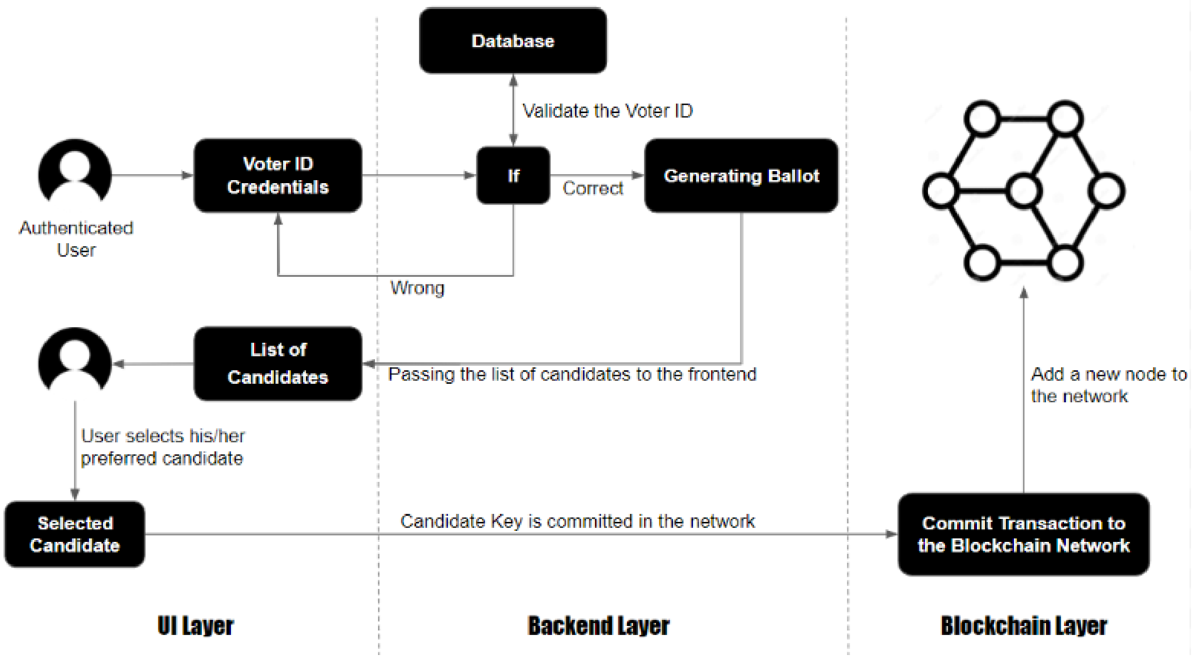
This module provides real-time insights into the election process, designed for administrators and election monitors.

Functionalities include:

- **Live Vote Tallying:**
  - Display live updates of votes as they are cast, showing results in real-time while maintaining confidentiality.
- **Monitor Vote Integrity:**
  - Track voting data integrity, ensuring no duplication or tampering occurs.
- **Security Logs:**
  - Record and display security events related to voter authentication, vote casting, and data access.

3.DESIGN

System Architecture


Database
Validate the Voter ID
Authenticated User
Voter ID Credentials
If
Correct
Generating Ballot
Wrong
List of Candidates
Passing the list of candidates to the frontend
Add a new node to the network
User selects his/her preferred candidate
Selected Candidate
Candidate Key is committed in the network
Commit Transaction to the Blockchain Network
UI Layer
Backend Layer
Blockchain Layer

3.1 System architecture

System Architecture of a Secure Digital Voting System

A secure digital voting system plays a vital role in ensuring fair, transparent, and tamper-resistant elections. The architecture is composed of several components and logical layers that work together to authenticate users, allow secure vote casting, ensure vote integrity, and deliver verifiable results.

1. Voter Authentication and Data Collection

Voter information such as Voter ID, Aadhaar number, or biometric data is collected during the login or registration phase. This data is crosschecked against official records to validate the voter's eligibility and prevent fraudulent access to the system.

2. Ballot Generation and Vote Casting

After successful verification, a digital ballot specific to the voter's constituency is generated. This ballot is displayed on the user interface, where the voter can select a preferred candidate. Once the selection is made, the vote is encrypted and securely submitted to the backend.

3. Vote Verification and Token Management

Each vote is attached to a unique encrypted token that ensures vote authenticity and prevents duplication. These tokens are stored temporarily and help confirm that a valid vote has been cast without linking it back to the voter's identity, maintaining anonymity.

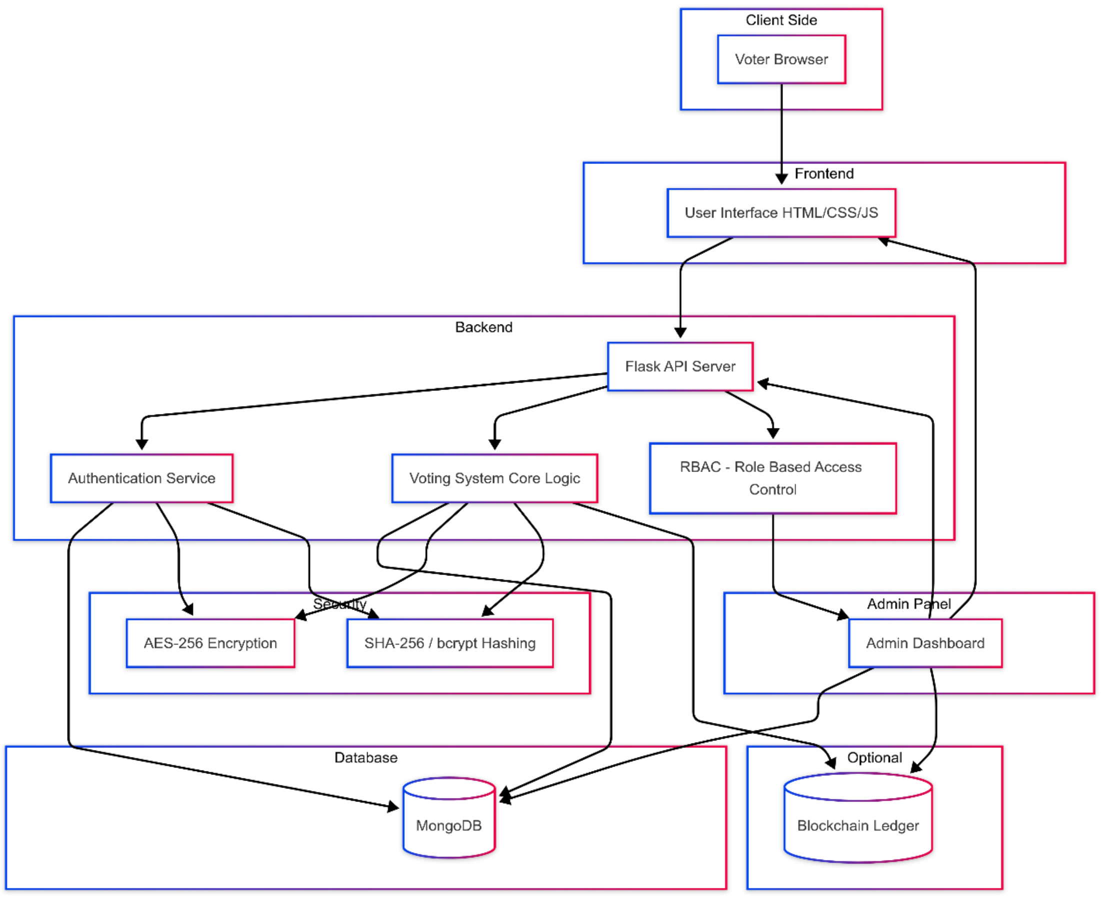
4. Vote Storage and Blockchain Integration

The encrypted votes are stored in a blockchain ledger. Each vote is added as a new block containing transactiondetails, timestamps, and cryptographic hashes. This decentralized storage ensures the vote cannot be modified or deleted, providing a secure and auditable trail of all votes cast.

5. Result Generation and Monitoring

Once voting concludes, election administrators can log in and trigger vote counting. The system compiles results from the blockchain and presents a complete report, including total votes per candidate, turnout statistics, and audit logs. Monitoring dashboards support real-time observation of the election process by authorized personnel.

Technical Architecture



3.2 Technical Architecture

4.IMPLEMENTATION

Flask Framework:

Flask is a lightweight, Python-based web framework that acts as the backend for the **Secured Digital Voting System**, handling all server-side logic, authentication, vote processing, and result computation.

Key Features of Flask:

Ease of Development:

Enables rapid development of secure, scalable web applications with minimal boilerplate code.

Integration with Authentication and Encryption:

Simplifies the integration of multi-factor authentication (MFA) and encryption techniques like AES-256 for securing votes and voter data.

Database Interaction:

Facilitates smooth communication with **MongoDB** for storing voter details and encrypted votes.

Real-time Updates:

Integrates seamlessly with technologies like **WebSockets** and **AJAX** for real-time vote tallying and result updates.

By using **Flask**, the application can efficiently handle user authentication, vote submission, and data encryption, making it a perfect choice for secure and scalable voting applications.

Development Environment

The development environment is optimized for building and deploying the **Secured Digital Voting System**, ensuring efficiency, scalability, and security.

Programming Language: **Python**

Core Development:

Python serves as the primary language for implementing the core functionalities of the system, including voter authentication, vote encryption, and backend processing.

Scalability and Versatility:

Python's simplicity and powerful libraries allow for scalable, secure, and maintainable code, making it ideal for large-scale systems like digital voting platforms.

Integrated Development Environment (IDE): **Visual Studio Code**

Development Efficiency:

VS Code provides a seamless environment for writing, testing, and debugging Python code. Its support for Flask and MongoDB extensions enhances productivity.

Real-Time Code Preview:

The IDE supports live previews and immediate testing, ensuring that the system's components are continuously integrated and tested.

Version Control:

Built-in Git support allows easy version management, ensuring smooth collaboration between team members.

Customizable Workspaces:

Developers can tailor their workspaces to streamline tasks, such as Flask development, database management, and encryption setup.

Additional Tools and Features:

Virtual Environments:

Virtual environments are used to isolate project dependencies and manage different Python libraries without causing conflicts with system-wide installations.

Debugging and Testing:

VS Code's powerful debugging tools and testing support make identifying issues and ensuring code quality efficient.

Documentation Support:

The IDE also integrates well with tools for maintaining project documentation, ensuring that the development process is well-documented and transparent.

Libraries Used:

Pandas: For processing and organizing voter data, vote details, and election metadata.

Joblib: For efficiently loading and saving machine learning models (if used for vote prediction or result analysis).

Flask: For creating the backend system, handling authentication, vote encryption, and real-time result updates.

PyCryptodome: For implementing AES-256 encryption to secure vote data.

MongoDB Driver (PyMongo): For interacting with MongoDB to store user data and encrypted vote records.

5-CONCLUSION

The Secured Digital Voting System provides a robust and reliable solution for online voting, ensuring a transparent, secure, and fraud-resistant voting process. By leveraging advanced technologies like multi-factor authentication, AES-256 encryption, and blockchain (optional) for immutable voting records, the system guarantees the integrity and authenticity of each vote.

The integration of Flask for backend development and MongoDB for secure data storage, along with an intuitive user interface built using Streamlit, ensures that voters can easily participate in elections while maintaining a high level of security. Real-time result updates via WebSockets and AJAX contribute to an interactive and efficient election process.

In conclusion, this system meets the highest standards of security, transparency, and user-friendliness, making it an ideal solution for modern, digital voting platforms. It is scalable, adaptable to various election types, and capable of preventing fraudulent activities, ensuring that every vote counts accurately and securely.

REFERENCES

1. **Pooja, N. P., & Anuja, T.** (2022). "Smart Voting System using Deep Learning Techniques." International Journal of Engineering Research & Technology (IJERT), ETEDM – 2022 (Volume 10 – Issue 08).
2. **Sirenjeevi, P., Preethi, C., Shaminee, S., & Gayathri, J.** (2022). "Smart Voting System using Deep Learning Techniques and Facial Authentication." International Journal of Engineering Research & Technology (IJERT), ICONNECT – 2022 (Volume 10 – Issue 09).
3. **Kavitha, K., & Khanam, M. H.** (2022). "Smart Voting System Using Deep Learning." International Journal of Research and Analytical Reviews (IJRAR), Volume 9, Issue 2.
4. **Choudhary, N., Agarwal, S., & Lavania, G.** (2019). "Smart Voting System through Facial Recognition." International Journal of Scientific Research in Computer Science and Engineering, Vol.7, Issue.2, pp.7-10.
5. **Nisha, P. P., & Anuja, T.** (2022). "Revolutionizing E-voting Systems With Facial Recognition Using Machine Learning and Deep Learning for Improved Identity Verification and Security." International Journal of Engineering Research & Technology (IJERT), ETEDM 2022 (Volume 10 Issue 08).
6. **Swasthik, N., & Murugan, R.** (2023). "An Efficient Smart Voting System through Facial Recognition." International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), Volume 12, Issue 4.
7. **Khaiyum, S., & Samitha, K.** (2022). "Revolutionizing E-voting Systems With Facial Recognition Using Machine Learning and Deep Learning for Improved Identity Verification and Security." International Journal of Engineering

Research & Technology (IJERT), ETEDM 2022 (Volume 10 Issue 08).

8. **Nisha, P. P., & Anuja, T.** (2022). "Smart Voting System using Deep Learning Techniques." International Journal of Engineering Research & Technology (IJERT), ETEDM – 2022 (Volume 10 – Issue 08).