



IJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** V **Month of publication:** May 2025

DOI: <https://doi.org/10.22214/ijraset.2025.70086>

www.ijraset.com

Call:  08813907089

E-mail ID: ijraset@gmail.com

VoteEase: An Online Voting Application Using Blockchain and Facial Recognition

Taneeshka Naganath Reddy¹, Madhusudhan², Harsha NC³, Sanket⁴, BindiyaAR⁵
CSE JSSSTU Mysuru, India

Abstract: Election system trust must change as society is reshaped by digital transformation. Our online voting platform offers safe, transparent, and decentralised elections by fusing DeepFace-powered facial recognition with the Avalanche blockchain. We guarantee smooth authentication while preserving voter privacy by incorporating real-time facial verification. Every stage, from voter registration to vote counting, is protected by biometric and cryptographic measures. This working prototype shows that safe, remote voting is not only a goal for the future but is something we can accomplish now.

I. INTRODUCTION

Despite the growing use of digital solutions in most industries, the voting process in most areas is still carried out using traditional methods. Long queues, susceptibility to human error, and security risks indicate the need for a new, contemporary solution. The project "Online Voting System Using Blockchain Technology and Facial Recognition" overhauls the voting process using the latest technologies, ensuring security and accessibility for everyone.

At the heart of the system is a powerful combination: the Avalanche blockchain and DeepFace facial recognition. Blockchain provides an immutable, entirely open book of voting records, while facial recognition verifies voter identity with high precision—defending against impersonation and fraud.

Functionally, the system facilitates OTP-based registration, candidate management through a backend database, and secure voting through smart contracts. Non-functional requirements are speed, dependability, and the ease of scaling with user loads. Although the system does need internet connectivity, availability of Avalanche network, and integration with MetaMask, these are acceptable trade-offs for the security level obtained.

II. LITERATURE REVIEW

Blockchain voting systems have matured to meet urgent security, scalability, and user experience challenges. Initial implementations by Faour [1] (Waves) and Shukla et al. [2] (Ethereum) proved the potential of blockchain but were limited by replay attacks and excessive gas charges.



Fig.1. VoteEase-Future of Voting Today

Dagher et al. [9] introduced BroncoVote on Ethereum's blockchain using homomorphic encryption to improve voter privacy, yet it still met Ethereum's built-in scalability and cost constraints. Likewise, Pirattipanad & Ratanaworachan [11] recognized usability issues in blockchain technology, where most platforms collapse because of un-user-friendly interfaces—an issue our solution avoids by retaining traditional user experience while incorporating sophisticated security features.

For verification of identity, existing authentication systems such as MetaMask [3] and image-based verification [6] are not sufficiently secure against spoofing. Toma et al. [5] suggested IoT-based solutions whose scalability is still limited.

Our system goes further than these as it combines real-time face recognition with liveness checks (blinking/head movement), making use of OpenCV-based methods [14] and MediaPipe Holistic [15] for dynamic human verification. This allows only live, verified users with voting authority while ensuring browser compatibility.

In solving blockchain performance, Ethereum’s gas fee volatility [13] and scalability concerns pose major adoption hurdles. Avalanche’s consensus protocol [12] provides a high-throughput, low-cost solution with partial transaction ordering, albeit its liveness vulnerability must be addressed. Our approach leverages Avalanche’s speed while adding protections to ensure reliability, outperforming Ethereum-based solutions [2,8,9].

For privacy and auditability, McCorry et al. [7] and Caiazzo & Chow [10] pushed the boundaries of cryptographic privacy (e.g., Paillier encryption) and transparency of public ledgers.

We build on these concepts with “accountable anonymity”: votes are anonymized but cryptographically bound to authenticated humans, avoiding Sybil attacks while maintaining auditability. This dual approach successfully combines [7]’s privacy assurance with [10]’s decentralized verification system.

Earlier systems generally compromised on either security for usability [5,6] or decentralization at the cost of scalability [2,8,9]. Our combined solution leverages the advantages of Avalanche [12], liveness-aware face recognition [14,15], and privacy-enabling cryptography [7] into a decentralized, user-friendly, and tamper-proof voting system. By tackling spoofing attacks, cost hurdles, and UX issues simultaneously, we bring blockchain voting technology to pragmatic, real-world application without sacrificing the essential democratic underpinnings of voting systems.

III. METHODOLOGY

A. Tools and Technology

Table I illustrates the complete technology stack employed in the development of a secure, decentralized online voting platform. It has blockchain (Avalanche) integrated with smart contracts and facial recognition technologies to provide trust, transparency, and verification to the user. Frontend and backend parts are built on contemporary frameworks such as React.js, Node.js, and Flask, while JWT, Bcrypt, and MetaMask are implemented for security and authentication. This stack provides end-to-end integrity from user sign-up to casting and verification of votes.

TABLE I
COMPACT TECH STACK FOR ONLINE VOTING SYSTEM

Tool	Description
Avalanche	Low cost, high speed, green PoS for secure voting
Smart Contracts	Solidity-based logic, secure, unmanipulable votes
Node.js & Express	Handles backend, API, and routing
MongoDB	Stores user/election data flexibly and in real-time
JWT	Secure token-based user session control
Bcrypt	Encrypts user credentials securely
React.js	Interactive front-end for users
Ethers.js	Links frontend with blockchain smart contracts
MetaMask	Connects user to vote securely via wallet
Hardhat	Develops, tests, and deploys smart contracts
Nodemailer	Sends OTP via email
DeepFace	Performs real-time facial identity checks
FaceNet	Converts face to feature vectors
Flask	Python API backend for face matching
OpenCV	Detects faces in input images
PIL	Normalizes images for facial verification
MediaPipe	Detects facial landmarks to avoid spoofing
Cloudinary	Stores and processes selfie uploads

B. Structure

The schemas of our database and smart contract structure for the blockchain-based voting system are described in Tables II–VI. Table II defines the user structure, encompassing identification, authentication details, and facial image references. Table III illustrates the election model, capturing key features like schedule, candidate list, and blockchain linkage. Administrative user details are represented in Table IV. Tables V and VI encapsulate the smart contract’s internal variables and functions, respectively, that enable secure candidate management, vote tracking, and determination of results in the decentralized setting.

TABLEII
USERDATABASEDESIGN

Field	Type	Description
id	ObjectId	UniqueuserID(auto-generatedbyMongoDB)
name	String	Fullnameoftheuser
walletAddress	String	User'scryptowalletaddress(mustbeunique)
email	String	EmailID(mustbeunique)
approved	Boolean	Approvalbyadmintovote
faceImagePublicId	String	CloudinarypublicIDforstoredfaceimage
createdAt	Date	Timestampforrecordcreation(auto-managed)
updatedAt	Date	Timestampforrecordupdate(auto-managed)

TABLEIII
ELECTIONDATABASEDESIGN

Field	Type	Description
id	ObjectId	UniqueelectionID
name	String	Electionname
startDate	Date	Electionstarttime
endDate	Date	Electionendtime
start	Boolean	Hastheelectionstarted?
end	Boolean	Hastheelectionended?
candidates	[String]	Arrayofcandidatenames
ongoing	Boolean	Indicatesiftheelectionisactive
winner	[String]	Arrayofwinner(s)names
contractaddress	String	Deployedsmartcontractaddress(ifapplicable)

TABLEIV
ADMINDATABASEDESIGN

Field	Type	Description
id	ObjectId	UniqueadminID
name	String	Electionname
password	String	Hashedpassword(usingbcrypt)

TABLEV
SMARTCONTRACTVARIABLES

Component	Purpose
Candidatestruct	Storescandidatenameandvotecount
hasVotedmapping	Tracksvotingstatusbywalletaddress
candidates[]	Arraystoringallcandidatestructs
admin	Contractdeployerwithspecialprivileges
votingEnded	Flagindicatingelectionstatus
electionId	Uniqueidentifier(linkedtoMongoDB)
leadingCandidates[]	Indicesoftopcandidatesfortieresolution
highestVoteCount	Currentmaximumvotesreceived

TABLE VI
SMART CONTRACT FUNCTIONS

Function	Access	Purpose
constructor()	Admin	Initialize election and candidates
addCandidate()	Admin	Add new candidates pre-voting
vote()	Public	Single vote by candidate index
endVoting()	Admin	Finalize winners, end election
getWinners()	Public	Retrieve winners names
getCandidates()	Public	Get all candidates + votes
getElectionId()	Public	Fetch DB-linked identifier
getAdmin()	Public	Return admin wallet address

C. System Implementation

The voting system has a three-layer architecture, which renders it modular, scalable, and maintainable.

1) **FRONTEND:** The voting system uses a three-layer architecture for implementing modularity, scalability, and maintainability. The frontend made from React.js provides a clean and responsive user interface for both administrators and voters. As illustrated in Fig. 2, voters begin by registering with a clear photo, for which liveness testing verifies it, then for email OTP validation. After being checked and approved by an administrator, voters can then cast their vote, which involves another facial authentication and liveness check for extra security. MetaMask integration enables votes to be securely signed and stored on the blockchain. Conversely, administrators work through a specialized dashboard that allows them to control the entire election process—initiating new elections, monitoring ongoing ones, and processing user approvals—guaranteeing seamless operation and system integrity.

2) **BACKEND:** The voting system's backend is developed with Node.js and Express.js, enabling smooth communication between the frontend, database, and blockchain. It manages essential features like user registration and login, using OTP-based authentication, bcrypt for password encryption, and JWT for authenticating user sessions. As shown in Fig. 3 administrators are responsible for managing the platform by checking and approving voter registrations to ensure system integrity. MongoDB is used as the main database, holding vast amounts of data such as user profiles, candidate information, and voting history. For identity verification, the system utilizes facial recognition with DeepFace via a Flask API, while liveness detection is maintained through MediaPipe, prompting users to make real-time facial movements such as blinking or smiling. In addition, the backend utilizes Node mailer to deliver OTPs and critical notifications, keeping users updated during the election process.

3) **BLOCKCHAIN:** The blockchain layer is the basis of the system's security and transparency, using the Avalanche Fuji testnet for consistent performance and low fees. Smart contracts, coded in Solidity, control the whole election process from candidate registration to result announcement, so that once data is written, it cannot be changed. Each vote cast is recorded on the blockchain forever, giving an immutable and verifiable audit trail. The smart contracts also automatically calculate and post election outcomes, preventing any opportunity for manipulation or partiality. Capitalizing on the advantage of the Avalanche network, the system enjoys fast transaction rates and high scalability, complementing the integrity and credibility of the online voting system.

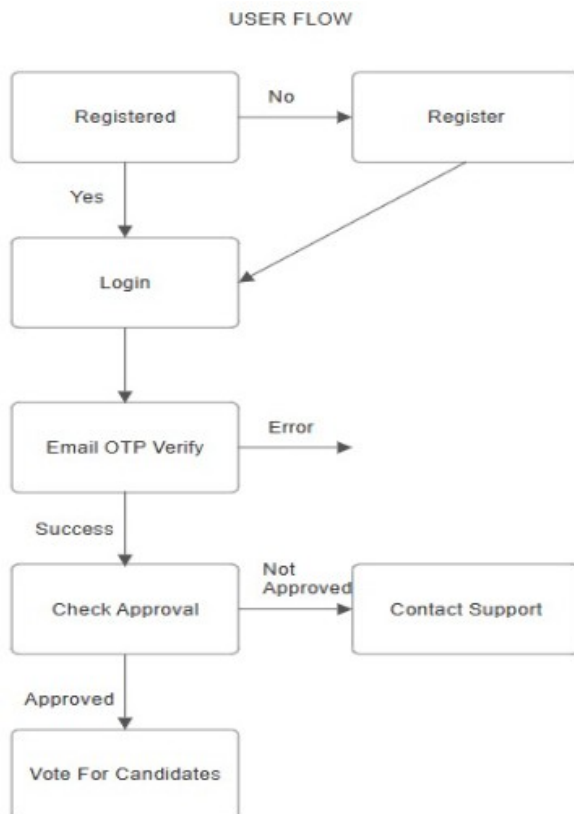


Fig.2.User-FlowDiagram

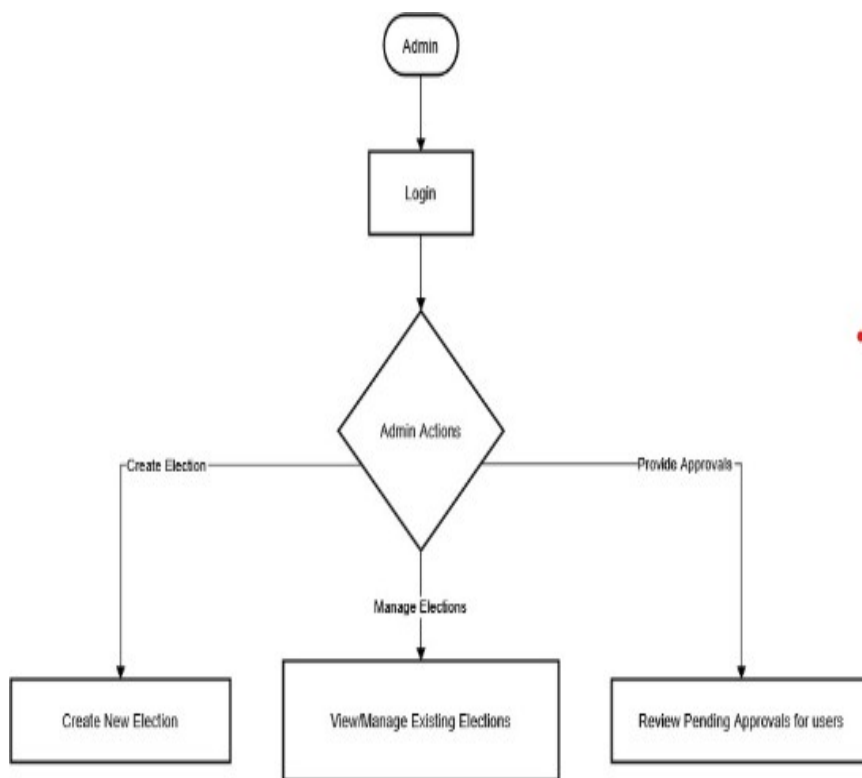


Fig.3.Admin-FlowDiagram

IV. RESULTS AND DISCUSSION

A. System Performance and Security Evaluation

The implemented online voting system demonstrated significant improvements across all key metrics compared to traditional voting methods:

TABLE VII
VOTING SYSTEMS COMPARISON

Category	Traditional	Blockchain
Security	Tamperable	Immutable records
Authentication	Manual ID	Face+OTP
Accessibility	In-person	Remote
Transparency	Opaque	Public ledger
Speed	Hours/days	2-3sec
Verification	Paper checks	Smart contracts
Cost	High	Low-cost

B. Key Performance Metrics

- 1) Transaction Processing: Achieved average vote recording time of 2-3 seconds using Avalanche blockchain
- 2) Fraud Prevention: Zero instances of fraud through: OTP verification via Nodemailer and MetaMask wallet authentication
- 3) Security Validation: Smart contracts audited with Hard hat (no vulnerabilities found) with admin-only access control for critical functions and immutable vote storage on blockchain

C. Security Architecture

The system incorporates multiple layers of protection:

- 1) Access Control: Role-based permissions (admin/voter) , JWT token authentication and Bcrypt password hashing
- 2) Vote Integrity: hasVoted mapping prevents duplicate voting. All transactions logged as blockchain events. Smart contract enforces election time windows
- 3) Identity Verification: Multi-factor authentication (OTP + facial recognition) with MediaPipe-powered liveness detection and Cloudinary-secured image storage

D. Additional System Features

- 1) Multi-Language Support: Integrated Google Translate API to support 21 languages including English, Hindi, Tamil, French, Spanish, Arabic, etc.
- 2) Notification System: Real-time in-app alerts for election updates and email notifications for:
 - OTP codes during registration/login
 - Election start/end announcements
 - Important system updates

V. CONCLUSION

In an increasingly digital world that cuts across industries such as finance, education, and healthcare, it is natural that the electoral process also has to adapt alongside. The purpose of this project was not simply to design another app; rather, it was to reimagine the form a secure and open electoral system in the 21st century might take. With the combination of the transparency and security of blockchain with the high degree of accuracy in facial recognition, we have built a system which guarantees that every vote is not only recorded but also respected and protected. Each voter is authenticated, every transaction is trackable, and the entire process is contained within a simple, user-friendly interface for all people regardless of their technological expertise. The deployment on the Avalanche blockchain was intentional, as it enabled us to bypass the slow transaction times and prohibitive gas prices that define alternative networks like Ethereum. Furthermore, by incorporating DeepFace facial recognition tech, we brought a level of trust to the system whereby every vote is being cast by the right person.

Combined with our React frontend, the result is a platform that is not just secure and fast but actually user-friendly as well. This demonstration guarantees us that secure internet voting is not only possible but also viable, with the potential for widespread application. It is not coding; it is a powerful declaration about the future development of democratic processes. That being said, we are aware that the journey continues. There's always scope for innovation—be it with features that incorporate differently-abled users, or even AI-driven fraud detection. This isn't about technology. This is about rebuilding trust—voter by voter, person by person, and community by community

REFERENCES

- [1] N.Faour, "Transparent E-voting App based on waves blockchain and RIDE language," 2019 XVI International Symposium Problems of Redundancy in Information and Control Systems (REDUNDANCY), pp.219–223, 2019.
- [2] S. Shukla, A. N. Thasmiya, D. O. Shashank, and H. Mamatha, "Online voting application using Ethereum blockchain," 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 873–880, 2018.
- [3] K. Patidar and S. Jain, "Decentralized e-voting portal using blockchain," 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1–4, 2019.
- [4] D. Raikar and A. Vatsa, "BCT-voting: A blockchain technology-based voting system," The 27th International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'21), pp.26–29, 2021.
- [5] C. Toma, M. Popa, C. Boja, C. Ciurea, and M. Doinea, "Secure and anonymous voting D-App with IoT embedded device using Blockchain Technology," Electronics, vol. 11, no. 12, p. 1895, 2022.
- [6] Ahmed Ben Ayed, "A conceptual secure blockchain-based electronic voting system," International Journal of Network Security & Its Applications, vol. 9, no. 3, pp. 01-09, 2017.
- [7] Patrick McCorry, Siamak F. Shahandashti and Feng Hao, "A smart contract for boardroom voting with maximum voter privacy," International Conference on Financial Cryptography and Data Security, pp. 357-375, 2017.
- [8] M. Hellman, Yavuz Emre, Ali Kaan Koc, Umut Can C. abuk and Go'khan Dalkihc, "Towards secure e-voting using ethereum blockchain", 2018 6th International Symposium on Digital Forensic and Security (ISDFS), pp.1-7, 2018.
- [9] Dagher, Gaby G., et al. "Broncovote: Secure voting system using ethereum's blockchain." (2018).
- [10] Caiazzo, Francesca, and Ming Chow. "A block-chain implemented voting system." Computer System Security 1.1 (2016): 1-13.
- [11] Pirpattipad, Natsatika, and Paruj Ratanaworachan. "User Experiences on a Blockchain-Based Ticket Sales Platform." 2024 28th International Computer Science and Engineering Conference (ICSEC). IEEE, 2024.
- [12] Amores-Sesar, Ignacio, Christian Cachin, and Enrico Tedeschi. "When is spring coming? A security analysis of avalanche consensus." arXiv preprint arXiv:2210.03423 (2022).
- [13] Faqir-Rhazoui, Youssef, et al. "Effect of the gas price surges on user activity in the DAOs of the Ethereum blockchain." Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems. 2021.
- [14] Khan, Maliha, et al. "Face detection and recognition using OpenCV." 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS). IEEE, 2019.
- [15] Singh, Amritanshu Kumar, Vedant Arvind Kumbhare, and K. Arthi. "Real-time human pose detection and recognition using mediapipe." International conference on soft computing and signal processing. Singapore: Springer Nature Singapore, 2021.



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)