JSS Mahavidyapeetha
JSS SCIENCE AND TECHNOLOGY UNIVERSITY
SRI JAYACHAMRAJENDRA COLLEGE OF ENGINEERING
JSS Techincal Institutions Campus , Mysuru - 570006

"Vote-Ease: An Online Voting Application Using Blockchain And Facial Recognition"

A technical project report submitted in partial fulfillment of the award of the degree of

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING

BY

TANEESHKA NAGANATH REDDY
(01JCE21CS116)

MADHUSUDHAN
(01JCE21CS060)

HARSHA NC
(01JST22UCS410)

SANKET
(01JST22UCS430)

Under the guidance of

BINDIYA AR
Assistant Professor
Department of Computer Science & Engineering
JSS STU Mysore

2024-25

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

JSS Mahavidyapeetha
JSS SCIENCE AND TECHNOLOGY UNIVERSITY

CERTIFICATE

This is to certify that the work entitled "**Vote-Ease: An Online Voting Application Using Blockchain And Facial Recognition**" is a Bonafide work carried out by **Taneeshka Naganath Reddy, Madhusudhan, Harsha NC, Sanket** in partial fulfillment of the award of the degree of Bachelor of Engineering in Computer Science and Engineering for the award of Bachelor of Engineering by Sri Jayachamarajendra College of Engineering, JSS Science and Technology University, Mysuru, during the year 2024-2025. It is certified that all corrections/suggestions indicated during CIE have been incorporated into the report. The project report has been approved as it satisfies the academic requirements concerning the project work prescribed for the Bachelor of Engineering degree.

Under the guidance of	**Head of the Department**
Bindiya A R	**Dr. Srinath S.**
Assistant Professor	Assoc. Prof and HOD
Dept of CS & E	Dept of CS & E
JSS STU, Mysuru -06	JSS STU, Mysuru -06

Name of Examiner	Signature with Date
1. ………………………	………………………….
2. ………………………	…………………………..
3. ………………………	…………………………..

CERTIFICATE OF PLAGIARISM CHECK

ORIGINALITY REPORT

10%	8%	5%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.ijraset.com Internet Source	2%
2	link.springer.com Internet Source	1%
3	Submitted to The University of Wolverhampton Student Paper	<1%
4	Submitted to Middlesex University Student Paper	<1%
5	Submitted to CSU, San Jose State University Student Paper	<1%
6	Submitted to University of Wales Institute, Cardiff Student Paper	<1%
7	drops.dagstuhl.de Internet Source	<1%
8	Submitted to Bournemouth University Student Paper	<1%
9	dr.ntu.edu.sg Internet Source	<1%
10	research.monash.edu Internet Source	<1%
11	"Soft Computing and Signal Processing", Springer Science and Business Media LLC, 2022	<1%

DECLARATION

We do hereby declare that the project titled **“VoteEase : An online voting application using Blockchain and Facial Recognition”** is carried out by **Taneeshka Naganath Reddy, Madhusudhan, Harsha NC And Sanket** under the guidance of **Bindiya A R** ,Assistant Professor, Department of Computer Science and Engineering, JSS Science and Technology University, Mysuru, in partial fulfilment of requirement for the award of Bachelors of Engineering by JSS Science and Technology University, Mysore, during the year 2024-2025.

We also declare that we have not submitted this dissertation to any other university for the award of any degree or diploma courses

Date:
Place: Mysore

Taneeshka Naganath Reddy(01JCE21CS116)
Madhusudhan(01JCE21CS060)
Harsha NC(01JST22UCS410)
Sanket(01JST22UCS430)

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.

ACKNOWLEDGEMENT

An endeavour is successful only when it is carried out under proper guidance and blessings. We would like to thank few people who helped us in carrying this work by lending invaluable assistance.We are grateful to Dr. C Nataraju, Principal, JSSTU, Mysuru and Dr. Srinath S, HOD, Department of Computer Science and Engineering, JSSSTU, Mysuru who encouraged us at this venture.It is our foremost duty to thank my project supervisor Bindiya A R for her encouragement, effective guidance and valuable suggestions right from the beginning of this project till its completion. We thank panel members for their support and guidance throughout the project. We also extend my regards to all the teaching and non-teaching members of Department of Computer Science and Engineering for their direct or indirect support towards the completion of this project.We would also like to thank our family and friends for their constant support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1. INTRODUCTION

1.1 PROBLEM STATEMENT

Despite technological advancement in fields like finance, he+althcare, and education, electoral systems in the majority of regions still rely on traditional, paper-based voting. These antiquated processes are prone to produce:

- Long lines at polling centers
- Counting and verification human error
- High vulnerability to security compromises and vote tampering
- Limited access for remote or disabled voters
- An antique procedure like this lacks the openness, speed, and convenience needed in today's democratic processes.

1.2 AIMS AND OBJECTIVES

Aim: To create and implement a secure, accessible, and transparent online voting system that utilizes blockchain and facial recognition technologies to identify voters and ensure election integrity.

Objectives:

- To utilize facial recognition (DeepFace) to verify voter identity in real-time.
- To use the Avalanche blockchain to provide tamper-proof vote records.
- To implement OTP-based voter registration to verify email ownership.
- To build a web-based platform using React.js and Node.js for convenience and scalability.
- To make use of smart contracts in order to autonomously perform election administration and results calculation.
- To implement MetaMask in order to ensure secure vote signing using wallet addresses.
- To have multiple language support through the Google Translate API.

1.3 APPLICATION AREAS

- National/State Elections: Enables safe remote voting with fraud prevention.

- University/College Elections: Simplifies forms and makes counting transparent.
- Private Companies/Organizations: Best used for board member voting, HR elections, or decision-making polls.
- Local Governance/Community Boards: Offers scalable and cost-effective solutions for voting.

1.4 EXISTING SOLUTIONS

Ethereum-based voting platforms (e.g., BroncoVote, Shukla et al.) provide immutability but are plagued by gas fee complexities and scalability problems. IoT and basic facial recognition methods lack liveliness detection and are subject to spoofing. Existing systems sacrifice user experience and require high technical literacy on the part of voters. Few platforms integrate both blockchain and biometric authentication into a seamless process.

1.5 PROPOSED SOLUTION

VoteEase has the following advantages as an end-to-end solution:

- Avalanche Blockchain: Clean energy usage, low cost, high speed.
- DeepFace + MediaPipe: Blink/smiling/head movement liveliness detection fused with face recognition.
- OTP + Facial Verification: Multi-factor authentication improves vote integrity.
- React.js + Google Translate API: Accessible and multilingual web frontend.
- Smart Contracts (Solidity): Clear, tamper-evident, automatically calculated outcomes.
- MetaMask Integration: Each vote is digitally signed, traceable, and genuine.

It eliminates cheating, makes it more convenient, and ensures that each vote counts—securely and reliably.

Fig. 1.1: VoteEase – Future of Voting Today

1.6 Gantt Chart

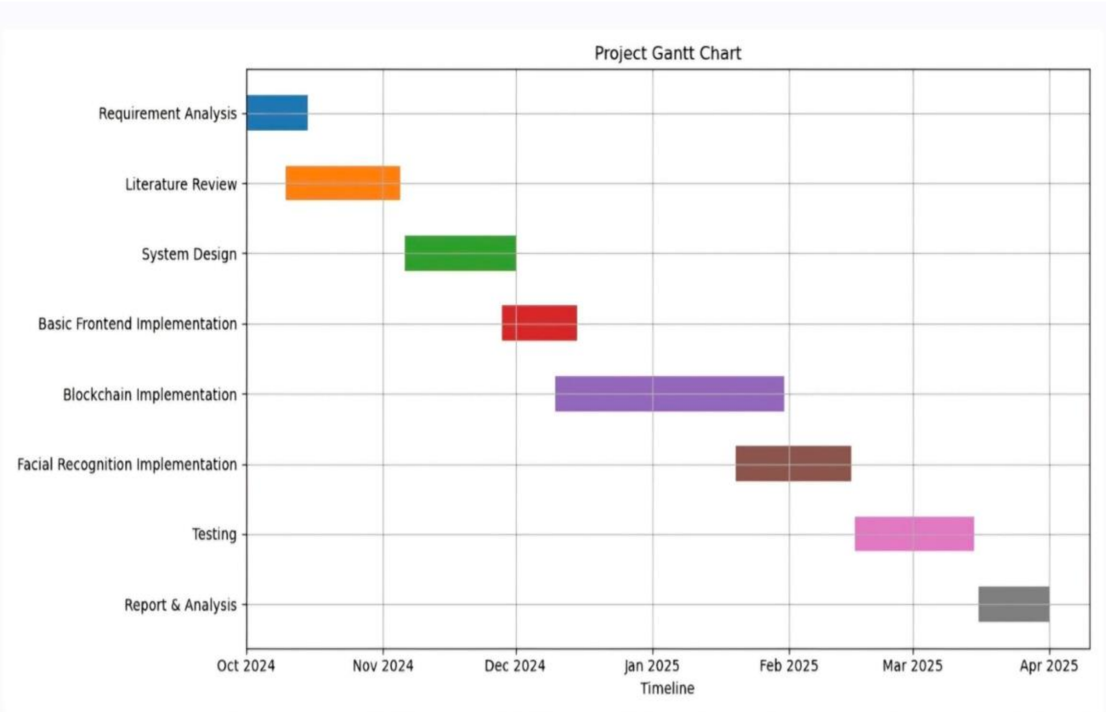

Fig. 1.2: Gantt Chart

CHAPTER 2. LITERATURE REVIEW

2.1 TRANSPARENT E-VOTING DAPP BASED ON WAVES BLOCKCHAIN AND RIDE LANGUAGE

Author(s): N. Faour

Source: 2019 XVI International Symposium Problems of Redundancy in Information and Control Systems (REDUNDANCY)

Findings: Faour's deployment demonstrated the application of the Waves blockchain to facilitate open e-voting through smart contracts in the RIDE language. The research emphasized how transparency and immutability can be realized in a decentralized way and how blockchain offers a solution alternative to centralized electoral processes. It also, however, referred to drawbacks like susceptibility to replay attacks and limited extensibility.

Technologies Used: Waves Blockchain, RIDE Language, Smart Contracts

Gap in Review: In comparison to VoteEase, this research does not have facial recognition, multi-factor authentication, and real-time identity verification. The research also employs a less established blockchain (Waves), which has less scalability and adoption prospects in comparison to Avalanche. The absence of the full-stack system with frontend/backend integration is also a major shortcoming of the prototype's usability and practicality.

2.2 ONLINE VOTING APPLICATION USING ETHEREUM BLOCKCHAIN

Author(s): S. Shukla, A. N. Thasmiya, D. O. Shashank, and H. Mamatha

Source: 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)

Findings: The e-voting application on the Ethereum platform showcased how decentralization can be properly leveraged to ensure vote immutability and auditability. The paper secured Ethereum's ability to conduct transparent, tamper-proof elections. It was not without problems such as high gas prices and low transaction speeds, which made it less ideal for mass deployments.

Technologies Used: Ethereum Blockchain, Smart Contracts, Web3.js

Gap in Review: VoteEase overcomes Ethereum's cost and speed constraints by employing Avalanche, providing faster and cheaper transactions. Additionally, the paper under reference does not have biometric authentication, liveliness detection, and an intuitive interface—all key aspects in VoteEase that enhance security and ease of use. The paper also did not consider fraud prevention through facial verification or multi-language capabilities.

2.3 DECENTRALIZED E-VOTING PORTAL USING BLOCKCHAIN

Author(s): K. Patidar and S. Jain

Source: 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)

Findings: This paper introduces a decentralized portal guaranteeing vote immutability and safe access via blockchain. The paper is centralized in focus but ambiguous regarding scalability and user experience. The study highlights how e-voting will minimize human involvement and election costs but is devoid of technical rigor in real-time implementation.

Technologies Used: Ethereum, Solidity, Blockchain-based DApp

Gap in Review: In comparison with VoteEase, this system leaves out real-time verification layers including OTP and face recognition. Whilst VoteEase involves DeepFace and MediaPipe as spoof-resistant implementations, this portal only uses foundational blockchain concepts. Furthermore, there is no discussion of backend implementation or wallet integrations like MetaMask, thereby making it not as robust.

2.4 BCT-VOTING: A BLOCKCHAIN TECHNOLOGY-BASED VOTING SYSTEM

Author(s): D. Raikar and A. Vatsa

Title: BCT-voting: A blockchain technology-based voting system

Source: The 27th International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'21)

Findings: BCT-voting brought an early architecture of blockchain voting with hashed credentials and voter registration processes. It's a proof of concept for blockchain voting that provides vote traceability and confidentiality. Nevertheless, the paper is not descriptive about real-time implementation, fraud detection processes, and user interface.

Technologies Used: Ethereum Blockchain, Voting Portal Interface

Gap in Review: VoteEase surpasses BCT-voting in incorporating dynamic authentication using facial recognition and liveliness checking. VoteEase also utilizes OTP authentication, admin verification, and multi-language, none of which are utilized by BCT-voting. This renders VoteEase more appropriate for actual usage, such as high-trust use in real life.

2.5 SECURE AND ANONYMOUS VOTING D-APP WITH IOT EMBEDDED DEVICE USING BLOCKCHAIN TECHNOLOGY

Author(s): C. Toma, M. Popa, C. Boja, C. Ciurea, and M. Doinea

Source: Electronics, vol. 11, no. 12, p. 1895, 2022

Findings: The authors designed a D-App that was coupled with IoT-based identification, suggesting security via decentralization and in-built verification. Nevertheless, it was not scalable because of the physical constraints of IoT devices and did not satisfactorily resolve concerns such as UI/UX design and mobile compatibility.

Technologies Used: Blockchain, IoT, D-Apps

Gap in Review: VoteEase avoids hardware constraints by taking advantage of browser-based technology with MediaPipe and Cloudinary, offering more scalability. VoteEase also improves verification through facial recognition and supports MetaMask for cryptographic wallet security—areas where the IoT model falls behind. The absence of real-time fraud resistance and admin control in the IoT model further emphasizes the superiority of VoteEase.

2.6 A CONCEPTUAL SECURE BLOCKCHAIN-BASED ELECTRONIC VOTING SYSTEM

Author(s): Ahmed Ben Ayed

Source: International Journal of Network Security & Its Applications, vol. 9, no. 3, pp. 01–09, 2017

Findings: Ayed presented a hypothetical e-voting framework that utilized blockchain to maintain vote integrity and anonymity. The system prioritized immutability, transparency, and trust of vote recording but did not conduct practical verification or user experience trials.

Technologies Used: Blockchain architecture, cryptographic hashing, electronic voter ledger

Gap in Review: While Ayed's system spoke about security basics, VoteEase builds on top of that with real-world application, dynamic identity verification with DeepFace, and anti-spoofing through MediaPipe. Ayed's theoretical system is missing OTP, UI design, and real-world deployment which VoteEase provides through a React frontend, Flask backend, and Avalanche blockchain.

2.7 A SMART CONTRACT FOR BOARDROOM VOTING WITH MAXIMUM VOTER PRIVACY

Author(s): Patrick McCorry, Siamak F. Shahandashti, and Feng Hao

Source: International Conference on Financial Cryptography and Data Security, 2017

Findings: This study aimed at optimizing voter secrecy in boardroom settings using zero-knowledge proofs and cutting-edge cryptographic methods. The contract had total anonymity, with vote count integrity.

Technologies Used: Smart Contracts, Cryptographic Voting Protocols, Zero-Knowledge Proofs

Gap in Review: VoteEase instead follows a practical path with "accountable anonymity" that strikes a balance between identifying someone and keeping the vote private. While McCorry's solution sacrifices usability for privacy, VoteEase does not. VoteEase is designed for public elections with facial authentication, while the aforementioned system is better applied to small, internal environments.

2.8 TOWARDS SECURE E-VOTING USING ETHEREUM BLOCKCHAIN

Authors: M. Hellman, Y. Emre, A. K. Koç, U. C. Çabuk, G. Dalkıhç

Source: 2018 6th International Symposium on Digital Forensic and Security (ISDFS)

Findings: This research investigated secure e-voting using Ethereum. The authors implemented a smart contract to manage the casting of votes, counting them, and publishing results. Despite this, they faced high gas prices, scaling problems, and transaction delays that rendered the Ethereum network unsuitable for practical deployment.

Technologies Used: Ethereum Blockchain, Solidity Smart Contracts, IPFS (for vote storage)

Gap in Review: VoteEase specifically answers the limitations emphasized in this paper. By leveraging the Avalanche network, it enjoys quicker processing (2–3 seconds per transaction) and very low fees. Ethereum's uncertainty of gas price defeats scalability—addressed in VoteEase using Avalanche's DAG-based consensus. Additionally, this paper does not have biometric identity verification checks and interface design, both being VoteEase's strengths. The addition of DeepFace, OTP authentication, and a React.js frontend raises VoteEase to the level of a complete voting product from being a concept demo.

2.9 BRONCOVOTE: SECURE VOTING SYSTEM USING ETHEREUM'S BLOCKCHAIN

Author: Dagher, Gaby G., et al.

Source: (2018)

Findings: BroncoVote improves vote secrecy through homomorphic encryption and votes are stored on Ethereum's blockchain. It aims to make vote contents private but verifiable. The system does mitigate some replay attacks, but still faces Ethereum's known cost and speed constraints.

Technologies Used: Ethereum, Homomorphic Encryption, Solidity

Gap in Review: BroncoVote is cryptographic-strong, yet it does not have the live identity verification VoteEase has. VoteEase introducing DeepFace, MediaPipe, and OpenCV for biometric authentication, no impersonation—for which BroncoVote does not account—is allowed. VoteEase also prevents the performance degradation by Ethereum using the higher TPS and green PoS consensus from Avalanche. While BroncoVote's encryption has strong conceptual prowess, VoteEase combines both crypto-strength along with user comfort, particularly on heterogeneous, open elections.

2.10 A BLOCKCHAIN-IMPLEMENTED VOTING SYSTEM

Authors: Francesca Caiazzo and Ming Chow

Source: Computer System Security Journal, 2016

Findings: This initial paper offers a voting system that has been applied on a public blockchain. It promotes decentralized verification and an unalterable ledger. The paper suggests essential features such as transparency and accountability but is not modulated and lacks real-world user management capabilities.

Technologies Used: Blockchain, Basic cryptographic hashing, Decentralized record keeping

Gap in Review: Caiazzo and Chow established groundwork in blockchain-based voting but did not address user authentication, liveliness detection, or front-end UX design. VoteEase extends their work and adds to it with facial recognition, OTP login, React.js UI, and complete backend integration with Flask and Node. Additionally, Caiazzo's solution was theoretical and failed to involve admin workflows or actual deployment issues—VoteEase makes the concept real with a proven system, smart contract functions, and role-based access controls.

2.11 USER EXPERIENCES ON A BLOCKCHAIN-BASED TICKET SALES PLATFORM

Authors: Pirpattipanad, Natsatika & Ratanaworachan, Paruj

Source: 2024 28th International Computer Science and Engineering Conference (ICSEC), IEEE

Findings: This paper investigates usability and interface design issues in blockchain-based systems, employing a ticketing system as a proxy. The research reveals that although blockchain is transparent, users find wallet integrations, transactions, and system feedback difficult to use, resulting in low adoption.

Technologies Used: Blockchain (platform unspecified), Wallet-based authentication, UI/UX feedback loops

Gap in Review: This paper's core theme—user experience—is well in line with VoteEase's design strength. VoteEase tackles these issues head-on: MetaMask is integrated seamlessly, and the UI is developed using React.js, prioritizing simplicity and accessibility. It further includes Google Translate API support for multilingual users, increasing inclusivity. VoteEase doesn't merely diagnose UX issues in the way the cited work does—it addresses them with actual design implementations that benefit elections, not merely ticketing.

2.12 WHEN IS SPRING COMING? A SECURITY ANALYSIS OF AVALANCHE CONSENSUS

Authors: Amores-Sesar, Ignacio, Cachin, Christian & Tedeschi, Enrico

Source: arXiv preprint arXiv:2210.03423, 2022

Findings: This study examines Avalanche's consensus protocol, which is characterized by both its high throughput and certain liveness vulnerabilities. The analysis is centered on theoretical attack vectors and finds areas where Avalanche's partial ordering would fail in highly adversarial environments

Technologies Used: Avalanche Consensus Protocol, Security modeling and attack simulation

Gap in Review: Whereas this paper warns against Avalanche's liveness in extreme scenarios, VoteEase responds to that by superimposing its design with biometric and admin verification processes. Not even a temporary failure of one node impacts the integrity of voter authentication (via facial recognition and OTP). VoteEase also does not simply depend on Avalanche mindlessly—it limits control-sensitive actions (such as closing elections) to smart contract-admin roles and time-locked voting sessions. The essay is negative about Avalanche's potential, but VoteEase shows its practical power in a real system.

2.13 EFFECT OF GAS PRICE SURGES ON USER ACTIVITY IN ETHEREUM DAOS

Authors: Faqir-Rhazoui, Youssef et al.

Source: Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems

Findings: This research indicates that the volatile gas fees of Ethereum impact user participation directly, leading users to withdraw from DAOs due to excessive charges. The uncertainty of gas costs erodes continuous participation.

Technologies Used: Ethereum, DAO activity tracking, Gas price analytics

Gap in Review: VoteEase prevents this issue by choosing Avalanche with its low, stable gas costs. While Ethereum-based platforms drive away users when prices spike, VoteEase's election platform remains accessible and cost-effective—vital for millions in democratic operations. VoteEase's smart contract gas consumption is also optimized and admin operations limited to necessary work. This research solidifies VoteEase's justification for choosing Avalanche over Ethereum.

2.14 FACE DETECTION AND RECOGNITION USING OPENCV

Authors: Khan, Maliha et al.

Source: 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), IEEE

Findings: This paper introduces a face recognition and detection system based on OpenCV. It demonstrates how facial features can be extracted and compared for basic identification purposes, like security access or tagging.

Technologies Used: OpenCV, Haar Cascades, Basic Euclidean face matching

Gap in Review: Khan's work is seminal but static recognition-only without liveliness detection or integration into real-world use. VoteEase builds upon this by employing DeepFace + MediaPipe Holistic to confirm not only who the voter is, but also that they're alive, blinking, and in attendance. Moreover, Khan's model is local/offline, whereas VoteEase runs securely in a web-integrated setting with real-time API verification, OTP, and cloud storage (Cloudinary). In essence, VoteEase brings OpenCV from a research experiment to a deployed, secure e-voting process.

2.15 REAL-TIME HUMAN POSE DETECTION AND RECOGNITION USING MEDIAPIPE

Authors: Singh, Amritanshu Kumar, Kumbhare, Vedant Arvind & Arthi, K.

Source: Springer Nature, International Conference on Soft Computing and Signal Processing

Findings: The application of MediaPipe for real-time detection of human pose landmarks in this paper explores human-computer interaction, which involves interpreting movements such as head nodding, blinking, or smiling for health monitoring and application in AR/VR.

Technologies Used: MediaPipe Holistic, Pose and facial landmark tracking, Python-based integration

Gap in Review: VoteEase utilizes the same pose-detection capability of MediaPipe but directs it toward preventing fraud—making sure that the voters are not employing photos, masks, or deepfakes. The paper is technically good but has no application in security areas. VoteEase takes advantage of these liveliness capabilities together with blockchain and web integration to create a multi-factor identity validation system. Singh et al. consider MediaPipe as a tool, while VoteEase converts it into a defense mechanism against electoral fraud.

CHAPTER 3. SYSTEM REQUIREMENTS AND ANALYSIS

3.1 FUNCTIONAL REQUIREMENTS

These outline the key operations the system needs to execute:

1.User Registration:

- Voter registers through email, uploads selfie, and enters wallet address.
- OTP to email for verification.
- Admin approves/rejects voter registration.

2. Voter Authentication:

- When voting, the system does real-time facial recognition and liveliness check.
- OTP verification assures ownership of email.
- MetaMask is needed to link wallet and authenticate transactions.

3. Election Management:

- Admins can create, track, and close elections.
- Admins can accept or reject voters.
- Voting and winner announcement are handled by smart contracts.

4.Vote Casting:

- Voter can vote only once in an election.
- Smart contract stores vote immutably on the Avalanche blockchain.

5. Result Announcement:

- After voting closes, results are computed and published on the blockchain.

6. Multilingual Support:

- Google Translate API enables users to see the interface in their desired language.

3.2 NON-FUNCTIONAL REQUIREMENTS

These specify the quality attributes of the system:

1. Security:

- Multi-level authentication (OTP + facial recognition + MetaMask).
- Smart contracts verified using Hardhat.
- Password storage via Bcrypt, session management via JWT.
- Performance:
- Completion of voting transactions within 2–3 seconds.
- Scalable for a large number of voters without slowing down.

2. Reliability:

- Immutability and fault tolerance ensured by blockchain-based records.
- Cloud-based selfie upload and email alerts add robustness.

3. Usability:

- Simple and responsive UI through React.js.
- Process of user onboarding remains intuitive even for non-technical users.
- Scalability:
- Based on Avalanche to handle high volumes of transactions at low expense.

3.3. SYSTEM CONSTRAINTS

1. Needs to run on stable internet connection.
2. Voters need to have crypto wallet (MetaMask).
3. Web browser needs to be capable of running MediaPipe and JavaScript.

3.4. ASSUMPTIONS

1. All voters are pre-informed and can use MetaMask and email OTP.
2. Admins are trusted parties with high privileges.
3. Voter images in Cloudinary are securely accessible using public IDs.

CHAPTER 4. TOOLS AND TECHNOLOGY

4.1 AVALANCHE BLOCKCHAIN

Avalanche was chosen as the core blockchain platform for this voting system due to its high throughput, low latency, and eco-friendly Proof-of-Stake (PoS) consensus mechanism. Its ability to process thousands of transactions per second ensures that the voting process remains smooth, secure, and scalable—essential features for any real-time decentralized application like online voting.

4.2 SMART CONTRACTS

Smart contracts, written in Solidity, are the heart of the voting logic. They automate essential processes such as adding candidates, recording votes, and retrieving results. Once deployed, these contracts guarantee transparency, immutability, and tamper-proof operations, ensuring that every vote counts—and stays counted. Development and testing were conducted using Hardhat, a powerful Ethereum-compatible environment.

4.3 NODE.JS & EXPRESS.JS

Node.js and Express.js were utilized to create a strong backend for dealing with all server-side functionality. This involves managing user sessions, processing API requests, and serving as the communication bridge between frontend and blockchain. RESTful APIs were created to process tasks like candidate data retrieval and user authentication.

4.4 MONGODB

MongoDB, a NoSQL database, is utilized to store critical data such as user information, candidate data, and system metadata. It enables rapid, consistent, and agile data access, which ensures smooth performance and correct data synchronization with the blockchain.

4.5 JWT (JSON WEB TOKENS)

To provide safe access, JWTs were used for authenticating users. The tokens authenticate every user's identity and permissions to ensure that only authenticated voters are able to vote using the voting system and vote. It's a tiny but powerful protector of your app's integrity.

4.6 BCRYPT

Security begins with secure credentials, and bcrypt is a big help by hashing user passwords prior to saving them in the database. This provides an added layer of security against unauthorized access, protecting user identities.

4.7 REACT.JS

React.js drives the application's frontend, providing a seamless, interactive, and responsive user interface. From rendering elections and candidates to wallet integration, React takes care of everything with flair—keeping the voting experience as intuitive and effortless as possible.

4.8 ETHERS.JS

Ethers.js connects the blockchain to the frontend, providing seamless integration of wallets, signing transactions, and communicating with smart contracts. It's strong yet light—ideal for a responsive decentralized application.

4.9 METAMASK

MetaMask is implemented on the platform to enable users to securely connect their wallets, handle tokens, and sign transactions. It is the user's portal to the blockchain, offering a comfortable and trusted interface for dealing with the voting system.

4.10 HARDHAT

Hardhat simplifies the development, deployment, and testing of smart contracts. It was heavily utilized to test voting situations on the Avalanche Fuji testnet to guarantee that all contracts operate smoothly prior to being deployed.

4.11 NODEMAILER

nodemailer assists in sending an otp to the user's email upon login and registration.

4.12 DEEPFACE

DeepFace is a strong Python-based face recognition system effortlessly integrated into our application to authenticate a voter's identity in the voting process. DeepFace supports multiple facial recognition models and backend detectors, giving users flexibility and accuracy. In our system, DeepFace guarantees that the individual casting the vote is actually the registered voter, providing an important layer of biometric security.

4.13 FACENET

Facenet is the deep model used by DeepFace in our project. Facenet maps face images to 128-dimensional embeddings, allowing comparison between two faces with high precision. Facetuned for reliability and high performance, Facenet is critical to deciding whether two faces are matches even under differing lighting, angle, and image quality, hence ideal for application in real-voting environments.

4.14 FLASK (PYTHON)

Flask is the light backend framework that drives our facial verification API. This Python micro web framework processes image requests by taking one image from the database (taken at registration) and another in base64 format (taken at voting). It then matches the two and returns an unambiguous, understandable result—whether the faces match or not—without filling the system to ensure real-time identity verification.

4.15 OPENCV

OpenCV is utilized as the backend face detector within the DeepFace system. It's tasked with finding the face in the image prior to entering the verification process. Praised for its speed and dependability, OpenCV makes sure that the system can recognize faces under varying lighting conditions and orientations of the face, even if the images are not aligned perfectly.

4.16 PIL (PYTHON IMAGING LIBRARY)

To ensure consistency and enhance precision when comparing faces, PIL is utilized to preprocess face images—particularly those arriving in base64 form. It converts images, resizes, and formats them such that inputs for the DeepFace model are as required and are of suitable quality. This is a crucial step to prevent mismatching as a result of differing image sizes.

4.17 MEDIAPIPE

MediaPipe's facial landmark detection behaves as a digital face lie detector, monitoring tiny movements such as blinks and smiles to detect photo or video spoofing. In real time and with pinpoint accuracy, it converts any camera into a security guard that can identify fake faces in a fraction of a second—no additional hardware required.

4.18 MONGODB

MongoDB takes care of all the behind-the-scenes data work—storing voter profiles, election information. Due to its malleable structure, we don't need to concern ourselves with fixed tables or complex schemas. It expands along with us, readily conforming when more users are added and more elections come on board, keeping everything in order and readily accessible in real-time.

4.19 CLOUDINARY

Cloudinary handles all voter selfies during registration.It makes uploading and retrieving images a breeze, without slowing things down. By taking care of image storage and optimization, Cloudinary helps us focus on what matters—making the voting experience smooth and secure.

CHAPTER 5. SYSTEM DESIGN

5.1 DATABASE DESIGN

Table 5.1 User Database

Field	Type	Description
_id	ObjectId	Unique user ID (auto-generated by MongoDB)
name	String	Full name of the user
walletAddress	String	User's crypto wallet address (must be unique)
email	String	Email ID (must be unique)
approved	Boolean	Approval by admin to vote
faceImagePublicId	String	Cloudinary public ID for stored face image
createdAt	Date	Timestamp for record creation (auto-managed)
updatedAt	Date	Timestamp for record update(auto-managed)

Table 5.2 Election Database

Field	Type	Description
_id	ObjectId	Unique election ID
name	String	Election name
startDate	Date	Election start time

endDate	Date	Election end time
start	Boolean	Has the election started?
end	Boolean	Has the election ended?
candidates	[String]	Array of candidate names
ongoing	Boolean	Indicates if the election is active
winner	[String]	Array of winner(s) names
contract_address	String	Deployed smart contract address (if applicable)

Table 5.3 Admin Database

Field	**Type**	**Description**
_id	ObjectId	Unique admin ID
name	String	Election name
password	String	Hashed password (using bcrypt)

5.2 SMART CONTRACT DESIGN

Table 5.4 Smart Contract variables

Component	**Purpose**
Candidate struct	Represents a candidate with their name and vote count

hasVoted mapping	Tracks whether a wallet address has already voted
candidates[]	Dynamic array to store all candidate structs
admin	The deployer of the contract who has exclusive control over certain actions
votingEnded	Boolean flag to mark the end of the election
electionId	Unique string to identify the election (linked to MongoDB)
leadingCandidates[]	Stores indices of leading candidates for tie handling
highestVoteCount	Tracks the most votes received by any candidate

Table 5.5 Smart Contract Functions

Function	**Access**	**Purpose**
constructor()	Admin-only	Initializes election, sets admin, adds candidates
addCandidate()	Admin-only	Adds new candidates before voting ends
vote()	Public	Allows users to vote once by index
endVoting()	Admin-only	Ends the voting period and finalizes the winner(s)
getWinners()	Public	Returns names of winning candidate(s) after voting ends
getCandidates()	Public	Returns full list of candidates and their votes

getElectionId()	Public	Returns the string ID for cross-checking with DB
getAdmin()	Public	Returns wallet address of the admin

5.3 SECURITY & TRANSPARENCY:

- Access Control: Only the admin can add candidates or end voting.
- Double Voting Prevention: hasVoted mapping ensures one vote per address.
- Immutable Results: Once voting is ended, results cannot be tampered with.
- Event Logging: Important actions emit events for on-chain transparency.

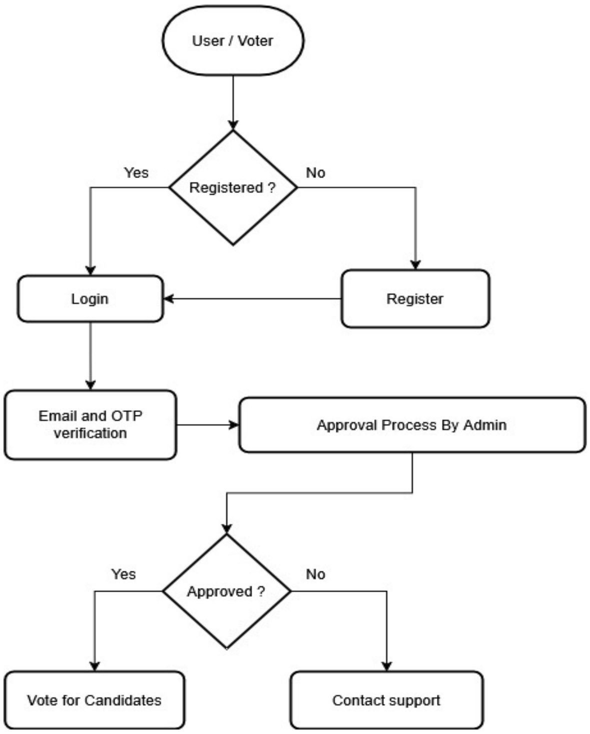

Fig. 5.1: User Flow Diagram

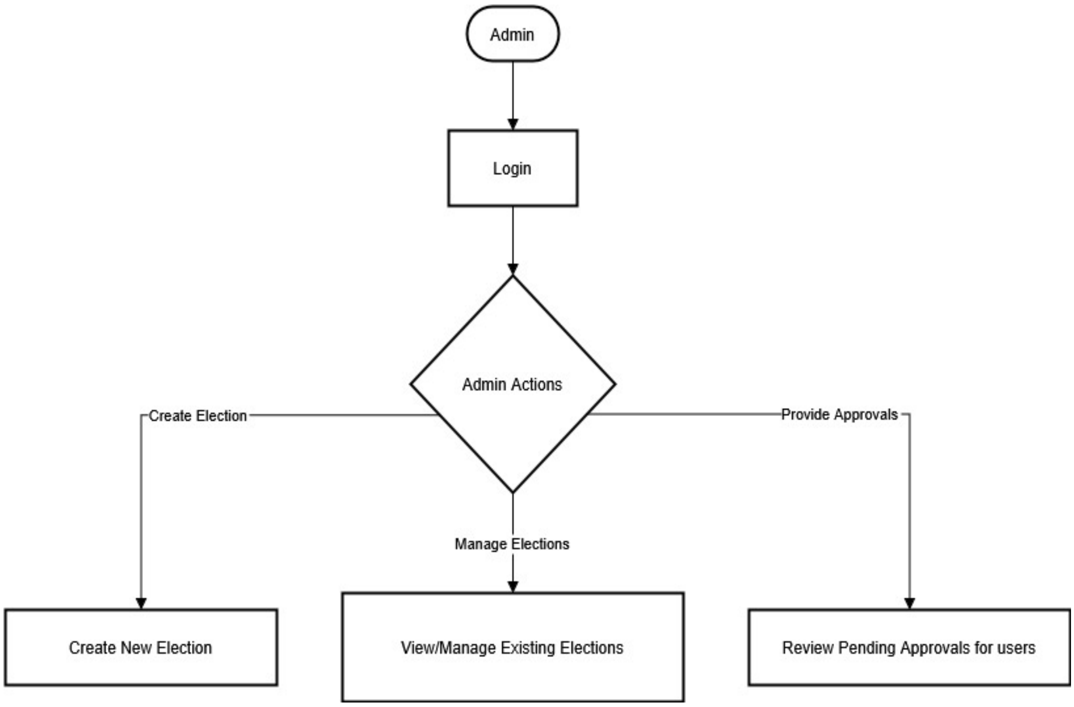

Fig. 5.2: Admin Flow Diagram

CHAPTER 6. SYSTEM IMPLEMENTATION

The voting system is designed using a three-tier architecture, making it modular, scalable, and easy to manage.

6.1 FRONTEND: THE USER INTERFACE

The frontend is crafted with React.js, making it intuitive, clean, and easy to navigate. It's the part of the system that all users will interact with, whether they're casting votes, managing candidate profiles, or overseeing elections.

- Voters: They can quickly register by taking a clear photo and their faces are verified by a liveliness check and then otp is sent to their emails and they can log in via otp. When the users cast their votes with a simple and easy-to-use interface they have to take a photo before clicking the vote button and that photo is compared with the one taken during registration along with a liveliness check. For added security, MetaMask is integrated, ensuring that each vote is securely recorded on the blockchain.
- Admins: Admins have a dashboard that lets them manage everything — from creating an election ,starting/ending an election to approving users.

6.2 BACKEND: THE LOGIC

The backend is the powerhouse of the system, built with Node.js and Express.js. It handles all the server-side operations and ensures smooth communication between the frontend, database, and blockchain.

● User Registration & Login: We've prioritized security, offering OTP-based verification to confirm that only legitimate users gain access. User passwords are securely hashed using bcrypt, and JWT tokens keep sessions safe.

● Admin Management: Admins are able to see and approve users. Only genuine people take part in the election, and hence the integrity of the system is not affected.

● Data Storage: User information, candidate information, voting information, and all other data related to the election are stored with MongoDB.

● Facial Recognition: For a secure voting process, we've integrated DeepFace with Flask. Voters are asked to verify their identity by comparing a live photo with the one taken during registration. This helps prevent impersonation and ensures only authorized voters can cast their votes.

● Liveliness Check: In order to keep only actual, physically present voters from voting, our system has MediaPipe's facial landmark detection. During registration and when voting, users will be asked to blink or smile—demonstrating they are not merely an image or clip. This extra layer prevents even advanced spoofing efforts without slowing down or being inconvenient.

● Email Notifications: Nodemailer sends otp on registration and login and also on the start and end of an election.

6.3 BLOCKCHAIN: THE BACKBONE OF SECURITY & TRANSPARENCY

The magic is at the blockchain level. Backed by the Avalanche Fuji testnet, it makes sure that every vote is stored immutably, thereby the process being tamper-proof and transparent.

- Smart Contracts: These are the brain of the voting system. Written in Solidity, the smart contracts manage everything — from adding candidates to recording votes and announcing results. Once deployed, the contract ensures all votes are safely stored on the blockchain and can never be changed. When admin starts an election , all details about the election are deployed as a smart contract on the avalanche blockchain.
- Vote Recording: Every time a voter casts a vote, it's recorded on the blockchain, ensuring full transparency and making it impossible for anyone to alter the results.
- Election Results: When the election concludes, the smart contract automatically calculates and announces the results, ensuring fairness and preventing manipulation.

- Security & Transparency: By using Avalanche, the system benefits from high-speed transactions and reduced costs (thanks to the testnet). Blockchain ensures that once a vote is cast, it’s final, making tampering impossible.

6.4 MULTI-LANGUAGE SUPPORT

- Added a language selection feature to allow users to interact with the application in their preferred language.
- We are using Google Translate embedding: a Google-powered language translator into our website.
- Language support : 21 languages
- English,Hindi,Tamil,Telugu,Malayalam,Bengali,Gujarati,Marathi,Punjabi,Kannada,Urdu,French,Spanish,German,Chinese,Japanese,Korean,Russian,Italian, Turkish,Arabic.

6.5 NOTIFICATIONS

- When an election is created , users get notified through the notification section.
- When an elections starts or ends, each user is sent an email on their registered email address.

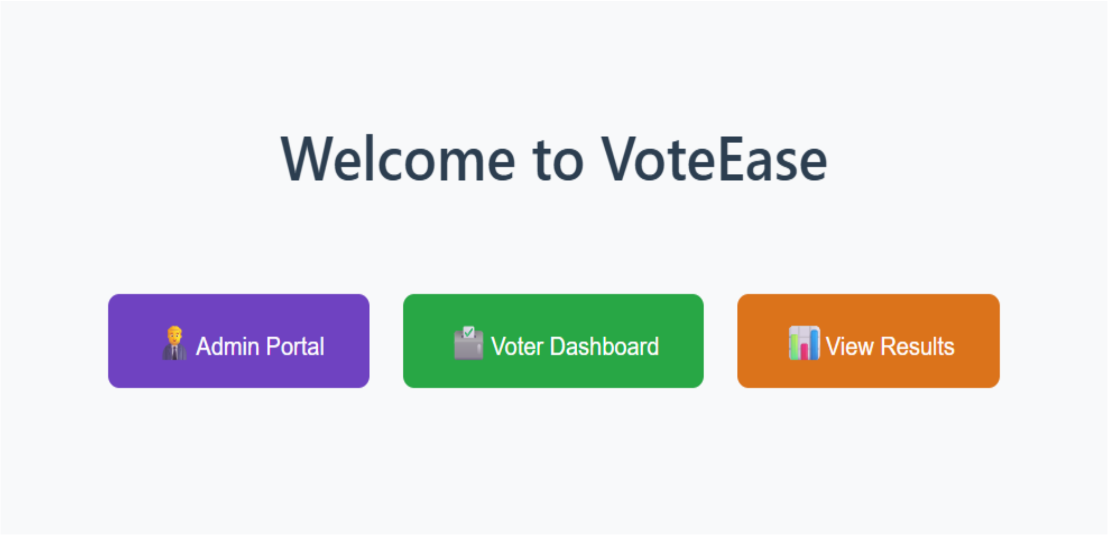

Fig. 7.1: Home Page (English)

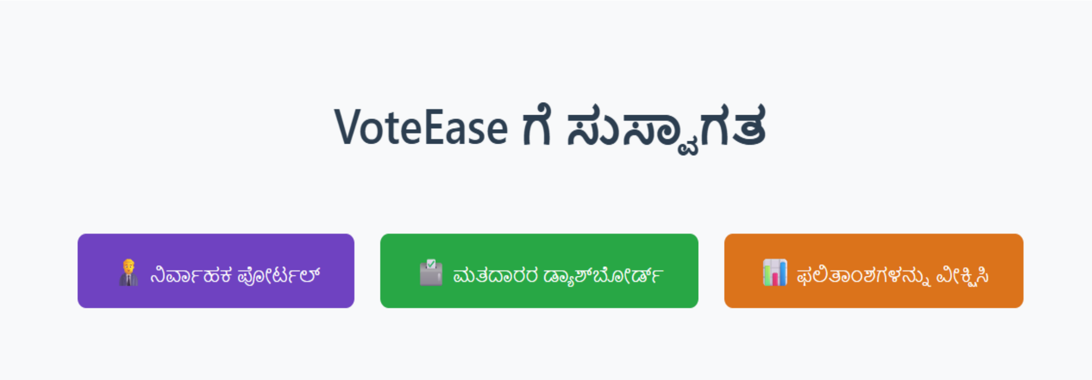

Fig. 7.2: Home Page (Kannada)

Fig. 7.3: Home Page (Hindi)

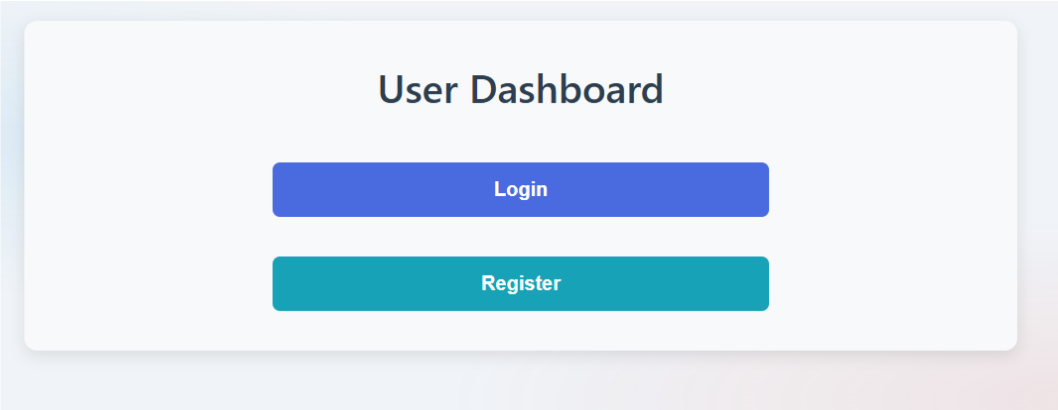

Fig. 7.4: User Dashboard

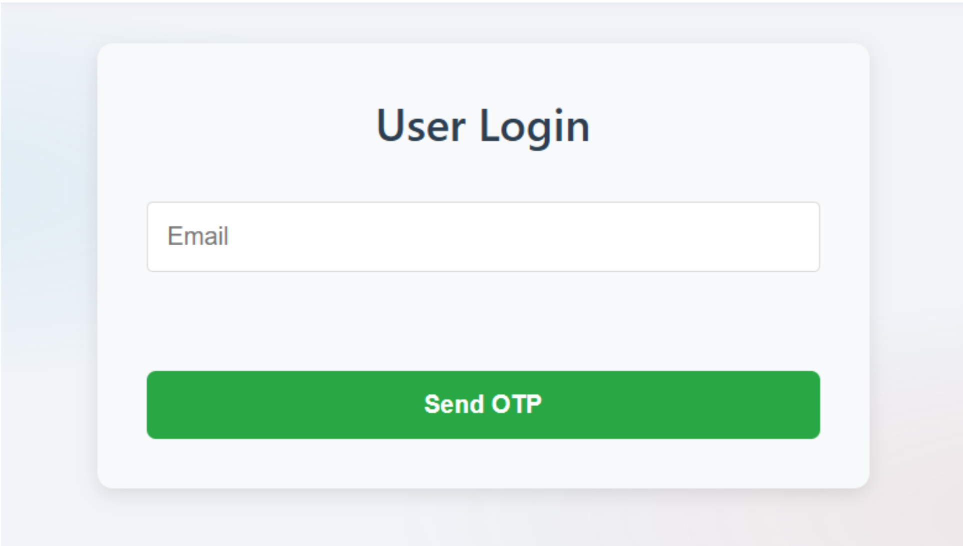

Fig. 7.5: User Login

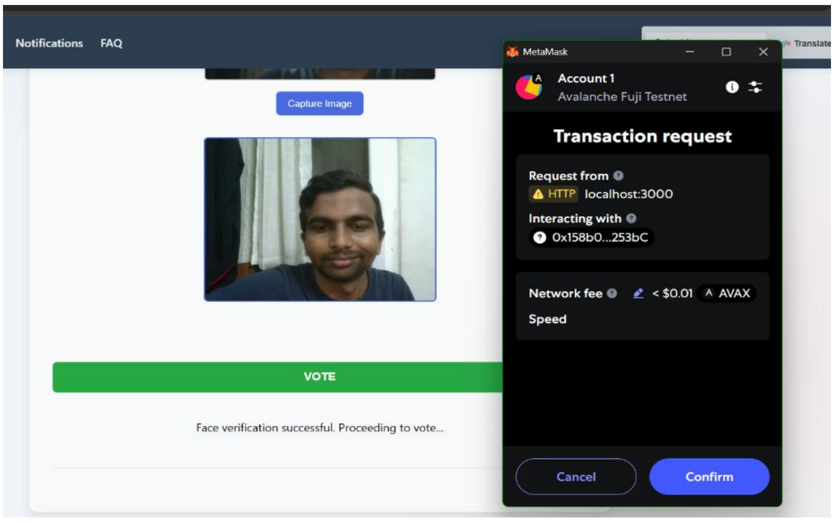

Fig. 7.6: Voting Process

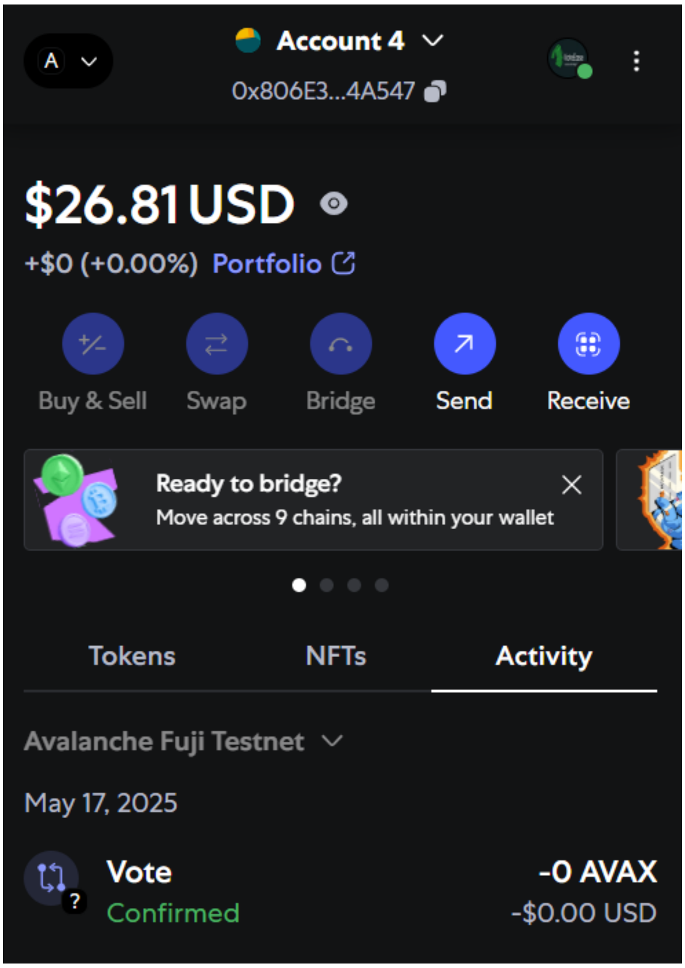

Fig. 7.7: MetaMask wallet

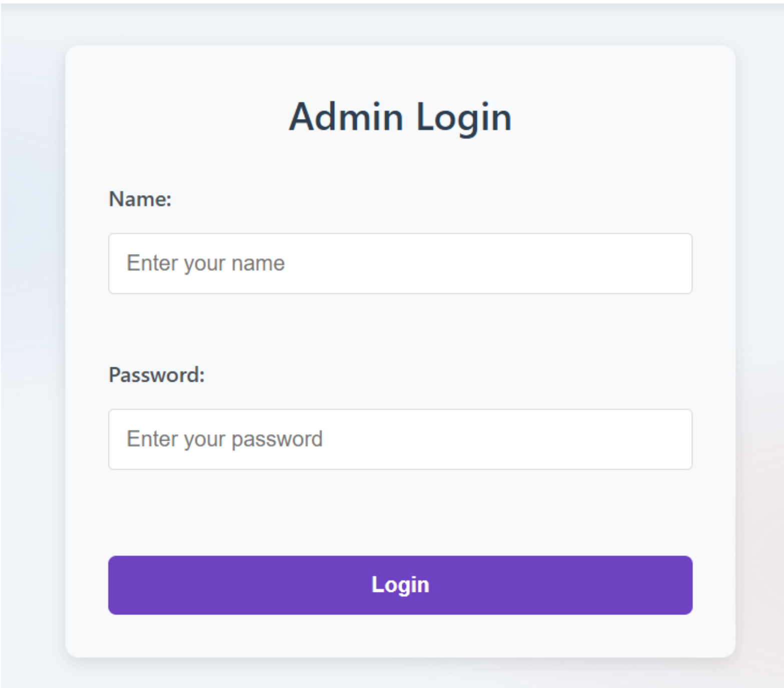

Fig. 7.8: Admin Login

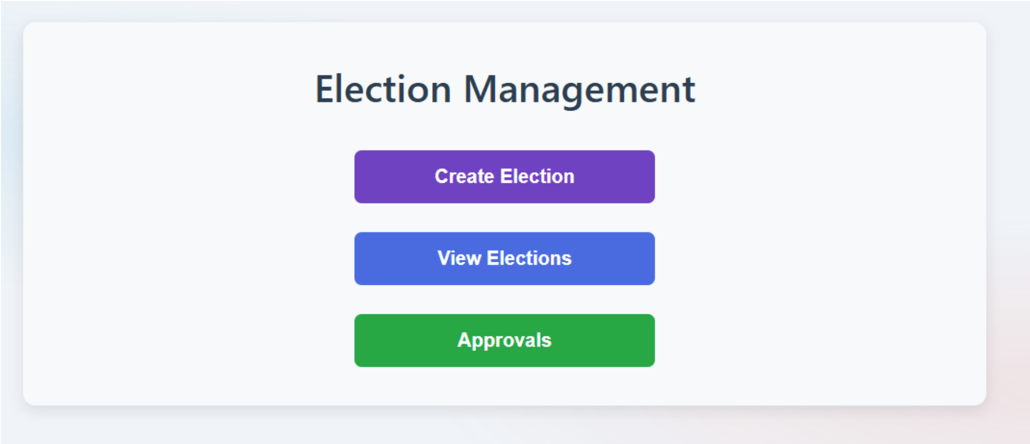

Fig. 7.9: Election Management

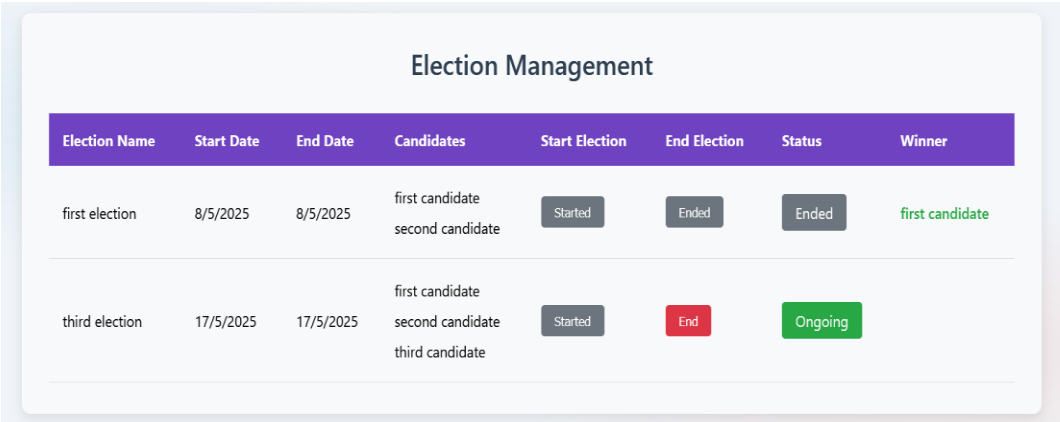

Fig. 7.10: View Elections

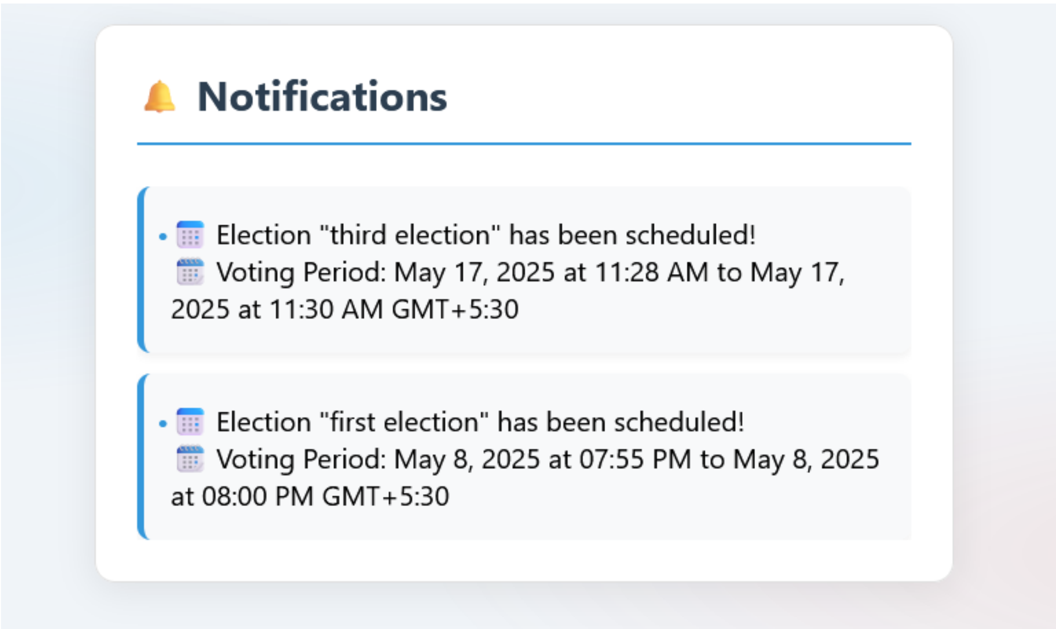

Fig. 7.11: Notifications

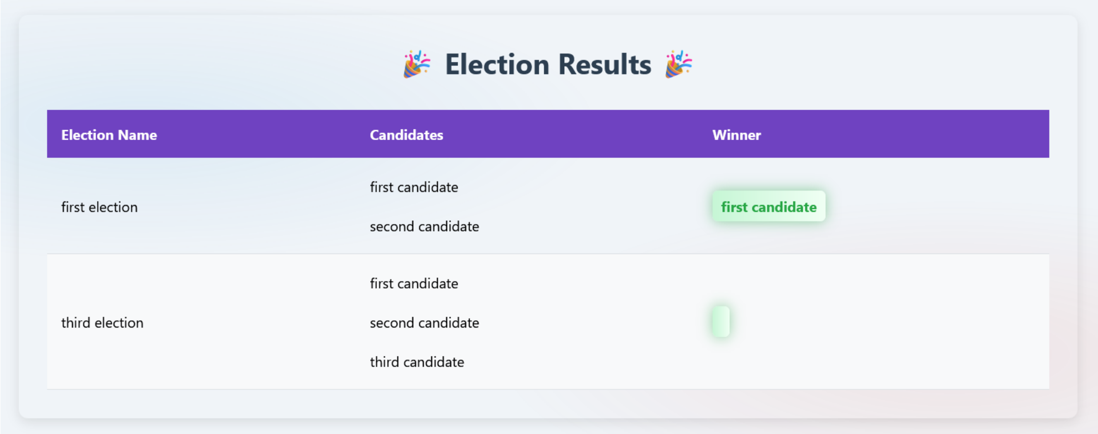

Fig. 7.12: Results Page

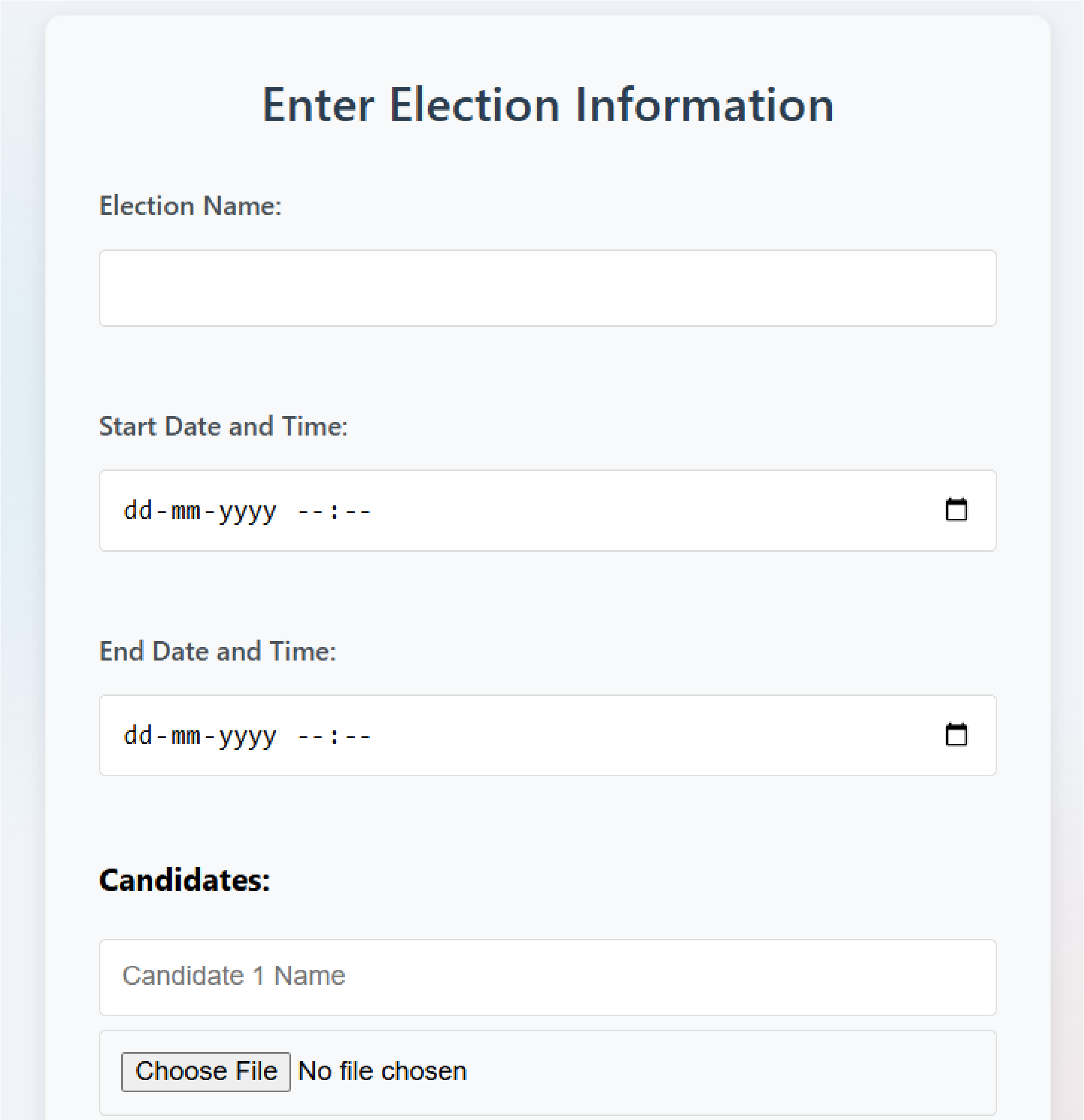

Fig. 7.13: Create Election

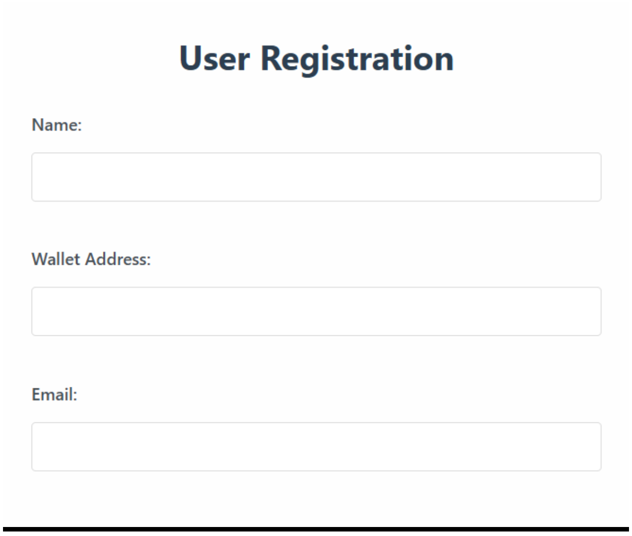

Fig. 7.14: User Registration- part 1

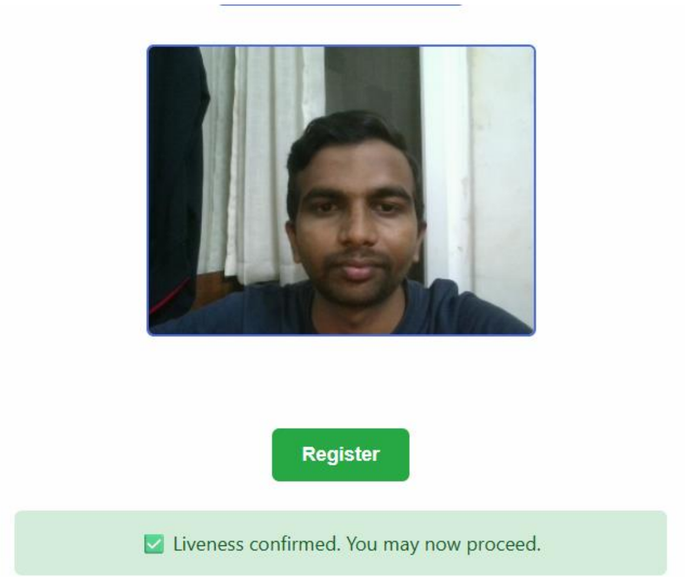

Fig. 7.15: User Registration – part 2

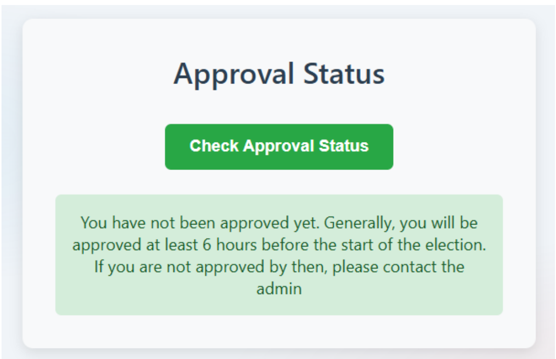

Fig. 7.16: Check Approval Status

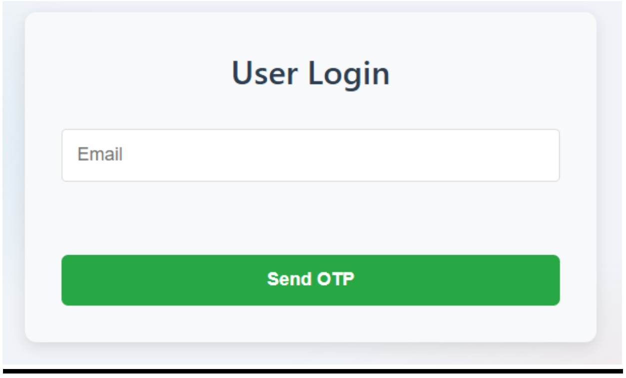

Fig. 7.17: User Login

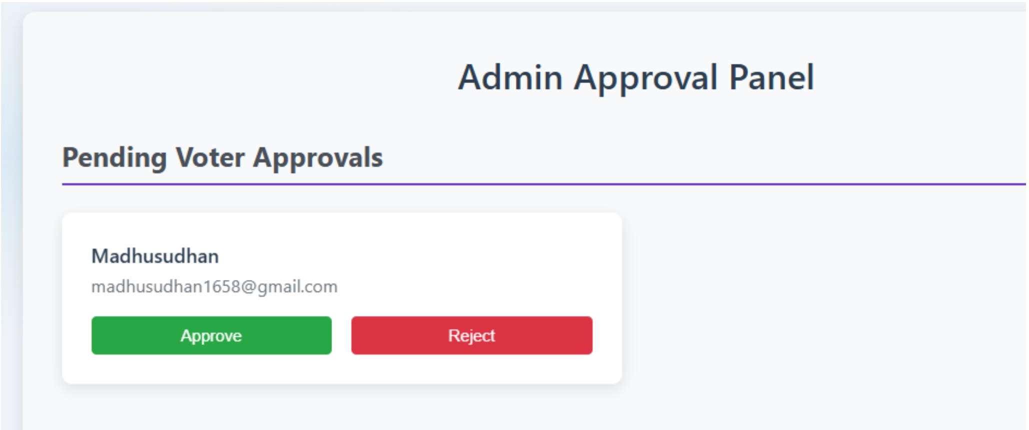

Fig. 7.18: Admin Approval Panel

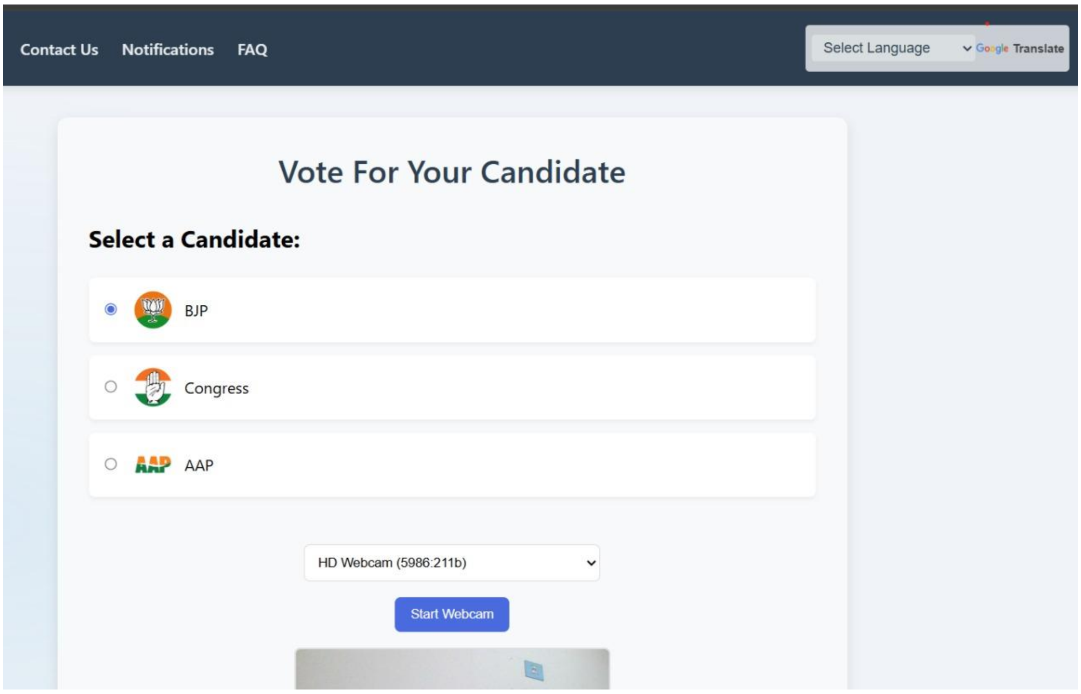

Fig. 7.19: Voting Page

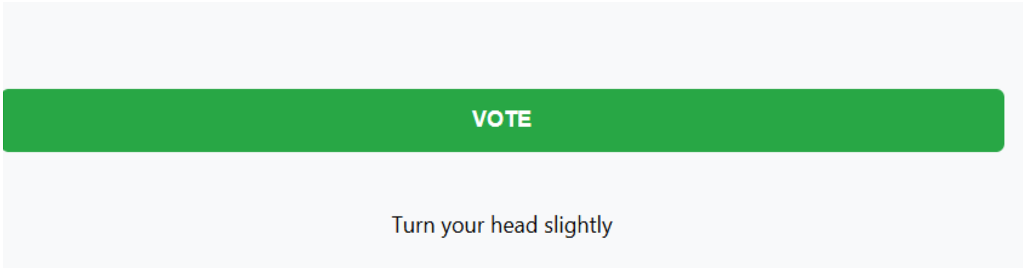

Fig. 7.20: Message during liveliness check

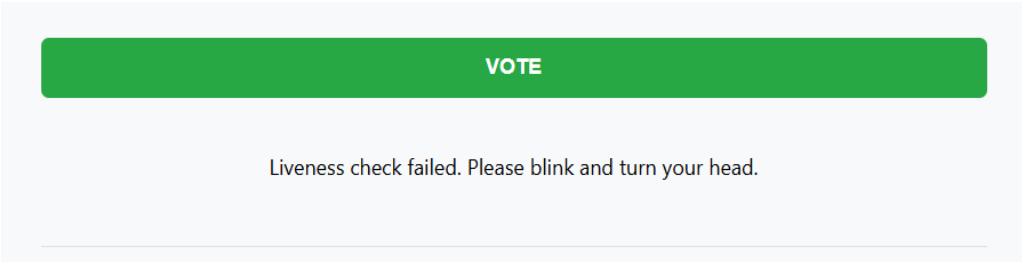

Fig. 7.21: Asking to blink and turn your head for liveliness verification

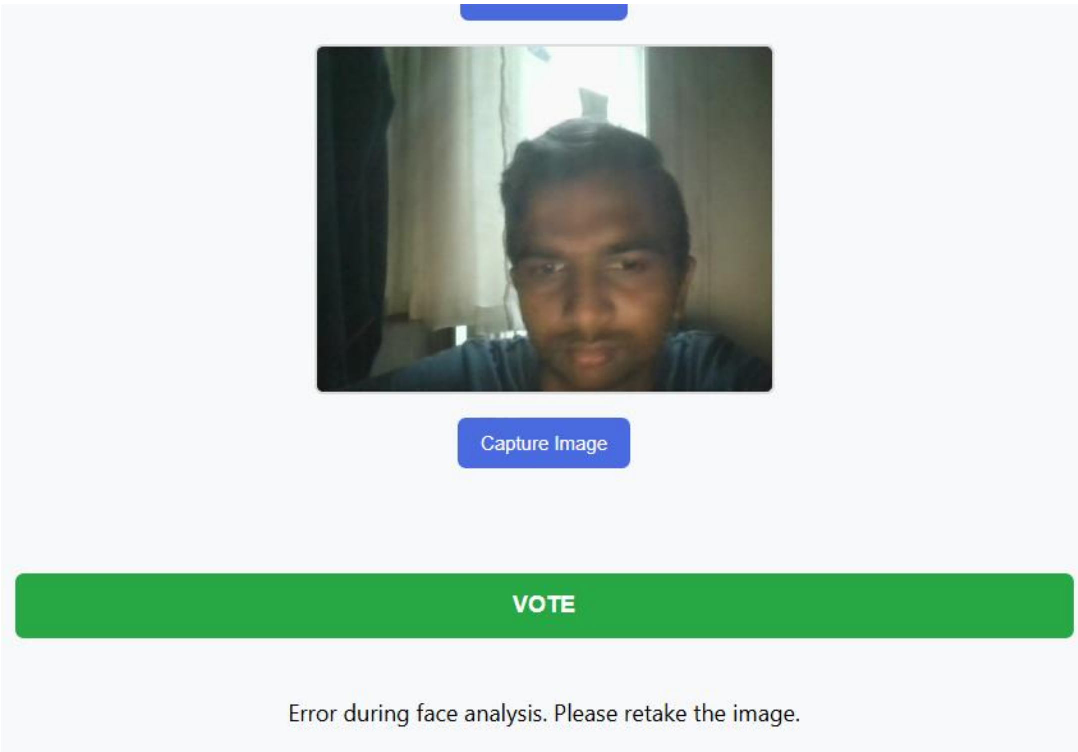

Fig. 7.22: Retake image as image is not clear

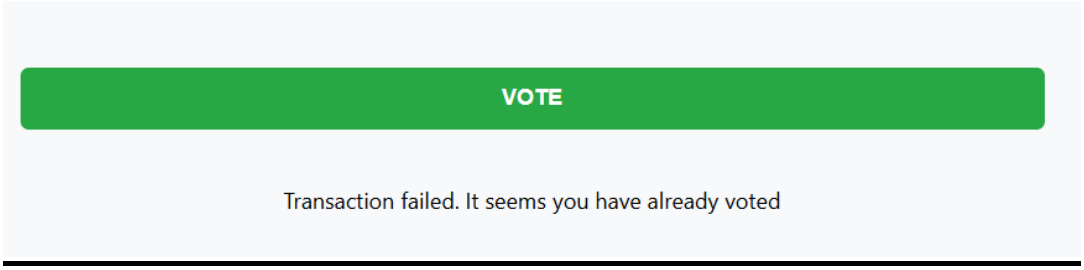

Fig. 7.23: no vote more than once

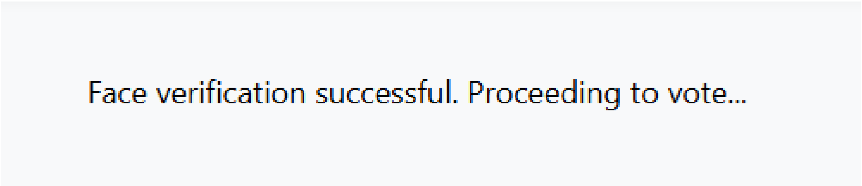

Fig. 7.24: After liveliness check and face verification

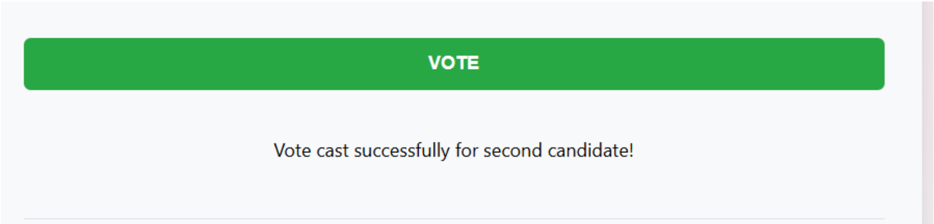

Fig. 7.25: Vote casted successfully

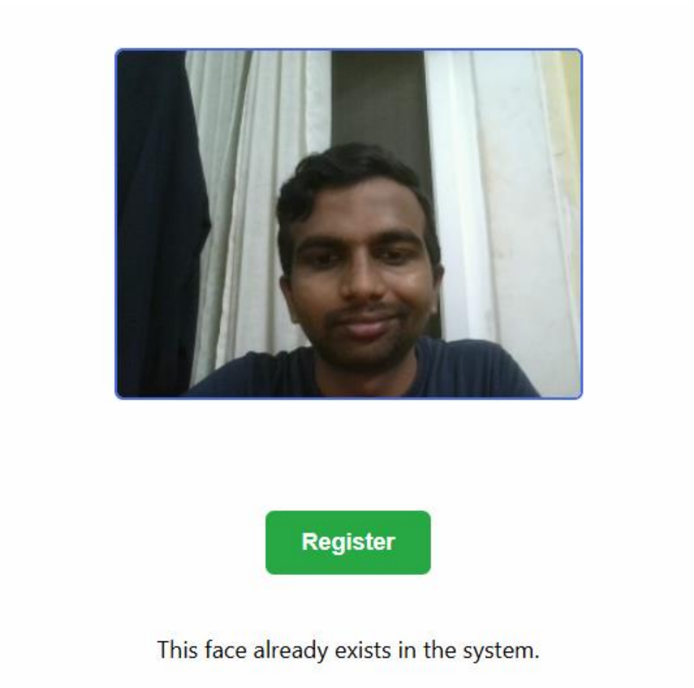

Fig. 7.26: No Registration with same face

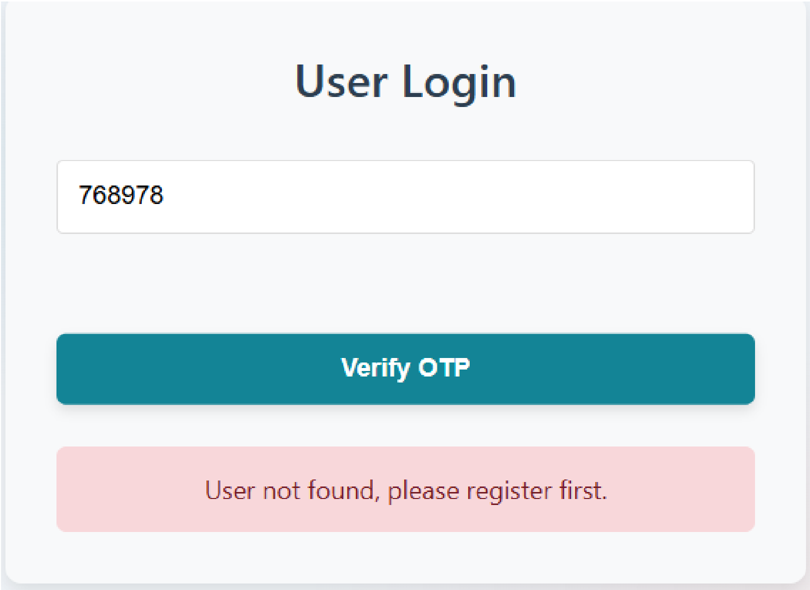

Fig. 7.27: Login not possible without Registration

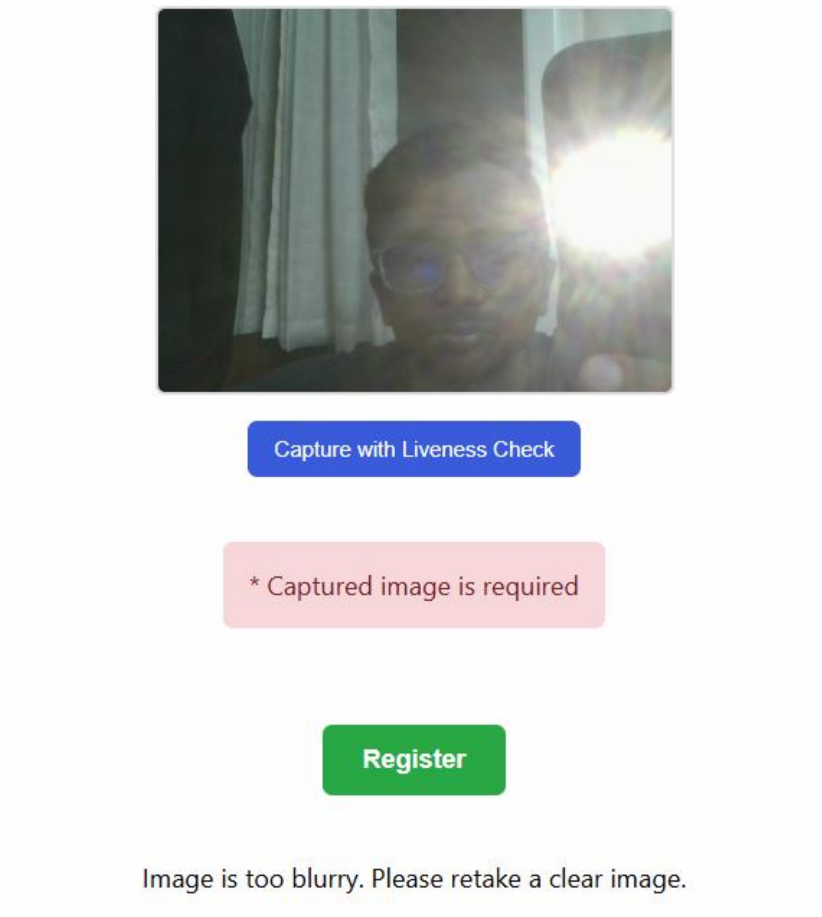

Fig. 7.28: Clear Image is required

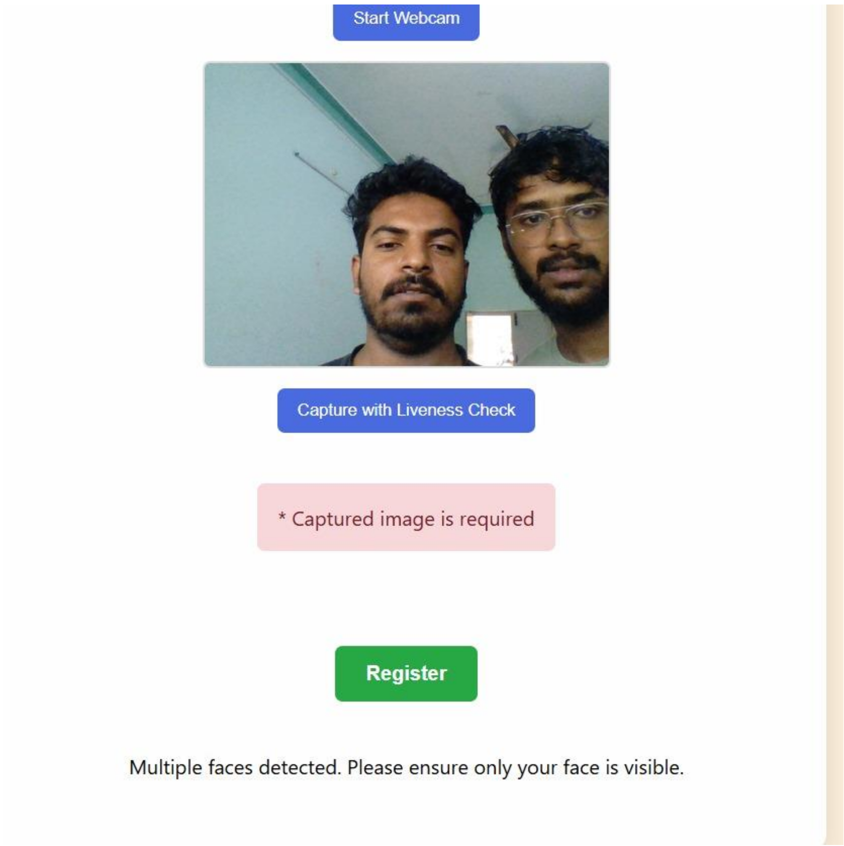

Fig. 7.29: Multiple Faces detected

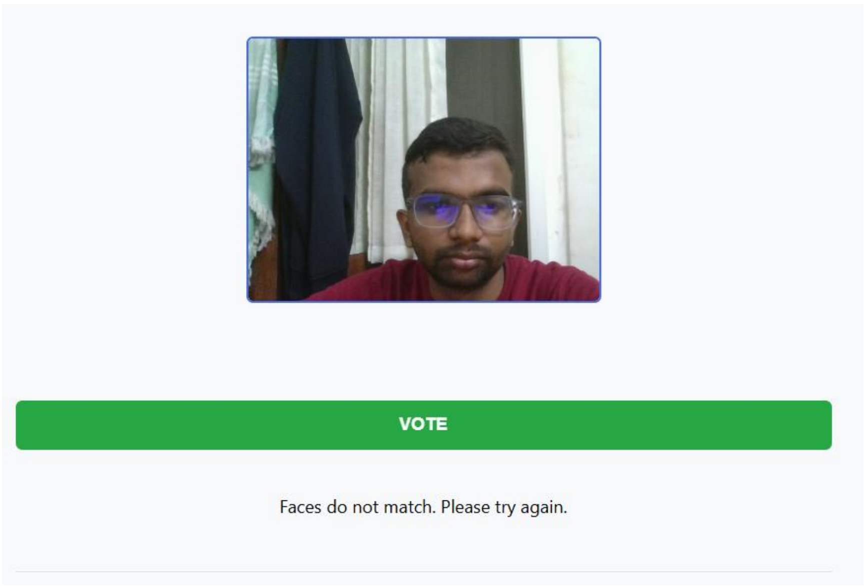

Fig. 7.30: Faces did not match

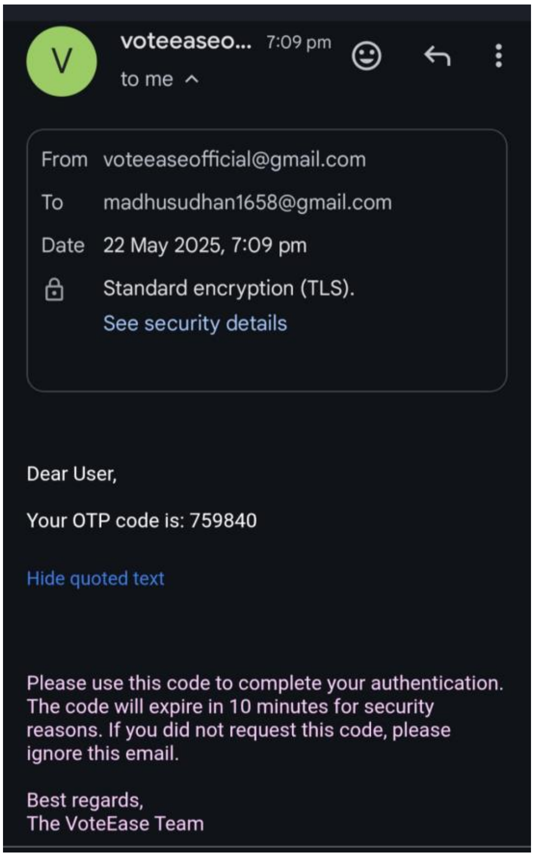

Fig. 7.31: Email for OTP

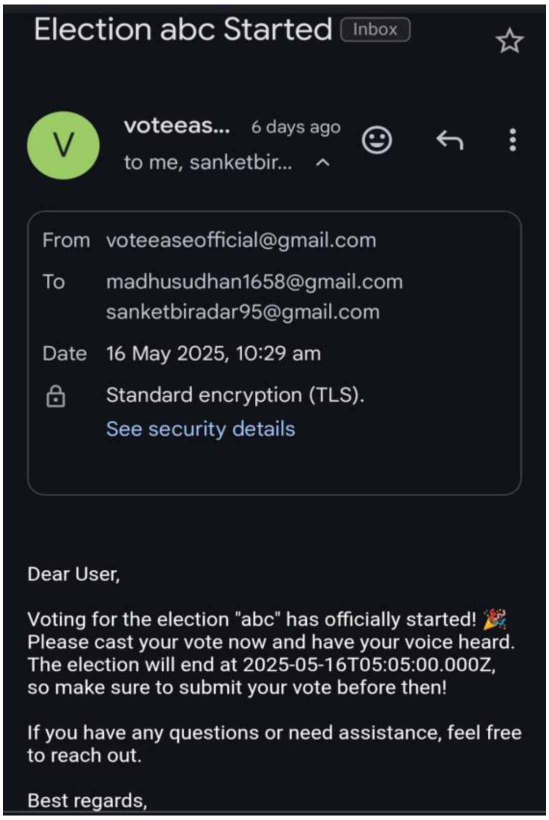

Fig. 7.32: Email for Election Started

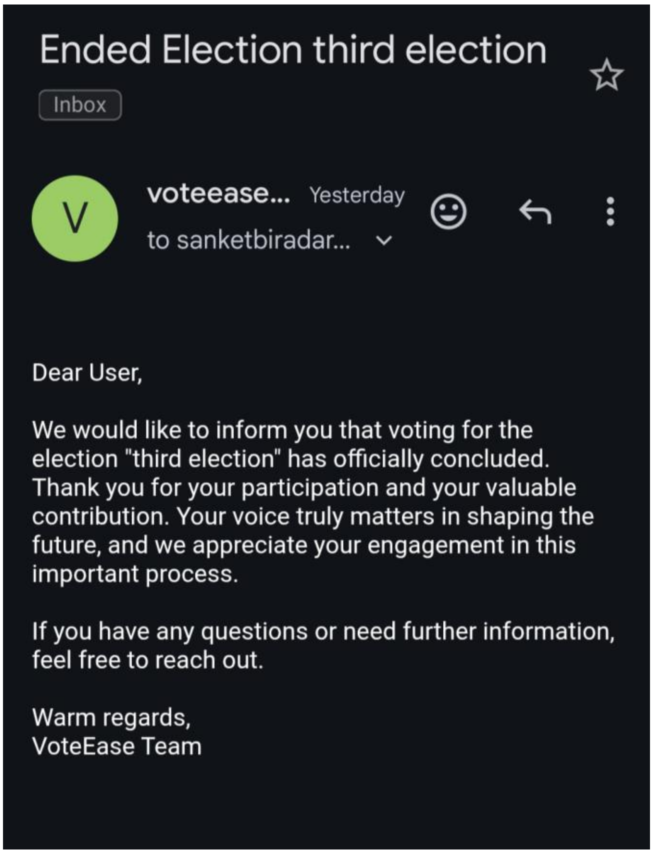

Fig. 7.33: Email for Election Ended

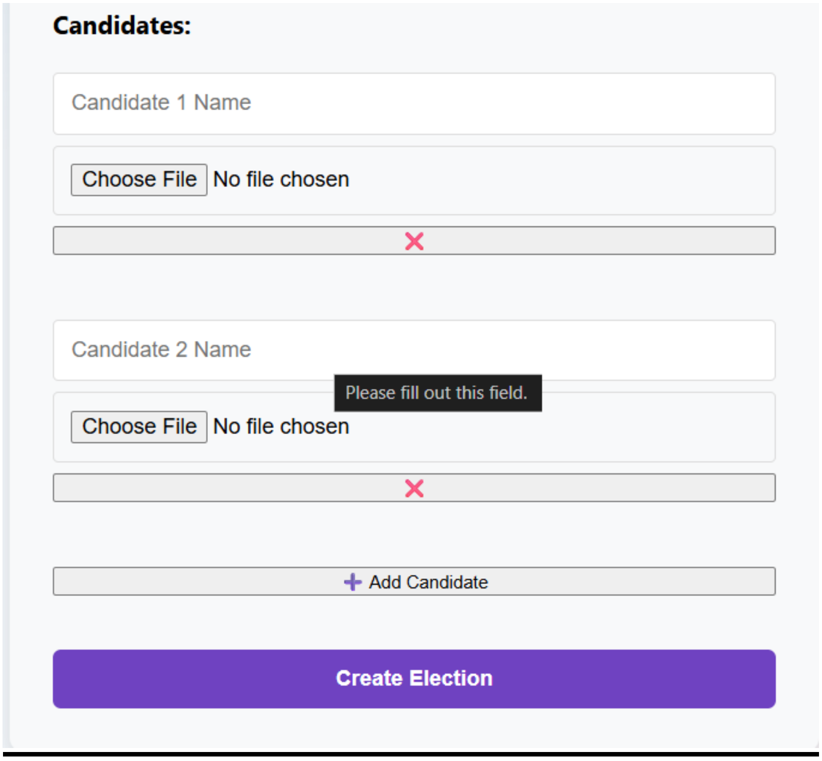

Fig. 7.34: Entering candidates during election creation

CHAPTER 8. CONCLUSION AND FUTURE WORK

8.1 CONCLUSION

In a world that's quick to go digital—from banking, to education, to healthcare—only makes it natural that the voting process goes digital as well. With this project, it wasn't so much our objective to create just another app, but to innovate what a 21st-century secure and democratic election system should be.

By blending the transparency and security of blockchain with the biometric precision of facial recognition, we've built a system where every vote is not just recorded, but respected and protected. Every voter is verified, every transaction is traceable, and the entire process is wrapped in a clean, user-friendly interface that anyone can navigate—even without tech skills.

We used the Avalanche blockchain for a purpose—it allowed us to avoid slow speeds and exploding gas fees we've experienced in other networks such as Ethereum. And with DeepFace facial recognition, we introduced a second layer of trust into the system to ensure every vote is cast by the correct person. Coupled with our React-powered frontend, the result is a site that is fast, secure, and even fun to use.

This is evidence that secure online voting is possible, feasible, and ready to scale. It's not just lines of code—it's a vision for what democracy in the future might look like.

8.2 FUTURE WORK

The ride does not stop here. As we keep adding to this system, a few interesting aspects are in the pipeline. We intend to add screen-reader support to allow differently-abled users to utilize it. Moreover, AI-driven fraud detection would detect suspicious voting patterns in real-time, enhancing system integrity further. We are also working towards adding Zero-Knowledge Proofs (ZKPs) to anonymize voters while keeping votes traceable. The presence of compatibility with digital ID infrastructure and offline voting via QR-code-based verification can further increase the usability. Lastly, the running of government-sponsored pilot schemes and third-party security audits will allow us to transition from prototype to nationwide deployment.

APPENDIX A - PROJECT TEAM DETAILS

Project Title	"VOTE-EASE: AN ONLINE VOTING APPLICATION USING BLOCKCHAIN AND FACIAL RECOGNITION"		
USN	**Team Members**	**Email**	**Mobile number**
01JCE21CS116	Taneeshka Naganath Reddy	taneeshkareddy@gmail.com	9359275252
01JCE21CS060	Madhusudhan	madhusudhan7931@gmail.com	8328151930
01JST22UCS410	Harsha N C	ncharsha7@gmail.com	7019135413
01JST22UCS430	Sanket	sanketbiradar95@gmail.com	9739105905

Taneeshka Naganath Reddy

Madhusudhan

Harsha N C

Sanket

APPENDIX B - COs, POs and PSOs

Mapping for the project work (20CS83P)

Course Outcomes:

CO1: Formulate the problem definition, conduct literature review and apply requirements analysis.

CO2: Develop and implement algorithms for solving the problem formulated.

CO3: Comprehend, present and defend the results of exhaustive testing and explain the major findings.

Program Outcomes:

PO1: Apply knowledge of computing, mathematics, science, and foundational engineering concepts to solve the computer engineering problems.

PO2: Identify, formulate and analyze complex engineering problems.

PO3: Plan, implement and evaluate a computer-based system to meet desired societal needs such as economic, environmental, political, healthcare and safety within realistic constraints.

PO4: Incorporate research methods to design and conduct experiments to investigate real-time problems, to analyze, interpret and provide feasible conclusion.

PO5: Propose innovative ideas and solutions using modern tools.

PO6: Apply computing knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to professional engineering practice.

PO7: Analyze the local and global impact of computing on individuals and organizations for sustainable development.

PO8: Adopt ethical principles and uphold the responsibilities and norms of computer engineering practice.

PO9: Work effectively as an individual and as a member or leader in diverse teams and in multidisciplinary domains.

PO10: Effectively communicate and comprehend.

PO11: Demonstrate and apply engineering knowledge and management principles to manage projects in multidisciplinary environments.

PO12: Recognize contemporary issues and adapt to technological changes for lifelong learning.

Program Specific Outcomes

PSO1: Problem Solving Skills: Ability to apply standard practices and mathematical methodologies to solve computational tasks, model real world problems in the areas of database systems, system software, web technologies and Networking solutions with an appropriate knowledge of Data structures and Algorithms.

PSO2: Knowledge of Computer Systems: An understanding of the structure and working of the computer systems with performance study of various computing architectures.

PSO3: Successful Career and Entrepreneurship: The ability to get acquaintance with the state-of-the-art software technologies leading to entrepreneurship and higher studies.

PSO4: Computing and Research Ability: Ability to use knowledge in various domains to identify research gaps and to provide solution to new ideas leading to innovations.

Justification for CO-PO and PSO mapping

The Online Voting Application aligns seamlessly with the Program Outcomes (POs), Course Outcomes (COs), and Program Specific Outcomes (PSOs) by addressing a critical societal need for secure and efficient voting. The project applies computing knowledge (PO1) and modern tools (PO5) to design a scalable system, analyzes security and privacy challenges (PO2), and ensures ethical practices and sustainability (PO6, PO7, PO8). It incorporates research (PO4) and teamwork (PO9) to implement algorithms for secure authentication and vote encryption (CO2), while effectively presenting the results after rigorous testing (CO3). The project utilizes database systems, web technologies, and networking solutions (PSO1) to create a robust system, demonstrating understanding of computer systems (PSO2) and showcasing potential for entrepreneurship and innovation (PSO3). Additionally, it bridges research gaps and fosters lifelong learning by exploring state-of-the-art technologies for future enhancements (PO12, PSO4).

Table of Mapping of CO, PO and PSO:

SUBJECT	**CODE**	**CO**	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
Project Work	20CS83P	CO1	1	3	2	2	3	3	1	2	3	3	1	2	2	2	1	2
		CO2	3	3	2	2	3	2	1	3	3	3	2	2	3	3	1	2
		CO3	3	3	2	3	3	2	2	3	3	3	2	2	3	2	1	2

Note:

Scale

0. –.Not Applicable

1 – Low relevance Scale

2 – Medium relevance Scale

3 – High relevance Scale

APPENDIX C - PUBLICATION DETAILS

Paper Title: VoteEase: An Online Voting Application Using Blockchain and Facial Recognition

Journal: International Journal for Research in Applied Science and Engineering Technology (IJRASET)

Volume: 13

Issue: V

Publication Date: May 2025

Paper ID: IJRASET70086

DOI/Link: https://www.doi.org/10.22214/ijraset.2025.70086

Status: Published

Figure. IJRASET Journal

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

VoteEase: An Online Voting Application Using Blockchain and Facial Recognition

Taneeshka Naganath Reddy[1], Madhusudhan[2], Harsha NC[3], Sanket[4], BindiyaAR[5]
CSE JSSSTU Mysuru,India

Abstract*: Election system trust must change as society is reshapedbydigitaltransformation.Ouronlinevotingplat- form offers safe, transparent, and decentralised elections by fusing DeepFace-powered facial recognition with the Avalanche blockchain. We guarantee smooth authentication while preserv- ing voter privacy by incorporating real-time facial verification. Everystage,fromvoterregistrationtovotecounting,isprotected by biometric and cryptographic measures. This working proto- type 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 indus- tries, 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, con- temporarysolution. Theproject"OnlineVotingSystemUsing BlockchainTechnologyandFacialRecognition"overhaulsthe voting process using the latest technologies, ensuring security and accessibility for everyone.

Attheheartofthesystemisapowerfulcombination: 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, candidatemanagementthroughabackenddatabase,andsecure 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 Meta- Mask, 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 implementationsbyFaour[1](Waves)andShuklaretal.[2](Ethereum)provedthepotentialofblockchainbutwere limited by replay attacks and excessive gas charges.

VoteEase
Future of Voting,Today

Figure. Published Paper

REFERENCES

[1] N. Faour, "Transparent E-voting dApp 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 Koç, Umut Can Çabuk and Gökhan Dalkihç, "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] Pirpattipanad, 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.