IN PARTNERSHIP WITH PLYMOUTH UNIVERSITY

Name: Delwakkada Nemsara
Student Reference Number: 10898858

Module Code: PUSL3190	Module Name: Computing Individual Project
Coursework Title: Blockchain-based Voting System for Organizations	
Deadline Date: 5th May 2025	Member of staff responsible for coursework: Mr. Gayan Perera
Programme: BSc (Hons) Computer Science	

Please note that University Academic Regulations are available under Rules and Regulations on the University website www.plymouth.ac.uk/studenthandbook.

Individual assignment: I confirm that I have read and understood the Plymouth University regulations relating to Assessment Offences and that I am aware of the possible penalties for any breach of these regulations. I confirm that this is my own independent work.

Signed:

Sinel Nemsara

Use of translation software: failure to declare that translation software or a similar writing aid has been used will be treated as an assessment offence.

I *have used/not used translation software.

If used, please state name of software……

Overall mark _____% Assessors Initials _____ Date_____

UNIVERSITY OF PLYMOUTH

PUSL3190 Computing Individual Project

Final Report

Blockchain-Based Secured Voting System for Organizations

Supervisor: Mr. Gayan Perera

Name: Delwakkada Nemsara
Plymouth Index Number: 10898858
Degree Program: BSc (Hons) Computer Science

Acknowledgement

I would like to express my gratitude to my project supervisor, **Mr. Gayan Perera**, for the support, encouragement, and constructive feedback he provided throughout the course of this project. His technical insights and academic support were instrumental in shaping the direction and quality of this work.

I am also thankful to the academic staff of the School of Computing, whose teaching and mentorship over the years laid the foundation upon which this project was built. Their contributions were significant in helping me acquire the skills required to conduct this study.

In addition, I would like to extend my gratitude to the participants of the requirement-gathering phase, particularly the survey respondents, as well as my colleagues, who provided feedback and helped with the relevance of the solution.

I would also like to acknowledge the support, patience, and encouragement I have received from my family and friends. Your belief in me has helped me stay focused and motivated during challenging times, and I sincerely appreciate all of you for being there throughout this journey.

Finally, I would like to thank everyone who supported me both directly and indirectly for the successful completion of this project.

Abstract

The integrity, transparency, and level of participation associated with elections are global concerns, especially in organisational and institutional settings where people do not trust manual or digital processes. This concern is mitigated in this project by designing and implementing a blockchain-based e-voting platform that uses decentralised technologies to ensure secure, tamper-resistant, and auditable electoral processes.

The primary goal of this study was to allow verified users to participate in digital elections, guaranteeing the uniqueness of votes, transparency of counting, resistance to vote tampering, and manipulation post-election. The approach used was the design and implementation of a voting framework using smart contracts on the Ethereum Sepolia Testnet and supplementary back-end logic with restricted access for better usability and control.

The key capabilities of the implemented system include role-based access control, voter verification via a one-time password (OTP), secure vote casting through wallet-based interaction, and automatic result computation. The testing strategy included API testing for the backend endpoints, component testing of the core front-end, and contract functionalities.

The findings confirm that the system properly enforces voting restrictions, issues event-driven updates for front-end monitoring and maintains transaction logs on the chain in a verifiable manner. Although the addition of sophisticated features, such as cryptographic anonymity layers, is planned for future iterations, the platform provides a functional proof-of-concept that meets the objectives.

This study demonstrates the potential of blockchain technology to enhance the security and trustworthiness of electoral processes in digital governance systems.

Table of Contents

List of Figures

List of Tables

1. Introduction

In an era when technology is transforming every part of an organisation, there is a greater need for voting systems that are secure, transparent, and efficient. Voting is an essential part of decision-making in any organisation, be it a company board meeting or an academic council. However, there are numerous issues regarding the effectiveness of traditional voting systems. Although they are considered fairly trustworthy, paper ballots are associated with operational inefficiencies, the risk of tampering, and a lack of verification in real time. Centralised electronic voting systems, though modern, also have their own drawbacks. They contribute to tightly coupled and poorly modularized system architectures, which introduce single points of failure, resulting in massive vulnerabilities in the data and critical system components.

Blockchain is one of the most transformative technologies that can potentially address these issues. A blockchain is a non-centralised, immutable ledger capable of enabling trustless consensus, transparent audit trails, and permanent record tampering. Once data is recorded on the blockchain, it cannot be altered. This makes blockchain a compelling candidate for electronic voting systems, particularly for organisational settings that require a lighter regulatory framework than national elections but require stringent trust and accountability.

This project showcases the design and implementation of Ballotix, an organisational voting system based on blockchain technology. The Ethereum blockchain is used alongside MetaMask wallet authentication, allowing for seamless login and confirmation that votes are recorded as transactions in immutable smart contracts. Testing in Sepolia does not incur any costs and enables realistic testing without mainnet gas fees or the risk of operational damage, mitigating exposure to net loss servicing costs, and overall system fragility. The architecture of the system contains a front-end built in React.js, a backend with Node.js + Express.js, and auxiliary data handled in a MongoDB database, where supporting services such as SendGrid /Mailtrap for email confirmation and Cloudinary for image storage contribute to a reliable production ready technology stack.

Ballotix aims to address several critical issues in electronic voting systems, including inadequate protection against tampering, weak encryption, voter inattention, and operational inefficiencies. By eliminating reliance on a central authority, trust, or a single institution, blockchain technology allows real-time auditable and cryptographically verifiable votes. Along with employing strong encryption, Ballotix incorporates design principles that ensure privacy, guaranteeing that every vote cast retains its integrity and confidentiality.

The project's motivation emerged from the convergence of distributed ledger technologies with the socio-political challenges of electoral distrust and administrative inefficiency. Preliminary research conducted during the development phase, supported by constituent surveys and stakeholder interviews, highlighted the requirement for intuitive interaction models and user-friendly voting interfaces. These conclusions assist in defining the project scope, confirming that the developed solution meets usability expectations, along with robust technical standards.

There have been attempts to build blockchain-based voting systems at various levels of academia and even to conduct pilot studies; however, most have yet to resolve scalability,

usability, and compliance with legal frameworks. Ballotix addresses these gaps using a modular, scalable architecture combined with a user-centred design. The platform provides role-based access control to election participants, such as administrators and voters, placing them within a secure onboarding framework that utilises email and Ethereum wallet verification. Election smart contracts that record votes and enforce predefined rules defined during the setup, written in Solidity, ensure that elections are conducted without inconsistency or bias once they commence.

The project adopted an agile development approach that allows for iterative feature releases, offers enhanced user involvement, and enables faster feedback cycles. Development practices also include containerisation with Docker and the implementation of CI/CD pipelines, which facilitates the streamlining of testing and deployment processes. These combined efforts resulted in a scalable deployable blockchain-based voting platform capable of supporting elections for multiple types of organisations.

This report documents the complete lifecycle of the Ballotix project, which begins with foundational research and requirement analysis, progressing towards design, implementation, rigorous testing, and reflection on the overall process. The following chapters outline the project goals, literature review, technical approach, system architecture, and detailed outcomes. Emphasis is placed on result evaluation, the unique challenges encountered, and the insights gained during the development phase.

Ultimately, the primary goal of Ballotix is to offer a practical solution for digital governance, while offering guidance on responsibly leveraging blockchain technology to enhance governance. Its inherent scalability, along with increased transparency and resilience, positions the system as a candidate for modernising elections within an organisation while simultaneously reinforcing trust in democratic processes through technological advancements.

2. Background, Objectives and Deliverables

2.1 Background

Voting is one of the most fundamental and universally applied mechanisms for decision-making across various domains, including political systems, corporate governance, academic institutions, and community-based organisations. However, as crucial as it is, the process suffers from a number of issues, particularly within the organisational and digital frameworks. Most voting systems rely on some centralised authority or body to manage and authenticate votes, which introduces vulnerabilities concerning information corruption, inflationary counting, reductions in accuracy, and lack of transparency. Even most contemporary web-based voting systems cannot offer complete privacy while ensuring auditability and verifiable user-centric keyed services.

The potential of blockchain technology can address these issues. Blockchain, as a decentralised and unalterable ledger, coupled with its required attributes of trustless record keeping, adds substantial value to matters of secure, auditable, and verifiable voting systems. Its immutable nature, combined with the absence of controlling focal points, enhances the level of trust and confidence in the voting process.

The project aims to develop a blockchain-based electronic voting platform for organisational elections, which employs the voting potential of Ballotix. The system is constructed on the Ethereum Sepolia test network and integrates React.js as a frontend, Node.js/Express as the backend, with auxiliary data stored in MongoDB. Smart contracts developed in Solidity encapsulate vital election functionalities, which facilitate the secure storage of votes and the checking of results. As described in (Pandey, et al., 2019), the decentralised nature of blockchain ensures voter anonymity while providing verifiable records for every activity carried out within the system. Bhasi and Chandrasekaran (2019) are equally credited for arguing its use in centralisation of power, added transparency, and better protection of the votes cast (Bhasi, et al., 2019). These frameworks contributed considerably to the approach of this project in attempting to understand how blockchain can be integrated into organisational governance systems, where the legal structures, rules, and compliances tend to be flexible, but trust and transparency are vital.

This chapter outlines the key milestones and outputs achieved during the design of Ballotix, a system focused on addressing real-world problem challenges identified through literature reviews and stakeholder analysis.

2.2 Objectives

The primary aim of this project is to design and develop a secure, decentralised, and transparent voting system that leverages blockchain technology to modernise electoral processes within organisations. The following key objectives were established to ensure that the system would be both technically robust and user-friendly:

1. **Develop a decentralised voting platform** that eliminates central authority dependence and offers an immutable record of votes.
2. **Design and deploy Ethereum smart contracts** to manage election logic such as candidate registration, vote casting, and result calculation.
3. **Implement wallet-based authentication** using MetaMask to verify voter identity and ensure one-vote-per-user policy.
4. **Build a responsive web interface** that supports both admin and voter functionalities, allowing users to interact seamlessly with the blockchain.
5. **Integrate third-party services** such as SendGrid for email verification and Cloudinary for secure image handling.
6. **Enable real-time vote updates and result visibility**, ensuring both transparency and usability for election administrators and participants.
7. **Prepare thorough documentation and user guidance** for deploying, managing, and interacting with the platform.
8. **Establish a plan for testing**, including unit and integration testing of both the frontend and smart contract logic, to be executed and reported in subsequent stages of development.

These objectives were shaped by real-world requirements and feedback gathered during preliminary research and iterative prototyping. Each objective aligns with the system's broader goals of enhancing trust, efficiency, and accessibility in electronic voting.

2.3 Deliverables

The Ballotix system has been designed to produce a comprehensive set of functional and technical deliverables that demonstrate both the feasibility and utility of blockchain-based organisational voting. These deliverables are grouped into system features, technical artifacts, and supporting documentation.

Core System Deliverables

- **Blockchain-based Voting Platform**: A complete web application integrating React frontend and Node.js backend with smart contracts on Ethereum's Sepolia testnet.
- **Smart Contracts**: Solidity-based contracts (ElectionFactory and Election) managing election creation, candidate registration, voting, and result computation. *(See Section 4.7 for explanation and code snippets).*
- **Admin and Voter Interfaces**: Role-based dashboards allowing election administrators to create and manage elections, and voters to authenticate and cast votes securely.

- **MetaMask Integration**: Ethereum wallet login enabling secure, cryptographically signed voting.
- **Cloudinary Integration**: For handling candidate profile images and other secure media storage.

Technical Documentation

- **Smart Contract Documentation**: A complete breakdown of contract structures including core functions such as (startElection, voteCandidate, endElection, etc.), and security mechanisms like admin only access controls and unique voter enforcement.
- **System Architecture**: A high-level three-tier architecture diagram detailing the interaction between the frontend, backend, and blockchain layers is provided in Chapter 6.
  (Diagram attached in Section 6.1 - Architecture Design)
- **API and Routing Docs**: Backend API routes and services are modularised and documented for maintainability.
- **User Guide**: A walkthrough document covering platform usage, wallet setup, and voting flow will be attached in the Appendix.

Deployment and Repository

- **Source Code**: Version-controlled GitHub repository covering all frontend, backend, and smart contract components.
- **Deployment Setup**: Scripts and instructions for deploying smart contracts to the Ethereum Sepolia network and the web client/backend to cloud platforms such as Render or Netlify.

Together, these deliverables establish a robust foundation for further enhancements, real-world deployment, or academic reference. They demonstrated the feasibility of a blockchain-based voting system that prioritises security, transparency, and usability for organisational decision-making.

3. Literature Review

3.1 Traditional Electronic Voting Systems: Strengths and Limitations

The implementation of electric voting systems sought to improve traditional methods of voting through the use of technology while simultaneously cutting costs and increasing productivity. Voting equipment has evolved from simple Direct Recording Electronic (DRE) machines to systems resembling Optical Scans, all of which provide significant advancements over paper ballots (Lebre et al. 2004, cited in CORE 2022). Furthermore, (Wadowski, et al., 2023) suggested that electronic systems are widely implemented to mitigate wait times for polling, reduce lifetime operational expenses, and enhance the experience of voters with disabilities.

Despite its numerous advantages, such as effortless vote tallying, increased accessibility, and streamlined procedures, electronic voting systems are plagued by considerable challenges. EODS (2021, p.9) chronicled some of the most pressing challenges as 'lack of transparency, limited openness and understanding of the system for non-experts, lack of agreed standards, and potential violation of the secrecy of the vote.' Further analysis revealed that a lack of decentralisation gave rise to cyber vulnerabilities and data tampering (Bhasi, et al., 2019)

McGaley and Gibson (2003, cited in EODS, 2021) put it appositely that "a voting system is only as good as the public believes it to be", stressing that perceived security gaps undermine public trust in an election's outcome even when the system is intact. Centralised electronic voting systems were shown to be vulnerable when MIT researchers revealed critical security weaknesses in the Voatz smartphone voting app, stating that exploitation "would be well within the capacity of a nation-state actor" (Collier & NBC News, 2020).

3.2 Blockchain Technology and Its Relevance to Secure Voting

The shortcomings of traditional electronic voting systems can be addressed through the decentralisation, immutability, and transparency features of blockchain technology. Blockchain is a distributed ledger with a list of records, known as blocks, that is ever growing and secured through cryptography, with previous blocks being hashed (Nakamoto, 2008).

Voting is one of the areas in which blockchain technology is applied because it enables the creation of records that cannot be altered and can be verified transparently. As noted by (Cole, 2024), "blockchain's decentralised architecture… distributes voting data across a network of nodes, making it virtually impossible for any single entity to tamper with the results." Tampering is no longer possible with records stored in other nodes, because electronic systems rely on a central point to function.

Once a vote is cast, the vote being immutable guarantees that it is part of an immutable ledger. Sharp et al. (2024) emphasise that this characteristic is essential for a voting system as it ensures that votes remain unchangeable after being cast. Moreover, audit trails can be created without compromising voter identity because of blockchain's transparent yet cryptographic nature.

3.3 Prior Research on Blockchain-Based Voting Systems

Interest in research on blockchain-based voting systems has grown and evolved since 2017, with Sharp et al. (2024) examining the architecture, cryptographic techniques, vote-counting processes, and security requirements of different systems. Their research introduced a novel approach that incorporates the Borda count and Condorcet methods to enhance accuracy when tallying votes.

Owing to the smart-contract functionality of the Ethereum platform, its use has drawn attention. Hjalmarsson et al. (2018) showed how Ethereum smart contracts automate and secure the voting process, allowing votes to be recorded as transactions on the blockchain. Pereira et al. (2023) detailed further capabilities of these contracts, illustrating how they can autonomously enforce election rules, validate individual votes, and prevent multiple votes from being cast commonly referred to as 'double voting' without trusted third parties.

A systematic review conducted by researchers in 2024 assessed different frameworks of blockchain technology for e-voting and found that Ethereum is capable of secure and portable elections. The review underscored the role of smart contracts in enabling "anonymity, privacy, verifiability, and fairness in the process of voting" (Ohize et al., 2024). Abdul and Saleem focused on other blockchain innovations, arguing that the rapid evolution of technology paves the way for a new generation of electronic voting systems that offer improved security, efficiency, and accessibility (Abdul & Saleem, 2024).

3.4 Key Challenges in Blockchain Voting

While blockchain technology presents promising enhancements for electoral systems, it also introduces a distinct set of technical, usability, and economic challenges that must be carefully addressed to ensure its viability in large-scale democratic processes. A fundamental limitation lies in scalability. As Abdul and Saleem (2024) categorise, many blockchain-based voting solutions encounter significant difficulties in managing high transaction throughput, particularly during peak voting periods. Early deployments of blockchain networks such as those built on Ethereum have exhibited bottlenecks under high loads, rendering them less suitable for national or high-turnout organisational elections.

The other major challenge, according to Aidynov and Goranin (2024), is privacy and voter anonymity. Indeed, many actively working systems that preserve the integrity of votes dominantly lack proper measures to support voter anonymity. The problem is how to validate a voter's eligibility in an election, while guaranteeing that the identity of the ballot caster remains confidential.

Explaining blockchain remains unfamiliar to many people. Those operating it range from the general public to very sophisticated tech-savvy individuals, and therefore, the interfaces need to be user-friendly. Clearly, tendered votes must not become too complicated, disrupting intuitive voting interfaces. Educational hurdles among voters and election officials, as introduced by Tambanis (2019), are also part of the unwillingness to endure potential discomfort by adopting newer approach-based voting systems.

In particular, gas expenses present a substantial problem for Ethereum-based implementation. The research conducted by Ohio State University (2023) segregates that “Ethereum gas fees” impose negative consequences on the economic feasibility of remotely executed on-chain voting, especially for elections which scale in size. Moreover, the research talks about possible smart contract exploits such as "reentrancy attacks" that need to be mitigated with proper security auditing (Chai & Ohio State University, 2020).

3.5 Gaps in Existing Solutions

Existing blockchain voting approaches leave several gaps which justify further research. Many current implementations do not seem to make use of all possibilities provided by blockchains, such as the confirmation of identity and verification of attendance during voting in real time (Cole, 2024). This lack of precision for those with voting rights undermines trust among stakeholders, which is essential for the successful implementation of the voting system.

Regulatory vagueness is the central gap. Mohammed Abdul et al. emphasise that many regions have a void of distinct regulatory politics pertaining to voting with blockchain technology; this legal gap highly hinders the application. This lack of clarity requires more innovative approaches built around compliance, using legislation crafted alongside technological advances (Abdul & Saleem, 2024).

Existing solutions also tend to misalign the consideration of decentralisation and performance enhancement, dominating other solutions. Hajian Berenjestanaki highlighted that some systems continue to use centralised elements which introduce weak points that contradict the decentralised essence of blockchain technology (Berenjestanaki et al., 2024).

Inadequate attention is often given to user experience issues. Current systems focus on what can be done technologically rather than how easy it is to use, which hampers adoption by non-technical users (Tambanis, 2019). This emphasises the security concerns that still exist alongside the need for simple interfaces.

3.6 Related Projects and Platforms

There appear to be a number of attempts, some more successful than others, to implement voting systems based on blockchain technology. “Follow My Vote” has created “end to end verifiable online voting software that is open source and truly revolutionary” (Follow My Vote 2024). Their system utilised blockchain technology to render the voting process secure and transparent. Voters were able to verify that their votes were accurately recorded without revealing their identity.

Agora offers “a blockchain based voting ecosystem allowing anyone anywhere to vote online from a digital device in a fully secure, easy and certain way” (Agora 2017). Their system solves the issues associated with online voting through “cryptographic security and a distributed and decentralised network architecture which defends election data from external interference.

Voatz is known to be one of the most broadly implemented blockchain voting applications, with limited usage during elections in the United States. However, researchers at the

Massachusetts Institute of Technology (MIT) identified critical vulnerabilities in the Voatz application that could potentially enable various adversaries to manipulate, halt, or deanonymise a user's vote (Collier & NBC News, 2020). This highlights the need for more scrupulous security testing and open-code auditing in blockchain voting systems.

3.7 Research Direction and Project Approach

An analysis of the literature reveals that although blockchain technology has potential solutions to the constraints of traditional voting systems, challenges still exist in developing secure, usable, and scalable implementations. The Ballotix project addresses these issues by concentrating on organisational elections first, as these can be implemented in a controlled manner devoid of the regulatory hurdles characteristic of governmental elections.

Building on the existing blockchain infrastructure, Ballotix integrates Ethereum's Sepolia testnet, Solidity smart contracts, and MetaMask wallet, in conjunction with innovations in user experience design through the React/Node. js tech stack. This focus on user experience is in sharp contrast to UC Berkeley's observations regarding the lack of attention given to user interfaces and interaction design by applying blockchain technologies (Sharp et al. 2024).

The project's focus on auditability and transparency addresses the concerns captured by Cole (2024) on real-time tracking functionality, whereas the use of cryptographic techniques for anonymising votes addresses privacy issues raised by Aidynov and Goranin (2024). In concentrating on organisational elections, Ballotix avoids the extreme scalability issues other projects have become stuck, while still providing a strong test for the principles of blockchain voting systems.

To summarise, prior studies recognise the strengths of blockchain voting systems while highlighting serious problems that remain unresolved. The Ballotix project is based on the work of prior implementations to develop a secure, transparent, and usable voting system for organisational elections.

4. Method of Approach

4.1 Requirement Gathering and Analysis

The Ballotix vote-casting platform utilising blockchain technology begins with a systematic multilevel requirement analysis to address all possible factors. With this step, the designers set out to address security, decentralisation, auditability, and other user-centric issues that are crucial for an organisational election in a tailored voting platform.

Fact-Finding Techniques

Multiple industry standard techniques were employed to ensure comprehensive requirement elicitation.

- **Literature Review:** The analysis of academic articles alongside voting systems based on blockchains revealed recurring problems such as centralisation, transparency, privacy, stakeholder ability, and bottlenecking. All of these issues became building blocks in conjunction with non-functions to establish gaps within systems.
- **Survey Research:** A Google Forms-based online survey was distributed across a diverse respondent base including corporate, academic, non-profit, and government sectors. The goal was to assess stakeholder expectations and operational pain. Key insights included:
    1. 85% of the participants (voters) emphasised the need for a secure and seamless voting experience.
    2. 7.5% (election administrators) required robust backend tools and real-time oversight features.
    3. 7.5% (IT/Admin staff) prioritised ease of maintenance and system reliability.
    4. 70% of all respondents were from corporate environments, affirming the need for a scalable and user-friendly solution for internal elections.

    (See Appendix 6 for detailed survey results.)

- **Observations:** All existing voting systems, both paper and electronic, were assessed for operational vulnerabilities, including critical failure points, ballot-box tampering, and audit-free real-time monitoring. This qualitative assessment proved the need for blockchain's fundamental attributes, such as immutability, decentralisation, and transparency, in the proposed solution.

Design Implications

The requirement analysis phase culminated in a user-centred system blueprint, emphasising the following:

- **Simplicity and trust**: Prioritised through a minimal, intuitive UI/UX to reduce voter errors and increase adoption.
- **Auditability**: Ensured via smart contracts and integration with blockchain explorers for transparent verification.
- **Security by Design**: Enforced through cryptographic safeguards and rigorous access control.

These insights directly inform architectural decisions, technology stack selection, and smart contract specifications, providing a robust foundation for subsequent development phases.

4.2 Development Methodology

To manage the complexity and evolving nature of this blockchain-based voting platform, the agile software development methodology was adopted. Agile was particularly well suited for this project because of its iterative structure, ability to accommodate change, and focus on delivering functional components early and frequently.

Justifying Agile for a Decentralized Architecture

Given the multiple interdependent layers of the system such as blockchain smart contracts, web frontend, decentralised storage, and backend services, Agile's emphasis on incremental development and continuous feedback provides a clear advantage. This approach ensured that the features could be evaluated in working increments and refined based on the technical feasibility and user feedback.

Sprint Structure

The project was divided into time-boxed sprints lasting 1–2 weeks, each following a structured cycle.

- **Sprint Planning**: Defined sprint goals (for example, MetaMask integration, vote casting logic, real-time result display).
- **Development**: Implemented frontend, backend, and smart contract modules aligned with sprint scope.
- **Testing and Validation**: Conducted unit and integration testing within each sprint to ensure functional reliability.
- **Review and Feedback**: Iterative feedback sessions were conducted to validate features against stakeholder expectations and technical criteria.

This cyclical process allows for the early detection of implementation issues (e.g. blockchain gas limitations and UI performance), enabling timely pivots and fixes without derailing the overall timeline. Sprints typically lasted two weeks, covering goals such as implementing authentication, integrating blockchain functionality, or enhancing UI responsiveness. GitHub Projects and GitHub Actions were used for sprint tracking, version control, and CI/CD pipeline execution.

An Agile development methodology was adopted, which allowed the project to progress through iterative sprints. Each sprint involved planning, implementation, and feedback collection phases to accommodate the changing user expectations and technical challenges. Agile was chosen because of its flexibility and emphasis on continuous delivery, which are key advantages for a blockchain-based system that depends on precise coordination between the frontend, backend, and smart contract layers. GitHub Projects and GitHub Actions were used for sprint tracking, version control, and CI/CD pipeline execution.

(See Appendix 4 for the Gantt chart which visualizes the project milestones within the iterative development methodology)

4.3 Feasibility Study

A comprehensive feasibility analysis was conducted to evaluate the viability of implementing the blockchain-based voting platform across the technical, operational, and financial dimensions. The findings support a strong case of sustainable implementation within organizational environments, particularly in academic and corporate settings.

4.3.1 Operational Feasibility

The operational feasibility of Ballotix was validated through early stage user engagement including surveys, interface mockups, and feedback from usability testing sessions. The system's core functionalities including secure voting, result transparency, and admin control were well received by the targeted user groups, including university students, academic staff, and corporate employees.

The key operational strengths include the following.

- **Role-Based Dashboards**: Tailored user interfaces for voters, administrators, and IT support staff.
- **Device and Platform Compatibility**: Developed using React.js, the frontend supports all major browsers and devices with responsive designs via the Tailwind CSS.
- **User-friendly logic mechanism**: Blockchain-savvy users benefit from MetaMask wallet authentication, while fallback mechanisms such as email verification ensure accessibility for less technical users.
- **Language and Interaction Simplicity**: Clean UI design and minimalistic interaction flows reduce training overhead and enhance adoption, as emphasised in survey feedback (see Appendix 6).

4.3.2 Technical Feasibility

Ballotix was architected using mature, interoperable technologies capable of supporting secure, real-time election operations. All the components were tested in a development environment that simulated real-world deployment scenarios.

Table 1: Technical Feasibility Assessment of System Components

Component	**Technology**	**Status**
Blockchain Layer	Ethereum Sepolia Testnet	Integrated and functional
Smart Contracts	Solidity-based, locally tested & verified	Completed
Frontend	React.js + Tailwind CSS + Web3.js	Fully implemented
Backend	Node.js + Express.js	API endpoints operational
Database	MongoDB Atlas (cloud-based)	Configured and scalable
File Storage	Cloudinary API (for candidate images)	Integrated
Email Services	SendGrid (production), Mailtrap (testing)	Fully functional

Additional security is provided through Ethereum's transaction validation, cryptographic signing, and strict access control of administrative endpoints.

4.3.3 Budget Considerations

Ballotix was deployed using free-tier infrastructure and public test nets to ensure cost-effectiveness during the development and initial testing phases. This decision significantly reduces hosting and operational expenses without compromising functionality.

Table 2: Estimated Costs of Project Infrastructure and Services

Component	**Service/Tool**	**Est. Monthly Cost**
Frontend Hosting	Netlify (Free Tier)	$0
Smart Contract Ops	Ethereum Sepolia Testnet	$0 (faucet-based ETH)
File Storage	Cloudinary (Free Tier)	$0
Email Service	Mailtrap / SendGrid	$0 / Pay-as-you-go
Backend & DB Hosting	Render (Free Tier)	$0

Note: A transition to production-grade infrastructure (for example, AWS for backend hosting and IPFS for decentralised storage) is planned for post-deployment, contingent on scale and organizational adoption.

4.4 System Architecture

The high-level system architecture comprises four primary layers:

- **Frontend Layer**: A React.js SPA that interfaces with MetaMask for wallet-based authentication and displays real-time election data.
- **Backend Layer**: Node.js with Express manages API routes, interacts with MongoDB, and communicates with the Ethereum blockchain via Web3.js.

- **Blockchain Layer**: Smart contracts govern election logic, including voting, candidate registration, and result calculation.
- **Storage Layer**: MongoDB stores non-sensitive metadata, and Cloudinary handles candidate images.

High level system architecture diagram is attached under the 6.1 in System Design section

4.5 Programming Languages and Tools

Table 3: Technologies and Tools Used Across Project Components

Component	**Technology Used**
Blockchain	Solidity (Ethereum), Web3.js, MetaMask
Frontend	React.js + Tailwind CSS, Vite
Backend	Node.js + Express.js, MongoDB
Hosting	Render and Netflify (for backend/frontend), Sepolia testnet
DevOps/Testing	GitHub Actions (CI/CD), Docker, Postman Mocha, Chai, Jest

4.6 Third-Party Libraries and Services

The following third-party services and libraries were integrated into streamlined development and added essential functionalities:

- MetaMask - Ethereum wallet integration for secure vote casting.
- SendGrid - Email verification for voter account setup.
- Cloudinary - Candidate image upload and retrieval.
- Framer Motion & Chakra UI - Frontend animation and component libraries.
- ShadCN UI - Accessible, Tailwind-based component library built on Radix UI for consistent, modern UI design.
- Mocha/Chai - Used for smart contract unit testing.
- Render - Cloud hosting for the backend.
- Netlify - Cloud hosting for the frontend.
- MongoDB Atlas - Cloud-based NoSQL data store for admin and election metadata.

4.7 System Development and Implementation

The Ballotix platform follows a modular, full-stack architecture combining a React-based frontend, an Express.js backend, and Ethereum smart contracts. This section outlines the project structure and the implementation approach across its three main components.

4.7.1 Project Directory Structure

To maintain scalability, clarity, and separation of concerns, the system is divided into three top-level components:

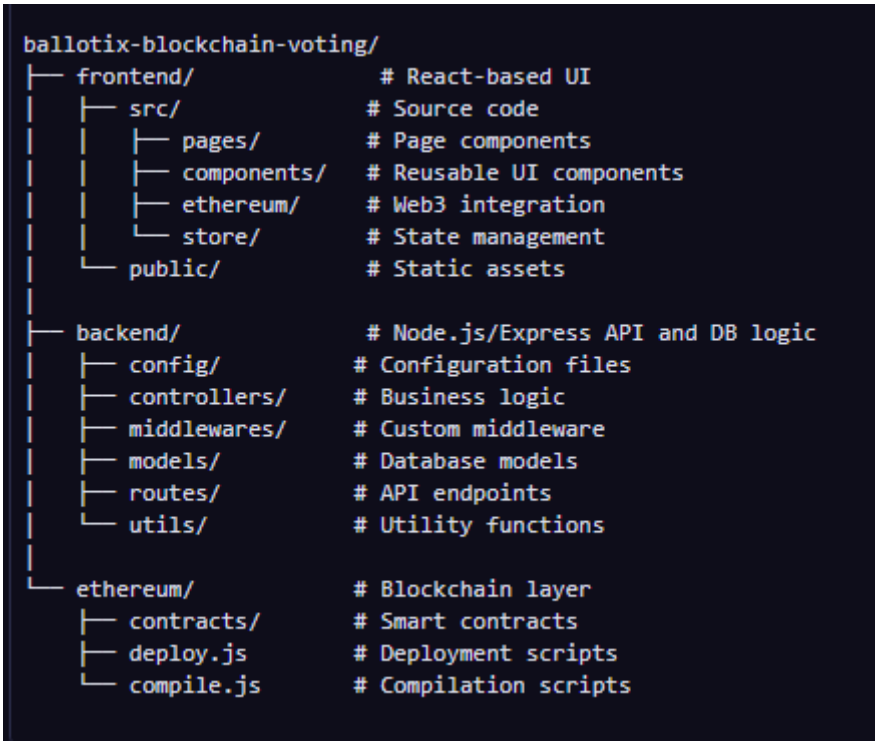


Figure 1: Project Directory Structure of Voting Platform

4.7.2 Smart Contract Implementation

The smart contract `Election.sol` includes functions such as:

- `startElection()`: Initializes a new election.
- `registerCandidate()`: Allows admin to add candidates.
- `voteCandidate()`: Lets verified users cast a vote; checks if already voted.
- `endElection()`: Closes voting and triggers winner determination.

Security considerations like access modifiers (`onlyAdmin`) and vote duplication checks were applied to ensure integrity.

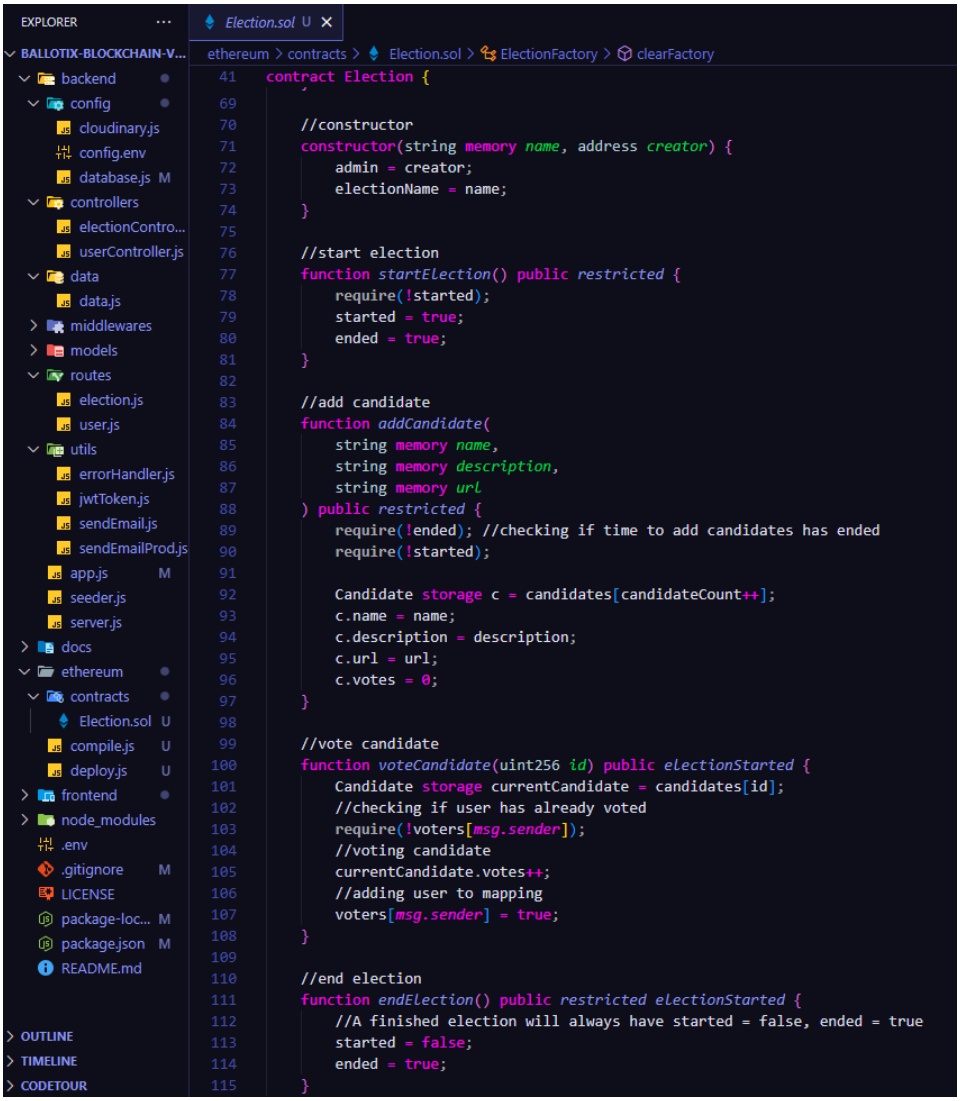


Figure 2: Smart Contract Implementation

4.7.3 Frontend Implementation

The frontend was developed using **React.js** with **Tailwind CSS** for styling and **Web3.js** for blockchain interaction. Major components include:

- Login.jsx: MetaMask wallet login and user authentication
- AddElection/, AddCandidate/: Admin management tools for election setup
- Election/: Voter dashboard with on-chain vote execution
- AllResults/, SingleResult/: Result retrieval and display
- AllUsers/, EditUser/: Admin user control panels

Key Blockchain Interaction:

- frontend/src/ethereum/web3.js: Initializes Web3 connection with MetaMask and Sepolia testnet
- frontend/src/ethereum/factory.js and election.js: Contract instance management
- frontend/src/pages/Election/: Handles voting logic via smart contract functions

```
web3.js U
frontend > src > ethereum > web3.js > ...
1   import Web3 from "web3";
2
3   let web3;
4
5   if (typeof window !== "undefined" && typeof window.ethereum !== "undefined") {
6     // We are in the browser and metamask is running
7     window.ethereum.request({ method: "eth_requestAccounts" });
8     web3 = new Web3(window.ethereum);
9   } else {
10    // We are on the server OR the user is not running metamask
11    const provider = new Web3.providers.HttpProvider(
12      import.meta.env.VITE_RINKEBY_ENDPOINT
13    );
14    web3 = new Web3(provider);
15  }
16
17  export default web3;
18
```

Figure 3: Web3 Initialization Logic for Blockchain Connectivity

Figure 4: Client-Side Development

4.7.4 Backend Implementation

The backend is built on Node.js and Express.js, serving as the bridge between frontend actions, MongoDB, and blockchain operations.

Main API Functionalities:

- **User Routes**: /login, /register, /vote, /edit, /allUsers, /delete/:id
- **Election Routes**: /startElection, /endElection
- **OTP Verification**: /generateOtp via SendGrid/Mailtrap

Technology Stack:

- **MongoDB Atlas**: Cloud database for user and election metadata
- **Cloudinary**: Candidate image storage
- **Smart Contracts**: Used for vote handling and election status

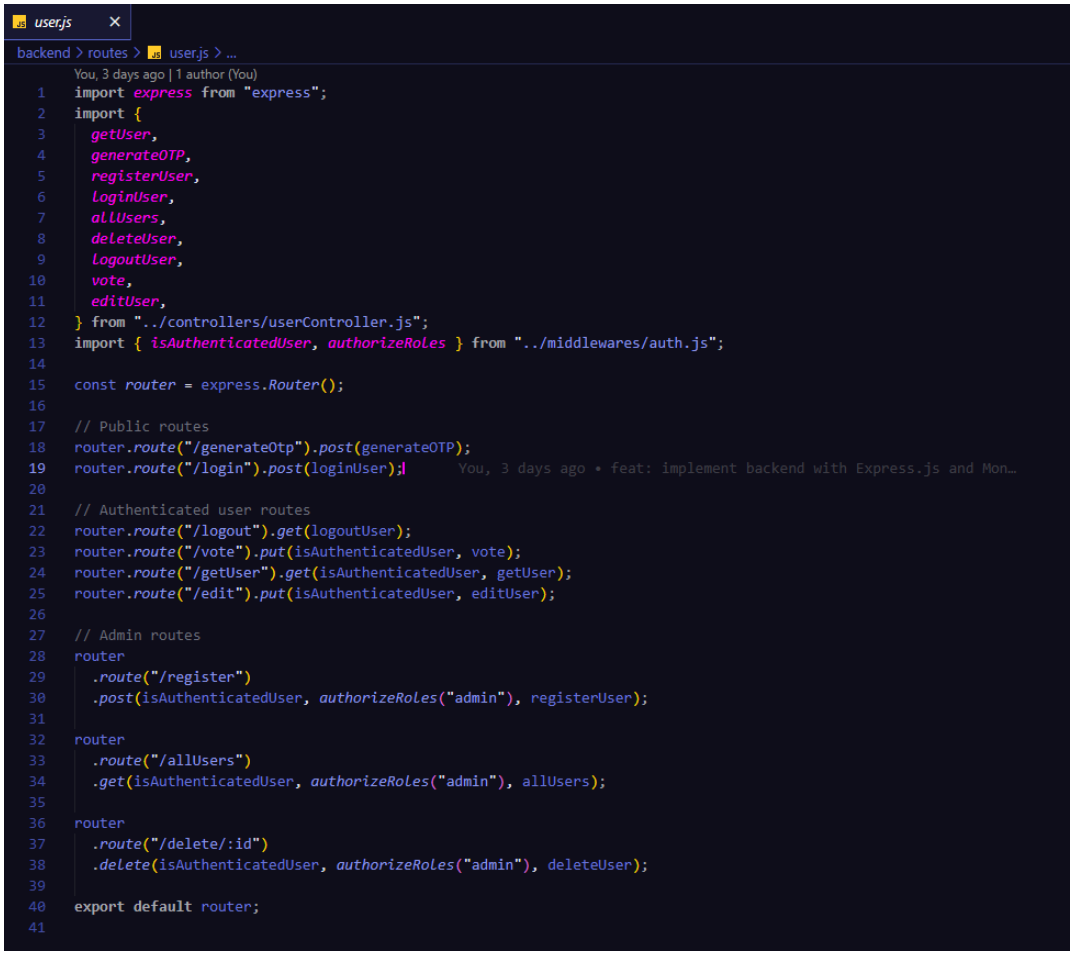


Figure 5: User API Routing and Middleware Integration

Above code snippet demonstrates the Express.js route configuration for user authentication and role-based access control in the system.

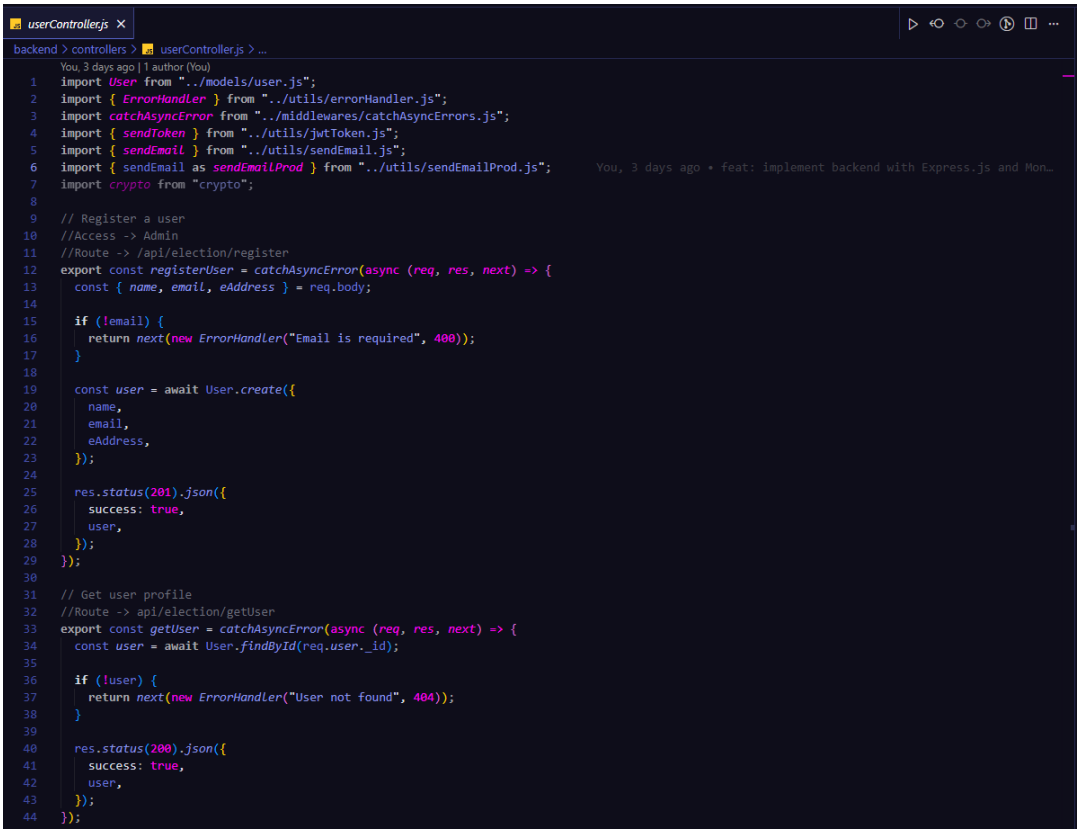


Figure 6: Controller Logic for User Authentication and Voting Operations

Figure 6 showcases the core business logic responsible for user registration, OTP-based verification, logic processing, and voting functionality. This component orchestrates communication between the API endpoints, database, and smart contracts, ensuring secure and validated operations.

4.8 Testing Plan

Ensuring reliability and correctness in a blockchain-integrated voting platform requires testing at multiple levels, including backend API flows, frontend interactions, and smart contract logic. This section outlines the testing strategies, tools used, and test evidence carried out as part of this project.

4.8.1 API Testing

API testing was a central part of the testing strategy due to the system's reliance on RESTful endpoints for user registration, authentication, OTP verification, and vote-related operations. The endpoints were tested using **Postman**, validating expected responses and status codes under different scenarios.

Two core endpoints were mainly tested:

1. **POST /api/election/generateOtp**
    - Accepts an email address.
    - Returns a success response and triggers an OTP email via Mailtrap.
2. **POST /api/election/login**
    - Accepts email and OTP.
    - Returns a JWT token and user metadata.

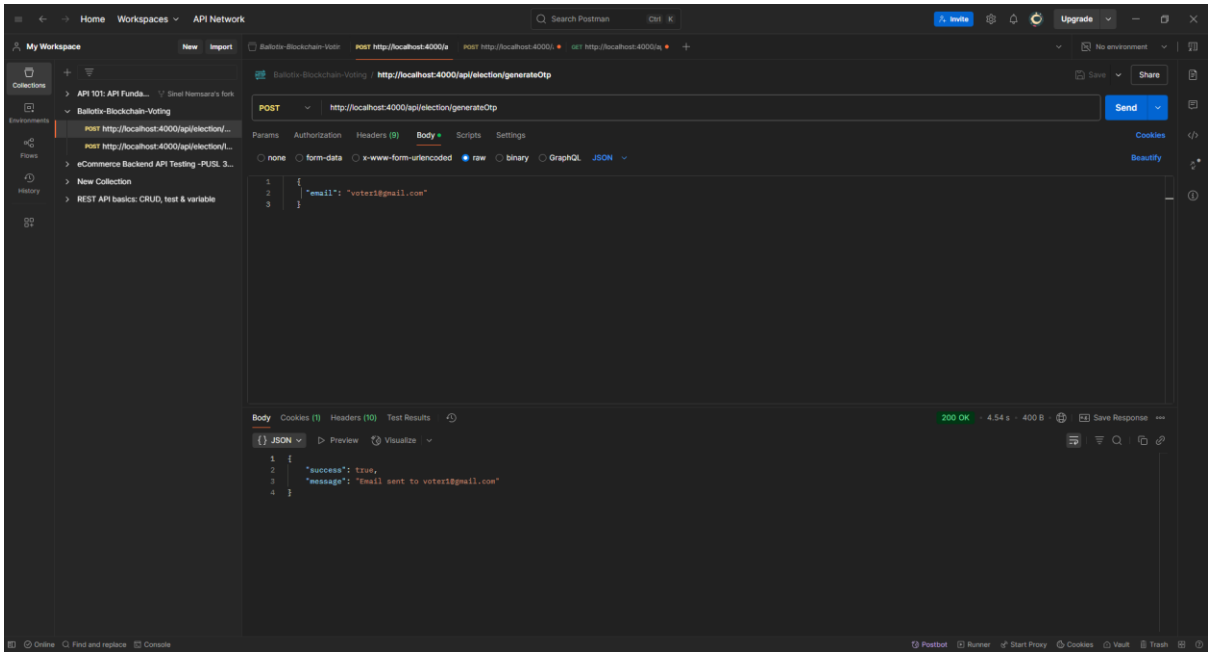


Figure 7: OTP Generation Endpoint Test

Validates the email-based OTP generation flow and confirms the successful trigger of the Mailtrap and SendGrid services.

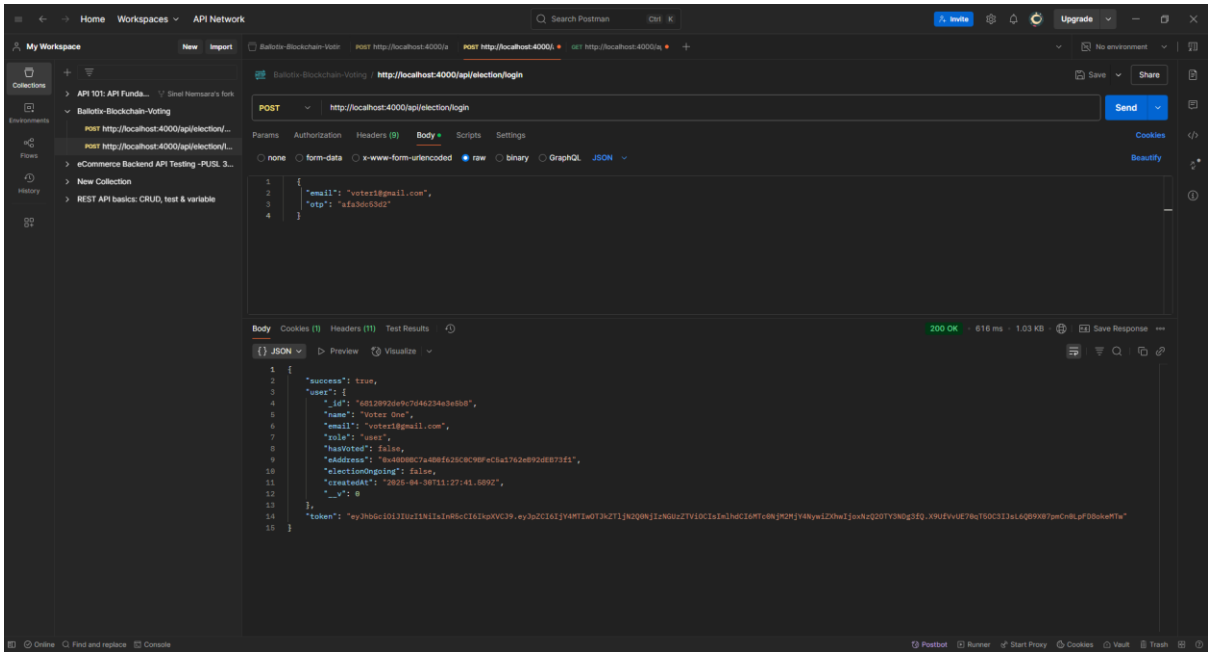


Figure 8: Login Endpoint with OTP Verification Test

Demonstrates successful login with email and OTP, returning a JWT token and user profile metadata. These tests validated the complete user login flow and confirmed secure backend handling of credentials and token generation. Edge cases such as invalid OTPs and empty inputs were also tested.

4.8.2 Frontend Testing

Frontend component testing was implemented using **Jest** and **React Testing Library**. The focus was on verifying user interaction flows, validation messages, loading states, and API integration responses.

Testing tools used:

- **Jest**: JavaScript testing framework
- **React Testing Library**: DOM-oriented unit testing
- **Mock Axios**: For simulating backend interactions
- **Mock React Router and Contexts**: To simulate navigation and authentication

1. Tested Functionalities (Login Component)

- Rendering of the email input and OTP input fields
- Form submission with empty inputs (error alerts)
- OTP request success and failure flows
- Successful login using mock OTP and email
- Countdown disabling of the OTP button
- Loading indicators during API calls

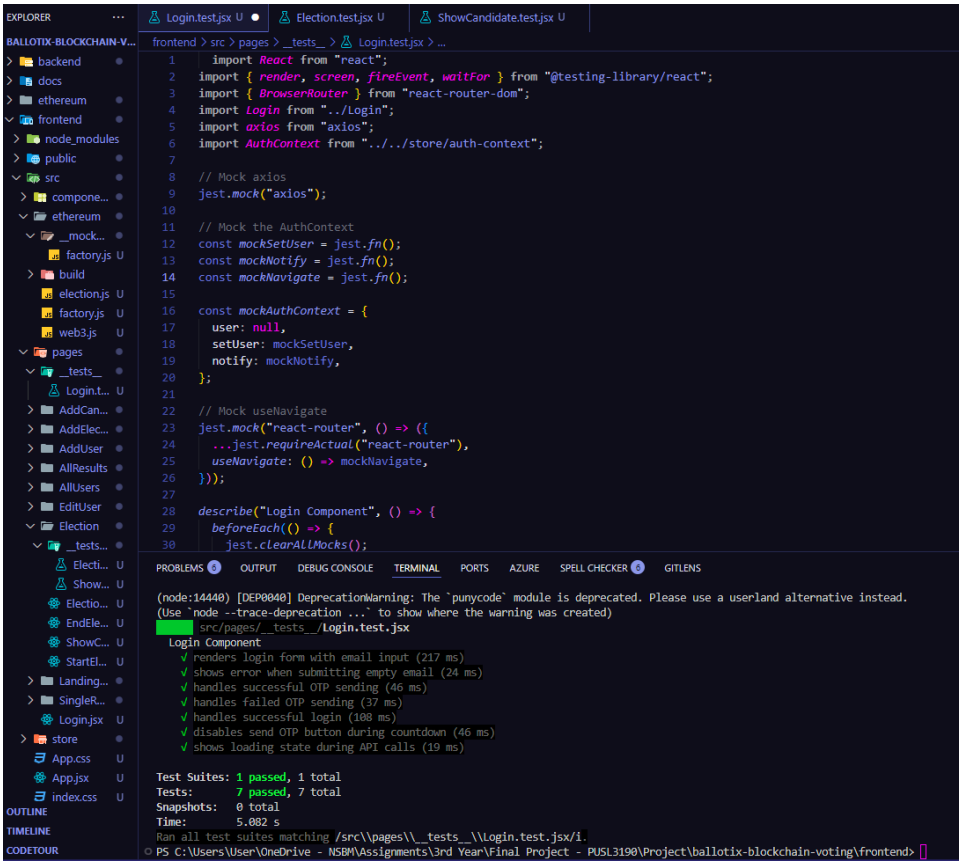


Figure 9: Frontend Unit Test Results for Login Component

These tests are part of the automated testing pipeline and can be re-run using the `npm test` command. Additional test scripts for other frontend components (eg., Election, ShowCandidate) are included in appendix 8 to demonstrate full test coverage.

5. Requirements

This chapter outlines the functional, non-functional, and environmental (hardware/software) requirements that guided the design and development of the Ballotix blockchain-based voting system. These requirements were identified through stakeholder analysis, survey data, and iterative prototyping.

5.1 Functional Requirements

Functional requirements specify the core behaviors and interactions the system must support. Ballotix is designed to serve two primary user roles that are **election administrators** and **voters** with distinct capabilities allocated to each.

Administrator Functionalities

- **Admin Authentication**: Authenticate securely via a wallet-based login system (MetaMask) and email verification.
- **Election Creation**: Create new elections with configurable names, timeframes, and descriptions.
- **Candidate Management**: Add, update, and remove candidates for an election, including image uploads and descriptions.
- **Election Control**: Start and end the election via smart contract interactions, enforcing state transitions.
- **Result Viewing**: View live vote counts and final results upon election conclusion.

Voter Functionalities

- **Voter Registration**: Register using a valid email address and verify identity through OTP-based authentication.
- **Wallet Integration**: Connect a valid Ethereum wallet via MetaMask to confirm voting eligibility.
- **Vote Casting**: Select a preferred candidate and cast a single vote per election, which is recorded immutably on the Ethereum blockchain.
- **Vote Confirmation**: Receive feedback on successful vote submission and confirmation of inclusion in the blockchain ledger.

System-Level Functionalities

- **Smart Contract Deployment**: Dynamically deploy and interact with smart contracts representing each election instance.
- **Event Handling**: Monitor smart contract events (e.g., `VoteCast`, `ElectionEnded`) for real-time frontend updates.
- **Result Tallying**: Automatically determine the winning candidate based on blockchain-verified vote counts.

5.2 Non-Functional Requirements

Non-functional requirements define quality attributes and operational constraints of the system. This ensures Ballotix is secure, scalable, maintainable, and user-friendly.

- **Security**:
    1. All votes must be cryptographically signed using the voter's Ethereum wallet.
    2. Admin-only operations (e.g., ending the election) are protected using Solidity access modifiers.
    3. OTP-based email verification adds a second layer of identity validation.
- **Transparency**:
    1. All vote transactions are publicly viewable on the blockchain, ensuring a tamper-proof audit trail.
    2. Contract events and election states are visible in the frontend in near real-time.
- **Anonymity and Privacy**:
    1. Voter identity is not linked to their vote in the smart contract to maintain ballot secrecy.
    2. Only hashed or encrypted metadata is stored in the backend for auxiliary services.
- **Usability**:
    1. The system must have a clean, responsive, mobile-friendly UI built with React and Chakra UI.
    2. MetaMask integration is intuitive, requiring no blockchain knowledge from end users.
- **Performance**:
    1. Blockchain interactions must complete within acceptable latency (typically <10s).
    2. Backend APIs should return results within 2s under normal load.
- **Scalability**:
    1. The system must handle elections with up to 1,000 voters and 10–20 candidates without degradation.
    2. Modular smart contracts allow for extension without rewriting the core logic.
- **Maintainability**:
    1. Backend and frontend components must be modular and well-documented.
    2. Smart contract logic should be reusable across different elections with minimal changes.
- **Reliability**:
    1. Backend services must provide fault tolerance through retry mechanisms and error logging.
    2. Smart contract state changes are irreversible, reducing the risk of transactional failure.

5.3 Hardware and Software Requirements

The following outlines the minimum and recommended system environments for development and deployment.

5.3.1 Hardware Requirements

Table 4: Hardware Requirements

Resource	Minimum Specification	Notes
Processor	Intel i5 / AMD Ryzen 5 or higher	Required for running local blockchain node and development environment
RAM	8 GB (16 GB recommended)	16 GB recommended for running multiple services (frontend, backend, blockchain)
Storage	50 GB free disk space	Required for node_modules, blockchain data, and development tools
Network	Stable internet connection	Required for blockchain interactions and API calls
GPU (optional)	Not required unless scaling UI/UX	No heavy graphics processing needed

5.3.2 Software Requirements

Table 5: Software Requirements

Component	Technology / Platform	Version/Details
Frontend	React.js, Vite	React 18.2.0, Vite 6.3.4
Frontend UI	Tailwind CSS, Framer Motion	Tailwind 3.3.5, Framer Motion 12.9.4
Backend	Node.js, Express.js	Node.js 18+
Database	MongoDB (Atlas Cloud)	Cloud-hosted MongoDB instance
Blockchain	Ethereum (Sepolia Testnet)	Web3.js 4.16.0
Smart Contract Tools	Solidity, Web3.js	Solidity for contract development
Development Tools	• Jest (Testing) • ESlint (Linting) • Postman (API Testing) • MetaMask (Wallet)	• For unit and integration testing • For code quality • For API development and testing • For blockchain interactions

Image Hosting	Cloudinary	-
Email Services	SendGrid / Mailtrap (dev)	-

Deployment scripts are optimized for platforms like Render and Netlify. The Ethereum testnet environment ensures zero-cost blockchain operations during development, while future migration to a mainnet or Layer-2 scaling platform remains feasible.

6. System Design

This chapter outlines the architectural and design elements of the Ballotix blockchain-based voting system. It captures the structural, technological, and interactional components that form the foundation of the system's functionality. Each diagram presented in this section serves to illustrate a specific design aspect, helping to visualise the internal operation and external communication of system modules.

6.1 System Architecture

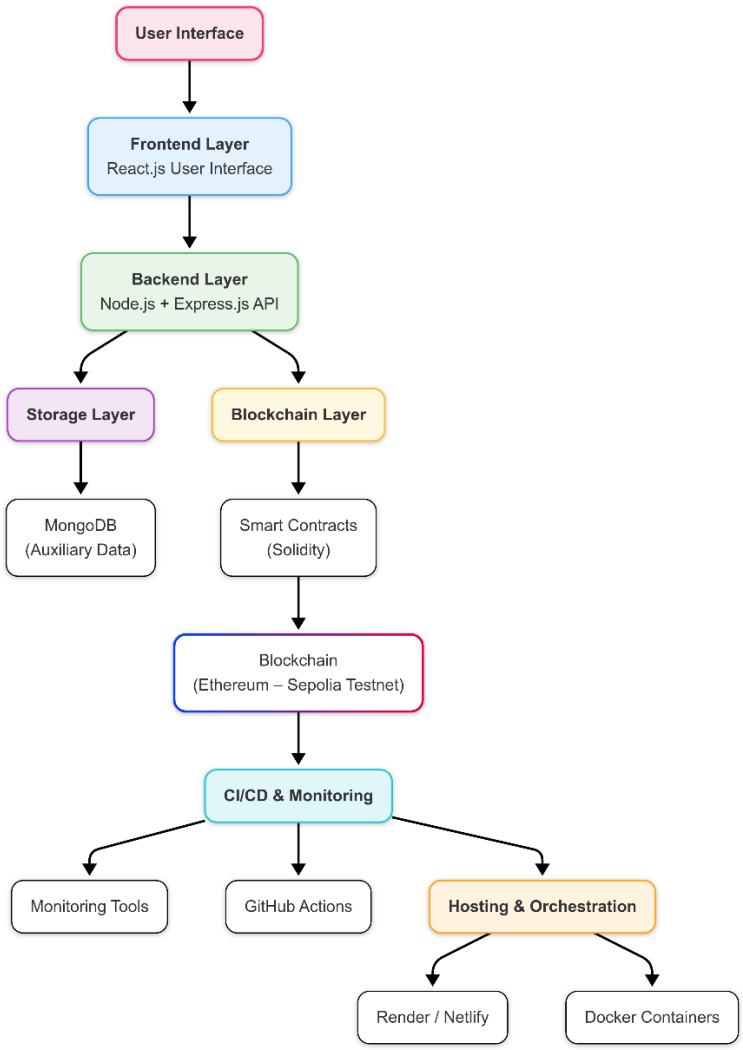


Figure 10: High-Level System Architecture Diagram

The system utilises a multi-tiered architecture style which separates all user interactions, application logic, data management, and blockchain functions into different layers. The uppermost layer, which is the user interface, is built with React.js to allow easy interaction for both voters and administrators through a flexibly responsive web interface. This links to a Node.js and Express.js backend that takes care of authentication, OTP verification, data validation, and blockchain interfacing.

Auxiliary non-sensitive data like voters' emails and OTP session logs are stored in MongoDB for data persistence. Logic pertaining to blockchains such as the management of elections, casting of votes, and computation of results is done through Solidity smart contracts on the Ethereum Sepolia testnet, forming the platform's contracts which decentralise trust.

The system is linked with CI/CD pipelines alongside minimal monitoring systems for automated testing and deployment. Continuous integration is done through GitHub Actions and backend hosting is done via Render while frontend hosting is done via Netlify. While Docker containerisation is planned to enhance future scalability, the current focus is on an easily deployable and lightweight structure. High-level system architecture is shown in Figure 10.

6.2 Use Case Diagram

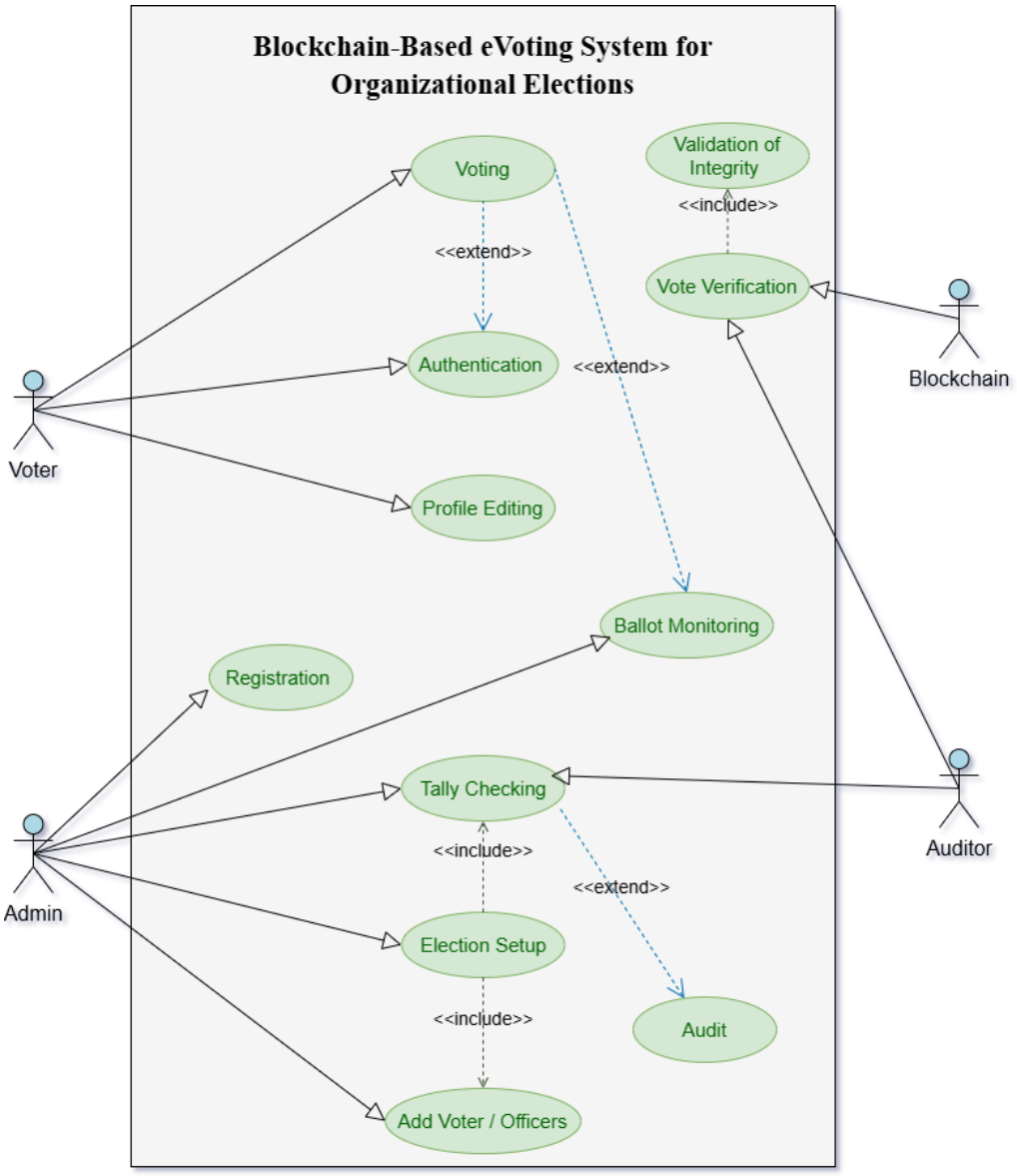


Figure 11: Use Case Diagram

The above use case diagram shows the most important connections between users and system parts. The main actors are Voter, Admin, Auditor, and the Blockchain. Every actor takes a particular action which is pertinent to their role such as registration, authentication, voting, and checking the tally.

The system allows the Voter to log in and vote, and if they choose, observe the ballot. The Admin takes care of configuration activities, like organising elections, controlling the list of candidates and voters, and starting the counting of results. The Blockchain is the trustless system that stores and verifies the votes.

6.3 Class Diagram

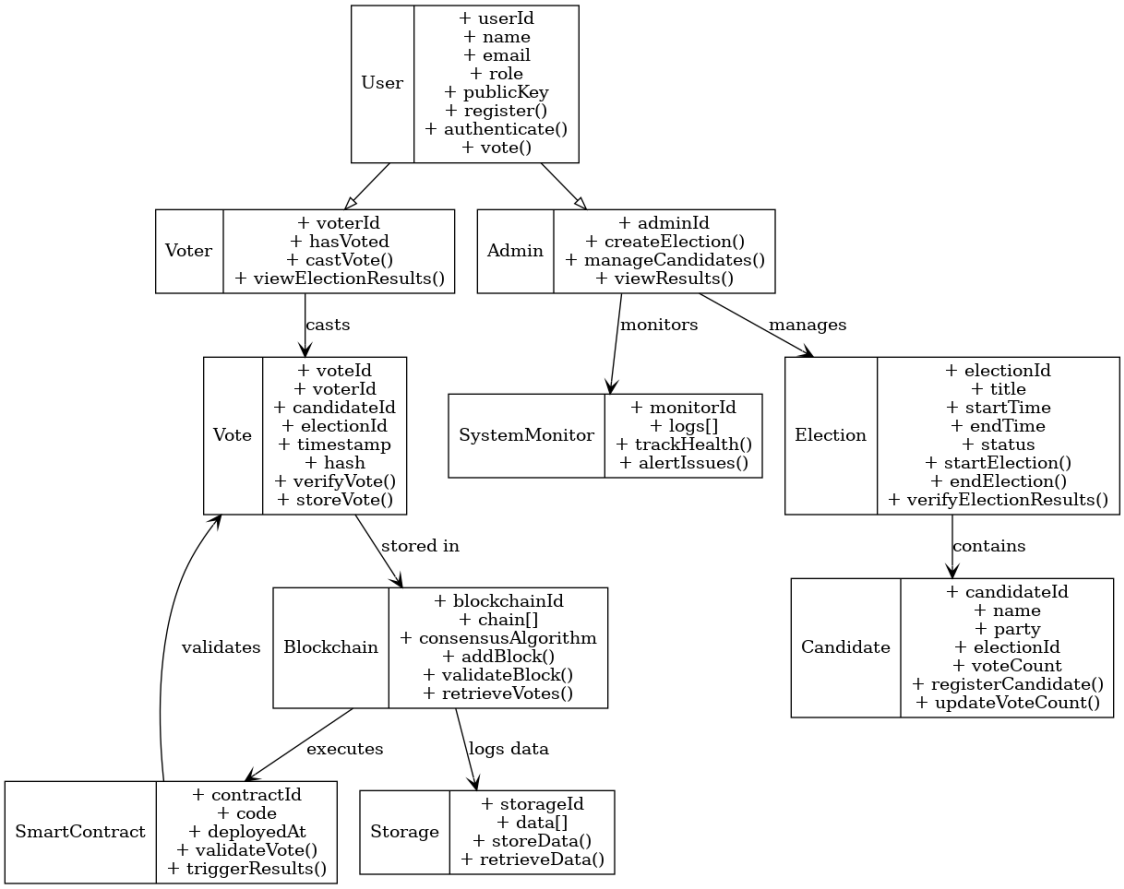


Figure 12: Class Diagram of the System

The above class diagram Illustrates the object-oriented design of the Ballotix voting system. It provides a high-level overview of the system's major entities, their attributes, and interrelationships.

At the core of the system is the abstract `User` class, which serves as the parent for both `Voter` and `Admin` roles. Voters are responsible for casting votes and viewing election results, while admins manage election setup, candidate registration, and result tracking. Votes are linked to elections and candidates and are validated and stored via smart contracts on the Ethereum blockchain.

The `Election` class contains key metadata, such as start and end times, and is linked to multiple `Candidate` entities. Each `Vote` includes attributes for tracking the voter's identity, selected candidate, and a hash for integrity verification.

The `Blockchain` class abstracts the functionality of storing validated votes, using the `SmartContract` class to execute secure transactions and compute results. Auxiliary data such as user metadata and OTP logs are handled by the `Storage` class, which interfaces with MongoDB.

6.4 Entity-Relationship Diagram

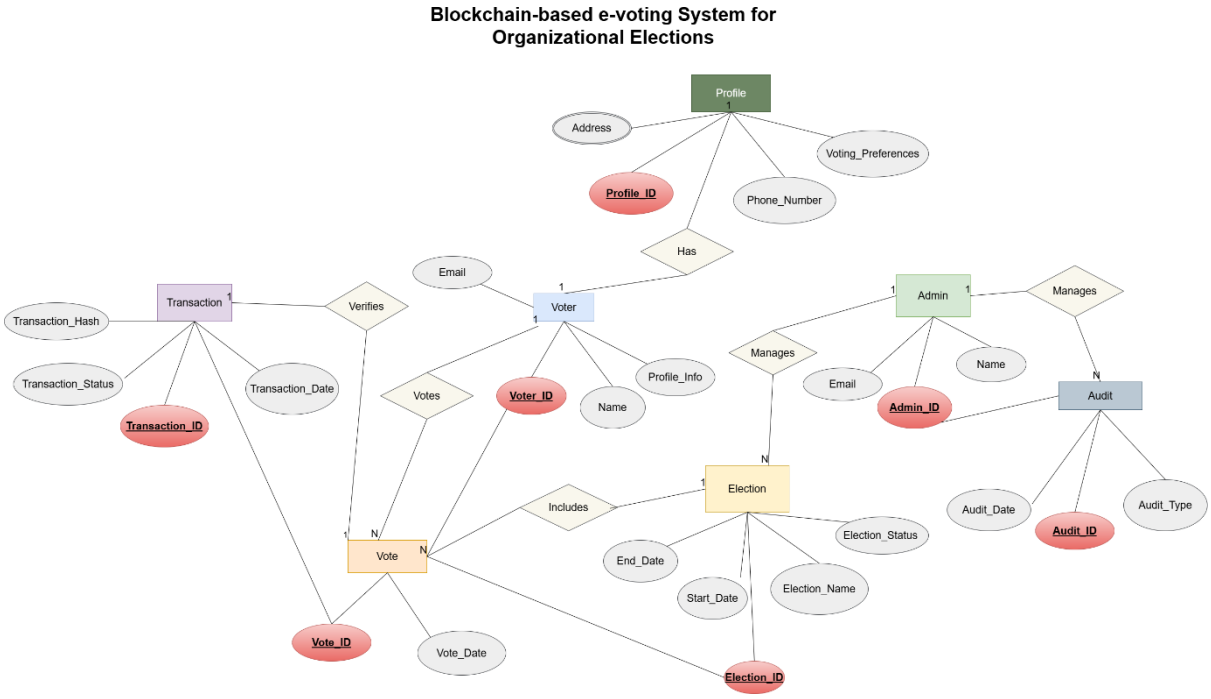


Figure 13: Entity-Relationship Diagram

The Entity-Relationship Diagram (ERD) depicted in Figure 13 outlines the logical data model used in the Ballotix voting system. It captures the relationships between key data entities such as users, elections, votes, and blockchain transactions.

Each voter is associated with a profile entity containing extended information such as contact details and preferences. The system ensures that each vote is uniquely linked to a specific election and voter and is further validated through a corresponding blockchain transaction. Administrators are responsible for managing elections and associated audit logs, which track system events and verification processes. The data model was designed to support both operational efficiency and future scalability. While core fields like voter identity, election metadata, and vote records are already in use within the MongoDB schema, additional entities like `Audit` and `Profile` are intended to support extended administrative features and voter analytics in subsequent development phases.

7. End-Project Report

7.1 Project Overview & Achievements

The Ballotix project set out to design and develop a secure, transparent, and user-centric voting platform leveraging blockchain technology for organisational elections. The core motivation was to address longstanding issues in traditional and centralised electronic voting systems, such as lack of transparency, single points of failure, and trust deficits. By adopting a decentralised architecture powered by the Ethereum blockchain and integrating modern web technologies, Ballotix aimed to demonstrate a practical, secure alternative tailored for use in non-governmental contexts like universities, corporate boards, and associations.

At the conclusion of the development phase, the following core deliverables were successfully implemented:

- A fully functional blockchain-based voting application, supporting admin and voter roles
- Smart contracts developed and deployed to the Ethereum Sepolia test network, enabling vote recording, election lifecycle management, and result calculation
- A MetaMask-integrated React frontend with OTP-based email verification for secure voter onboarding
- A Node.js and Express backend to manage authentication, image uploads, and auxiliary services
- Real-time result fetching via smart contract events and event-driven frontend updates
- Successful API and frontend testing using Postman, Jest, and React Testing Library

Despite some pending components like comprehensive smart contract testing and implementation of zero-knowledge proofs (ZKPs) to enhance voter anonymity, the project has reached a demonstrably functional and robust state that validates its original vision and feasibility.

7.2 Evaluation of Objectives

The Ballotix project began with a clear set of technical and strategic objectives aimed at addressing known issues in electronic voting systems, particularly around centralisation, trust, and auditability. This section provides a critical evaluation of whether these objectives were met and highlights any significant deviations from the initial plan.

One of the primary objectives was to design and develop a decentralised voting platform using blockchain technology. This was successfully achieved through the implementation of Ethereum-based smart contracts and MetaMask wallet integration, which collectively eliminated the need for a central authority in vote recording and result computation. The election lifecycle creation, candidate registration, voting, and result publication was also fully

implemented using Solidity smart contracts, with real-time interaction enabled via a React-based frontend and Web3.js.

The project further aimed to ensure secure, single-vote-per-user enforcement. This was realised by combining OTP-based email verification with Ethereum wallet checks. The smart contract maintained a mapping of voter addresses to voting status, effectively preventing duplicate voting and unauthorised access. The use of smart contract access control also ensured that sensitive actions such as ending the election were restricted to administrative users.

Another key goal was to develop a responsive and intuitive frontend interface that would make the system accessible to non-technical users. This was successfully delivered through the use of React, Chakra UI, and tailored UX patterns, making tasks such as wallet connection, vote casting, and result viewing seamless and user-friendly across devices.

Comprehensive testing was another defined objective. While extensive API and frontend testing was carried out using tools like Postman, Jest, and React Testing Library, formal smart contract testing using Mocha and Chai is still in progress. Although not yet fully complete, plans are in place to conduct contract-level testing before deployment.

One objective that was not achieved in this phase was the use of zero-knowledge proofs (ZKPs) to enhance voter anonymity. The initial plan included integrating ZKP-based mechanisms to preserve ballot secrecy while maintaining vote integrity. However, due to the time constraints, technical complexity, and the lack of ready-to-use Solidity-based ZKP libraries, this enhancement was deferred. The current system supports transparent vote tracking while maintaining basic identity obfuscation by avoiding direct storage of voter details on-chain.

Additionally, the proposed use of AWS hosting and IPFS-based file storage was replaced with simpler alternatives such as Render for web hosting and Cloudinary for media storage. This change allowed the project to meet its delivery timeline and maintain a working deployment pipeline, albeit with fewer advanced infrastructure features.

Overall, the project met the majority of its functional and architectural objectives and demonstrated the viability of a decentralised voting solution for organisational use. Where deviations occurred, they were pragmatic and made in favour of stability, usability, and time management, with a clear roadmap for future improvements.

7.3 Realisation of Business Objectives

From a strategic perspective, the project successfully addressed the business and ethical concerns surrounding organisational elections. The platform enhances trust by providing an immutable record of every vote on a public blockchain. It reinforces security and privacy through cryptographic signing of votes and decoupling voter identity from vote content.

Further, the admin panel and real-time result interface streamline election management, reducing overheads for organisations. By keeping the platform open to wallet-based interactions, the system remains extensible and future-proof, aligning with long-term goals of scalability and adaptability.

The true business value lies in demonstrating that blockchain technology often criticized as overly complex or speculative can be pragmatically applied to solve real-world issues with clarity, transparency, and control.

7.4 Changes Made During the Project

Several key changes were made over the course of the project in response to feasibility considerations, time constraints, and development complexity:

1. **Removal of IPFS & AWS Hosting from Initial Deployment**

   Originally, files (e.g., candidate images) were to be hosted on IPFS and the platform deployed on AWS. These were substituted with **Cloudinary** and **Render** respectively, enabling smoother integration and eliminating setup overhead.

2. **Testing Adjustments**

   The initial testing plan included full unit and integration tests for smart contracts using Mocha/Chai. However, due to parallel development tasks, emphasis shifted to **API and frontend testing**. Smart contract testing is currently in progress.

3. **Smart Contract Simplification**

   The contract logic was refactored during development to eliminate early design redundancies and ensure more efficient event handling and gas optimisation on the Sepolia testnet.

4. **Frontend Experience Improvements**

   Usability was enhanced through more detailed loading indicators, real-time error feedback, and design refinements using Chakra UI and Framer Motion. These improvements were prioritised based on feedback from early testers and developers.

5. **Deferred Implementation of Cryptographic Anonymity Features**

   One of the initially proposed enhancements involved integrating zero-knowledge proof (ZKP) mechanisms to provide stronger voter anonymity while preserving eligibility verification. Due to the complexity of ZKP integration in smart contracts and the lack of readily available libraries within the Solidity ecosystem, this feature was postponed. However, work on this enhancement is ongoing, and the system architecture has been designed to accommodate future integration of ZKP-based vote masking and validation techniques.

These changes were necessary trade-offs to ensure delivery of a stable, functional system within the academic project timeline. Importantly, they were made with a long-term vision in mind, and the architecture remains open to these features being added in future iterations.

8. Project Post-Mortem

The post-mortem section offers a critical reflection on the completed project, assessing the decisions made, the processes followed, and the outcomes achieved. This retrospective evaluation highlights areas of success, identifies limitations, and captures lessons that can inform future project endeavors.

Evaluation of Project Objectives

The primary objectives of the project are to develop a secure, transparent, and accessible blockchain-based voting system for organizational elections were largely appropriate and well-aligned with both the academic aims and real-world challenges of voting systems. Each goal was intended to address an existing shortcoming in traditional voting infrastructure. While the implemented system effectively delivered core functionalities such as secure voter authentication, immutable vote storage, and real-time transparency, a few intended enhancements like advanced privacy mechanisms (e.g., Zero-Knowledge Proofs) remained in the research or early experimentation phase due to time constraints and implementation complexity.

Specification and Alignment with Business Goals

The product was appropriately specified with features tailored to improve trust, accessibility, and efficiency in organizational voting contexts. These included Ethereum-based smart contracts, email OTP-based authentication, and IPFS-based decentralized vote storage. However, scope adjustments during development specifically regarding UI refinements and planned integrations with identity systems meant some specifications were re-scoped for practicality. Despite this, the final system successfully reflects transparency and integrity as outlined in the business objectives.

Client Relationship and Feedback

As this project was not developed for an external client but for academic purposes, client-like roles were fulfilled via supervisor interactions, peer feedback, and target user surveys conducted during the requirement gathering phase. Feedback gathered during early surveys indicated strong interest in secure and anonymous digital voting mechanisms, validating the core design decisions. Supervisor feedback was also instrumental in shaping the technical depth and alignment with academic standards.

Evaluation of Development Process

The Agile methodology was effective for rapid prototyping and iterative feature delivery. However, more structured sprint planning could have improved final-stage integration and testing efficiency. While core features were implemented on time, overlapping deadlines occasionally limited focus on refinements such as frontend polish and documentation.

Technologies Chosen

The selected tech stack Ethereum, Solidity, React, Node.js, MongoDB was well-suited for the project's objectives. Integration of MetaMask and Web3.js ensured secure blockchain interactions. Planned cryptographic enhancements like Zero-Knowledge Proofs were deferred due to technical complexity but remain under active research for future inclusion.

Personal Performance

Time management, problem-solving, and full-stack coordination were handled effectively. Challenges arose in smart contract testing and last-mile polish, primarily due to concurrent deliverables. Earlier allocation of time for system evaluation and deployment would have reduced final-stage bottlenecks.

Lessons Learned

Key lessons include:

- Prioritise testing earlier in the lifecycle.
- De-risk complex technologies with clearer feasibility scoping.
- Maintain flexible planning to adapt to scope shifts.
- Continuous documentation throughout the project cycle reduces project implementation and documentation overlapping.

The project successfully demonstrated the feasibility of decentralised voting in a real-world context and laid a scalable foundation for future privacy enhancements, including ZKP integration.

9. Conclusion

This blockchain-based voting system project sets out to explore the application of blockchain technology to the pressing need for transparent, secure, and verifiable digital voting in organisational settings. At its core, the project aimed to replace centralised vote counting and data management processes with a decentralised, immutable ledger that can foster trust, reduce manipulation risks, and enhance accessibility for both administrators and voters.

Over the course of its development, Ballotix successfully implemented key functional components that reflect its original vision. These include a Solidity-based smart contract infrastructure deployed on the Ethereum Sepolia testnet, MetaMask integration for secure wallet-based voter authentication, OTP-based identity verification using email, a full-stack React and Node.js web application, and real-time election result tracking through blockchain event listening. Additional third-party services, including Cloudinary and SendGrid, were also integrated to enhance media handling and communication features. A range of backend and frontend functionalities were tested to ensure system reliability and correctness under typical usage scenarios.

The project has not only demonstrated the technical feasibility of decentralised voting but has also highlighted its practical value in addressing long-standing limitations of existing electronic voting systems. By enabling immutable vote recording and transparent result verification, this voting system contributes towards a more accountable digital voting experience. However, the project also encountered certain limitations. Notably, the planned integration of Zero-Knowledge Proofs (ZKPs) to improve voter anonymity while promising in theory proved to be beyond the scope of the current implementation due to its technical complexity and the limited availability of stable Solidity-based ZKP libraries. This enhancement remains an active area of investigation and is earmarked for future development cycles.

Several adjustments were made throughout the project lifecycle to accommodate realistic timeframes and streamline development. Originally proposed components such as AWS-based deployment and IPFS for file storage were substituted with Render and Cloudinary to simplify infrastructure setup and reduce technical overhead. These decisions ensured a functional and demonstrable system could be delivered within the project's constraints.

In reflecting on the overall journey, Ballotix has achieved its core goals and established a scalable, modular foundation upon which future improvements can be made. The next steps planned include completing smart contract test coverage, implementing ZKPs for enhanced voter anonymity, and exploring user-focused extensions such as multilingual support and voter analytics.

In conclusion, the Ballotix platform demonstrates the viability of blockchain-based voting solutions within non-governmental, organisational contexts. It provides a practical alternative to conventional systems like paper ballots and offers a strong foundation for further research, academic inquiry, and real-world adoption.

References

Abdul, M. & Saleem, S., 2024. *Navigating Blockchain's Twin Challenges: Scalability and Regulatory Compliance.* [Online]
Available at: https://www.mdpi.com/2813-5288/2/3/13

Aidynov, T. & Goranin, N., 2024. *A Systematic Literature Review of Current Trends in Electronic Voting System Protection Using Modern Cryptography*. [Online]
Available at: https://www.mdpi.com/2076-3417/14/7/2742
[Accessed 29 October 2024].

Berenjestanaki, M. H., Barzegar, H. R., El Ioini, N. & Pahl, C., 2024. *Blockchain-Based E-Voting Systems: A Technology Review*. [Online]
Available at: https://www.mdpi.com/2079-9292/13/1/17
[Accessed 2024].

Berenjestanaki, M. H., R. Barzegar , H., El Ioini , N. & Pahl, C., 2023. *Electronics | Free Full-Text | Blockchain-Based E-Voting Systems: A Technology Review*. [Online]
Available at: https://www.mdpi.com/2079-9292/13/1/17/review_report
[Accessed 15 October 2024].

Bhasi, M., Chandrasekaran, K. & Pandey, A., 2019. *VoteChain: A Blockchain Based E-Voting System.* [Online]
Available at: https://ieeexplore.ieee.org/abstract/document/8978295
[Accessed 20 October 2024].

Chai, J. & Ohio State University, 2020. Blockchain based voting system with Ethereum Blockchain. *Research Thesis,* p. 25.

Cole, J., 2024. *Blockchain Development and Security: Best Practices - BlockApps Inc.*. [Online]
Available at: https://blockapps.net/blog/security-best-practices-in-blockchain-development/
[Accessed 25 October 2024].

Cole, J., 2024. *The Impact of Blockchain on Voting Systems and Governance - BlockApps Inc.*. [Online]
Available at: https://blockapps.net/blog/the-impact-of-blockchain-on-voting-systems-and-governance/

Collier, K., 2020. *Voatz smartphone voting app has significant security flaws, MIT researchers say*. [Online]
Available at: https://www.nbcnews.com/tech/security/voatz-smartphone-voting-app-has-significant-security-flaws-mit-researchers-n1136546

Collier, K. & NBC News, 2020. *Voatz smartphone voting app has significant security flaws, MIT researchers say,* s.l.: s.n.

Council, B., 2023. *Role of Blockchain in Cybersecurity.* [Online] Available at: https://www.blockchain-council.org/blockchain/blockchain-in-cybersecurity/ [Accessed 29 October 2024].

Daraghmi, E., Hamoudi, A. & Helou, M. A., 2024. *Decentralizing Democracy: Secure and Transparent E-Voting Systems with Blockchain Technology in the Context of Palestine.* [Online]
Available at: https://www.mdpi.com/1999-5903/16/11/388
[Accessed 29 October 2024].

Groopman, J. & Insights, K., 2023. *8 best practices for blockchain security.* [Online] Available at: https://www.techtarget.com/searchsecurity/tip/8-best-practices-for-blockchain-security

Hjalmarsson, F. P., Hreioarsson, G. K., Hamdaqa, M. & Hjalmtysson, G., 2018. *Blockchain-Based E-Voting System.* [Online]
Available at: https://ieeexplore.ieee.org/document/8457919/
[Accessed 07 October 2024].

Ohize, H. O. et al., 2024. Blockchain for securing electronic voting systems: a survey of architectures, trends, solutions, and challenges.

Pandey, A., Bhasi, M. & Chandrasekaran, K., 2019. *VoteChain: A Blockchain Based E-Voting System.* [Online]
Available at: https://ieeexplore.ieee.org/abstract/document/8978295
[Accessed 20 October 2024].

Pereira, B. M. B. et al., 2023. *Blockchain-Based Electronic Voting: A Secure and Transparent Solution.* [Online]
Available at: https://www.mdpi.com/2410-387X/7/2/27

Sharp, M., Njilla, L., Huang, C.-T. & Geng, T., 2024. *Blockchain-Based E-Voting Mechanisms: A Survey and a Proposal.* [Online]
Available at: https://www.mdpi.com/2673-8732/4/4/21
[Accessed 29 October 2024].

Tambanis, D., 2019. *Blockchain Applications: Election Voting - Blockchain Philanthropy Foundation - Medium.* [Online]
Available at: https://medium.com/bpfoundation/blockchain-applications-election-voting-a1436e7d10cb
[Accessed 20 October 2024].

Wadowski, G. M., Otte, L. S., Bernardo, N. D. & Macht, G. A., 2023. A Comparative Study of Electronic Voting and Paper Ballot. p. 21.

Bibliography

Abdul, M. & Saleem, S., 2024. *Navigating Blockchain's Twin Challenges: Scalability and Regulatory Compliance.* [Online]
Available at: https://www.mdpi.com/2813-5288/2/3/13

Aidynov, T. & Goranin, N., 2024. *A Systematic Literature Review of Current Trends in Electronic Voting System Protection Using Modern Cryptography*. [Online]
Available at: https://www.mdpi.com/2076-3417/14/7/2742
[Accessed 29 October 2024].

Berenjestanaki, M. H., Barzegar, H. R., El Ioini, N. & Pahl, C., 2024. *Blockchain-Based E-Voting Systems: A Technology Review*. [Online]
Available at: https://www.mdpi.com/2079-9292/13/1/17
[Accessed 2024].

Berenjestanaki, M. H., R. Barzegar , H., El Ioini , N. & Pahl, C., 2023. *Electronics | Free Full-Text | Blockchain-Based E-Voting Systems: A Technology Review*. [Online]
Available at: https://www.mdpi.com/2079-9292/13/1/17/review_report
[Accessed 15 October 2024].

Bhasi, M., Chandrasekaran, K. & Pandey, A., 2019. *VoteChain: A Blockchain Based E-Voting System.* [Online]
Available at: https://ieeexplore.ieee.org/abstract/document/8978295
[Accessed 20 October 2024].

Chai, J. & Ohio State University, 2020. Blockchain based voting system with Ethereum Blockchain. *Research Thesis,* p. 25.

Cole, J., 2024. *Blockchain Development and Security: Best Practices - BlockApps Inc.*. [Online]
Available at: https://blockapps.net/blog/security-best-practices-in-blockchain-development/
[Accessed 25 October 2024].

Cole, J., 2024. *The Impact of Blockchain on Voting Systems and Governance - BlockApps Inc.*. [Online]
Available at: https://blockapps.net/blog/the-impact-of-blockchain-on-voting-systems-and-governance/

Collier, K., 2020. *Voatz smartphone voting app has significant security flaws, MIT researchers say*. [Online]
Available at: https://www.nbcnews.com/tech/security/voatz-smartphone-voting-app-has-significant-security-flaws-mit-researchers-n1136546

Collier, K. & NBC News, 2020. *Voatz smartphone voting app has significant security flaws, MIT researchers say,* s.l.: s.n.

Council, B., 2023. *Role of Blockchain in Cybersecurity.* [Online] Available at: https://www.blockchain-council.org/blockchain/blockchain-in-cybersecurity/ [Accessed 29 October 2024].

Daraghmi, E., Hamoudi, A. & Helou, M. A., 2024. *Decentralizing Democracy: Secure and Transparent E-Voting Systems with Blockchain Technology in the Context of Palestine.* [Online]
Available at: https://www.mdpi.com/1999-5903/16/11/388
[Accessed 29 October 2024].

Groopman, J. & Insights, K., 2023. *8 best practices for blockchain security.* [Online] Available at: https://www.techtarget.com/searchsecurity/tip/8-best-practices-for-blockchain-security

Hjalmarsson, F. P., Hreioarsson, G. K., Hamdaqa, M. & Hjalmtysson, G., 2018. *Blockchain-Based E-Voting System.* [Online]
Available at: https://ieeexplore.ieee.org/document/8457919/
[Accessed 07 October 2024].

Ohize, H. O. et al., 2024. Blockchain for securing electronic voting systems: a survey of architectures, trends, solutions, and challenges.

Pandey, A., Bhasi, M. & Chandrasekaran, K., 2019. *VoteChain: A Blockchain Based E-Voting System.* [Online]
Available at: https://ieeexplore.ieee.org/abstract/document/8978295
[Accessed 20 October 2024].

Pereira, B. M. B. et al., 2023. *Blockchain-Based Electronic Voting: A Secure and Transparent Solution.* [Online]
Available at: https://www.mdpi.com/2410-387X/7/2/27

Sharp, M., Njilla, L., Huang, C.-T. & Geng, T., 2024. *Blockchain-Based E-Voting Mechanisms: A Survey and a Proposal.* [Online]
Available at: https://www.mdpi.com/2673-8732/4/4/21
[Accessed 29 October 2024].

Tambanis, D., 2019. *Blockchain Applications: Election Voting - Blockchain Philanthropy Foundation - Medium.* [Online]
Available at: https://medium.com/bpfoundation/blockchain-applications-election-voting-a1436e7d10cb
[Accessed 20 October 2024].

Wadowski, G. M., Otte, L. S., Bernardo, N. D. & Macht, G. A., 2023. A Comparative Study of Electronic Voting and Paper Ballot. p. 21.

Appendixes

Appendix 1: User Guide

Ballotix Blockchain Voting System - User Guide

Introduction

This user guide provides step-by-step instructions for installing, configuring, and using the Ballotix Blockchain Voting System. It includes platform requirements, installation procedures, and essential usage guidelines for demonstration and deployment.

Minimum Platform Specification

- Processor: Intel i5 / AMD Ryzen 5 or higher
- RAM: 8 GB (16 GB recommended)
- Storage: 50 GB free disk space
- Network: Stable internet connection
- OS: Windows 10+, macOS 11+, or Linux
- Node.js: v18.x or higher
- Browser: Chrome/Firefox (MetaMask support)

Prerequisites

- Node.js (v18+) and npm
- Git for cloning the repository
- MetaMask browser extension
- MongoDB Atlas or local MongoDB
- SendGrid/Mailtrap account
- Cloudinary account (optional)

Installation Steps

1. Clone the Repository:
   git clone <your-repo-url>
   cd ballotix-blockchain-voting

2. Install Dependencies:
   - Frontend: cd frontend && npm install
   - Backend: cd ../backend && npm install
   - Smart Contracts: cd ../ethereum && npm install

3. Configure .env in frontend/backend with required credentials

4. Deploy Smart Contracts (Sepolia):
   npx hardhat compile
   npx hardhat run scripts/deploy.js --network sepolia
   Copy deployed address into .env

Running the Application

- Start Backend: cd backend && npm start
- Start Frontend: cd frontend && npm run dev

Access at http://localhost:5173

Using the System

- Registration/Login via Email & OTP
- MetaMask must be connected to Sepolia Testnet
- Admins: Create elections, add candidates, control lifecycle
- Users: View elections, cast votes, view results

Troubleshooting

- MetaMask not detected: Ensure correct network
- Email not received: Check spam/junk or email settings
- Contract issues: Confirm Sepolia ETH and correct contract address
- MongoDB error: Re-check URI and access settings

Demo & Reset

- Clear DB and redeploy contracts as needed
- Use admin tools to clear old elections

Support

For further help, refer to README.md or contact the development team.

Figure 14: User Guide

Appendix 2: Project Initiation Document (PID)

PUSL3190 Computing Individual Project

Project Initiation Document

Blockchain-Based Secured Voting System for Organizations

Supervisor: Mr. Gayan Perera

Name: Delwakkada Nemsara

Plymouth Index Number: 10898858

Degree Program: BSc (Hons) Computer Science

Table of Contents

Table of Figures

1. Introduction

Voting is fundamental to how organisations make decisions and ensure that everyone has a voice. It also promotes democratic participation and encourages collective decision-making. However, there are a number of drawbacks to conventional voting methods, such as paper ballots and centralised electronic voting systems, such as inefficiencies, security vulnerabilities, and issues with transparency. These challenges not only undermine the integrity of elections but also diminish trust among people. When people do not fully trust the voting process, it undermines the entire concept of democratic decision-making. It's clear that we need to develop better solutions that address these concerns properly.

The inspiration for this initiative comes from the growing recognition of blockchain technology as a transformative tool for improving security, transparency, and efficiency in wide range of domains. Blockchain's decentralised and immutable nature, as proved in financial applications such as cryptocurrencies, provides an unparalleled opportunity to revolutionise voting systems without a single point of failure and ensuring the integrity of election results. This technology offers immense potential for organizational elections, where the stakes are very high, but the regulatory complexities are more manageable compared to national elections.

The project's goal is to develop a blockchain-based voting system designed specifically for organisational elections, addressing major concerns including vote tampering, lack of transparency, and inefficiencies in vote counting and results tabulation. This technology aims to rebuild trust in organisational decision-making processes by exploiting blockchain's key properties such as decentralisation, immutability, and auditability.

The project's expected impact goes beyond addressing the immediate issues of organisational voting. It proposes a scalable and user-friendly platform that can be tailored to a variety of organisational scenarios, paving the door for greater use of blockchain-based voting systems. Furthermore, the system's emphasis on voter anonymity, real-time monitoring, and resource efficiency is consistent with current expectations for sustainable and privacy-preserving technology solutions. The project's goal with this initiative is to establish a standard for secure and transparent voting systems, as well as to contribute to the larger conversation about using technology for societal benefit.

2. Business Case

The following section discusses the project's necessity, the issues it addresses, and how it aligns with organizational goals.

2.1 Business Need

Traditional voting systems, whether paper or centralised electronic systems, are plagued by inefficiencies, high costs, and significant vulnerabilities. These include potential risks of vote tampering, lack of transparency, and the possibility of human error, all of which may undermine the legitimacy of elections. Such flaws not only lead to administrative burdens but also diminish trust among stakeholders, which is particularly harmful in organizational elections where decision impact governance, strategy, and staff engagement.

According to a Deloitte study, 83% of employees in large organizations consider transparent decision-making is vital for trust in leadership. Furthermore, a 2019 Pew Research Centre survey found significant concerns regarding the accuracy and security of voting systems in both governmental and organizational contexts. These findings emphasise the urgent requirement for innovative approaches to restore confidence in the outcome of elections.

Blockchain technology offers a compelling alternative by preventing single points of failure, assuring tamper-proof data storage, and offering a transparent, immutable ledger. By overcoming these critical challenges, the proposed blockchain-based voting system addresses the prevailing demand for secure, efficient, and trustworthy election processes.

2.2 Business Objectives

The proposed project meets the organization's strategic objectives of improving decision making processes' scalability, operational efficiency, and trust. The specific business objectives are outlined below.

1. **Improve Election Security:** By implementing blockchain's cryptographic protocols into practice, it may reduce the risk of vote tampering and unauthorised access, enhancing the integrity of the election.
   Key metric: A 100% reduction in vote manipulation incidents compared to previous methods.

2. **Enhance Transparency and Accountability:** Utilise blockchain's open ledger that enables for real-time vote audits and monitoring, which will improve stakeholder trust.
   Key metric: 97.3% of voters and auditors report increased confidence in the election process.

3. **Optimise Resource Utilisation:** Through the utilisation of digital automation, the dependence on centralised infrastructure and physical materials can be reduced. Key Metric: A 50% reduction in operational costs compared to traditional voting methods.

4. **Promote Scalability and Flexibility:** Develop a system that can handle a range of organisational needs, such as modular adaptability for specific requirements and scalability for larger elections.

5. **Enhance Stakeholder Engagement:** Increase voter turnout by offering a user-friendly, accessible platform that ensures privacy and anonymity.

By aligning with these business objectives, the project addresses critical challenges while also achieving broader strategic goals for innovation and operational excellence.

3. Project Objectives

The project aims to design, develop, and implement a blockchain-based voting system that addresses key challenges in organisational elections. These objectives are focused on the specific deliverables and measurable outcomes that the project expects to accomplish.

1. **Design a Secure Architecture:** Implement a secure architecture for the blockchain-based voting system to prevent unauthorised access and maintain data integrity.

2. **Develop an Easy-to-Access and User-Friendly Interface for End Users:** Build a user-friendly voting interface that meets varied needs and prioritises security.

3. **Implement Cryptographic Protocols:** Integrate blockchain cryptographic techniques like hashing and digital signatures to secure voting records and secure data transmission.

4. **Real-Time Auditing and Monitoring:** Develop features for stakeholders to audit and verify vote results in real time.

5. **Evaluate Stakeholder Feedback:** Collect and analyse feedback from pilot users to fine-tune system features and ensure alignment with organisational requirements.

6. **Ensure Compliance with Legal and Regulatory Standards:** To avoid legal concerns, ensure that the system meets all applicable legal and regulatory requirements.

4. Literature Review

This section highlights existing research on blockchain-based voting systems, focusing on key contributions, limitations, and research gaps that the proposed project addresses. The review outlines the theoretical foundation for the project and demonstrates how the system aligns with previous findings while advancing the field.

4.1 Introduction

Blockchain technology has increasingly gained attention as a transformative solution for secure, transparent and efficient voting systems. Numerous research have explored blockchain's potential to overcome traditional challenges with voting like tampering, inefficiency, and transparency issues. Despite these advancements, significant gaps remain, requiring additional research in order to develop practical, scalable, and reliable solutions.

4.2 Existing Research

- **Scalability Challenges**

Most blockchain voting systems do not scale for large elections. For example, Voatz and similar platforms have shown success in small-scale implementation but experience performance bottlenecks in large-scale implementation due to high transaction loads in peak voting periods (Abdul & Saleem, 2024). Thus, there is a need for research which could provide scalable solutions capable of maintaining performance in high-stakes scenarios.

- **Transparency and Trust**

The aspects of public trust and transparency are crucial to the success of electronic voting systems. The current implementations often lack features that are important for the building of stakeholders' confidence such as real time audit trails and voter verification. Findings defines that trust levels vary across demographic groups and thus need tailored strategies to address such variances (Tambanis, 2019).

- **Decentralisation and Security**

While blockchain is inherently decentralised, many systems still include centralised components that jeopardise security. This reliance on centralised parts exposes risks that threaten data integrity (Hajian Berenjestanaki *et al.*, 2024). Developing fully decentralised architecture is critical to addressing these challenges.

- **Privacy and Anonymity**

Despite the fact that blockchain offers security, voter privacy and anonymity remain key issues of concern. The majority of these systems do not incorporate more complex cryptographic techniques, including zero-knowledge proofs, which helps to ensure the voters' identity is protected while the process is being validated (Aidynov *et al.,* 2024). Further studies of these strategies need to be conducted in order to effectively deal with privacy issues.

- **Real-Time Tracking and Auditability**

Existing systems lack comprehensive mechanisms for real-time vote tracking and auditability. This undermines the concerns of more transparent electoral processes and increases the chances of questioning the legitimacy of elections (Cole, 2024; Satybaldina et al., 2024). Integrating real-time tracking capabilities will provide an opportunity for significant improvement.

- **Regulatory Compliance**

The lack of clear legal frameworks for blockchain voting systems causes uncertainty among developers and stakeholders. Most of the current approaches do not fully comply with electoral regulations (Mohammed Abdul et al., 2024; Tambanis, 2019). Therefore, it's crucial to establish regulatory norms that apply innovation with democratic principles.

- **Education and Awareness**

Insufficient educational resources for stakeholders, including voters and election officials, limit blockchain voting system adoption. A lack of understanding of blockchain technology may hinder acceptance and usage (Tambanis, 2019). Educational initiatives are needed to raise awareness and support for these systems.

4.3 Research Gaps and Contribution

While existing research reveals the potential of blockchain in voting systems, significant shortcomings remain as follows.

- Insufficient scalability during large-scale elections.
- Inadequate mechanisms for transparency in order to ensure trust.
- Security risks associated with centralized components
- Limited implementation of active privacy-preserving cryptographic primitives.
- Absence of real-time tracking and audit capabilities.
- Lack of clarity regarding regulatory compliance
- Insufficient educational resources for stakeholders.

The proposed blockchain-based voting system overcomes these gaps by,

- Implementing scalable and decentralised architecture.
- Integrating real-time tracking and audit features to improve transparency.
- Utilising modern cryptographic approaches can improve privacy and security.
- Ensure compliance with legal and regulatory standards.
- Create a user-friendly interface to increase stakeholder engagement.
- Providing educational resources to promote understanding and trust.

4.4 Conceptual Diagram

A conceptual diagram will be attached below that depicts the connections between existing research, identified gaps, and the proposed system's contributions.

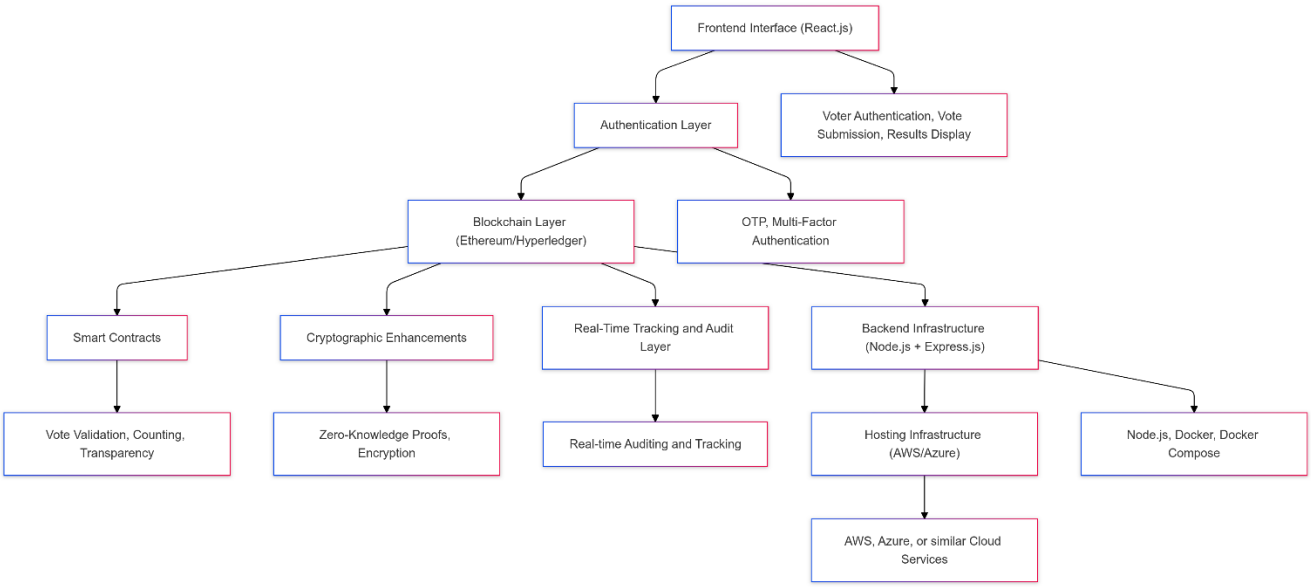


Figure 15: Conceptual Diagram

5. Method of Approach

This section outlines the methodology and tools to be utilized for developing the blockchain-based e-voting system. The approach will follow the Agile methodology to encourage flexibility, and iterative development, assuring that each functionality can be tested, refined and improved through the project life cycle. The agile methodology is compatible with the need for frequent updates, rapid adaptation to feedback, and the ability to provide the user with working increments of the system.

5.1 Agile Methodology

Agile approach was chosen based on its iterative nature and the level of flexibility needed to manage such project complexity. Tasks are divided into smaller units called sprints, which last 1-2 weeks. For every sprint performed, it delivers a working version of the product, with the capability for continuous development and rapid adaptation to changing requirements.

Reasons for Choosing Agile:

- **Flexibility in adapting to changes:** The project's scope may change with input once the project is in progression. Hence, it is critical that an approach be in place to accommodate new requirements or adjustments to existing ones.
- **Frequent communication and collaboration:** Agile prioritises frequent interactions between developers and stakeholders to ensure that the system meets user needs and expectations.
- **Incremental Development:** Agile facilitates the delivery of usable features at the end of each sprint, allowing stakeholders to see actual outcomes early on and providing an opportunity to incrementally improve the product.

5.2 Tools and Technologies

To make sure the development of a secure, scalable, and efficient blockchain-based voting system, we will utilise a combination of cutting-edge technologies addressing both functional and non-functional requirements.

- **Blockchain Platform:** Ethereum or Hyperledger will be used as the underlying blockchain platform. Ethereum is ideal for implementing smart contracts in a decentralised environment, enabling immutability and transparency of data (Abdul & Saleem, 2024). Alternatively, Hyperledger could be employed due to its flexibility, and it allows support for permissioned blockchains, which is convenient in an enterprise-level implementation.
- **Smart Contracts:** Solidity will be utilized to write smart contracts, that will automate and secure the voting process by making sure votes are valid and tamper-proof.

- **Frontend:** The user interface will be developed using React.js, ensuring a responsive, dynamic and user-friendly experience.
- **Backend:** For backend operations, we'll use Node.js and Express.js to develop RESTful APIs that enable seamless communication between the frontend and the blockchain network.
- **Storage:** IPFS (InterPlanetary File System) will be utilised for storing vote data in a decentralised manner, while MongoDB will be used for auxiliary data storage, that include user profiles and voting metadata.
- **Hosting:** The application will be hosted on AWS (using Amazon EC2) or Azure (using Azure Virtual Machines or Azure App Service) ensuring scalability, security, and availability. These platforms provide the infrastructure required to handle increased traffic and maintain system availability during critical election periods.
- **CI/CD and Testing:** GitHub Actions will automate continuous integration and continuous deployment, which leads to frequent code updates and testing. This will facilitate a more streamlined development process and prompt identification of faults.
- **Containerization and Orchestration:** Docker will be used to deploy the application consistently across environments, and Docker Compose will handle the orchestration to scale effectively during high traffic periods.

5.3 High-Level Architecture

The architecture of the system will be designed to ensure decentralization, scalability, and security. Following is the high-level architecture diagram illustrating how different components of the system will interact.

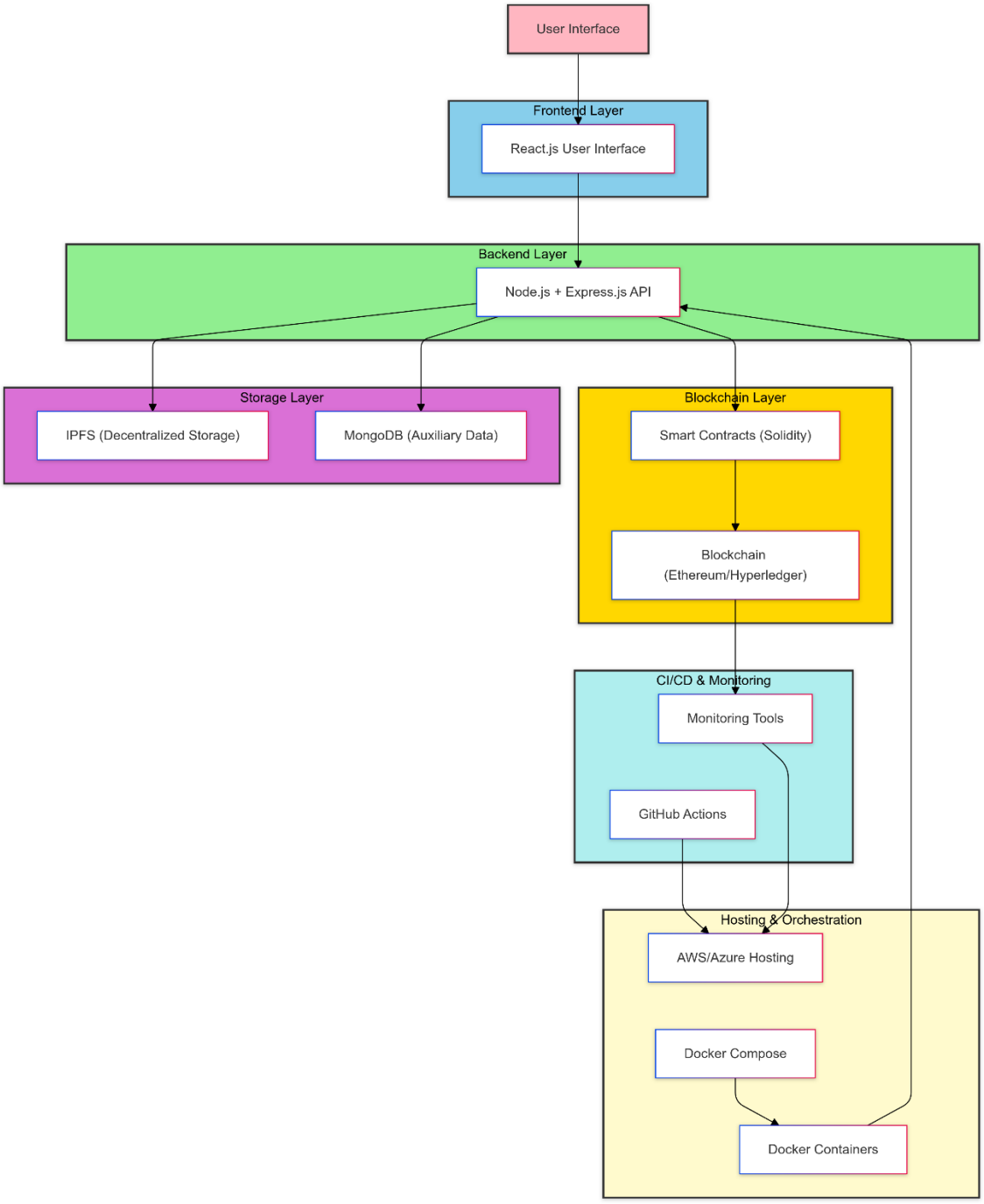


Figure 16: High-Level Architecture Diagram

1. **Front-end User UI:** Voters interact with the system using a responsive React.js interface, which allows them to vote and track their status in real time.

2. **Backend:** The Node.js backend processes frontend requests, interacts with the blockchain using smart contracts, and stores auxiliary data in MongoDB.

3. **Blockchain Layer:** The blockchain layer (either Ethereum or Hyperledger) securely records all voting transactions, ensuring immutability and transparency. Election regulations are followed, and votes are validated via smart contracts.

4. **Storage Layer:** IPFS stores encrypted voting data, while MongoDB manages non-sensitive data including user information and vote metadata.

5. **Cloud Hosting and CI/CD:** For scalability, the system is deployed on cloud platforms (AWS or Azure), and deployment workflows are automated using GitHub Actions. Docker containers enable portability, while Docker Compose manages orchestration.

6. Initial Project Plan

6.1 Gantt Chart

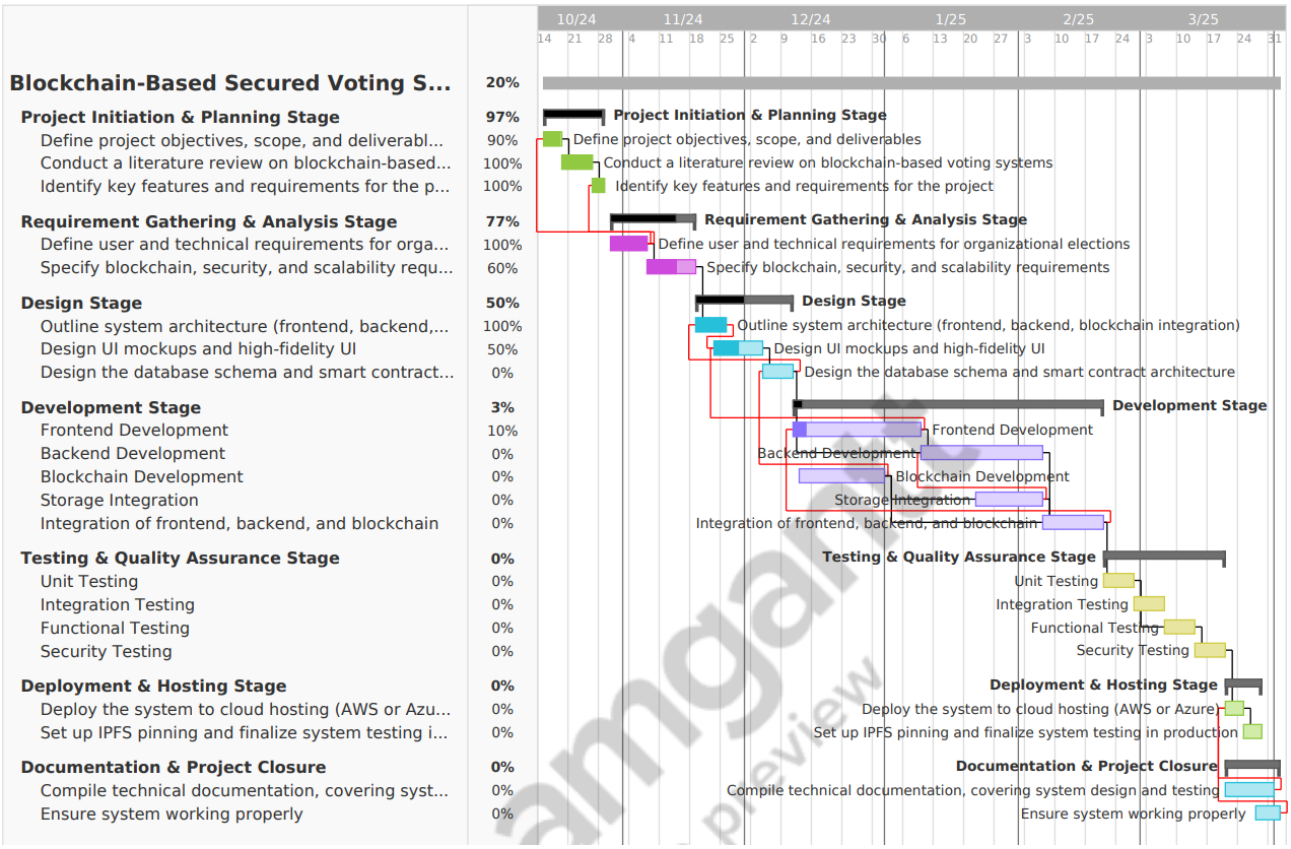


Figure 17: Gantt Chart

For a clear version of the timeline, visit: [Nemsara, D. (2025). Timeline of the Blockchain-based Voting System] (https://nsbm365-my.sharepoint.com/:b:/g/personal/dlsnemsara_students_nsbm_ac_lk/Ee0VvJrZWgNNhDP6v4xMXI8BEaXCrYfmSxxSyRqlse6cgg?e=vzEA68) [Accessed 5 Jan 2025].

The project's initial timeline comprises the key milestones that will be instrumental in guiding its development from initiation to final deployment. The attached Gantt chart clearly demonstrates the tasks, deadlines, and responsibilities. This is a structured iterative approach, with phases that include project initiation, requirement gathering, design, development, testing, deployment, and documentation.

Following is a high-level view of the key stages:

1. **Project Initiation and Planning (Oct 15 – Oct 29, 2024):**
   Define the project's objectives, scope, and deliverables, followed by a literature review to identify critical features and requirements.

2. **Requirement Gathering and Analysis (Oct 30 – Nov 19, 2024):**
   Collect and finalize user and technical requirements, with a focus on blockchain and security.

3. **Design (Nov 20 – Dec 10, 2024):**
   Design the system architecture, UI mockups, and smart contract structure.

4. **Development (Dec 11, 2024 – Feb 19, 2025):**
   Implement the front-end, back-end, and integration of storage systems.

5. **Testing & Quality Assurance (Feb 20 – Mar 19, 2025):**
   Perform unit, integration, functional, and security testing to ensure system functionality.

6. **Deployment & Hosting (Mar 20 – Mar 27, 2025):**
   Deploy to cloud infrastructure and conduct final system testing to ensure system functionality.

7. **Documentation & Final Report (Mar 28 – Apr 1, 2025):**
   Prepare technical documentation and a final report summarising key findings.

7. Risk Analysis

Risk analysis is one of the crucial factors to identify and address any concerns that could risk the successful implementation of the blockchain-based voting system. Since blockchain is a complex technology with various dependencies involved, certain risks are associated with it. Major potential risks are outlined below, along with their likelihood, potential impact, and their mitigating strategies.

7.1 Technology Risks

Technology risks stem from dependency on third-party services and libraries, thus which may encounter challenges such as version deprecation, compatibility issues, or downtime. This might disrupt the voting system's functionality.

Mitigation:

- Rely on trusted services and perform frequent system updates.
- Configure monitoring for quick issue detection and alternatives.
- Use Docker for consistent software packaging and deployment across different environments.

7.2 User Adoption Risks

Users may have difficulty understanding how to use the voting system efficiently, particularly if the user interface (UI) is not easy to understand.

Mitigation:

- Use user-centered design and iterative testing to enhance usability.
- Offer extensive user guidance, training, and support.

7.3 Privacy and Security Risks

Security flaws or breaches could compromise voter data, potentially leading to legal action and a loss of trust.

Mitigation:

- Perform robust cryptography measures and secure coding practices.
- Regularly audit security features and comply with data protection laws.

7.4 Budget Constraints

Unexpected costs, mainly related to cloud services or infrastructure, might exceed the budget that has been allocated.

Mitigation:

- Monitor expenses on a regular basis and optimise the utilisation of resources.

- Maintain a contingency fund to cover unexpected costs.

7.5 Scalability Risks

The system may struggle to handle large numbers of votes, impacting reliability and performance during peak times.

Mitigation:

- Build scalability into the system's design by leveraging cloud infrastructure.
- Load tests the system to ensure it can handle a large number of users.

7.6 Regulatory and Legal Risks

Legal concerns with data privacy or election laws could undermine the project's progress or deployment.

Mitigation:

- Consult with legal professionals to ensure complete compliance with applicable rules.
- Stay up to date on regulatory changes so that the system can adapt accordingly.

8. Additional Sections

8.1 Stakeholder Analysis

The primary stakeholders for the blockchain-based voting system, once deployed, are as follows.

1. **Voters (85%)**

Voters will be the most active user group, participating in internal elections within corporations, universities, and non-profit organizations. This group requires a seamless and secure voting experience.

2. **Election Administrators (7.5%).**

These stakeholders manage the election process and will rely on the system for election setup,

voter authentication, and real-time tracking.

3. **IT/Admin Support (7.5%)**

The IT staff will oversee system maintenance, security, and troubleshooting. They will require a reliable, secure system, as well as proper training for their role.

Organization Types:

- Corporate (70%)
- Academic (15%)
- Non-profit (7.5%)
- Government (7.5%)

The system will primarily target corporate organizations but will also cater to academic, non-profit, and government elections. The conclusions from this survey are, voters need a user-friendly interface, whereas election administrators and IT support teams require comprehensive administrative tools.

Further analysis of these results is provided in the survey section.

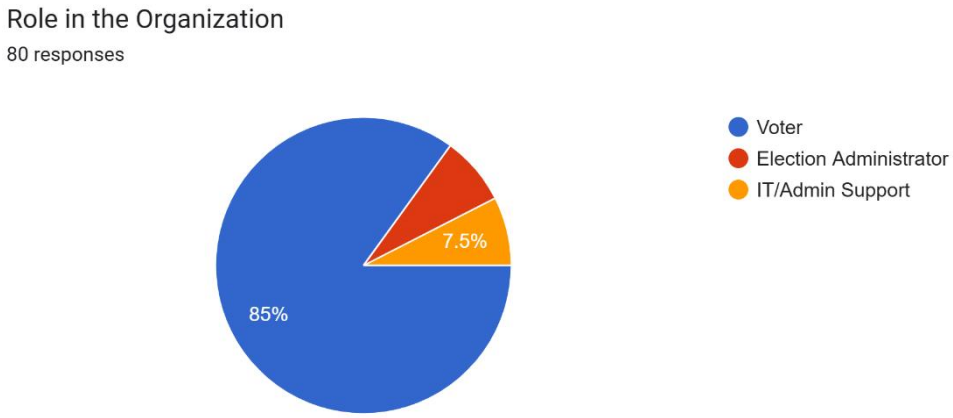
8.2 Budget Justification

The budget for this project primarily covers the costs of hosting services, backend servers, and decentralized storage. These costs are required to ensure the system's availability, security, and functionality. The estimated monthly cost is provided below.

Tool/Technology /Service	Estimated Cost (USD)	Justification
Frontend Hosting	$5 - $8/month	Cloud hosting services (e.g.: AWS or Azure): Provides infrastructure needed to host the frontend, ensuring scalability and reliability.
Backend Server	$6 - $10/month	Backend Hosting: Hosted on a cloud server (e.g.: AWS EC2) using Node.js with Express.js for API requests and business logic.
Decentralized Storage (IPFS Pinning Service)	$5 - $10/month	Decentralizes and secures vote data, enhancing availability and integrity.
Total Estimated Monthly Cost	**$15 - $20/month**	Represents the monthly cost for cloud infrastructure and decentralized storage.

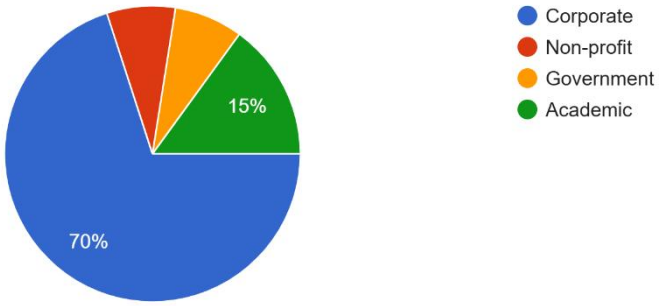
8.3 Survey

A survey was conducted to better understand the requirements and preferences of potential users across various organisational roles and types. The outcomes of this analysis are depicted below.



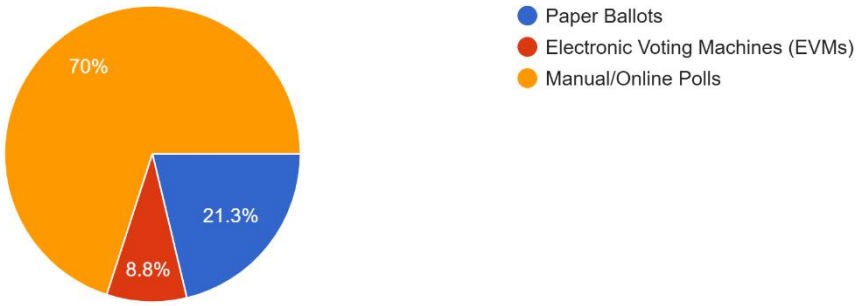
Organization Type

80 responses



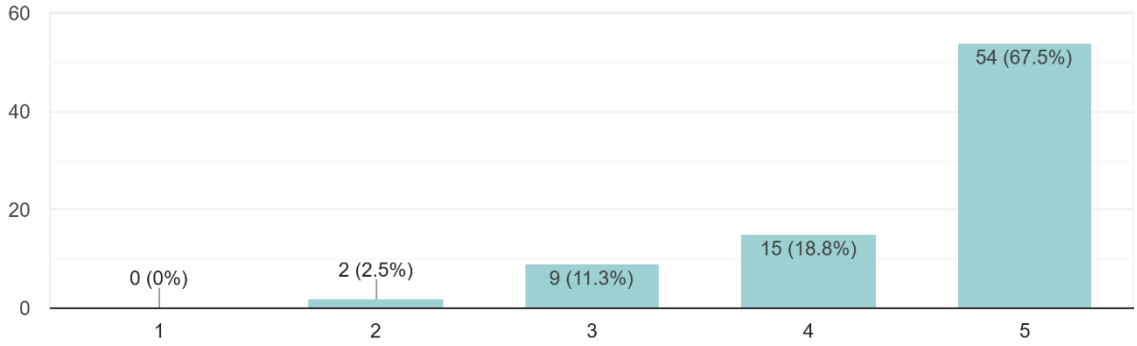
What current voting system do you use?

80 responses



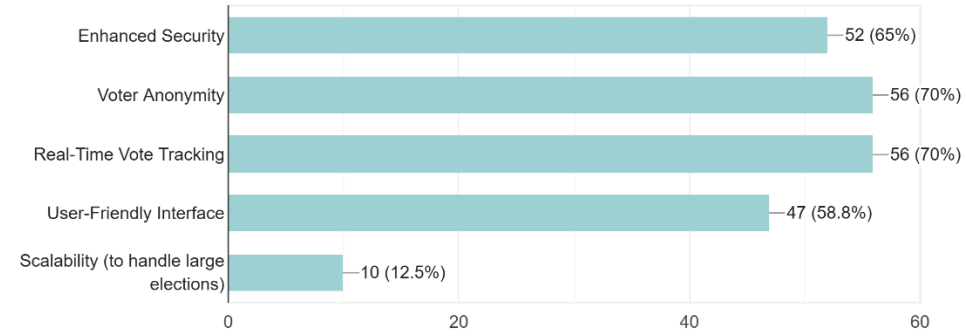
How important is it to ensure the security and tamper-proof nature of the voting process in your organization?

80 responses



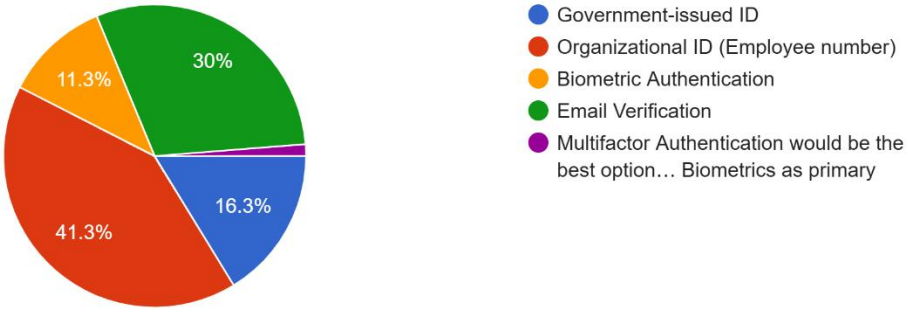
What features would you consider most important in a new voting system?

80 responses



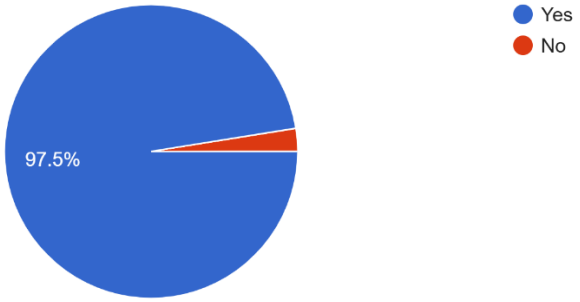
What is your preferred method for verifying voter identity?

80 responses



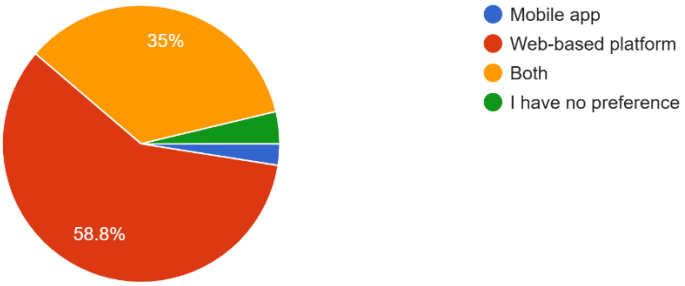
Do you think decentralized systems (blockchain) improve trust in the voting process?

79 responses



Would you like to have the option for a mobile app or just a web-based platform?

80 responses



Appendices

Project Keywords

- Blockchain Voting System
- Organizational Election Platform
- Data Integrity and Security
- Smart Contract Automation
- Voter Anonymity and Privacy
- Decentralization
- Transparency and Auditability
- Tamper-Resistance

High-level System Architecture Diagram

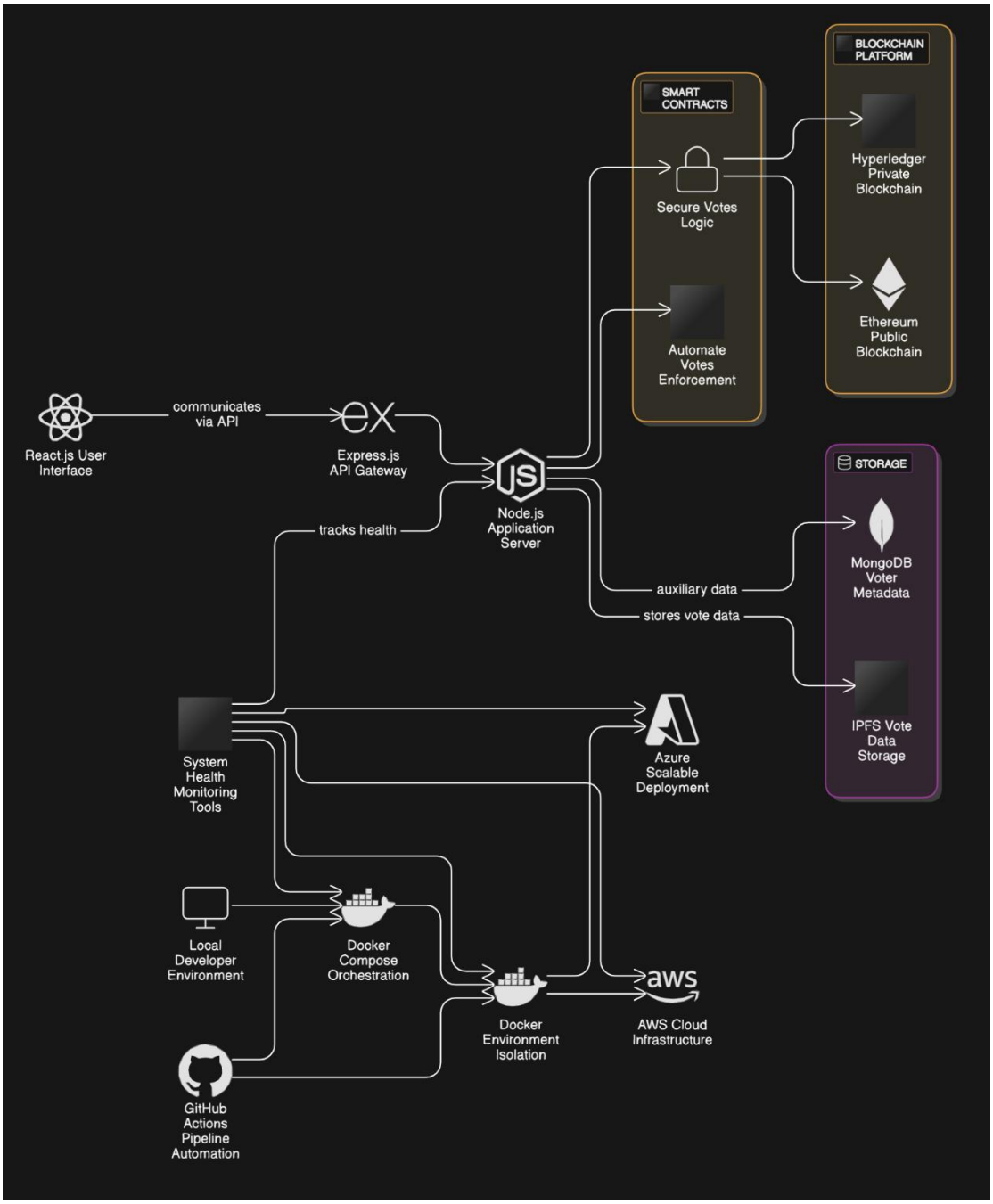


Figure 18: High Level System Architectural Diagram

References

Abdul, M. & Saleem, S., 2024. *Navigating Blockchain's Twin Challenges: Scalability and Regulatory Compliance.* [Online]
Available at: https://www.mdpi.com/2813-5288/2/3/13

Aidynov, T. & Goranin, N., 2024. *A Systematic Literature Review of Current Trends in Electronic Voting System Protection Using Modern Cryptography*. [Online]
Available at: https://www.mdpi.com/2076-3417/14/7/2742
[Accessed 29 October 2024].

Berenjestanaki, M. H., Barzegar, H. R., El Ioini, N. & Pahl, C., 2024. *Blockchain-Based E-Voting Systems: A Technology Review*. [Online]
Available at: https://www.mdpi.com/2079-9292/13/1/17
[Accessed 2024].

Berenjestanaki, M. H., R. Barzegar , H., El Ioini , N. & Pahl, C., 2023. *Electronics | Free Full-Text | Blockchain-Based E-Voting Systems: A Technology Review*. [Online]
Available at: https://www.mdpi.com/2079-9292/13/1/17/review_report
[Accessed 15 October 2024].

Bhasi, M., Chandrasekaran, K. & Pandey, A., 2019. *VoteChain: A Blockchain Based E-Voting System.* [Online]
Available at: https://ieeexplore.ieee.org/abstract/document/8978295
[Accessed 20 October 2024].

Chai, J. & Ohio State University, 2020. Blockchain based voting system with Ethereum Blockchain. *Research Thesis,* p. 25.

Cole, J., 2024. *Blockchain Development and Security: Best Practices - BlockApps Inc.*. [Online]
Available at: https://blockapps.net/blog/security-best-practices-in-blockchain-development/
[Accessed 25 October 2024].

Cole, J., 2024. *The Impact of Blockchain on Voting Systems and Governance - BlockApps Inc.*. [Online]
Available at: https://blockapps.net/blog/the-impact-of-blockchain-on-voting-systems-and-governance/

Collier, K., 2020. *Voatz smartphone voting app has significant security flaws, MIT researchers say*. [Online]
Available at: https://www.nbcnews.com/tech/security/voatz-smartphone-voting-app-has-significant-security-flaws-mit-researchers-n1136546

Collier, K. & NBC News, 2020. *Voatz smartphone voting app has significant security flaws, MIT researchers say*, s.l.: s.n.

Council, B., 2023. *Role of Blockchain in Cybersecurity.* [Online] Available at: https://www.blockchain-council.org/blockchain/blockchain-in-cybersecurity/ [Accessed 29 October 2024].

Daraghmi, E., Hamoudi, A. & Helou, M. A., 2024. *Decentralizing Democracy: Secure and Transparent E-Voting Systems with Blockchain Technology in the Context of Palestine.* [Online]
Available at: https://www.mdpi.com/1999-5903/16/11/388
[Accessed 29 October 2024].

Groopman, J. & Insights, K., 2023. *8 best practices for blockchain security.* [Online] Available at: https://www.techtarget.com/searchsecurity/tip/8-best-practices-for-blockchain-security

Hjalmarsson, F. P., Hreioarsson, G. K., Hamdaqa, M. & Hjalmtysson, G., 2018. *Blockchain-Based E-Voting System.* [Online]
Available at: https://ieeexplore.ieee.org/document/8457919/
[Accessed 07 October 2024].

Ohize, H. O. et al., 2024. Blockchain for securing electronic voting systems: a survey of architectures, trends, solutions, and challenges.

Pandey, A., Bhasi, M. & Chandrasekaran, K., 2019. *VoteChain: A Blockchain Based E-Voting System.* [Online]
Available at: https://ieeexplore.ieee.org/abstract/document/8978295
[Accessed 20 October 2024].

Pereira, B. M. B. et al., 2023. *Blockchain-Based Electronic Voting: A Secure and Transparent Solution.* [Online]
Available at: https://www.mdpi.com/2410-387X/7/2/27

Sharp, M., Njilla, L., Huang, C.-T. & Geng, T., 2024. *Blockchain-Based E-Voting Mechanisms: A Survey and a Proposal.* [Online]
Available at: https://www.mdpi.com/2673-8732/4/4/21
[Accessed 29 October 2024].

Tambanis, D., 2019. *Blockchain Applications: Election Voting - Blockchain Philanthropy Foundation - Medium.* [Online]
Available at: https://medium.com/bpfoundation/blockchain-applications-election-voting-a1436e7d10cb
[Accessed 20 October 2024].

Wadowski, G. M., Otte, L. S., Bernardo, N. D. & Macht, G. A., 2023. A Comparative Study of Electronic Voting and Paper Ballot. p. 21.

Appendix 3: Project Interim Document

PUSL3190 Computing Individual Project

Project Interim Report

Blockchain-Based Secured Voting System for Organizations

Supervisor: Mr. Gayan Perera

Name: Delwakkada Nemsara

Plymouth Index Number: 10898858

Degree Program: BSc (Hons) Computer Science

Table of Contents

Chapter 01 Introduction

1.1 Introduction

Voting is a fundamental process in decision making within organizations, ensuring fairness, transparency, and democratic participation. Traditional voting techniques, both paper-based and computerised, face numerous issues, including security vulnerabilities, inefficiencies, and a lack of transparency. Paper ballots require extensive resources for secure storage, counting, and transportation, making the process prone to logistical delays and human error (Pandey, et al., 2019). Meanwhile, electronic voting systems are often centralized, creating single points of failure that increase susceptibility to cyber-attacks and data manipulation (Pandey, et al., 2019).

Blockchain technology provides a transformative solution to these challenges by introducing decentralisation, immutability, and cryptographic security into voting systems. By leveraging smart contracts and cryptographic techniques, blockchain based voting systems can deliver a verifiable and tamper-proof electoral process. The immutability feature of blockchain ensures that once votes are cast, they cannot be altered or deleted, thereby securing the integrity of election outcomes (Nakamoto, 2008). Furthermore, blockchain enables real-time tracking of votes, improving transparency and voter confidence.

The goal of this project is to develop a blockchain-based voting system for organizational elections. By utilizing blockchain technology, the proposed system will enhance the election security, making the elections fraud-proof and improve the efficiency of the voting process. The focus of organizational elections allows for experimentation on a manageable scale avoiding the regulatory complexities associated with national elections while signifying blockchain's viability as a secure voting mechanism.

1.2 Problem Definition

Traditional voting methods, whether manual or electronic, are constrained by security risks, inefficiencies, and trust issues, reducing their reliability.

Key concerns include:

- **Election Fraud:** Paper ballots pose risks such as vote manipulation, misplacement, or miscounting, while electronic voting systems can be hacked and are subject to cyber-attacks (Pandey, et al., 2019).
- **Lack of Transparency:** Centralised voting systems cannot be verified in real time, raising doubts about election integrity and results.
- **High Costs and Logistical Challenges:** Traditional voting employs significant financial and human resources in printing ballots, distributing them, counting, and auditing.

- **Voter Accessibility and Anonymity:** Remote voters, such as expatriates and individuals with disabilities, often face difficulties in casting their votes. Additionally, traditional electronic voting systems may fail to ensure voter anonymity.

Blockchain technology presents an opportunity to overcome these challenges through the provision of a decentralized, tamper-proof, and verifiable voting platform. Blockchain-based voting will minimize fraud, enhance transparency, lower costs, and create a more accessible electoral process.

1.3Project Objectives

The project aims to design, develop, and implement a decentralised blockchain-based voting system to address significant challenges in organizational elections. The objectives below define the specific deliverables and measurable outcomes that this project intends to achieve.

1. **Develop a Secure Voting Mechanism**: Implement blockchain-based security protocols to ensure vote integrity and tamper-proof election results.
2. **Enhance Transparency and Auditability**: Utilize blockchain's public ledger feature to enable real-time vote verification while maintaining privacy.
3. **Ensure Voter Anonymity and Privacy**: Use cryptographic techniques such as zero-knowledge proofs to protect voter identities while maintaining vote authenticity.
4. **Facilitate Accessibility and Scalability**: Design a web-based voting platform to support various devices, allowing wider participation in organizational elections.
5. **Implement Smart Contracts for Automation**: Use Solidity-based smart contracts to manage voting processes, preventing human errors and vote tampering.
6. **Optimize Performance and Scalability**: Analyze system performance in handling large-scale elections, ensuring the blockchain framework is efficient and cost-effective.

This interim report outlines the design, feasibility, and technical considerations of the proposed blockchain-based voting system. It conducts a thorough study of the problem domain, systemrequirements, architectural design, and development approach, as well as identifying critical challenges and future implementation paths.

Scope of the Project

This project focuses on the development of a blockchain-based voting system tailored for organizational elections. Unlike national elections, which involve legal, political, and large-scale infrastructure considerations, this system is intended for use in private institutions, corporate environments, and academic settings. The project primarily focuses upon enhancing security, transparency, and accessibility in the process of voting, with a web-based platform serving as the primary voting interface.

While blockchain technology has significant advantages, this system is designed for organisational elections rather than large-scale governmental use. It does not address national election policies, large-scale governmental infrastructure, or integration with national electoral commissions. However, this project incorporates secure voter authentication mechanisms suitable for organizational-level elections, ensuring secure access and verification within its intended scope. In addition, considerations for scalability in larger deployments will be explored in future research.

Chapter 02 System Analysis

2.1 Facts Gathering Techniques

To ensure a comprehensive understanding of the requirements for the blockchain-based voting system, multiple fact gathering techniques were conducted, and these included surveys, observations, literature reviews, and an analysis of existing blockchain-based voting systems.

Survey

A survey was conducted using an online questionnaire to collect feedback from potential users across corporate, academic, non-profit, and government sectors. The survey evaluated key themes related to voter experience, election administration, and IT support needs. The key takeaways from the survey were:

- Voters (85%) prioritize a seamless and secure voting experience.
- Election administrators (7.5%) require robust election management tools, real-time tracking, and authentication systems.
- IT/Admin Support (7.5%) focuses on system maintenance, security, and troubleshooting.

The majority of the participants represented corporate organisations (70%), followed by academic institutions (15%) and non-profit and government entities (7.5% each). The findings emphasized the need for an intuitive user interface for voters and comprehensive administrative tools for election officials.

The survey questions and detailed response analytics are provided in the Appendix.

Observations

Observations were carried out to analyse both traditional and electronic voting systems to identify their strengths and weaknesses. The study assessed common vulnerabilities such as:

- **Security risks** in paper-based voting, including fraud, ballot tampering and coercion.
- **Technical failures** in electronic voting machines, like software glitches and hacking attempts.
- **Lack of transparency** in some electronic voting systems that do not provide audit trails.

This observation studied how blockchain technology could improve concerns such as security, transparency, and scalability to mitigate these issues.

Literature Review

As part of the fact gathering process, a comprehensive literature review was also conducted, which covered research on existing blockchain-based voting solutions. This review guided in identifying key challenges and advancements in the field. The main areas of research included:

- **Scalability Issues:** Many blockchain voting systems struggle to handle high transaction volumes during large scale elections (Abdul & Saleem, 2024)**.** Potential solutions like sharding and off-chain transactions could be applied to address this challenge.

- **Trust and Transparency:** The public's trust in blockchain voting varies, requiring real-time audit trails and voter verification mechanisms (Tambanis, 2019).

- **Decentralisation and Data Security:** Several systems still rely on centralised components, resulting in security vulnerabilities (Hajian Berenjestanaki *et al.*, 2024).

- **Data Privacy and Anonymity:** Privacy concerns persist, as many systems lack cryptographic techniques like zero-knowledge proofs to ensure voter anonymity (Aidynov *et al.*, 2024).

- **Real-Time Tracking and Auditability:** Many current blockchain voting solutions lack transparency tools for tracking votes in real time (Cole, 2024; Satybaldina et al., 2024)**.**

Review of Existing Blockchain-Based Voting Systems

Several blockchain voting systems were analysed, focusing on system architecture, security mechanisms, and usability. Key takeaways included the implementation of decentralized ledgers, cryptographic security, and real-time auditing capabilities.

- **Decentralised ledgers** to prevent tampering and improve security.
- **Cryptographic security** measures such as end-to-end encryption and hashing for vote integrity.
- **Real-time auditing capabilities** to allow election monitoring and enhance trust in the system.

An analysis of the case studies of blockchain voting platforms like Voatz and Follow My Vote demonstrated both potential benefits and challenges. While secure and transparent voting mechanisms have been successfully implemented by the platforms, they also faced issues like scalability bottlenecks, privacy concerns, and regulatory restrictions.

2.2 Existing System

Traditional voting systems, including paper ballots and electronic voting machines are affected by several drawbacks, such as inefficiency, lack of transparency, and security vulnerabilities. The literature review demonstrated that while electronic voting systems enhance accessibility, they remain susceptible to cyber threats and centralized control, raising concerns about manipulation and trust.

To explore potential solutions, existing blockchain voting systems analysed. Many blockchain voting projects highlight security and transparency but continue to face challenges such as scalability, voter anonymity, and regulatory acceptance. While solutions like Voatz have demonstrated success in pilot projects, large-scale electoral implementation would become a challenge because of performance bottlenecks and privacy concerns (Abdul & Saleem, 2024).

2.3 Use Case Diagram

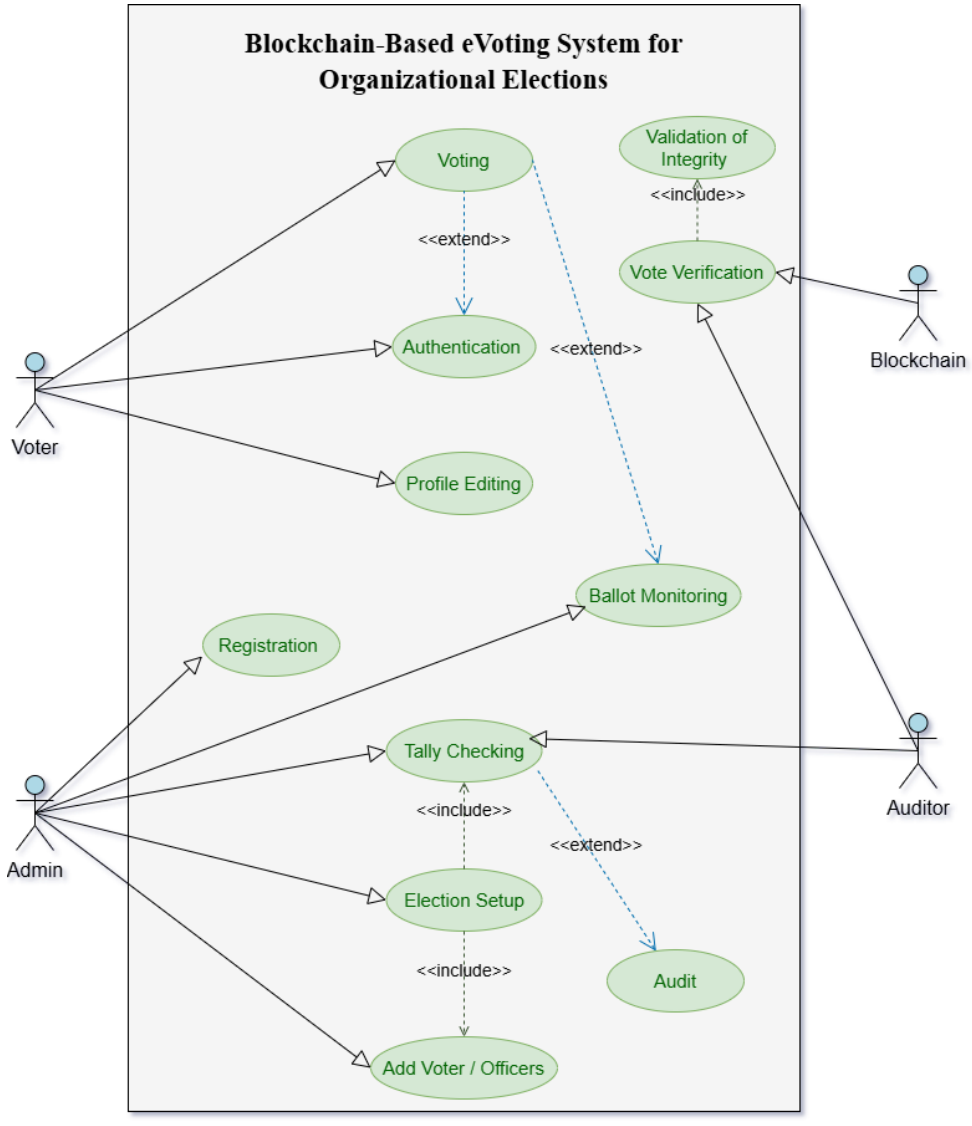


Figure 19: Use Case Diagram

2.3 Drawbacks of the Existing System

Despite advancements in electronic and blockchain-based voting systems, there are several challenges that hinder their widespread adoption and effectiveness. Traditional voting techniques, including paper ballots and electronic machines, continue to face significant concerns of security, transparency, and scalability. While blockchain technology delivers effective solutions, its adoption in voting systems is still limited by privacy concerns, legal issues, and user adoption barriers. The following section discusses the major drawbacks of existing voting systems.

1. **Security Vulnerabilities**

Traditional electronic voting systems are vulnerable to cyber threats including hacking, data breaches, and data tampering. Centralised databases make them high value targets for such attacks (Hajian Berenjestanaki *et al.*, 2024).

2. **Lack of Transparency**

Many current systems lack mechanisms for real-time vote tracking and auditability. These results in a lack of public trust in election outcomes (Cole, 2024; Satybaldina et al., 2024).

3. **Scalability Challenges**

Blockchain voting solutions, while promising, often struggle to scale effectively. During peak periods of voting, high transaction volumes may cause system slowdowns or failures. To address this, strategies such as sharding or off-chain transactions could be integrated to improve system performance.

4. **Privacy Concerns**

Existing systems fail to fully implement cryptographic techniques like zero-knowledge proof to protect voter anonymity while ensuring vote integrity (Aidynov & Goranin, 2024).

5. **Regulatory and Legal Uncertainty**

The legal context for blockchain-based voting remains unclear, resulting in an uncertainty in adoption. Many jurisdictions have yet to build regulatory frameworks that adhere to democraticprinciples (Abdul & Saleem, 2024; Tambanis, 2019).

6. **Educational Barriers**

There is a significant knowledge gap among stakeholders, including voters, election officials, and policymakers, about blockchain technology and its application in voting systems. Without adequate education and awareness initiatives, many individuals are hesitant to engage in blockchain-based voting, resulting in low adoption rates and slower implementation of these systems (Tambanis, 2019).

Chapter 03 Requirements Specification

3.1 Functional Requirements

The blockchain-based voting system must meet key functional requirements to ensure secure, transparent, and reliable voting processes. The key functional requirements are outlined below:

Requirement Category	Description
Secure and Transparent Voting Process	• Votes must be immutably stored on the blockchain to prevent unauthorized modifications. • Smart contracts must validate votes and enforce election rules autonomously.
Voter Authentication and Authorization	• Users must register with verified credentials before voting. • Identity verification should integrate blockchain wallets (Ex: MetaMask) to ensure legitimacy.
Real-Time Vote Confirmation	• Voters should receive instant confirmation of their vote submission. • A blockchain transaction hash must be provided as proof of a successful vote.
Decentralised Storage and Accessibility	• The system must store voter data securely using IPFS (InterPlanetary File System). • MongoDB should be used for auxiliary data such as election metadata and voter registration details.
Election Management	• Administrators should be able to create and configure elections. • The system must support multiple candidates and custom voting periods.
Audit and Transparency Mechanisms	• Real-time audit trails should be provided via a user-friendly interface. • All transactions must be publicly viewable on the blockchain while maintaining voter anonymity.
Role-Based Access Control	• The system must provide different access levels for voters, election administrators, and auditors. • Smart contracts should restrict unauthorized activities.

3.2 Non-Functional Requirements

Requirement Category	Description
Performance and Scalability	• The system must handle high traffic while maintaining low latency. • If using Ethereum, sharding and Layer-2 scaling solutions should be considered.
Security and Privacy	• Cryptographic hashing (SHA-256) must be utilised to secure sensitive voter information. • Smart contracts should be inspected to prevent vulnerabilities such as reentrancy attacks. • Voter anonymity should be protected using techniques like Zero-Knowledge Proofs (ZKPs).
Reliability and Availability	• The system should be deployed on AWS/Azure for high uptime or cost-effective alternatives like Netlify/Render. • Blockchain nodes must be replicated in multiple geographic zones for redundancy.
Data Integrity and Auditability	• Cryptographic signatures must be employed to prevent vote duplication and tampering. • Election results should be verifiable through public blockchain explorers.
Usability and Accessibility	• The frontend (built with React.js) must be mobile-responsive and user-friendly across multiple devices.

3.3 Hardware / Software Requirements

3.3.1 Hardware Requirements

Component	**Minimum Requirement**
Processor	2 GHz or higher
RAM	4 GB or more
Disk Space	100 GB or more
GPU (Optional)	Required for ZKPs or AI-based security features

3.3.2 Software Requirements

Software/Tool	**Version**
Node.js	18.14.0
Web3.js	1.8.2
Truffle	5.7.6
Solidity	0.5.16
Ganache	7.7.3
MetaMask	Latest
Python	3.9
FastAPI	Latest
MongoDB	Latest
Ethereum	Latest

3.4 Networking Requirements

As a decentralized web application (dApp), the blockchain-based voting system requires specific networking configurations to ensure secure and efficient interactions between system components.

Requirement Category	**Description**
Ethereum Node Connectivity	• The system must connect to the Ethereum testnet/mainnet for deploying smart contracts and processing transactions.
Cloud-Based Deployment	• Backend services should be hosted on cloud platforms like AWS EC2, Azure VMs, or DigitalOcean Droplets to ensure availability and scalability.
Security and Access Control	• **Reverse Proxies:** Use Nginx or HAProxy to control and distribute incoming requests securely. • **Firewall Rules:** Restrict unauthorized access and permit only trusted communication. • **Virtual Private Cloud (VPC):** Isolate blockchain nodes and backend servers from direct public exposure to enhance security.

IPFS Gateways	• Decentralized storage should be accessible via public or private IPFS gateways for fast and secure data retrieval.

By implementing these networking configurations, the system ensures secure interactions between voters, blockchain nodes, and backend services while maintaining high availability.

Chapter 04 Feasibility Study

4.1 Operational Feasibility

Stakeholder Readiness

Implementing a blockchain-based voting system involves active engagement and acceptance from multiple stakeholders, including election officials, administrators, and voters. The readiness of these entities depends on:

- **Election Authorities:** Their readiness to adopt digital voting solutions while maintaining legal and regulatory compliance.
- **Voters:** Awareness and familiarity with blockchain technology and digital voting systems.
- **Administrator:** Training and adaptability to manage and secure blockchain-based elections.

User Experience and Accessibility:

To enable widespread adoption, the system should have an intuitive and user-friendly interface. This includes:

- **Responsive UI:** A React.js-based interface that adapts to different screen sizes and devices.
- **Multilingual Support:** Ensures accessibility for diverse user groups.
- **Training and Support:** Tutorials and guidance to help users securely cast their ballots.

Legal & Regulatory Considerations

The legal framework governing blockchain-based elections is an important consideration for practical feasibility. Factors include:

- **Data Protection Laws:** Compliance with GDPR and other relevant regulations.
- **Election Regulations:** Complying with national and international electoral laws.
- **Identity Verification:** Ensuring a legal and secure method of voter authentication.

Scalability of Operations

The system must accommodate elections of varying sizes, from corporate decision-making to national elections. Scalability considerations include:

- Performance during peak voting periods.
- Handling an increasing number of registered voters.
- Ensure minimal downtime and seamless user experience.

4.2 Technical Feasibility

Technology Stack and Suitability

The project employs a combination of cutting-edge technologies to ensure security, transparency, and scalability:

- **Blockchain Platform:** Ethereum for decentralized voting or Hyperledger for a permissioned blockchain setup.
- **Smart Contracts:** Written in Solidity to ensure tamper-proof voting mechanisms.
- **Frontend:** React.js for a dynamic and responsive voting interface.
- **Backend:** Node.js and Express.js to handle API requests and logic.
- **Storage:** IPFS for decentralized vote storage, and MongoDB for auxiliary metadata.
- **Hosting:** AWS or Azure for scalable cloud deployment.
- **CI/CD & Testing:** GitHub Actions for automated testing and deployment.
- **Containerization:** Docker and Docker Compose for consistent application deployment.

Consensus Mechanism and Justification

The blockchain platform implemented in this project involves a consensus mechanism to validate transactions and ensure the security and integrity of the voting process. The system uses [Proof of Stake (PoS)] / [Proof of Authority (PoA)] as its consensus mechanism for the following reasons:

- **Energy Efficiency:** Unlike Proof of Work (PoW), which requires substantial computational power, PoS/PoA consumes considerably less energy.

- **Scalability:** PoS/PoA offers faster transaction confirmation times, which is critical for processing an excessive number of votes in a limited voting window.
- **Security and Trust:** By relying on validators rather than miners, PoS/PoA minimises risks like 51% attacks while increasing network trustworthiness.

For this project, Ethereum (PoS) / Hyperledger (PoA) is utilized as the blockchain platform, ensuring decentralized yet efficient voting processes.

System Architecture & Infrastructure

The system follows a decentralised, layered architecture:

1. **Frontend UI:** Voters interact via a React.js web application.
2. **Backend API:** Node.js handles business logic and API requests.
3. **Blockchain Layer:** Securely records votes using Ethereum or Hyperledger.
4. **Storage Layer:** IPFS stores encrypted vote data; MongoDB stores non-sensitive metadata.
5. **Cloud Hosting:** AWS/Azure ensures scalability and availability.
6. **CI/CD Pipeline:** Automated testing and deployment streamline development.

Performance and Scalability Considerations

The system must handle a high volume of transactions efficiently. Key considerations include:

- **Optimised Smart Contracts:** Minimizing gas fees on Ethereum.
- **Layer-2 Scaling Solutions:** Use zk-Rollups or Optimistic Rollups to increase transaction throughput.
- **Load Testing:** Ensures that the system can support a large number of concurrent voters.

Integration with Existing Systems

The system needs to be integrated seamlessly with existing electoral databases and identity verification systems. Considerations include:

- APIs for voter authentication and data synchronization.
- Interoperability with existing election management software.

4.3 Outline Budget

The estimated budget covers development, hosting, security, and maintenance. It may vary based on actual implementation and hosting choices.

Tool/Technology /Service	**Estimated Cost (USD)**	**Justification**
Frontend Hosting	$5 - $8/month	Cloud hosting services (AWS/Azure) for frontend scalability.
Backend Server	$6 - $10/month	Cloud-hosted backend on AWS EC2 or Azure VM.
Decentralized Storage	$5 - $10/month	IPFS pinning service for secure, tamper-proof vote storage.
Blockchain Deployment	TBD	Ethereum gas fees (testnet/mainnet), smart contract audits.
Total Estimated Monthly Cost	**$15 - $20/month**	Represents the monthly cost for cloud infrastructure and decentralized storage.

Chapter 05 System Architecture

5.1 Class Diagram of Proposed System

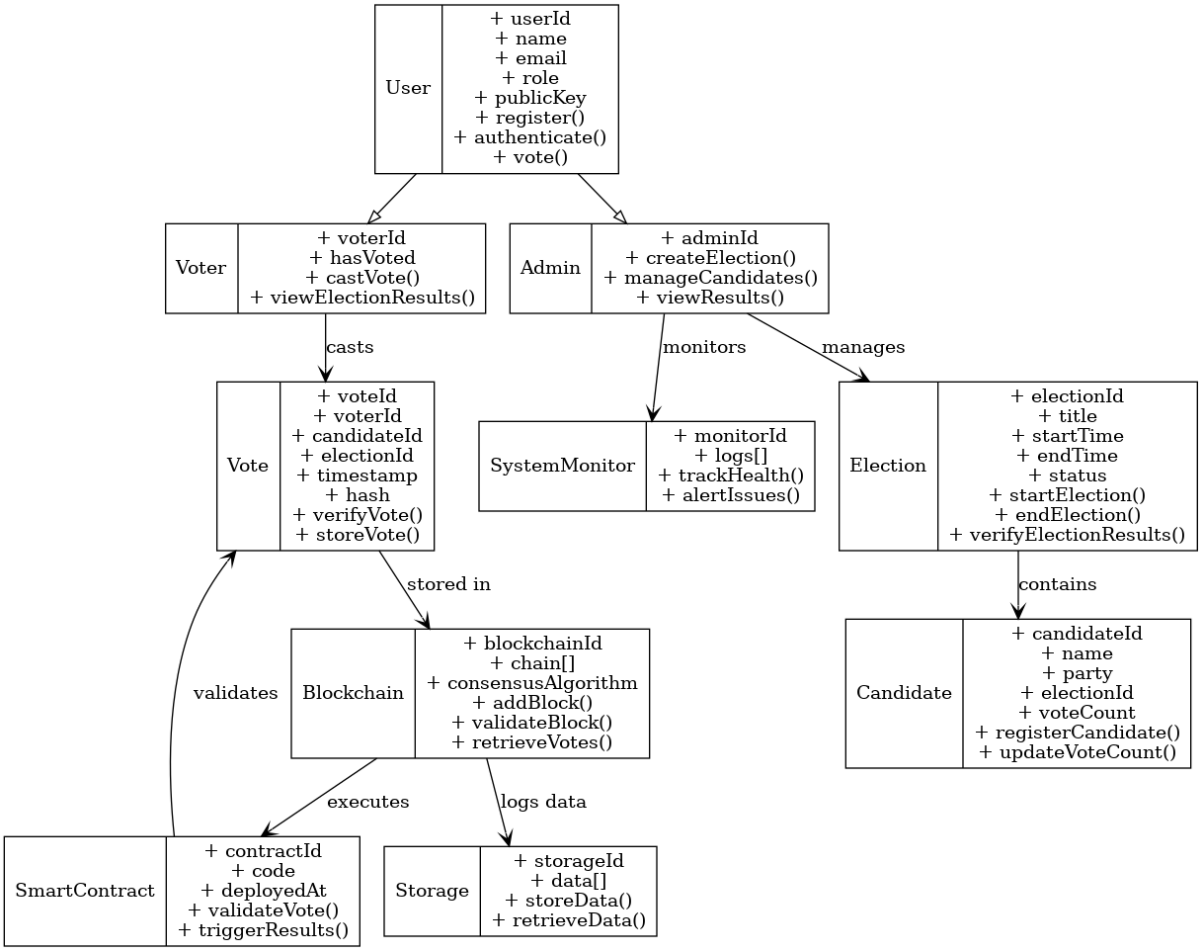


Figure 20: Class Diagram

5.2 ER Diagram

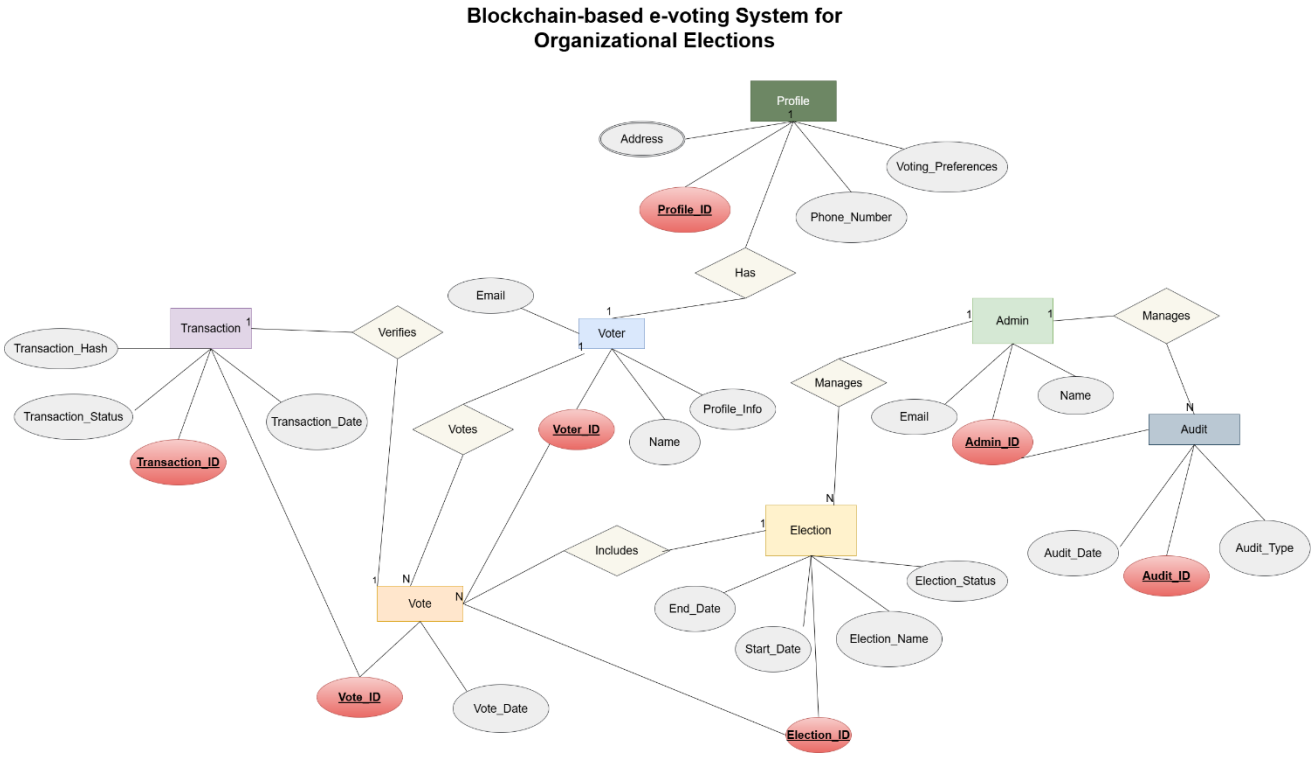


Figure 21: ER Diagram

5.3 High-level Architectural Diagram

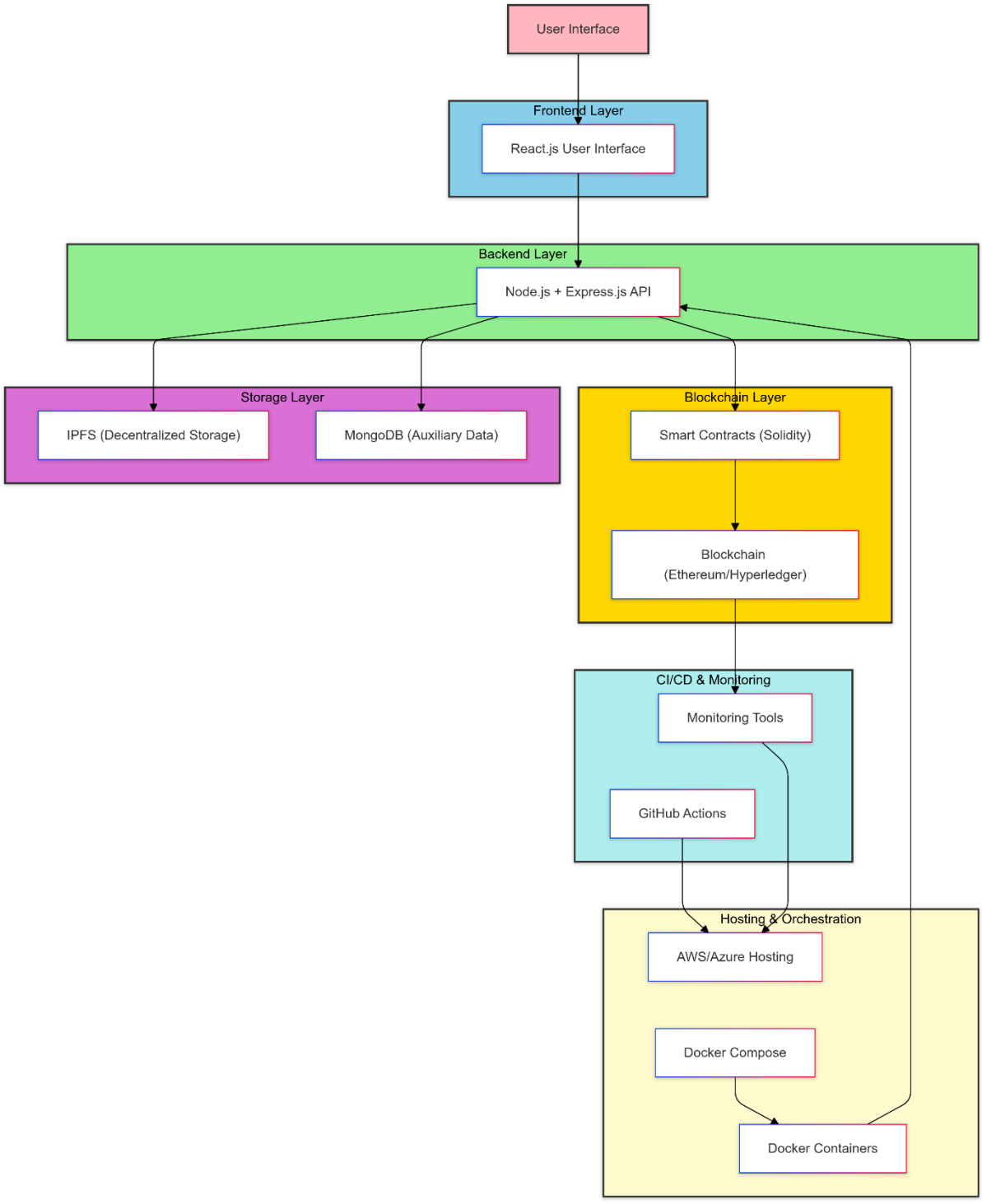


Figure 22: High-level Architectural Diagram

5.4 Networking Diagram

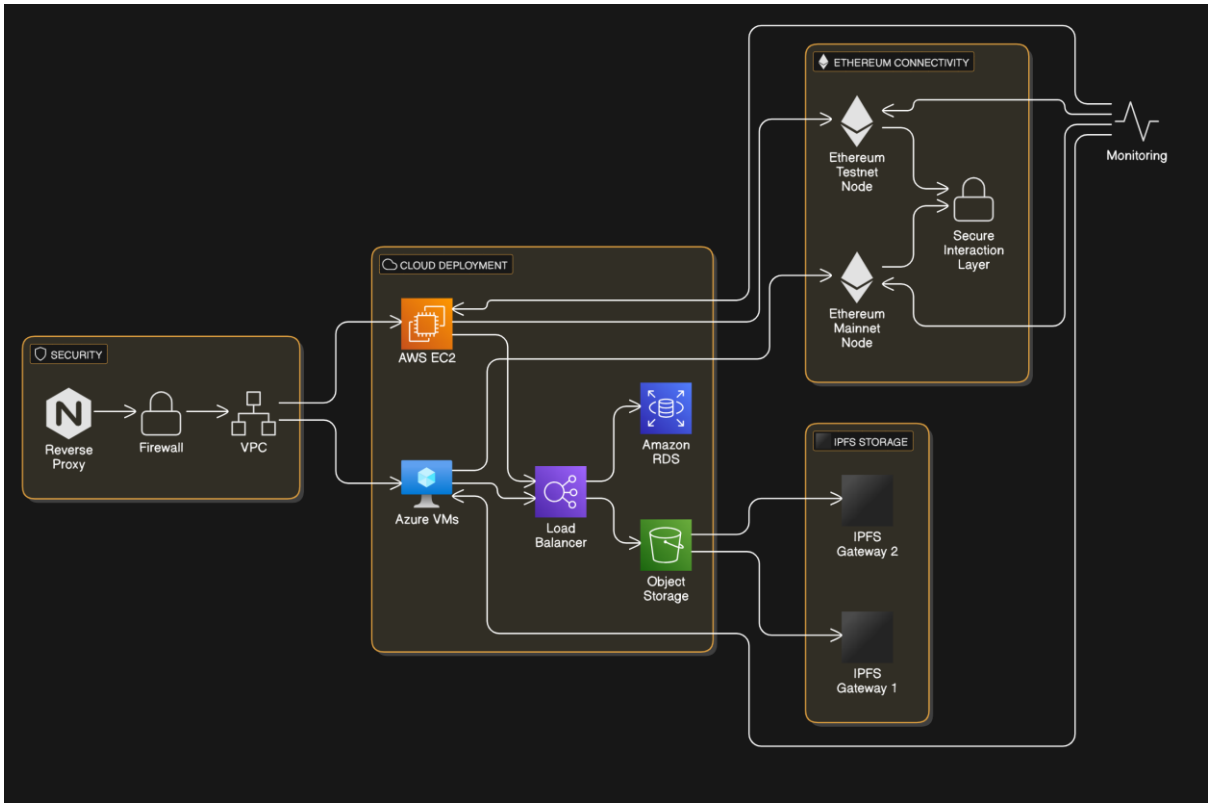


Figure 23: Networking Diagram

Chapter 06 Development Tools and Technologies

6.1 Development Methodology

The development of the blockchain-based voting system will follow the Agile methodology, prioritizing flexibility, iterative enhancements, and continuous stakeholder engagement. Progressive projects with complex, evolving requirements, like this one, where the exact scope and features may need to adapt throughout the development cycle, are ideally suited for the agile software development methodology. This approach enables rapid prototyping, regular feedback, and incremental delivery of working solutions.

Agile Methodology in Detail:

The development phase will be mainly divided into sprints which are small, time-boxed periods, typically lasting between 1-2 weeks. Each sprint will deliver a working model of the system, whereas it can be reviewed and refined based on feedback. The primary principles that influence this methodology include:

1. **Flexibility and Adaptation:** Agile enables the project scope to evolve in accordance with stakeholder feedback and adjustments in technical, security, or legal requirements. This adaptability is crucial for a blockchain-based electronic voting system as technology and the legal frameworks surrounding it are continuously evolving.

2. **Frequent Communication:** Collaboration is an important factor in the Agile approach. All team members including developers, stakeholders, and users are kept informed and in sync throughout the development process by conducting regular meetings like standups and sprint reviews. Incorporating end-user input into the system's design, particularly with regard to usability and security issues, depends on this communication.

3. **Incremental and Deliverable Outcomes:** The Agile methodology allows the team to demonstrate measurable outcomes at the conclusion of each sprint by splitting the system into smaller, more manageable features. This ensures that early in the development cycle, working versions of the e-voting system are available for continuous enhancement.

In addition to providing adaptability to address emerging challenges in blockchain technology, voter authentication, and system security, the Agile methodology will ensure that both functional and non-functional requirements of the blockchain-based voting system are met.

6.2 Programming Languages and Tools

The system will be developed using a range of cutting-edge programming languages, frameworks, and tools to incorporate the functional and non-functional requirements of a secure, decentralized, scalable e-voting platform.

Blockchain Platform

The choice of blockchain platform is crucial for assuring transparency, immutability, and scalability. Two blockchain platforms, Ethereum and Hyperledger, will be examined based on their different benefits:

- **Ethereum (PoS):** Ethereum, with its Proof of Stake (PoS) consensus mechanism, is chosen because of its ability to support smart contracts, which are necessary for automating the voting process and ensuring vote security and immutability. Ethereum is an ideal platform for public blockchain applications due to its decentralised nature and robust smart contract capabilities, which utilise Solidity for contract development.

- **Hyperledger (PoA):** For permissioned blockchain applications, Hyperledger provides enhanced network control, making it a perfect option for enterprise-level solutions. Proof of Authority (PoA) provides rapid transaction completion, which is ideal for high-throughput systems like voting platforms.

Smart Contracts

Solidity will be used to write smart contracts for the Ethereum platform. These contracts will automate the voting process, assuring vote validity, election regulations, and immutable records. Smart contracts are crucial for preventing tampering with voting results, ensuring transparency and trust in the voting process.

Front-end Development

The front end of the system will be built using React.js, a popular JavaScript framework for building interactive, dynamic user interfaces. React.js was chosen for its:

- Component-based architecture, which enables reusable and maintainable UI components.
  Fast rendering, leveraging virtual DOM to enhance application performance.

- Responsive design capabilities ensure that the application is accessible across various devices, including smartphones, tablets, and desktops.

- React.js, along with Redux for state management, ensures that the frontend can efficiently handle real-time interactions like vote submission, confirmation, and user feedback.

Backend Development

The backend will be built with Node.js and Express.js. Node.js is a JavaScript runtime built on Chrome's V8 engine that is ideal for building scalable and performant backend services. Express.js, a Node.js-based lightweight web framework, will be used to construct RESTful API. These APIs will enable communication between the frontend, blockchain, and storage layers. The key roles of the backend include:

- **Voter Authentication:** It involves integrating with blockchain wallets (such as MetaMask) to ensure safe user authentication.
- **Vote Management:** involves ensuring that votes are correctly validated and recorded on the blockchain using smart contracts.
- **Data Retrieval and Processing:** Fetching auxiliary data (e.g., election results, user profiles) from MongoDB or IPFS.

Storage

To support decentralised vote storage of data, IPFS (InterPlanetary File System) will be used. IPFS is a distributed storage mechanism that guarantees the security and immutability of vote data, which will be stored in an encrypted form. MongoDB, a NoSQL database, will be used to store auxiliary data such as election metadata, voter information, and other non-sensitive data. MongoDB's flexible schema allows it to handle a variety of data structures that might arise during election configuration.

Hosting and Development

To ensure the application's scalability and security, cloud hosting will be done using AWS (Amazon Web Services) or Microsoft Azure. AWS provides a package of services including EC2 for virtual machines, S3 for file storage, and RDS for database management, which will be needed to run blockchain nodes and support backend services. Deployment pipelines will be automated with GitHub Actions, making continuous integration and continuous deployment (CI/CD) more efficient.

Containerization and Orchestration

For seamless portability across environments, Docker is used for containerisation, allowing the application to run consistently on any machine. Docker Compose will be used to manage multi-container environments, ensuring that the application scales effectively during peak traffic periods such as elections times.

6.3 Third-Party Components and Libraries

To streamline the development process and leverage external solutions, several third-party libraries and services are integrated into the system.

- **Web3.js:** This JavaScript framework will enable communication between the frontend and the Ethereum blockchain. It enables the frontend to interact with smart contracts and retrieve blockchain data.

- **MetaMask:** MetaMask will serve as the cryptocurrency wallet for user authentication. Voters will authenticate with their MetaMask wallet, which is connected with the Ethereum network to provide secure and seamless voting.

- **Truffle Suit:** Truffle Suite will be used to create, test, and deploy smart contracts. Truffle offers a suite of tools for developing and managing smart contracts on the Ethereum network.

- **Ganache:** A local Ethereum blockchain used to test smart contracts during development before they are deployed to the testnet or mainnet.

- **IPFS.js:** This JavaScript library will use IPFS to store and retrieve encrypted vote data.

6.4 Algorithms

The key algorithms for this project will focus on security, scalability, and vote validation:

- **Zero-Knowledge Proofs (ZKPs):** ZKPs will be used to improve voter anonymity by ensuring that voter identities are never exposed on the blockchain. This cryptographic method allows voters to prove the validity of their vote without revealing any sensitive information.

- **Hashing Algorithms (SHA-256):** To protect voter data and user information, SHA-256 hashing will be employed. This ensures that all sensitive data, including voter credentials, are kept secure.

- **Merkle Trees:** It will be used to store hashes of voting data in order to provide efficient and secure verification. This structure enables efficient verification of large data sets without the need to access the entire data set.

- **Sharding and Layer-2 Scaling:** To address scalability, sharding (in Ethereum) and Layer-2 scaling techniques like zk-Rollups and optimistic rollups will be used. These solutions enable the system to manage high numbers of transactions while maintaining speed and security.

Chapter 07 Discussion

7.1 Overview of the Interim Report

The interim report provides a comprehensive evaluation of the progress made on the blockchain-based voting system designed for organizational elections. It documents key tasks completed, assesses the outcomes, and discusses any deviations from the initial plan. Additionally, the report highlights the challenges encountered, risk management strategies implemented, and upcoming milestones necessary to complete the project within the revised timeline.

7.2 Summary of the Report

Since the initiation of the project, multiple key milestones have been established. The initial stages focused on literature review, requirement gathering, and system design. To identify critical improvements, a thorough analysis of existing voting systems was conducted. A high-level system architecture, which includes UI mockups, database schemas, and smart contract structures, has been developed. While full-scale implementation is yet to progress significantly, extensive research on blockchain integration, security mechanisms, and decentralisation principles has been conducted. Additionally, updates to the risk assessment and timeline have been incorporated to align with project realities.

7.3 Challenges Faced

The challenges that have affected the project's progress thus far are outlined below.

- **Technical Complexities:** Integrating blockchain technology, ensuring secure data handling, and achieving system scalability requires comprehensive research and testing. Additionally, implementing smart contracts with robust security measures to prevent attacks (e.g. reentrancy, overflow) has been a major challenge.
- **Resource Constraints:** It been difficult to set up the necessary blockchain environment and identifying and configuring compatible hosting solutions has been challenging.
- **Development Delays:** Unexpected challenges in implementing smart contracts caused changes to the initial plan, which led to a revised development timeframe.
- **Knowledge Gaps:** Implementing Zero Knowledge Proofs (ZKPs) for voter privacy is a complex process, requiring deep cryptographic knowledge.

7.4 Tasks Undertaken & Outcomes

- **Requirement Gathering and Analysis:** Completed with an emphasis on security, decentralisation, and requirements specific to elections.
- **Design Phase:** UI mockups, system architecture, and database schema are developed.
- **Development:** Initial front-end development commenced, validating UI feasibility before deeper implementation.
- **Research:** Extensive study of blockchain security, scalability, and smart contract vulnerabilities.

7.5 Future Plans / Upcoming Work

To ensure successful project completion, the following steps are prioritised:

1. **Implementation Acceleration:** Progress in blockchain layer, front-end, and backend with a focus on core voting functionalities.
2. **Testing and Validation:** Conduct unit, integration (frontend – backend – blockchain), and security testing.
3. **Deployment and Hosting:** Exploring cloud-based deployment solutions for efficiency.

References

Abdul, M. & Saleem, S., 2024. *Navigating Blockchain's Twin Challenges: Scalability and Regulatory Compliance.* [Online]
Available at: https://www.mdpi.com/2813-5288/2/3/13

Aidynov, T. & Goranin, N., 2024. *A Systematic Literature Review of Current Trends in Electronic Voting System Protection Using Modern Cryptography*. [Online]
Available at: https://www.mdpi.com/2076-3417/14/7/2742
[Accessed 29 October 2024].

Berenjestanaki, M. H., Barzegar, H. R., El Ioini, N. & Pahl, C., 2024. *Blockchain-Based E-Voting Systems: A Technology Review*. [Online]
Available at: https://www.mdpi.com/2079-9292/13/1/17
[Accessed 2024].

Berenjestanaki, M. H., R. Barzegar , H., El Ioini , N. & Pahl, C., 2023. *Electronics | Free Full-Text | Blockchain-Based E-Voting Systems: A Technology Review*. [Online]
Available at: https://www.mdpi.com/2079-9292/13/1/17/review_report
[Accessed 15 October 2024].

Bhasi, M., Chandrasekaran, K. & Pandey, A., 2019. *VoteChain: A Blockchain Based E-Voting System.* [Online]
Available at: https://ieeexplore.ieee.org/abstract/document/8978295
[Accessed 20 October 2024].

Chai, J. & Ohio State University, 2020. Blockchain based voting system with Ethereum Blockchain. *Research Thesis,* p. 25.

Cole, J., 2024. *Blockchain Development and Security: Best Practices - BlockApps Inc.*. [Online]
Available at: https://blockapps.net/blog/security-best-practices-in-blockchain-development/
[Accessed 25 October 2024].

Cole, J., 2024. *The Impact of Blockchain on Voting Systems and Governance - BlockApps Inc.*. [Online]
Available at: https://blockapps.net/blog/the-impact-of-blockchain-on-voting-systems-and-governance/

Collier, K., 2020. *Voatz smartphone voting app has significant security flaws, MIT researchers say*. [Online]
Available at: https://www.nbcnews.com/tech/security/voatz-smartphone-voting-app-has-significant-security-flaws-mit-researchers-n1136546

Collier, K. & NBC News, 2020. *Voatz smartphone voting app has significant security flaws, MIT researchers say,* s.l.: s.n.

Council, B., 2023. *Role of Blockchain in Cybersecurity.* [Online] Available at: https://www.blockchain-council.org/blockchain/blockchain-in-cybersecurity/ [Accessed 29 October 2024].

Daraghmi, E., Hamoudi, A. & Helou, M. A., 2024. *Decentralizing Democracy: Secure and Transparent E-Voting Systems with Blockchain Technology in the Context of Palestine.* [Online]
Available at: https://www.mdpi.com/1999-5903/16/11/388
[Accessed 29 October 2024].

Groopman, J. & Insights, K., 2023. *8 best practices for blockchain security.* [Online] Available at: https://www.techtarget.com/searchsecurity/tip/8-best-practices-for-blockchain-security

Hjalmarsson, F. P., Hreioarsson, G. K., Hamdaqa, M. & Hjalmtysson, G., 2018. *Blockchain-Based E-Voting System.* [Online]
Available at: https://ieeexplore.ieee.org/document/8457919/
[Accessed 07 October 2024].

Ohize, H. O. et al., 2024. Blockchain for securing electronic voting systems: a survey of architectures, trends, solutions, and challenges.

Pandey, A., Bhasi, M. & Chandrasekaran, K., 2019. *VoteChain: A Blockchain Based E-Voting System.* [Online]
Available at: https://ieeexplore.ieee.org/abstract/document/8978295
[Accessed 20 October 2024].

Pereira, B. M. B. et al., 2023. *Blockchain-Based Electronic Voting: A Secure and Transparent Solution.* [Online]
Available at: https://www.mdpi.com/2410-387X/7/2/27

Sharp, M., Njilla, L., Huang, C.-T. & Geng, T., 2024. *Blockchain-Based E-Voting Mechanisms: A Survey and a Proposal.* [Online]
Available at: https://www.mdpi.com/2673-8732/4/4/21
[Accessed 29 October 2024].

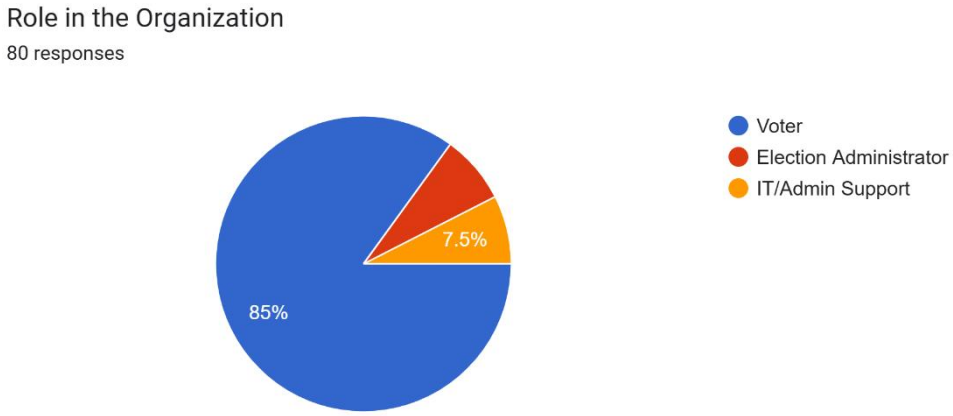
Tambanis, D., 2019. *Blockchain Applications: Election Voting - Blockchain Philanthropy Foundation - Medium.* [Online]
Available at: https://medium.com/bpfoundation/blockchain-applications-election-voting-a1436e7d10cb
[Accessed 20 October 2024].

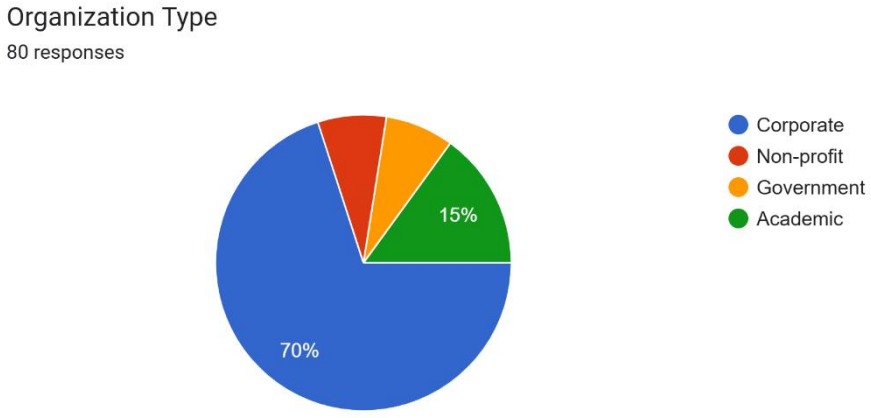
Wadowski, G. M., Otte, L. S., Bernardo, N. D. & Macht, G. A., 2023. A Comparative Study of Electronic Voting and Paper Ballot. p. 21.

Appendix A: Survey Results and Analysis

This appendix contains the results of an online survey conducted to gather feedback from potential users of the blockchain-based voting system. The survey was designed to evaluate key themes related to voter experience, election administration, and IT support needs.

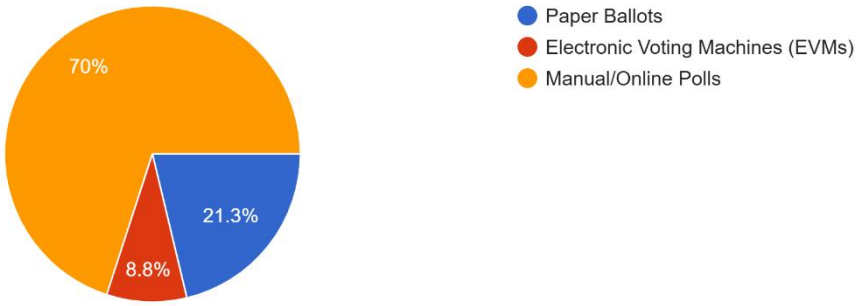
Below are detailed survey results and analytics for each question.





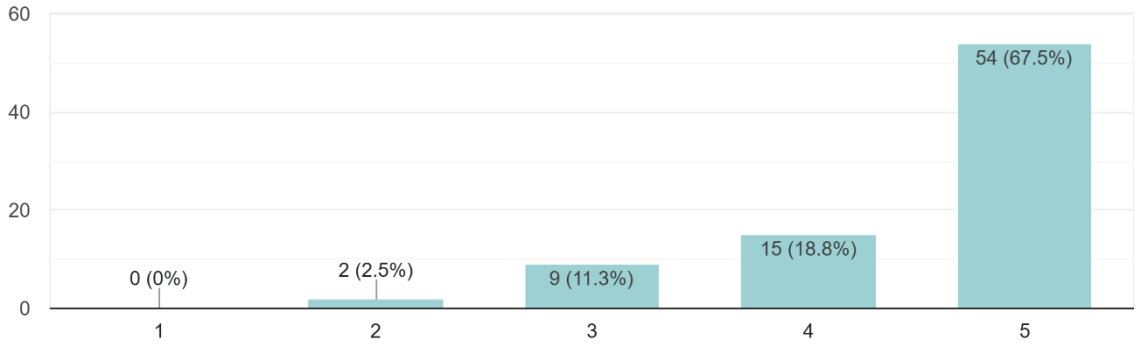
What current voting system do you use?

80 responses



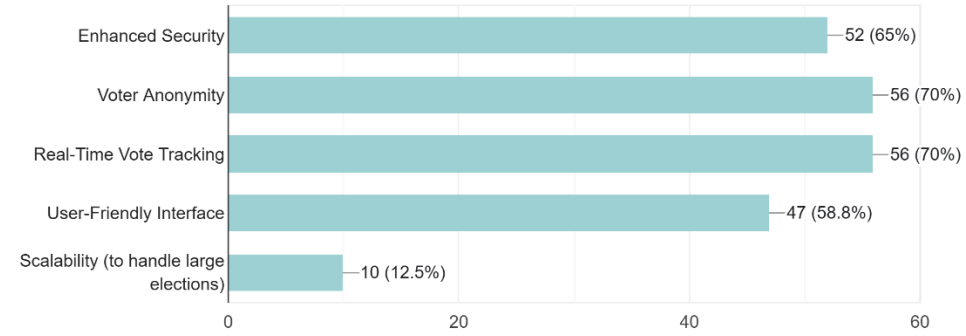
How important is it to ensure the security and tamper-proof nature of the voting process in your organization?

80 responses



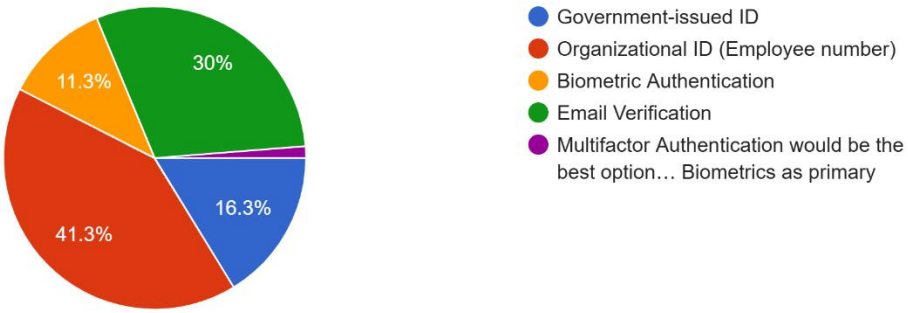
What features would you consider most important in a new voting system?

80 responses



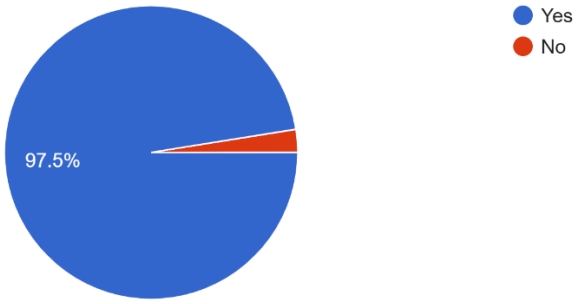
What is your preferred method for verifying voter identity?

80 responses



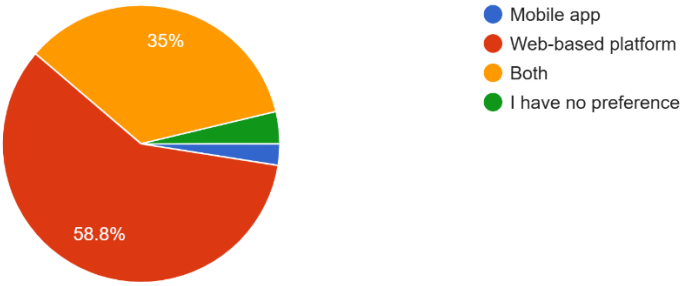
Do you think decentralized systems (blockchain) improve trust in the voting process?

79 responses



Would you like to have the option for a mobile app or just a web-based platform?

80 responses



Appendix B: High-level System Architectural Diagram

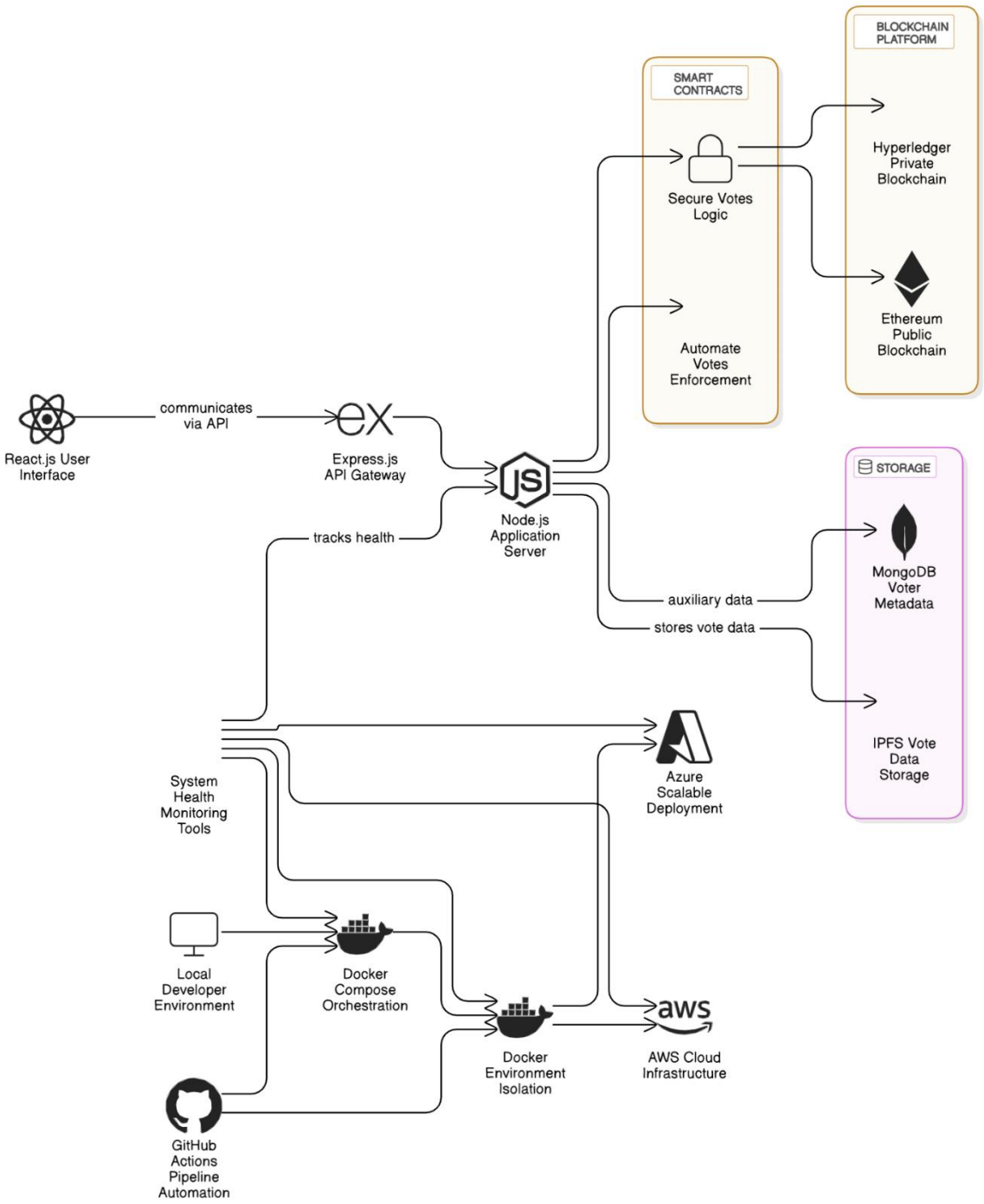


Figure 24: High-level System Architectural Diagram

Appendix 4: Stage Plans

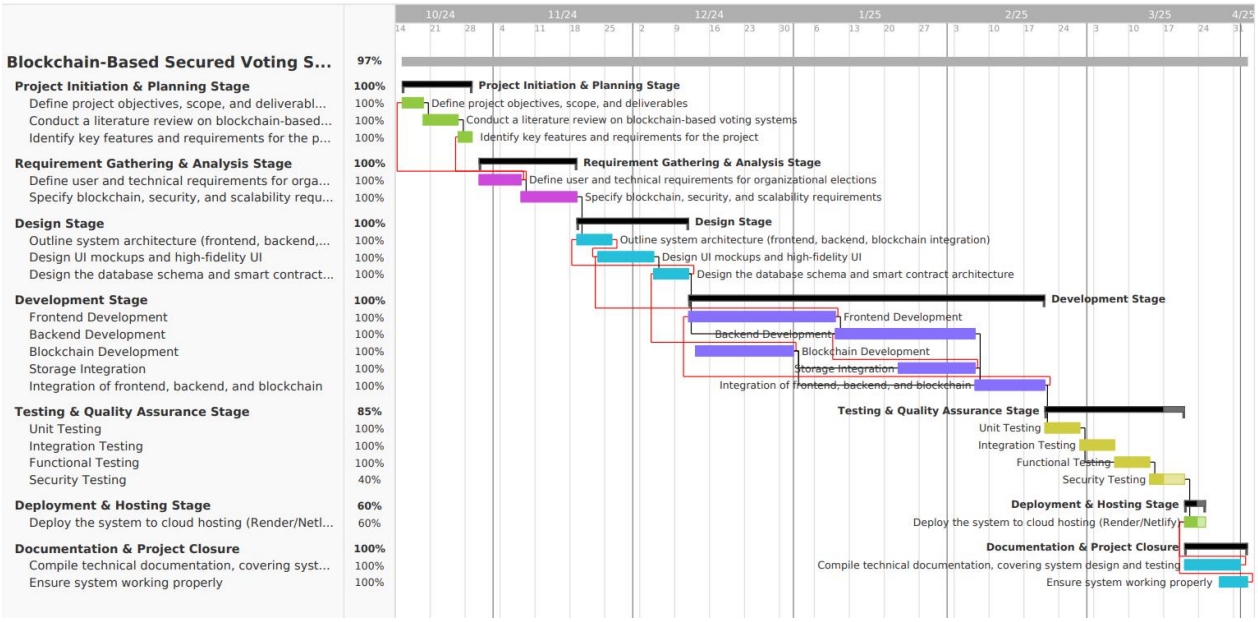
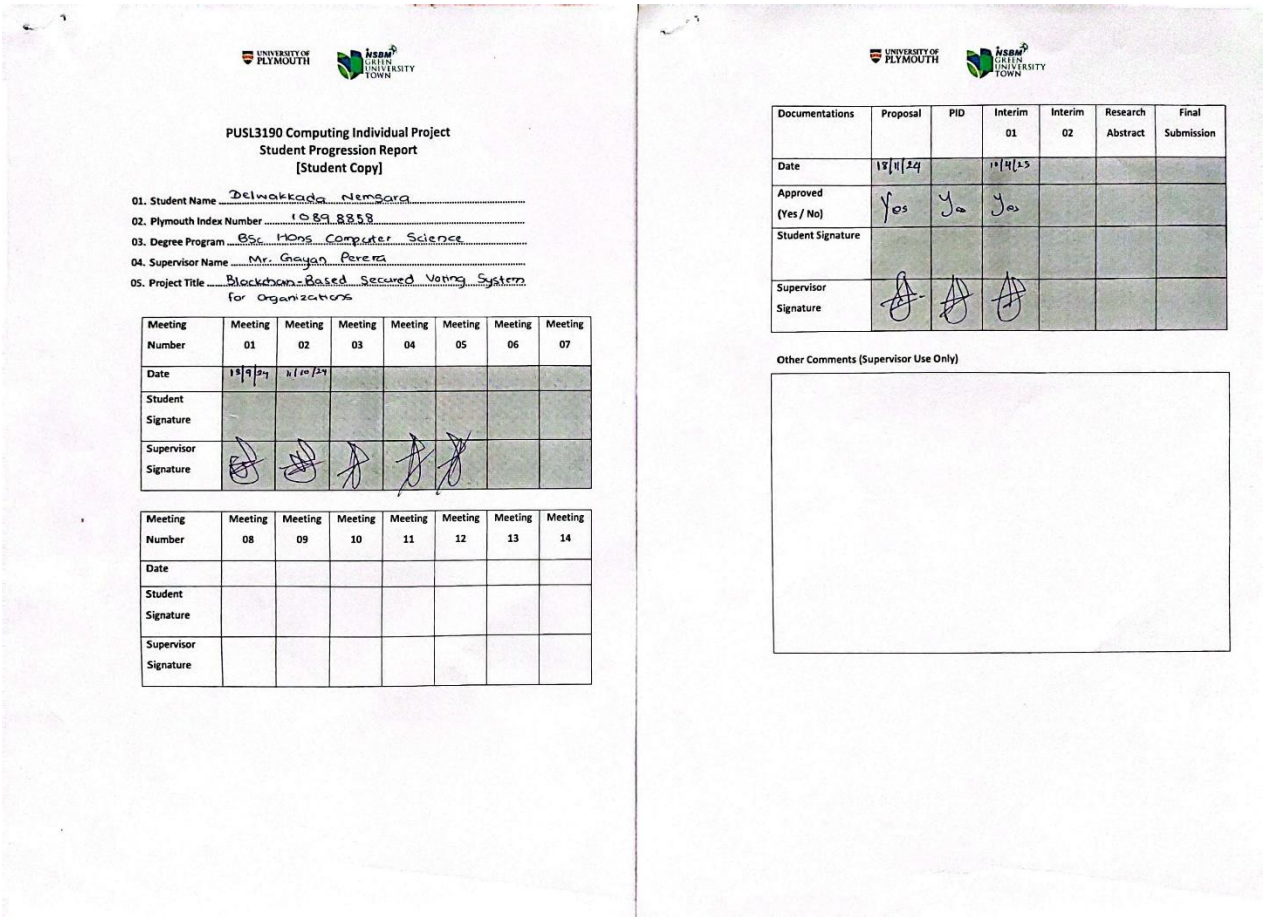


Figure 25: Gantt Chart Outlining Project Stages and Timelines

Appendix 5: Records of Supervisory Meetings



PUSL3190 Computing Individual Project
Student Progression Report
[Student Copy]

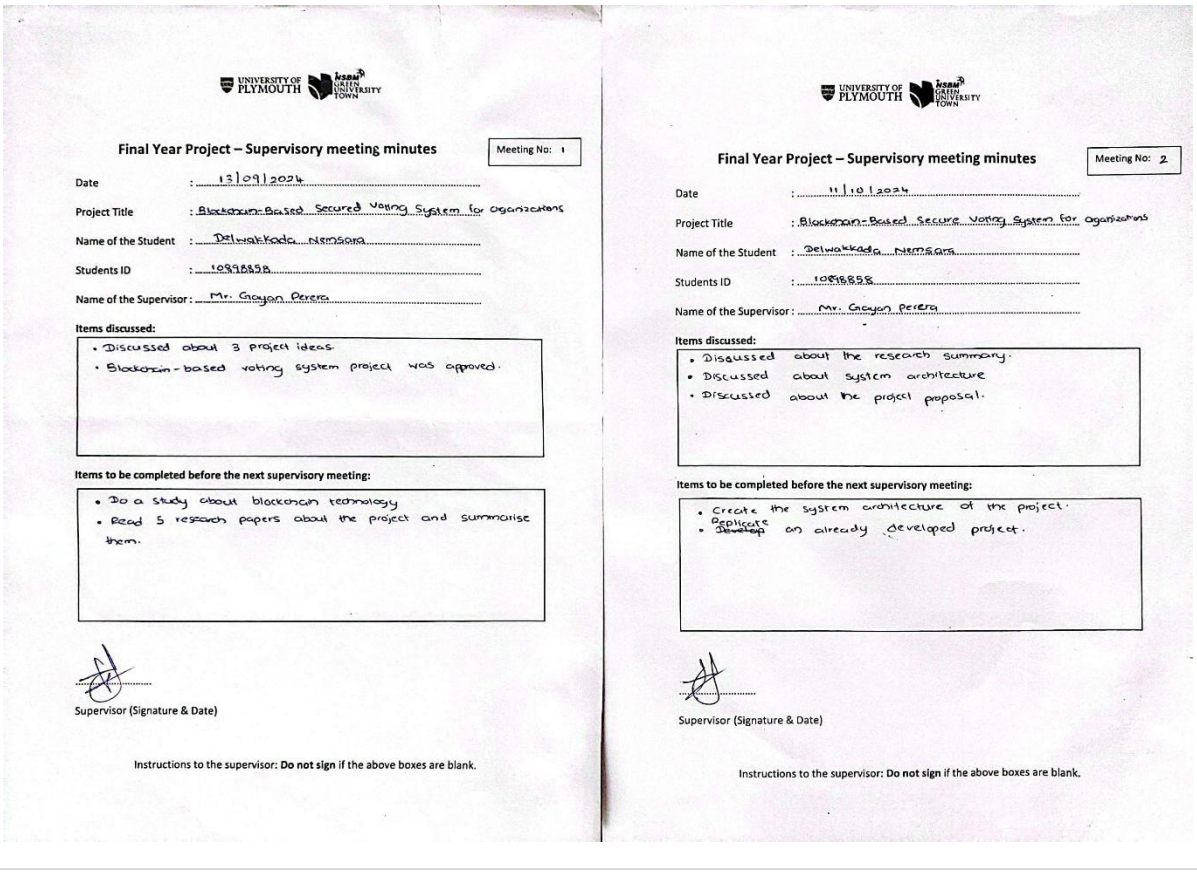
01. Student Name: Delwakkada Nemsara
02. Plymouth Index Number: 10898858
03. Degree Program: BSc Hons Computer Science
04. Supervisor Name: Mr. Gayan Perera
05. Project Title: Blockchain-Based Secured Voting System for Organizations

Meeting Number	Meeting 01	Meeting 02	Meeting 03	Meeting 04	Meeting 05	Meeting 06	Meeting 07
Date	18/9/24	1/10/24					
Student Signature							
Supervisor Signature							

Meeting Number	Meeting 08	Meeting 09	Meeting 10	Meeting 11	Meeting 12	Meeting 13	Meeting 14
Date							
Student Signature							
Supervisor Signature							

Documentations	Proposal	PID	Interim 01	Interim 02	Research Abstract	Final Submission
Date	18/11/24		10/1/25			
Approved (Yes / No)	Yes	Yes	Yes			
Student Signature						
Supervisor Signature						

Other Comments (Supervisor Use Only)



Final Year Project – Supervisory meeting minutes — Meeting No: 1

Date : 13/09/2024
Project Title : Blockchain-Based Secured Voting System for organizations
Name of the Student : Delwakkada Nemsara
Students ID : 10898858
Name of the Supervisor : Mr. Gayan Perera

Items discussed:
- Discussed about 3 project ideas.
- Blockchain-based voting system project was approved.

Items to be completed before the next supervisory meeting:
- Do a study about blockchain technology
- Read 5 research papers about the project and summarise them.

Supervisor (Signature & Date)

Instructions to the supervisor: **Do not sign** if the above boxes are blank.

Final Year Project – Supervisory meeting minutes — Meeting No: 2

Date : 11/10/2024
Project Title : Blockchain-Based Secure Voting System for organizations
Name of the Student : Delwakkada Nemsara
Students ID : 10898858
Name of the Supervisor : Mr. Gayan Perera

Items discussed:
- Discussed about the research summary.
- Discussed about system architecture
- Discussed about the project proposal.

Items to be completed before the next supervisory meeting:
- Create the system architecture of the project.
- Replicate ~~Develop~~ an already developed project.

Supervisor (Signature & Date)

Instructions to the supervisor: **Do not sign** if the above boxes are blank.

UNIVERSITY OF PLYMOUTH | iconn GREEN UNIVERSITY TOWN

Final Year Project – Supervisory meeting minutes

Meeting No: 3

Date : 22/11/2024

Project Title : Blockchain-Based Secure Voting System for Organizations.

Name of the Student : Delwakkada Nemsara

Students ID : 10898858

Name of the Supervisor : Mr. Gayan Perera

Items discussed:

- Discussed about the high-level system architecture of the system.
- ~~Draft of the~~ Project proposal document were presented
- Discussed about the consensus algorithms that can be used to implement the system.

Items to be completed before the next supervisory meeting:

- Implementation of the smart contracts for the voting.
- Document the blockchain implementation requirements.

Supervisor (Signature & Date)

Instructions to the supervisor: **Do not sign** if the above boxes are blank.

UNIVERSITY OF PLYMOUTH | iconn GREEN UNIVERSITY TOWN

Final Year Project – Supervisory meeting minutes

Meeting No: 4

Date : 07/02/2025

Project Title : Blockchain-Based Secure voting System for Organizations

Name of the Student : Delwakkada Nemsara

Students ID : 10898858

Name of the Supervisor : Mr. Gayan Perera

Items discussed:

- Discussed about the design of the platform. (Dashboard or a standard app)
- Discussed about the contents drafted in the PID document.
- Implemented ~~draft~~ smart contract using Solidity was presented demonstrated & discussed.

Items to be completed before the next supervisory meeting:

- Integration of blockchain layer to the React.js / Node.js web application.
- Consensus algorithm implementation which is suggested.

Supervisor (Signature & Date)

Instructions to the supervisor: **Do not sign** if the above boxes are blank.

UNIVERSITY OF PLYMOUTH | iconn GREEN UNIVERSITY TOWN

Final Year Project – Supervisory meeting minutes

Meeting No: 5

Date : 10/09/2025

Project Title : Blockchain-Based Secure Voting System for Organizations

Name of the Student : Delwakkada Nemsara

Students ID : 10898858

Name of the Supervisor : Mr. Gayan Perera

Items discussed:

- Interim Document was presented and discussed the changes needed for the final report.
- Development approach followed was demonstrated using the source code.
- Discussed the requirement of consensus algorithm like ~~proofs of~~ zero proofs knowledge to enhance the voter anonymity.

Items to be completed before the next supervisory meeting:

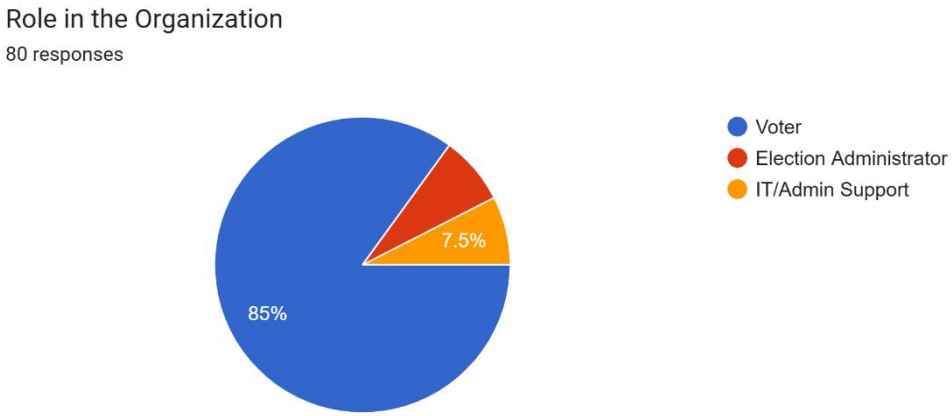
- Checking the feasibility to implement a algorithm like zero proofs knowledge and integrate it to the system to remove the risk of voter data getting disclosed on the blockchain.

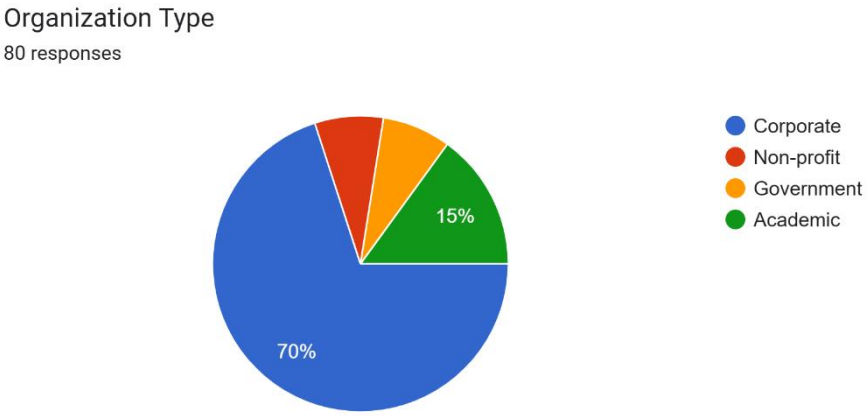
Supervisor (Signature & Date)

Instructions to the supervisor: **Do not sign** if the above boxes are blank.

Appendix 6: Requirement Gathering Survey Results

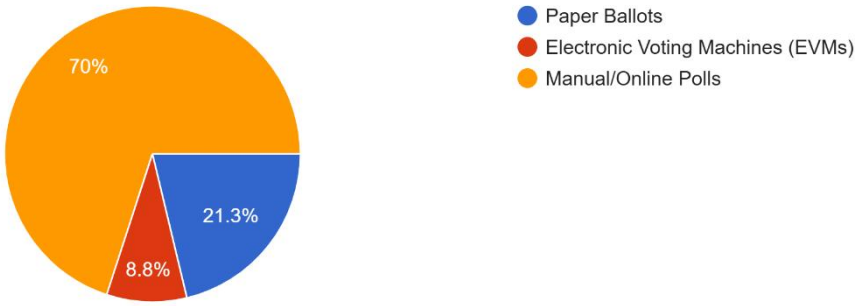
Below are detailed survey results and analytics for each question.





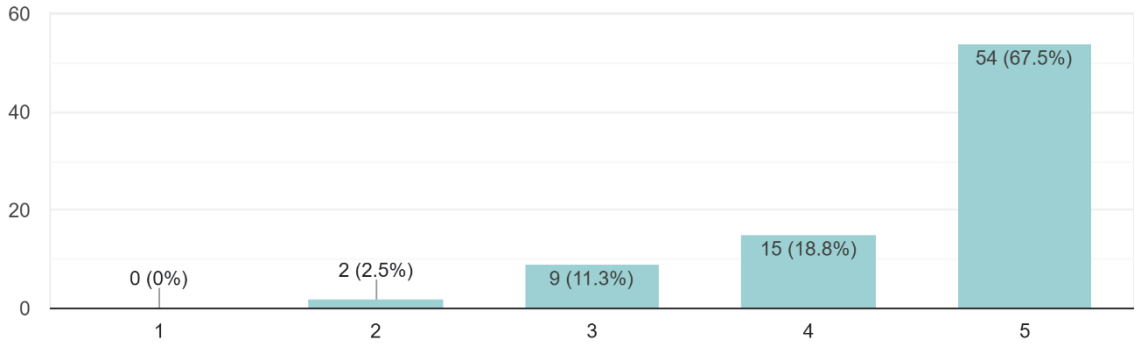
What current voting system do you use?

80 responses



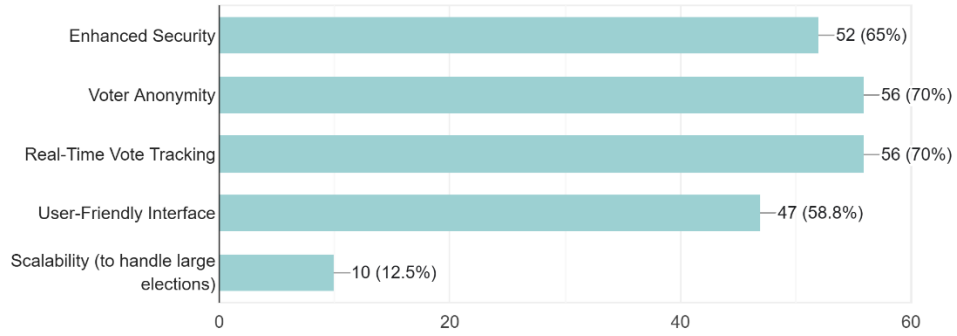
How important is it to ensure the security and tamper-proof nature of the voting process in your organization?

80 responses



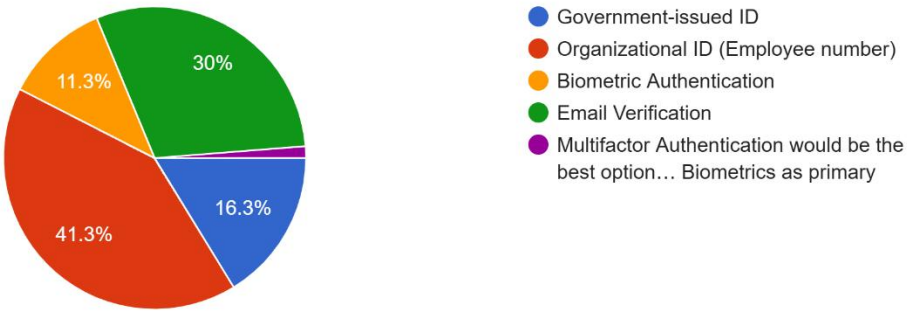
What features would you consider most important in a new voting system?

80 responses



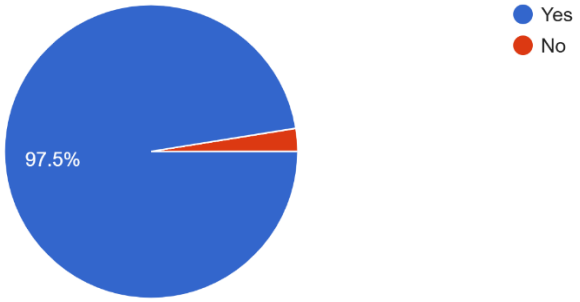
What is your preferred method for verifying voter identity?

80 responses



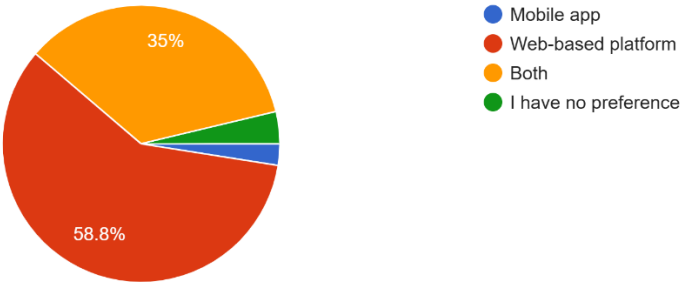
Do you think decentralized systems (blockchain) improve trust in the voting process?

79 responses



Would you like to have the option for a mobile app or just a web-based platform?

80 responses



Appendix 7: Preliminary UI Designs

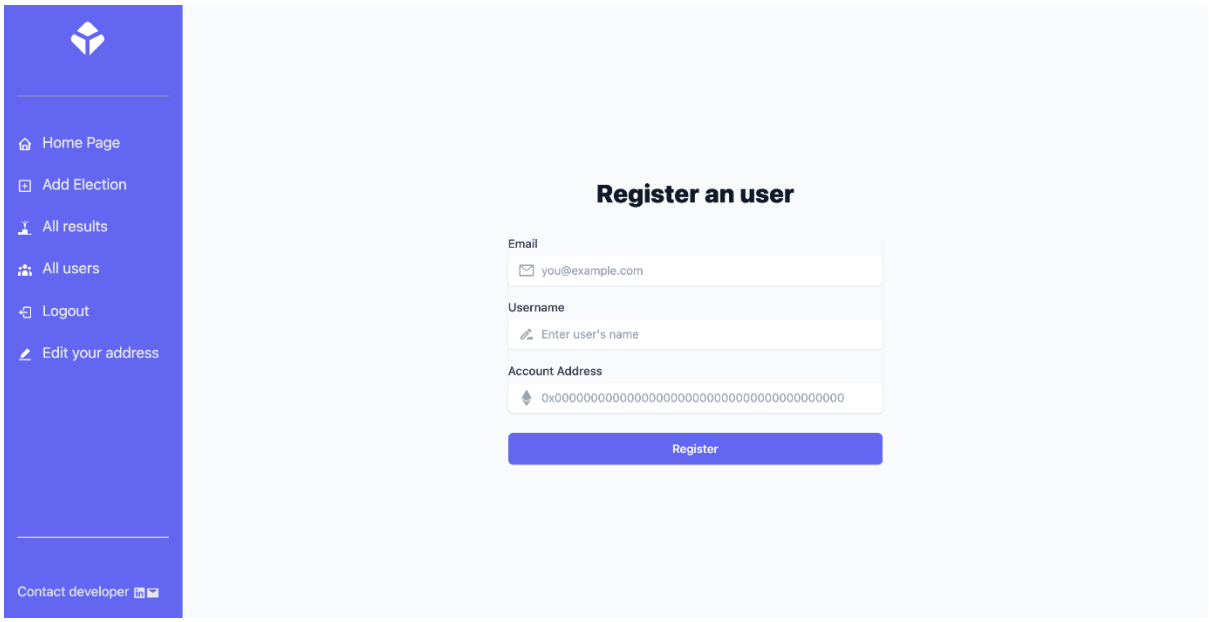


Figure 26: Registration Page UI Design

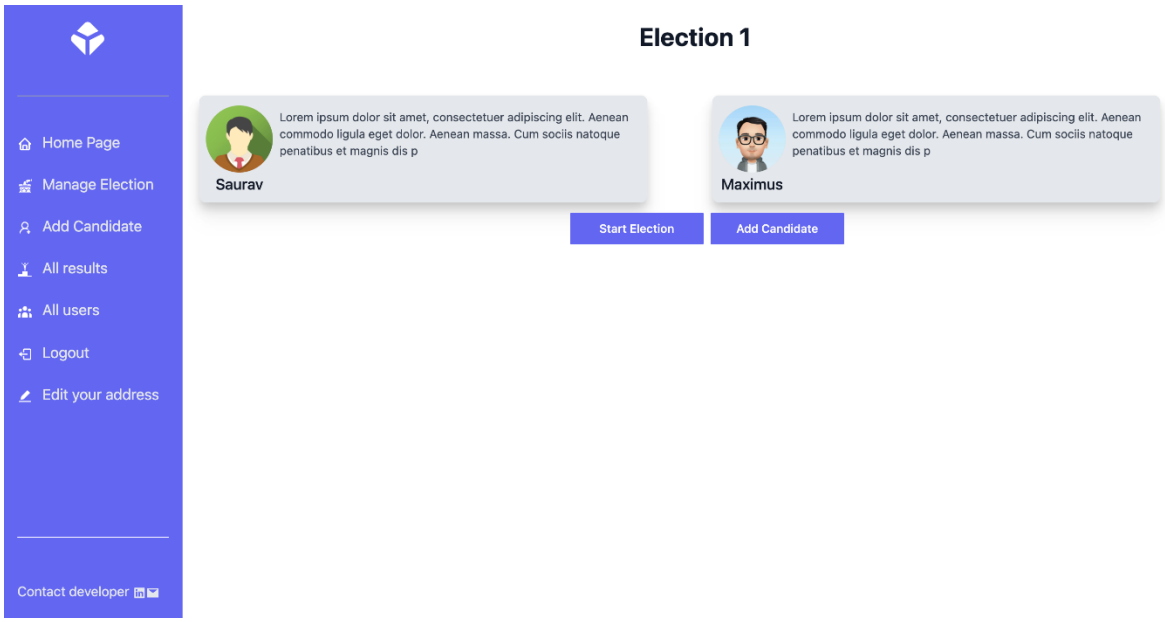


Figure 27: Election Page UI Design

Appendix 8: Frontend Testing Scripts

Figure 28: Frontend Unit Test for the Election Page

Frontend unit test for the Election page, validating election creation, input handling, and component rendering using Jest and React Testing Library

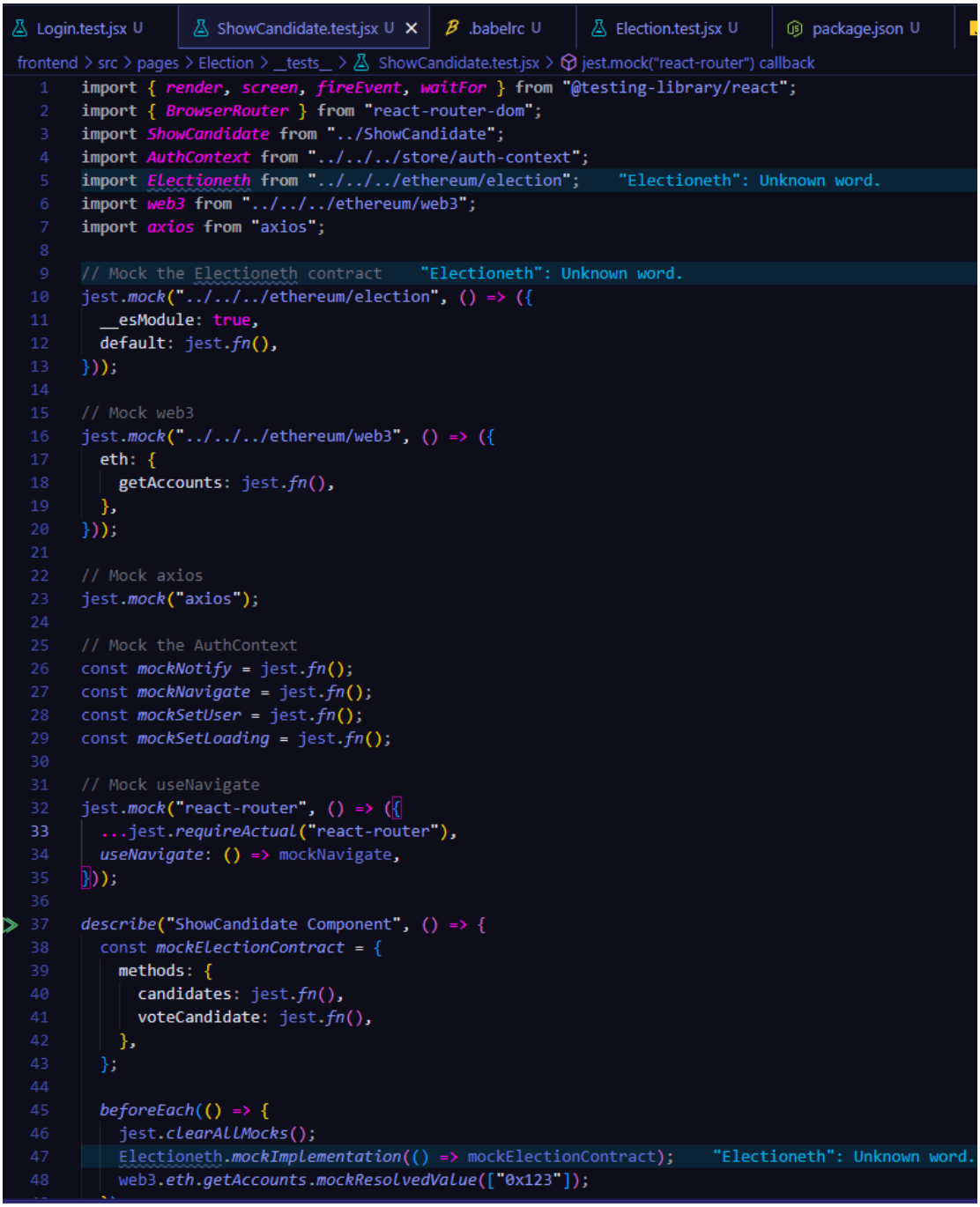


Figure 29: Test Script for Validating Candidate Display and Voting Functionality

Test script for the candidate display component, ensuring accurate rendering of candidate data and voting functionality under different user roles.

Appendix 9: Smart Contract Voting Workflow

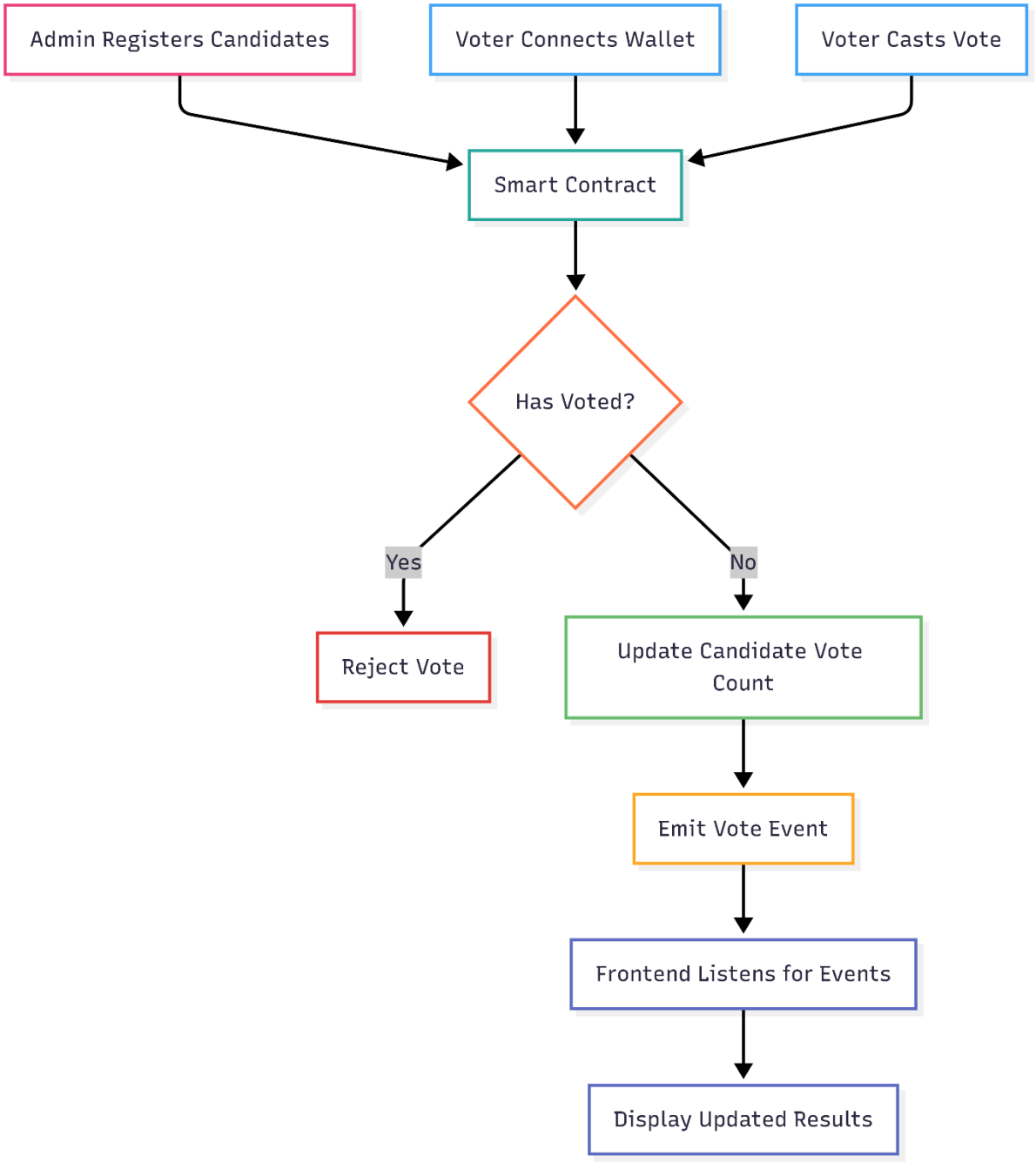


Figure 30: Smart Contract Voting Workflow

This diagram illustrates the operational flow of the Ethereum smart contract during voting. This highlights how candidate registration, wallet connection, vote casting, and vote validation are managed via the smart contract, including event emissions and front-end result updates.