IMPLEMENTATION OF WEB-BASED E-VOTING SYSTEM USING HYPERLEDGER FABRIC

MOYEGBEMI AJINOMISANGHAN

A project report submitted in conformity with the requirements for the degree of Master's of Science in Information Technology

Department of Mathematical and Physical Sciences
Faculty of Graduate Studies
Concordia University of Edmonton

IMPLEMENTATION OF WEB-BASED E-VOTING SYSTEM USING HYPERLEDGER FABRIC

MOYEGBEMI AJINOMISANGHAN

Approved:

Rossitza Marinova, Ph.D.

Supervisor Date

Committee Member Name, Ph.D.

Committee Member Date

Patrick Kamau, Ph.D.

Dean of Graduate Studies Date

Abstract

This document presents the design of a web-based voting system powered by Hyperledger Fabric, a permissioned blockchain framework, aiming to leverage blockchain's decentralized and immutable features to address the limitations of traditional voting. The architecture includes multiple organizations (voters, candidates, electoral authorities, and auditors) and essential components like a user management system, a web interface for voting, and a governing smart contract. Voters cast their votes using secured digital identities, with transactions being encrypted, validated by endorsing peers, and confirmed through Hyperledger Fabric's consensus mechanism. This approach ensures election transparency, verifiability, and resistance to tampering. However, challenges like voter privacy and regulatory compliance must be tackled for comprehensive adoption. In essence, this Hyperledger Fabric-based voting platform promises enhanced security and transparency in voting, yet demands further exploration and refinement to fully transform the democratic process.

Keywords: decentralized, immutability, Hyperledger Fabric, chaincode, permissioned blockchain framework

Acknowledgments

I want to express my appreciation to my amazing professor, Prof. R. Marinova, for her consistent support and insightful advice during this attempt. I was able to finish my project thanks to her expertise and perseverance, and I gained a lot of knowledge from working with her.

In addition, I value the educational and adventurous atmosphere that the Concordia University of Edmonton provides. This endeavor was successful thanks to the institution's resources, facilities, and intellectual stimulation.

My classmates continue to be a source of motivation and company for me. My journey was made better by their insightful advice, constructive criticism, and steadfast support. The success of this initiative can be attributed to our mutual enthusiasm for blockchain technology.

Contents

List of Figures

Listings

1 Introduction

1.1 Background

Blockchain technology has arisen as a potentially game-changing invention as a direct result of efforts to create election systems that are more safe, transparent, and efficient. Because of its durability, scalability, and permissioned nature, Hyperledger Fabric stands out among the different blockchain platforms now available [7]. As a result, it is an excellent option for the implementation of web-based electronic voting systems. This research investigates the feasibility of utilizing Hyperledger Fabric to modernize voting procedures and overcome existing issues associated with conventional voting systems. In recent years, concerns about the integrity of traditional voting methods have led to a growing desire for innovative solutions that offer better transparency and resilience against possible threats such as tampering, duplicate voting, and illegal access to voting data. These concerns have contributed to a rise in the number of countries adopting electronic voting systems. This research intends to investigate how the decentralized nature and immutability of blockchain technology might improve the credibility and trustworthiness of electoral outcomes. This will be done by merging web-based electronic voting with Hyperledger Fabric. The fact that the voting method under consideration is web-based gives a plethora of advantages to voters. For example, it enables voters to cast their ballots from any location as long as they have access to the internet, which increases both voter participation and accessibility [4]. The employment of digital identities for the purpose of voter authentication helps to ensure that the voting process is only undertaken by those who are qualified to do so, hence reducing the likelihood of voter fraud occurring.

The consensus mechanism of Hyperledger Fabric ensures that all transactions are validated and agreed upon by network participants. This eliminates the requirement for centralized authority and fosters a form of distributed governance. The immutability of the blockchain ensures that vote records will be permanent, protecting the honesty of the voting process while also providing a clear audit trail for verification purposes.

Despite the fact that the project places its primary emphasis on the benefits of web-based electronic voting using Hyperledger Fabric, it does recognise the existence of potential problems. These challenges include protecting the privacy of voters and addressing concerns over scalability for large-scale elections. The research intends to provide solutions to optimize the technology so that it can be adopted by a large number of people by examining these obstacles.

In the end, the incorporation of web-based electronic voting with Hyperledger Fabric presents a revolutionary chance to alter the democratic process. This opportunity will strengthen citizen engagement and faith in the results of election processes. This project aims to contribute to the development of more modern and robust election systems that preserve the ideals of transparency, security, and inclusivity by making use of the potential offered by blockchain technology [13].

1.2 Statement of Problem

Current voting systems lack transparency, safety, and confidence. Traditional voting techniques are inefficient, expensive, and may not accommodate for a huge community's varied viewpoints. This project aims to create an internet-accessible blockchain voting infrastructure that uses Hyperledger Fabric to swiftly overcome these issues and provide a more trustworthy voting procedure. Blockchain technology's decentralised and immutable qualities can improve election results and allow eligible voters to securely and easily submit their ballots from anywhere with internet access. Voting should be easy from anyplace with an internet connection.

But putting together a complicated vote system that uses Hyperledger Fabric presents a number of problems that need to be carefully looked into. Before the platform can be widely accepted and used, questions about privacy, scalability, and compliance with regulations need to be answered. This is a must if you want to solve basic problems.

With the help of Hyperledger Fabric's technology, the goal of this study is to make and use a safe blockchain voting system that can be accessed over the internet. By using blockchain technology, the proposed answer aims to improve the integrity and transparency of the voting process, making it less likely that voting data will be tampered with or accessed by people who shouldn't have access to it.

If a blockchain voting system that can be used over the internet can be made and put to use without any problems, it could change the way elections are run in a way that no one has ever seen before. This new change could make people more likely to vote and keep election results honest and clear. This project wants to make a big difference in improving democratic practices and government all over the world by getting around the problems with standard voting methods and using cutting-edge technology.

1.2.1 Potential Advantages of E-Voting Systems

Convenience and Accessibility: Electronic voting technologies allow voters to vote online even if they are not at the polling station. Because of this, voters no longer need to physically attend polling stations to cast their ballots, which benefits those with mobility issues or who live in rural areas. Electronic voting may boost voter turnout, making democracy more inclusive and participatory.

Real-Time Vote Counting: Electronic voting systems tabulate votes in real time, unlike paper ballot voting methods, which require human counting. This speeds up election results announcement, reducing wait time and improving voting efficiency. Eliminating human error in vote counting improves accuracy and reliability.

Cost Savings and Environmental Impact:eVoting eliminates ballot sheets, printing, and logistics, reducing election expenses. Governments and electoral bodies can use this cost-effectiveness to fund other vital services. eVoting also promotes ecologically friendly behaviours by reducing paper ballot waste.

1.2.2 Concerns Regarding Security, Privacy, and Trust in Existing E-Voting Solutions

Security Vulnerabilities: eVoting has many advantages, but security issues remain. Cyber risks and data breaches could jeopardize election integrity by manipulating voting data or accessing voter information. Maintaining vote confidentiality and integrity requires strict security measures.

Preserving Voter Privacy and Anonymity: Ensuring voter privacy and anonymity is essential to protect individuals from coercion or intimidation. E-Voting systems must ensure that votes are recorded without revealing the voter's identity, striking a delicate balance between verifiability and voter anonymity [5].

Building Trust and Transparency: Building trust among stakeholders, including voters, election authorities, and candidates, is paramount to the successful adoption of eVoting systems. Providing transparency in the voting process and ensuring that the results are verifiable are essential elements to gain confidence in the integrity of the system.

To help with these concerns, blockchain technology, particularly Hyperledger Fabric, emerges as a promising solution to create a secure, transparent, and trustworthy eVoting platform. By leveraging blockchain's decentralized and immutable nature, a Web-based eVoting Website using Hyperledger Fabric can address security vulnerabilities, ensure privacy and verifiability, and enhance trust in the democratic voting process [15]. The subsequent sections will delve into the proposal for this blockchain-based eVoting system, outlining how it effectively mitigates the challenges and sets a new standard for modern, secure, and transparent democratic elections.

2 Project Objectives

This project aims to provide a safe, transparent, and efficient Hyperledger Fabric-based Web-based eVoting Website. The project's main goals are to fix existing eVoting system issues:

2.1 Enhancing Security and Integrity: The first and foremost objective of the project is to develop an eVoting platform that ensures the highest level of security and integrity throughout the voting process. By leveraging the decentralized and tamper-resistant nature of blockchain, Hyperledger Fabric will be utilized to protect voting data from unauthorized access and manipulation. The cryptographic techniques and consensus mechanisms in Hyperledger Fabric will strengthen the resilience of the system against cyber threats, ensuring that votes are recorded accurately and securely. The project aims to create an ecosystem where voters can have full confidence that their votes are protected and cannot be altered or tampered with by any malicious entity.

2.2 Ensuring Privacy and Anonymity: Addressing concerns related to voter privacy and anonymity is a critical objective of the project. Hyperledger Fabric's permissioned network will be utilized to verify and authenticate voters while ensuring that their identities remain anonymous during the voting process [2]. Through the use of cryptographic techniques such as zero-knowledge proofs, the system will guarantee that votes are cast in a confidential manner, preventing any form of voter coercion or intimidation. By ensuring privacy and anonymity, the project aims to foster a sense of trust and confidence among voters, encouraging broader participation in the electoral process.

2.3 Providing Verifiability and Transparency: Another vital objective is to create a transparent and auditable eVoting system. Hyperledger Fabric's distributed ledger technology will record every casted vote on the blockchain, making it visible to all network participants [6]. This transparency will allow voters to verify that their votes were accurately recorded and counted. Election authorities can conduct audits to ensure the integrity of the electoral process, providing stakeholders with a verifiable and transparent voting experience. By making the voting process more transparent, the project aims to enhance trust in the system and reduce skepticism surrounding election outcomes.

2.4 Improving Accessibility and Convenience: The project aims to improve accessibility and convenience for voters by developing a Web-based eVoting platform. Citizens will be able to cast their ballots from any location with an internet connection, eliminating the need for physical attendance at polling stations. This increased accessibility is expected to enhance voter turnout and foster a more inclusive democratic process. By making voting more convenient and accessible, the project aims to promote civic engagement and strengthen democratic participation.

2.5 Reducing Costs and Enhancing Efficiency: Leveraging blockchain technology can lead to cost savings in election administration. The elimination of physical ballots and manual vote counting will reduce operational costs for electoral bodies. Moreover, the use of Hyperledger Fabric's smart contracts will streamline the voting process, leading to quicker and more efficient elections [9]. By reducing costs and enhancing efficiency, the project aims to make the electoral process more sustainable and resource-efficient.

2.6 Leveraging Hyperledger Fabric's Advantages: The project emphasizes the utilization of Hyperledger Fabric as the framework for its unique advantages. Hyperledger Fabric's modular architecture allows for customization and scalability, enabling the system to adapt to diverse voting scenarios. The platform's permissioned network ensures that only verified participants can engage in the voting process, enhancing security and trust. Furthermore, the smart contract capabilities of Hyperledger Fabric streamline the execution of voting rules and enable automatic vote tallying, leading to greater efficiency and accuracy in election administration.

By achieving these objectives, the proposed Web-based eVoting Website using Hyperledger Fabric seeks to revolutionize the way elections are conducted, offering a secure, transparent, and user-friendly eVoting solution that instills trust and confidence in the electoral process. Through the utilization of blockchain technology, the project aims to set a new standard for modern democratic elections, paving the way for more reliable and inclusive voting systems.

3 Literature Review and Theoretical Framework

3.1 Overview of E-Voting Systems:

Electronic Voting (eVoting) systems have emerged as potential alternatives to traditional paper-based voting methods, offering the promise of increased accessibility, efficiency, and voter participation. Existing eVoting systems can be categorized into online voting platforms and blockchain-based solutions [18]. Online voting systems enable voters to cast their ballots remotely, reducing the need for physical polling stations and increasing convenience. However, these systems have faced scrutiny due to concerns over cybersecurity, voter authentication, and the potential for vote manipulation.

On the other hand, blockchain-based eVoting systems have gained attention for their decentralized and tamper-resistant nature [1]. Blockchain technology ensures transparency and immutability by recording votes on a distributed ledger, making it challenging for any single entity to alter the results without consensus from the entire network. The use of smart contracts in blockchain-based eVoting further enhances security and automation in the voting process. Despite these advantages, the adoption of blockchain-based eVoting faces challenges related to scalability, voter privacy, and the technical complexity of the underlying blockchain platforms.

3.2 Blockchain Technology in E-Voting:

The blockchain technology that forms the basis of cryptocurrencies such as Bitcoin has developed beyond the scope of its initial application to change a variety of industries, one of which is electronic voting. Because of blockchain's distributed and decentralized structure, there is no way for a centralized authority to exert control over the voting process. This significantly lowers the likelihood that votes will be manipulated or fraudulently cast. Blockchain-based electronic voting provides voters with a verifiable audit trail of every ballot they cast by recording votes in an immutable ledger. This enables voters to independently audit and verify election results [20].

A number of different blockchain technologies have been investigated for use with eVoting, with Ethereum and Hyperledger Fabric emerging as the two most popular options. The features of smart contracts offered by Ethereum make it possible to build decentralized voting systems, which eliminates the requirement for third-party intermediaries and guarantees the transparent execution of voting rules. On the other hand, Hyperledger Fabric, which has an architecture that is both modular and extensible, provides increased control over privacy and permissioning. As a result, it is an excellent choice for the development of enterprise-level electronic voting applications. Despite these benefits, blockchain-based electronic voting still faces issues in the areas of voter anonymity, identity verification, and maintaining the integrity of data sources that are not part of the blockchain [3].

3.3 Hyperledger Fabric as a suitable platform

Hyperledger Fabric, a prominent framework under the Linux Foundation, provides a robust foundation for building secure and scalable enterprise blockchain applications, including eVoting systems. Its modular architecture allows for the fine-tuning of network components, ensuring privacy and confidentiality among participants. Hyperledger Fabric's permissioned network model ensures that only verified and authorized entities can participate in the voting process, addressing concerns related to Sybil attacks and unauthorized access [14].

The endorsement policy of Hyperledger Fabric ensures that transactions (votes) are validated by authorized peers before committing them to the ledger, enhancing the overall integrity of the eVoting system. Moreover, its pluggable consensus mechanism allows for flexibility in choosing an appropriate consensus algorithm based on the specific requirements of the eVoting application. Hyperledger Fabric's support for off-chain data through external services enhances the integration of real-world data into the voting process without compromising privacy.

By adopting Hyperledger Fabric as the underlying framework for the proposed Web-based eVoting Website, this project aims to harness the power of blockchain technology to address the limitations of traditional voting systems and enhance the security, transparency, and trust in the eVoting process. The theoretical framework presented in this literature review sets the stage for the subsequent sections, where the project's objectives and design will be detailed to realize the vision of a reliable and democratic eVoting platform.

The Execute-Order-Validate (EOV) architecture in Hyperledger Fabric is a novel approach that distinguishes it from traditional blockchain platforms. As illustrated in Figure 1, this architecture can be elucidated in three distinct phases:

- **Execution**: Endorsing peers perform smart contract logic to execute transactions. An endorsement policy in the chaincode (smart contract) specifies endorsing peers. The endorsing peer(s) sign the proposed ledger change (also known as a "proposal response" or "endorsed transaction").
- **Ordering**: After signing, the ordering service orders all incoming transactions into a block. The ordering service creates a transaction history by arranging all network transactions.
- **Validation**: Network peers receive ordered transaction blocks. The peers then review the blocks' transactions for endorsement policy compliance and double-spend problems. The ledger commits transactions after this validation phase.

This separation of the execution and validation processes allows the network to perform transactions in parallel without a global order of execution, increasing flexibility and scalability. Ordering transactions after execution reduces invalid transaction effort, boosting performance.

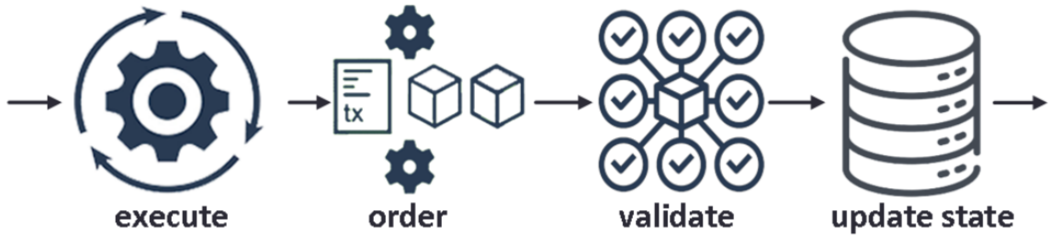


Figure 1: Execute-Order-Validate Architecture in Hyperledger Fabric. Source: **Hyperledger Fabric Documentation**

3.3.1 Permissioned Ledger:

One of the fundamental reasons for choosing Hyperledger Fabric is its support for a permissioned ledger. In an eVoting system, it is essential to ensure that only verified and authorized participants can access and participate in the network. Hyperledger Fabric's permissioned network model allows for fine-grained control over access, ensuring that only trusted entities, such as voters, candidates, and the Election Commission, are allowed to interact with the blockchain. This feature mitigates the risk of Sybil attacks and unauthorized access, safeguarding the integrity of the voting process.

3.3.2 Consensus Mechanism:

A pluggable consensus mechanism in Hyperledger Fabric lets the eVoting application choose a consensus algorithm. Choose consensus algorithms like Practical Byzantine Fault Tolerance (PBFT) or Raft for an eVoting system that prioritizes security and efficiency. PBFT provides Byzantine fault tolerance to allow malevolent nodes and yet reach consensus on transaction validity. Raft has a simpler consensus method for quick transaction finality. Hyperledger Fabric is ideal for eVoting due to its consensus mechanism flexibility.

3.3.3 Privacy Controls:

Since voters must conceal their identities and voting intentions, electronic voting creates privacy problems. Hyperledger Fabric's channels feature lets users create private sub-networks within the larger network, adding privacy to the platform. This isolates sensitive voting data and only shares it with relevant participants to protect voter names and preferences. Hyperledger Fabric allows private data collecting [19]. This lets authorized parties store confidential data off-chain with cryptographic integrity assurances. Hyperledger Fabric is a good choice for building a safe, privacy-protected electronic voting system.

3.3.4 Scalability and Performance:

Hyperledger Fabric's modular architecture and endorsement policies boost performance and scalability. E-Voting system throughput increases by partitioning the network into many channels and using endorsement policies to control transaction endorsement [17]. Hyperledger Fabric's scalability allows the network to conduct a large number of voting transactions without affecting performance, making it perfect for large voter populations [10].

The web-based eVoting website is designed with a high-level architecture that leverages the capabilities of Hyperledger Fabric to ensure a secure and transparent eVoting process. The architecture consists of several components, each with specific roles and responsibilities.

Endorsing peers, ordering nodes, and the client application form the Hyperledger Fabric network. Only permitted and legitimate transactions are added to the blockchain by endorsing peers. Ordering nodes manage consensus and order and add agreed-upon transactions to the blockchain [12].

4 Project Design

4.1 Architectural Overview

As shown in Figure 2, the architecture of the system is designed to facilitate a secure and transparent voting process. Voters and the Election Commission interact with the system through a client application. This application serves as the main interface for users to cast their votes and for the Election Commission to oversee the process.

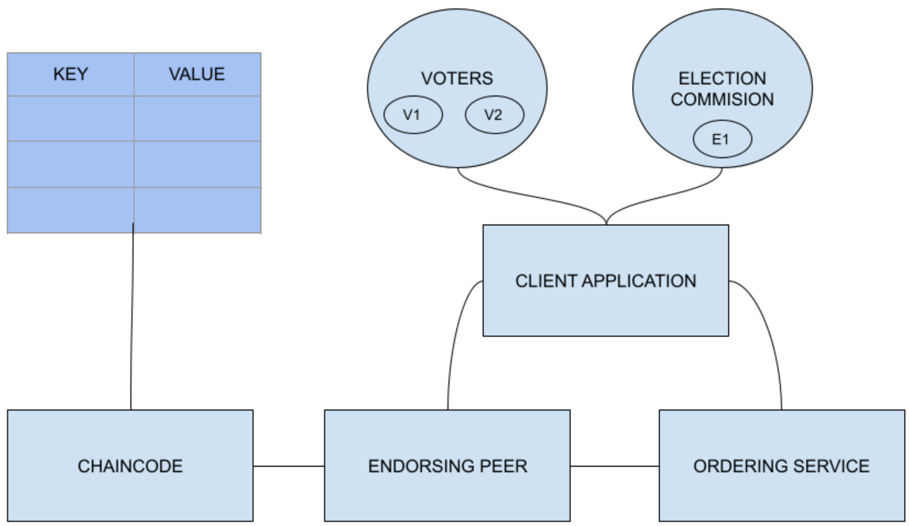


Figure 2: System Architecture

The client application communicates with the Ordering Service, a key component responsible for ordering transactions into blocks. This creates a consistent and transparent history of all transactions. Following the proposal of transactions by the client application, they are sent to the Endorsing Peer. This is a network node that simulates transactions and provides endorsement, thereby ensuring the validity of each vote.

The Endorsing Peer utilizes a chaincode, also known as a smart contract, which contains the business logic of how votes are cast and counted. After the chaincode is executed and the transactions are endorsed, they are ordered by the Ordering Service and subsequently update the Vote Ledger.

The Vote Ledger is a comprehensive record of all votes cast, offering a transparent and immutable chronicle of the voting process. This architecture thus presents a secure and efficient platform for conducting elections, making the most of blockchain's transparency, security, and verifiability.

4.2 Network Configuration

In the area of blockchain technology as it exists today, Hyperledger stands out as the leading center for the development of new applications of the technology. Its Hyperledger Fabric module offers a sturdy foundation upon which the architectural strata of applications, such as voting systems, can be created. These applications may include voting systems. The capacity to carry out strict validation checks is one of the primary benefits that comes with putting into operation a permissioned blockchain platform such as Hyperledger Fabric. This assures that the voting process is only open to registered voters, whose identities have been thoroughly checked, and that these voters are the only ones who may take part in it. Additionally, while the system is checking voter credentials, it maintains a shroud of confidentiality. This ensures that voter names stay secret, which maintains the concepts of anonymous voting and helps uphold its principles.

The voting application has been modeled after a system that is both streamlined and thorough. A single client application acts as the medium via which users communicate with one another, but with a variety of user interfaces specifically designed to cater to different user types. The modular approach helps to ensure that user management and data separation are carried out in an effective manner.

The importance of endorsements from peers cannot be overstated. Within the context of this ecosystem, a particular endorsing peer has been given the task of overseeing the execution of chaincode within the Docker container that corresponds to it. In addition, the ordering node, which is an essential component, is responsible for ensuring that the transactions are sequenced in the appropriate order and, as a result, the integrity of the voting sequence is preserved.

Diving into the network configurations, this application employs the sample network 'basicnetwork', which initiates the following instances and situates them within Docker containers:

- **Orderer**: Central to the voting application, the orderer processes transactions (votes), organizing them sequentially and distributing blocks to network peers. This systematic operation ensures that each transaction is processed in the right order, contributing to a consistent view of the ledger across the network. Each vote, being a transaction, is ordered and validated before it is added to the blockchain.

- **Certifying Authority**: The Certifying Authority (CA) is integral to maintaining the security and authenticity of the network. The CA issues digital identities to all participants, such as voters and network nodes (like the peer and orderer). Only verified voters are permitted to participate, with the CA taking charge of issuing these verified identities. This strategy minimizes the risk of fraudulent votes by ensuring each vote cast on the network originates from a verified voter.

- **Org1 maintaining Peer0**: In this system, the peer maintained by Org1 hosts both the ledger and the voting smart contract (chaincode). The peer, upon receiving ordered blocks of votes from the orderer, adds them to the ledger. Each vote is treated as a transaction processed by the peer. Additionally, the peer handles the execution of the voting chaincode, potentially containing rules and logic exclusive to the voting process (e.g., eligibility criteria, vote counting).
- **CouchDB**: Serving the voting application, CouchDB acts as the state database storing the current ledger state. This state represents the latest voting results at any given time, offering an efficient method to query the current voting status and allowing swift retrieval of voting results as required.
- **CLI**: The Command Line Interface (CLI) provides a tool that allows developers or administrators to engage with the network. It handles a variety of tasks, such as creating and joining channels, installing and instantiating the voting chaincode, and querying the current state of voting. Administrators may deploy new versions of the voting chaincode or query the current state of votes using the CLI.

Hyperledger Fabric's permissioned and distributed nature allows only verified voters to vote in this project. The blockchain securely stores each vote, creating a trustworthy and transparent voting mechanism. This greatly eliminates fraud and tampering.

Hyperledger Fabric's permissioned and distributed nature allows only verified voters to vote in this project. The blockchain securely stores each vote, creating a trustworthy and transparent voting mechanism. This greatly eliminates fraud and tampering.

4.3 Project Timeline

The Gantt chart shown in Figure 3 provides a comprehensive representation of the project's timeline, illustrating the structured sequencing and anticipated completion dates of the various tasks and milestones associated with the endeavor.

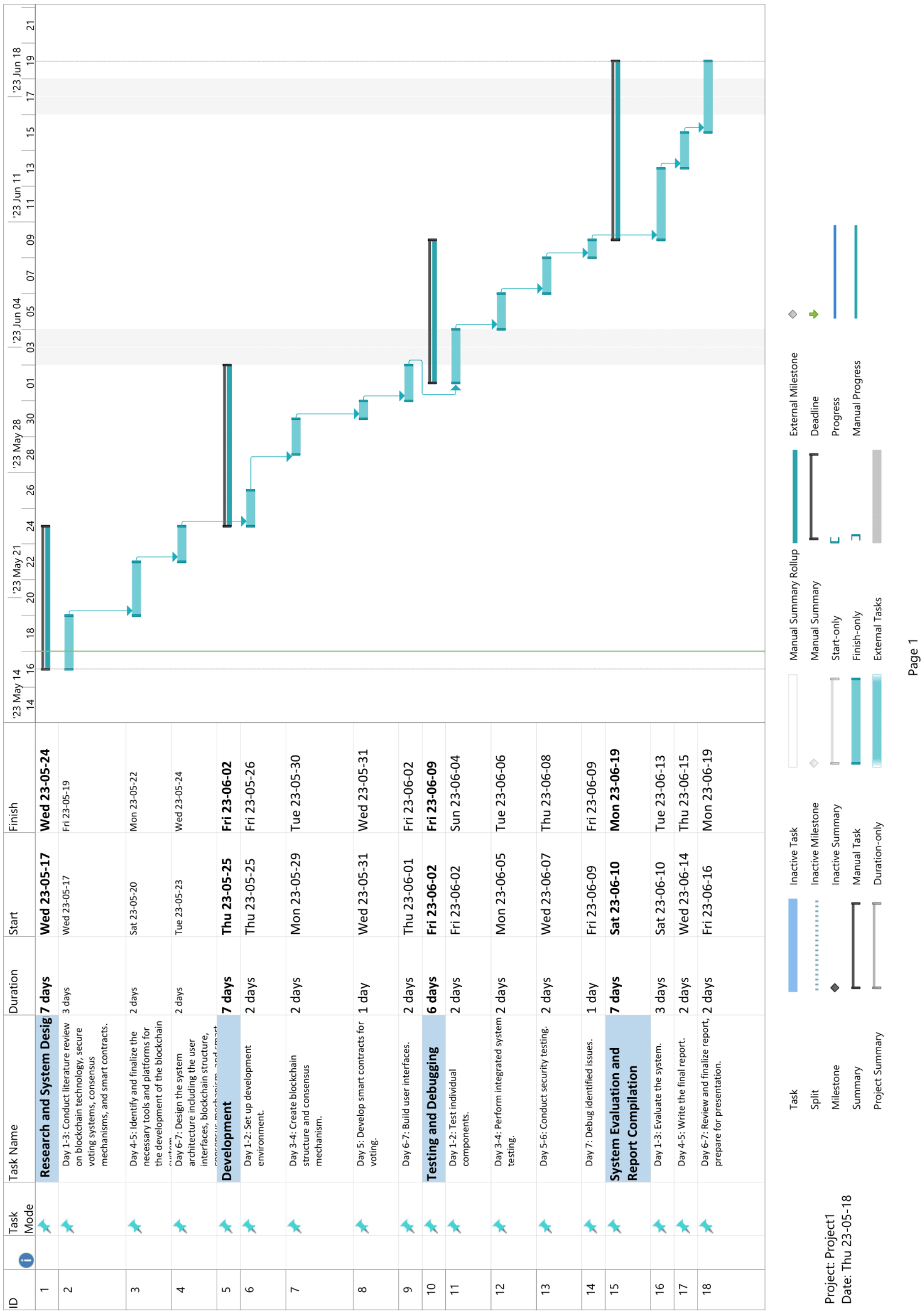


Figure 3: Project Timeline

4.4 User Interfaces

4.4.1 EC-Dashboard

Figure 4 introduces the EC-Dashboard, a comprehensive tool designed for the Election Commission to manage the entire election process. Post authentication, the Election Commission can register new voters, create unique voter IDs and PINs, and send these credentials to voters via email. The dashboard also enables the commission to add candidates to the election roster and set the end time for the election, thereby marking the start of the voting process. The real-time monitoring feature provides a dynamic view of voter turnout, and once the voting concludes, the commission can readily access election results and related statistics. All these functionalities reinforce the transparency and efficiency of the election process.

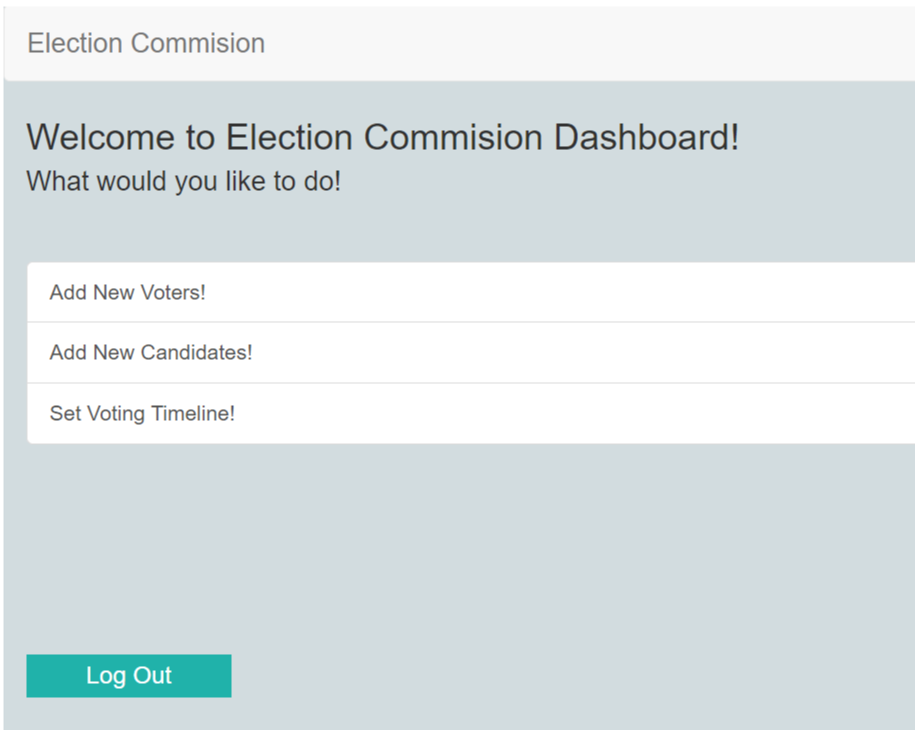


Figure 4: Election Commission Dashboard

4.4.2 Voter-Dashboard

As shown in Figure 5 the Voter-Dashboard is designed for registered voters to participate in the voting process securely. Voters can access the dashboard using their unique voter IDs and PINs received via email. Upon login, voters can view the list of candidates and their details, enabling them to make an informed voting decision. The dashboard allows voters to cast their votes by entering the candidate ID of their preferred candidate. Once a vote is cast, it cannot be modified, ensuring the integrity of the voting process. The Voter-Dashboard also provides a feature for voters to query and verify their vote by using their vote IDs.

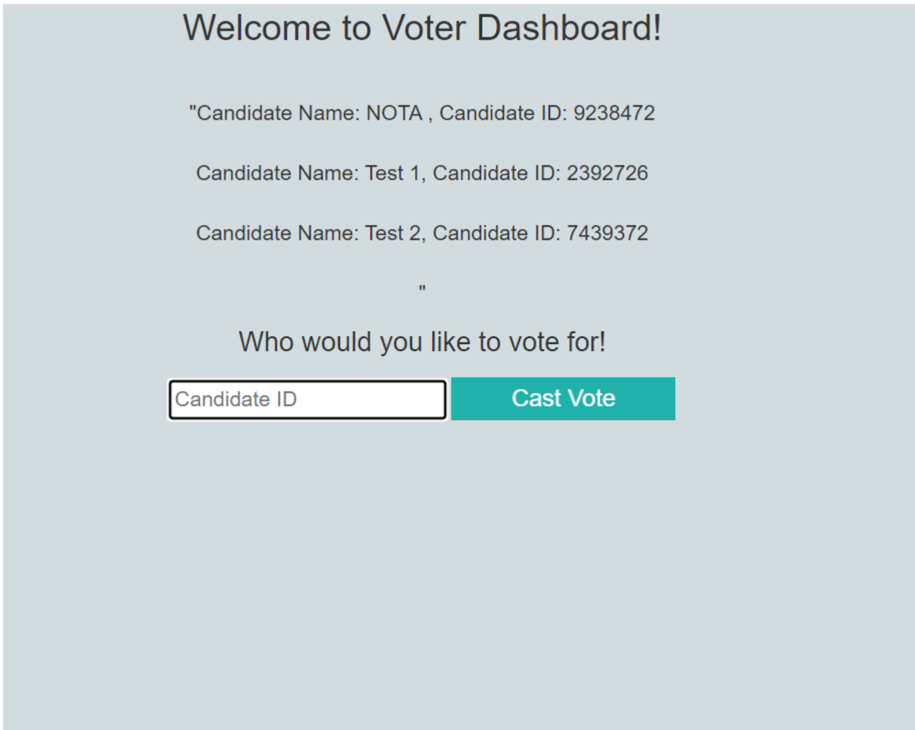


Figure 5: Voter Dashboard

4.4.3 Public Bulletin

As shown in Figure 6, the public bulletin operates as an important information center for the voting process. It offers essential election insights that are available to both the voters and the Election Commission. The public bulletin primarily comprises of Election Results and Voter Turnout data, providing real-time updates that reflect transparency in the democratic process.

In addition to these fundamental components, the bulletin also allows for querying specific votes that have been cast, a feature which bolsters the transparency and auditability of the voting process. Users can also locate a particular vote using the Vote ID, further enhancing the platform's user-friendliness and interactive nature.

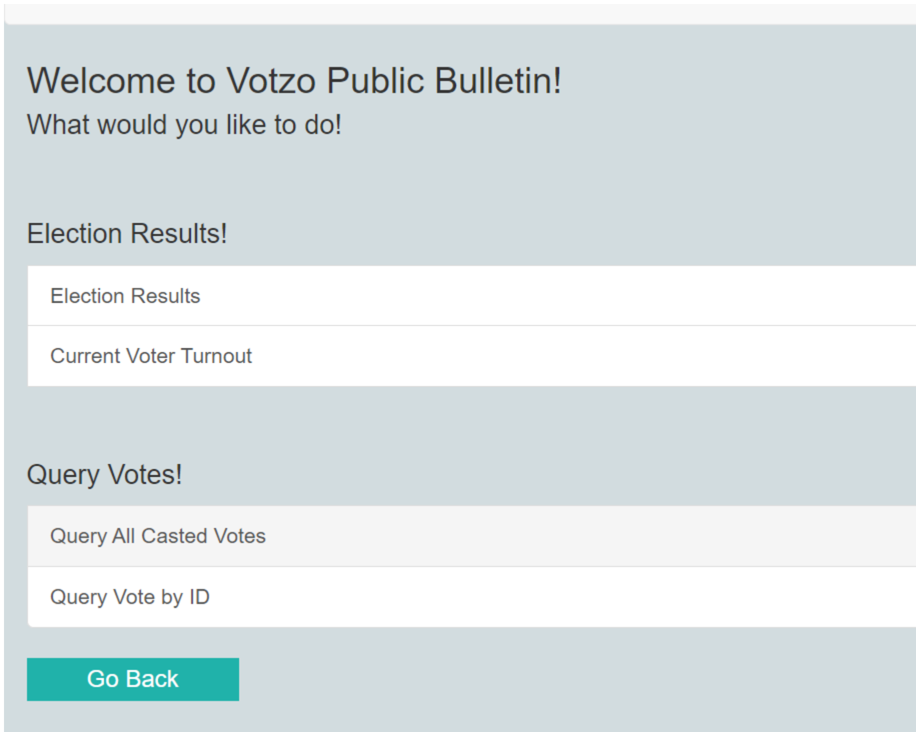


Figure 6: Public Bulletin

The web-based eVoting website's user interfaces are carefully designed to cater to the needs of different stakeholders, ensuring a smooth, secure, and trustworthy eVoting experience for all participants.

5 Project Implementation and Results

5.1 Smart Contract (Chaincode) Functions

Smart contracts, often referred to as chaincode in the Hyperledger Fabric context, are programs that encode business logic, rules, and processes [8]. They provide the functionality needed to interact with the ledger, effectively enabling transaction execution.

- **createVoteObj**: This function is essential in the initialization of the voting process. It creates a new voting object with necessary properties, like candidates' details and vote counts, all set to zero initially. It's invoked by the election commission during the setup of a new election, and this voting object is subsequently stored in the ledger, awaiting votes from voters.
- **setEndTime**: The setEndTime function contributes to the security and integrity of the voting process. Voting periods are time-bound, providing a clear timeline for voters and mitigating potential manipulation. Once the end time has been reached, no more votes can be accepted for that particular voting session.
- **sendVoteObj**: Voting is the central function in the voting process. The sendVoteObj function is responsible for accepting a voter's choice, updating the voting object, and ensuring the vote is recorded securely and anonymously.
- **getResults**: After the voting session has ended, the getResults function is invoked. It retrieves the voting object, calculates the results based on received votes, and returns the outcome of the election. This function ensures transparency and immediacy of results, as they are available as soon as the voting period ends.

5.2 Voting Procedure

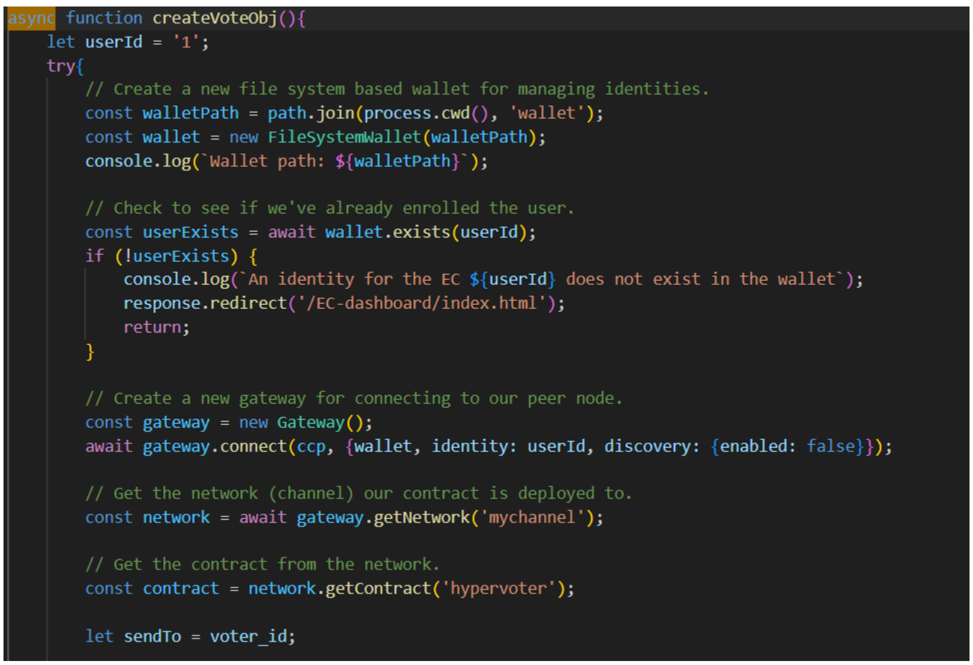
The voting procedure is designed to be straightforward and user-friendly while ensuring maximum security and transparency. Here is a step-by-step overview:

The election commission initializes a new voting session using the createVoteObj function, setting up the candidates and initiating vote counts. The end time for voting is set using the setEndTime function, establishing the time-bound period for voting. Voters, upon verification, receive their secure voting objects and are then eligible to vote. Voters cast their votes using the sendVoteObj function before the set end time. Each vote is securely recorded in the voting object, updating the vote counts while maintaining voter anonymity. Once the voting period ends, the getResults function is triggered. It calculates the results based on the final state of the voting object and immediately returns the election outcome.

5.3 Technologies Used

The application leverages several powerful technologies:

Hyperledger Fabric: This blockchain framework forms the core of the application, providing an infrastructure that enables confidentiality, scalability, and security. It allows the use of chaincodes (smart contracts) for business logic implementation and houses the ledger where all transactions (votes) are recorded [11]. Figure 7 shows Express.js application interacting with Hyperledger fabric.



```
async function createVoteObj(){
    let userId = '1';
    try{
        // Create a new file system based wallet for managing identities.
        const walletPath = path.join(process.cwd(), 'wallet');
        const wallet = new FileSystemWallet(walletPath);
        console.log(`Wallet path: ${walletPath}`);

        // Check to see if we've already enrolled the user.
        const userExists = await wallet.exists(userId);
        if (!userExists) {
            console.log(`An identity for the EC ${userId} does not exist in the wallet`);
            response.redirect('/EC-dashboard/index.html');
            return;
        }

        // Create a new gateway for connecting to our peer node.
        const gateway = new Gateway();
        await gateway.connect(ccp, {wallet, identity: userId, discovery: {enabled: false}});

        // Get the network (channel) our contract is deployed to.
        const network = await gateway.getNetwork('mychannel');

        // Get the contract from the network.
        const contract = network.getContract('hypervoter');

        let sendTo = voter_id;
```

Figure 7: Express.js Application Interacting with Hyperledger Fabric

Docker Containers: Docker ensures that the application runs smoothly across different computing environments [16]. Each component of the application (like the Hyperledger peers and ordering service) is packaged into a Docker container, ensuring easy deployment and scaling.

This Docker Compose snippet in Listing 1 illustrates the setup of a Certificate Authority (CA), crucial for managing network identities in a Hyperledger Fabric environment. The CA, defined in its Docker container, handles certificate issuance, validating and managing user identities, and thus playing a critical role in network security. The encapsulation of the CA service within a Docker container ensures a consistent, easily replicable environment, underlining Docker's utility in achieving smooth and efficient application deployment.

```
services:
  ca.example.com:
    image: hyperledger/fabric-ca
```

```
    environment:
      - FABRIC_CA_HOME=/etc/hyperledger/fabric-ca-server
      - FABRIC_CA_SERVER_CA_NAME=ca.example.com
      - FABRIC_CA_SERVER_CA_CERTFILE=/etc/hyperledger/fabric-ca-server-config/ca.org1.example.com-cert.pem
      - FABRIC_CA_SERVER_CA_KEYFILE=/etc/hyperledger/fabric-ca-server-config/4239aa0dcd76daeeb8ba0cda701851d14504d31aad1b2ddddbac6a57365e497c_sk
    ports:
      - "7054:7054"
    command: sh -c 'fabric-ca-server start -b admin:adminpw'
    volumes:
      - ./crypto-config/peerOrganizations/org1.example.com/ca/:/etc/hyperledger/fabric-ca-server-config
    container_name: ca.example.com
    networks:
      - basic

  orderer.example.com:
    container_name: orderer.example.com
    image: hyperledger/fabric-orderer
    environment:
      - FABRIC_LOGGING_SPEC=info
      - ORDERER_GENERAL_LISTENADDRESS=0.0.0.0
      - ORDERER_GENERAL_GENESISMETHOD=file
      - ORDERER_GENERAL_GENESISFILE=/etc/hyperledger/configtx/genesis.block
      - ORDERER_GENERAL_LOCALMSPID=OrdererMSP
      - ORDERER_GENERAL_LOCALMSPDIR=/etc/hyperledger/msp/orderer/msp
    working_dir: /opt/gopath/src/github.com/hyperledger/fabric/orderer
    command: orderer
    ports:
      - 7050:7050
    volumes:
      - ./config/:/etc/hyperledger/configtx
      - ./crypto-config/ordererOrganizations/example.com/orderers/orderer.example.com/:/etc/hyperledger/msp/orderer
      - ./crypto-config/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/:/etc/hyperledger/msp/peerOrg1
    networks:
      - basic

  peer0.org1.example.com:
    container_name: peer0.org1.example.com
    image: hyperledger/fabric-peer
    environment:
      - CORE_VM_ENDPOINT=unix:///host/var/run/docker.sock
      - CORE_PEER_ID=peer0.org1.example.com
      - FABRIC_LOGGING_SPEC=info
      - CORE_CHAINCODE_LOGGING_LEVEL=info
      - CORE_PEER_LOCALMSPID=Org1MSP
      - CORE_PEER_MSPCONFIGPATH=/etc/hyperledger/msp/peer/
```

```
      - CORE_PEER_ADDRESS=peer0.org1.example.com:7051
      - CORE_VM_DOCKER_HOSTCONFIG_NETWORKMODE=${COMPOSE_PROJECT_NAME}_basic
      - CORE_LEDGER_STATE_STATEDATABASE=CouchDB
      - CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb:5984
      - CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=
      - CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=
    working_dir: /opt/gopath/src/github.com/hyperledger/fabric
    command: peer node start
    ports:
      - 7051:7051
      - 7053:7053
    volumes:
      - /var/run/:/host/var/run/
      - ./crypto-config/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/msp:/etc/hyperledger/msp/peer
      - ./crypto-config/peerOrganizations/org1.example.com/users:/etc/hyperledger/msp/users
      - ./config:/etc/hyperledger/configtx
    depends_on:
      - orderer.example.com
      - couchdb
    networks:
      - basic

  couchdb:
    container_name: couchdb
    image: hyperledger/fabric-couchdb
    environment:
      - COUCHDB_USER=
      - COUCHDB_PASSWORD=
    ports:
      - 5984:5984
    networks:
      - basic

  cli:
    container_name: cli
    image: hyperledger/fabric-tools
    tty: true
    environment:
      - GOPATH=/opt/gopath
      - CORE_VM_ENDPOINT=unix:///host/var/run/docker.sock
      - FABRIC_LOGGING_SPEC=info
      - CORE_PEER_ID=cli
      - CORE_PEER_ADDRESS=peer0.org1.example.com:7051
      - CORE_PEER_LOCALMSPID=Org1MSP
      - CORE_PEER_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp
      - CORE_CHAINCODE_KEEPALIVE=10
    working_dir: /opt/gopath/src/github.com/hyperledger/fabric/peer
    command: /bin/bash
```

```
    volumes:
      - /var/run/:/host/var/run/
      - ./../chaincode/:/opt/gopath/src/github.com/
      - ./crypto-config:/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/
    networks:
      - basic
```

Listing 1: Docker Compose File for Hyperledger Fabric Network

CouchDB: This NoSQL database, when used as a state database (world state) with Hyperledger Fabric, allows for rich queries against the JSON voting data, providing efficient and flexible data handling capabilities.

Node.js and Express.js: These JavaScript runtime and library are used for building the server-side of the application. They handle requests from the client-side, interacting with the chaincode and the ledger, and serve responses back to the client.

Listing 2 showcases a Node.js script that interacts with a Hyperledger Fabric network. The script specifically interacts with the network's Certificate Authority (CA) to enroll an 'admin' user.

The code leverages the Hyperledger Fabric SDK for Node.js to create a new CA client and a new file system-based wallet for identity management. It then checks if an 'admin' user already exists in the wallet. If not, it enrolls a new 'admin' user with the CA and imports the new identity into the wallet

```
async function createVoteObj() {
    let userId = '1';
    try {
        // Create a new file system based wallet for managing identities.
        const walletPath = path.join(process.cwd(), 'wallet');
        const wallet = new FileSystemWallet(walletPath);
        console.log(`Wallet path: ${walletPath}`);

        // Check to see if we've already enrolled the user.
        const userExists = await wallet.exists(userId);
        if (!userExists) {
            console.log(`An identity for the EC ${userId} does not exist in the wallet`);
            response.redirect('/EC-dashboard/index.html');
            return;
        }

        // Create a new gateway for connecting to our peer node.
        const gateway = new Gateway();
        await gateway.connect(ccp, { wallet, identity: userId, discovery: { enabled: false } });

        // Get the network (channel) our contract is deployed to.
        const network = await gateway.getNetwork('mychannel');
```

```
        // Get the contract from the network.
        const contract = network.getContract('hypervoter');

        let sendTo = voter_id;
        await contract.submitTransaction('createVoteObj', sendTo);
        console.log("Added Voter and created VOTE Obj successfully!\n");
        await gateway.disconnect();
    } catch (error) {
        console.log(`Failed to submit transaction: ${error}`);
        //response.redirect('/EC-dashboard/ec-dashboard.html');
        process.exit(1);
    }
}
createVoteObj();
response.redirect('/EC-dashboard/ec-dashboard-add-voter.html');

app.get('/EC-dashboard/ec-dashboard-add-candidate.html', function(request, response) {
    response.sendFile(path.join(__dirname + '/public/EC-dashboard/ec-dashboard-add-candidate.html'));
});

app.get('/EC-dashboard/ec-set-time.html', function(request, response) {
    response.sendFile(path.join(__dirname + '/public/EC-dashboard/ec-set-time.html'));
});

let et = "Not Yet Set";
app.post('/EC-dashboard/set-time', function(request, response) {
    var duration_d = request.body.duration_d;
    var duration_h = request.body.duration_h;
    var duration_m = request.body.duration_m;

    let duration = ((parseInt(duration_d) * 24 * 60) + (parseInt(duration_h) * 60) + parseInt(duration_m)).toString();
    let timestr;

    async function setTime() {
        let userId = '1';
        try {
            // Create a new file system based wallet for managing identities.
            const walletPath = path.join(process.cwd(), 'wallet');
            const wallet = new FileSystemWallet(walletPath);
            console.log(`Wallet path: ${walletPath}`);

            // Check to see if we've already enrolled the user.
            const userExists = await wallet.exists(userId);
            if (!userExists) {
                console.log(`An identity for the EC ${userId} does
```

```
not exist in the wallet`);
                response.redirect('/EC-dashboard/index.html');
                return;
            }

            // Create a new gateway for connecting to our peer node.
            const gateway = new Gateway();
            await gateway.connect(ccp, { wallet, identity: userId, discovery: { enabled: false } });

            // Get the network (channel) our contract is deployed to.
            const network = await gateway.getNetwork('mychannel');

            // Get the contract from the network.
            const contract = network.getContract('hypervoter');

            timestr = await contract.submitTransaction('setEndTime', duration);
            timestr = new Date(timestr);
            et = timestr;
            console.log(`Setting End Time To: ${timestr}`);
            console.log("Voting End Time set Successfully!\n");
            let outString = "Voting End Time Set to: " + timestr;
            response.render(__dirname + "/public/EC-dashboard/ec-set-time.html", { _: outString });

            await gateway.disconnect();
        } catch (error) {
            console.log(`Failed to submit transaction: ${error}`);
            process.exit(1);
        }
    }
    setTime();
});
```

Listing 2: Enrolling an Admin User on a Hyperledger Fabric Network using Node.js

HTML5, CSS3, and JavaScript: These technologies form the backbone of the front-end. HTML5 structures the content, CSS3 styles the interface, and JavaScript ensures interactivity. They make the application interface intuitive and easy-to-use, making voting a breeze for voters.

The given HTML snippet in Listing 3 corresponds to the main landing page of the Votzo e-voting system. This page features a clear, navigable layout where users can easily access various system features. The central logo adds a visual identity to the system. With a straightforward navigation bar at the top and a list of accessible dashboards, the interface facilitates quick transitions between different aspects of the voting process. An additional feature displayed is the 'Election End Time', keeping users informed about the election timeline.

```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Votzo</title>

    <!-- External Scripts and CSS -->
    <script src="http://ajax.googleapis.com/ajax/libs/jquery/2.0.0/jquery.min.js"></script>
    <script src="https://apis.google.com/js/client.js?onload=onClientLoad" type="text/javascript"></script>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">

    <!-- Internal Styles -->
    <style>
        body {
            background-color: rgb(210, 220, 223);
        }

        .center {
            display: block;
            margin-left: auto;
            margin-right: auto;
            margin-bottom: 30px;
            width: 50%;
        }
    </style>

</head>

<body>

    <!-- Navbar Section -->
    <nav class="navbar navbar-default">
        <div class="container-fluid">
            <div class="navbar-header">
                <a class="navbar-brand" href="#">Votzo</a>
            </div>
            <ul class="nav navbar-nav">
                <li class="active"><a href="#">Dashboard Links</a></li>
                <li><a href="./about.html">About</a></li>
            </ul>
        </div>
    </nav>

    <!-- Main Content Section -->
    <div class="container-fluid">
```

```

        <!-- Logo Section -->
        <img src="votzo_logo.jpg" style="width:200px;height:120px;" class="center">

        <!-- Dashboard Links -->
        <div class="list-group">
            <a href="./EC-dashboard/index.html" target="_self" class="list-group-item">EC Dashboard</a>
            <a href="./voter-dashboard/index.html" target="_self" class="list-group-item">Voter Dashboard</a>
            <a href="./public-bulletin/index.html" target="_self" class="list-group-item">Public Bulletin</a>
        </div>

        <!-- Election Time Display -->
        <p>Election End Time: <%- _ %></p>

    </div>

</body>

</html>
```

Listing 3: Main Dashboard Interface of the Votzo E-Voting System

In essence, each of these technologies plays a critical role, coming together to create a secure, efficient, and user-friendly e-voting system.

5.4 Conclusion and Future Works

This project taught me much about blockchain, smart contracts, and secure, transparent voting systems. This initiative offered a great chance to study blockchain technology's practical applications beyond cryptocurrency. Hyperledger Fabric, Docker containers, and other cutting-edge technologies created an electronic voting system. The project was about creating a new e-voting system.

The initial study topics investigated blockchain technology's promising use in electoral systems. These inquiries influenced my method selection, result interpretation, and literature review. I reviewed voting system articles that used blockchain technology. I also studied this technology's voting benefits and drawbacks. My findings strongly supported this strategy, showing that the blockchain may provide a trustworthy, transparent, and secure foundation for electronic voting systems. I utilized createVoteObj, setEndTime, sendVoteObj, and getResults to make voting seamless. These functions each provided a unique contribution.

As shown by the real-world execution of this project, blockchain technology will assist in constructing an electoral system. Based on these interesting findings, blockchain technology can significantly improve electronic voting system security, efficiency, and transparency when used appropriately.

Docker containers can manage and reproduce complex programs effortlessly, improving our system's efficacy and adaptability. We learned to use Node.js, Express.js, HTML5, CSS3, and JavaScript to construct an interactive and user-friendly voting experience.

This initiative helped me understand blockchain technology and its exciting real-world applications. It stressed the need to synchronize research topics, literature reviews, techniques, and outcomes to build a thorough and informative project that offers progress and success. This project's theoretical and practical understanding showed blockchain technology's enormous potential. The blockchain can transform voting, government, and digital currencies, creating intriguing future possibilities.

5.5 Contributions

My journey began with the pioneering design and construction of a blockchain-based voting system utilizing Hyperledger Fabric, demonstrating its potential to improve modern democracy. I introduced a prototype that assures each vote's confidentiality and resistance against threats using Hyperledger Fabric's inherent security capabilities.

I designed a system that smoothly integrates endorsing peers, orderers, and certifying authorities. This architecture streamlines transaction processing and preserves vote chronological integrity. I used HTML5, CSS3, and JavaScript to create an intuitive user interface to make system use easy. Identity assurance was another key contribution. I created a core identity management platform that prioritized voter authentication and anonymity. I also examined the complex legal framework of blockchain-based voting, highlighting its real-world ramifications and importance.

The 'Future Works' section describes future prospects and challenges. I always strive to improve democratic processes' transparency, efficiency, and accessibility.

5.6 Future Works

As I developed a blockchain-based voting system using Hyperledger Fabric, I lay the framework for something that could transform democratic elections. Although my effort has been useful, I know the road ahead is long and there is much more to investigate and perfect. The system's performance, accessibility, and resilience may improve with future opportunities and challenges. I will mention future research and enhancements that could improve the system in this part. I'd also like to reflect on my important contributions to this endeavor.

- **Exploring consensus algorithms**: Hyperledger Fabric's normal consensus mechanism was employed. Other consensus methods may be employed to test their efficacy and safety in vote systems.

  Adding more advanced cryptography techniques, such as zero-knowledge proofs or homomorphic encryption, could improve security.

- **Scalability and performance testing**: Run many performance and scalability tests to find out where the existing system fails and what changes may be needed for larger voting situations.

- **Building a user-friendly interface**: The existing project employed HTML5, CSS3, and JavaScript, although future work could simplify the interface. React or Angular could enhance the experience.

- **Strong identity management**: Identity management should be more comprehensive. This ensures that only qualified voters can vote anonymously.

- **Adding post-election audits**: Future development could incorporate post-election audits to make the system more trustworthy and open.

  Understanding blockchain-based voting system legalities is crucial for real-world applications.

- **Integration with existing voter registration systems**: Future research can determine how to effectively connect blockchain-based voting systems with voter registration databases.

- **Analysis of environmental impact:** Concerns regarding blockchain technologies' environmental impact are developing, thus a study can compare blockchain-based voting systems to regular ones.

References

[1] João Alves and António Pinto. On the use of the blockchain technology in electronic voting systems. In *Ambient Intelligence–Software and Applications–, 9th International Symposium on Ambient Intelligence*, pages 323–330. Springer, 2019.

[2] Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, Srinivasan Muralidharan, Chet Murthy, Binh Nguyen, Manish Sethi, Gari Singh, Keith Smith, Alessandro Sorniotti, Chrysoula Stathakopoulou, Marko Vukolić, Sharon Weed Cocco, and Jason Yellick. Hyperledger fabric: A distributed operating system for permissioned blockchains. In *Proceedings of the Thirteenth EuroSys Conference*, EuroSys '18, New York, NY, USA, 2018. Association for Computing Machinery.

[3] A. V. Aswin and Bineeth Kuriakose. An analogical study of hyperledger fabric and ethereum. In S. Balaji, Álvaro Rocha, and Yi-Nan Chung, editors, *Intelligent Communication Technologies and Virtual Mobile Networks*, pages 412–420, Cham, 2020. Springer International Publishing.

[4] Chuan Chen, Jiajing Wu, Hui Lin, Wuhui Chen, and Zibin Zheng. A secure and efficient blockchain-based data trading approach for internet of vehicles. *IEEE Transactions on Vehicular Technology*, 68(9):9110–9121, 2019.

[5] Ross Clarke, Luke McGuire, Mohamed Baza, Amar Rasheed, and Maazen Alsabaan. Online voting scheme using ibm cloud-based hyperledger fabric with privacy-preservation. *Applied Sciences*, 13(13), 2023.

[6] Gaby G. Dagher, Jordan Mohler, Matea Milojkovic, and Praneeth Babu Marella. Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology. *Sustainable Cities and Society*, 39:283–297, 2018.

[7] Julian Dreyer, Marten Fischer, and Ralf Tönjes. Performance analysis of hyperledger fabric 2.0 blockchain platform. In *Proceedings of the Workshop on Cloud Continuum Services for Smart IoT Systems*, CCIoT '20, page 32–38, New York, NY, USA, 2020. Association for Computing Machinery.

[8] Vaidotas Drungilas, Evaldas Vaičiukynas, Mantas Jurgelaitis, Rita Butkienė, and Lina Čeponienė. Towards blockchain-based federated machine learning: Smart contract for model inference. *Applied Sciences*, 11(3), 2021.

[9] Javier Díaz-Santiso and Paula Fraga-Lamas. E-voting system using hyperledger fabric blockchain and smart contracts. *Engineering Proceedings*, 7(1), 2021.

[10] Denis Kirillov, Vladimir Korkhov, Vadim Petrunin, and Mikhail Makarov. Performance of the secret electronic voting scheme using hyperledger fabric per-

missioned blockchain. In Osvaldo Gervasi, Beniamino Murgante, Sanjay Misra, Chiara Garau, Ivan Blečić, David Taniar, Bernady O. Apduhan, Ana Maria A. C. Rocha, Eufemia Tarantino, Carmelo Maria Torre, and Yeliz Karaca, editors, *Computational Science and Its Applications – ICCSA 2020*, pages 25–36, Cham, 2020. Springer International Publishing.

[11] Sergey Kyazhin and Vladimir Popov. Yet another e-voting scheme implemented using hyperledger fabric blockchain. In Osvaldo Gervasi, Beniamino Murgante, Sanjay Misra, Chiara Garau, Ivan Blečić, David Taniar, Bernady O. Apduhan, Ana Maria A. C. Rocha, Eufemia Tarantino, Carmelo Maria Torre, and Yeliz Karaca, editors, *Computational Science and Its Applications – ICCSA 2020*, pages 37–47, Cham, 2020. Springer International Publishing.

[12] Yacov Manevich, Artem Barger, and Gal Assa. Redacting transactions from execute-order-validate blockchains. In *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–9. IEEE, 2021.

[13] Prodipta Promit Mukherjee, Arika Afrin Boshra, Mallik Mohammad Ashraf, and Milon Biswas. A hyper-ledger fabric framework as a service for improved quality e-voting system. In *2020 IEEE Region 10 Symposium (TENSYMP)*, pages 394–397, 2020.

[14] Mohammed Khaled Mustafa and Sajjad Waheed. An e-voting framework with enterprise blockchain. In Asis Kumar Tripathy, Mahasweta Sarkar, Jyoti Prakash Sahoo, Kuan-Ching Li, and Suchismita Chinara, editors, *Advances in Distributed Computing and Machine Learning*, pages 135–145, Singapore, 2021. Springer Singapore.

[15] Aneta Poniszewska-Marańda, Stanislaw Rojek, and Michal Pawlak. Decentralized electronic voting system using hyperledger fabric. In *2022 IEEE International Conference on Services Computing (SCC)*, pages 339–348, 2022.

[16] Amit M Potdar, DG Narayan, Shivaraj Kengond, and Mohammed Moin Mulla. Performance evaluation of docker container and virtual machine. *Procedia Computer Science*, 171:1419–1428, 2020.

[17] Shatha H Saeed, Suha M Hadi, and Ali H Hamad. Performance evaluation of e-voting based on hyperledger fabric blockchain platform. *Revue d'Intelligence Artificielle*, 36(4), 2022.

[18] Ruhi Taş and Ömer Özgür Tanrıöver. A systematic review of challenges and opportunities of blockchain for e-voting. *Symmetry*, 12(8):1328, 2020.

[19] Shan Wang, Ming Yang, Yue Zhang, Yan Luo, Tingjian Ge, Xinwen Fu, and Wei Zhao. On private data collection of hyperledger fabric. In *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, pages 819–829. IEEE, 2021.

[20] Bin Yu, Joseph K. Liu, Amin Sakzad, Surya Nepal, Ron Steinfeld, Paul Rimba, and Man Ho Au. Platform-independent secure blockchain-based voting system. In Liqun Chen, Mark Manulis, and Steve Schneider, editors, *Information Security*, pages 369–386, Cham, 2018. Springer International Publishing.