SAVITRIBAI PHULE PUNE UNIVERSITY

A PRELIMINARY PROJECT REPORT ON

“DECENTRALIZED AND SECURE VOTING SYSTEM USING BLOCKCHAIN TECHNOLOGY”

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)

BY

ARVIND SUDARSHAN	SEAT NO: B190214209
CHATANE SHREE ATUL	SEAT NO: B190214228
EKSAMBEKAR YASH SAGAR	SEAT NO: B190214241
GADKARI GAURAV SUDHIR	SEAT NO: B190214244

Under the guidance of

Prof. Snehal S. Kolte

AISSMS
COLLEGE OF ENGINEERING
Approved by AICTE, New Delhi, Recognized by
Govt. of Maharashtra, Affiliated to Savitribai Phule Pune University
and recognized 2(f) and 12(B) by UGC (Id.No. PU / PN/ Engg. / 093 (1992)
Accredited by NAAC with ‘A+’ Grade

No. 1, Kennedy Road, Near RTO Office Sangamvadi, Shivajinagar,
Pune - 411001

AISSMS
COLLEGE OF ENGINEERING
Approved by AICTE, New Delhi, Recognized by
Govt. of Maharashtra, Affiliated to Savitribai Phule Pune University
and recognized 2(f) and 12(B) by UGC (Id.No. PU / PN/ Engg. / 093 (1992)
Accredited by NAAC with 'A+' Grade

CERTIFICATE

This is to certify that the project report entitles

"DECENTRALIZED AND SECURE VOTING SYSTEM USING BLOCKCHAIN TECHNOLOGY"

Submitted by

ARVIND SUDARSHAN	SEAT NO: B190214209
CHATANE SHREE ATUL	SEAT NO: B190214228
EKSAMBEKAR YASH SAGAR	SEAT NO: B190214241
GADKARI GAURAV SUDHIR	SEAT NO: B190214244

is a bonafide work carried out by them under the supervision of Prof. Snehal S. Kolte and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University Pune for the award of the degree of Bachelor of Engineering (Computer Engineering). This project work has not been earlier submitted to any other Institute or University for the award of any degree or diploma.

Prof. Snehal S. Kolte Project Guide	Dr. S. V. Athawale H.O.D. Computer Department	Prof. V. V. Navale Project Coordinator

Dr. D.S. Bormane
Principal
AISSMS College of Engineering

Place: Pune
Date:

PROJECT APPROVAL SHEET

A

Project

on

“DECENTRALIZED AND SECURE VOTING SYSTEM USING BLOCKCHAIN TECHNOLOGY”

Is successfully completed by

ARVIND SUDARSHAN	SEAT NO: B190214209
CHATANE SHREE ATUL	SEAT NO: B190214228
EKSAMBEKAR YASH SAGAR	SEAT NO: B190214241
GADKARI GAURAV SUDHIR	SEAT NO: B190214244

at

Department of Computer Engineering
AISSMS College of Engineering, Pune
Savitribai Phule Pune University
2022-2023

Prof. Snehal S. Kolte

Project Guide

Dr. S V Athawale
H.O.D.
Department of Computer Engineering

ACKNOWLEDGEMENT

We would like to extend my sincere gratitude and thanks to my guide Prof. Snehal S. Kolte, for her invaluable guidance and for giving us useful inputs and encouragement time and again, which inspired us to work harder. Due to her forethought, appreciation of the work involved and continuous imparting of useful tips, this report has been successfully completed.

We are extremely grateful to Dr. S V Athawale, Head of the Department of Computer Engineering, for his encouragement during the course of the project work.

We also extend our heartfelt gratitude to the staff of the Computer Engineering Department for their cooperation and support.

We also take this opportunity to thank all our classmates, friends and all those who have directly or indirectly provided their overwhelming support during our project work and the development of this report.

Arvind Sudarshan
Chatane Shree
Eksambekar Yash
Gadkari Gaurav

ABSTRACT

Voting in democratic country is a fundamental right granted to every eligible individual by the constitution. Current e-Voting system used isn't transparent and can be improved in a few aspects. All voting data from Electronic Voting Machines (EVMs) are stored on a central server. This creates a single point of failure which can be exploited and tampered with easily. Such flaws cause mistrust in the electoral process. Blockchain is a shared immutable ledger that facilitates the process of recording transactions in a network. It is an emerging technology whose full potential is yet to be realized. Blockchain became popular in 2009 when bitcoin was introduced and used as an alternative to tangible currency and has evolved since. It is a reliable system that can be used in various critical industrial applications. Blockchain has potential to improve the voting system to contest transparent and fair voting. Using this modern technology, a voting system can be implemented which provides transparency leading to fairness in the system. Furthermore, this will overcome the current system flaw of having a single point of failure caused by storing data in a centralized server. In addition to this, election results can be declared faster compared to the current system which might take a few days. The proposed system in this paper shows implementation of voting using blockchain technology.

Contents

List of Figures

List of Tables

CHAPTER 1
SYNOPSIS

1.1 Project Title

DECENTRALIZED AND SECURE VOTING SYSTEM USING BLOCKCHAIN TECHNOLOGY

1.2 Project Option

Sponsored

1.3 Internal Guide

Prof. Snehal S. Kolte

1.4 Sponsorship and External Guide

Elite Softwares

1.5 Problem Statement

Create a private network responsible for storing and maintaining blockchain. Upon this network, develop a decentralized blockchain based Voting System system based on smart contract which is the optimum solution for frauds, forgery, human error, traceability, stagnancy occurring in the current system providing enhanced security, privacy, efficiency and speed in the voting process.

1.6 Abstract

Voting in democratic country is a fundamental right granted to every eligible individual by the constitution. Current e-Voting system used isn't transparent and can be improved in a few aspects. All voting data from Electronic Voting Machines (EVMs) are stored on a central server. This creates a single point of failure which can be exploited and tampered with easily. Such flaws cause mistrust in the electoral process. Blockchain is a shared immutable ledger that facilitates the process of recording transactions in a network. It is an emerging technology whose full potential is yet to be realized. Blockchain became popular in 2009 when bitcoin was introduced and used as an alternative to tangible currency and has evolved since. It is a reliable system that can be used in various critical industrial applications. Blockchain has potential to improve the voting system to contest transparent and fair voting. Using this modern technology, a voting system can be implemented which provides transparency leading to fairness in the system. Furthermore, this will overcome the current system flaw of having a single point of failure caused by storing data in a centralized server. In addition to this, election results can be declared faster compared to the current system which might take a few days. The proposed system in this paper shows implementation of voting using blockchain technology.

1.7 Goals and Objectives

To design a decentralized e-voting system that,

1. Eliminate humans from the process.

2. Does not depend on a trusted third party for controlling the collected data, whilst still staying immune to attacks and guaranteeing user privacy.

3. Provides transparency, ensuring security and does not jeopardize voter privacy.

1.8 Plan of Project Execution

	June	July	August	September	October	November	December	January	February	March	April	May
Project scouting	🟨											
Feasibility Study		🟨										
Literature Survey			🟨	🟨								
Project definition				🟨								
Project interface definition					🟨							
Requirement gathering					🟨							
Documentation					🟨	🟨						
Phase 1 implementation							🟨	🟨				
Phase 2 implementation								🟨	🟨			
Journal paper publication									🟨	🟨	🟨	
Project report finalizing											🟨	🟨
Final project implementation											🟨	🟨

Table 1.1: Timeline Chart

CHAPTER 2
TECHNICAL KEYWORDS

2.1 Area of Project

1. Blockchain Technology
2. Secured Voting

2.2 Technical Keywords

1. Blockchain Technology
2. Ethereum Virtual Machine
3. Voting System
4. Secure Voting
5. Central Server
6. Transparency
7. Single Point of Failure

CHAPTER 3
INTRODUCTION

3.1 Overview

Blockchain technology can be used in the E-voting system to conduct a fair election and reduce injustice. The physical voting systems have many flaws in it as well as the digital voting systems are not perfect enough to be implemented on a large scale. This paper presents an overview of Blockchain based voting systems. The proposed platform will provide a framework that can be implemented to conduct voting activity digitally through blockchain. Our proposed system will use a flexible blockchain with consensus algorithms. The Chain security algorithm used in makes voting transactions more secure and integrated.

Blockchain: Blockchain is like a distributed ledger technology consisting of records of transactions called as blocks that are linked together using mathematical cryptography. Blockchain technology is popular because of its distributed nature which makes it immutable and practically impossible to hack.

Distributed Ledger: Ledger is a records collection containing information about transactions. Distributed ledger is a data structure capable of storing said transactions which is distributed across different computers on the network. DLT (Distributed Ledger Technology) is technology that distributes transaction records to all the users participating in the network. Blockchain is a type of Distributed Ledger Technology(DLT). Hence the data is shared among all its users providing transparency and preventing corruption.

Decentralized Application(dApp): dApp is an application built to run on decentralized computing, blockchain or any other distributed ledger system. It makes use of smart contracts for its functioning.

Smart Contract: Smart contracts are computer programs that automatically execute or control actions according to the terms of a contract or an agreement. It is used to enforce rules in blockchain transactions. Smart contracts are considered fundamental building blocks for cryptocurrencies and NFTs.

Consensus: Consensus ensures that all the different users participating in blockchain come to a mutual agreement regarding the state of blockchain. There are numerous consensus mechanisms that are used by different blockchain applications.

Cryptography: Cryptography is based on mathematical theory and computer science. Cryptography's importance is to provide methods to secure and protect data and communications using encryption related techniques.

Ethereum: Ethereum is an open-source blockchain which provides smart contract functionality using solidity programming language.

3.2 Motivation

India is said to have the largest democracy. For the electoral voting system India is using modern "Electronic Voting Machines" which are responsible for taking user input and storing the votes entered by the end user. Later these votes are carried to a central location where they are counted and the result is declared. Though, there is a possibility that these votes data can be tampered with. Having a single central server to store data creates a single point of failure for the entire system. These issues cause mistrust regarding the current system. In the traditional ballot paper system, vote authenticity was questionable and had its separate issues for which it was replaced with modern Electronic Voting Machines. Blockchain is based on Distributed Ledger Technology(DLT). In DLT, replicas of the same data are made on different nodes in the network due to which there is no single point of failure. Moreover, due to consensus algorithms implemented in blockchain, all nodes decide and agree upon whether a block is valid and must be added to the network or not.

Using Blockchain Technology will overcome issues in the present electronic voting system. It will also contribute to increasing the overall trust in the system and benefit democracy. Blockchain will provide a secure system for vote acceptance, storage and provide faster election results.

3.3 Objective

To design a decentralized e-voting system that,

1. Reduces human interaction in the process of voting and elections in general.
2. Does not depend on a trusted third party for controlling the collected data, whilst still staying immune to attacks and guaranteeing user privacy.
3. Provides transparency, ensuring security and does not jeopardize voter privacy.

CHAPTER 4
LITERATURE SURVEY

4.1 Study of Research Paper

1. **Paper Name:** Blockchain challenges and opportunities: a survey [1]
   Authors: Zibin Zheng, Shaoan Xie, Hong-Ning Dai, Xiangping Chen and Huaimin Wang
   Abstract: Blockchain has numerous benefits such as decentralisation, persistency, anonymity and auditability. There is a wide spectrum of blockchain applications ranging from cryptocurrency, financial services, risk management, internet of things (IoT) to public and social services. Although a number of studies focus on using the blockchain technology in various application aspects, there is no comprehensive survey on the blockchain technology in both technological and application perspectives. To fill this gap, we conduct a comprehensive survey on the blockchain technology. In particular, this paper gives the blockchain taxonomy, introduces typical blockchain consensus algorithms, reviews blockchain applications and discusses technical challenges as well as recent advances in tackling the challenges. Moreover, this paper also points out the future directions in the blockchain technology.
   Inference:
   Blockchain gained most of its fame from bitcoin, a cryptocurrency.
   It is the backbone of bitcoin. The concept of blockchain was introduced in 2008 and implemented a year later. Satoshi Nakamoto is dubbed as the creator of blockchain. Blockchain has several different characteristics like decentralisation, persistence, anonymity and auditability. Blockchain has various diverse applications other than bitcoin.Its ability to conduct transactions without banks make it a strong contender for online payments.

2. **Paper Name:** A Survey on Blockchain Technology: Evolution, Architecture and Security [2]
   Authors: Bhutta, Muhammad Nasir Mumtaz, Amir A. Khwaja, Adnan Nadeem, Hafiz Farooq Ahmad, Muhammad Khurram Khan, Moataz A. Hanif, Houbing Song, Majed Alshamari, and Yue Cao
   Abstract: Blockchain is a revolutionary technology that is making a great impact on modern society due to its transparency, decentralization, and security properties. Blockchain gained considerable attention due to its very first application of Cryptocurrencies e.g., Bitcoin. In the near future, Blockchain technology is determined to transform the way we live, interact, and perform businesses. Recently, academics, industrialists, and researchers are aggressively investigating different aspects of Blockchain as an emerging technology. Unlike other Blockchain surveys focusing on either its applications, challenges, characteristics, or security, we present a comprehensive survey of Blockchain technology's evolution, architecture, development frameworks, and security issues. We also present a comparative analysis of frameworks, classification of consensus algorithms, and analysis of security risks & cryptographic primitives that have been used in the Blockchain so far. Finally, this paper elaborates on key future directions, novel use cases and open research challenges, which could be explored by researchers to make further advances in this field.
   Inference:

This paper covers the Blockchain technology's evolution, its security and architecture.
Blockchain 1.0 is used in bitcoin founded by anonymous person with pseudoname Satoshi Nakamoto. Cryptocurrencies are the first applications of Blockchain technology and are already functional as a digital payment alternative on the World Wide Web.
Blockchain 2.0 is based on Smart Contracts. It is used for transferring assets like bonds, stocks, loans, Properties etc. Afterwards it is realised that that Blockchain can revolutionise for all the industries.
Blockchain 3.0 provides a platform for development of secure applications for all the industries rather than only money exchange. It supports large scale interconnection using web technology.

3. **Paper Name:** Blockchain characteristics and consensus in modern business processes [3]
Authors: Viriyasitavat, Wattana and Danupol Hoonsopon
Abstract: Blockchain technology has attracted a great deal of attentions as an effective way to innovate business processes. It has to be integrated with other Business Process Management system (BPM) components to implement specified functionalities related to the applications. The current efforts in integrating this technology into BPM are at a very early stage. To apply Blockchain into business processes efficiently, Blockchain and business process characteristics must be identified. Inconsistency of confirmation settlement that heavily relies on the implementation of consensus protocol poses a major challenge in business process operations, especially ones that are time-critical. In addition, validators, nodes responsible for performing consensus operations in a Blockchain system, can introduce bias and as a result are not trustable. This paper first defines Blockchain and also investigates the characteristics of Blockchain and business processes. Then, we suggest an architecture of business processes in Blockchain era to overcome the problems of time inconsistency and consensus bias. The architecture provides persistency, validity, auditability, and disintermediary that Blockchain offers. The architecture also provides flexibility by allowing business partner to select nodes in performing consensus; thus bias is mitigated.
Inference:

- **Cost effective:** In traditional systems there is a need of some central organizaion to verify the validity of transactions. Blockchain is a peer to peer network(P2P). The absence of a central agency reduces the cost per transaction.
- **Persistence:** Blockchain has one important property of persistence, Each block has the ability to maintain its records. This helps in keeping the data tamper proof.
- **Validity:** The execution is not carried out every time. This system has three roles, proposer, acceptor, learner.
- **Anonymity & Identity:** The centralised systems require to know you as

a person. You need to register your identity proof with those authorities. Whereas with blockchain, the user data remains fairly anonymous.

- **Auditability:** The block added to the blockchain remains there forever. This helps in checking transaction history. In private blockchain the auditability is least and depends on the central entity. In permissioned blockchain there is a little bit of auditability. In public blockchain have the most auditability.

4. **Paper Name:** An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends [4]
   Authors: Zheng, Zibin and Xie, Shaoan and Dai, Hong-Ning and Chen, Xiangping and Wang, Huaimin
   Abstract: Blockchain, the foundation of Bitcoin, has received extensive attentions recently. Blockchain serves as an immutable ledger which allows transactions take place in a decentralized manner. Blockchain-based applications are springing up, covering numerous fields including financial services, reputation system and Internet of Things (IoT), and so on. However, there are still many challenges of blockchain technology such as scalability and security problems waiting to be overcome. This paper presents a comprehensive overview on blockchain technology. We provide an overview of blockchain architechture firstly and compare some typical consensus algorithms used in different blockchains. Furthermore, technical challenges and recent advances are briefly listed. We also lay out possible future trends for blockchain.
   Inference:
   In Blockchain Technology, there exists three major types: Privete, Consortium and Public blockchain. Since a network will have to be created, a comparison must be made on what kind of network must be used for the same. The types are compared on the basis of Consensus determination, Read permission, Immutability, Efficiency, whether it is centralised or not and the Consensus process. Depending upon the application, a developer must decide what kind of blockchain should be used. Table 4.1 shows a comparision between the types of blockchains.

Property	**Types of Blockchain**		
	Private	**Consortium**	**Public**
Consensus determination	Single Organisation	Selected nodes	All participating miners
Read permission	Could be public or restricted	Could be public or restricted	Public
Immutability	Could be tampered	Could be tampered	Nearly impossible to tamper
Efficiency	High	High	Low
Centralised	Yes	Partial	No
Consensus process	Permissioned	Permissioned	Permissionless

Table 4.1: Types of Blockchains

5. **Paper Name:** Comparative analysis of blockchain consensus algorithms [5]
   Authors: L. M. Bach, B. Mihaljevic and M. Zagar
   Abstract: Cryptocurrencies have seen a massive surge in popularity and behind these new virtual currencies is an innovative technology called the blockchain: a distributed digital ledger in which cryptocurrency transactions are recorded after having been verified. The transactions within a ledger are verified by multiple

clients or "validators," within the cryptocurrency's peer-to-peer network using one of many varied consensus algorithms for resolving the problem of reliability in a network involving multiple unreliable nodes. The most widely used consensus algorithms are the Proof of Work (PoW) algorithm and the Proof of Stake (PoS) algorithm; however, there are also other consensus algorithms which utilize alternative implementations of PoW and PoS, as well as other hybrid implementations and some altogether new consensus strategies. In this paper, we perform a comparative analysis of typical consensus algorithms and some of their contemporaries that are currently in use in modern blockchains. Our analysis focuses on the algorithmic steps taken by each consensus algorithm, the scalability of the algorithm, the method the algorithm rewards validators for their time spent verifying blocks, and the security risks present within the algorithm. Finally, we present our conclusion and some possible future trends for consensus algorithms used in blockchains.

Inference:

The Oxford Dictionary meaning of word consensus is: an opinion that all members of a group agree with. In Blockchain Technology every transaction must be verified and agreed upon by validators in the network before being inserted. To do so, various consensus algorithms have been defined for use. Few of the most used consensus algorithms are Proof of Stake (PoS) algorithm and Proof of Work (PoW) algorithm.

In this paper referenced, the authors have mentioned a few of the consensus algorithms used and their comparative analysis. Some High-profile consensus algorithms mentioned in the paper are Proof of Work (PoW), Proof of Stake (PoS), Proof of Importance (PoI), Delegated Proof of Stake (dPOS), Stellar Consensus Protocol (SCP) and Ripple Protocol Consensus Algorithm (RPCA). The authors have compared these algorithms on the basis of Security, Scalability and Power Consumption. A consensus algorithm characteristic comparison is provided in Table 4.2.

Property	**Algorithm Name**					
	PoW	**RPCA**	**PoS**	**SCP**	**dPOS**	**PoI**
Energy Saving	No	Yes	Partial	Yes	Partial	Yes
Tolerated power of adversary	<25% computing power	<20% faulty nodes	<51% stake	Variable	<51% validators	<50% importance
Example	Bitcoin	Ripple	Cardano	Stellar	EOS	NEM

Table 4.2: Consensus Algorithm Comparision

6. **Paper Name:** Solutions to Scalability of Blockchain: A Survey [6]
   Authors: Q. Zhou, H. Huang, Z. Zheng and J. Bian
   Abstract: Blockchain-based decentralized cryptocurrencies have drawn much attention and been widely-deployed in recent years. Bitcoin, the first application of blockchain, achieves great success and promotes more development in this field. However, Bitcoin encounters performance problems of low throughput and high transaction latency. Other cryptocurrencies based on proof-of-work also inherit the flaws, leading to more concerns about the scalability of blockchain. This paper

attempts to cover the existing scaling solutions for blockchain and classify them by level. In addition, we make comparisons between different methods and list some potential directions for solving the scalability problem of blockchain.

Inference:

- **Bitcoin-Cash:** It is another form of cryptocurrency. It is the hard fork from bitcoin's codebase. The main idea is to solve the scalability issue of the blockchain. The solution here is to increase the block size while keeping the block interval the same. The block size of bitcoin was 1 MB, but now bitcoin and bitcoin cash has increased its blocksize to 8 MB. After that there was further increase in the block size of bitcoin cash to 32 MB.However we cannot increase the blocksize infinitely as there is a limitation to the intra-blockchain bandwidth. The larger block sizes may also lead to the problems of centralization.
- **Block compression:** In compact block relay technique the data structure is little altered. The block contains the headers and some short transaction IDs, used to match transactions already available to the receivers. The compact block messages are sent and receivers process these messages. The node A sends the compact block to the node B. When node B receives the compact block it calculates the transaction IDs in the memory pool and matches it with the transactions IDs in the compact block. If all transactions are available the block is reconstructed if not node B sends the getblocktxn message to node A to receive all transaction data. After that the block is constructed. In low bandwidth relaying the compact blocks are sent only if the request is made.

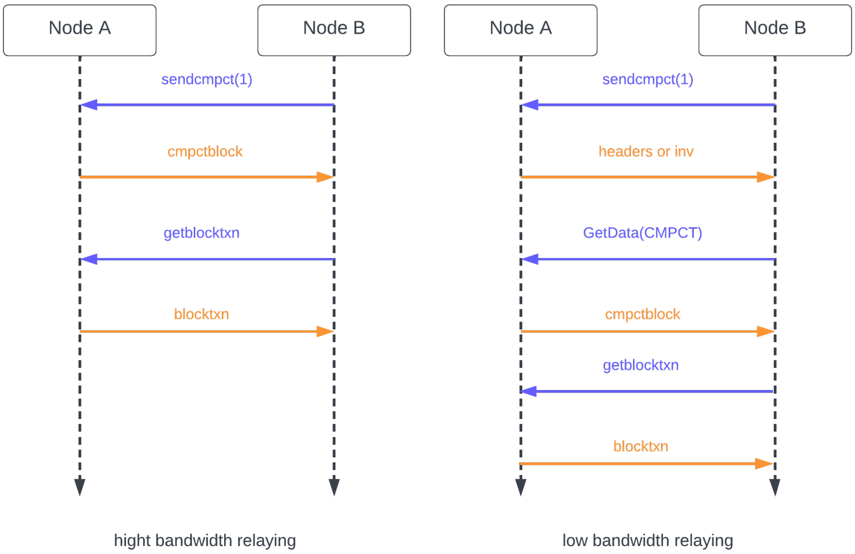


Figure 4.1: Block Compression

- **Storage scheme Optimization:** This scheme has a consensus unit. It has a number of nodes in it. The blocks of the whole chain are assigned to a single unit. This reduces the total query cost. Jidar is another approach. The main idea is to store only relevant data they are interested in. Small part of the data is stored including the relevant transactions and the merkel branch. If at all we need all the block data, the nodes ask the other nodes and the complete blocks are constructed out of it. There needs to be some incentives for other blocks to help the block asking for the complete data.

7. **Paper Name:** An Empirical Analysis Of Using Blockchain Technology In E-Voting Systems [7]
Authors: J. V. Cadiz, N. A. M. Mariscal and A. M. Ceniza-Canillo
Abstract: An analysis on the use of blockchain technology in e-voting systems has been done to compare its performance and security to that of a centralized e-voting system. Blockchain technology is one of the emerging technologies of today as a distributed, decentralized database technology, providing much promise to transparency and reducing cybersecurity risks. A Blockchain-based E-Voting System (BEVS) and a Centralized E-Voting System (CEVS) were developed, and compared in terms of performance and security testing. The results show that the BEVS processes requests slightly slower than the CEVS. This is due to the additional processes of block validation and creation in a blockchain network, whereas utilizing a local private blockchain as the database of an e-voting system handles requests faster than utilizing a public blockchain network. However, the BEVS is more reliable when it comes to the efficiency of an e-voting system as it has a complete 0.00% error rate compared to the CEVS with its errors generated by the internal server. The speed at which the BEVS may also be affected by heavy loads, however, all requests were fulfilled, and no system crashes occurred. The BEVS also proves to be a more secure e-voting system with less vulnerabilities detected.
Inference:
This paper discusses the difference between a centralised e-voting system and a blockchain based e-voting system in terms of performance and security. Both systems were designed to have similar interfaces. The only difference between them was the systems used to design the backend. With blockchain based systems, the users needed to use their digital wallet's private key as their credentials to connect to the network. If all the conditions of the contact are satisfied, a new block is created, processed and then added to the blockchain, which serves as the database for this system. The centralised system required users to log in with their voter ID. The votes are processed by the server and stored in a database, which automatically counts the number of votes for each candidate. Results showed that BEVS is slightly slower than CEVS. This is due to the additional processes of block validation and creation while using a local private blockchain, it processes requests faster than a public blockchain network. However, the BEVS is more reliable when it comes to efficiency as it has a zero error rate compared to the CEVS with its errors. through the internal server. However, the rate at which the BEVS works can be affected even by heavy loads. The BEVS is also proving to be a more secure

electronic voting system with fewer identified vulnerabilities.

8. **Paper Name:** A Framework to Make Voting System Transparent Using Blockchain Technology [8]
   Authors: M. S. Farooq, U. Iftikhar and A. Khelifi
   Abstract: A widespread mistrust towards the traditional voting system has made democratic voting in any country very critical. People have seen their fundamental rights being violated. Other digital voting systems have been challenged due to a lack of transparency. Most voting systems are not transparent enough; this makes it very difficult for the government to gain voters' trust. The reason behind the failure of the traditional and current digital voting system is that it can be easily exploited. The primary objective is to resolve problems of the traditional and digital voting system, which include any kind of mishap or injustice during the process of voting. Blockchain technology can be used in the voting system to have a fair election and reduce injustice. The physical voting systems have many flaws in it as well as the digital voting systems are not perfect enough to be implemented on large scale. This appraises the need for a solution to secure the democratic rights of the people. This article presents a platform based on modern technology blockchain that provides maximum transparency and reliability of the system to build a trustful relationship between voters and election authorities. The proposed platform provides a framework that can be implemented to conduct voting activity digitally through blockchain without involving any physical polling stations. Our proposed framework supports a scalable blockchain, by using flexible consensus algorithms. The Chain Security Algorithm applied in the voting system makes the voting transaction more secure. Smart contracts provide a secure connection between the user and the network while executing a transaction in the chain. The security of the blockchain based voting system has also been discussed. Additionally, encryption of transactions using cryptographic hash and prevention of attack 51% on the blockchain has also been elaborated. Furthermore, the methodology for carrying out blockchain transactions during the process of voting has been elaborated using Blockchain Finally, the performance evaluation of the proposed system shows that the system can be implemented in a large-scale population.
   Inference:
   This paper explains about the transparent voting system using blockchain technology. It reduces a lot of resources and efforts in the polling system. Unlike the traditional voting system which stores votes on a centralised system, blockchain based voting systems are not easily tampered. Blockchain provides a high level of security that can be trusted more than the previously used technologies.
   Here voters need to complete verification into the voting management system. The nation's database is integrated with the system's database to keep voters' integrity. For every vote, a transaction is generated against the voter's National ID then the transaction is saved in blockchain. After casting a vote his/her vote coin is used. After casting vote blockchain verifies his voting system by comparing with the national voting ids. Then miners analyse them to remove malicious votes before adding them to the chain.

9. **Paper Name:** Decentralised E-voting system based on Smart Contract by using Blockchain Technology [9]
   Authors: M. Al-madani, A. T. Gaikwad, V. Mahale and Z. A. T. Ahmed
   Abstract: Nowadays the use of the Internet is growing; E-voting system has been used by different countries because it reduces the cost and the time which used to consumed by using traditional voting. When the voter wants to access the E-voting system through the web application, there are requirements such as a web browser and a server. The voter uses the web browser to reach to a centralized database. The use of a centralized database for the voting system has some security issues such as Data modification through the third party in the network due to the use of the central database system as well as the result of the voting is not shown in real-time. However, this paper aims to provide an E-voting system with high security by using blockchain. Blockchain provides a decentralized model that makes the network Reliable, safe, flexible, and able to support real-time services.
   Inference:
   This Paper Explains the Difference between Ballot paper voting and Blockchain based decentralised casting For better integrity of the voting system.
   Here in Decentralised Blockchain System data like the name of voter or votes is saved on a decentralised ledger. This data neither be accessed nor be changed by any third party authority. Currency ballot paper is a widely used voting system worldwide. But this doesn't guarantee the correctness of the result Due to availability of data at a single resource. This problem is faced by a centralised voting system and is solved by Blockchain based voting systems. Here in Blockchain Technology the data is not stored on a central location but is allotted in different locations on different servers. The data which is distributed across each device connected to the blockchain using a peer-to-peer system.
   In the Decentralised E-voting system candidate registration is done before the voting process starts and then voters identity is verified before the creating account.In this system. The authorised person authenticates the voter and then blockchain ensures the double voting is not allowed.

10. **Paper Name:** BlockVOTE : An Architecture of a Blockchain-based Electronic Voting System [10]
    Authors: C. Angsuchotmetee, P. Setthawong and S. Udomviriyalanon
    Abstract: Electronic voting systems provide many advantages over traditional ballot based voting systems mainly over the accuracy and speed of the tallying process of the voting. However, electronic voting systems suffer from many technical and security issues which have limited its deployment in voting scenarios such as company voting and political elections. Centralized electronic voting systems are, by nature not secure, and there are many avenues of cyber-attacks that could tamper the voting result. Electronic voting system should be highly secured, tamperedproof guaranteed, and the voting should be trusted worthy. In this study, we propose BlockVOTE, a Blockchain-based electronic voting system. Our proposal uses Blockchain to ensure that the voting process can be kept secure and trustable through the consensus handling mechanism of the Blockchain. The architecture

design and implementation suggestion are provided in this study. The implementation of the proposal was developed and tested via experimentation. The experiment result and the discussion on the possibility of adopting our proposal in an actual election is provided at the end of this study.

Inference:

The paper proposes a Blockchain-based voting system called BlockVOTE. It focuses on keeping the voting system secure and trustable using consensus handling mechanisms. The paper covers Traditional Ballot based Voting systems and Electronic based Voting Systems with both their issues. In the proposed architecture, the paper defines upon processes Poll Creation, Voting and Result Tallying. Authors of the paper implemented the system using Ethereum and HyperLedger and compared the results. The system will have the following actors: Poll Creator, Contract Handler and Voter performing tasks of Poll Creation, Contract creation and deployment and Voting.

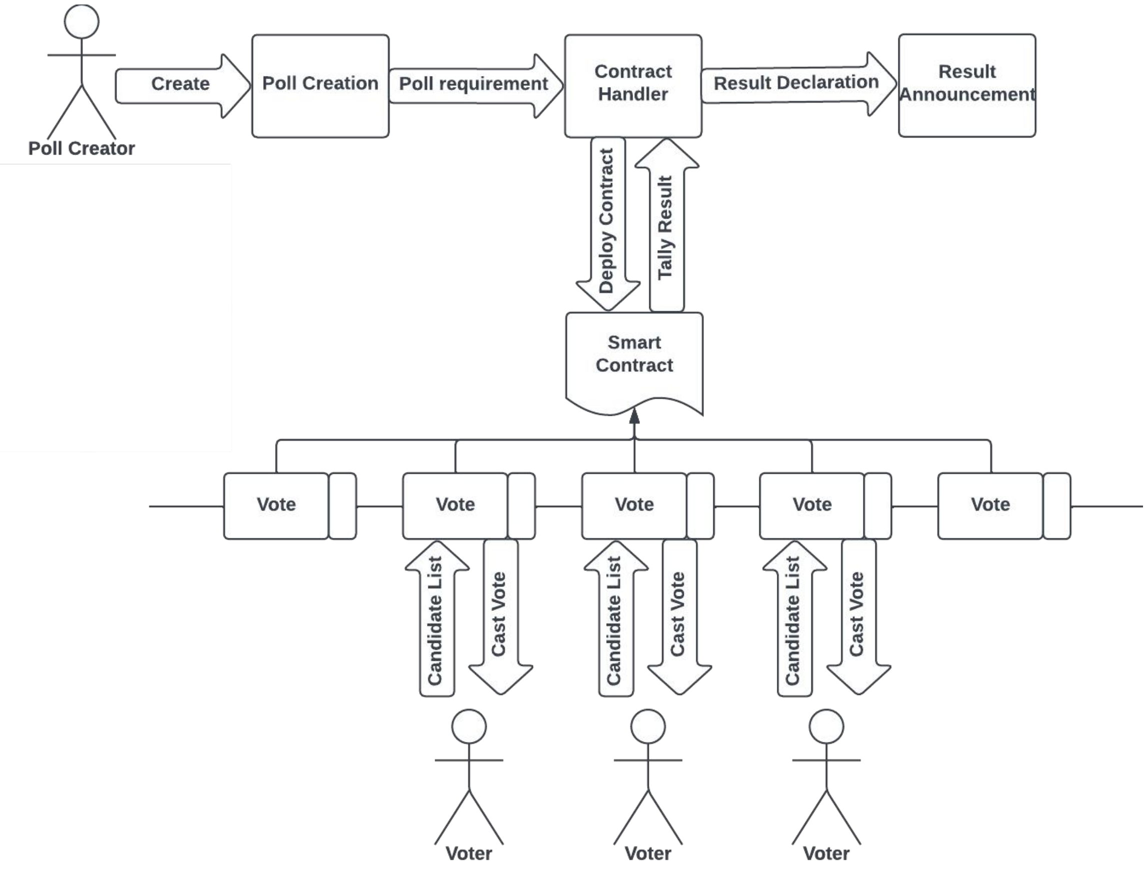


Figure 4.2: Blockchain based e-Voting Architecture

11. **Paper Name:** Analysis of Blockchain Solutions for E-Voting: A Systematic Literature Review [11]
    Authors: A. Benabdallah, A. Audras, L. Coudert, N. El Madhoun and M. Badra
    Abstract: To this day, abstention rates continue to rise, largely due to the need

to travel to vote. This is why remote e-voting will increase the turnout by allowing everyone to vote without the need to travel. It will also minimize the risks and obtain results in a faster way compared to a traditional vote with paper ballots. In fact, given the high stakes of an election, a remote e-voting solution must meet the highest standards of security, reliability, and transparency to gain the trust of citizens. In literature, several remote e-voting solutions based on blockchain technology have been proposed. Indeed, the blockchain technology is proposed today as a new technical infrastructure for several types of IT applications because it allows to remove the TTP and decentralize transactions while offering a transparent and fully protected data storage. In addition, it allows to implement in its environment the smart-contracts technology which is used to automate and execute agreements between users. In this paper, we are interested in reviewing the most revealing e-voting solutions based on blockchain technology.
Inference:
This paper reviews the latest innovations in the blockchain based e-voting system to understand their particulars and compares them with each other as well as the traditional voting process. Various blockchain based e-voting applications here are being compared based on many parameters like implementations used, algorithms used, voter identification methods, vote encryption methods, how they fare against attacks and their security properties. After comparing these based on the said factors, limitations and constraints of these systems. Even though the blockchain based systems for voting may be in their initial phases, they offer an interesting solution to the problems of traditional voting.

12. **Paper Name:** Blockchain Based E-Voting System: Open Issues and Challenges [12]
Authors: Z. Khudoykulov, U. Tojiakbarova, S. Bozorov and D. Ourbonalieva
Abstract: Blockchain technology has become very trendy and penetrated different domains, mostly due to the popularity of cryptocurrencies. Blockchain technology offers decentralized nodes for e-voting and is used to create e-voting systems, mainly because of their end-to-end verification benefits. This technology is an excellent replacement for traditional e-voting solutions with distributed performance, reliability and security. The following article provides an overview of e-voting systems based on blockchain technology. The main purpose of this analysis was to examine the current state of blockchain-based voting systems, as well as any associated difficulties in predicting future events.
Inference:
This paper analyses current research on blockchain electronic voting systems and identifies issues in it. E-Voting defines the process that uses electronic tools in the process of elections for voting and counting purposes. The procedure for electronic voting varies from country to country. which may include voting machines on polling stations, centralised accounting of paper bills and voting on the Internet. In many countries, centralised calculations are used. Sometimes, however, they also use electronic voting machines in places of voting. However, the use of the internet is minimal. In particular, electronic approaches have been tested in several places

and problems with security and reliability were noted. This paper also talks about several factors that makes an e-voting system secure like,

i **Anonymity:** Any correlation between registered voters and voter identities shall be anonymous.

ii **Auditability and Accuracy:** The results should be accurate and should precisely correspond to the voter sentiment.

iii **Democracy/Singularity:** Every eligible person should be able to vote. There shall be no duplication of votes.

iv **Vote Privacy:** No one should be able to associate a particular vote to an individual.

v **Robustness and Integrity:** It is the proof that registered voters will abstain without problems. It also encourages others to cast their legitimate votes.

vi **Transparency and Fairness:** No one outside the people involved with the counting process can find out about the results before they are announced.

vii **Availability and Mobility:** The systems should always be available during the electoral process.

viii **Verifiable Participation/Authenticity:** The authorities should be able to check if someone abstained from voting.

ix **Recoverability and Identification:** It should be able to track and restore data in order to avoid attacks or data losses.

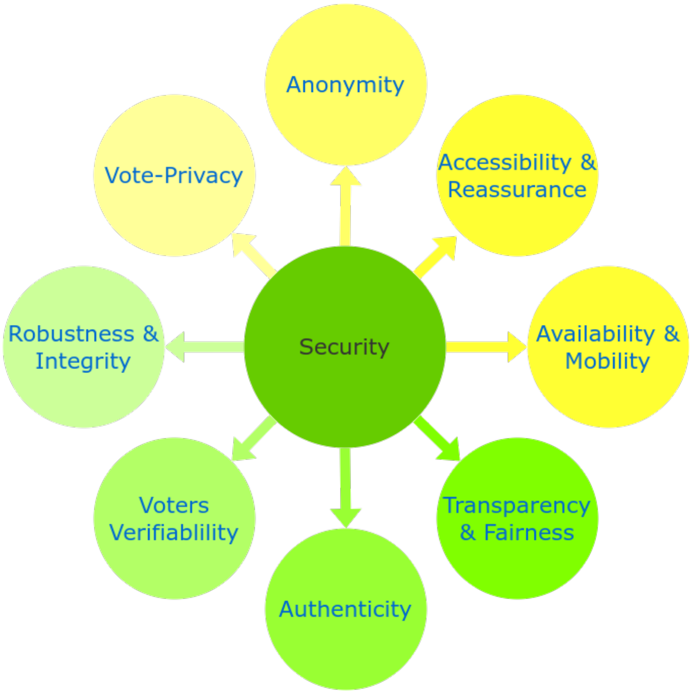


Figure 4.3: Security Requirements for Electronic Voting System

13. **Paper Name:** Blockchain for Electronic Voting System—Review and Open Research Challenges [13]
Authors: Jafar, Uzma, Mohd Juzaiddin Ab Aziz, and Zarina Shukur
Abstract: Online voting is a trend that is gaining momentum in modern society. It has great potential to decrease organizational costs and increase voter turnout. It eliminates the need to print ballot papers or open polling stations—voters can vote from wherever there is an Internet connection. Despite these benefits, online voting solutions are viewed with a great deal of caution because they introduce new threats. A single vulnerability can lead to large-scale manipulations of votes. Electronic voting systems must be legitimate, accurate, safe, and convenient when used for elections. Nonetheless, adoption may be limited by potential problems associated with electronic voting systems. Blockchain technology came into the ground to overcome these issues and offers decentralized nodes for electronic voting and is used to produce electronic voting systems mainly because of their end-to-end verification advantages. This technology is a beautiful replacement for traditional electronic voting solutions with distributed, non-repudiation, and security protection characteristics. The following article gives an overview of electronic voting systems based on blockchain technology. The main goal of this analysis was to examine the current status of blockchain-based voting research and online voting systems and any related difficulties to predict future developments. This study provides a conceptual description of the intended blockchain-based electronic voting application and an introduction to the fundamental structure and characteristics of the blockchain in connection to electronic voting. As a consequence of this study, it was discovered that blockchain systems may help solve some of the issues that now plague election systems. On the other hand, the most often mentioned issues in blockchain applications are privacy protection and transaction speed. For a sustainable blockchain-based electronic voting system, the security of remote participation must be viable, and for scalability, transaction speed must be addressed. Due to these concerns, it was determined that the existing frameworks need to be improved to be utilized in voting systems.
Inference:
A blockchain includes the following components:

i **Node:** It refers computer or user participating in the blockchain network.

ii **Transaction:** The is the data stored in blocks making it the most fundamental part of a blockchain system.

iii **Block:** It is the data structure which is used for storing the transaction information.

iv **Chain:** A sequence of blocks which are organized in some specific order.

v **Miners:** Specific nodes which perform the block verification process.

vi **Consensus:** Algorithm which ensures validity of block and decides whether it should be inserted to chain or not.

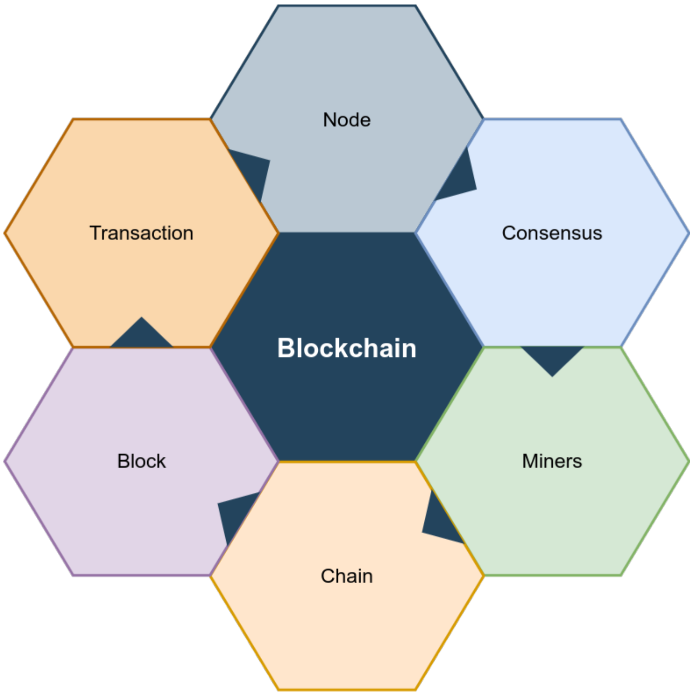


Figure 4.4: Core Components of Blockchain Architecture

4.2 Outcome of Literature Survey

Blockchain is a new emerging technology. It is currently under research to implement with perfection. Blockchain has shown some promising applications. Some of its famous applications include finance industry, healthcare industry, art industry etc. The concept of blockchain was proposed in 2008 by a person named Satoshi Nakamoto and was actualized a year later. Based on the application of the blockchain there are different characteristics of it that make it different. They are cost effective. Reduces the overall cost of conducting transactions on our case voting. There is no need for a central agency. Persistence helps us keep track of all the logs of voting in the system. There are three roles in the blocks. The one who proposes a transaction/vote. The one who checks or validates whether the votes are valid or not and applies it to itself. And lastly the one who learns from all these transactions and implements the votes to itself. Blockchain systems provides you with anonymity over traditional systems. They do not collect the users private data. The blocks added to the system remain there forever. This facilitates auditing of the transaction or in our cases votes. There are different tradeoffs to make blockchain scalable. Lately there was a fork made to bitcoin to increase its scalability. Bitcoin having a block size of 1MB could accommodate a smaller number of transactions, whereas the fork (now known as bitcoin-cash) has a block size of 8-32 MB. This increases the number of transactions that can be accommodated in a block. Another technique to increase scalability is to tweak the data structure of the block. It calculates the transaction IDs in the memory pool and matches it with the transaction IDs in the compact block.

Advantages:

1. The votes cannot be manipulated as consensus algorithms are used.
2. Before a node (i.e a vote) is added to the chain, its legitimacy is verified by various nodes in the network.
3. Every node in the network has a copy of the ledger and therefore the transparency of the process increases.
4. There is no central server/system which reduces the possibility of data loss and malicious attacks.
5. Also since there is no central server/system it increases the reliability of the system as there is no single point of failure.
6. The counting process also becomes more efficient and use of blockchain increases the accuracy of the system.
7. Mechanisms can be put in place to prevent duplicate voting.

Disadvantages:

1. As the number of nodes increases, the amount of time required to process a transaction also increases.
2. There is a chance that the voters can be intimidated or coerced to vote against their will.
3. There needs to be proper power supply and backup in order for the system to function seamlessly.
4. As blockchain is resource intensive the cost of hardware increases and power consumption increases.
5. If the number of active nodes decrease then the security of the blockchain aso decreases.
6. People in general are reluctant to use new technology.
7. As the number of nodes increases, it becomes increasingly difficult to maintain the nodes.

4.3 Conclusion of Literature Survey

As we have seen, all the above-mentioned papers have their own advantages. By evaluating all the advantages we can develop the platform for blockchain based E-voting systems. The proposed platform provides a framework that can be implemented to conduct voting activity digitally through blockchain without involving any malpractices. Our goal is to create an E-voting system that can be trusted by voters and to encourage them to vote.

CHAPTER 5
PROBLEM STATEMENT AND SCOPE

5.1 Problem Statement

Blockchain technology can be used in the E-voting system to conduct a fair election and reduce injustice. The physical voting systems have many flaws in it as well as the digital voting systems are not perfect enough to be implemented on a large scale. Our aim here is to create a private network responsible for storing and maintaining blockchain. Upon this network, develop a decentralized blockchain based Voting System system based on smart contract which is the optimum solution for frauds, forgery, human error, traceability, stagnancy occurring in the current system providing enhanced security, privacy, efficiency and speed in the Voting process.

5.1.1 Goals and Objectives

To design a decentralized e-voting system that,

1. Reduces human interaction in the process of voting and elections in general.
2. Does not depend on a trusted third party for controlling the collected data, whilst still staying immune to attacks and guaranteeing user privacy.
3. Provides transparency, ensuring security and does not jeopardize voter privacy.

5.1.2 Scope

Blockchain technology has proven itself in recent years as a tamper proof solution to store information. When it comes to lack of trust in the system and possibility of cheating we can trust on blockchain. Blockchain is almost unhackable with traditional techniques providing exceptional security. The goal of this project is to reduce corruption and conduct fair elections. The scope currently is to provide the proof of concept and then scale it. Also in order to scale it we need to do the potential tradeoffs to optimize performance and reduce the harm it causes to the environment. Using efficient algorithms to reduce the number of calculations required.

5.2 Major Constraints

1. The application should be able to manage a high amount of requests and transactions while maintaining the correctness of the results.
2. The application must also ensure that the identities and personal information of the users are safeguarded.
3. The systems must be usable by all users, regardless of their technological abilities or accessibility requirements.

5.3 Methodologies of Problem Solving and Efficiency Issues

5.4 Expected Outcome

1. Promote transparency in the electoral process.

2. Improve security by encrypting vote data and preventing unauthorized access.
3. All transactions are recorded on a public ledger that cannot be altered, which can also help to prevent tampering with voting results.
4. The automation of numerous electoral processes, such as vote counting and auditing.

5.5 Applications

1. A decentralized voting application can be used for corporate governance, allowing shareholders to vote on crucial issues such as mergers, acquisitions, and board appointments.
2. A decentralized voting application can be used for public opinion polls, allowing people to voice their views on political, social, and economic topics in a secure and anonymous manner.
3. A decentralized voting application can be used in community governance, allowing members of a community to vote on crucial choices such as community initiatives, resource allocation, and public works.
4. Referendums can be held using a decentralized voting application, which allows citizens to vote on key matters such as constitutional amendments, changes in public policy, and large infrastructure projects.

CHAPTER 6
PROJECT PLAN

6.1 Project Estimates

Use Agile model and associated streams derived from assignments 1,2,3,4 and5 (Annex A and B) for estimation.

6.1.1 Sequential Phases in Agile Methodology

The software development model that is used in this project is Agile Model. It is an iterative and incremental approach to software development that emphasizes collaboration, flexibility and adaptability. It consists of the following stages:

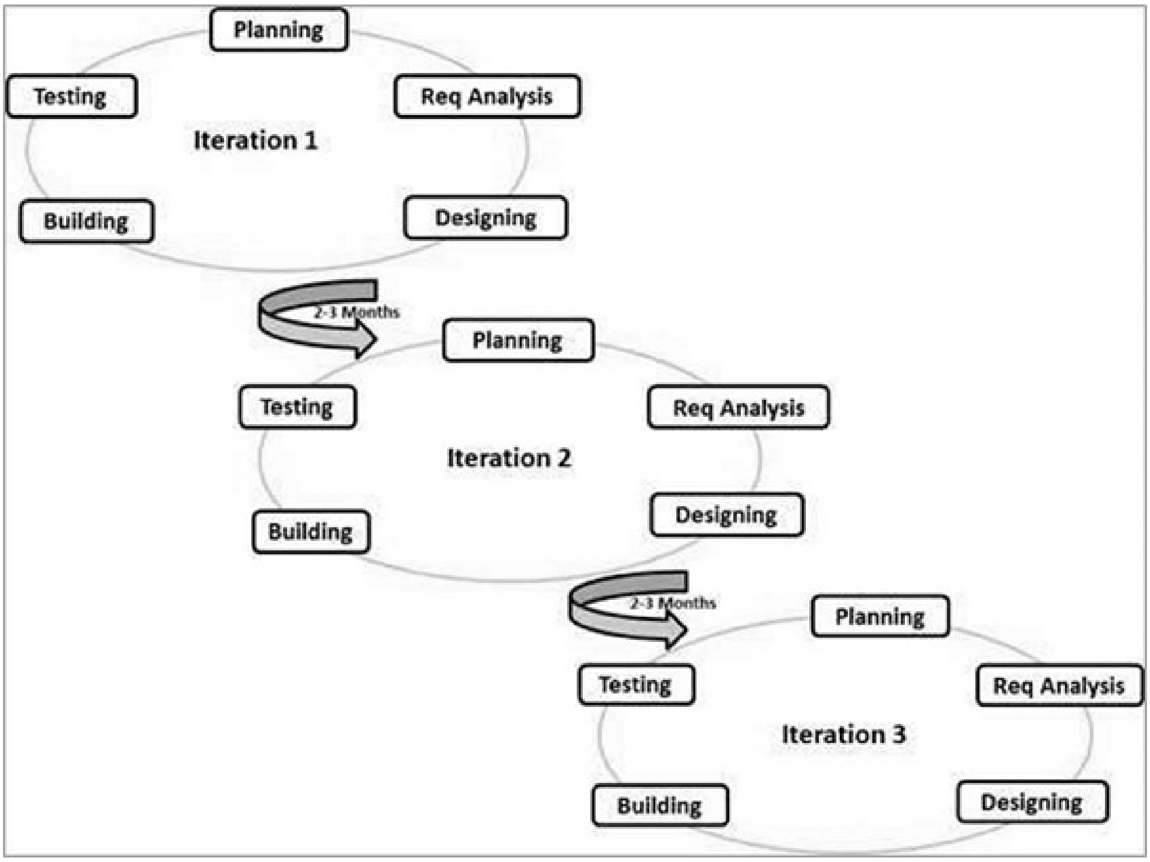


Figure 6.1: Agile Methodology

1. **Planning :** During this stage, the development team determines the project's needs and scope. Following that, the team develops a plan outlining the tasks and deliverables for each sprint.

2. **Requirements analysis :** In this stage, the development team improves the requirements and builds user stories. User stories are brief descriptions of features written from the perspective of the end user.

3. **Design :** Based on the requirements and user stories, the development team designs a design for the feature or product at this stage.

4. **Implementation :** The development team constructs the feature or product according to the design at this stage. This stage is frequently divided into smaller sprints, with each sprint producing a functional product or feature.
5. **Testing :** The development team tests the feature or product at this stage to confirm that it meets the requirements and is free of faults and defects.
6. **Deployment :** The feature or product is deployed to the client or end-users at this step. During this stage, the development team may also provide training and support to the customer.
7. **Review :** At this point, the development team and the customer go over the sprint outcomes and provide input. This feedback is utilized to improve the design and implementation of the feature or product by refining the requirements.

6.1.2 Project Resources

Well configured Laptop, 8 GB RAM, 40 GB Storage, Intel i5 Processor, Remix IDE.

6.2 Risk Management

The process of identifying the risk (Product risk or Project risk) , analyzing the risk, and then mitigating the risk or controlling the risk is known as Risk Management. We identified following risks that were managed as below:

6.2.1 Risk Identification

Risk management begins with risk identification. By utilizing certain approaches, we must identify both project and product risk. The use of risk-free templates, stakeholder interviews, project retrospectives, etc. are some of the most popular methods that may be used to detect various risks.

- Risk: Risk is the expectation of suffering a loss or encountering a potential issue in the future, whether or not it does so. Usually, it results from a lack of knowledge, power, or time. A software risk is the potential for financial loss during the software development process.
- Product Risk: Risks associated with a software product or programme that may develop as a result of its inability to function as users want it to. These risks include any kind of unforeseen or unpredictable incident or activity that could happen and harm a project's progress.
- Some of the classic strategies include:
    1. Avoid it: Sometimes a plan can be developed to entirely prevent a risk. It might be conceivable to use a different supplier, for instance, if there's a chance that a certain one will supply a necessary item too late to meet your deadline.

2. Accept it: It might not be worthwhile to handle the risk at all if the impact and probability are low.
3. Reduce it: Establishing contingency plans and funding, adding team members with the knowledge to take over a crucial task in the event that others leave, and cultivating a favorable relationship with persons outside the team whose cooperation is crucial for success are some approaches to decrease risks.
4. Transfer it: In some cases, you can delegate control of a particular risk to a supplier, an independent consultant, or even a stakeholder.

6.2.2 Risk Analysis

Probability	Value	Description
High	Probability of occurrence is	>75%
Medium	Probability of occurrence is	26 – 75 %
Low	Probability of occurrence is	<25%

Table 6.1: Risk Probabilities

Impact	Value	Description
Very High	>10 %	Schedule impact or Unacceptable quality
High	5 – 10 %	Schedule impact or some parts of the project have low quality
Medium	<5 %	Schedule impact or barely noticeable degradation in quality Low impact on schedule or Quality can be incorporated

Table 6.2: Risk Impact Definitions

6.2.3 Overview of Risk Mitigation, Monitoring, Management

Following are the details for each risk.

6.3 Project Schedule

6.3.1 Project Task Set

Major Tasks in the Project stages are:

- Task 1: Correctness
- Task 2: Availability
- Task 3: Integrity

6.3.2 Task Network

Project tasks and their dependencies are noted in this diagrammatic form.

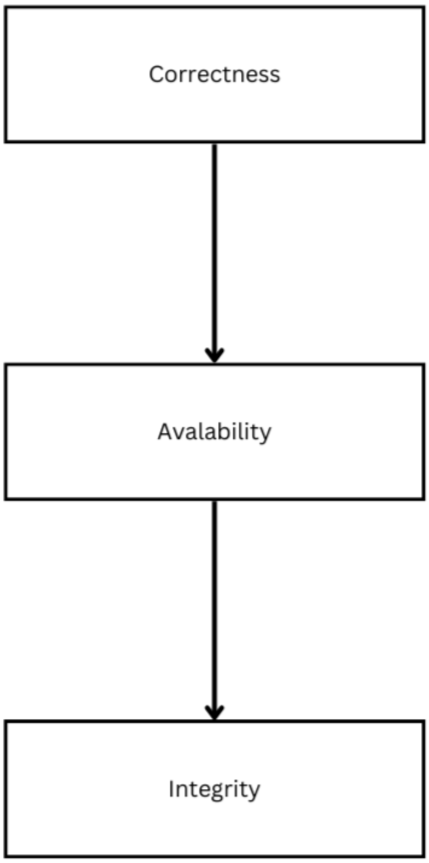


Figure 6.2: Task Network

6.3.3 Timeline Chart

A project timeline chart is presented. This may include a timeline for the entire project. Above points is also covered in Project Planner as Annexure C.

6.4 Team Organization

Team consists of 4 members and proper planning mechanism is used androles of each member are defined.

6.4.1 Team Structure

The team structure for the project is identified. There is total 4 members in our team and roles are defined. All members are contributing in all the phases of project.

Month Scheduled	Phase	Members	Work Done
July - August	Topic searching	All	Topic Searched
August-September	Topic selection	All	Topic Selected and finalized
August-September	Project Confirmation	All	Project approved by Internal and External guide
August-September	Literature Survey	All	Literature Survey was done
September – October	Requirement Analysis	All	Requirement analysis was done
October-November	Survey paper publishing	All	Survey Paper Published
November-December	Network Creation	Yash	A private network was created
December-January	Creation of Blockchain and Genesis Block	Yash, Arvind	Blockchain was implemented and genesis block was created
December-January	Implementation of Solidity Contracts	Arvind, Gaurav	Solidity Contracts were implemented for voting and authentication
January-February	Implementation of Front End using React	Gaurav	
February-March	Optimization and Performance Evaluation	Shree, Gaurav	
March-April	Code Integration and testing	Shree, Yash, Gaurav	Code was combined and test cases were generated
April-May	Implementation Paper Publishing	All	Implementation Paper Published
April-May	Documentation	All	Documentation Done

Table 6.3: Team Structure Phases

6.4.2 Management reporting and communication

- Well planning mechanisms are used for progress reporting and inter/intra team communication is identified as per requirements of the project.
- Each team member had worked on modules. Each unit was worked on and finished simultaneously. After the competition of all modules, unit testing was performed, Final modules were integrated together, and integration testing was performed.

CHAPTER 7
SOFTWARE REQUIREMENT SPECIFICATION

7.1 Introduction

7.1.1 Purpose and Scope of Document

A software requirements specification is the basis for your entire project. It lays the framework that every team involved in development will follow. It is used to provide critical information to multiple team's development quality assurance, operations, and maintenance. This keeps everyone on the same page. Using the SRS helps to ensure requirements are fulfilled: And it can also help you make decisions about your products lifecycle for instance, when to retire a feature. Writing an SRS can also minimize overall development time and costs. Embedded development teams especially benefit from using an SRS.

7.1.2 Overview of responsibilites of Developer

A Developer must perform project design and development activities according to customer specifications. Work with Manager in developing the project plan, budget, and schedule. He has to coordinate with management in preparing project proposals and contractual documents. A Project developer has to keep track of project progress regularly and develop status reports to management. Ensure that the project is completed within the allotted budget and timelines. Follow company policies and safety regulations for operational efficiency, Other responsibilities consist of research and recommend new technologies to carry out project development tasks. Provide assistance to other Developers, perform peer reviews and provide feedback for improvements. To develop cost reduction mitigative while maintaining quality and productivity.

7.2 System Requirements

7.2.1 System Interface

Hardware Interface

- **RAM :** 8 GB (As we are using Blockchain hosted on private networks. Hence minimum Ram requirement is 8 GB)
- **Hard Disk :** 40 GB
- **Processor :** Intel i5 Processor

Software Interface

- **Operating System :** Portable. Latest Operating System that supports all types of installation and development Environment.
- **IDE :** Remix IDE for Solidity Smart Contracts. It is a robust toolbox that users of any skill level can utilize for the whole process of contract development and deployment.
- **Languages :** Solidity, JavaScript, HTML, CSS,Python

7.3 Usage Scenarios

7.3.1 User profiles

1. Voter : The voter has limited access to the system. His function is to register. Then to register. There will be a provision to give him a coin. The coin will be consumed when the voter casts a vote.

2. Admin : The admin is mostly the moderator of the system. He has access to the candidate data. The voter data will be anonymous. But candidate data will be accessible. He can add different parties in the voting system.

3. Candidate : Candidates basically define a party. They are basically the on which voters vote. Candidates are registered by admins. They can only see the results in their dashboard . They can't cast votes.

7.3.2 Use Case View

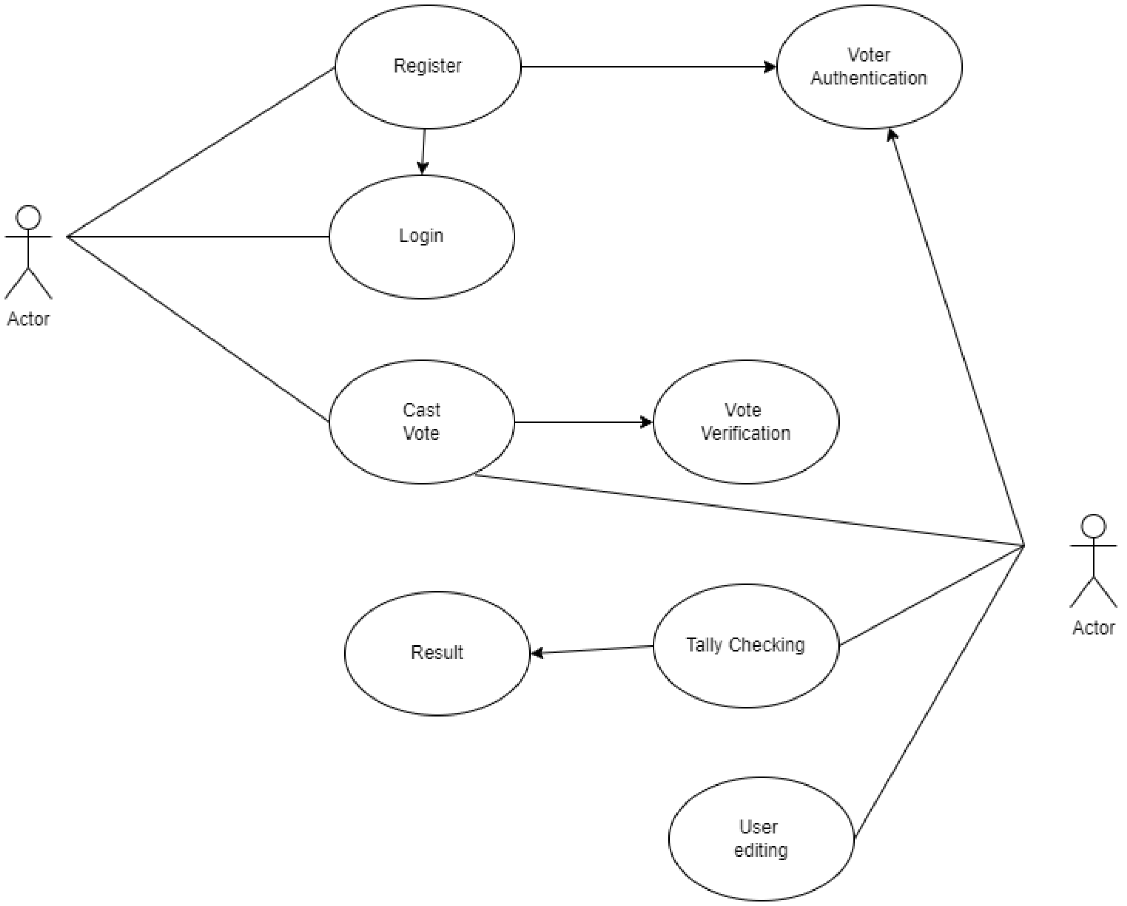


Figure 7.1: Use Case Diagram

7.4 Data Model and Description

7.4.1 Data Description

Data objects that will be managed/manipulated by the software are described in this section. The database entities or files or data structures required to be described. For data objects details can be given as below.

7.4.2 Data Objects and Relationships

Data objects and their major attributes and relationships among data objects are described using an ERD like form.

7.5 Functional Model and Description

A description of each major software function, along with data flow (structured analysis) or class hierarchy (Analysis Class diagram with class description for object-oriented system) is presented.

7.5.1 Data Flow Diagram

7.5.2 Activity Diagram

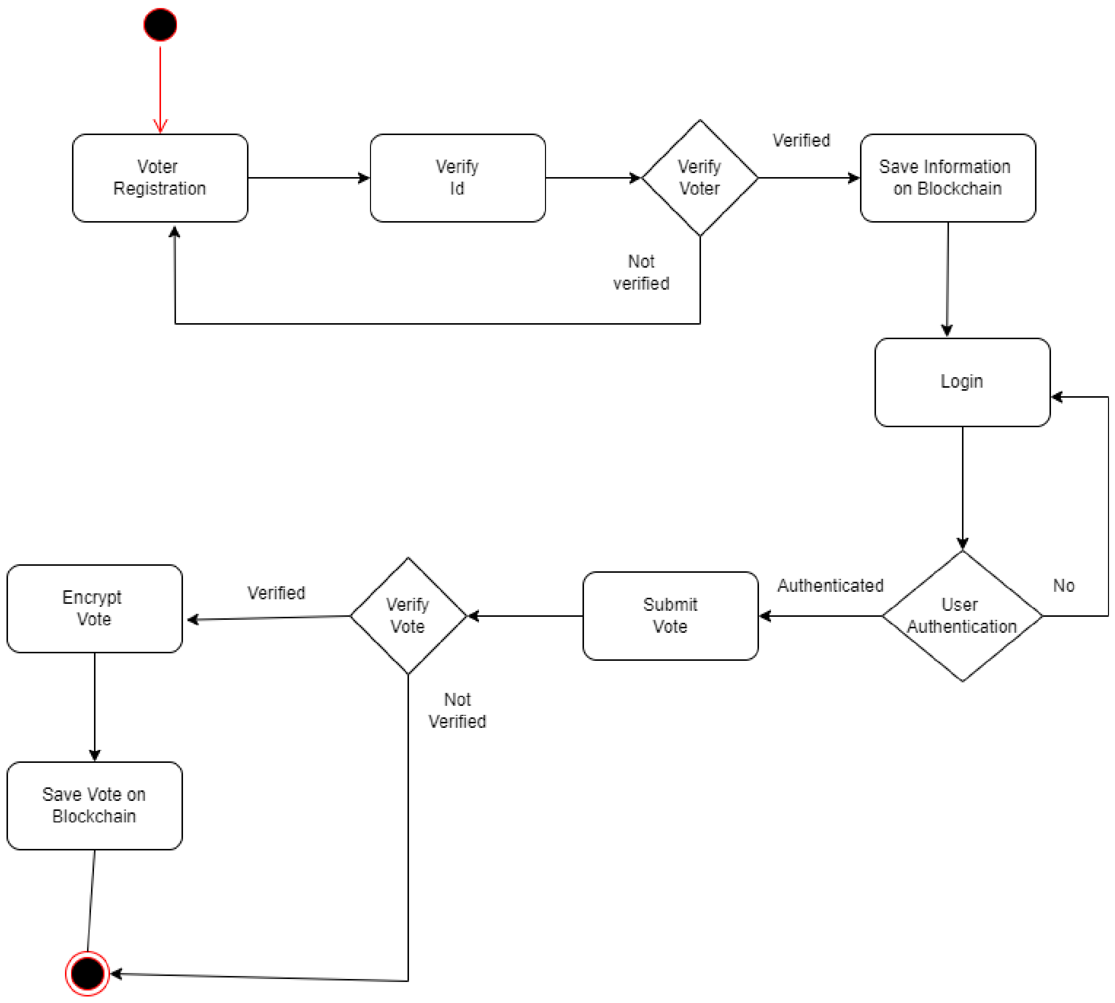


Figure 7.2: Activity Diagram

7.5.3 Non-Functional Requirements

- **Performance Requirements**
  The performance of the functions and every module must be well. The overall performance of the software will enable the users to work efficiently. Performance of encryption of data should be fast.

- **Safety Requirement**
  The application is designed in modules where errors can be detected and fixed easily. This makes it easier to install and update new functionality if required.

- **Software Quality Attributes**
  Our software has many quality attribute that are given below:
  1. **Adaptability :** This software is adaptable by all users.
  2. **Maintainability :** After the deployment of the project if any error occurs then it can be easily maintained by the software developer.
  3. **Reliability :** The performance of the software is better which will increase the reliability of the Software.
  4. **User Friendliness :** Since, the software is a GUI application; the output generated is much user friendly in its behavior.
  5. **Integrity :** Integrity refers to the extent to which access to software or data by unauthorized persons can be controlled.
  6. **Security :** Users are authenticated before casting votes.
  7. **Test ability :** The software will be tested considering all the aspects.

7.5.4 Design Constraints

7.5.5 Software Interface Description

The first screen that appears as the module is started is the Login Page which asks for user credentials to log in to the system.

After the user is logged in, he will be greeted with another page, where he would have a choice to vote for a candidate from multiple choices.

For admin there is an Admin Dashboard where he can monitor voters who have votes as well as the number of votes of the candidates.

CHAPTER 8
DETAILED DESIGN DOCUMENT USING ANNEXURE A AND B

8.1 Introduction

India is said to have the largest democracy. For the electoral voting system India is using modern “Electronic Voting Machines” which are responsible for taking user input and storing the votes entered by the end user. Later these votes are carried to a central location where they are counted and the result is declared. Though, there is a possibility that these votes data can be tampered with. Having a single central server to store data creates a single point of failure for the entire system. These issues cause mistrust regarding the current system. In the traditional ballot paper system, vote authenticity was questionable and had its separate issues for which it was replaced with modern Electronic Voting Machines.

Blockchain is based on Distributed Ledger Technology(DLT). In DLT, replicas of the same data are made on different nodes in the network due to which there is no single point of failure. Moreover, due to consensus algorithms implemented in blockchain, all nodes decide and agree upon whether a block is valid and must be added to the network or not

Decentralized voting systems can provide increased transparency, as they often rely on blockchain technology or other distributed ledger systems. Every vote cast can be recorded on the blockchain, making the process transparent and verifiable by anyone. This transparency helps build trust in the integrity of the voting system. Centralized voting systems are vulnerable to various security risks, such as hacking, tampering, or manipulation of results. In contrast, decentralized voting systems leverage the security features of blockchain technology, which makes it extremely difficult for any single entity or malicious actor to tamper with the results. Each vote is recorded in a transparent and immutable manner, ensuring the integrity of the voting process.

In centralized voting systems, there is a risk of censorship by the central authority. Decentralized voting systems aim to mitigate these risks by empowering individual voters to directly participate and control their votes. By removing the need for intermediaries, decentralized systems provide a higher level of autonomy and reduce the potential for manipulation or censorship.

8.2 Architectural Design

The architecture of a system describes its major components, their relationships(structures), and how they interact with each other. Software architecture and design includes several contributory factors such as Business strategy, quality attributes, human dynamics, design and IT environment.

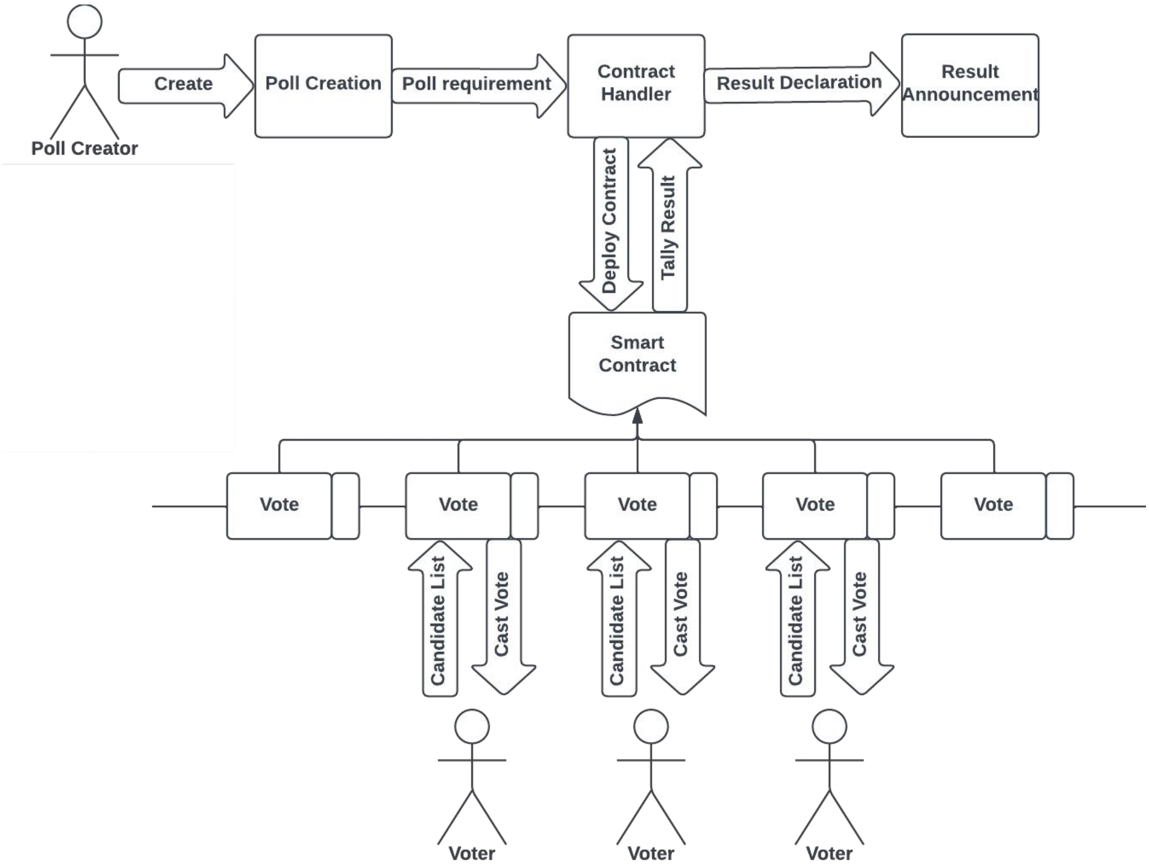


Figure 8.1: Architecture Diagram

8.3 Data Design (Using Annexure A and B)

8.3.1 Internal Software Data Structure

Data design is the first design activity, which results in less complex, modular and efficient program structure. The information domain model developed during the analysis phase is transformed into data structures needed for implementing the software. The data objects, attributes and relationships depicted in entity relationship diagrams and the information stored in data dictionaries provide a base for data design activity. During the data design process, data types are specified along with the integrity rules required for the data.

8.3.2 Database Description

In a Blockchain network LevelDB is used to store the blocks. LevelDB is an open-source key-value storage library developed by Google. It is designed to provide a simple, lightweight, and efficient solution for storing and retrieving data. LevelDB is written in C++ and provides bindings for various programming languages, making it accessible for developers across different platforms.

8.4 Component Design

Component diagrams are used in modeling the physical aspects of object-oriented systems that are used for visualizing, specifying, and documenting component- based systems and also for constructing executable systems through forward and reverse engineering. Component diagrams are essentially class diagrams that focus on a system's components that are often used to model the static implementation view of a system.

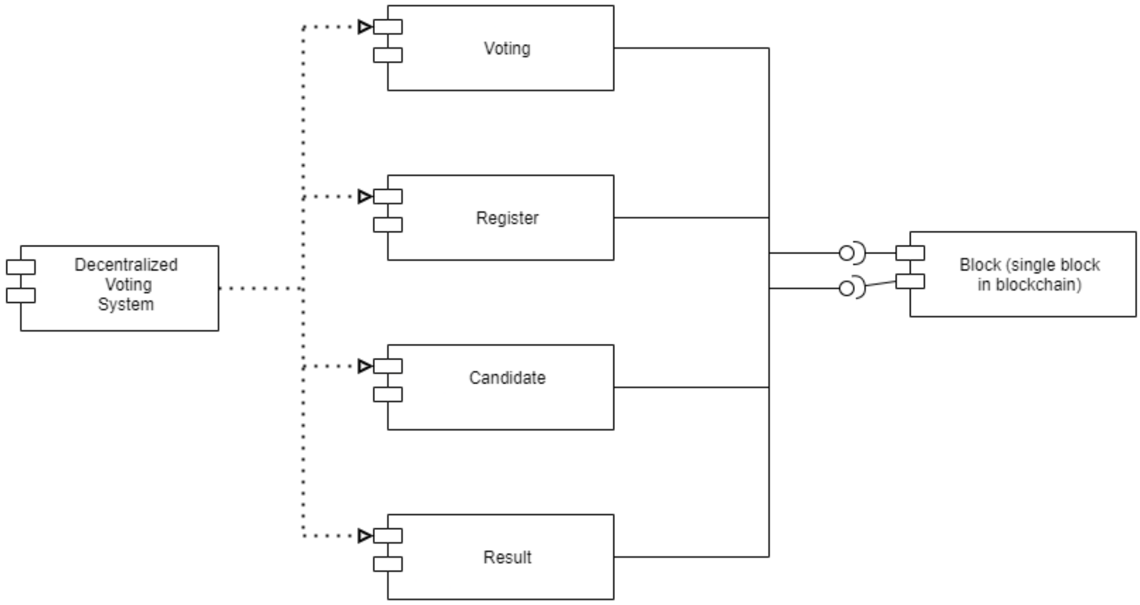

Voting
Register
Decentralized Voting System
Block (single block in blockchain)
Candidate
Result

Figure 8.2: Component Diagram

8.4.1 Class Diagram

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

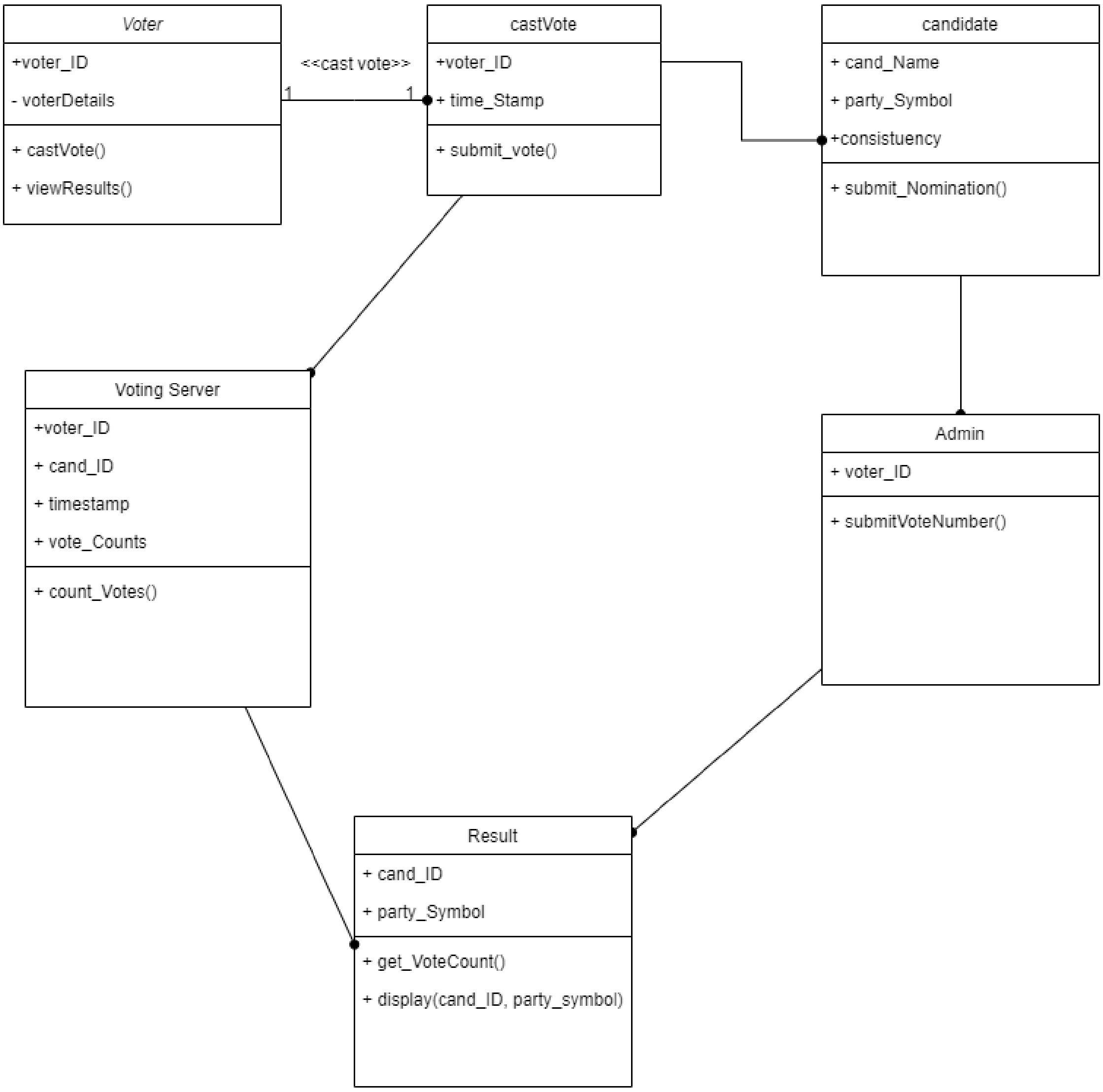


Figure 8.3: Class Diagram

8.4.2 Sequence Diagram

Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focused and they show the order of the interaction visually by using the vertical axis of the diagram to represent time, what messages are sent and when.

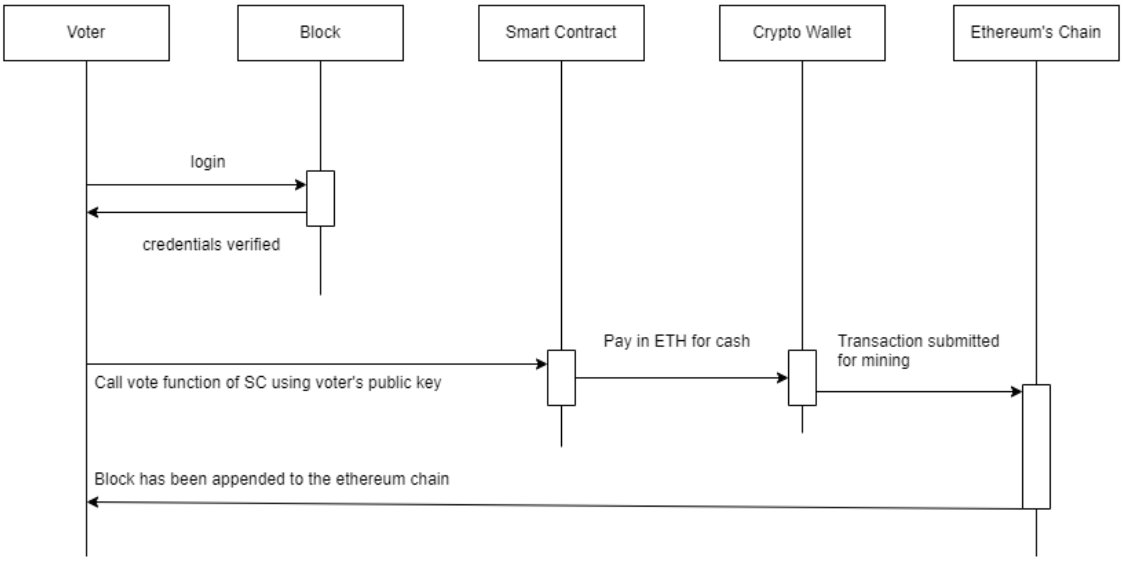

Voter
Block
Smart Contract
Crypto Wallet
Ethereum's Chain
login
credentials verified
Pay in ETH for cash
Transaction submitted for mining
Call vote function of SC using voter's public key
Block has been appended to the ethereum chain

Figure 8.4: SequenceDiagram

CHAPTER 9
PROJECT IMPLEMENTATION

9.1 Introduction

9.2 Module Description

- **Private Network**
  A blockchain private network, also known as a permissioned blockchain network, is a type of blockchain network that restricts access to participating nodes or entities. Unlike public blockchains, where anyone can join and participate, private blockchain networks are designed for specific organizations, consortia, or groups with predefined access controls.
  In a private blockchain network, the participating nodes are typically known and identified, and their permissions and roles within the network are predefined. This allows for more control and privacy compared to public blockchains, where all transactions and data are visible to the entire network.
  Private blockchain networks offer several advantages:

  1. Enhanced privacy
  2. Increased scalability and efficiency
  3. Customized governance
  4. Reduced energy consumption

  Private blockchain networks are commonly used in industries such as finance, supply chain management, healthcare, and government, where data privacy, security, and controlled access are of utmost importance. These networks enable participants to collaborate, share data, and execute transactions securely and efficiently within a trusted ecosystem.

- **Smart Contract**
  A smart contract is a self-executing contract with predefined terms and conditions written in code. It is built on blockchain technology, most commonly associated with platforms like Ethereum, and allows for the automation and enforcement of agreements without the need for intermediaries.
  Smart contracts operate on the principle of ”if-then” statements, where the terms of the contract are translated into lines of code. Once deployed on the blockchain, these contracts are immutable and execute automatically when certain predetermined conditions are met. This automation eliminates the need for intermediaries, such as lawyers or escrow agents, as the contract’s execution is based on predefined rules and cannot be altered or tampered with.
  The key features of smart contracts include:

  1. Autonomy
  2. Transparency
  3. Security
  4. Efficiency

Smart contracts have a wide range of applications across various industries. They can be used for financial transactions, supply chain management, real estate transactions, intellectual property rights, voting systems, and more. By eliminating the need for intermediaries and relying on the security and transparency of blockchain technology, smart contracts offer a new level of efficiency and trust in conducting business and executing agreements.

- **Frontend**
  Frontend refers to the client-facing part of the website. It encompasses the user interface (UI) and user experience (UX) components that users interact with directly. The frontend is responsible for presenting data and functionality in a visually appealing and intuitive manner, enabling users to interact with the application or website effectively.

- **Synthetic Data**
  Synthetic data refers to artificially generated data that mimics the characteristics and statistical properties of real-world data. It is created using algorithms or models to simulate data patterns, distributions, and relationships found in actual datasets. Synthetic data is designed to maintain privacy and confidentiality by not containing any personally identifiable information (PII) or sensitive data.

9.3 Tools and Technologies Used

- **Programming Languages**
  - Solidity
    Solidity is a programming language specifically designed for developing smart contracts on blockchain platforms, with Ethereum being the most popular one. It is a statically-typed, high-level language that enables developers to write code for executing smart contracts on the Ethereum Virtual Machine (EVM).
  - JavaScript
    JavaScript is a versatile and widely-used programming language primarily used for web development. It is a high-level, interpreted language that allows developers to add interactivity and dynamic behavior to websites. JavaScript can be executed on both the client side (in web browsers) and the server side (with the help of frameworks like Node.js).
  - Python
    Python is a high-level, interpreted programming language known for its simplicity, readability, and versatility. It was created by Guido van Rossum and first released in 1991. Python's design philosophy emphasizes code readability and a clean syntax, making it easier to write and understand compared to many other programming languages.
    Python is widely used in various domains, including web development, scientific computing, data analysis, artificial intelligence, machine learning, automation, scripting, and more. Its versatility, readability, and extensive library

support make it a popular choice for both beginners and experienced developers alike.

  - TypeScript
    TypeScript is a programming language developed by Microsoft that is a superset of JavaScript. It enhances JavaScript by adding optional static typing, which allows developers to catch errors and improve code quality during the development process. TypeScript code is translated into plain JavaScript and can be executed in any JavaScript runtime environment.
  - HTML
    HTML (Hypertext Markup Language) is the standard markup language used for creating the structure and content of web pages. It provides a set of tags (markup elements) that define the various components and elements of a web page. HTML is the backbone of the World Wide Web and is interpreted by web browsers to display web content.
  - CSS
    CSS (Cascading Style Sheets) is a style sheet language used to describe the presentation and visual appearance of web pages written in HTML and XML. It provides a set of rules that define how elements on a web page should be displayed, including their layout, colors, fonts, spacing, and other visual properties.

- **Frameworks/Libraries/Packages**
  - ReactJS (JavaScript)
    ReactJS is a popular JavaScript library for building user interfaces (UIs) and creating interactive web applications. It was developed by Facebook and released in 2013. ReactJS follows a component-based architecture, where UIs are broken down into reusable and self-contained components that manage their own state and update efficiently.
    ReactJS is widely used for building single-page applications, mobile applications (with React Native), and complex web interfaces. Its modular and efficient architecture, along with the vibrant community support, has made it a popular choice among developers for creating interactive and scalable UIs.
  - Chai (JavaScript)
    Chai is a popular assertion and expectation library for testing JavaScript code. It provides a clean and expressive syntax for making assertions about the behavior and expected outcomes of functions and code blocks. Chai can be used in various JavaScript environments, including browser-based applications and Node.js.
    Chai provides a powerful and expressive way to write tests and make assertions about your JavaScript code's behavior. Its flexibility, wide range of assertions, and integration with popular testing frameworks make it a popular choice for JavaScript testing.
  - Hardhat (JavaScript)
    Hardhat is a development environment and task runner for building and test-

ing Ethereum smart contracts and decentralized applications (DApps). It is designed to streamline the development workflow by providing a comprehensive set of tools and utilities specifically tailored for Ethereum development. Hardhat has gained popularity among Ethereum developers due to its flexibility, extensibility, and comprehensive toolset. It simplifies the development and testing of Ethereum-based applications, making it an excellent choice for building and deploying smart contracts and DApps on the Ethereum blockchain.

- Web3 (Python)
  Web3.py is a Python library that allows developers to interact with the Ethereum blockchain and build decentralized applications (DApps) using Python. It provides a convenient and user-friendly interface for working with Ethereum smart contracts, executing transactions, and interacting with blockchain data.
  Web3.py is a powerful tool for Python developers interested in building Ethereum-based applications. It simplifies the process of interacting with the Ethereum blockchain and enables seamless integration with existing Python codebases, making it a popular choice for Ethereum development in the Python ecosystem.
- TailwindCSS (CSS)
  Tailwind CSS is a highly customizable utility-first CSS framework that provides a comprehensive set of pre-built CSS classes for building user interfaces. Unlike traditional CSS frameworks that come with pre-designed components, Tailwind CSS focuses on providing low-level utility classes that can be combined to create custom designs and layouts.
  Tailwind CSS is known for its simplicity, flexibility, and performance. By using utility classes as building blocks, developers have fine-grained control over the design and layout of their applications. It is particularly suitable for projects that require custom designs, rapid prototyping, or highly responsive interfaces.

- **Ethereum Client**
  Geth, short for Go Ethereum, is one of the most widely used client implementations for the Ethereum blockchain. It is a command-line interface (CLI) tool and a full-featured Ethereum node written in the Go programming language. Geth allows users to interact with the Ethereum network, run their own Ethereum nodes, and deploy and execute smart contracts.
  Key features and functionalities of Geth include:

  1. Ethereum Node
  2. Blockchain Synchronization
  3. Smart Contract Execution
  4. Ethereum Account Management
  5. Transaction Submission and Mining
  6. Network Configuration
  7. JSON-RPC API

8. Developer Tools and Utilities

Geth is a versatile and powerful tool for interacting with the Ethereum blockchain. It is used by developers, miners, and other participants in the Ethereum ecosystem to run their own Ethereum nodes, deploy smart contracts, interact with the network, and contribute to the decentralized nature of the Ethereum blockchain.

- **Ethereum Dashboard**
  Ethstats is a real-time network monitoring and statistics platform for Ethereum. The Ethstats dashboard provides an overview of key metrics and information about the Ethereum network, including block propagation, mining activity, network health, and other relevant statistics. It aims to provide users with insights into the performance and status of the Ethereum network.
  The Ethstats dashboard offers various features and information, including:

  1. Network Statistics
  2. Mining Information
  3. Block Propagation
  4. Network Health Indicators
  5. Peer Information
  6. Historical Data
  7. Customization and Filtering

  The Ethstats dashboard provides a comprehensive view of the Ethereum network, allowing users to monitor network health, mining activity, block propagation, and other critical metrics. It serves as a valuable tool for Ethereum stakeholders, developers, miners, and enthusiasts to gain insights into the performance and status of the Ethereum blockchain in real-time.

9.4 Methodologies/Algorithm Details

To implement and test a blockchain based voting system, a few components have to be designed. A privately hosted closed blockchain network is required for our contract to run on. This is created using ethereum's Go language based client. To do so, a script is written in JavaScript to automate the following tasks :

1. Ethereum wallet accounts are created for the number of nodes required.
2. On these accounts, the genesis block is initialized.
3. Bootnode is initialized. Bootnode is a node which connects other nodes to each other.
4. Geth (Go Ethereum Client) is run using the accounts created using bootnode initialized in the previous step.

A Solidity based contract is written to execute the application's core functionality. This contract works like the backend for the entire application. This contract is written in solidity version 0.8.19 :

```
ElectoralContract {
struct Voter {
string voterName;
bool hasVoted;
bool isLoggedIn;
string voterPassword;
address nodeAddress;
uint timestamp;
}
struct Candidate {
string candidateName;
uint256 voteCount;
}
// func addCandidate : adds candidate for voters to vote
// func addVoter : adds voter
// func login : login functionality for voter
// func logout : logout functionality for voter
// func getResult : gets result of elections
// func getCandidates : gets list of candidate names
// func getVotersDetailed : gets all voters information
// func vote : cast vote to candidate functionality
};
```

Algorithm to cast vote :

```
Require  :  Voter must have not cast vote earlier
Require  :  Voter must be logged in
Update   :  candidate.voteCount++            //Update vote count for candidate
Update   :  voter.hasVoted = true            //Update voting status for voter
Update   :  voter.timestamp = timestamp      //Update voting timestamp for voter
Func     :  logout(voterId)                  //Logout voter
```

The overall flow followed is :

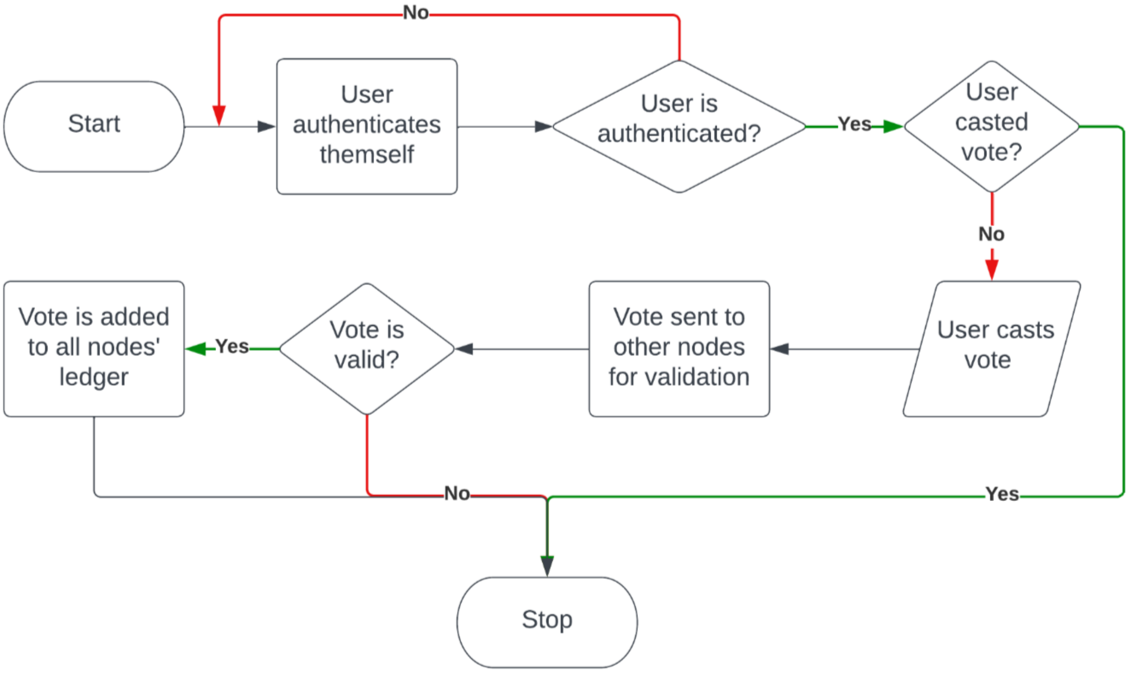

No
Start
User authenticates themself
User is authenticated?
Yes
User casted vote?
No
Yes
Vote is added to all nodes' ledger
Yes
Vote is valid?
Vote sent to other nodes for validation
User casts vote
No
Yes
Stop

Figure 9.1: Flow of the system

Consensus algorithm used in this chain is Proof of Authority (Clique). In PoA consensus, some nodes are given permission to generate new blocks. These nodes are called 'Validators'. PoA leverages the importance of identities. The validators are not staking coins like PoS but rather their own reputation instead. PoA is suited for private networks such as this one.

Finally, to evaluate the system, some synthetic data is generated and provided to the system and performances tested. To perform this evaluation, python script is written which will perform these tasks :

1. Register a single candidate.
2. Register required number of voters with random names.
3. Login with all voters' credentials.
4. Cast votes by voters with preset probability.
5. Logout all logged in voters.
6. Get result and compile csv.
7. Plot graphs from results.

Along with these components, a frontend client is made for a citizen/voter to interact and cast their votes. Dashboards are also deployed to monitor the network and the data flowing through it. The system's performance is tested by generating synthetic data which is passed to the system and the time taken for it to process the data and store it in blockchain. The number of nodes in the system are scaled in range from 1 to 50 while ranging the voters from 10 to 250.

9.5 Verification and Validation

Following table shows time taken in seconds by the system to insert data to the private blockchain network and generate results without artificial delay included.

		Number of nodes									
		1	2	3	4	5	10	15	20	25	50
Voters	10	0.46	7.08	2.4	5.89	20.26	6.27	6.76	10.15	5.94	4.84
	20	0.9	13.56	3.75	5.97	35.4	13.16	17.4	14.87	12.44	56.11
	30	1.33	19.15	6.16	10.16	45.24	81.77	22.4	19.29	19.11	41.7
	40	1.76	8.88	9.25	13.58	61	31.21	30.11	31.57	29.68	33.01
	50	2.3	35.77	11.73	15	65.16	43.83	39.35	37.42	32.23	28.8
	100	4.61	22.82	21.81	30.53	139.74	87.84	71.14	77.71	70.99	111.91
	150	7.93	35.41	34.93	45.02	166.75	100.94	113.08	112.71	111.59	161.79
	200	11.18	49.99	46.33	72.57	-	167.5	143.24	164.77	155	273.67
	250	13.34	152.75	59.7	105.66	-	191.49	169.56	189.36	184.24	209.19

Table 9.1: Time Taken - (Number Of Nodes Vs Voters)

During evaluation of the system, we faced many failures due to block creation taking a long time, blockchain not being synchronized properly across the network, running out of resources for mining and some unexpected system crashes. Due to these crashes, results for 5 nodes, 200 and 250 voters could not be obtained.

CHAPTER 10
SOFTWARE TESTING

10.1 Test Plan

Software testing is an activity that helps in finding out bugs/defects/errors in a software system under development, in order to provide a bug free and reliable system/solution to the customer. Testing of software is always carried out in two parts viz.. Verification and validation. Verification refers to the set of activities that ensure that software correctly implements a specific function. Validation refers to a different set of activities that ensure that the software has been built is traceable to customer requirements.

Software scope describes the functions and features that are to be delivered to end users, the data that are input and output, and the content that is presented to the users as consequence of using the software. Test plan for this project involves the testing of the individual components for modules interface.

The software is tested using two levels of testing viz. black box testing and white box testing. White box testing could be carried out in three different phases viz. unit testing system/ integration testing and validation testing.

10.2 Test Cases and Test Result

10.2.1 Smoke Testing

Whenever a new build is provided by the development team then the software testing team validates the build and ensures that no major issue exists. The testing team ensures that the build is stable and a detailed level of testing is carried out further. Smoke Testing checks that no show stopper defect exists in the build which will prevent the testing team to test the application in detail. If testers find that the major critical functionality is broken down at the initial stage itself then testing team can reject the build and inform accordingly to the development team. Smoke Testing is carried out to a detailed level of any functional or regression testing.

10.2.2 Sanity Testing

Sanity Testing is done to determine if a new software version is performing well enough to accept it for a major testing effort or not. If an application is crashing for the initial use then the system is not stable enough for further testing. Hence a build or an application is assigned to fix it.

10.2.3 Unit Testing

Unit Testing, also known as Module Testing, focuses verification efforts on the module. The module is tested separately and this is carried out at the programming stage itself. Unit Test comprises the set of tests performed by an individual programmer before integration of the unit into the system. Unit test focuses on the smallest unit of software design- the software component or module.

10.2.4 Integration Testing

It is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with the interface. It takes the unit tested modules and builds a program structure. All the modules are combined and tested as a whole. Integration of all the components to form the entire system and an overall testing is executed.

10.2.5 Regression Testing

Testing an application as a whole for the modification in any module or functionality is termed as Regression Testing. It is difficult to cover all the systems in Regression Testing, so typically automation testing tools are used for these types of testing.

10.2.6 Validation Testing

Validation tests succeed when the software functions in a manner that can be reasonably expected by the client. Software validation is achieved through a series of black box testing which confirms the requirements. Black box testing is conducted at the software interfaces. The test is designed to uncover interface errors, is also used to demonstrate that software functions are operational, input is properly accepted, output is produced, and that the integrity of external information is maintained. Both the plan and procedure are designed to ensure that all functional requirements are satisfied, all behavioral characteristics are achieved, all performance requirements are attained, documentation is correct, and human engineered, and other requirements are met.

10.2.7 System Testing

Tests to find the discrepancies between the system and its original objective, current specifications and system documentation. The system software is tested as a whole. It verifies all elements mesh properly to make sure that all system functions and performance are achieved in the target environment.

The focus areas are:

- System functions and performance.
- System reliability and recoverability (recovery test).
- System behavior in the special conditions (stress and load test).
- System user operations (acceptance test/alpha test).
- Hardware and software integration collaboration.
- Integration of external software and the system.

10.2.8 Output Testing

Output of test cases compared with the expected results during design of test cases. Asking the user about the format required by them tests the output generated or displayed by the system under consideration. Here, the output format is considered into two ways, one is on screen and another one is printed format.

The output on the screen is found to be correct as the format was designed in the system design phase according to user needs. The output comes out as the specified requirements as the user's hardcopy.

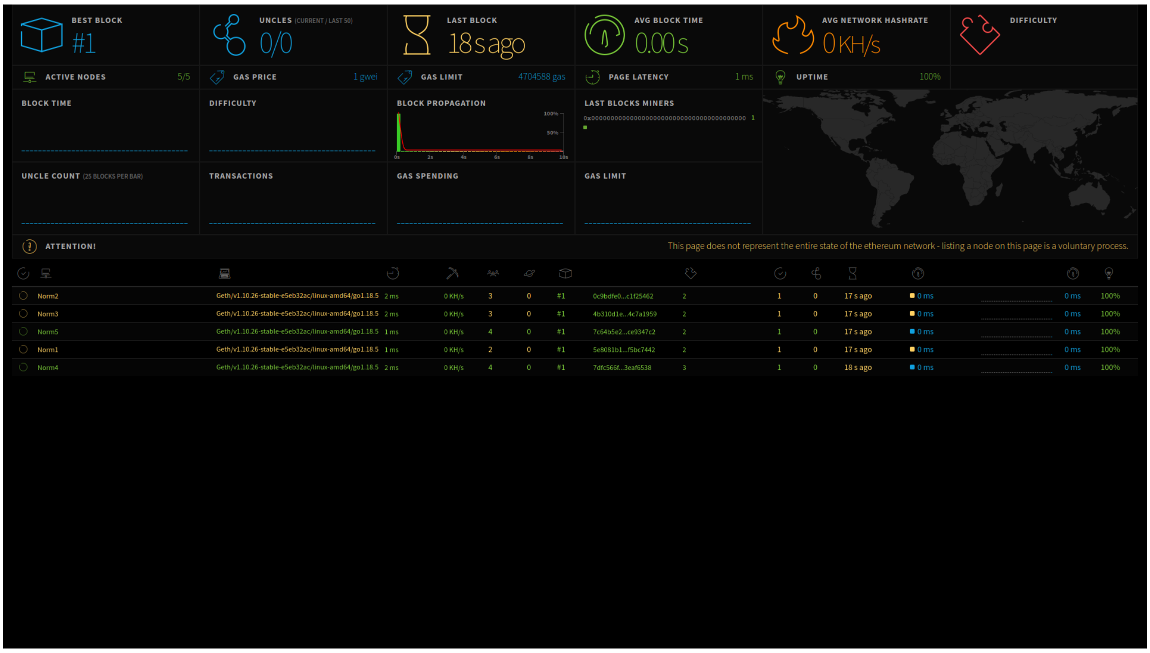
10.2.9 Performance Testing

Performance testing determines the amount of execution time spent in various parts of the unit, program throughput, and the response time and device utilization of the program unit. It occurs throughout all steps in the testing process.

10.2.10 GUI Testing

Graphical User Interface (GUIs) present interesting challenges for software engineers. Because of reusable components provided as part of GUI development environments, the creation of the user interface has become less time consuming and more precise.

CHAPTER 11
SCREENSHOTS



BEST BLOCK #1

UNCLES (CURRENT / LAST 50) 0/0

LAST BLOCK 18 s ago

AVG BLOCK TIME 0.00 s

AVG NETWORK HASHRATE 0 KH/s

DIFFICULTY

ACTIVE NODES 5/5

GAS PRICE 1 gwei

GAS LIMIT 4704588 gas

PAGE LATENCY 1 ms

UPTIME 100%

BLOCK TIME

DIFFICULTY

BLOCK PROPAGATION

LAST BLOCKS MINERS

0x0000000000000000000000000000000000000000 1

UNCLE COUNT (25 BLOCKS PER BAR)

TRANSACTIONS

GAS SPENDING

GAS LIMIT

ATTENTION! This page does not represent the entire state of the ethereum network - listing a node on this page is a voluntary process.

Norm2	Geth/v1.10.26-stable-e5eb32ac/linux-amd64/go1.18.5	2 ms	0 KH/s	3	0	#1	0c9bdfe0...c1f25462	2	1	0	17 s ago	🟨 0 ms	0 ms	100%
Norm3	Geth/v1.10.26-stable-e5eb32ac/linux-amd64/go1.18.5	2 ms	0 KH/s	3	0	#1	4b310d1e...4c7a1959	2	1	0	17 s ago	🟨 0 ms	0 ms	100%
Norm5	Geth/v1.10.26-stable-e5eb32ac/linux-amd64/go1.18.5	1 ms	0 KH/s	4	0	#1	7c64b5e2...ce9347c2	2	1	0	17 s ago	🟦 0 ms	0 ms	100%
Norm1	Geth/v1.10.26-stable-e5eb32ac/linux-amd64/go1.18.5	1 ms	0 KH/s	2	0	#1	5e8081b1...f5bc7442	2	1	0	17 s ago	🟨 0 ms	0 ms	100%
Norm4	Geth/v1.10.26-stable-e5eb32ac/linux-amd64/go1.18.5	2 ms	0 KH/s	4	0	#1	7dfc566f...3eaf6538	3	1	0	18 s ago	🟦 0 ms	0 ms	100%

Figure 11.1: Ethstats

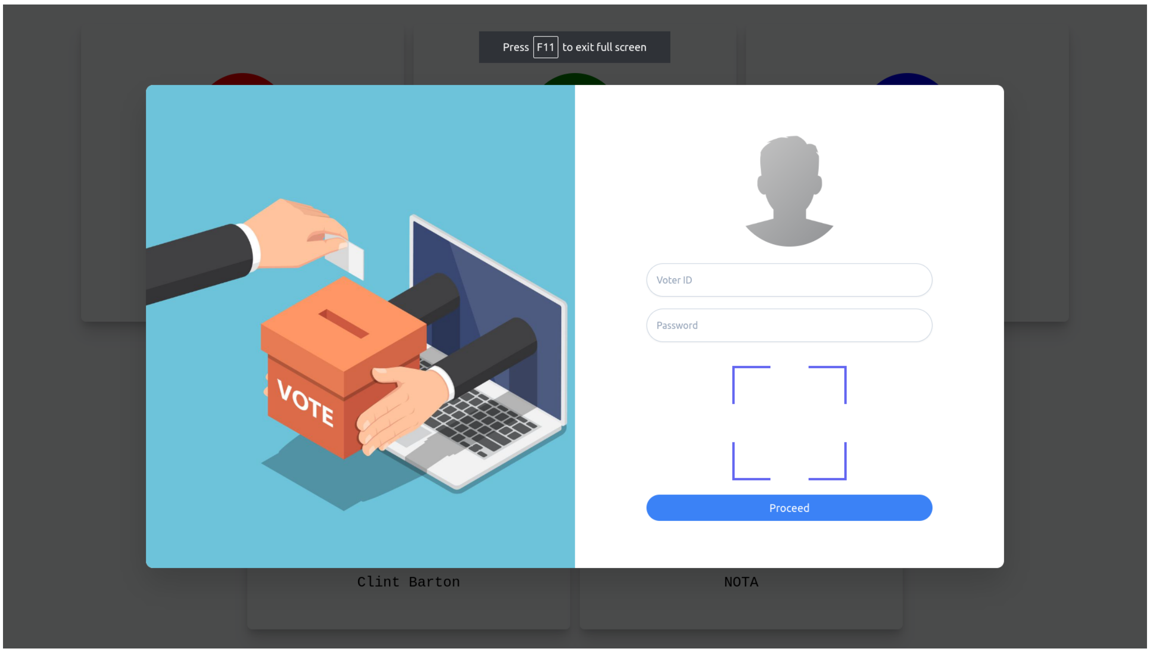


Figure 11.2: Login

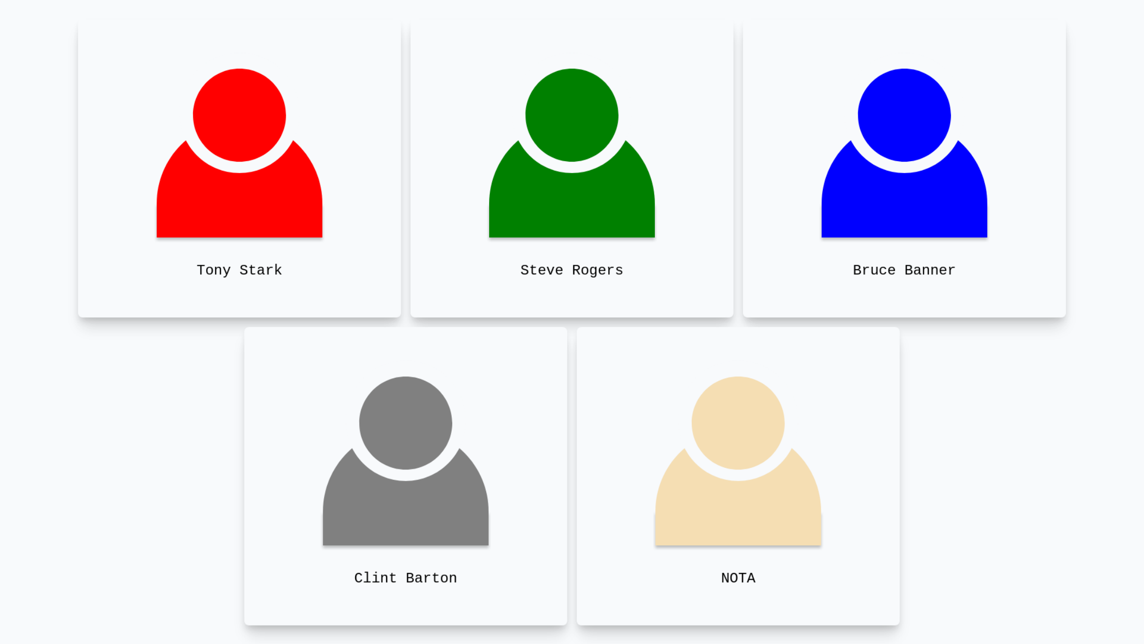


Figure 11.3: Before selecting candidate

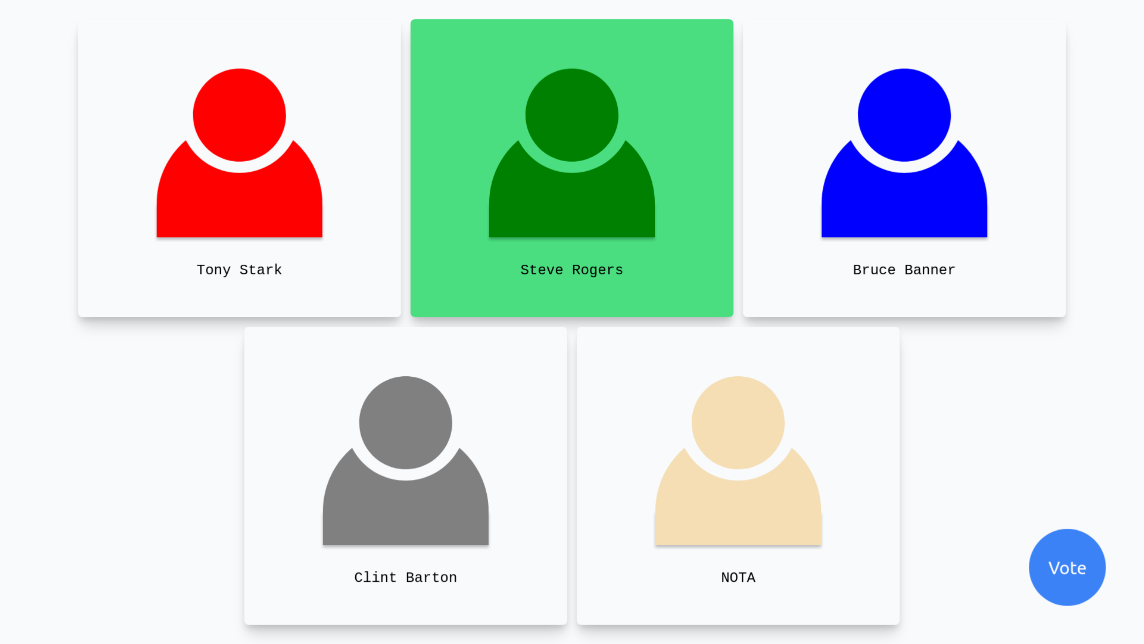
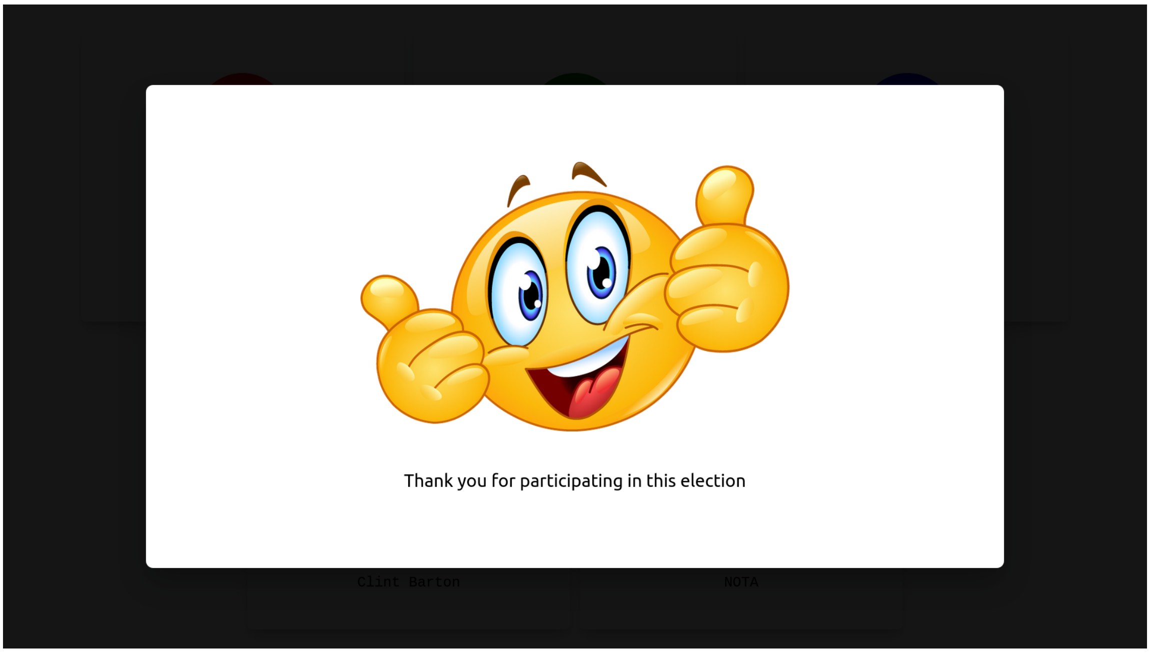
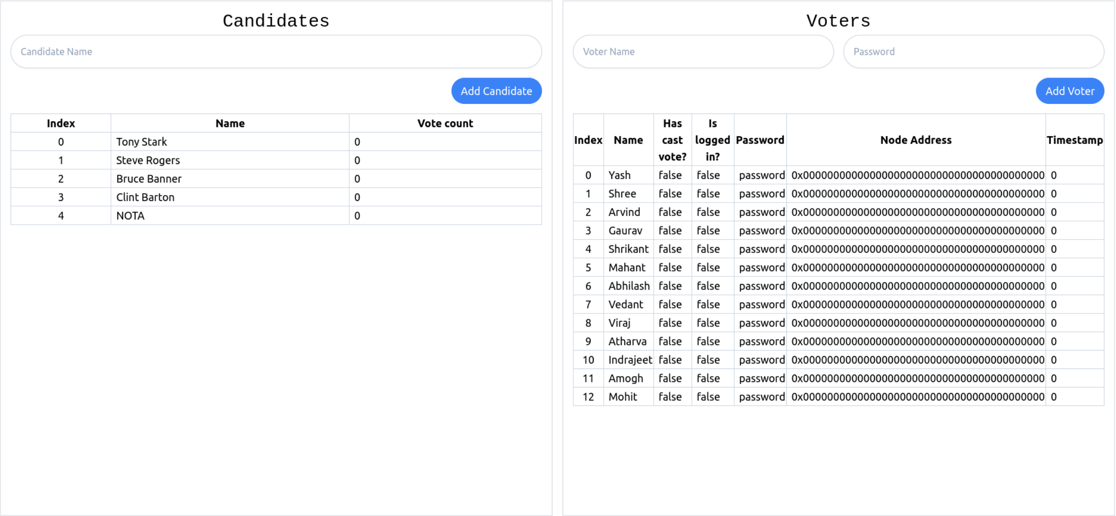


Figure 11.4: After selecting candidate



Thank you for participating in this election

Figure 11.5: After casting vote



Index	Name	Vote count
0	Tony Stark	0
1	Steve Rogers	0
2	Bruce Banner	0
3	Clint Barton	0
4	NOTA	0

Index	Name	Has cast vote?	Is logged in?	Password	Node Address	Timestamp
0	Yash	false	false	password	0x0000000000000000000000000000000000000000	0
1	Shree	false	false	password	0x0000000000000000000000000000000000000000	0
2	Arvind	false	false	password	0x0000000000000000000000000000000000000000	0
3	Gaurav	false	false	password	0x0000000000000000000000000000000000000000	0
4	Shrikant	false	false	password	0x0000000000000000000000000000000000000000	0
5	Mahant	false	false	password	0x0000000000000000000000000000000000000000	0
6	Abhilash	false	false	password	0x0000000000000000000000000000000000000000	0
7	Vedant	false	false	password	0x0000000000000000000000000000000000000000	0
8	Viraj	false	false	password	0x0000000000000000000000000000000000000000	0
9	Atharva	false	false	password	0x0000000000000000000000000000000000000000	0
10	Indrajeet	false	false	password	0x0000000000000000000000000000000000000000	0
11	Amogh	false	false	password	0x0000000000000000000000000000000000000000	0
12	Mohit	false	false	password	0x0000000000000000000000000000000000000000	0

Figure 11.6: Admin Dashboard

CHAPTER 12
DEPLOYMENT AND MAINTENANCE

12.1 Deployment

- When the system meets approval and is ready to go live, we deploy it to the production environment and assist the transition to the new system.
- The development platforms we use support a broad range of deploymentoptions.
- There is no need to install any software on end-users' machine since we plan to convert this application into executable file format.

12.2 Maintenance

- Systems evolve after deployment. We will continue to make improvements and changes to our system as needed.
- With Survey, delivering the latest revisions to the users will be our priority, and will be completely transparent to the users.
- We can also train the user to maintain the system.

CHAPTER 13
CONCLUSION AND FUTURE SCOPE

13.1 Conclusion

Through the numerous trial runs and performance testing, we were able to make a few observations. These observations were made based on the performance table in the Result section and self experiences while evaluating the system.

1. As the number of voters increases a gradual increase is observed in time taken to insert transactions and get results.
2. The time taken also increases as the number of nodes in the system is increased. This is due to blockchain synchronization across the network.
3. Less number of nodes showed instability and frequent crashes since load not being distributed well enough. As the nodes increased, the stability increased and less crashes were observed.

From the observations, we can say that voting using blockchain technology is feasible and will be beneficial due to its security and distributed nature. Since voting is carried on large scales, it will be stable and solve few issues faced by the current voting system.

13.2 Future Scope

To improve accuracy we need to increase the number of nodes. Increasing the number of nodes also affects the speed of the system but increases security. Balancing all these characteristics and producing results in optimal time would be the aim of the system in future. Also hardware resource optimization is also the aim. It makes the blockchain environment friendly and causes less harm to the environment. Moreover hardware optimization helps to include nodes with medium to less computing capacity to participate in mining. This increases miners and would help the spread of the chain.

CHAPTER 14
REFERENCES

Bibliography

[1] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *International journal of web and grid services*, vol. 14, no. 4, pp. 352–375, 2018.

[2] M. N. M. Bhutta, A. A. Khwaja, A. Nadeem, H. F. Ahmad, M. K. Khan, M. A. Hanif, H. Song, M. Alshamari, and Y. Cao, "A survey on blockchain technology: Evolution, architecture and security," *Ieee Access*, vol. 9, pp. 61048–61073, 2021.

[3] W. Viriyasitavat and D. Hoonsopon, "Blockchain characteristics and consensus in modern business processes," *Journal of Industrial Information Integration*, vol. 13, pp. 32–39, 2019.

[4] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *2017 IEEE international congress on big data (BigData congress)*, pp. 557–564, Ieee, 2017.

[5] L. M. Bach, B. Mihaljevic, and M. Zagar, "Comparative analysis of blockchain consensus algorithms," in *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 1545–1550, Ieee, 2018.

[6] Q. Zhou, H. Huang, Z. Zheng, and J. Bian, "Solutions to scalability of blockchain: A survey," *Ieee Access*, vol. 8, pp. 16440–16455, 2020.

[7] J. V. Cadiz, N. A. M. Mariscal, and A. M. Ceniza-Canillo, "An empirical analysis of using blockchain technology in e-voting systems," in *2021 1st International Conference in Information and Computing Research (iCORE)*, pp. 78–83, IEEE, 2021.

[8] M. S. Farooq, U. Iftikhar, and A. Khelifi, "A framework to make voting system transparent using blockchain technology," *IEEE Access*, vol. 10, pp. 59959–59969, 2022.

[9] A. M. Al-Madani, A. T. Gaikwad, V. Mahale, and Z. A. Ahmed, "Decentralized e-voting system based on smart contract by using blockchain technology," in *2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing (ICSIDEMPC)*, pp. 176–180, IEEE, 2020.

[10] C. Angsuchotmetee, P. Setthawong, and S. Udomviriyalanon, "Blockvote: An architecture of a blockchain-based electronic voting system," in *2019 23rd International Computer Science and Engineering Conference (ICSEC)*, pp. 110–116, IEEE, 2019.

[11] A. Benabdallah, A. Audras, L. Coudert, N. El Madhoun, and M. Badra, "Analysis of blockchain solutions for e-voting: A systematic literature review," *IEEE Access*, 2022.

[12] Z. Khudoykulov, U. Tojiakbarova, S. Bozorov, and D. Ourbonalieva, "Blockchain based e-voting system: Open issues and challenges," in *2021 International Conference on Information Science and Communications Technologies (ICISCT)*, pp. 1–5, IEEE, 2021.

[13] U. Jafar, M. J. A. Aziz, and Z. Shukur, "Blockchain for electronic voting system—review and open research challenges," *Sensors*, vol. 21, no. 17, p. 5874, 2021.

CHAPTER 15
ANNEXURE A
LABORATORY ASSIGNMENTS ON PROJECT ANALYSIS OF ALGORITHMIC DESIGN

15.1 Introduction

15.2 Objective

15.3 Architectural Design

CHAPTER 16
ANNEXURE B
LABORATORY ASSIGNMENTS ON PROJECT QUALITY AND RELIABILITY TESTING OF PROJECT DESIGN

CHAPTER 17
ANNEXURE C
PROJECT PLANNER

CHAPTER 18
ANNEXURE D
PLAGARISM REPORT

CHAPTER 19
ANNEXURE E
TERM II - PROJECT LABORATORY ASSIGNMENTS

CHAPTER 20
ANNEXURE F
INFORMATION OF PROJECT FROUP MEMBERS