DEPARTMENT OF COMPUTER SCIENCE ENGINEERING,

SCHOOL OF ENGINEERING AND TECHNOLOGY,

SHARDA UNIVERSITY, GREATER NOIDA

Internet Voting System Using Blockchain

A project submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Technology in Computer Science and Engineering

by

Pranav Kumar (2018008769)
Praveen Anand (2018011512)
Shubh Shukla (2018006057)
Paras Gera (2018005034)

Supervised by:
Ms. Sonam Nagar, Asst. Professor (CSE)

May, 2022

CERTIFICATE

This is to certify that the report entitled "**Internet Voting System Using Blockchain**" submitted by Pranav Kumar (2018008769), Praveen Anand (2018011512), Shubh Shukla (2018006057), Paras Gera (2018005034) to Sharda University, towards the fulfillment of requirements of the degree of "**Bachelor of Technology"** is a record of bonafide final year Project work carried out by him/her in the Department of Computer Science and Engineering, School of Engineering and Technology, Sharda University. The results/findings contained in this Project have not been submitted in part or full to any other University/Institute for an award of any other Degree/Diploma.

Signature of Supervisor

Name: Ms. Sonam Nagar

Designation: Asst. Professor (CSE)

Signature of Head of Department

Name: Prof. (Dr.) Nitin Rakesh

Place: Sharda University

Date:

Signature of External Examiner

Date:

ACKNOWLEDGEMENT

A major project is a golden opportunity for learning and self-development. We consider ourselves very lucky and honored to have so many wonderful people lead us through the completion of this project.

First and foremost, we would like to thank Dr. Nitin Rakesh, HOD, CSE who gave us an opportunity to undertake this project.

My grateful thanks to **Ms. Sonam Nagar** for their guidance in our project work.

Ms. Sonam Nagar, who in spite of being extraordinarily busy with academics, took time out to hear, guide, and keep us on the correct path. We do not know where we would have been without her help.

CSE department monitored our progress and arranged all facilities to make life easier. We choose this moment to acknowledge their contribution gratefully.

Name and signature of Students

Pranav Kumar (2018008769)
Praveen Anand (2018011512)
Shubh Shukla (2018006057)
Paras Gera (2018005034)

Abstract

Every country's citizens have the right to vote. An Internet voting system is one in which the election process is digitally notated, recorded, saved and processed making vote management easier than with the traditional paper-based method. Blockchain is opening up new possibilities for the creation of new sorts of digital services. While research on the subject is still in its early stages, it has mostly concentrated on technological and legal difficulties rather than utilizing this innovative notion to develop enhanced digital services. BEV (blockchain-enabled e-voting) has the potential to eliminate election fraud while also increasing voter access. Eligible voters used a computer or smartphone to vote anonymously. BEV employs a tamper-proof personal ID and an encrypted key. Electronic credibility services have evolved into a necessary component of the information landscape. It's possible to create more complex systems that rely on them, such as an electronic voting system, with the reliable execution of vital services like digital signature and authentication.

Idea for developing a system of electronic voting utilizing blockchain technology applied in this project. The two-level design ensures safe voting without relying on current (non-blockchain) technologies for redundancy. To make the whole project integrated and function together, the blockchain-based voting project contains two parts:

(a) The Election Commission is one among, which in charge of putting together elections, as well as registering parties and people to run for office using smart contracts.
(b) The voter's module, on the other hand, will allow each individual to cast a ballot for their Assembly District, which the transaction will be recorded on the blockchain and hence secured.

Index Terms- voting on the web, security, digital voting on the distributed ledgers, confidentiality, smart contracts.

Table of Contents

List of Tables

List of Figure

Chapter-1: Introduction

Not just for democratic countries, but also for state voters' trust and accountability, election data is essential. In this sense, democratic voting procedures is critical. Electronic voting is a beneficial move for the government has the potential to boost voter engagement and trust while also reigniting interest in the voting system. Elections have long been a societal issue as a viable mechanism for making democratic decisions. People of India increasingly conscious importance of the system of elections when the number of real-life votes increases [1,2]. Judges choose who will represent them in political and commercial governance using a voting system. Democracy is a system in which citizens vote to election representatives [3,4]. The effectiveness of such a strategy is largely driven by people's faith in the electoral process. It is a well-known trend to create legislative organizations that will represent the people will among public. Student unions and constituencies are examples of such political groups. Over time, the vote has evolved into the principal means of voters expressing their will by picking from a menu of options [2].

Traditional or paper-based polling methods boosted the public's belief in the majority vote choices. It really has aided the democratic process as well as the electoral system for electing constituencies and governments. All of them are either completely faulty or a mix of the two. [5,6]. Since the commencement of the voting system, the secret voting concept has been utilised to foster trust in democratic systems. It is vital to ensure that voter trust does not decrease. Research found conventional voting procedure isn't entirely sanitary, raising various problems about justice, equality, and the people's will, all of which were not fully defined and comprehended in the form of governance.

New voting technologies have been created by experts from across the world that give some anti-corruption protection while retaining vote accuracy. New electronic voting procedures and approaches have been developed as a result of technological advancements, which is crucial and poses substantial democratic system is under attack. Electronic voting increases election reliability as compared to human polling. In comparison to past voting methods, it has increased the process' efficiency and integrity [10]. Because of its versatility, simplicity of use, and cheap cost compared to general elections, electronic voting is commonly employed in a variety of situations [11]. Despite this, current electronic voting methods run the danger of altered data and over-authority, compromising voting fairness, secrecy, anonymity, and transparency.

In contrast, electronic voting processes use a single controller to manage the entire voting procedure [12]. Due to the central authority's dishonesty, the process leads to erroneous picks that are impossible to correct using current approaches (election commission). A decentralised network might be utilised as a contemporary electronic voting mechanism to avoid relying on the central authority.

For online or electronic voting, technology offers a decentralised node. Because of its end-to-end verification capabilities, distributed ledger technologies like blockchain have recently been employed to construct electronic voting systems [13]. Decentralization, non-repudiation, and security protection are just a few of the advantages that blockchain offers over traditional 10 electronic voting methods. It is suitable for both private and public voting [8]. A blockchain is a

growing collection of cryptographic links between blocks that began as a chain of blocks and is still developing. Each block includes the preceding block's hash, timestamp, and transaction data. When the blockchain was built, security was a top objective. Voting is a new phase of blockchain technology, and academics are attempting to capitalise on features such as transparency, confidentiality, and nonrepudiation, which are all required for voting [14]. Efforts to use blockchain technology to protect and repair elections have lately attracted a lot of attention [15] because to its use in electronic voting systems.

1.1: Motivation

The goal of this project is to create a safe voting environment and demonstrate that blockchain technology can be used to build a trustworthy Internet voting system. Because if everyone with a computer or a mobile phone has access to the Internet, people and members will be able to make whatever administrative decision they want and at the very least, people viewpoints will be more apparent and accessible to politicians and executives. It'll be eventually just for humanity, lead to true direct democracy [6]. It's crucial for us because elections are readily tainted or influenced, particularly in small towns and even larger cities in corrupt governments. Furthermore, traditional elections on a broad scale are exceedingly expensive in the long run, especially when there are hundreds of polling locations throughout a vast geographic area and millions of voters [7]. Voter turnout at polling places is low because people may not live at the address where their name is on the list, or they may be away on vacation or at another job. If done correctly, I-voting would be able to solve these problems. The notion of electronic voting is far older than blockchain. As a result, well instances up to this point have relied on centralized computation and storage structures.

Estonia serves as an excellent example, as the Estonian government was one of the first to developed completely system of internet voting that is both online and offline [8]. Electronic voting was first discussed in the country in 2001, and it was formally implemented by the government in the summer of 2003 [9]. Their system is still in use, with several enhancements and adjustments to the basic concept. It is now highly robust and dependable, as reported. For person-to-person identification, they employ smart digital ID cards and personal card readers (provided by the government) [10]. There is a dedicated web site as well as a desktop software for residents to participate in elections by listing candidates and voting.

On the website (http://rahvaalgatus.ee), anyone may also generate digital petitions and recommendations for actions and laws. Any person who wants to support the proposal can use their smart ID card to digitally sign the petitions. Proposals that get a certain number of signatures are discussed in parliament. Another fantastic illustration of how technology may promote democracy is this. The Estonian model, despite its considerable success and approximately 30% penetration rate during recent elections, is not without flaws. A centralized solution, by its very nature, has a single point of failure and is susceptible to trying to hack attempts. Assaults, for example: might cause damage to the software, servers or databases that are used. During an election, the administrators of such a system may behave maliciously and

steal, if they are unable to influence some vital information. Another concern is the system's scalability. Because Estonia has a tiny population, it's difficult to predict if such a system would operate perfectly developed by China. The continual requirement for an ID card and scanner equipment is also inconvenient, given the additional costs of making, distributing, and carrying (for voters).

Switzerland is another of the few countries that have embraced the trend of computerized voting. Every person over the age of 18 in Switzerland, which is noted for its extensive democracy, has the option of participating actively or passively in elections, which can be held on a variety of issues and for a variety of purposes. They've also started working on a voting mechanism known as remote voting [11]. For example, at Sierra Leone's general election in March 2018, Swiss start-up Agora tallied votes in two districts. Following the election, a team of certified observers from several places manually input almost 400,000 ballots into the blockchain system. Similar commercial or experimental work was carried out in the Russian city of Moscow for its Active Citizen initiative, which included the partial deployment of blockchain and the verification of votes via blockchain. In December of this year. For voting and to make the vote results publicly auditable, the program began employing a blockchain. Each subject that is debated by the community and put up for a vote is moved to a blockchain-based e-voting system. The outcomes of the vote are recorded in a ledger that contains all of the previous polls [10].

http://www.strawpoll.me/ is a popular and free online polling service that may be used instead of an Internet-voting system. Straightforward a site that enables everyone may create surveys and vote. Demonstrates the effectiveness of Internet-voting by allowing anybody to easily access the election, cast their ballots, and declare their preferences. People can share private URLs to any generated poll (as long as they know the link), and everyone who has the link can vote, but only one browser can vote. In terms of voter verification, duplicate votes, and non-repudiation of votes, the security here is pretty poor. http://www.strawpoll.me/ has faith in people to not tamper with the election process while benefiting from ease of access and e-voting capabilities. As a result, it can't be used in real-life scenarios such as choosing a department chairman [2].

1.2: Overview

Modern democracies are built on the pillars of traditional balloting and electronic voting (e-voting). Electronic voting machines, or EVMs have been under fire in recent years due to inaccurate election results reporting. Many concerns have been expressed concerning the design and internal architecture of these devices, as well as their vulnerability to attacks. Several ways for tampering with EVMs were investigated in this study [1]. The use of online voting is being marketed as a way to recruit young people and non-citizens to the country. A variety of security and performance requirements must be addressed for a trustworthy online election system, including transparency, correctness, auditability, data privacy and so on.

We developed the following concepts using two separate sets of modules: electoral commission and voter registrations. The Election Commission sets up elections and invites registered individuals and political parties to run for office. The specifics of an election are displayed to the

front-end of the voter for casting the vote using an election's REST API housed on Ethereum's Blockchain. The vote is then saved on our blockchain infrastructure during polling from which the Election Commission retrieves the vote total. The issue we've encountered as a result of not utilizing the standard method of smart contracts is that the blockchain framework we've created can't operate on the main net since it needs to be hosted, therefore we'll have to utilize a separate web3 provider to be utilized for dealing with it and the lack of a public API for voter ID provides the disadvantage of voter authentication. The most crucial aspect of this application is the smooth integration of a blockchain framework with both modules for voting.

1.3: Expected outcome

The expected outcome of the internet voting system using blockchain is to provide a reliable system that will be providing security and privacy to the system and reduces the workload of setting up an election center. The system's main goal is to: -

- Technology to improve the present online voting system.
- To decrease the time and effort required to set up a voting booth and conduct physical elections.
- Voting system process is entirely online.
- We are expected to understand the concept of Blockchain and how it may be applied to various industries.

1.4: Gantt Chart

A Gantt chart is a type of bar chart that is used to show project schedules. Any project involving effort, resources, milestones, or delivery dates can benefit from the use of Gantt charts. Gantt charts are now the preferred method of project management in all fields. Gantt charts help project managers to keep track of the whole project's progress. The project manager may keep track of individual activities as well as the overall project progress using Gantt charts.

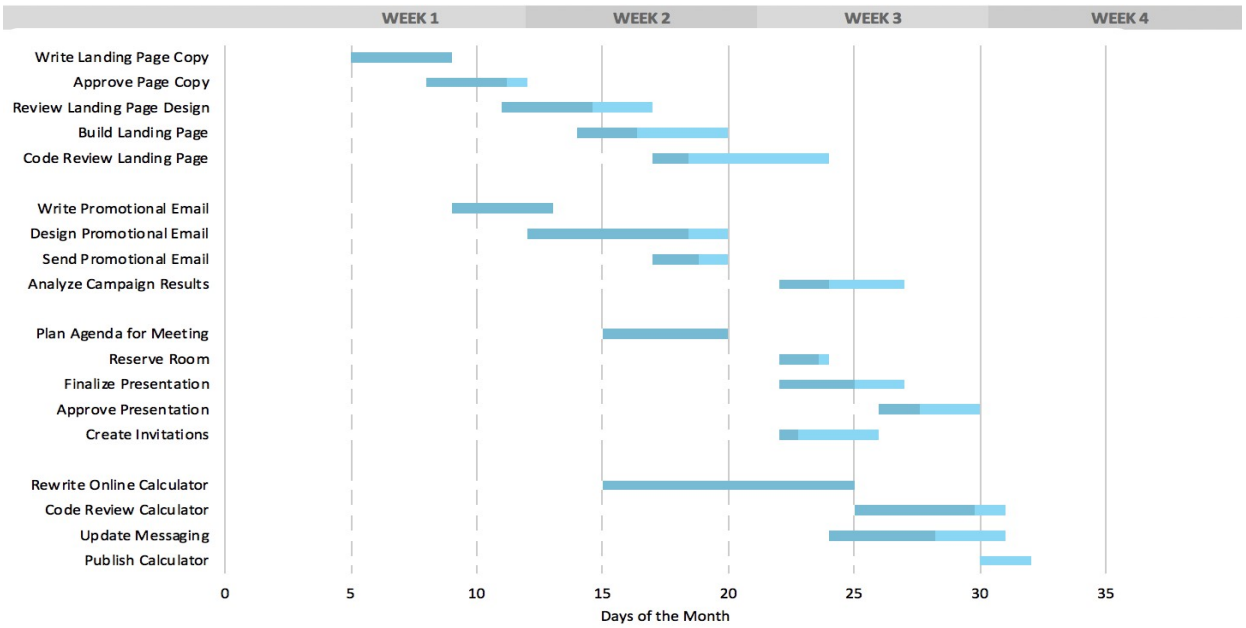


Figure 1: Gantt Chart

Following is the resource allocation table:

Table 1: Resource Allocation

Task Name	Start Date	Duration	Progress
Project	25/04/2021	80d	100%
Getting Started	25/04/2021	12d	100%
Analysis	25/05/2021	5d	100%
Requirement Gathering	03/06/2021	7d	100%
System Design and Architecture	12/06/2021	25d	100%
Detailed Analysis and Requirement Gathering	12/06/2021	10d	100%
Prepare Design Documentation	26/06/2021	15d	100%
Implementation	17/07/2021	37d	100%
Research Paper	25/08/2021	20d	100%
Setting up the environment	16/09/2021	5d	100%
Development	23/09/2021	25d	100%

Quality Testing	28/10/2021	7d	100%
Deployment	08/02/2022	6d	100%
Setup Production	08/02/2022	4d	100%
Release	12/02/2022	2d	100%

1.5: Possible risks

Possible risks for the development of the Internet Voting System Using Blockchain tampering of the vote and easy-to-use interface of the software, provide privacy and security. The main factors are: -

- **Only valid voter**: It should be possible to vote only if you are a registered voter.
- **Unreusability**: Each voter is only allowed to vote once.
- **Confidentiality**: No one else has access to information about the people's choice save the voter.
- **Objectivity**: There is no method to receive results from intermediate votes.
- **Validity**: During the counting process, invalid ballots should be identified and not counted.
- **Accuracy of tally**: All lawful ballots shall be counted in accordance with the law.

1.6: SRS (Software Requirement Specification)

The Online Election System utilising Blockchain software package and supporting documentation employ the terminology, acronyms, and abbreviations listed below: **-**

Table 2: Definitions and Abbreviations

Abbreviations	Definitions
EC	Election Commission
ETH	Ethereum
API	Application Programming Interface
IDE	Integrated Development Environment
JSON	JavaScript Object Notation
SRS	Software Requirement Specifications
SDLC	Software Development Life Cycle
STLC	Software Testing Life Cycle

1.7: Literature Review

Authentication of voters:

Authenticating voters can be done in a variety of ways. Voter verification, according to Kriti Patidar and Dr. Jain, may be done using private key cryptography, which must be provided to voters prior to the voting process. Some authority should register voters at the same time, and keys should be produced and delivered to voters on the spot (2019).

Cosmas Krisna Adiputra (2018) describes a public-private key architecture in which an election key pairs (PE; SE) is generated by the electoral commission (or another election administration) and then used to encrypt and decode voter communications. After that, each voter must create their own key pair. The key pair of Voter X is indicated by (PV X; SV X). After that, the voter's signature is added to the message she produced herself using this key combination.

Voters must register their public key PV X with the election commission using a valid ID in order to vote. The electoral commission next checks each voter's identification and either accepts or rejects the linked public key PV X if the person is ineligible. It is critical that each voter retain their public key private and only share it with the governing body.

The technology recognises and validates each participant by providing an electronic ID from Auokenni (2019) and the matching 6-digit PIN at the voting booth. Fririk. Hjálmarsson has a different idea; he wants to utilise a 6-digit pin that voters may use to verify their identity. If there isn't any regulation, things will get out of hand.

Roopak (2020) proposed an innovative approach for storing voter information that included the Aadhar database. The proposed framework is an electronic voting system that makes use of a UIDAI-issued unique virtual ID. The Aadhar database is used to collect demographic and fingerprint data on voters. The fingerprint is converted to a digital signature during the encryption process, which may be used to confirm the vote in the block is secure.

Anonymity is one of the most significant voting requirements, as others are unable to observe how someone voted. Individuals must however be qualified to vote and there must be a system in place to verify this. It's challenging to strike a balance between these two requirements. We want the user to be able to see that their vote has been recorded on the blockchain, but we don't want anybody else to be aware of what's going on since it makes it harder to guarantee that the voting is fair.

Nonetheless, several nations are pushing ahead with efforts to implement blockchain voting, like Brazil [13], which stores election data on the Ethereum blockchain. The effort of collecting and validating the information of over 145 million registered voters is enormous. As a result, various problems must be addressed in order to perform entirely blockchain-based e-voting. Biometric technology such as face comparison, fingerprint, Iris, and retinal scan have been used to verify voter identity in some studies, although they are potentially inaccurate and readily manipulated or stolen. However, we believe that using advanced algorithms that are difficult to break is one

strategy to safeguard stolen biometrics data. Instead of preserving the biometric information as binary data and then storing it as a reference string, it can be hashed using any hashing technique. During the validation and identification process, the sample model should be transformed to a hash value and then compared to the reference value.

Table 4. Literature Survey

S. No	Name of paper	Technique
1.	Survey on Blockchain Based E-Voting Recording System Design	1. AES algorithm
2.	Online Voting: Voting System Using Blockchain	1.Cryptographic verification 2. Homomorphic Encryption Technique:
3.	Blockchain-Based E-Voting System	1.Quorum 2.Geth: Go-Ethereum
4.	Blockchain Based E-Voting Recording System Design	1.ECDSA (Elliptic Curve Digital Signature Algorithm) 2. SHA-256 algorithm
5.	Decentralized Voting Platform Based on Ethereum Blockchain	1.HTML5 web-app compiled using Apache Cordova 2. Ethereum network
6.	Votereum: An Ethereum-based E-voting system	1. External Personal Account (EOA) 2. Contract Account 3. Votereum

1.8: Existing System

Prior to 2004, India used a paper-based voting system. It's referred to as the ballot paper system. Voters were obliged to go to a polling station and cast their vote on ballot paper by putting a seal in front of the candidate's insignia. The votes were counted and the results were revealed. The winner was determined by who garnered the most votes. Ballot paper voting is unreliable, time-consuming, and difficult to count in a country with a population of more than 120 million people. There are further difficulties such as changing vote paper boxes with duplicates, ballot paper deterioration and marking stamp seals for more than one candidate, all of which must be addressed these issues. In order to solve these issues, electronic voting machines (EVMs) were made available. Electronic voting machines (EVMs) are made up of two basic parts:

1. ECI utilise the Control Unit to store and assemble votes.

2. Ballot Unit: A ballot unit is a device that is put in an election booth and utilised by voters.

A 5m wire connects the two units, with the other end permanently connected to a voting unit. The control unit houses a battery pack that powers the system. The voting unit comes with 16 candidate buttons, with the unused buttons covered by a plastic hide tab within the device. If

there are more than 16 candidates, an additional voting unit may be attached. The socket beneath the first vote unit can be used to connect a second ballot unit. EVMs are also known as DREs in the international community (Direct recording Electronic). Since the 2004 national election, when ballot papers were prohibited, electronic voting machines (EVMs) have been used in India elections. Votes are successfully recorded when EVMs are used, and there are no issues with counting, scalability, accuracy, quick results declaration, or system robustness. The main issue is that voting may not be done by a legitimate individual. Other issues such as political parties taking voting booths, underage voters casting ballots, and voter fraud are possible. The voter id card, issued by the Indian government, serves as evidence of identification. Name misprinting, missing names, no clear photo on photo id cards, and other issues can be seen on voter identification cards.

1:9: Proposed System

Several studies on the use of computer technology to enhance elections have been conducted. These studies discuss the dangers of using an electronic voting system due to software issues, insider threats, network vulnerabilities, and auditing issues. We've recommended redesigning the current online voting system to use Blockchain technology.

In comparison to the present system, the suggested system has the following advantages:

• Users may vote from anywhere in the globe until they become citizens of the nation, and the votes are saved on the Blockchain, making them tamper-proof.

• It will save a lot of time and lessen the burden because there will be no line for casting votes.

Our e-Voting system will meet four essential requirements, which are depicted below:

- **Authentication**: Only persons who have previously registered to vote are allowed to vote. A registration procedure will not be supported by our system. In order to comply with current requirements, registration often necessitates the verification of specific information and documents, which cannot be done properly online. As a result, before enabling voters to vote just once, the system should be able to validate their identities against a pre-confirmed database.
- **Confidentiality and privacy**: No links between voters identities and ballots should be allowed in the E-Voting system. During and after the election, voters must keep their identities hidden.
- **Correctness:** Votes must be counted and cannot be altered, replicated, or withdrawn.
- **Data integrity:** To ensure that all votes are counted accurately, the system should be verifiable. Our solution provides mobility, flexibility, and efficiency in addition to addressing the fundamental need. However, in this study, we will concentrate on the four most important requirements.

Initial Investigation

The primary goal of the preliminary study is to pinpoint the issue. The necessity for a new or improved system must first be identified. Only once the requirement has been identified can the proposed system be compared and further study performed. We had to identify the issue and potential at this point, and we researched the existing system and discovered that there were a few spots where we could combine additional technologies to make the system better than it was. It was determined that the suggested system could be developed with the resources available, and that it could prove to be a viable option.

Project Planning

The most important aspect of constructing a project is project planning. It outlines the phases, activities, and tasks required to complete a project. The project plan also includes the durations needed to complete the project, as well as the resources and milestones.

The project scope is initially specified, and the relevant procedures for project completion are selected. The time necessary to complete the various tasks is then indicated and organised into a work breakdown structure.

Several parts of a project, such as project plans, workloads, and team and individual management, are typically organised using project planning. The logical links between tasks are defined using an activity network diagram, which allows the important path to be discovered. Because project planning must be completed before the project can begin, it is inherently uncertain. As a result, the duration of activities is commonly measured as a weighted average of optimistic, normal, and pessimistic conditions. The critical chain technique incorporates "buffers" into project planning to account for unexpected delays in project execution. Project management software may be used to determine float or slack time in the schedule. After that, the needed resources may be calculated, and the expenditures for each activity can be assigned to each resource, resulting in the final project cost. In order to satisfy the project objectives, the project schedule may be optimised at this point to achieve the appropriate balance between resource use and project length. Once prepared and agreed upon, the project timetable is referred to as the baseline schedule. Progress will be tracked against the baseline schedule throughout the project's lifecycle. The technique of evaluating progress against the baseline timetable is known as earned value management.

A planning of project is a graphical representation of the approach the project's group intends to fulfill the project's objectives. Takes into account a variety of important aspects of the process, such as its scope, timeliness, and risks. The implementation plan may be thought of as a "contract" between project team members and reviewers. It lays out the measures that will be followed to accomplish the objectives, as well as the roles and duties that will be assumed. Other essential project management functions it supports include estimation performance monitoring and control, alternative analysis and decision-making, and performance forecasting and predicting.

Project plan must include the following elements:

- Scope statement
- Schedule
- Requirements
- Quality criteria
- Project resources
- Communications Plan

Scope Statement

It is a declaration of what work is and is not included in the project. A proper scope declaration minimises the risk of project overruns and unanticipated turbulence greatly. The scope description for this project is as follows: "This project is for the development of an online electoral system using Blockchain technology." There will be a website for the Election Commission as well as for voters. The user interface will be created as part of the project, and it will include all of the essential elements at both ends.

Schedule

By alerting all participants of the estimated arrival time, the project timetable keeps the project manager's hands on the wheel. Because projects are established as transitory endeavours with a stated start and finish date, the bulk of initiatives require exact placement of that end date. We'll go over the details of this project's timeline in a subsequent section titled Project Scheduling.

Requirements

All projects have criteria that are drafted at the start based on the demands of the customer. The requirements for this project include the module for generating elections and adding candidates to run in the elections. Requirement Specifications contains a detailed overview of the requirements.

Criteria for Quality

It is one of the most important aspects of project planning because if software is not thoroughly reviewed before being released to the market, it may cause a number of issues, putting strain on the maintenance team. The quality requirements, including pass/fail standards and the techniques used to guarantee the quality criteria are satisfied, should be identified in the project.

Materials for the Project

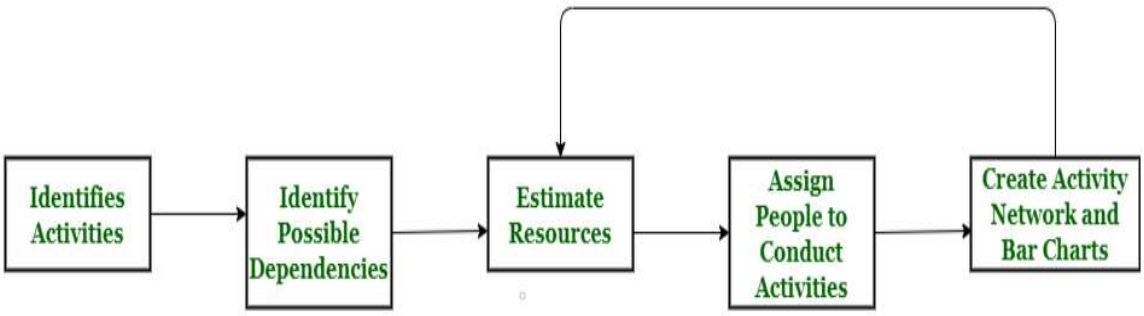
During the implementation of a project, assets generally require the greatest planning and organisation. This is due to the fact that they come late, require unanticipated maintenance, fail to meet requirements, or any of a variety of other factors that might cause a project to stall. Refers to the technical stack that will be used in the creation of the software. This is discussed in further depth later in this paper.

Scheduling Projects

In order for the timetable to take shape, we must follow several well laid-out stages. It is a means of giving information on when actions should be begun and how long they should take to finish. The following are some basic project scheduling principles:

• Defined duties - Each assignment is assigned to a certain team member.
• Specified outcomes - Every job that is scheduled for a software project should have a specified outcome, such as a work product.
• Establish milestones - Each job or group of tasks should be assigned a project milestone.
• When one or more work products have been examined and authorised by the team leader, a milestone has been reached.

PERT and GANTT Chart were developed to represent the project schedule and track the different tasks.



Chapter-2: Methodology

We'll use a permissioned blockchain in our proposal, which is a type of consortium-based chain that employs a consensus approach based on proof-of-authority. Transactions take place in proof-of-authority networks and blocks are validators have checked it, who are accounts that are authorized. Validators do not need to keep an eye on their computers all of the time because this method is automated. We might limit the ability of a small group of well-known entities to validate and certify blockchain transactions and filter transactions unilaterally using a permissioned blockchain that uses the POA consensus process, putting their identity and reputation at stake. Otherwise, proof-of-work consensus miners on a public blockchain would have to do it.

1) **Smart Contracts:** In a decentralized setting, smart contracts are trackable and irreversible programs (e.g., blockchain). After a smart contract has been started, no one can modify the code or the execution behaviour. Smart contract execution ensures that parties are bound by the terms of the agreement as written. This generates a new form of trust connection that isn't dependent on a single person. Because smart contracts are self-verifying and self-executing, they allow for improved management of the realization and administration of digital agreements.

   Nick Szabo, a graduate in CSE, invented the term "smart contracts." He stated objective behind smart contracts development for electronic trade protocols between strangers on the Internet using complex legal processes. Ethereum is a blockchain technology that can be used to construct smart contracts and is open-source. To construct these contracts, Ethereum developed a new programming language called Solidity. A permissioned blockchain underpins our electronic voting system, which is based on a smart contract.

2) **Ganache:** Ganache is use for setting up private blockchain that makes it simple to create Ethereum and Corda distributed apps. Ganache may be used at any step in the development process to safely and predictably create, deploy, and test your dApps. Ganache includes two types: user interface and command-line interface.

3) **MetaMask:** The software wallet for cryptocurrencies for digital transaction of the digital money in this MetaMask is used for transaction for the vote which has some amount of balance for a transaction to cast vote.

2.1: Product/System View

Design objectives are key properties of the system to be optimized that may get an influence upon that system performance overall design. The border between system design and requirements is thin. Design objectives are qualities that the designers aim to make "as excellent as possible" without precise criteria for acceptability, whereas requirements are particular values that must be reached in order for the product to be acceptable to the customer.

(i) Election Commission: In this module, the Election Commission will be in charge of setting up the smart contract, registering candidates and parties, and initiating an election.

(ii) Election Test: This is the module where we utilize Mocha Framework to conduct unit tests on our application to test our smart contract.

(iii) Voter Module: In this module, voters who have been given a personal ETH wallet will use the MetaMask extension to import into the voting portal and cast their vote.

2.2: System Components & Functionalities

1) Eligibility: This attribute specifies that only those who are eligible to vote are allowed to do so. Those who have received certification from the Election Commission.
2) Privacy: One of the most crucial features of democratic voting is privacy.
3) The privacy of voters should be protected. No one should be able to see how a certain voter voted or who that individual voted for.
4) Resistance to state intervention: No one should be able to force a voter to vote the way they were told to vote, and no one should be able to tell if the voter voted the way they were told to vote.
5) Verifiability: Everyone participating in the voting process should be able to check the outcomes. The election will be more transparent as a result of this. Additionally, an individual voter should be allowed to confirm their vote.
6) Irreversibility: A voter's vote should be irreversible. No one should be allowed to influence a voter's vote without the voter's consent. All records should be unchangeable.

2.3: Data & Relational Views

The project has been separated into multiple components, with separate modules for each feature. Any software is made up of various systems, each of which has several sub-systems, each of which has its own sub-systems. As a consequence, creating a comprehensive system in one go that covers all needed features is a time taking operation that, owing to its massive size, may result in several failures. If the partitioned modules are individually solvable, changeable, and compliant, an effective modular design can be created. The project modules are as follows:

(i) Election Commission: In this module, the Election Commission will be in charge of setting up the smart contract, registering candidates and parties, and launching an election.

(ii) Election Test: This is the module where we utilise Mocha Framework to conduct unit tests on our application to test our smart contract.

(iii) Voter Module: In this module, voters who have been given a personal ETH wallet will use the MetaMask extension to import into the voting portal and cast their vote.

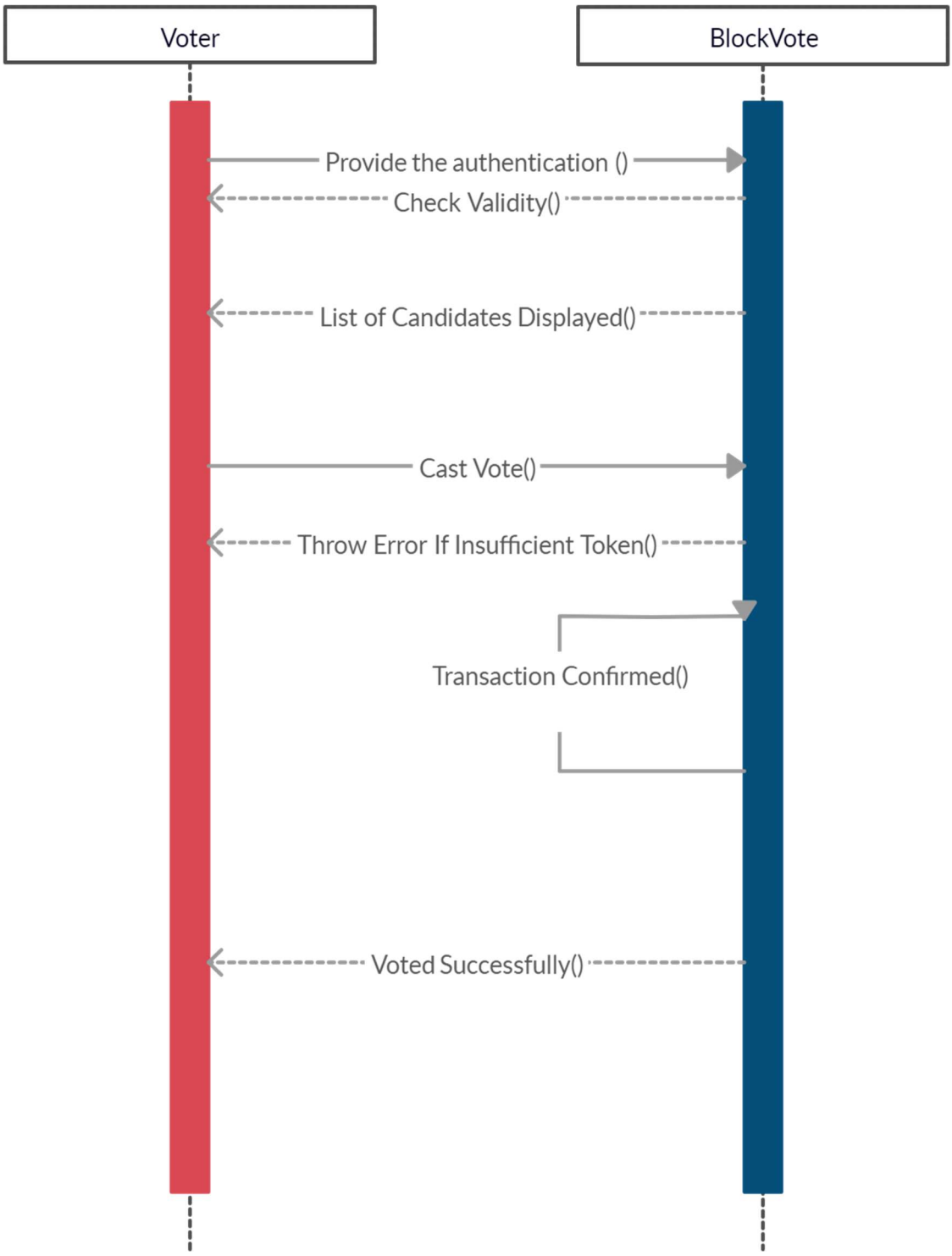


Figure 2: Sequence Diagram

2.4: Software Development Paradigm

This project employs agile approaches as well as an iterative approach. Let's go a little more into what's going on. Iterative and incremental development are the most often used terms to describe agile software development methodologies. Agile methodologies like SCRUM, DSDM, and FDD, to mention a few, are built on the iterative process. The fundamental idea is to break software development into cycles that reoccur (iterations). Each iteration is given a timebox, which is a set amount of time. In most cases, a timeframe is 2-4 weeks long. The ADCT (Analysis, Design, Code, Test) wheel is a more formal version of the PDCA cycle. For each iteration of the PDCA cycle, the team follows the approach outlined below:

• P (Plan) - Planning Iterations
During this discussion, the team works to discuss the following iterations goals. It also establishes the team backlog for the next iteration and describes the work performed.

• D (Design) – Execution of Iterations
This is the 'do' process, during which the programme is developed, designed, and coded. Functionality testing is carried out in the second or third iteration. The team collects user stories and gets ready for the Iteration Review, which comes next.

• C (Check) – Iteration Evaluation
The Product Owner participates in iteration review, sometimes known as the 'check' process. The group submits the tested deliverable to the Product Owner, who assesses the task and verifies that all of the requirements have been satisfied.

• A (Adjust) – Retrospect Iteration
The team assesses the whole iteration process, starting with the first step, in this event. It essentially builds on whatever advancements made in prior generations. New issues and their causes are discovered. The team's backlog is updated for future reference before the team begins the following phase. For refinements and modifications, iterations are repeated, and lessons acquired from prior cycles are applied to the current cycle. Until such time as a completely working software is available.

In comparison to previous approaches, agile methodologies offer the following benefits:

User Contribution -Users can participate in Agile Iterative Development. Customer input is received after each iteration cycle, and the product is subsequently subjected to appropriate revisions based on that information. Feature makes the project's structure more adaptable.

Software Evolution - Planning is a continuous effort in the Agile Iterative development process, allowing for the growth of ideas rather than full planning, which occurs right before the waterfall approach includes operation and evaluation.

Assessment of the Risk - Early development process using this model of development, you may identify and minimise risks, preventing barriers later.

Quick Delivery - Work is broken down into manageable cycles, which allows team members to concentrate only on the task at hand and complete it on time. Furthermore, every iteration includes testing in addition to coding and design, drastically reducing the project's duration.

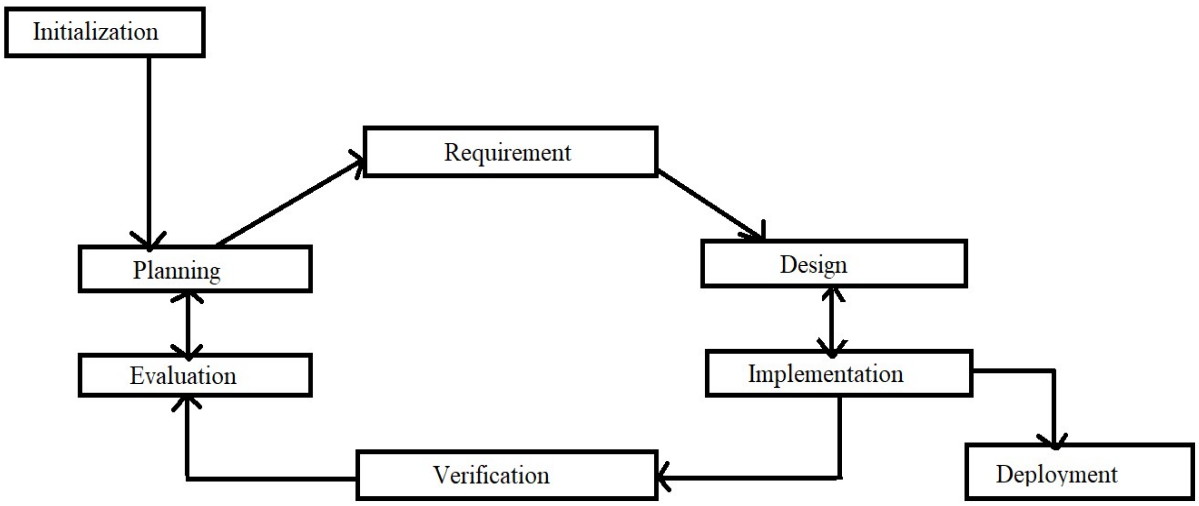


Figure 3: Iterative Model of Software Development

2.5: Process

- **Planning & Needs:** The initial part, like with other development projects, is to create a blueprint for the specification papers, describe hardware/software requirements, and prepare for the next stages of the cycle.
- **Analysis & Design:** After the planning phase, an analysis is conducted to establish the required business logic, database models, and other aspects for this stage of the project. The design step also takes place here, detailing any technological needs (languages, data layers, services, and so on) that will be employed to meet the requirements of the analysis stage.
- **Execution:** Once the planning and analysis are complete, the execution and coding phase may begin. All planning, specification, and design papers have been coded and included into the project's first iteration up to this point.
- **Testing:** Following the completion and implementation of this build iteration, the next step is to run a set of tests to identify and correct any potential flaws or issues.
- **Assessment:** Now that you've accomplished once you've completed all of the preceding processes, it's time to evaluate them. how far you've come. The above allows the entire

group, and also customers or even other key parties, to assess where the project stands, where it needs to go, and what changes may or should be made, among other things.

2.6: Advantages

• **Inherent Versioning:** Versioning will be used in most software development life cycles to indicate the product's newest version. The iterative model, on either hand, makes things a lot easier by assuring that every iteration improved on the preceding one progressively. Furthermore, if a new iteration fundamentally disrupts a system in a negative way, a previous iteration may be rapidly and simply installed or "rolled back" with minimal loss, which is especially useful for post-release maintenance and online application.

• **Quick Turnaround:** Although each phase of a iterative process could looks similar to that of an extra traditional system, such as the waterfall method, trying to imply that The iterative application's appeal is that each step may be efficiently trimmed down into smaller and smaller time spans, depending on the project or organization's demands. While the first iteration of all phases may take some time, each subsequent iteration will become faster and faster, reducing the time between each new iteration to days or even times.

• **Designed for Agile Teams:** Even though a stage process strategy like the waterfall model may work for large corporations with hundreds of people, the iterative model excels when employed by a smaller, more agile team. An entire "iteration process" may be efficiently completed by a number of individual team members with little to no outside input or assistance, especially when paired with the capabilities of contemporary version control systems.

• **Easy Adaptability:** An essential feature of the iterative process is its capacity to quickly adapt to changing project or client demands, which is based on the fundamental strength of providing frequent, frequent updates on a regular basis. Fundamental changes to the underlying code structure or implementations (such as a new database system or service implementation) may usually be implemented rapidly and inexpensively since any undesirable modifications can be found and rolled back to a previous iteration.

Chapter-3: Design Criteria

The design criteria for the project are to provide reliable and easy use for users. The registration process is the first step in our design, and establishing a voter's identity is crucial to the system's security. It's crucial to guarantee that someone's identity isn't used fraudulently, especially when it comes to voting because every vote counts. Our recommended solution employs recognition devices and authentic identity card numbers to cross-check if a person is in the database and eligible to vote, allowing people to register to vote. The voter is then given a unique hash address that he may use to vote. Ethers are provided to each hash, which he may use to vote once. On election day, the voter will walk to the polling station, go through a verification process, and then vote using the address provided to him. He will then be automatically logged out. The voters will also be informed of the current voting status. The system design is given in Figure 4. Each block is described in this Section.

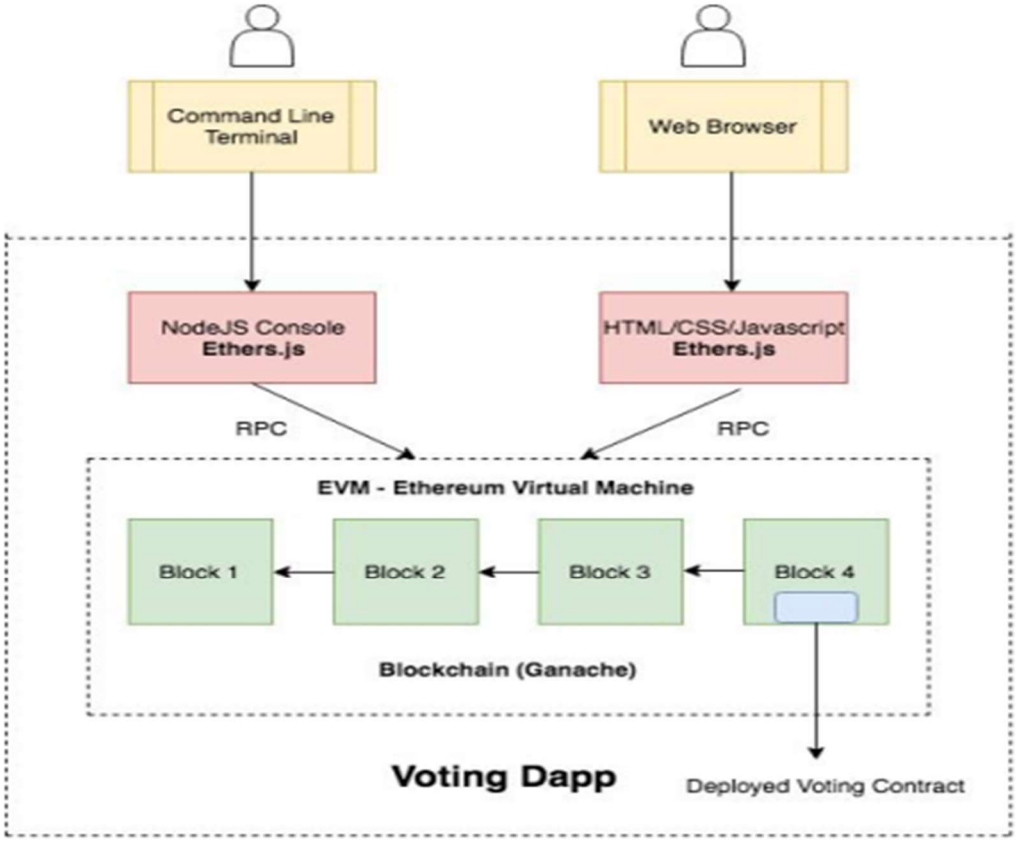


Figure 4: System Design

3.1: System Design

Listed below refer to distinct levels or layers in which activities take place:

Client: A client is an application that wishes to vote on the system. Clients interact with the system through a presentation layer.

Presentation Layer: This layer is in charge of client-side data display, i.e., it provides an interface for the end-user to vote in the programme.

Manager of Resources: It is in charge of organizing the data required to support the application logic (storage, indexing, and retrieval). The Local Blockchain Server, which is maintained by Ganache, is this resource manager.

The logic of the application: The application logic determines what the system performs in practice. It is concerned with the implementation of business regulations and the establishment of business processes.

The project has been separated into multiple components, with separate modules for each feature. Any programming is made up of various systems, each of which has several sub-systems, each of which has its own sub-systems. As a result, creating a comprehensive system in one go that contains all needed features is a time-consuming operation that, owing to its massive size, may result in several failures. System module has been divided into two types:

i) **Admin Module:** The admin will be able to add candidates and start the election and stop the election and declare the election.

ii) **Voter Module:** The vote will be able to cast vote for the selected candidate with their unique private key.

The System design consists of an admin login to start the election. The admin gives the unique private key to voters to vote, as well as the voter goes to cast tab to cast vote using their private key by selecting a candidate and then cast vote and the voter will be able to see the result ones it is declared by admin.

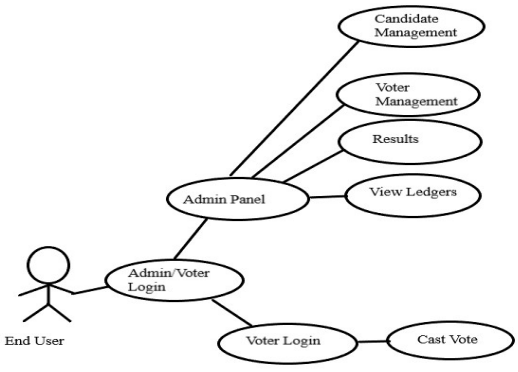


Figure 5: System Design Modules

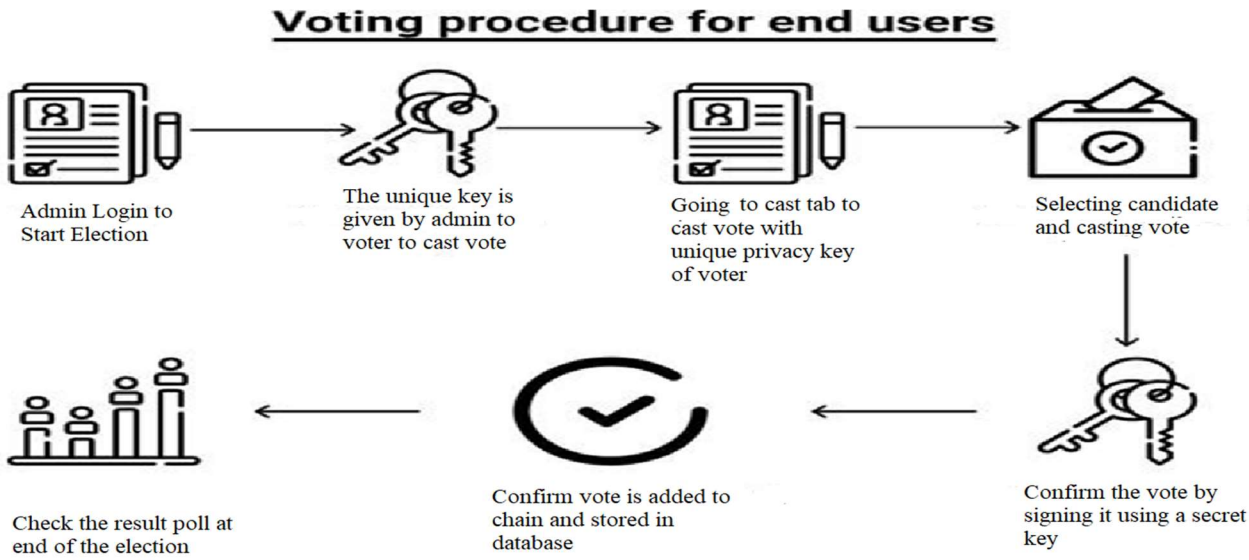


Figure 6: Voting Procedure

3.2: Design Diagrams

Entity Relationship Diagram

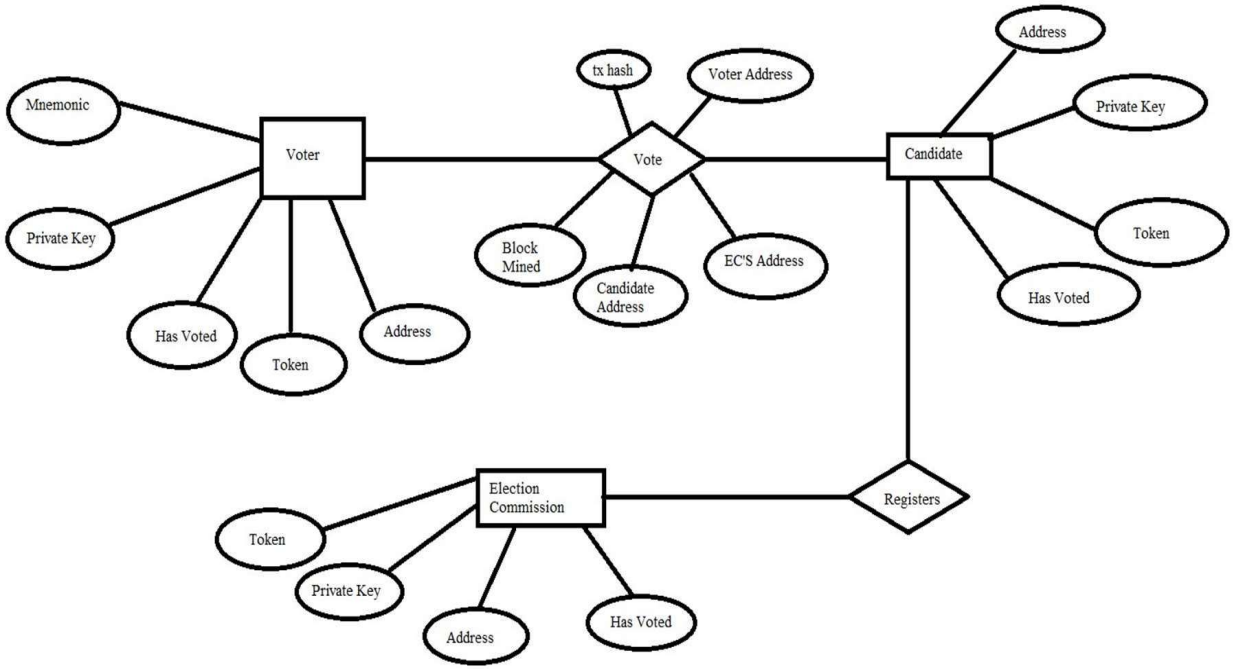


Figure 7: ER Diagram

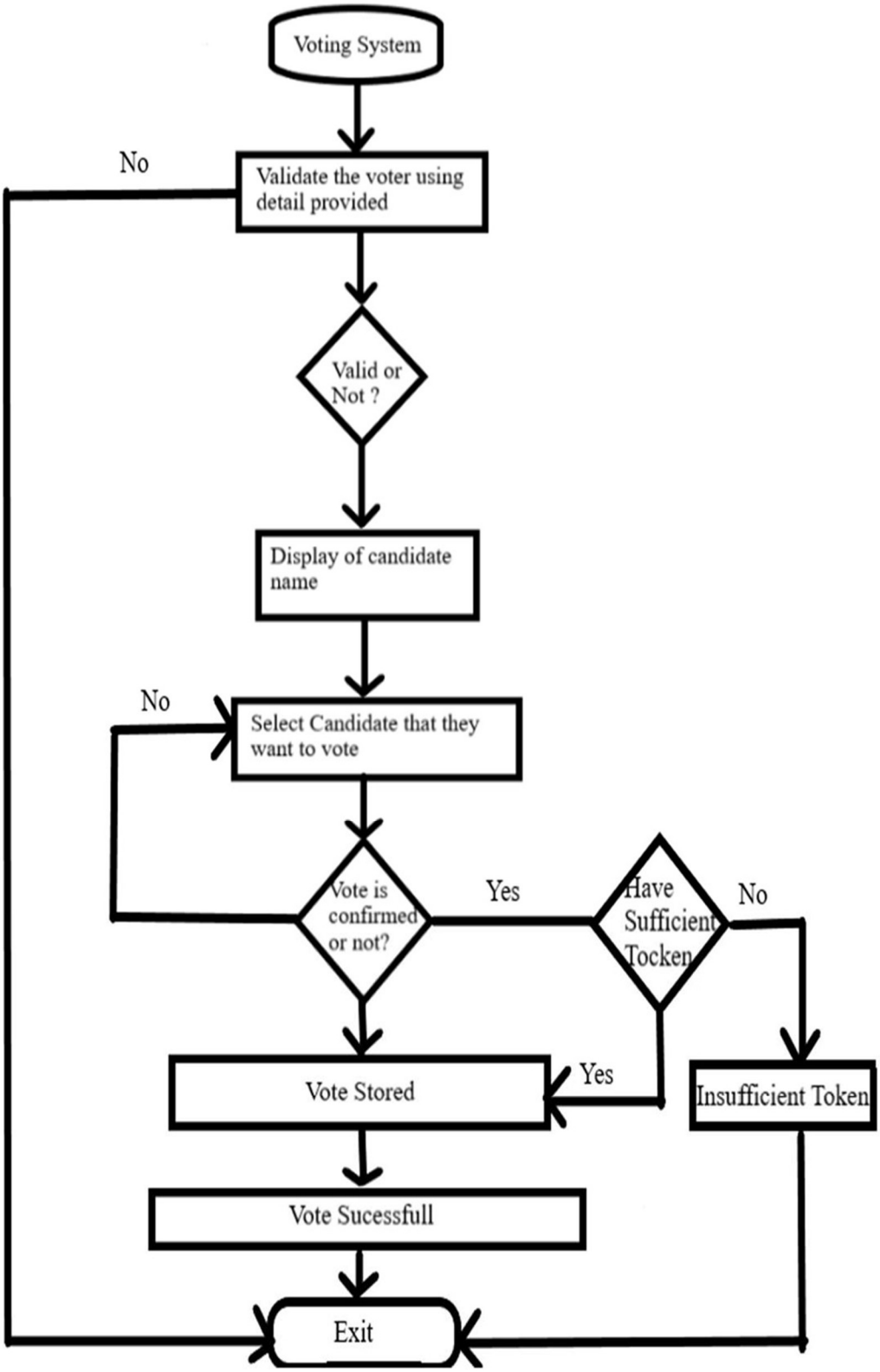


Figure 8: Flowchart

3.3: System Architecture

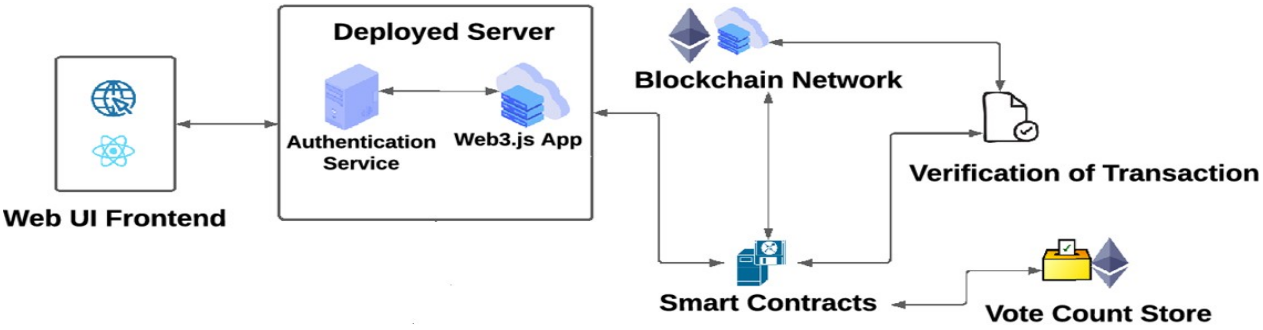


Figure 9: Implementation Diagram of System

3.4: Information & Communication Design

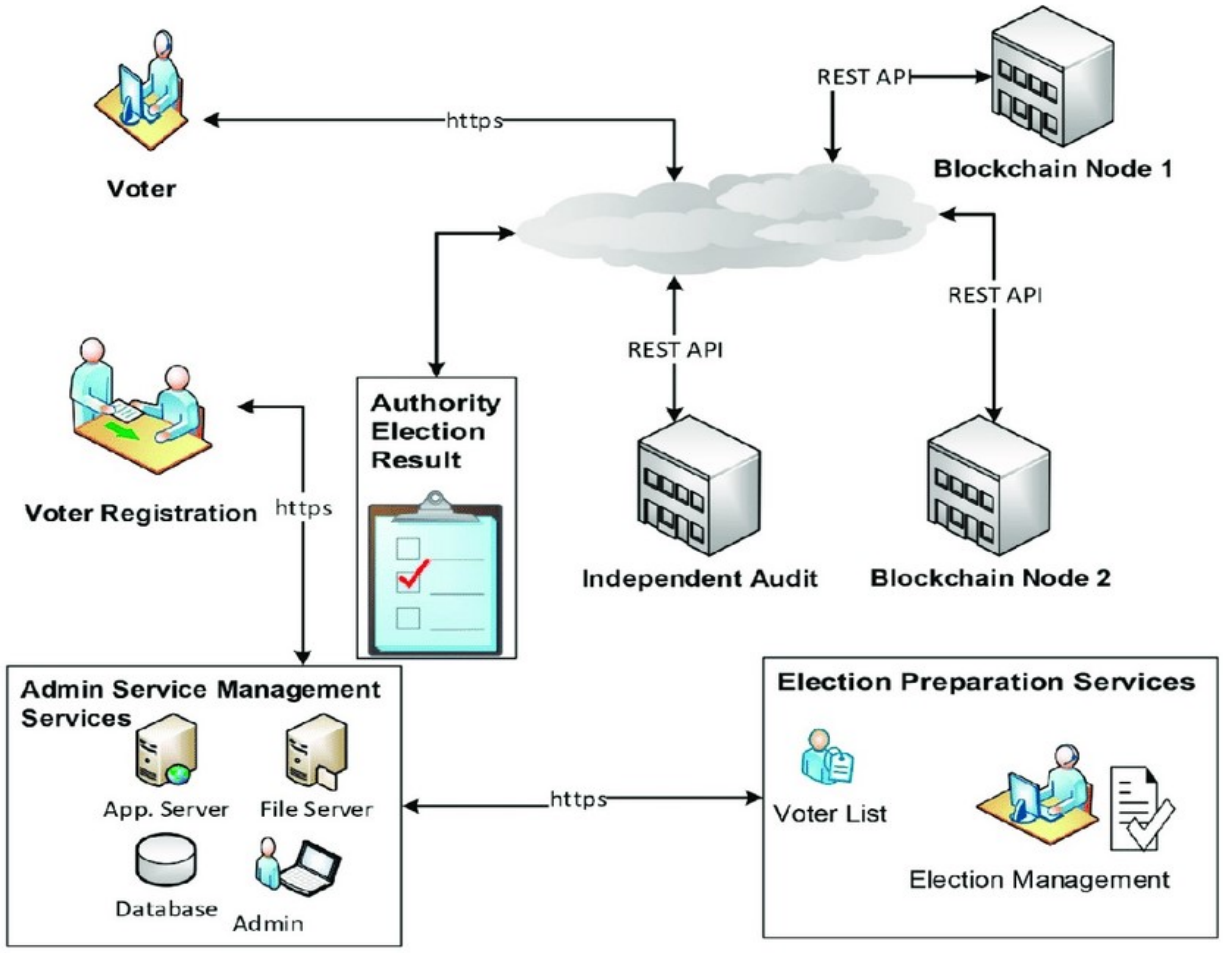


Figure 10: Blockchain Voting Systems Architectural Overview

Chapter-4: Development & Implementation

Development

Development has given below to different levels or layers where activities occurs:

Users: A user is any user or programme that wants to do anything with the system. Clients can interact with the system via a presentation layer.

Presentation Layer: The layer manages the client-side presentation of data, i.e., it provides a voting interface for the app's end-user.

Manager of Resources: The resource manager is in charge of the data organisation (storage, indexing, and retrieval) that is necessary to support the application logic. This resource manager is Ganache's Local Blockchain server.

Application Logic: It is in charge of determining how the system really works. It is in charge of implementing business rules and creating business procedures. The blockchain voting system is designed and implemented using a three-tier architecture.

Implementation

We have use MetaMask and Ganache to established smart contract through which transaction is occurs and I have installed various module such as truffle migrate which use to migrations JavaScript files that assist in the deployment of Ethereum contracts. These files are in charge of staging your deployment activities, and they're created with the expectation that your deployment requirements may vary over time. You'll write fresh migration scripts as your project advances on the blockchain and npm run dev module for deployment of server. After that admin start the election than voter able to cast vote to their selected candidate.

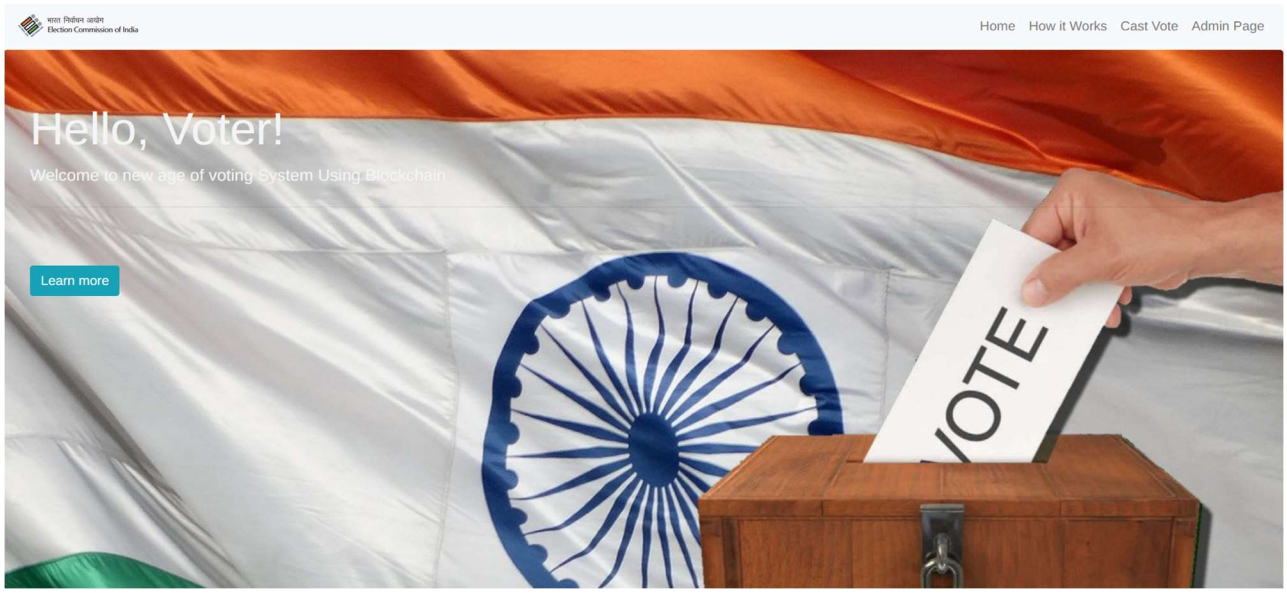


Figure 11: Homepage

Figure 12: How Does the Voting System Using Blockchain Work

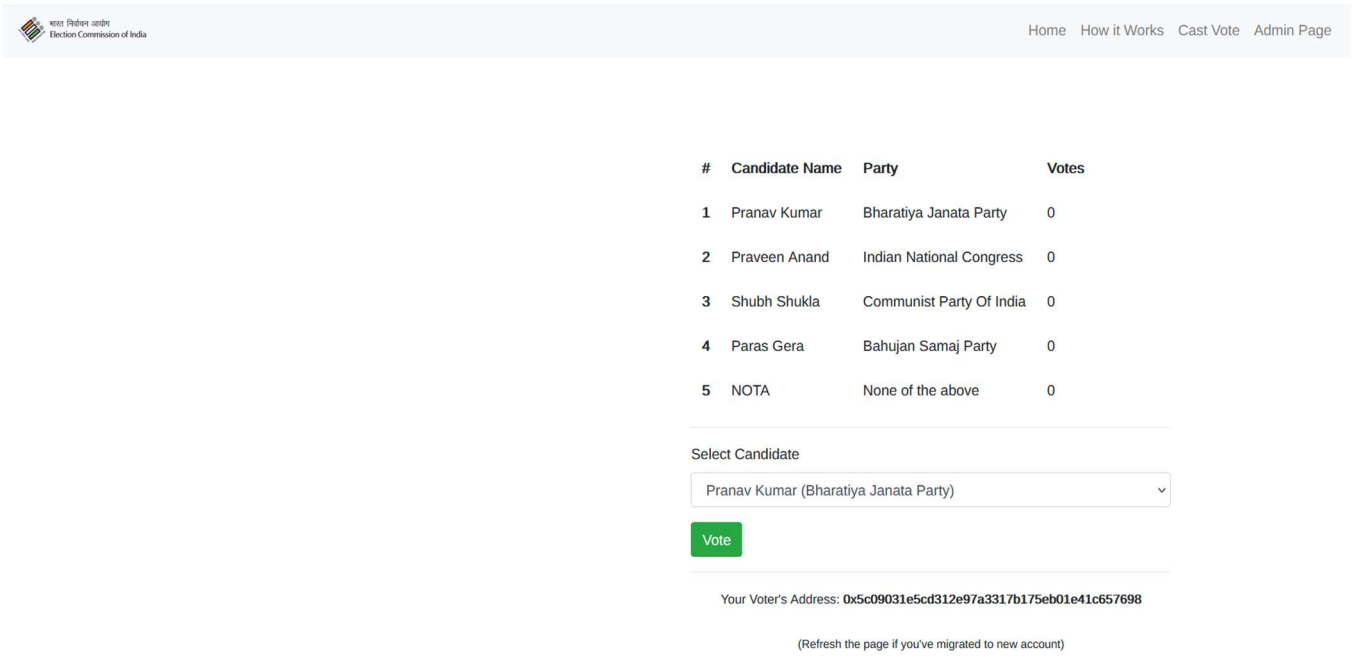


Figure 13: Cast Vote

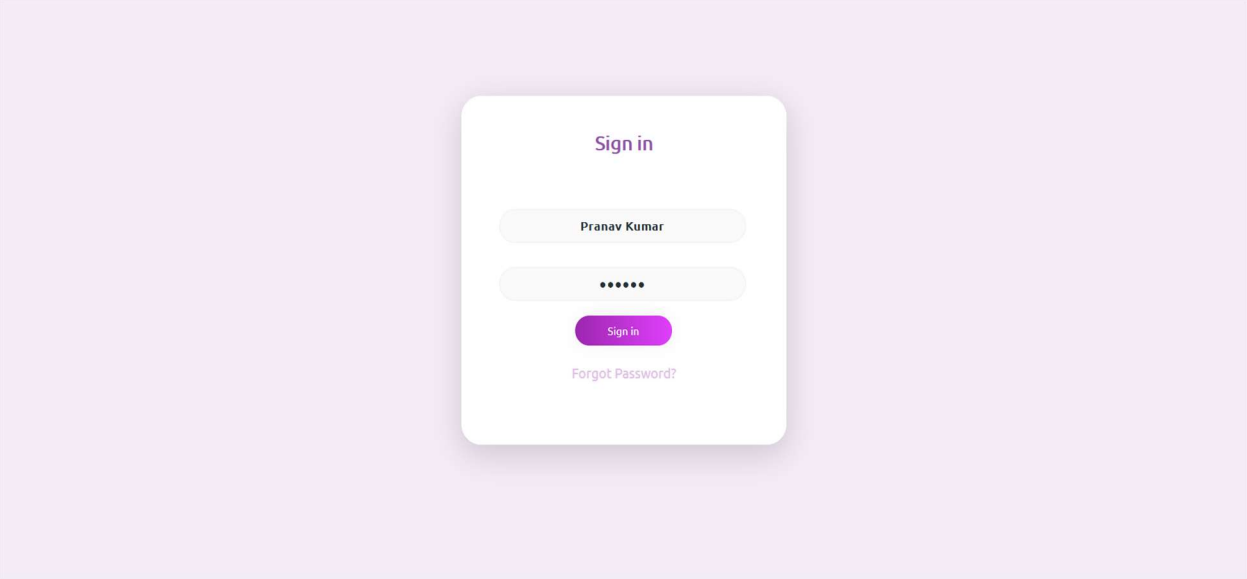


Figure 14: Admin Login

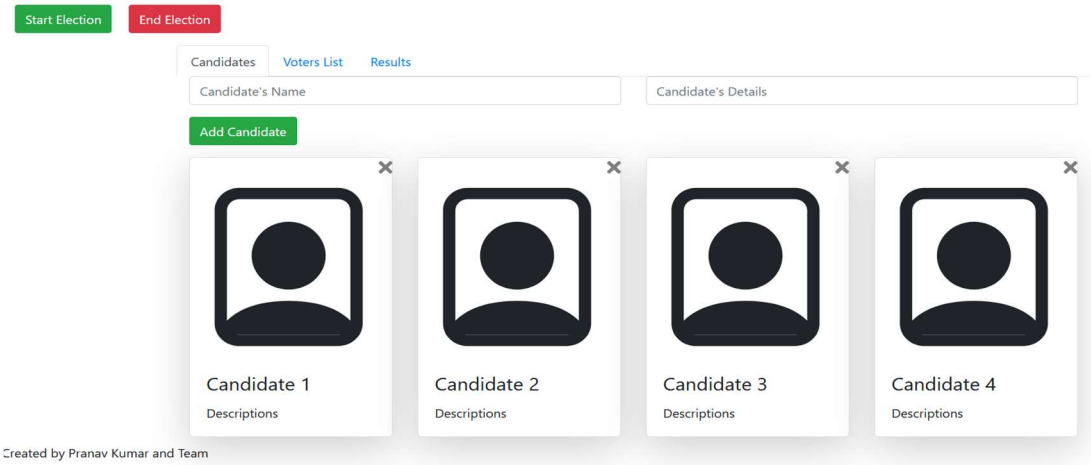


Figure 15: Admin Module

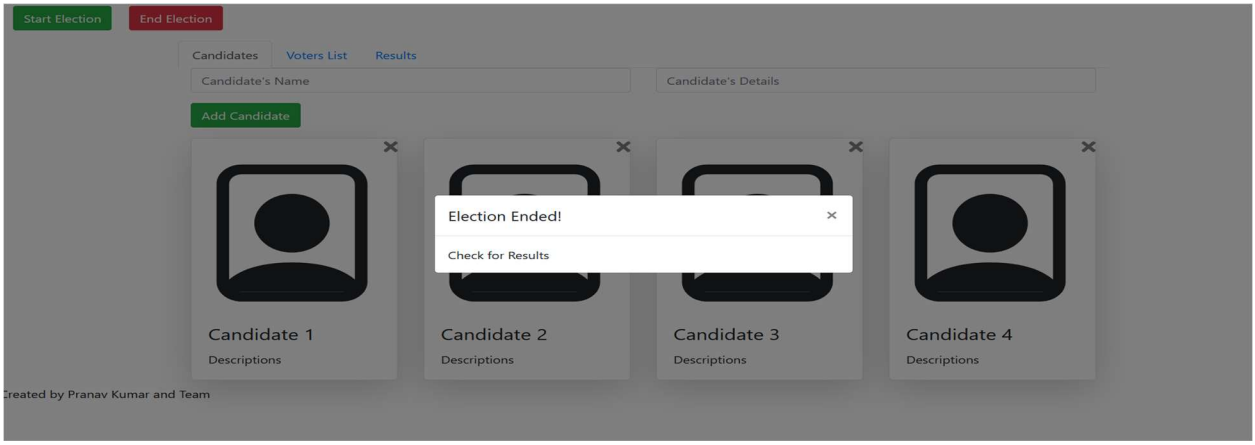


Figure 16: End Election

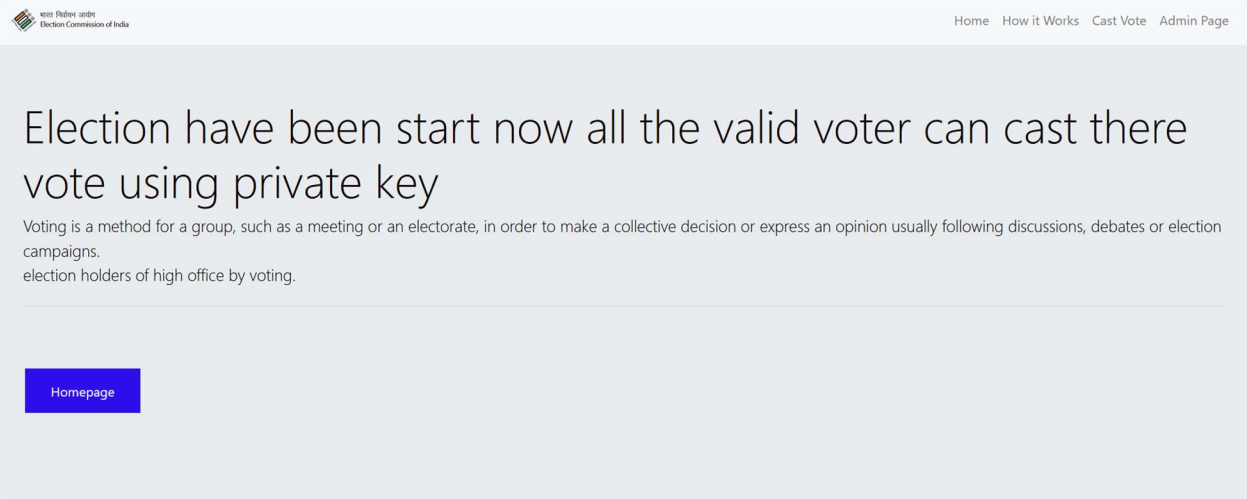


Figure 17: Start Election

Coding

The above system design is converted into code, which is a machine-readable format. It essentially converts a human-readable format to a machine-readable format. This is handled by the code generation stage.

When translating a system design into code, the following factors are taken into account:

• Have you double-checked your initializations?
• Have the data types been assigned correctly?
• Is a memory leak being addressed?
• Is it in accordance with the coding standard?

Coding Standardization

The efficiency of our code that has been transformed from the system design is referred to as coding standardisation. The effectiveness of the code is essentially determined by:

• **Readability:** The code should be legible, with suitable indentation and space, to make the contents of all modules evident.

• **Portability:** The code is portable enough to run on a variety of platforms as long as all of the required dependencies are present.

• **Easily debug:** As far as feasible, the coding should be error-free.

Source Code

```
pragma solidity >=0.5.16;

contract Migrations {
  address public owner;
  uint public last_completed_migration;

  modifier restricted() {
    if (msg.sender == owner) _;
  }

  constructor () public{
    owner = msg.sender;
  }

  function setCompleted(uint completed) public{
    last_completed_migration = completed;
  }

  function upgrade(address new_address) public {
    Migrations upgraded = Migrations(new_address);
    upgraded.setCompleted(last_completed_migration);
  }
}
```

Figure 18: Migration Smart Contract

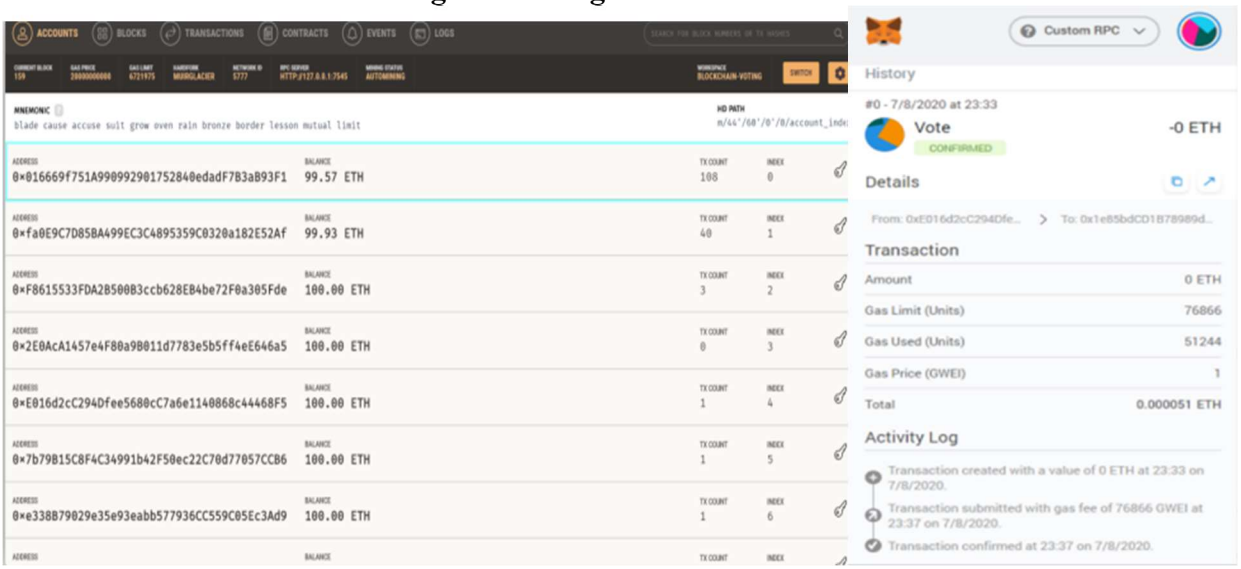


Figure 19: Smart Contract Establishment

```
pragma solidity >=0.5.16;

contract Election {
    struct Candidate {
        uint id;
        string name;
        string party;
        uint voteCount;
    }
    mapping(address => bool) public voters;
    mapping(uint => Candidate) public candidates;
    uint public candidatesCount;


    event votedEvent (
        uint indexed _candidateId
    );

    constructor () public {
        addCandidate("Pranav Kumar","Bharatiya Janata Party");
        addCandidate("Praveen Anand","Indian National Congress");
        addCandidate("Shubh Shukla","Communist Party Of India");
        addCandidate("Paras Gera","Bahujan Samaj Party");
        addCandidate("NOTA","None of the above");
    }
```

Figure 20: Election Smart Contract

```
var Election = artifacts.require("./Election.sol");

contract("Election", function(accounts) {
  var electionInstance;

  it("initializes with five candidates along with the parties", function() {
    return Election.deployed().then(function(instance) {
      return instance.candidatesCount();
    }).then(function(count) {
      assert.equal(count,6);
    });
  });

  it("it initializes the candidates with the correct values", function() {
    return Election.deployed().then(function(instance) {
      electionInstance = instance;
      return electionInstance.candidates(1);
    }).then(function(candidate) {
      assert.equal(candidate[0], 1, "contains the correct id");
      assert.equal(candidate[1], "Raju Bista", "contains the correct name");
      assert.equal(candidate[2], "Bharatiya Janata Party", "contains the correct party");
      assert.equal(candidate[3], 0, "contains the correct votes count");
      return electionInstance.candidates(2);
    }).then(function(candidate) {
      assert.equal(candidate[0], 2, "contains the correct id");
      assert.equal(candidate[1], "Sankar Malakar", "contains the correct name");
```

Figure 21: Election.js

```
init: function () {
  return App.initWeb3();
},

initWeb3: function () {
  // TODO: refactor conditional
  if (typeof web3 !== 'undefined') {
    // If a web3 instance is already provided by Meta Mask.
    App.web3Provider = web3.currentProvider;
    web3 = new Web3(web3.currentProvider);
  } else {
    // Specify default instance if no web3 instance provided
    App.web3Provider = new Web3.providers.HttpProvider('http://localhost:7545');
    ethereum.enable();
    web3 = new Web3(App.web3Provider);
  }
  return App.initContract();
},
```

Figure 22: Initializing the Web3 Connection on Front-End

System Security Measures

Data Security- It is very important to recognize the dangers associated with blockchain solutions. Security main component of voting system so that the data is not manipulated that's why we determine type of blockchain and the risks involved with a blockchain solution and look at the following many sorts of distributed ledgers, beginning with the least risky and working our way up to the most secure:

• Because public blockchains are open to the public, anybody may join and confirm transactions. Often more dangerous (for example, cryptocurrencies). It contains the following possibility of anyone being able to participate in the blockchain without any control or constraints.

• It is governed by just one organisation (control board) consortia; private blockchains are restricted and generally confined to corporate networks.

• Processors are not restricted in permissionless blockchains.

• Permissioned blockchains encrypt the ledger so that it can only be seen by relevant parties and decrypted by those who fulfil a need-to-know condition.

There are a variety of different dangers associated with blockchain solutions, which may be divided into three categories:

• **Governance And Business:** Business risks include financial ramifications, reputational problems, and regulatory risks, to name a few. Due to the decentralised nature of blockchain technology, decision criteria, governance guidelines, authentication, and access controls all need to be tightly controlled.

• **Procedure:** Certain risks are connected to the numerous procedures required for the creation and operation of a blockchain system.

• **Tech:** It is used to fulfil numerous operations and corporate needs, which may not always be the best option, resulting in security issues.

Blockchain Security Threat Models

The threat model should also be considered while evaluating a solution's security. By its very nature, blockchain ensures record integrity; yet, in other portions of a distributed ledger's application, a number of things might go wrong, resulting in data compromise and loss. Weak access restrictions, poor key and certificate management safeguards, and inadequate communication security are just a few examples. The key to successfully safeguarding such an application is to develop a comprehensive threat model for it and to plug any gaps that are found.

Blockchain applications typically include external components owned and maintained such as ownership and access control systems (IAM), multi-object authentication (MFA), primary public infrastructure (PKI), and control and research systems. Because they are built or managed by third parties, these systems must be thoroughly investigated before they can be applied to the overall solution. This should be taken into account when developing a threat model for the blockchain system.

Blockchain-Specific Security Measures

• API security best practises are employed to protect API-based transactions.

• Data categorization is used as a method to protect data/information.

• Based on company contracts, suitable endorsement policies are created and supported.

• A secret store is used for both application and privileged access.

4.1: Developmental Feasibility

In this part, we report the results of our poll on the possibility of incorporating blockchain technology into electronic voting systems. When we talk about feasibility, we're talking about a system that's expense, flexibility, safety, and easiness (or subsystem). While global assessments of these features are tough to come by, we've looked at a few thresholds or parameters to see if replacing existing (or future) structures with blockchain-based counterparts is a viable alternative. Over a three-year period with at least one election per year, any blockchain-based solution should be (noticeably) less expensive than traditional elections. It also shouldn't be more expensive than non-blockchain alternatives. Depending on the country, the size of the business, or the purpose, the system should be able to support millions of individuals.

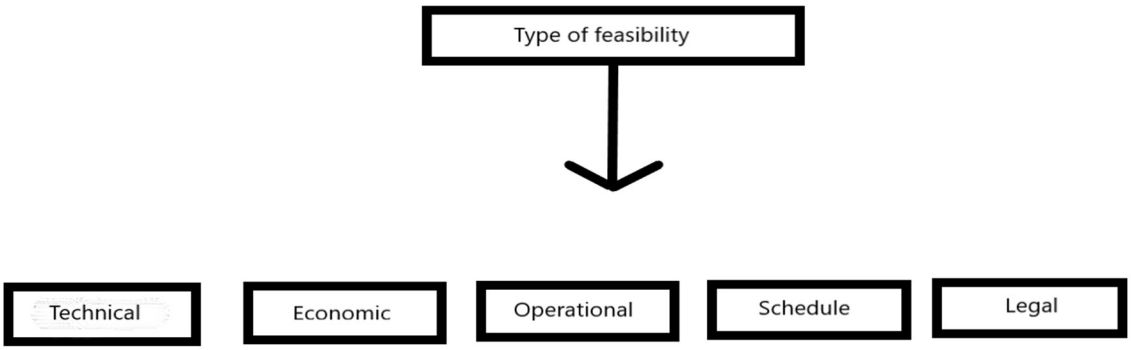


Figure 23: Type of Feasibility

Social Aspects

The societal implications of internet voting and Blockchain-based apps are far-reaching. The value derived from the provided simplicity of use, as well as people's sentiments of confidence in these so-called "high-tech" systems, are among these repercussions. Individuals, particularly those who live in remote areas and who are highly busy or mobile, have had wider, easier, and faster access to government services as a result of e-Government services. As a result, it might be viewed as a useful instrument for improving government-citizen relations. While eGovernment is not synonymous with democracy, the notion of e-voting broadens the scope of eGovernment to incorporate democratic instruments, culminating in e-Democracy.

Financial Aspects

Using automated electronic solutions such as online portals and mobile applications would undoubtedly cut administrative costs over time, despite the higher initial investment expenditures. A preliminary comparison of traditional versus electronic voting infrastructure and maintenance costs was recently released by prior study. Switching to an online voting system, according to the report, might save you money on an annual basis. When two or more elections take place in the same year, the disparity is magnified. Another research from Estonia looked at the price from the voters' perspective. Voters who live and over 30 minutes away from their polling places have higher (effort and cash) expenses and are therefore more willing to vote online, as per this study. Nevertheless, as they noted out, a version to computers among many of the elderly is often ignored. The majority of conventional elections costs (ballot papers, boxes, etc.) are made up of operational cost, followed by people and supply chain management. The price of e-voting systems includes software development, infrastructure components, and related maintenance. Applying distributed ledgers solutions could help you save even more money on your software because most blockchain packages are open-source projects with configurable APIs.

Security and Reliability

When compared to alternative database systems, the blockchain delivers superior security. The system's availability and fault tolerance are high because all nodes keep a duplicate of the data and to keep the system stable, they double-check one another. Transparency and anonymity are both provided by the blockchain. Privacy is not a goal, but it may be achieved.

Legal Feasibility

It assesses if the proposed system violates legal requirements; in this case, because we did not attempt to execute anything on the public domain, the project is lawful. It's critical that the project adheres to all of the prerequisites for getting started, such as certifications, copyrights, company insurance, tax identification number, health and safety precautions, and so on. In a legal feasibility study, there are various aspects to examine, such as ethical concerns and some societal issues. These are the concerns of privacy and accountability. Everything in this project is developed with all legal terms in mind, and no real-world data or privacy of any citizen of this nation has been violated in order to utilise it as a sample voter to construct this application.

4.2: Implementation Specifications

The mechanism, which is validated by EC and furnished with the ETH wallet address and private key, allows voters to vote from anywhere. The most important aspects of our blockchain voting method are security and anonymity. Our assumptions and dependencies for the system's correct operation are as follows:

• **MetaMask Browser Plugin:** Extension is used by users may maintain their accounts and keys in a number of methods, including hardware wallets, while being separate from the website.

• **Ganache:** It's a private blockchain that allows you to easily construct Ethereum and Corda distributed apps.

•**Truffle:** The EVM is used to create a global development environment, testing framework, and asset pipeline for blockchains, with the goal of making life easier for developers.

• **NodeJS:** It's a JavaScript runtime based on the V8 JavaScript engine in Chrome.

Structural features and software requirements of the Online Election System utilising Blockchain Technology are described in this document. Our country has used a manual voting method for many years. However, many people in our nation are unable to vote due to a variety of factors. For example, persons may not be in their own registration zone at any given time, preventing them from exercising their right to vote. To address these difficulties, an online election voting system is required. Keeping in mind that EVM votes may be tampered with, this online election voting system will be connected with Blockchain Technology to make it tamper-proof.

Functional Requirements

Table 3: Software Requirements

Software	Type	Version
Ganache	Ethereum Blockchain Server	2.4.0
MetaMask	Ethereum Wallet	7.7.9
Truffle	Development framework for ETH	5.1.31
Node	JavaScript Runtime	12.17.0
Visual Studio Code	Integrated development environment	1.46
Remix	Solidity's IDE	0.10.1
Windows 10	Operating System	1809

4.3: System Modules and Flow of Implementations

The system is intended to respond in a fair amount of time. The voter should be able to import his or her Election Commission-provided wallet in a matter of seconds, while keeping network stability in mind. The system's performance varies depending on the mode it is in:

i) **Election Mode:** The time it takes to deploy smart contracts in this phase is entirely dependent on the number of miners connected to the blockchain and the amount of GAS we decide to sign off the transaction as validated, but because we're working locally, it's only a matter of a half-minute or so.

ii) **Voting Mode:** During this phase, the system will reply in seconds since we won't have to sign off a transaction only to get the list of candidates for the elections, however the aforementioned performance may be delayed depending on network stability and web3 connection. Following that, depending on the miners and GAS limit, signing off the transaction may take a minute or two.

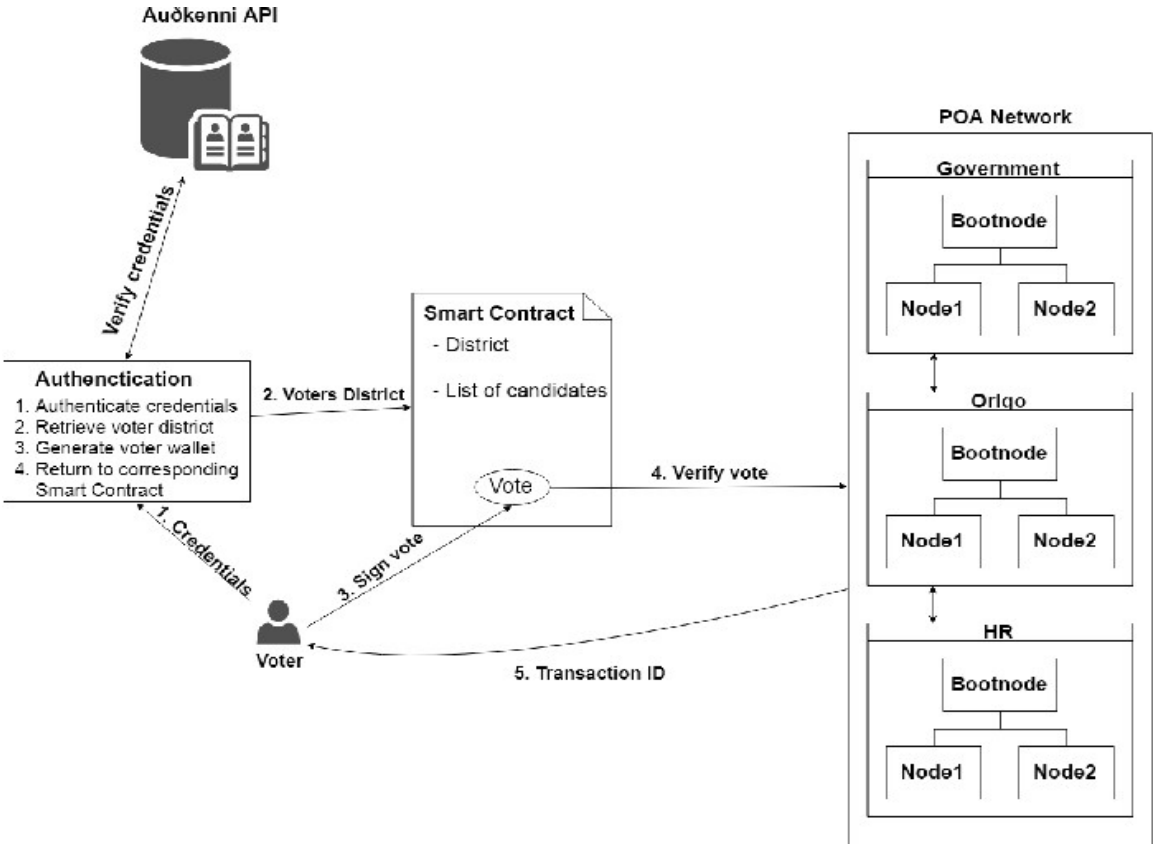


Figure 24: Flow of implementations

4.4: Critical modules of product/system

The critical modules of internet voting system using blockchain is voting module in which voter cast vote with their private key and every voter can cast vote only one time and this module cannot tamper and result cannot be manipulated.

Chatper-5: Results & Testing

5.1: Result

Our Internet-voting solutions include all of the components of a secure E-voting system, such as anonymity, authentication, accuracy, and verifiability. Only listed users should be able to vote, and the E-voting method should provide for anonymity during and after the election. Following that, the vote must be correct; no duplicate or redundant votes will be counted. This system's fundamental dependability and flexibility may be verified. When numerous people vote at the same time in this Blockchain-based E-voting system, an issue may arise that is tied to the preceding hash. As a result, we employed the Longest Chain Rule to solve this problem. However, in terms of openness and fairness, a blockchain-based e-voting system is more secure than a traditional voting method. For the academic community and real-world applications, we created an Internet voting method based on the blockchain (simulation). Begin with use a private key to cast their ballots and to produce a signature for use in the E-voting system. As a result, voters can submit their vote to Blockchain by sealing it. We utilised 500 transactions and 20 blocks in this experiment. Our simulation findings were based on 500 transactions and 20 blocks in this experiment. The E-voting system has 25 votes in each of the blocks. the outcomes of our simulation the E-voting system has 25 votes in each of the blocks.

5.1.1: Success Cases

In this project success result we have done testing in the framework of the Internet-voting systems. In which it has gone to several test case in term of privacy and security such generation of private key caste vote by user by using key given by user and it has passed all test cases as result and given 90% accuracy. The available system mechanisms for Internet-voting systems, as well as the laws that govern the preservation of results in general, are discussed in the project, which supports the development of a new technique based on cutting-edge technology.

5.1.2: Failure Cases

There were several failure cases in this while doing transaction using MetaMask but doing testing of various module of project we have resolved those issues and now we are having successful transaction of digital currency using MetaMask.

5.2: Testing

The testing has involved into two types are: -

- Understanding Blockchain Architecture: We examine and comprehend the business and functional requirements throughout this phase. This explains the application's behaviour as well as the user's interaction with it.

- Full Methodology of Evaluation Designing: The testing strategy for testing an application is described in this step. This should be done meticulously in order to ensure that all objectives are met.

Mocha is the testing framework used in this project to unit and integration test all of the applications test cases.

5.2.1: Type of testing adapted

i) **Unit Testing:** Unit testing is a term that refers to the process of this is the most basic and crucial stage of testing. Its requirement begins when a coder creates a unit of code. Every unit is put through its paces in a variety of circumstances. Detecting and correcting defects early in the software lifecycle helps to lower the cost of subsequent repairs. Finding and eliminating issues throughout the early phases of the application development process is far more cost effective. As a result, Unit Testing is the most critical of all testing tiers. Finding and fixing issues gets more and more expensive as the software project proceeds.

The following are the elements of unit testing:

- Making a test plan is the first step.
- Gather data and test cases.
- Write scripts to run the test cases if at all possible.
- Run the test cases when the code has been completed.
- If any flaws are discovered, they must be corrected and the code retested.
- Repeat the test cycle until the unit is bug-free.

ii) **Integration Testing:** The integration strategy dictates how distinct modules will be connected during integration testing. Individual modules can be linked one at a time or all at once. The Integration Strategy determines how the components will be connected. We used a bottom-up integration strategy to integrate test our application.

In Bottom-Up Integration, we work from the bottom up, which implies that the components below are written first and then integrated. The process of integration begins at the bottom and works its way up. If the calling component has not yet been constructed, a Driver, which is a specifically designed component, is used in its stead.

iii) **System Testing:** System testing is the last step in the verification procedure. We've evaluated if the full set of integrated components is performing optimally at this stage. The method is vital for the quality life cycle, and testers are tasked with determining if the system fulfils quality standards and meets all of the system's requirements. To guarantee neutrality, this technique is examined by testers who were not involved in the application development process. Furthermore, this method's atmosphere is quite similar to that of the manufacturing phase. System testing is crucial because it ensures that the application meets its functional, technical, and business requirements. System testing is critical since it ensures that the app fulfils all

of the user's operational and technical needs.

iv) **Acceptance Testing**- it is the last stage of the QA testing process. It aids in determining whether or not the programmed is suitable for consumer use. Typically, testers perform this stage with the assistance of customer service employees who analyses the application. As a result, they'll check to see if the software can perform all of the tasks stated. Requirements are frequently misconstrued during software development. As a result, Acceptance Testing is critical for identifying any misinterpretations of corporate objectives and providing the solution that your customers desire. You can go on to the application's production phase once you've completed this stage. It's probable that if you skip this stage, clients won't get what they want and won't return.

Testing Designs

```
it("initializes with five candidates along with the parties", function() {
  return Election.deployed().then(function(instance) {
    return instance.candidatesCount();
  }).then(function(count) {
    assert.equal(count,6);
  });
});
```

Figure 25: Candidate Count Unit Test

```
it("throws an exception for invalid candiates", function() {
  return Election.deployed().then(function(instance) {
    electionInstance = instance;
    return electionInstance.vote(99, { from: accounts[1] })
  }).then(assert.fail).catch(function(error) {
    assert(error.message.indexOf('revert') >= 0, "error message must contain revert");
    return electionInstance.candidates(1);
  }).then(function(candidate1) {
    var voteCount = candidate1[3];
    assert.equal(voteCount, 1, "candidate 1 did not receive any votes");
    return electionInstance.candidates(2);
  }).then(function(candidate2) {
    var voteCount = candidate2[3];
    assert.equal(voteCount, 0, "candidate 2 did not receive any votes");
  });
});
```

Figure 26: Double Voting Unit Test

```
it("throws an exception for double voting", function() {
  return Election.deployed().then(function(instance) {
    electionInstance = instance;
    candidateId = 2;
    electionInstance.vote(candidateId, { from: accounts[1] });
    return electionInstance.candidates(candidateId);
  }).then(function(candidate) {
    var voteCount = candidate[3];
    assert.equal(voteCount, 1, "accepts first vote");
    // Try to vote again
    return electionInstance.vote(candidateId, { from: accounts[1] });
  }).then(assert.fail).catch(function(error) {
    assert(error.message.indexOf('revert') >= 0, "error message must contain revert");
    return electionInstance.candidates(1);
  }).then(function(candidate1) {
    var voteCount = candidate1[3];
    assert.equal(voteCount, 1, "candidate 1 did not receive any votes");
    return electionInstance.candidates(2);
```

Figure 27: Invalid Candidate Unit Test

```
it("it initializes the candidates with the correct values", function() {
  return Election.deployed().then(function(instance) {
    electionInstance = instance;
    return electionInstance.candidates(1);
  }).then(function(candidate) {
    assert.equal(candidate[0], 1, "contains the correct id");
    assert.equal(candidate[1], "Raju Bista", "contains the correct name");
    assert.equal(candidate[2], "Bharatiya Janata Party", "contains the correct party");
    assert.equal(candidate[3], 0, "contains the correct votes count");
    return electionInstance.candidates(2);
  }).then(function(candidate) {
    assert.equal(candidate[0], 2, "contains the correct id");
    assert.equal(candidate[1], "Sankar Malakar", "contains the correct name");
    assert.equal(candidate[2], "Indian National Congress", "contains the correct party");
    assert.equal(candidate[3], 0, "contains the correct votes count");
    return electionInstance.candidates(3);
  }).then(function(candidate) {
    assert.equal(candidate[0], 3, "contains the correct id");
    assert.equal(candidate[1], "Saman Pathak", "contains the correct name");
    assert.equal(candidate[2], "Communist Party Of India (Marxist)", "contains the correct party");
    assert.equal(candidate[3], 0, "contains the correct votes count");
    return electionInstance.candidates(4);
```

Figure 28: Candidate Initialization Unit Test

Test Report

```
2_deploy_contracts.js
=====================

   Replacing 'Election'
   --------------------
   > transaction hash:    0xb83c5e582bae28a99d3e33f30ee90f81b6736eaf1900593622a51c382432b5ef
   > Blocks: 1            Seconds: 4
   > contract address:    0x93E4DCbE913e279932c98Ee119252C8Bdd0a16Af
   > block number:        3
   > block timestamp:     1648875052
   > account:             0x19730041487D52D745C0e9f18AEcC3eCDe29DeDd
   > balance:             99.98155932
   > gas used:            692441 (0xa90d9)
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.01384882 ETH
```

Figure 29: Test Report

5.2.2: Test Results of Various Stages

Result of various stages of testing are: -

1. **Unit Testing**- The first level of software testing is unit testing. We have to analyses individual components of the system at this point to verify if they are operating properly on their own. Individual functions or programmed processes, for example, might constitute these units if software is based on procedural programming. In object-oriented systems, however, these units can take the shape of a single class. These units might differ from one another depending on the tester frame and the problem. The selected components are then double-checked to ensure that they are fully functionality. This step must be fully completed, the examiner must be trained and knowledgeable about finer levels of data. These processes must be followed when making any modifications to the codebase. We have ensured that any concerns are rapidly fixed by using unit testing in code changes.

2. **Integration Testing**- We have been following step of testing, we will do integration testing. We've tested the system's separate components before putting it through its paces a whole. This allows software testers to evaluate the overall productivity of individual modules as well as identify any flaws in module and function interfaces. Regardless of how well a single component works, you won't know if the product is accomplishing its goal unless you conduct integration testing. Individual components can be assessed in a variety of ways as a group, although approaches differ depending on how each item is defined.

3. **System Testing-** System testing is the level of the verification procedure. We've assessed if the full set of connected parts are working at their full potential at this stage. For the quality life cycle, the method is critical, and testers are vital are tasked with determining if the system fulfils quality standards and meets all of the system's requirements. This approach is evaluated by testers who were not engaged in the application's development process to ensure objectivity. Furthermore, the environment for this procedure is quite similar to the production phase. System examine is essential because it ensures that the application meets its functional, technical, and business requirements. Device test is critical since it guarantees that the application meets all of the user's technical and functional requirements.

4. **Acceptance Testing-** The final level of the QA test cycle is testing. It assists in deciding if a software is appropriate for consumer use. This process is usually carried out by testers with the help of customer care representatives who examine the application. As a consequence, they'll examine the programmed to verify if it can complete all of the duties listed. During software development, requirements are commonly misinterpreted. As by this testing is critical for spotting any misinterpretations of business objectives and providing the solution that your consumers desire. You can go on to the application's production phase once you've completed this stage. If you neglect this step, it's likely that consumers will not receive what they desire and will not return.

5.2.3: Conclusion of Testing

We introduced testing conclusion to provide a Blockchain-based safe Internet-voting system in this project and reliable with error free by going through various testing methods such as functional testing, Integration Testing, Performance Testing and Node Testing which allows to provide error free distributed database to vote in a contemporary manner. We have demonstrated that blockchain technology can address concerns of security, transparency, fairness, and trust, as well as lowering the hurdles to E-voting systems. Nobody will be able to corrupt Blockchain technology since it will be freely verified and disseminated. We also discussed the disadvantages of our E-voting technology, which will be used in the future research.

5.3: Success of System

The client have been provided with error free and reliable by the product going through various testing phase and provide reliable system for voter and admin (Election Commission of India) which provide private key to vote which have unique id (key) for every voter to cast vote to selected candidates The time it takes to deploy smart contracts in this phase is entirely dependent on the number of miners connected to the blockchain and the amount of GAS we decide to sign off the transaction as validated, but because we're working locally, it's only a matter of a half-minute or so.

Chapter-6: Conclusion & Future Improvements

Conclusion

- These blockchains are controlled independently to eliminate the possibility of votes for certain parties being coordinated and returned to individual voters, while still allowing for the recording of who voted and how many times they voted.
- Every voter is a provider with a unique private key and voters will be able to vote once only.
- Our solution uses a blockchain to store voter registration information and verify that each voter is unique.
- Several additional concerns about openness, confidentiality, and integrity have also been examined as part of the suggested solution, to retain everything as a feature of the system.
- Only after casting a vote will the voter will be able to see the vote count.

Future Improvements

- System may be including with method for verifying with unique id which is unshareable.
- Using data from the government's voting system to pull the plan.
- Increasing system's security.
- Improving interface to make it more user-friendly for both voters and ECI (Election Commission of India).
- Local languages can be incorporated, which will benefit persons living in rural regions as well as the ignorant.
- A candidate's previous social work and qualifications might be included to provide a voter with a more informed decision.

6.1: Performance Estimation

The experimental setting for the proposed PSC-blockchain consensus model was created in MATLAB using laboratory, and the performance was evaluated using an experimental simulation on the Amazon EC2. Number of nodes are in simulated network for t2 range from 200 to 1000. As a single node, with vcpu = 2 and RAM (GiB) = 4. It compared the proposed PSC-scalability blockchain's throughput and latency against those of current techniques for reaching an agreement like as PoS and PoW. Built a simulation network with 200 nodes in each node set, then scaled it up to 1000 nodes four times. We analysed delay and assessed throughput for 10 blocks. Second, we put both the PoW and PoS protocols to the test on the same simulated network, which had anything from 200 to 1000 nodes (network size). We attempted ten blocks of PoW and ten blocks of PoS to calculate the latency. Figure 30 shows how the proposed consensus model compares to the existing consensus approaches (PoW, PoS) (PSC-blockchain). As demonstrated by the data bars, the PoW consensus technique had the largest latency of around 63s, while the PoS consensus method had the lowest latency of 10s. The suggested PSC-blockchain consensus mechanism has a latency of around 27 seconds. Furthermore, as seen in Figure 30, increasing the number of nodes did not enhance the rate at which PoW or PoS transactions were completed. When the number of nodes

increased, however, the proposed PSC-blockchain design was able to handle more transactions. The suggested PSC-throughput blockchain's surpassed 60Tps when the number of nodes was increased to 1000, which is faster than PoW and PoS, which had throughputs of 5TPS and roughly 25TPS, respectively. As a consequence, the findings reveal that the proposed PSC-blockchain is extremely scalable when combined with the approach.

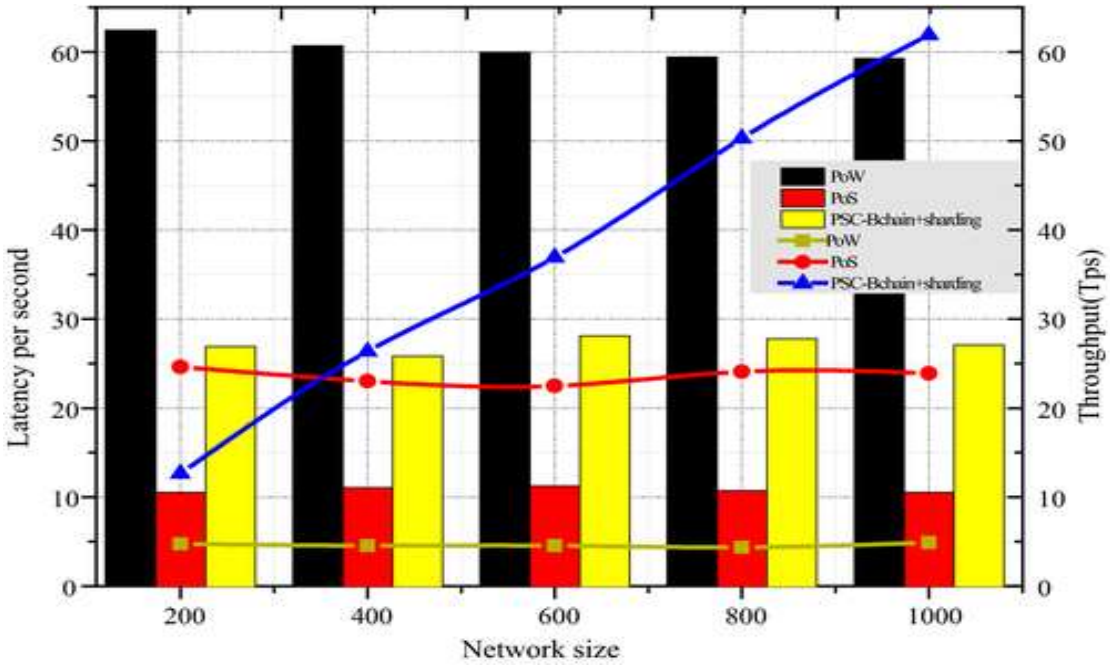


Figure 30: The Performance Evaluation

Cost Estimation of The Project

The term "cost estimation" refers to a rough estimate of a project's expenses. Because there are too many variables involved in calculating a cost estimate, such as human, technological, environmental, and political factors, cost estimating can never be a precise science. Furthermore, any process including a substantial human aspect can never be completely precise since humans are just too complicated to be completely predictable. Furthermore, due to the complexity of software systems, every reasonable-sized project will undoubtedly have a number of activities with complexities that are difficult to assess.

Typically, cost estimation is expressed in terms of effort. Person months or years are the most commonly used metrics (or man months or years). The effort is defined as the amount of time it takes for one person to work for a given length of time. Because no two development environments are the same, it is critical to consider the individual features of the development environment when comparing the effort of two or more projects. The length of time individuals work in various nations is a prominent illustration of disparities in development environments; the normal workweek in North America is 40 hours per week, whereas the typical workweek in Europe is 35 hours per week. As a result, when comparing a North American project to a European one, a conversion factor must be applied to both for an exact comparison. Cost estimation can be based on a variety of variables, making it difficult to compare projects if standard models or methods are not employed. A cost estimate, for example, might contain elements from management, development (e.g., training, quality assurance), and other areas unique to a company.

Project Planning and Cost Estimation

Cost estimating is an essential instrument that can have an impact on project planning and budgeting. Because a project's resources are limited, all of the features included in a requirements document may not be incorporated in the final result. A cost estimate completed at the start of a project will assist in determining which features can be incorporated within the project's resource limits (e.g., time). Prioritize requirements to guarantee that the most critical features are included in the final product. Because the complexity of a project rises with its size, there is greater possibility for mistakes as development advances, the risk of a project is decreased when the most crucial aspects are included at the start. As a result, cost estimation can have a significant influence on a project's life cycle and timeline. The allocation of resources can also be influenced by cost estimation. It is sensible for a corporation to devote more resources to costlier initiatives, such as more experienced people. The phrase "manpower loading" refers to the quantity of engineers and managers assigned to a project in a specific period of time. Most of the time, a price project failing is worse for a corporation than a less costly one failing. When estimating tools are employed, managers and developers can even try trading off certain resources or factors for others while keeping the project's cost constant. One compromise may be to invest in a more capable integrated development environment in order to lower the number of people working on a project.

The Estimator

Cost estimators may be directly or indirectly responsible for a project's implementation, such as a developer or management. An analogy-based technique, which is a frequent way of estimating for small companies and small projects, might be used by someone who knows the company and prior projects to compare the present project to previous initiatives. The past data is frequently confined to the estimator's recollection. In this scenario, the estimator would need to be knowledgeable and have worked for the firm for some time. Some argue that it is preferable to have the estimations done by outsiders since there is less risk of bias. It is true that those who work outside of an organisation will generally encounter fewer corporate politics than those who work within it. For example, a company's developer may wish to satisfy the management by providing an unduly optimistic estimate. The drawback of getting an outside estimate is that the individual will have less expertise of the development environment, especially if they are not from the organisation. An empirical approach of estimation, such as the Constructive Cost Model, would therefore be necessary (COCOMO). All sorts of estimators can employ empirical methods of estimation. There may be considerable scepticism about employing an empirical approach of estimating because it is unclear whether a model can surpass a human expert.

Aside from the cost drivers, various inputs and restrictions must be addressed throughout the actual cost assessment process. The financial constraint, which is the amount of money that can be budgeted or assigned to the project, is one of the most important limitations of the software cost estimate. There are also additional restraints, such as staff shortages and deadlines. Other inputs include architecture, which describes the components that make up the system as well as their interrelationships. Some businesses will have a specific software process or architecture in place, therefore software cost estimates for these businesses must be based on these factors. There are just a few instances where software needs remain constant. So, how do we handle changes in software requirements, ambiguities, or inconsistencies? A competent estimator will spot ambiguities and inconsistencies in the

requirements during the estimating process. The estimator will attempt to resolve all of these uncertainties as part of the estimating process by altering the requirements. If ambiguities or inconsistencies in requirements are not resolved, the estimation accuracy will suffer.

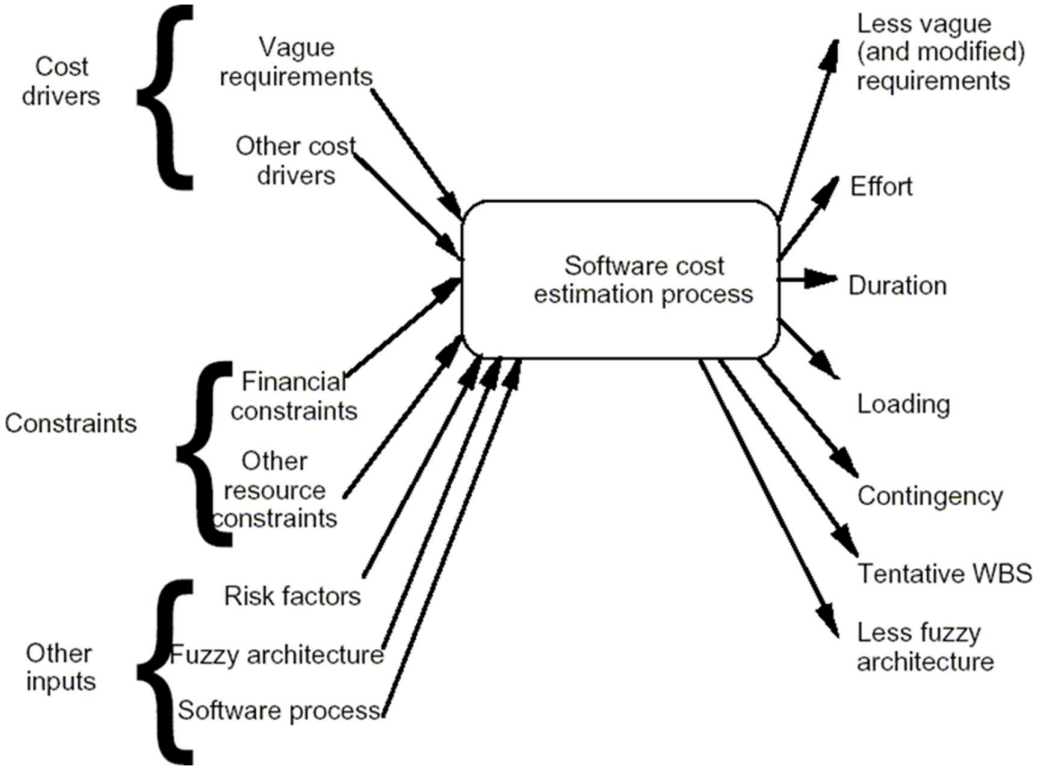


Figure 31: Actual Cost Estimation Process

6.2: Usability of Product/System

- The system's user interface will be simple and easy.
- A tutorial on how to use the system will be provided to those who are using it for the first time.
- Voters who have been given permission to vote should be permitted to do so (equal opportunity about accessibility and place).
- Authentication and Validation.
- The voter has the ability to check whether or not their vote was counted.
- The vote(transaction) should be immutable which disallow to tamper and involved the third-party.
- It must allow admin to verify the results.
- Should provide admin control.

6.3: Limitations

Despite the fact that blockchain technology is becoming increasingly popular, the majority of people are ignorant of it and have little understanding of it. As a result, it will take time for it to surface and be taken into account, which is a substantial technological limitation. Another flaw in the system is that a 51 percent assault may put our proposed design in trouble.

The voting system using blockchain may have security attack such as Phishing attacks, Routing attacks and Sybil attacks. Blockchains are difficult to grow because of their redundancy. Every network transaction has replicated on every system. As a result, hundreds of duplicates of the same data are created. It demands a lot of storage, and the bigger the chain network gets, the more computer power the nodes will require. Regulating your blockchain will be nearly impossible even if you meet all of your digital, software, and hardware criteria. While blockchain has a number of security issues, cyber security professionals can do a lot to address them. IT professionals with analytical and technical skills will be in the best position to deploy blockchain safely.

6.4: Scope of Improvement

The following are some methods to improve the system:

• Connecting the program to data from the government's voting system.
• Increasing the systems security.
• Improving interface to make it more user-friendly for both voters and ECI (Election Commission of India).
• Local languages can be incorporated, which will benefit persons living in rural regions as well as the ignorant.
• A candidate's previous social work and qualifications might be included to provide a voter with a more informed decision.
• Also, a recommendation mechanism for voters will be added, allowing the public to make ideas to the current winner.
• A complaint mechanism that allows individuals to make complaints about candidates can be incorporated.

References

[1] Liu, Y.; Wang, Q. An E-voting Protocol Based on Blockchain. IACR Cryptol. Print Arch. 2017, 2017, 1043. [Google Scholar].

[2] Shahzad, B.; Crowcroft, J. Trustworthy Electronic Voting Using Adjusted Blockchain Technology. IEEE Access 2019, 7, 24477–24488. [Google Scholar] [CrossRef].

[3] Racsko, P. Blockchain, and Democracy. Soc. Econ. 2019, 41, 353–369. [Google Scholar] [CrossRef]

[4] Yaga, D.; Mell, P.; Roby, N.; Scarfone, K. Blockchain technology overview. arXiv 2019, arXiv:1906.11078. [Google Scholar]

[5] The Economist. EIU Democracy Index. 2017. Available online: https://infographics.economist.com/2018/DemocracyIndex/ (accessed on 18 January 2022).

[6] Cullen, R.; Houghton, C. Democracy online: An assessment of New Zealand government websites. Gov. Inf. Q. 2000, 17, 243–267. [Google Scholar] [CrossRef]

[7] Schinckus, C. The good, the bad and the ugly: An overview of the sustainability of blockchain technology. Energy Res. Soc. Sci. 2020, 69, 101614. [Google Scholar] [CrossRef].

[8] Gao, S.; Zheng, D.; Guo, R.; Jing, C.; Hu, C. An Anti-Quantum E-Voting Protocol in Blockchain with Audit Function. IEEE Access 2019, 7, 115304–115316. [Google Scholar] [CrossRef].

[9] Kim, T.; Ochoa, J.; Faika, T.; Mantooth, A.; Di, J.; Li, Q.; Lee, Y. An overview of cyberphysical security of battery management systems and adoption of blockchain technology. IEEE J. Emerg. Sel. Top. Power Electron. 2020. [Google Scholar] [CrossRef].

[10] Hang, L.; Kim, D.-H. Design and implementation of an integrated IoT blockchain platform for sensing data integrity. Sensors 2019, 19, 2228. [Google Scholar] [CrossRef] [PubMed].

[11] Chang, V.; Baudier, P.; Zhang, H.; Xu, Q.; Zhang, J.; Arami, M. How Blockchain can impact financial services–The overview, challenges, and recommendations from expert interviewers. Technol. Forecast. Soc. Chang. 2020, 158, 120166. [Google Scholar] [CrossRef] [PubMed].

[12] Wang, B.; Sun, J.; He, Y.; Pang, D.; Lu, N. Large-scale election based on blockchain. Procedia Computer. Sci. 2018, 129, 234–237. [Google Scholar] [CrossRef].

[13] Ometov, A.; Bardinova, Y.; Afanasyeva, A.; Masek, P.; Zhdanov, K.; Vanurin, S.; Sayfullin, M.; Shubina, V.; Komarov, M.; Bezzateev, S. An Overview on Blockchain for Smartphones: State-of-the-Art, Consensus, Implementation, Challenges and Future Trends. IEEE Access 2020, 8, 103994–104015. [Google Scholar] [CrossRef].

[14] Hakak, S.; Khan, W.Z.; Gilkar, G.A.; Imran, M.; Guizani, N. Securing smart cities through blockchain technology: Architecture, requirements, and challenges. IEEE Netw. 2020, 34, 8–14. [Google Scholar] [CrossRef].

[15] C¸ abuk, U.C.; Adiguzel, E.; Karaarslan, E. A survey on feasibility and suitability of blockchain techniques for the e-voting systems. arXiv 2020, arXiv:2002.07175. [Google Scholar] [CrossRef].

List of Paper Publication

1. Implementation of blockchain for fair polling system in Information & Communication Technology (ISICT-2022), 01st & 02nd April 2022 in Sharda University, Greater Noida, Uttar Pradesh, India **(Published).**

2. Implementation of blockchain for fair polling system in 3rd International Conference on Applied Sciences, Engineering, Technology & Management (ICASETEM-2022) ISBN: 978-93-92105-42-5. **(Accepted).**

3. Implementation of blockchain for fair polling system in 2nd Series of IEEE ICONAT (The International Conference for Intelligent Technologies), Hubballi, Kartakata, India. Host Institution -K. L. E. Institute of Technology, Hubballi **(Accepted).**

4. Implementation of blockchain for fair polling system in International Conference on "Contemporary Innovations in Mechanical Engineering" (CIME-2022)**. (Accepted).**

5. Review of Blockchain for Internet Voting Systems and Open Research Challenges in International Conference on Advancements in Interdisciplinary Research (AIR-2022), 6th- 7th May 2022 Allahabad, Prayagraj (U.P.), India **(Accepted).**

6. Review of Blockchain for Internet Voting Systems and Open Research Challenges in IJISRT (International Journal of Innovative Science and Research Technology), ISSN No.: 2456-2165 **(Accepted).**

7. Review of Blockchain for Internet Voting Systems and Open Research Challenges in 2nd International conference on Advancement in Electronics & Communication Engineering (AECE-2022) **(Accepted).**

GitHub Repository Link:-
Link-https://github.com/PranavKumar8769/Internet-Voting-System-Using-Blockchain-Technology