

CP-III Project Report on

# **E-voting Using Blockchain at U. V. Patel College of Engineering**



**Internal Guide:**  
Prof. Pooja Thakkar

**Prepared By:**  
Mr. Ujjawal Patel (19012011139)  
Mr. Kavya Patel (19012011116)

**B.Tech Semester VII  
Computer Engineering  
Nov-Dec, 2022**

Submitted to,  
Department of Computer Engineering  
U.V. Patel College of Engineering  
Ganpat University, Kherva - 384 012

# U.V. PATEL COLLEGE OF ENGINEERING



26/11/22

## CERTIFICATE

**TO WHOM SO EVER IT MAY CONCERN**

This is to certify that **Mr. Ujjawal Patel** student of **B.Tech. Semester VII (Computer Engineering)** has completed his full semester on site project work titled “**E-voting using blockchain**” satisfactorily in partial fulfillment of the requirement of Bachelor of Technology degree of Computer Engineering of Ganpat University, Kherva, Mehsana in the year 2022-2023.

**Prof. Pooja Thakkar**  
**College Project Guide**

**Dr. Paresh M. Solanki**  
**Head, Computer Engineering**

# U.V. PATEL COLLEGE OF ENGINEERING



30/11/22

## C E R T I F I C A T E

**TO WHOM SO EVER IT MAY CONCERN**

This is to certify that **Mr. Kavya Patel** student of **B.Tech. Semester VII (Computer Engineering)** has completed his full semester on site project work titled “**E-voting using blockchain**” satisfactorily in partial fulfillment of the requirement of Bachelor of Technology degree of Information Technology of Ganpat University, Kherva, Mehsana in the year 2022-2023.

**Prof. Pooja Thakkar**  
**College Project Guide**

**Dr. Paresh M. Solanki**  
**Head, Computer Engineering**

## ACKNOWLEDGEMENTS

This satisfaction that successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation it made it possible, whose constant guidance and encouragement crown all efforts with success. I am grateful to our guide **Prof. Pooja Thakkar** for the guidance, inspiration, and constructive suggestions that helpful us in the preparation of this project. I also thank our colleagues who have helped in successful completion of the project. I would like to express my gratitude towards my parents & member of U V Patel Collage of Engineering for their kind co-operation and encouragement which help me in completion of this project. My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

## ABSTRACT

Electronic voting or e-voting has been used in varying forms since the 1970s with fundamental benefits over many techniques-based systems such as increased efficiency and reduced errors. However, there remain challenges to achieve widespread adoption of such systems especially with respect to improving their resilience against potential faults. Blockchain is a disruptive technology of the current era and promises to improve the overall resilience of e-voting systems.

Voting is a fundamental part of democratic systems; it gives individuals in a community the faculty to voice their opinion. In recent years, voter turnout has diminished while concerns regarding integrity, security, and accessibility of current voting systems have escalated. E- voting was introduced to address those concerns; however, it is not cost-effective and still requires full supervision by a central authority. The blockchain is an emerging, decentralized, and distributed technology that promises to enhance different aspects of many industries.

Expanding e-voting into blockchain technology could be the solution to alleviate the present concerns in e-voting. In this paper, we propose a blockchain-based voting system, named e- Vote, that preserves voter privacy and increases accessibility, while keeping the voting system transparent, secure, and cost-effective. e-Vote implements a university-scaled voting framework that utilizes Ethereum's blockchain and smart contracts to achieve voter administration and auditable voting records. In addition, e-Vote utilizes a few cryptographic techniques, including homomorphic encryption, to promote voter privacy. Our implementation was deployed on Ethereum's Testnet to demonstrate usability, scalability, and efficiency.

The blockchain-based voting project has two modules to make the whole project integrated and work along. One will be the Election Commission who will be responsible for creating elections, adding registered parties and candidates contesting for the election added under the smart contracts. The other end will be the voter's module where everyone can cast a vote for their respective Assembly Constituency and the vote will be registered on the blockchain to make it tamper proof.

# INDEX

1. INTRODUCTION.....	1
1.1 Introduction.....	1
1.2 Aim and objective.....	2
1.3 Existing System.....	3
1.4 Problem Statement.....	3
1.5 Proposed System.....	3
2. LITREATURE SURVEY.....	5
2.1 Security analysis of India's electronic voting machines.....	5
2.2 A conceptual secure blockchain- based electronic voting system.....	5
2.3 Blockchain Based E-Voting Recording System Design.....	6
2.4 Platform-independent Secure Blockchain-Based Voting System.....	6
2.5 System Analysis.....	7
3. SYSTEM SPECIFICATION.....	9
3.1 Functional Requirements.....	9
3.2 Non-Functional Requirements.....	10
3.2.1 Reliability.....	10
3.2.2 Security.....	10
3.2.3 Maintainability.....	10
3.2.4 Performance.....	10
3.2.5 Portability.....	10
3.2.6 Scalability.....	10
3.2.7 Flexibility.....	10
3.3 Hardware Requirements.....	11
3.4 Software Requirements.....	11
3.5 Project Planning.....	11
4. SYSTEM DESIGN AND ARCHITECTURE.....	14
4.1 Software Design.....	14
4.2 System Design.....	18
4.2.1 Use case: Users.....	18
4.2.2 Use case: Admin.....	19
4.2.3 Use case: Add New Users.....	19
4.2.4 Use Case: System use case.....	20
4.3 Class Diagram.....	20
4.4 Data Flow Diagram.....	21
4.5 Data Strategy.....	23
4.5.1 Admin.....	23
4.5.2 Candidate.....	23
4.5.3 Voters.....	24
4.5.4 Position.....	25
4.5.5 Vote.....	25
5. IMPLEMENTATION.....	26
5.1 Flowchart.....	26
5.2 Modules.....	27
5.3 Overall Implementation Process.....	28

6. TERMONOLOGIES.....	32
6.1 Languages and Technology.....	32
6.1.1 PHP.....	32
6.1.2 WAMPP.....	32
6.1.3 phpMyAdmin.....	32
6.1.4 HTML.....	33
6.1.5 Cascading Style Sheets.....	33
6.1.6 JAVASCRIPT.....	33
6.1.7 BOOTSTRAP.....	34
6.2 Smart Contract Concept.....	34
6.3 Methodology.....	36
7. TESTING.....	39
7.1 Testing.....	39
7.2 Cost Estimation and Project Planning.....	39
8. PROTOTYPE.....	42
8.1 Home Screen.....	42
8.2 Signin and Signup.....	45
8.3 Registration form of voting.....	45
8.4 Voter's Dashboard.....	47
8.5 Connect Wallet.....	48
8.6 Preview ballot.....	48
8.7 Transaction for confirm vote's.....	49
8.8 Successfully Vote Submitted.....	50
8.9 Admin Dashboard.....	50
8.10 Voter's Verify Dashboard.....	51
8.11 Candidates List.....	51
8.12 Ballot Position.....	52
9. CONCLUSION.....	53
9.1 Conclusion.....	53
9.2 Further Enhancements.....	53
10. BIBLIOGRAPHY AND REFERENCES.....	54
10.1 Book Used.....	54
10.2 Reference Used.....	54

## LIST OF FIGURES

Figure 4.1 Iterative Model of SDLC.....	14
Figure 4.2 Sequence Diagram.....	17
Figure 4.3 User Login Page.....	18
Figure 4.4 User Login Page.....	19
Figure 4.5 Add New Users.....	19
Figure 4.6 System use case.....	20
Figure 4.7 Class Diagram.....	20
Figure 4.8 0- level DF.....	21
Figure 4.9 1- level DF.....	21
Figure 4.10 2- level DF.....	22
Figure 5.1 Flowchart for Voting.....	26
Figure 5.2 Private Blockchain Concept.....	28
Figure 6.1 Transfer contract terms into code.....	35
Figure 6.2 The code is stored in a blockchain and replicated between participants.....	35
Figure 6.3 When a term is satisfied, computers in the network verify its correctness.....	36
Figure 6.4 Architecture of Purposed system.....	37
Figure 7.1 Actual Cost Estimation Process.....	40
Figure 8.1 Home Screen.....	44
Figure 8.2 Signin and Signup.....	45
Figure 8.3 Registration form of voting.....	46
Figure 8.4 Voter's Dashboard.....	47
Figure 8.5 Connect Wallet.....	48
Figure 8.6 Preview ballot.....	48
Figure 8.7 Transaction for confirm vote's.....	49
Figure 8.8 Successfully Vote Submitted.....	50
Figure 8.9 Admin Dashboard.....	50
Figure 8.10 Voter's Verify Dashboard.....	51
Figure 8.11 Candidates List.....	51
Figure 8.12 Ballot Position.....	52



**LIST OF TABLES**

Tables 4.1 Admin.....23

Tables 4.2 Candidate.....23

Tables 4.3 Voters.....24

Tables 4.4 Position.....25

Tables 4.5 Vote.....25

# Chapter 1

## INTRODUCTION

### 1.1 Introduction

Internet is the greatest thing invented humanity. But there are some flaws on the. But there are some flaws on the internet. Consider a situation where you are depositing money or casting a vote, there is a single point of authority, and we are supposed to believe him/her with our data/money/vote. The limitation of the present system is a single point of control/failure. The Authority may or may not Authority be telling the truth or corrupted. The solution to this is to employ a decentralized and distributed system where the consensus of the users/peers is used to evaluate the transactions. /votes/data.

#### What is Blockchain ?

A blockchain is a collection of blocks linked together with chains using cryptography. Blockchain is one of the emerging technologies with strong cryptographic foundations enabling applications to leverage these abilities to achieve resilient security solutions. Here the data is divided into blocks and linked together. Each block is associated with a hash value (which represents the block), and the link is made link is made possible by listing the possible by listing the hash of the previous block into the current block. To summarize a block consists of the data section, hash, section, hash, previous hash.

Now the created chain of blocks does not get stored in a single computer in a single computer. All the users he users have their own copy of the blockchain which known as Distributed Ledger.

If someone tries to tamper with the data, the hash value gets changed, hash value gets changed, and the link is broken and the link is broken. To make the attempt successful, the attacker needs to change and recalculate the hashes of subsequent blocks. Each block, when created, is curated by the users and based on their consensus, and the block may be added or rejected. Hence the blockchains provide Security, Immutability and Transparency.

There are three main types of blockchains in practice; they are Public, Private and Consortium Blockchains.

#### Three Parts of Blockchain

A blockchain can be studied as a database that A blockchain can be studied as a database that is distributed across its users. The essential distributed across its users. The essential requirements of a requirements of a blockchain are Peer blockchain are Peer to Peer networking, Asymmetric to Peer networking, Asymmetric Cryptography, Hashing Cryptography, Hashing.

#### Ethereum

Ethereum is a decentralized, open-source blockchain featuring smart contract functionality. Ether (ETH) is the native cryptocurrency of the platform. It is the second-largest cryptocurrency by market capitalization, after Bitcoin. Ethereum is the most actively used blockchain. Ethereum was proposed in 2013 by programmer Vitalik Buterin. Development was crowdfunded in 2014, and the network went live on 30 July 2015, with 72 million coins premised.

The Ethereum Virtual Machine (EVM) can execute Turing-complete scripts and run decentralized applications. Ethereum is used for decentralized finance, and has been utilized for many initial coin offerings. A smart contract is a piece of code lying on the block, which is used to make decisions/transactions. Ethereum uses Solidity Programming Language to write its **Smart Contracts**.

Ethereum has a native cryptocurrency called Ether (ETH) which is similar to a Bitcoin. Being a programmable blockchain, many developers can use this blockchain service in their applications.

**Smart Contracts** are tools that can automatically execute transactions if certain conditions are met without requiring the help of an intermediary company or entity. They are often associated with Ethereum, a blockchain that was designed to accommodate smart contracts, but the idea isn't restricted to any particular platform or network. So to perform or deploy the contract, a cost is associated with it, called Gas.

In general, it is expensive and slow to execute on the shared network than to perform network in a traditional setup.

### **Election Process**

The election is a formal way of making decisions. A democratic society has its foundations from voting. Elections are powerful as they are the deciding factors for the fate of an organization/country organization/country. The question of Transparency and patency and Security is still unanswered.

- Administrator – Manages and conducts the election
- Candidate – Participant in the election
- Voter – Person who is entitled to vote

Traditional elections use a centralized system where a body is trusted to conduct and manage the whole process. Some problems with this structure are administrative authority may be compromised, tampered may be occurred.

## **1.2 Aim and objective**

The main objective of this project is to build a web application using blockchain technology where people can vote from anywhere if he/she possess a valid Citizenship of respective country where he/she wants to vote and protect each and every vote to ensure that each and every vote matters.

The vast majority of the ongoing work discusses security, exactness, respectability, quickness, protection, and review capacity however existing frameworks are powerless for assaults at some degree.

### **Disadvantages of Existing System**

1. Centralized architecture.
2. Attack prone.
3. Not trustable.
4. Non-transparent vote casting process.

The existing systems are prone to attacks and are either easily hackable or very difficult to maintain.

Data integrity and security are the major concerns and the proposed solution should be able to address all the shortcomings of the existing systems.

### **1.3 Existing System**

Voting is an integral part of a democratic society. It is a decision-making mechanism and security plays an important role in voting. The existing systems are:

1. Ballot System: In India, before 2004 there was a paper-based voting system. This is called as ballot Paper system. It is placed in the election booth and is used by the voters.
2. Electronic Control System: In order to overcome duplication and damage of ballot problems Electronic Voting Machines Were introduced. It stores and assembles votes, used by poll workers.
3. Current Digital Voting Systems: A number of digital voting systems are currently in use in countries around the world. We researched some of these systems to familiarize ourselves with current implementations, particularly Estonia.

Estonia has had electronic voting since 2005 and in 2007 was the first country in the world to allow online voting. In the 2015 parliamentary election 30.5% of all votes were made through the nation's i-voting system (Vabariigi Valimiskomisjon, 2016). The bases of this system are the national ID card that all Estonian citizens are given.

These cards contain encrypted files that identify the owner and allow the owner to carry out a number of online and electronic activities including online banking services, digitally signing documents, access their information on government databases and i-voting. (Electronic ID Card, no date)

### **1.4 Problem Statement**

Several studies have been done on using computer technologies to improve elections. These studies talk about the risks of adopting electronic voting system, because of the software challenges, insider threats, network vulnerabilities, and the challenges of auditing.

### **1.5 Proposed System**

We have proposed to design the existing online voting system which is integrated with the Blockchain technology. The proposed system has the following advantages as compared to the existing system as discussed on last page

- Users' can vote from anywhere in the world until he possesses a citizenship of the country.
- The voting is stored in the Blockchain which makes it tamper proof.
- As there's no standing in queue for casting vote it will save a lot of time and reduce the workload.

We have worked the following ideas by having the two different set of modules: election commission and the voter(s). Election Commission creates elections and adds registered candidates along with the parties for contesting the election. Using an election's REST API hosted on Ethereum's Blockchain,

the details are shown at the front-end of the voter for casting the vote. Then, while polling the vote is stored on our blockchain framework of which the Election Commission fetches the vote count. The limitation which we have faced due to not using the traditional way of smart contracts is that the blockchain framework which we have coded cannot run on the main net as it needs to be hosted and a separate web3 provider must be used for interacting with it and not having a public API of voter ID creates a drawback of

not having authentication of a voter.

The objectives for developing the project are as follows:

- To improve the existing online voting system using Blockchain technology.
- To reduce the workload of setting up an election booth and conducting elections in physical form.
- Non-Resident Indian can cast their votes as it is totally online.

We are supposed to learn the concept of Blockchain and how it can be utilized to work on different sectors.

## Chapter 2

### LITREATURE SURVEY

#### 2.1. Security analysis of India's electronic voting machines

##### Abstract

Election is a very important event in a modern democracy but large sections of society around the world do not trust their election system which is major concern for the democracy. Even the world's largest democracies like India, United States, and Japan still suffer from a flawed electoral system. Vote rigging, hacking of the EVM (Electronic voting machine), election manipulation, and polling booth capturing are the major issues in the current voting system. In this paper, we are investigating the problems in the election voting systems and trying to propose the E-voting model which can resolve these issues. Also, this article aiming to evaluate the application of blockchain as service to implement distributed electronic voting systems. The section of paper will highlight some of the popular blockchain frameworks that offer blockchain as a service and associated electronic E-voting system which is based on blockchain that addresses all limitations respectively, it also preserves participant's anonymity while still being open to public inspection.

In this paper [2], it has highlighted about the major problem in voting security where in the 2016 US Presidential Elections, EVM's were likely to be intercepted and votes were tampered. The study found that this old voting equipment is not only more prone to failures and crashes but is also notoriously easy to hack and tamper with.

#### 2.2. A conceptual secure blockchain- based electronic voting system

##### Abstract

Blockchain is offering new opportunities to develop new types of digital services. While research on the topic is still emerging, it has mostly focused on the technical and legal issues instead of taking advantage of this novel concept and creating advanced digital services. In this paper, we are going to leverage the open source Blockchain technology to propose a design for a new electronic voting system that could be used in local or national elections. The Blockchain-based system will be secure, reliable, and anonymous, and will help increase the number of voters as well as the trust of people in their governments.

In this study [3] by Ayed, Ahmed, et al., it has been proposed an electronic voting system based on the Blockchain technology. The system is decentralized and does not rely on trust. Any registered voter will have the ability to vote using any device connected to the Internet. The Blockchain will be publicly verifiable and distributed in a way that no one will be able to corrupt it. Rifa and Budi has come to a conclusion that if we use of hash values in recording the voting results of each polling station linked to each other makes this recording system more secure and the use of digital signatures makes the system more reliable.

## 2.3. Blockchain Based E-Voting Recording System Design

### Abstract

Increasingly digital technology in the present helped many people lives. Unlike the electoral system, there are many conventional uses of paper in its implementation. The aspect of security and transparency is a threat from still widespread election with the conventional system (offline). General elections still use a centralized system, there is one organization that manages it. Some of the problems that can occur in traditional electoral systems is with an organization that has full control over the database and system, it is possible to tamper with the database of considerable opportunities. Blockchain technology is one of solutions, because it embraces a decentralized system and the entire database are owned by many users. Blockchain itself has been used in the Bitcoin system known as the decentralized Bank system. By adopting blockchain in the distribution of databases on e-voting systems can reduce one of the cheating sources of database manipulation.

This research discusses the recording of voting result using blockchain algorithm from every place of election. Unlike Bitcoin with its Proof of Work, this thesis proposed a method based on a predetermined turn on the system for each node in the built of blockchain. The use of the sequence proposed in the blockchain creation process in this system considers that in an electoral system not required for mining as in the Bitcoin system because the voter data and numbers are clear and are not allowed to select more than once, the proposed sequence ensures that all nodes Which is legally connected and can avoid collision in transportation [4].

## 2.4 Platform-independent Secure Blockchain-Based Voting System

### Abstract

Cryptographic techniques are employed to ensure the security of voting systems in order to increase its wide adoption. However, in such electronic voting systems, the public bulletin board that is hosted by the third party for publishing and auditing the voting results should be trusted by all participants. Recently a number of blockchain-based solutions have been proposed to address this issue. However, these systems are impractical to use due to the limitations on the voter and candidate numbers supported, and their security framework, which highly depends on the underlying blockchain protocol and suffers from potential attacks (e.g., force-abstention attacks). To deal with two issues, we propose a practical platform-independent secure and verifiable voting system that can be deployed on any blockchain that supports an execution of a smart contract. Verifiability is inherently provided by the underlying blockchain platform, whereas cryptographic techniques like Paillier encryption, proof-of-knowledge, and linkable ring signature are employed to provide a framework for system security and user-privacy that are independent from the security and privacy features of the blockchain platform. We analyze the correctness and coercion-resistance of our proposed voting system. We employ Hyperledger Fabric to deploy our voting system and analyze the performance of our deployed scheme numerically.

Bin, Joseph, et al., has come to a conclusion that the current blockchain voting system cannot provide the comprehensive security features, and most of them are platform dependent, we have proposed a blockchain based voting system that the voters' privacy and voting correctness are guaranteed by homomorphic encryption, linkable ring signature, and PoKs between the voter and blockchain [5]

## **2.5 System Analysis**

### **Identification of Need**

Identification of need is a process of determining what and how an end-user would expect a product to perform after the deployment at production level. There's also nontechnical needs of an end-user or a business client which reflects the users' perception of the product and not the actual technical workaround, but they are closely related to the technical need at times. By implementing a needs identification system, the organization helps to ensure the proper allocation of assets to different project within the organization.

### **Identifying Problems**

Identifying potential problems before the start of a project can save the organization significant amounts of time and money. Problem analysis is one of the most critical stages of project planning because this stage helps to guide all subsequent analysis and decision-making. If the project does not advance past this stage with solutions that the organization can implement, the project should not go forward in its current form.

### **Observations**

The needs for a project are identified after the organization makes observations about the project. Observations are often subjective and therefore someone with expertise about the proposed project should help to make observations. A good observer can identify the needs of the project by answering key questions about the project. If the observations take into consideration the project itself and the outcome of the project, the observations should meet all of the needs of the project.

### **Gathering Information**

Observation and gathering information represent two processes. Observations highlight what is needed. On the other hand, gathering information highlights the processes needed to execute the proposed project. Both observations and the actual gathering of information should include comments from the group that ultimately will benefit from the completed project.

### **Objectives and Opportunities**

Once the organization has analyzed the needs and identified the objectives, the organization needs to allocate funds to capitalize the project. By successfully identifying the needs, an organization can begin to allocate resources to pay for the project. Additionally, a business needs to consider the potential future cash flow of the project. This allows the business to analyze potential cost savings to minimize costs and maximize the efficiency of the project.



## **Preliminary Investigation**

The main aim of preliminary investigation is to identify the problem. First, need for the new or the enhanced system is established. Only after the recognition of need, then the proposed system is compared and then further analysis is possible. At this stage, we had to perceive the problem and opportunities, the existing system is studied and found out that there were few areas where we can integrate with other technology to make the system better than the existing system. It was analyzed that such proposed system would be possible to develop with given and it might turn out to be the feasible solution. In this project, the biggest challenge was to integrate the existing online voting system with the designed blockchain framework and on further development levels we encountered various unit level problems such as the model for the Election Commission to create votes and store the necessary details of candidates along with the election details. On the later part of this document, we have come up with the features which can be added to our software to make it better than the initial deployment.

# SYSTEM SPECIFICATION

### 3.1 Functional Requirements

Functional Requirement defines a function of software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing and other specific functionality. The functional requirements of the project are one of the most important aspects in terms of entire mechanism of modules.

**The functional requirements here are:**

- **Maintaining user:** interface responsiveness: If the application needs to perform a time-consuming task, multiple threads can be used to prevent user interface from becoming unresponsive while the task is in progress. If the program is downloading information from the Internet, this will keep the user-interface running at nearly full-speed while the download is in progress.
- **Simple Multitasking:** Multitasking allows to execute multiple instances of a process quit easily. The downloading routine just mentioned can be extended so that the program can transfer multiple files simultaneously and still keep the user interface well behaved. All that is needed is to create another thread for each file to download.
- **Building Multi-user Applications:** Multithreading is often used when building server applications. Server applications wait for request to arrive and then establish conversations with the requester.
- **Multiprocessing:** Many operating systems support machines with multiple processors. Most of these systems are unable to break a single thread of execution for execution on different processors. By breaking an application into different Threads, it is possible to make the best use of processing power.

## **3.2 Non-Functional Requirements**

### **3.2.1 Reliability**

The framework ought to be dependable and solid in giving the functionalities. When a client has rolled out a few improvements, the progressions must be made unmistakable by the framework. The progressions made by the Programmer ought to be unmistakable both to the Project pioneer and in addition the Test designer.

### **3.2.2 Security**

Aside from bug following the framework must give important security and must secure the entire procedure from smashing. As innovation started to develop in quick rate the security turned into the significant concern of an association. A great many dollars are put resources into giving security. Bug following conveys the greatest security accessible at the most noteworthy execution rate conceivable, guaranteeing that unapproved clients can't get to imperative issue data without consent. Bug following framework issues diverse validated clients their mystery passwords so there are limited functionalities for all the clients.

### **3.2.3 Maintainability**

The framework observing and upkeep ought to be basic and target in its approach. There should not be an excess of occupations running on diverse machines such that it gets hard to screen whether the employments are running without lapses.

### **3.2.4 Performance**

The framework will be utilized by numerous representatives all the while. Since the framework will be facilitated on a solitary web server with a solitary database server out of sight, execution turns into a noteworthy concern. The framework ought not succumb when numerous clients would be utilizing it all the while. It ought to permit quick availability to every last bit of its clients. For instance, if two test specialists are all the while attempting to report the vicinity of a bug, then there ought not to be any irregularity at the same time.

### **3.2.5 Portability**

The framework should be effectively versatile to another framework. This is obliged when the web server, which is facilitating the framework gets adhered because of a few issues, which requires the framework to be taken to another framework.

### **3.2.6 Scalability**

The framework should be sufficiently adaptable to include new functionalities at a later stage. There ought to be a typical channel, which can oblige the new functionalities.

### **3.2.7 Flexibility**

Flexibility is the capacity of a framework to adjust to changing situations and circumstances, and to adapt to changes to business approaches and rules. An adaptable framework is one that is anything but difficult to reconfigure or adjust because of diverse client and framework

prerequisites. The deliberate division of concerns between the trough and motor parts helps adaptability as just a little bit of the framework is influenced when strategies or principles change.

### **3.3 Hardware Requirements**

- Processor type: Intel core i5 and above
- Processor speed: Minimum 2.00 GHz and above
- RAM: 6-10 GB
- HARD DISK: 400 GB or more
- Monitor: 800x600 or higher resolution
- Keyboard: 110 keys enhanced

### **3.4 Software Requirements**

- Operating System: Windows 7 (32 bit and 64 bit) and Above
- Development Environment: Solidity Programming, Web Development (CSS, HTML, JAVA SCRIPT)
- Scripting Language: Solidity Programming, PHP
- Decentralized Applications: Ethereum Framework ie. Ganache
- Browser: Google Chrome
- Add-on in Browser: Metamask
- Software: Visual Studio or Similar IDE

## **User Requirements**

The program works by considering the input from the user as he/she need to prove his identity.

- Browser: Google Chrome
- Add-on in Browser: Metamask and Internet Connectivity

### **3.5 Project Planning**

Project Planning is the most essential thing in developing a project. It sets out the phases, activities and task needed to deliver a project. The timeframes required to deliver the project, along with the resources and milestones are also shown on the project plan.

Initially, the project scope is defined and the appropriate methods for completing the project are determined. Following this step, the durations for the various tasks necessary to complete the work are listed and grouped into a work breakdown structure. Project planning is often used to organize different areas of a project, including project plans, workloads and the management of teams and individuals. The logical dependencies between tasks are defined using an activity network diagram that enables identification of the critical path. Project planning is inherently uncertain as it must be done before the project is started. Therefore, the duration of the tasks is often estimated through a weighted average of optimistic, normal, and pessimistic cases. The critical chain method adds “buffers”; in the planning to anticipate potential delays in project execution. Float or slack time in the schedule can be calculated

using project management software. Then the necessary resources can be estimated and costs for each activity can be allocated to each resource, giving the total project cost. At this stage, the project schedule may be optimized to achieve the appropriate balance between resource usage and project duration to comply with the project objectives. Once established and agreed, the project schedule becomes what is known as the baseline schedule. Progress will be measured against the baseline schedule throughout the life of the project. Analyzing progress compared to the baseline schedule is known as earned value management.

A project plan is a model of the process that the project team intends to follow to realize the project objectives. It brings together several important aspects of this process including its scope, timing and associated risks. The project plan can be viewed as a type of “contract” between the project team members and the reviewers. It defines the process by which objectives will be achieved, and the responsibilities in carrying out this process. It also underpins a number of other key project management functions including estimating and forecasting, options analysis and decision-making, and performance monitoring and control.

The essential elements of a project plan are:

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

### **Scope statement**

It is a statement of what work is included within the project, and what is not. A good scope statement significantly reduces risk of project overruns and unexpected turbulence. In this project, the scope statement is as follows:

“This project is for the creation of an online election system using Blockchain technology. There will be a website for Election Commission and for the voters. The user interface will be designed as part of the project which will contain necessary details at both the end.”

### **Schedule**

The project schedule communicates to all stakeholders what the expected arrival time will be, and serves to keep the project manager’s hands on the throttle throughout the project. Since projects are temporary endeavors with a defined beginning and end, the exact location of that end date is a primary consideration for most projects.

The details of this project’s schedule will be discussed later under Project Scheduling.

### **Requirements**

All projects have requirements which are drafted at the beginning as per client’s needs. In this project, the requirements are such as the module of creating elections, adding candidates contesting the elections. Detailed discussion of the requirements is discussed under Requirement Specifications.

## **Quality Criteria**

It is one of the essential elements of project planning as if a software is not inspected properly and then deployed to the market it might cause few problems which will then create pressure among the maintenance. The quality criteria should be identified in the project plan, including pass/fail requirements, as well as the methods used to ensure the quality criteria will be met.

## **Project Resources**

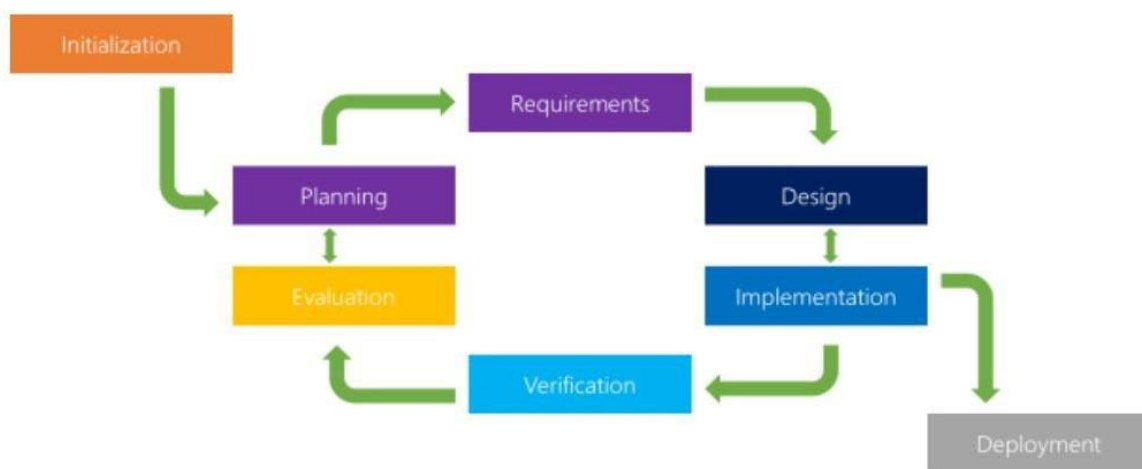
Resources often require the most planning and coordination throughout the project's execution. That's because they arrive late, require unexpected maintenance, don't meet specifications, or any other host of issues that can trip up a project. Resources are the technology stack which will be used in developing the software. Details about this is discussed at the later part of this documentation.

## Chapter 4

### SYSTEM DESIGN AND ARCHITECTURE

#### 4.1 Software Design

A project plan is a model of the process that the project team intends to follow to realize the project objectives. It brings together a number of important aspects of this process including its scope, timing and associated risks. The project plan can be viewed as a type of “contract” between the project team members and the reviewers. It defines the process by which objectives will be achieved, and the responsibilities in carrying out this process. It also underpins a number of other key project management functions including estimating and forecasting, options analysis and decision-making, and performance monitoring and control.



*Fig 4.1 : Iterative Model of SDLC*

This project uses an iterative model approach using Agile methodologies. Let’s discuss this in details. Agile methods of software development are most commonly described as iterative and incremental development. The iterative strategy is the cornerstone of Agile practices, most prominent of which are SCRUM, DSDM, and FDD. The general idea is to split the development of the software into sequences of repeated cycles (iterations). Each iteration is issued a fixed-length of time known as a timebox. A single timebox typically lasts 2-4 weeks. The ADCT (Analysis, Design, Code, Test) wheel is more technically referred to as the PDCA (Plan, Design, Check, Adjust) cycle. The team implements the PDCA cycle on each iteration separately in the following manner:

- **P (Plan) – Iteration Planning**

In this event, the team collaborates to discuss the objectives for the next iteration. It also summarizes the work done and determines the team backlog required for the next iteration.

- **D (Design) – Iteration Execution**

This is the ‘do’ step where the development of the software, its design and coding takes place. If it’s a second or third iteration, then functionality testing is also conducted. The team collects user stories and prepares for the next step, that is the Iteration Review.

- **C (Check) – Iteration Review**

Also known as the ‘check’ step, Iteration Review is carried out with the Product Owner. The team shows the tested deliverable to the Product Owner, who then reviews the completed work and ascertains whether all criteria have been met.

- **A (Adjust) – Iteration Retrospect**

In this event, the team evaluates the entire process of the iteration from the first step. It essentially works on any improvements that are gathered in previous iterations. New problems are identified along with their causes. Before the team starts the next cycle again, team backlog is refined for future reference. The iterations are repeated for optimizations and improvisations and, the lessons learned from previous cycles are applied in the next cycle. Until a fully functional software is ready to hit the market.

Agile methodologies have the following advantages over other methods:

**Customer Involvement** – Agile Iterative development encourages user contribution. After each iterative cycle, customer feedback is obtained, and the product is then subjected to necessary changes based on that feedback. This aspect brings adaptability into the project’s framework.

**Favors Evolution** – The planning in the Agile Iterative development process is a continuous feat, that allows space for evolving ideas, instead of extensive planning that only precedes execution and testing in Waterfall.

**Risk Assessment** – Agile iteration allows risk identification and mitigation early in the development to avoid speed bumps later down the timeline.

**Rapid Delivery** – The work is divided into small cycles, allowing team members to dedicate their focus and deliver on time. Moreover, testing is conducted simultaneously in coding and design in every iteration, which greatly reduces the time needed to achieve completion.

## **Process**

- **Planning & Requirements:** As with most any development project, the first step is to go through an initial planning stage to map out the specification documents, establish software or hardware requirements, and generally prepare for the upcoming stages of the cycle.

- **Analysis & Design:** Once planning is complete, an analysis is performed to nail down the appropriate business logic, database models, and the like that will be required at this stage in the project. The design stage also occurs here, establishing any technical requirements (languages, data layers, services, etc.) that will be utilized in order to meet the needs of the analysis stage.

- **Implementation:** With the planning and analysis out of the way, the actual implementation and coding process can now begin. All planning, specification, and design docs up to this point are coded and implemented into this initial iteration of the project.

- **Testing:** Once this current build iteration has been coded and implemented, the next step is to go through a series of testing procedures to identify and locate any potential bugs or issues that have cropped up.



• **Evaluation:** Once all prior stages have been completed, it is time for a thorough evaluation of development up to this stage. This allows the entire team, as well as clients or other outside parties, to examine where the

The essential elements of a project plan are:

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

### **Scope statement**

It is a statement of what work is included within the project, and what is not. A good scope statement significantly reduces risk of project overruns and unexpected turbulence. In this project, the scope statement is as follows:

“This project is for the creation of an online election system using Blockchain technology. There will be a website for Election Commission and for the voters. The user interface will be designed as part of the project which will contain necessary details at both the end”.

### **Schedule**

The project schedule communicates to all stakeholders what the expected arrival time will be, and serves to keep the project manager's hands on the throttle throughout the project. Since projects are temporary endeavors with a defined beginning and end, the exact location of that end date is a primary consideration for most projects. The details of this project's schedule will be discussed later under Project Scheduling.

### **Requirements**

All projects have requirements which are drafted at the beginning as per client's needs. In this project, the requirements are such as the module of creating elections, adding candidates contesting the elections. Detailed discussion of the requirements is discussed under Requirement Specifications.

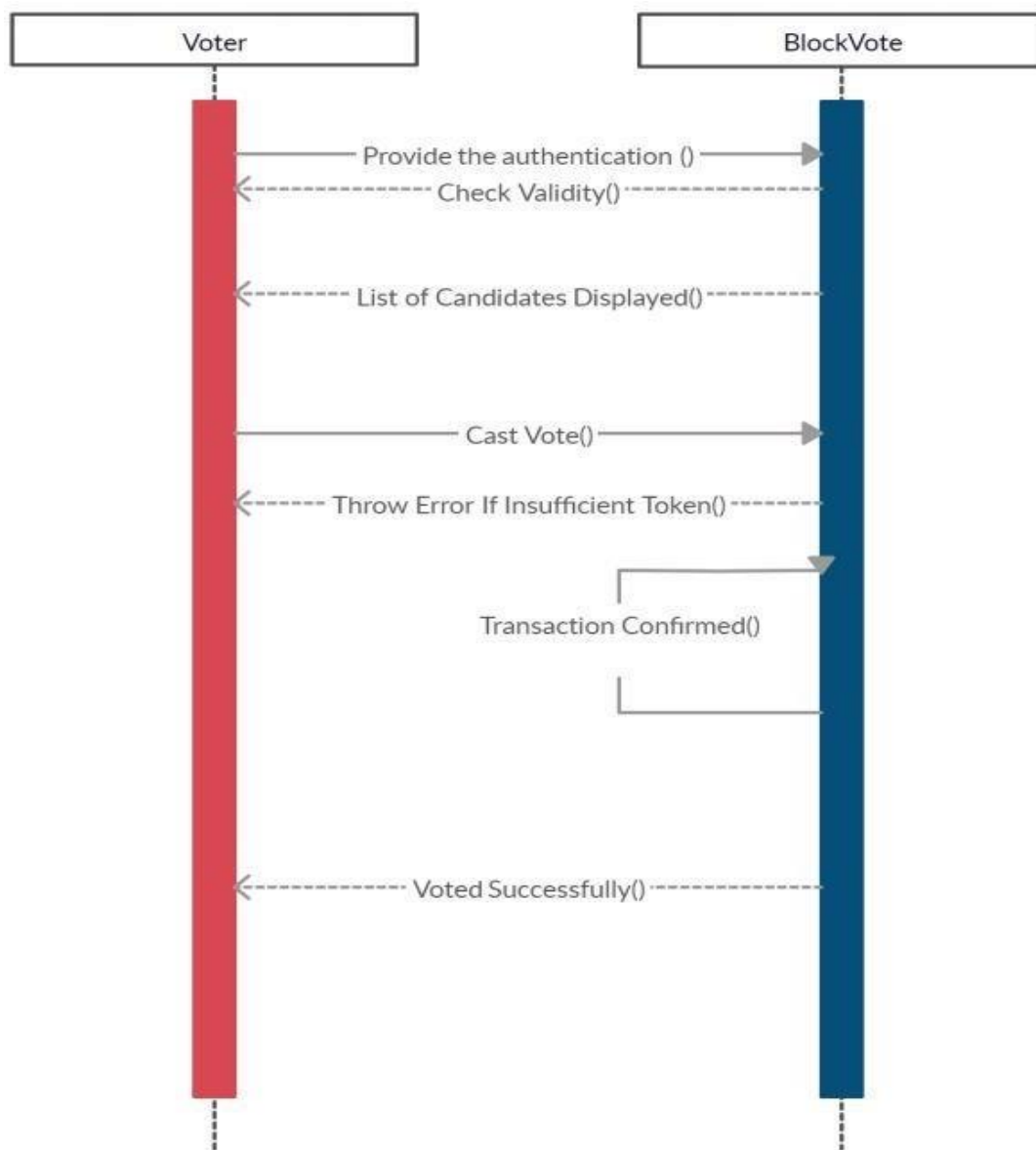
### **Quality Criteria**

It is one of the essential elements of project planning as if a software is not inspected properly and then deployed to the market it might cause few problems which will then create pressure among the maintenance. The quality criteria should be identified in the project plan, including pass/fail requirements, as well as the methods used to ensure the quality criteria will be met.

## Project Resources

Resources often require the most planning and coordination throughout the project's execution. That's because they arrive late, require unexpected maintenance, don't meet specifications, or any other host of issues that can trip up a project. Resources are the technology stack which will be used in developing the software. Details about this is discussed at the later part of this documentation.

## Sequence Diagram



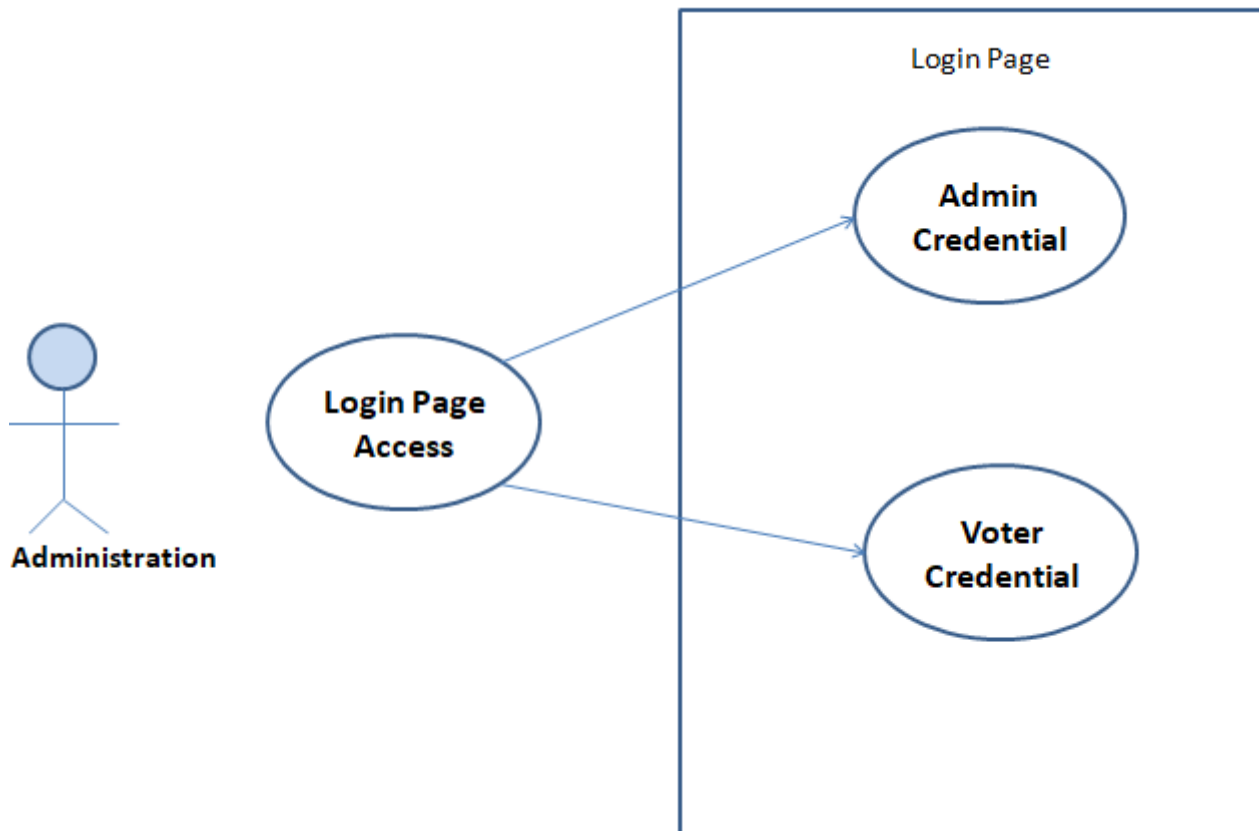
**Fig 4.2 : Sequence Diagram**

**MetaMask** was created to increase the accessibility of the Ethereum blockchain to the average user. A plug-in for Chrome, MetaMask acts as an Ethereum browser, allowing users to manage their Ethereum wallet and interact with decentralized applications and smart contracts without running a full node. Through MetaMask, users are able to manage multiple accounts and easily switch between different networks. In order to allow users the flexibility of using the Ethereum blockchain without running a full

node, MetaMask relies on trusted nodes to broadcast the transactions of MetaMask users in order to be mined. Since transactions are signed using the sender’s private key, which is stored locally on the user’s machine, MetaMask cannot impersonate the user and send transactions on the user’s behalf. Acting as an intermediary between Chrome and the Ethereum blockchain, MetaMask allows users the convenience and security of the blockchain within a popular browser.

## 4.2 System Design

### 4.2.1 Use case: Users



*Fig 4.3 : User Login Page*

#### 4.2.2 Use case : Admin

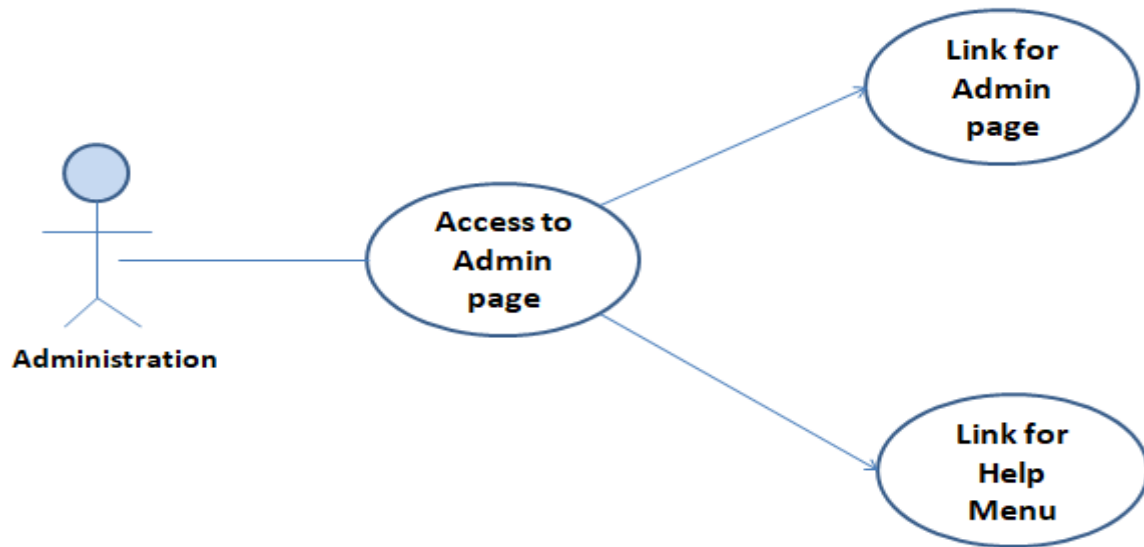


Fig 4.4 : Admin Login Page

#### 4.2.3 Use case : Add New Users

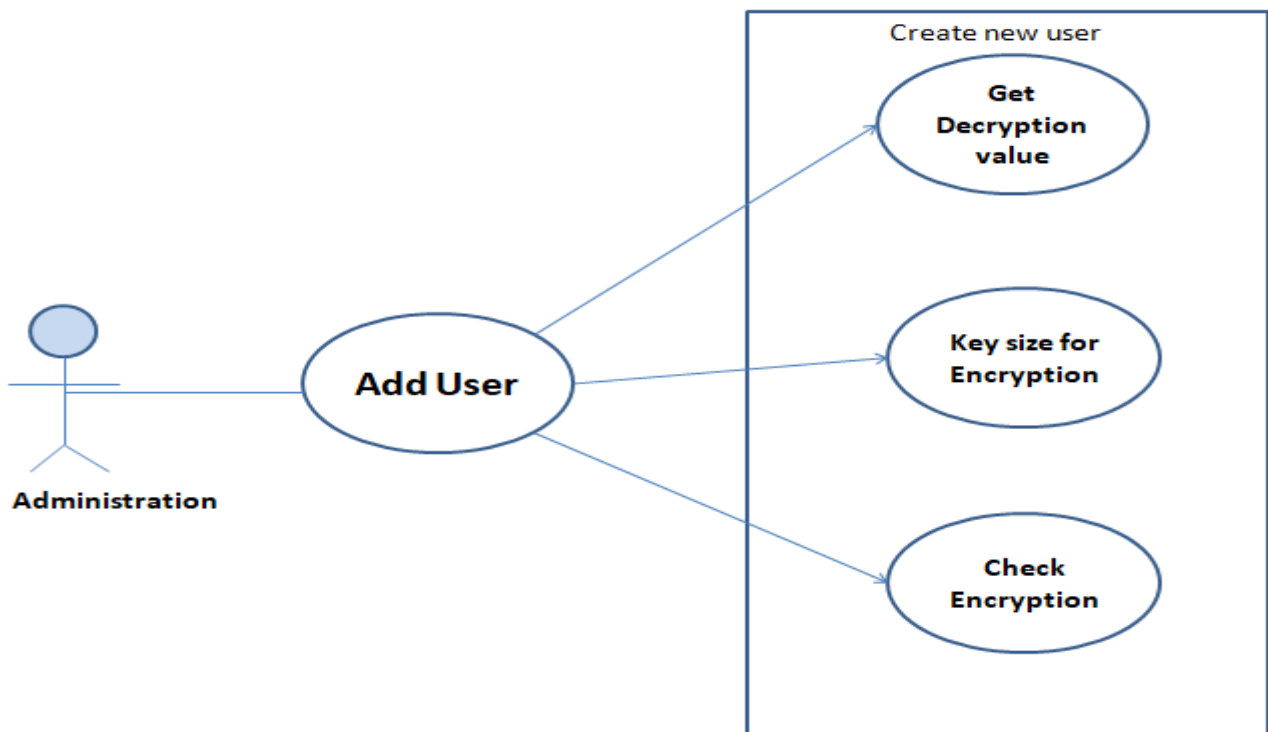
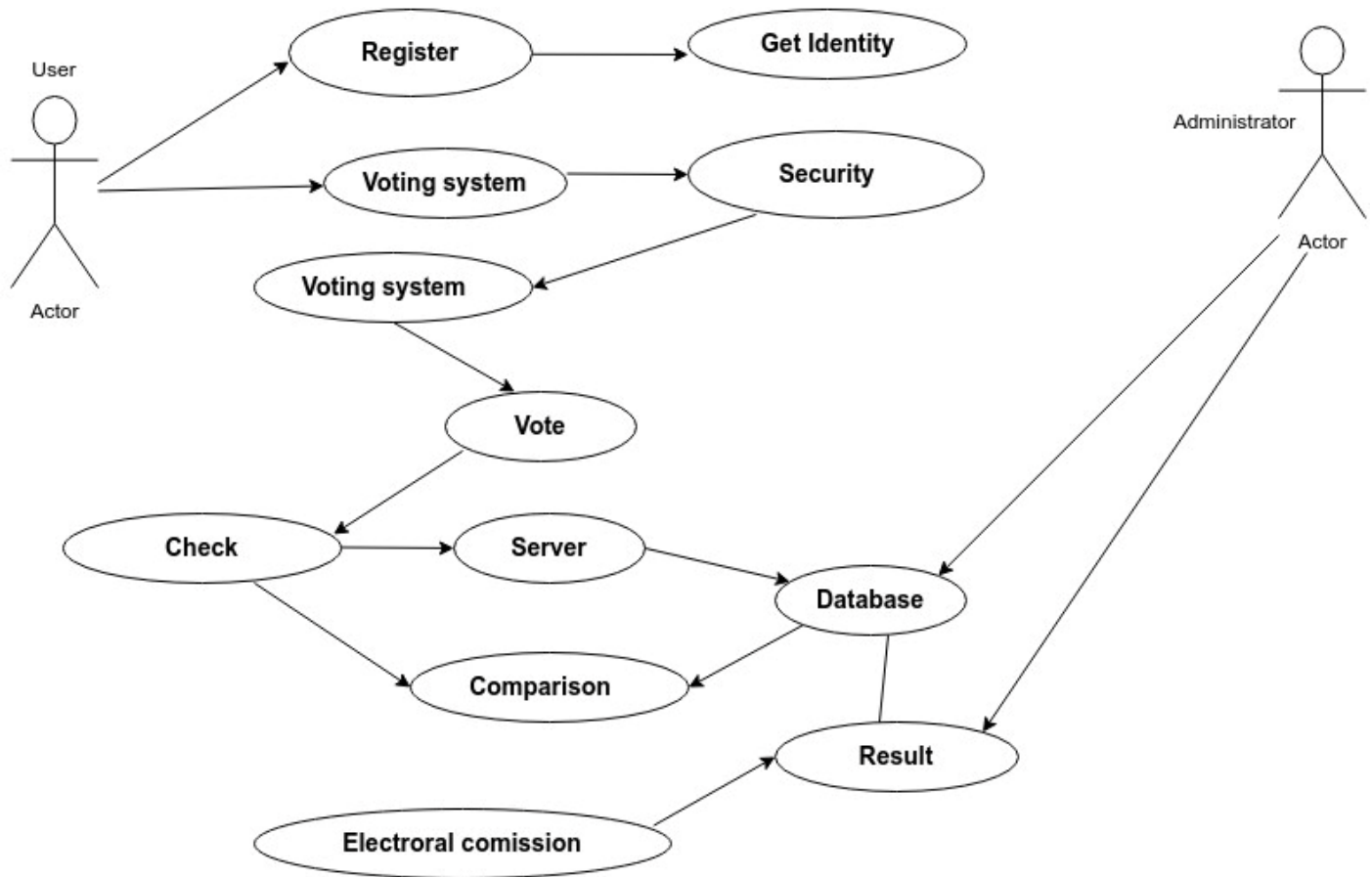


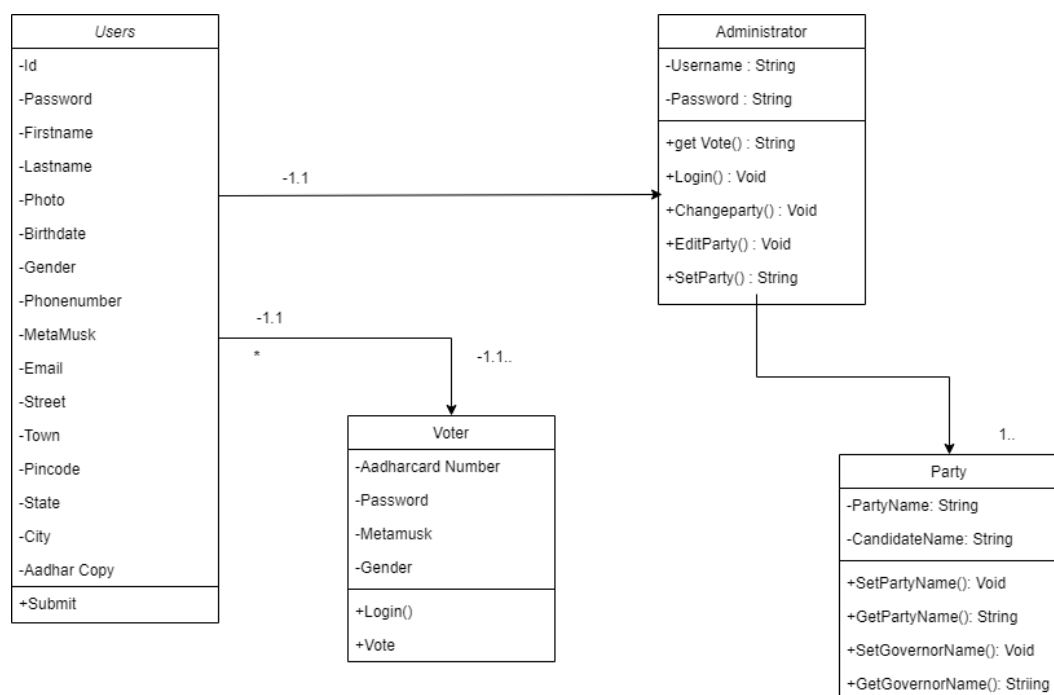
Fig 4.5 : Add New Users

#### 4.2.4 Use Case: System use case



*Fig 4.6 System use case*

#### 4.3 Class Diagram



*Fig 4.7 Class Diagram*

4.4. Data Flow Diagram:



Fig:7 0- level DFD

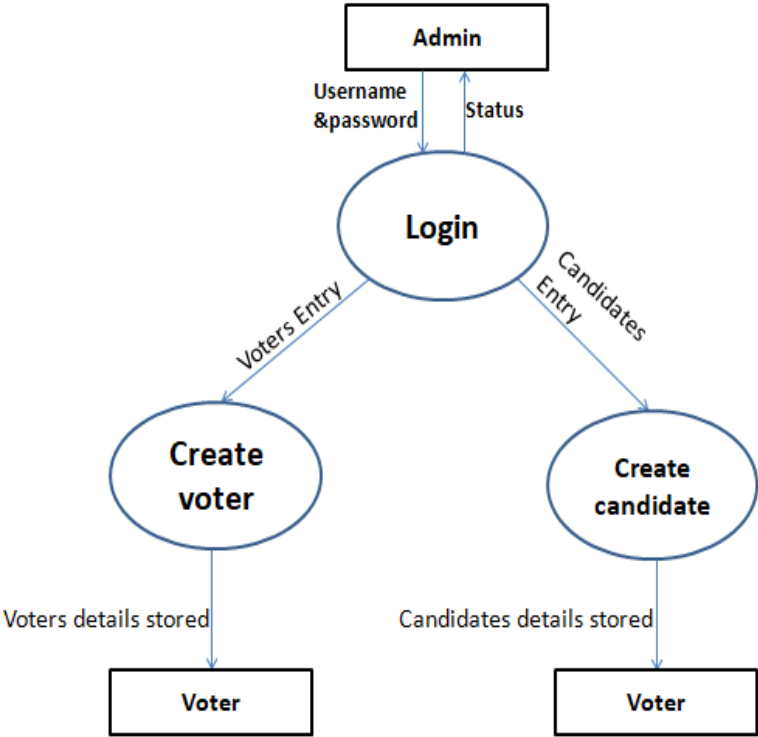
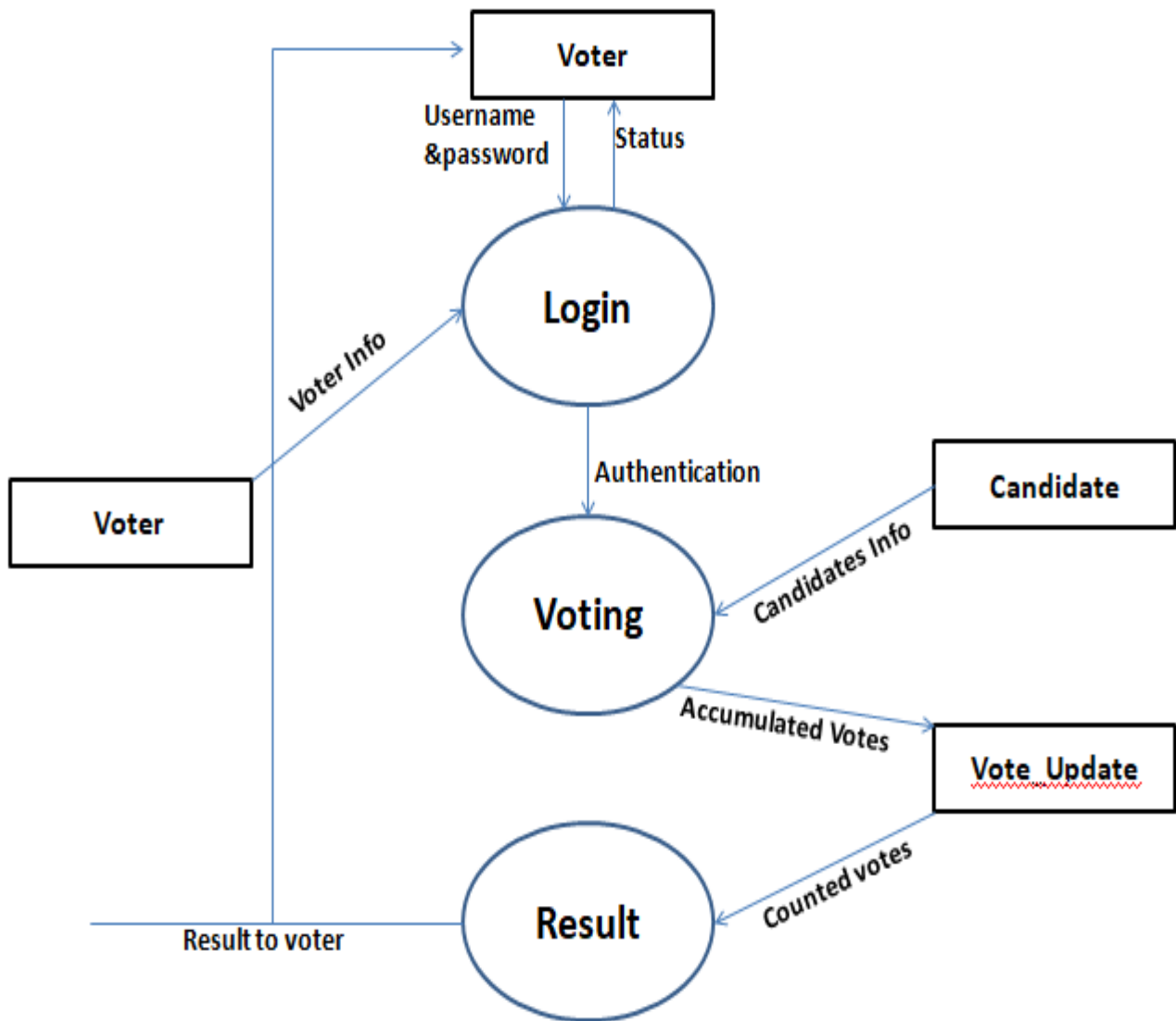


Fig:8 1- level DFD



*Fig:9 2- level DFD*

## 4.5 Data Strategy

### 4.5.1 Admin

Sn No.	Field Name	Data Type	Description	Constraints
1	admin_id	int	Sr no.	Primary Key
2	Name	varchar	Admin name	Not null
3	First name	varchar	First Name	Not null
4	password	varchar	Security password	Not null
5	Img	varchar	Admin photo	Not null
6	Last name	varchar	Name	Not null
7	Date	varchar	Date	Not null

*Table 4.1 Admin*

### 4.5.2 Candidate

Sn No.	Field Name	Data Type	Description	Constraints
1	Admin_id	int	Sr no.	Primary Key
2	Position_id	varchar	Admin name	Not null
3	First name	varchar	Candidate Name	Not null
4	Last name	varchar	Candidate Name	Not null
5	Symbol	varchar	Party Symbol	Not null
6	Image	varchar	Image	Not null
7	Platform	text	Commitment	Not null

*Table 4.2 Candidate*



### 4.5.3 Voters

Sr No.	Field Name	Data Type	Constraints
1	Id	Int	Primary Key
2	Voters Id	Varchar	Not Null
3	Password	Varchar	Not Null
4	First Name	Varchar	Not Null
5	Last Name	Varchar	Not Null
6	Photo	Varchar	Not Null
7	Birth Date	Varchar	Not Null
8	Gender	Varchar	Not Null
9	Phone Number	Varchar	Not Null
10	Meta Mask	Varchar	Not Null
11	Email	Varchar	Not Null
12	Street 1	Varchar	Not Null
13	Street 2	Varchar	Not Null
14	Town	Varchar	Not Null
15	Pincode	Int	Not Null
16	State	Varchar	Not Null
17	City	Varchar	Not Null
18	Aadhar Copy	Varchar	Not Null
19	Ccity	Varchar	Not Null
20	Town Village	Varchar	Not Null

***Table 4.3 Voters***

#### 4.5.4 Position

Sr No.	Field Name	Data Type	Constraints
1	Id	Int	Primary Key
2	Description	Varchar	Not Null
3	Max vote	Int	Not Null
4	Priority	Int	Not Null

*Table 4.4 Position*

#### 4.5.5 Vote

Sr No.	Field Name	Data Type	Constraints
1	Id	Int	Primary Key
2	Voters Id	Int	Not Null
3	Candidate Id	Int	Not Null
4	Position Id	Int	Not Null

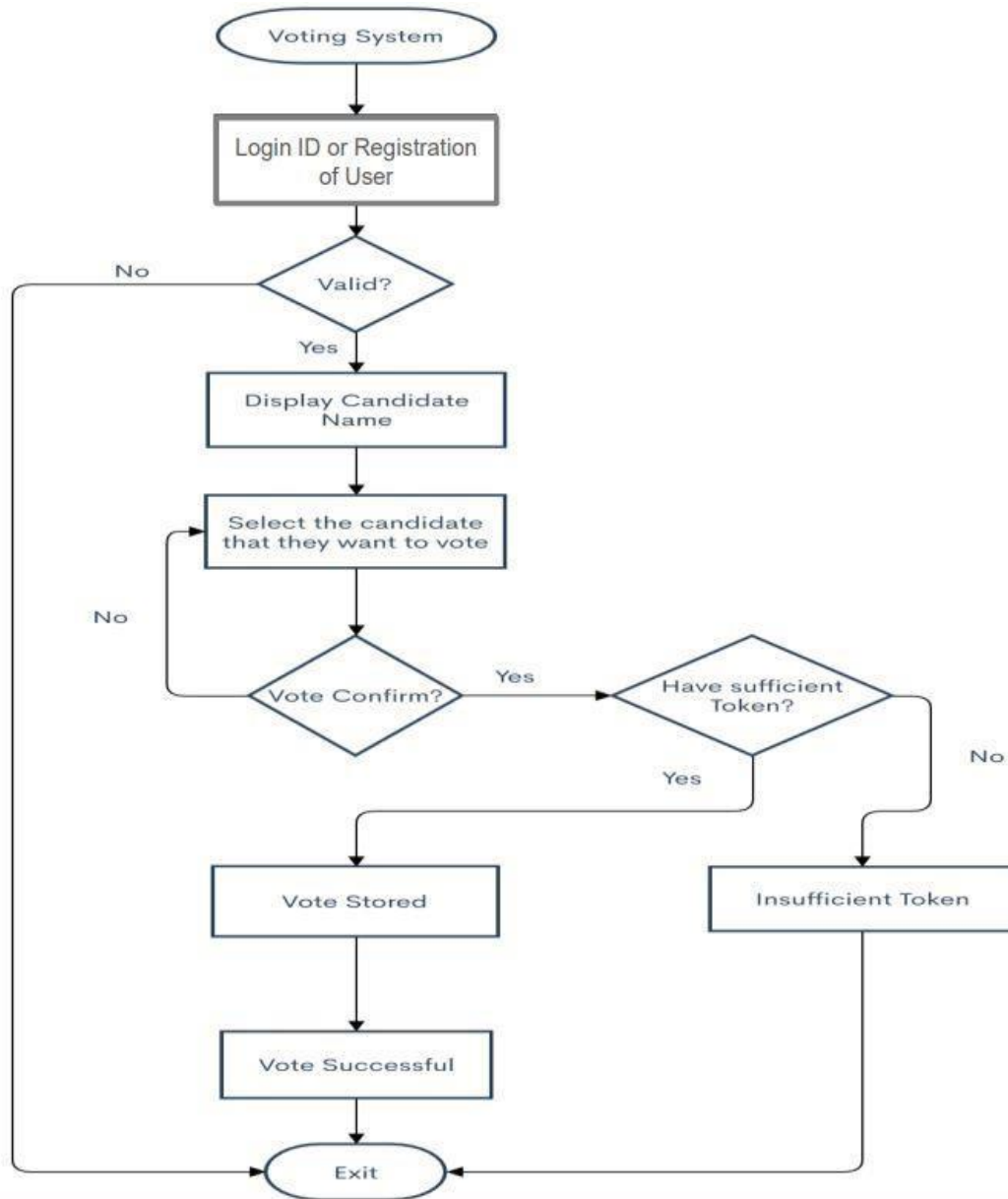
*Table 4.5 Vote*

## Chapter 5

### IMPLEMENTATION

#### 5.1 Flowchart

Below is the Overall Flowchart of this project.



*Fig 5.1: Flowchart for Voting*

## 5.2 Modules

The project has been divided into many modules in which for every functionality we have designated modules. Any software comprises of many systems which contains several sub- systems and those sub-systems further contains their sub-systems. So, designing a complete system in one go comprising of each and every required functionality is a hectic work and the process can have many errors because of its vast size.

Effective modular design can be achieved if the partitioned modules are separately solvable, modifiable as well as compliable. Following are the project modules:

- **Candidate:** The candidates should be a set of list. A candidate in a organization/ community who stand for election should submit the details to RA. Candidate The candidates should be a set of list For each candidate to vote can be defined as  $C_i$
- **Registration Authority:** The voter should register in RA to get ready to vote. The candidate should register in RA with his information and the RA will give him the id of candidate.
- **Voter:** The voters should be a set of list For each voter to vote can be defined as  $V_i$  The voter should transfer his public key(PKI/Token) to EA. : In this module, voters who have been provided with the personal ETH wallet will import onto the voting portal using the Metamask extension and cast their vote. Voter registers in our system with a valid student/employee ID and e-mail address to vote on given ballot ID numbers.
- **Election Authority (Admin):** The EA is responsible for starting the election, creating a vote, limiting the voter numbers of The voting, paying the voting fees for the Transaction generated automatically in the backend. In this module, an entity named Election Commission will be responsible to setup the smart contract and register candidates, parties and start off an election. **Administrator** is responsible for deploying the initial Registrar and Creator smart contracts. The administrator also has the ability to grant or revoke ballot creation permission for registered voters/creators.
- **Solidity Programming:** It acts as the record and gate keeper. It keeps track of all registered voters and creators, ballot IDs, voting contract addresses, and whitelisted e- mail domains. The information regarding the voter and different ballots are linked together in the contract. This allows the contract to perform voter verification, permission modification, and Voting.sol address retrieval. The owner of this contract is the administrator.

The tiers given below alludes to **different level or layers where activities occur.**

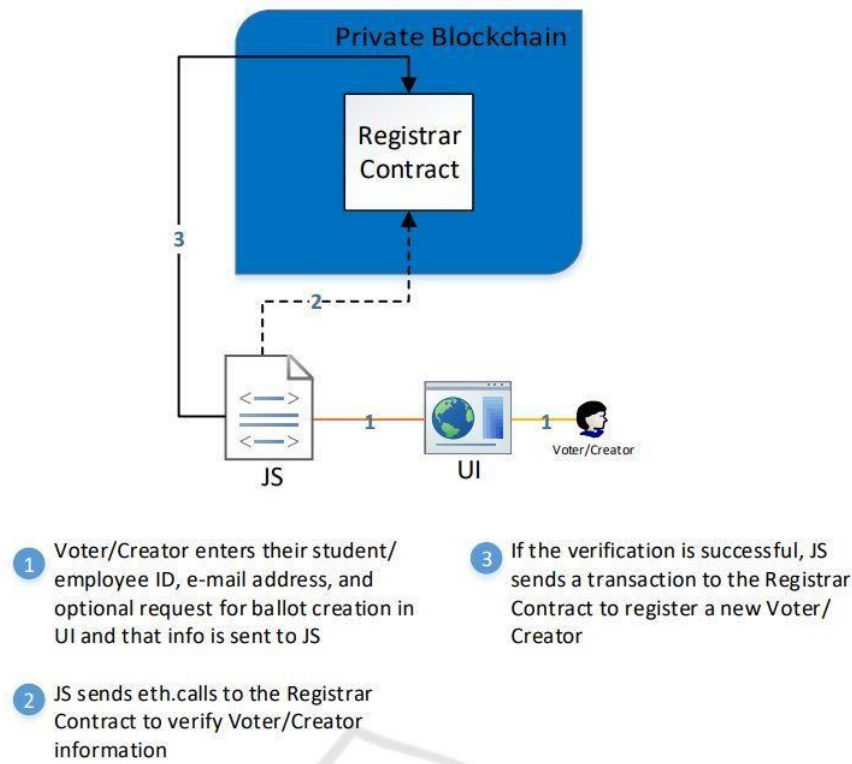
**Client:** Client is any user or program that wants to perform an operation over the system. Clients interact with the system through a presentation layer.

**Presentation Layer:** This layer is responsible for the presentation of data at the client side, i.e., it provides an interface for the end-user into the application to cast the votes.

**Resource manager:** The resource manager deals with the organization (storage, indexing and retrieval) of the data necessary to support the application logic. This resource manager here is the Local Blockchain server maintained by Ganache.

**Application logic:** The application logic figures out what the system does. It takes care implementing the business rules and establishing the business processes. Blockchain voting system is designed and

implemented according to the three-tier architecture.



*Fig 5.2: Private Blockchain Concept*

### 5.3 Overall Implementation Process

#### Initial Setup

The administrator is responsible for the initial deployment of both the Registrar and Creator contracts to activate the system and enable users to start registering, voting, and creating new voting contracts. When deploying the Registrar Contract, the administrator is also responsible for whitelisting a set of e-mail domains that are allowed to register to be part of the voting system.

#### Register Voter

This phase of the voting process will be the most similar to the current system. A trusted third party will still be required to take the role of the Electoral Commission and oversee the registration process. Voters will still be required to register for their constituency votes, with the only difference being that user IDs for the voting application will be sent out, rather than polling cards as in the current system. It makes eth.calls (Ethereum) to the registrar contract to verify the domain provided is part of the whitelist and if the user has previously registered. If those checks are passed, then it sends a transaction to the registrar contract to store the new voter information, including the voter's ID, Ethereum address, and e-mail. It links the user's Ethereum address and e-mail address so that they cannot double register. Individuals can also request access to create ballots during the registration process; these requests are planned to be manually processed by the administrator but currently are granted automatically.

## Create Ballot

If the user has permissions to create a ballot, the user can spawn a new voting contract by entering the required information in Registration.php . In order to create a ballot, the creator must provide their registered e-mail address then decide whether to create an election or poll, determine the title of the ballot, voting options, and number of votes allowed per voter. During this process, the creator can also elect to have a whitelisted ballot. If a whitelisted ballot is chosen, the creator enters the list of e-mail addresses allowed to vote on their ballot. If the creator chooses to not make a whitelisted ballot, everyone with a e-mail address that has the whitelisted domain will be allowed to vote. Lastly, the creator sets the end date and time of the election or poll.

## Load Ballot

Using the ballot ID provided by the Creator of the Voting Contract, a voter can check the results or vote on the ballot, provided the voting period has not passed. Once the voter enters the ballot ID in E-vote Interface Voting App Java Script sends an eth.call to the Registrar Contract to determine the validity of the ballot ID. If the ballot ID is valid, the voting options, title, and encrypted vote count for each choice if the voting period has ended unless it is a poll. If the ballot type was a poll, then the results are displayed live. Before the vote count can be displayed, there is another step involved, that involves sending the encrypted vote count to the metamask server so that we can display the tallied vote for each choice on E-vote Interface.

## Vote

The voting process itself will be vastly different. Registered voters will use the provided IDs to log into the application, where they will have the ability to create a transaction on each of the channels on which they are eligible to vote. This transaction will allow them a choice of available candidates for the channel, or the option "None of the above" for those wishing to register a process vote. Upon submission, this transaction will alter the ownership of the associated ballot to the chosen candidate. Once ownership of the ballot is changed, it cannot be reverted - like the finality of dropping the vote into a ballot box in the current system.

Once the ballot has been loaded, the user can vote for a particular choice on the ballot with his/her registered e-mail address. When the voter clicks vote, VotingApp.js receives the information and sends eth.calls to the Registrar Contract to verify the voter, it checks the voter registration and Ethereum address. If the voter is verified, an eth.call is sent to the Voting Contract to check whether the ballot is whitelisted or not.

## Get Votes

This phase will benefit the most from the increased transparency afforded by the voting process. Immediately upon the end of the voting period, an event will be triggered granting read-only access across all channels to all participants. All participants will be able to scrutinize the votes cast in each channel, or constituency, for discrepancies. Shared queries will allow the participants and electoral authorities to immediately understand who won, and, in elections where demographic information is kept, the sections of the electorate that backed the winners. getVotes acts as a data retrieval function. Whenever a user loads the ballot or successfully votes on a ballot, getVotes is invoked in Voting App JavaScript file, getVotes sends an eth.call with the hashed choices to get the current total encrypted votes. Depending on the time limit and election type, it would either decrypt the votes and display them or display the time when users can check back for the results. To decrypt the votes, getVotes sends the encrypted vote count to the truffle or metamask server to be decrypted by the private key.

## **Data Security**

Security is about risk management, so it is important to start with an understanding of the risk associated with the blockchain solutions. The specific risks of a blockchain solution depends on the type of blockchain being used. Let us take a look at the various types of blockchains with decreasing level of risks and increasing levels of security:

- Public Blockchains are public and anyone can join them and validate transactions. They are generally riskier (for example, cryptocurrencies). This includes risks where anyone can be part of the blockchain without any level of control or restrictions.
- Private blockchains are restricted and usually limited to business networks; membership is controlled by a single entity (regulator) or consortium.
- Permissionless blockchains have no restrictions on processors.
- Permissioned blockchains allow the ledger to be encrypted so that only relevant participants can see it, and only those who meet a need-to-know criterion can decrypt it.

There are a number of other risks with blockchain solutions, and they can be broadly categorized into three areas:

- Business and governance: Business risks include financial implications, reputational factors, and compliance risks. Governance risks emanate primarily from the decentralized nature of blockchain solutions, and require strong controls on decision criteria, governing policies, identity, and access management.
- Process: These risks are associated with the various processes that a blockchain solution requires in its architecture and operations.
- Technology: The underlying technology used to implement various processes and business needs may not always be the best choice, and this can ultimately lead to security risks.

## **Security controls unique to blockchain**

- API security best practices are used to safeguard API-based transactions.
- Data classification are adopted for the approach to safeguard data/information.
- The appropriate endorsement policies are defined and endorsed based on business contracts.
- Secrets-store for both application and privileged access is leveraged.

## **ADVANTAGES**

- Cost cutting
- Transparent voting
- Faster process
- Lesser manpower
- Tamper proof voting

**Others Advantages:**

- Accurate results and speed in vote count
- Low cost of setup because just internet connection cost is required to vote across all the available e-voting platforms
- Enhanced security as voting takes place over secure communication channels
- Accessibility from any corner of the world just by having an internet connection
- Fraud prevention due to less human intervention therefore avoiding the fraud that could possibly take place at the polling stations
- Reduced influence by family members or peers as voters can change their opinion until the end of the voting day several times as only the last vote will be considered

**DISADVANTAGES**

- In many developing countries internet access is not available to everyone, example: In rural areas low wage workers could not afford internet also many people don't know how to use and access the web
- E-voting machines use software to register the vote and it is built by a company, general public do not know how a software works that might lead to fraudulent results being generated, vendors could also be bribed and in return they could tweak the software to work in their favor
- In the internet voting voter has to login by providing their personal and ID details, which will result in "Voter Anonymity" issue.



## Chapter 6

### TERMONOLOGIES

#### 6.1 Languages and Technology

##### 6.1.1 PHP

**PHP: Hypertext Pre-processor** is a widely used, general-purpose scripting language that was originally designed for web development to produce dynamic web pages. For this purpose, PHP code is embedded into the HTML source document and interpreted by a web server with a PHP processor module, which generates the web page document.

As a general-purpose programming language, PHP code is processed by an interpreter application in command-line mode performing desired operating system operations and producing program output on its standard output channel. It may also function as a graphical application. PHP is available as a processor for most modern web servers and as standalone interpreter on most operating systems and computing platforms.

PHP was originally created by Rasmus Lerdorf in 1995 and has been in continuous development ever since. The main implementation of PHP is now produced by the PHP Group and serves as the *de facto* standard for PHP as there is no formal specification. PHP is freeware released under the PHP License.

##### 6.1.2 WAMPP

**WAMPP** is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. Since most actual web server deployments use the same components as WAMPP, it makes transitioning from a local test server to a live server possible.

WAMPP's ease of deployment means a WAMP or LAMP stack can be installed quickly and simply on an operating system by a developer. With the advantage of common add-in applications such as WordPress and Joomla! can also be installed with similar ease using Bitnami. Officially, WAMPP's designers intended it for use only as a development tool, to allow website designers and programmers to test their work on their own computers without any access to the Internet. To make this as easy as possible, many important security features are disabled by default. WAMPP has the ability to serve web pages on the World Wide Web.[14] A special tool is provided to password-protect the most important parts of the package.[15] WAMPP also provides support for creating and manipulating databases in MariaDB and SQLite among others.

##### 6.1.3 phpMyAdmin :

**phpMyAdmin** is a free and open source administration tool for MySQL and MariaDB. As a portable web application written primarily in PHP, it has become one of the most popular MySQL administration tools, especially for web hosting services. By that time, phpMyAdmin had already become one of the most popular PHP applications and MySQL administration tools, with a large community of users and contributors. In order to coordinate the growing number of patches, a group of three developers (Olivier Müller, Marc Delisle and Loïc Chapeaux). registered *The phpMyAdmin Project* at SourceForge and took over the development in 2001. In July 2015, the main website and the downloads left SourceForge and moved to a content delivery network. At the same time, the

releases began to be PGP-signed. Afterwards, issue tracking moved to GitHub and the mailing lists migrated. Before version 4, which uses Ajax extensively to enhance usability, the software used HTML frames.

#### **6.1.4 HTML**

HTML 5 is a markup language used for structuring and presenting content on the World Wide Web. It is the fifth and current version of the HTML standard.

It was published in October 2014 by the World Wide Web Consortium (W3C) to improve the language with support for the latest multimedia, while keeping it both easily readable by humans and consistently understood by computers and devices such as web browsers, parsers, etc. HTML5 is intended to subsume not only HTML 4, but also XHTML 1 and DOM Level 2 HTML.

HTML5 includes detailed processing models to encourage more interoperable implementations; it extends, improves and rationalizes the markup available for documents, and introduces markup and application programming interfaces (APIs) for complex web applications. For the same reasons, HTML5 is also a candidate for cross-platform mobile applications, because it includes features designed with low-powered devices in mind.

#### **6.1.5 Cascading Style Sheets**

**Cascading Style Sheets (CSS)** is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications. CSS is designed primarily to enable the separation of presentation and content, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

Separation of formatting and content makes it possible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. It can also display the web page differently depending on the screen size or viewing device. Readers can also specify a different style sheet, such as a CSS file stored on their own computer, to override the one the author specified.

#### **6.1.6 JAVASCRIPT**

JavaScript is one of the three core technologies of World Wide Web content production; the majority of websites employ it, and all modern Web browsers support it without the need for plug-ins. JavaScript is a multi-paradigm language, since it supports prototype-based with first-class functions, imperative, and functional programming paradigms. It has an API for working with text, arrays, dates, regular expressions, and basic manipulation of the DOM, but does not include network, storage, or sophisticated graphics APIs, relying instead upon APIs made available by its host environment.

Although there are strong outward similarities between JavaScript and Java, including language name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design; JavaScript was influenced by programming languages such as Self and Scheme. JavaScript is also used in environments that are not Web-based, such as PDF documents, site-specific browsers, and desktop widgets. Newer and faster JavaScript virtual machines (VMs) and platforms built upon them have also increased the popularity of JavaScript for server-side Web applications. On the client side, developers have traditionally implemented JavaScript as an interpreted language, but more recent browsers perform just-in-time compilation. Programmers also use JavaScript in video-game development and in desktop and mobile applications.

### 6.1.7 Bootstrap

**Bootstrap** is a free and open-source front-end web framework for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many web frameworks, it concerns itself with front-end development only.

Bootstrap is the second most-starred project on GitHub, with more than 107,000 stars and 48,000 forks. Bootstrap 3 supports the latest versions of the Google Chrome, Firefox, Internet Explorer, Opera, and Safari (except on Windows). It additionally supports back to IE8 and the latest Firefox Extended Support Release (ESR).

Since 2.0, Bootstrap supports responsive web design. This means the layout of web pages adjusts dynamically, considering the characteristics of the device used (desktop, tablet, mobile phone)

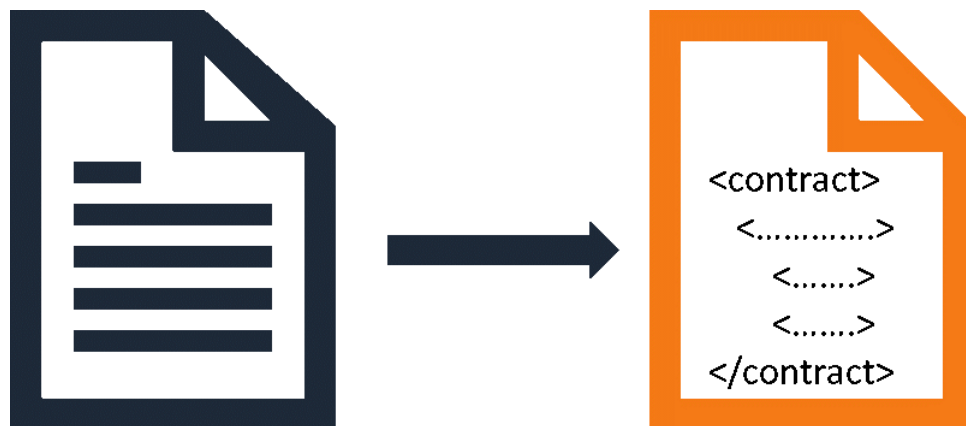
## 6.2 Smart Contract Concept

“Smart contracts” is a term used to describe computer code that automatically executes all or parts of an agreement and is stored on a blockchain-based platform. As discussed further below, the code can either be the sole manifestation of the agreement between the parties or might complement a traditional text-based contract and execute certain provisions, such as transferring funds from Party A to Party B. The code itself is replicated across multiple nodes of a blockchain and, therefore, benefits from the security, permanence and immutability that a blockchain offers. That replication also means that as each new block is added to the blockchain, the code is, in effect, executed. If the parties have indicated, by initiating a transaction, that certain parameters have been met, the code will execute the step triggered by those parameters. If no such transaction has been initiated, the code will not take any steps. Most smart contracts are written in one of the programming languages directly suited for such computer programs, such as Solidity.

The smart contract is executed through a blockchain network, and the code of the contract is replicated on many computers that comprise the network. This ensures a more transparent and secured facilitation and performance of the contractual terms. Moreover, smart contracts do not require a middleman to execute because the code of a smart contract is verified by all the participants in the blockchain network. The removal of the middleman from the contract helps to substantially reduce the costs for counterparties.

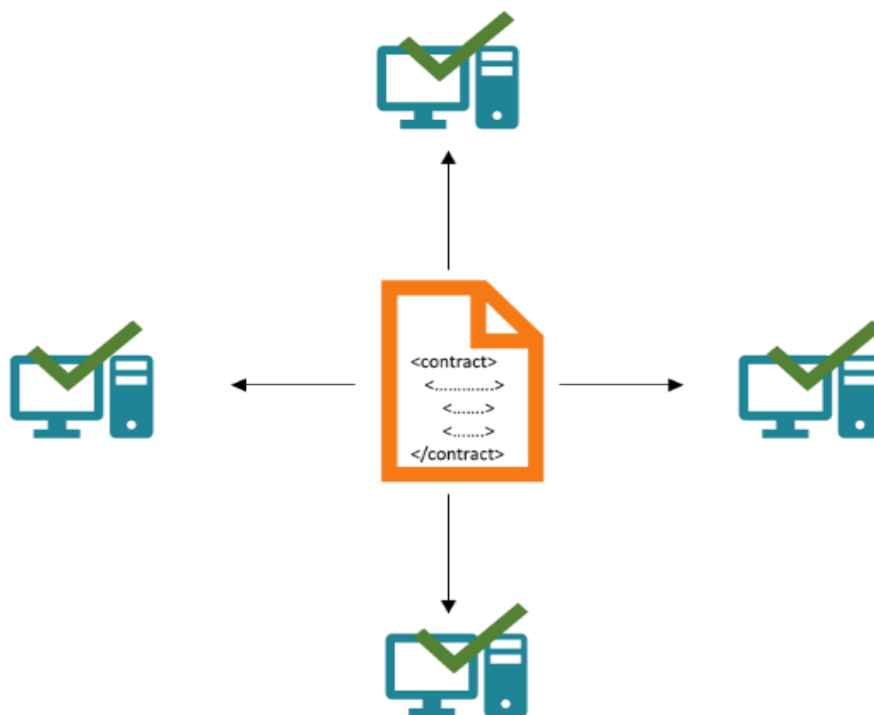
## How do smart contracts work?

First, the contractual parties should determine the terms of the contract. After the contractual terms are finalized, they are translated into programming code. Basically, the code represents several different conditional statements that describe the possible scenarios of a future transaction.



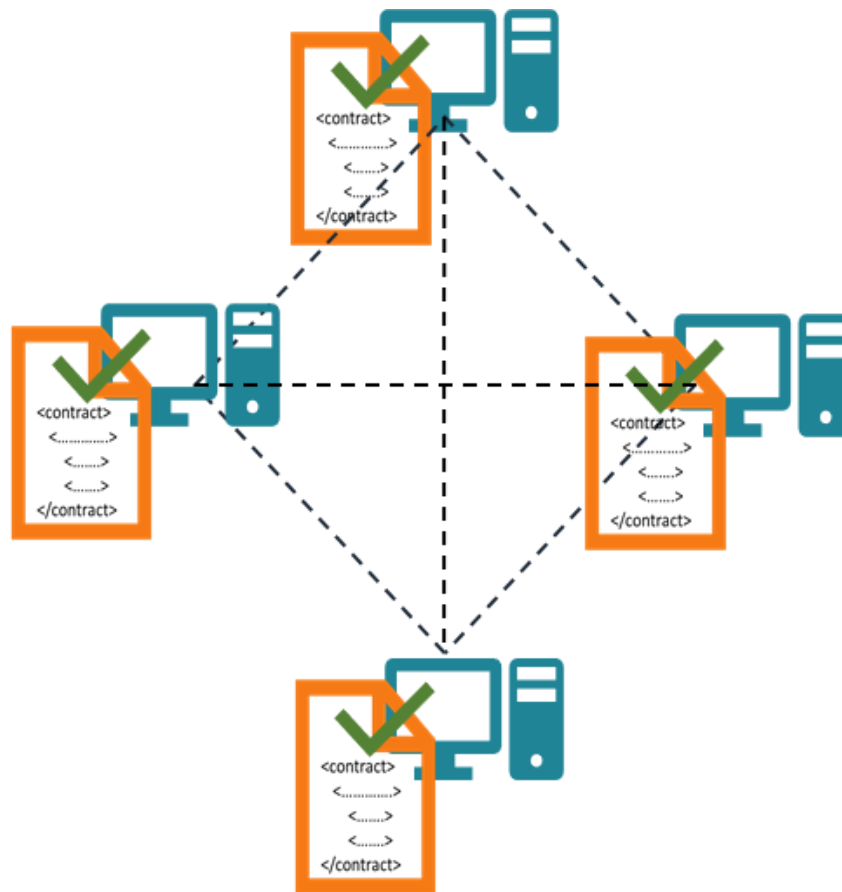
**Fig 6.1: Transfer contract terms into code**

When the code is created, it is stored in the blockchain network and is replicated among the participants in the blockchain.



**Fig 6.2: The code is stored in a blockchain and replicated between participants**

Then, the code is run and executed by all computers in the network. If a term of the contract is satisfied and it is verified by all participants of the blockchain network, then the relevant transaction is executed.

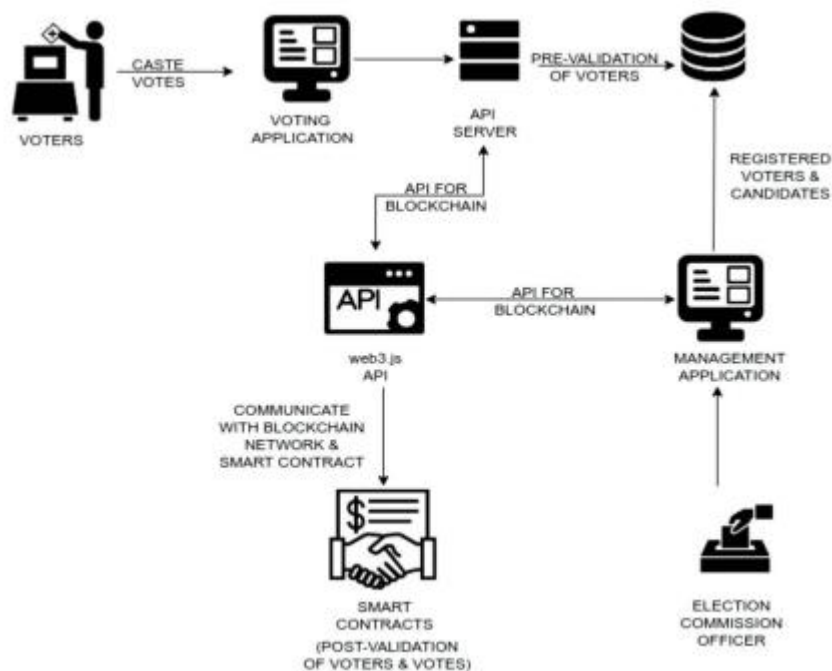


**Fig 6.3:** *When a term is satisfied, computers in the network verify its correctness.*

### 6.3 Methodology

Within our proposal we have tried to design a service and system that minimizes the size of attack vectors to prevent potential malicious attacks. We have tried to evaluate and analyze our design from various perspectives to make sure we have thought about each step of the voting process. This section of the report discusses the potential risks associated with our proposal and suggests actions that can be taken to help mitigate them. One risk is if a voter were to forget their ID, password, or polling card on the day of voting. In this case the voter will be unable to cast their vote as they cannot enter the system. Possible risk mitigations include the voter returning later that day with the correct information or the implementation of a backup authentication service such as by phone. Alternatively, a forgotten password system could be added to the voter registration website; this could work in much the same way as recovering a password works on other websites. However, this increases the risk of a hacker attempting to change a voter's password without their knowing. A 51% attack is a potential threat to our proposed design. The basis of the attack being that someone could theoretically control a majority of the digital voting mining hash-rate, leading to them being able to manipulate the public ledger. The chances of this type of attack occurring are slim due to the immense cost needed to purchase hardware capable of this scale of processing. We also have the added security of an auditor who checks and

keeps track of people connecting to the network and the locations of each node. This is a feature that current systems such as bitcoin lack. (Learn cryptography.com, 2016) The online aspect of the voting within our system is the largest attack vector for hackers as they could potentially exploit voters through their own devices in a host of ways. To combat this software could be developed that could be downloaded onto the client's device to establish a secure connection to the polling station. When it is time to vote, authentication of a user requires three distinct pieces of evidence; their identification number (e.g. UK citizens have national insurance numbers), the password supplied on registration, their ballot card which contains a QR code. As there are two methods of voting (web browser, physical polling station) the way the user will input the authentication details shall differ; however, in order to vote they are required to provide all three pieces of information. It is also important to note that each user will have been registered at a certain constituency so they will only be able to vote at a local polling station within that constituency or via the internet at the URL provided on the ballot card. (Each constituency is to be equipped with its own web server and URL to ensure votes are aggregated within the right network.) Behind the scenes the polling station will consult the voter blockchain to ensure the voter has not already used up their vote. If the user does have a vote, then the station will then allow the user to continue to the voting screen. If not, then system will respond to the user appropriately. See diagram Appendix B Figure 6 to see the process. After selecting their vote (from the selection of options including abstention) and then confirming the submission, the vote will become a transaction, it will be encrypted with the relevant constituency's public key. This transaction is then passed to the constituency node where it is added to a block and the update is then pushed to all other nodes connected to that constituency node. The connected nodes then pass the data on to their peers until the whole network is updated. Once the vote has been confirmed the polling station will then generate a transaction to remove the user's vote within the voter blockchain. It is important to note that there are two distinct blockchains being held; one which contains transactions relating to which users have registered and which users still have a vote, the second containing the contents of the vote (such as what party was voted for.). Through the use of these two distinct blockchains we ensure voter anonymity when selecting their vote.



**Fig 6.4 Architecture of Purposed system**

Preceding the introduction to our voting system, it merits mentioning that the Ethereum protocol utilized as part of our system has not been modified in any way. Our system, E-voting, uses existing functionality and features provided by Ethereum to provide the ability for creating and voting on ballots. Our implementation consists of three smart contracts coded in Ethereum's Solidity language, two scripts written in JavaScript, and one HTML page. E-voting is an open source project and the entirety of the code is available for public use. We assume the administrator, creators, and voters have the MetaMask plugin downloaded in their browser or running an Ethereum node to create and manage Ethereum accounts as well as interact with our system. We utilize Ethereum's Web3 framework internally, this allows our users to easily manage signed transactions and interactions with the Ethereum blockchain. Using MetaMask and Web3 eliminates the need for users to download full or even partial Ethereum blockchains on their local machines in order to broadcast transactions. The only action required of users when registering, voting, or creating ballots is to use their passwords to unlock their Ethereum accounts in the MetaMask plugin and securely interact with the blockchain. If the user decides not to utilize the Metamask plugin then they are responsible for running a node on their local machine and syncing it with the blockchain to interact with our system using Web3.

**Web3.** js is a collection of libraries that allow you to interact with a local or remote ethereum node using HTTP, IPC or WebSocket.

## Chapter 7

# TESTING

### 7.1 Testing

In the world today, technology is used to create several machines and make life easier. The software could have multiple bugs and might not be working as it is intended to. Hence testing is required to control and make sure that the software is error-free. It identifies all the defects that the software and makes sure the software is meeting the required specifications. Testing is also very cost-effective. It prevents failure from occurring in the future. It will also be cheap to fix the bugs when it is in an earlier stage. It also improves the quality of the products after it is tested. This project uses *Mocha* as the testing framework to unit test and integration test all of our test cases for the application. Following strategies are used:

(i) **Unit Testing:** This is the first and the most important level of testing. Its need begins from the moment a programmer develops a unit of code. Every unit is tested for various scenarios. Detecting and fixing bugs during early stages of the Software Lifecycle helps reduce costly fixes later on. It is much more economical to find and eliminate the bugs during early stages of application building process. Hence, Unit Testing is the most important of all the testing levels. As the software project progresses ahead it becomes more and more costly to find and fix the bugs. Steps for Unit Testing are:- Step 1: Creation of a Test Plan Step 2: Creation of Test Cases and the Test Data Step 3: Creation of scripts to run the test cases wherever applicable Step 4: Execution of the test cases, once the code is ready Step 5: Fixing of the bugs if present and re testing of the code Step 6: Repetition of the test cycle until the Unit is free from all types of bugs.

**Integration Testing:** Integration strategy stands for how individual modules will be combined during Integration testing. The individual modules can be combined in one go, or they can be joined one by one. A decision on how to put the pieces together is called the Integration Strategy. We have used bottom-up integration approach to integrate test our application. In Bottom Up Integration, we move from the bottom to top i.e. the components below are first written and these are integrated first. The integration happens from bottom to top. If the calling component is yet to be developed, it is replaced by a specially written component called a Driver.

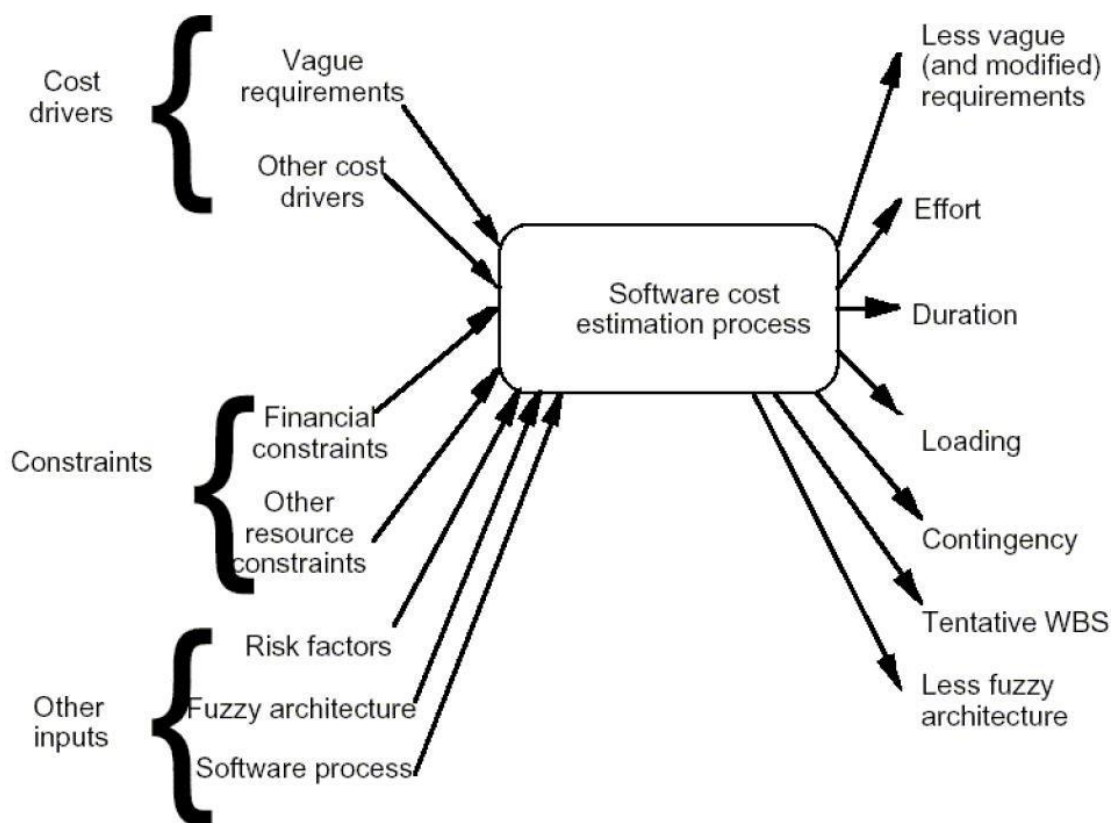
### 7.2 Cost Estimation and Project Planning

The people who do the cost estimates could be either directly or indirectly responsible for the implementation for a project, such as a developer or manager, respectively. Someone who has knowledge of the organization and previous projects could use an analogy-based approach to compare the current project with previous projects, which is a common method of estimation for small organizations and small projects. The historical data is often limited to the memory of the estimator. In this case, the estimator would need to be experienced and would likely have been with the company for a while. Some people believe it is better if the estimates are done by outsiders so that there is less chance of bias. It is true that people outside an organization will likely have to deal with fewer company politics than people within the organization. For example, the developer for a company may want to please the manager and so give an estimate that is overly-optimistic. The disadvantage of



having an outside estimate is that the person would have less knowledge of the development environment, especially if the person is from outside the company. An empirical method of estimation would then be required, such as the Constructive Cost Model (COCOMO). Empirical methods of estimation can be used by all types of estimators. There may be some resistance to using an empirical method of estimation because there may be some question on whether a model could outperform an expert. People who are accurate estimators are rare in our experience, and so it is best to get the opinion of several people or tools.

In the actual cost estimation process, there are other inputs and constraints that needed to be considered besides the cost drivers. One of the primary constraints of the software cost estimate is the financial constraint, which are the amount of the money that can be budgeted or allocated to the project. There are other constraints such as manpower constraints, and date constraints. Other input such as architecture, which defines the components that made up the system and the interrelationships between these components. Some company will have certain software process or an existing architecture in place; hence for these companies the software cost estimation must base their estimates on these criteria. There are only very few cases where the software requirements stay fixed. Hence, how do we deal with software requirement changes, ambiguities or inconsistencies? During the estimation process, an experienced estimator will detect the ambiguities and inconsistency in the requirements. As part of the estimation process, the estimator will try to solve all these ambiguities by modifying the requirements. If the ambiguities or inconsistent requirements stay unsolved, which will correspondingly affect the estimation accuracy





**Fig 7.1 : Actual Cost Estimation Process**



Cost estimation is an important tool that can affect the planning and budgeting of a project. Because there are a finite number of resources for a project, all of the features of a requirements document can often not all be included in the final product. A cost estimate done at the beginning of a project will


help determine which features can be included within the resource constraints of the project (e.g., time). Requirements can be prioritized to ensure that the most important features are included in the product. The risk of a project is reduced when the most important features are included at the beginning because the complexity of a project increases with its size, which means there is more opportunity for mistakes as development progresses. Thus, cost estimation can have a big impact on the life cycle and schedule for a project.

### 8.1 Home Screen

 [ujjawalpatel19@gnu.ac.in](mailto:ujjawalpatel19@gnu.ac.in)

 9157400902






[Home](#) [About](#) [Voter Guide](#) [Political Parties](#) [News](#) [Contact](#) [My Account](#)


### GUJARAT STATE

# Chief Electoral Officer

Conducts Elections to Gujarat Legislative Assembly & Parliamentary seats from Gujarat. Elections are conducted according to the constitutional provisions, supplemented by laws made by Parliament. Gujarat State Election Commission is an autonomous and statutory body constituted in Indian state of Gujarat for ensuring that elections in Gujarat are conducted in free, fair and unbiased way.

[More Information](#)





**Smt P. Bharathi, IAS**  
**Chief Electoral Officer, Gujarat**

#### ABOUT US

## CEO Message

The common perception about the election process is that it is not citizen-friendly. This perception is primarily due to lack of information about the process. This website aims to make the election process easier to understand and increase citizen participation in the election process. While the main concern of the citizen is how to enrol his name in the electors list, the lack of information about how to go about this process is the main hurdle for the citizen. This website aims to remove that hurdle. A new feature of the forthcoming elections will be the affidavits that will be filed by the candidates. All these affidavits will be made available on the website to enable the voters to make informed choices. While every effort is made to make the website as citizen-friendly as possible, suggestions are always welcome.

#### VOTER GUIDE

## Happy To Help You



Registration



Connect Wallet



KYC



e-Learning



Fraud Scam

### What do you need to do today?

Once a citizen is eligible to vote and has enrolled, a voter slip from the ECI will be issued that confirms the name on the voter list. This slip, along with a stipulated photo ID proof, can act as a voter card, in case one does not have a voter ID card.

- ✓ Enroll To Vote
- ✓ Check My Enrollment
- ✓ Update My Details
- ✓ Find My Ward number
- ✓ Know My BLO



#### OUR POLITICAL PARTIES

## See Our Political Parties



Indian National Congress

Hand



Aam Aadmi Party

Broom



Shiv Sena

Bow and Arrow



Samajwadi Party

Bicycle



Bharatiya Janata Party

Lotus

RECENT NEWS

## Check Our News



Economic Times

19 Aug 2022

### Ahead of assembly polls, unspent Rs 950 cr allocated for SC welfare schemes

New Delhi: In a major political outreach to the Scheduled Caste community, the Centre has decided to allocate ₹950 crore of unutilised budget of eight infrastructure ministries to the Ministry of Social Justice and Empowerment for infrastructure development and income-generation schemes for SC welfare.



By: Nidhi Sharma

[Discover More](#)



Economic Times

16 Aug 2022

### If elected, AAP will ensure that Gujarat has top-notch education

Arvind Kejriwal, on Tuesday that if elected in the Gujarat elections, he will deliver high-quality education.



The Hindu

17 Aug 2022

### State BJP chief C.R. Paatil welcomed Naresh Raval and former Raju Parmar into the party by offering them saffron scarves and caps.



India Today

17 Aug 2022

### Congress leaders in Gujarat, Himachal Pradesh join BJP ahead of state polls

## Get In Touch With Us Now



012-345-6789



gujarat-election@email.com



Sardar Patel Bhavan, Sector No. 10 Block 9, 6th Floor, Gujarat 382010

Name

Your Email

Subject

Message

[Send Message Now](#)

## 8.2 Signin and Signup

The image shows a web interface titled "Login for Vote". It features a white "Sign in" panel on the left and a blue "Hello, Voter's!" panel on the right. The "Sign in" panel has input fields for "AadharCard" and "Password", and a "SIGN IN" button. The "Hello, Voter's!" panel has the text "Enter your personal details and start voting today" and a "SIGN UP" button. The background is a light gray with a colorful illustration of Indian architecture at the bottom.

*Fig 8.2 Signin and Signup*

## 8.3 Registration form of voting

### REGISTRATION FORM OF E-VOTING

The image shows a registration form titled "REGISTRATION FORM OF E-VOTING". The form is divided into two main sections: "PERSONAL INFORMATION" and "METAMASK WALLET ADDRESS". The "PERSONAL INFORMATION" section includes fields for "FIRST NAME", "LAST NAME", "BIRTH DATE", "GENDER", and "PHONE NUMBER". The "METAMASK WALLET ADDRESS" section includes a field for "METAMASK WALLET ADDRESS". The form is set against a background with a stylized illustration of a person holding a tablet and a ballot box.

**PERSONAL INFORMATION**

FIRST NAME: FIRST NAME

LAST NAME: LAST NAME

BIRTH DATE: MM-DD-YYYY

GENDER: Male Female

PHONE NUMBER: 1234567890

**METAMASK WALLET ADDRESS**

METAMASK WALLET ADDRESS:

**AADHAR CARD NUMBER**

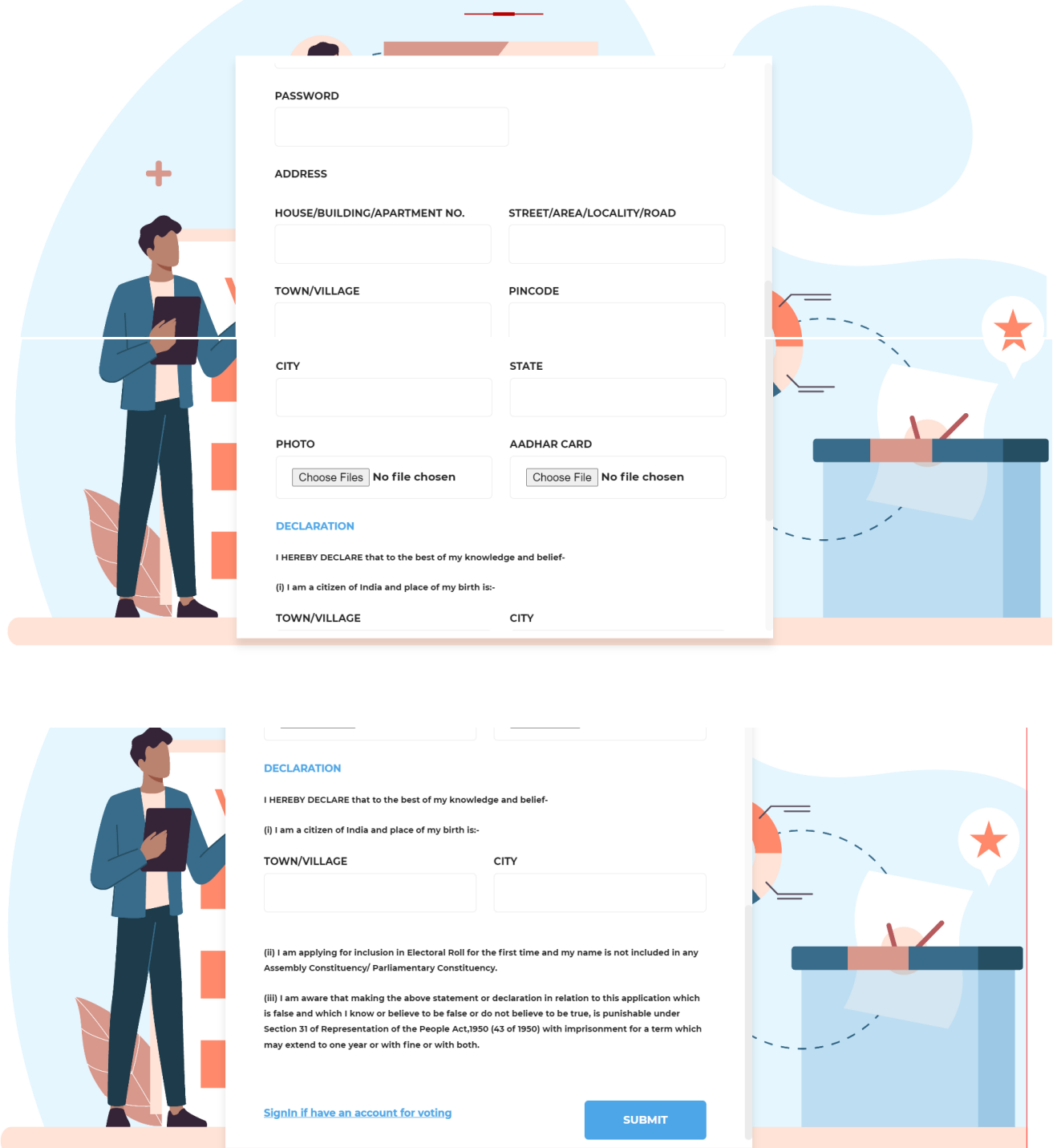
AADHAR CARD NUMBER: 1234-1234-1234

**EMAIL**

EMAIL: EXAMPLE@EMAIL.COM



## REGISTRATION FORM OF E-VOTING



The registration form is presented in two views. The top view shows the initial registration steps, including password creation, address entry, and photo/AADHAR card upload. The bottom view shows the declaration section, where the user confirms their citizenship and provides additional details. The form is set against a background with a man holding a tablet and a ballot box with a star.

**PASSWORD**

**ADDRESS**

**HOUSE/BUILDING/APARTMENT NO.** **STREET/AREA/LOCALITY/ROAD**

**TOWN/VILLAGE** **PINCODE**

**CITY** **STATE**

**PHOTO** **AADHAR CARD**

**DECLARATION**

I HEREBY DECLARE that to the best of my knowledge and belief.

(i) I am a citizen of India and place of my birth is:-

**TOWN/VILLAGE** **CITY**

**DECLARATION**

I HEREBY DECLARE that to the best of my knowledge and belief.

(i) I am a citizen of India and place of my birth is:-

**TOWN/VILLAGE** **CITY**

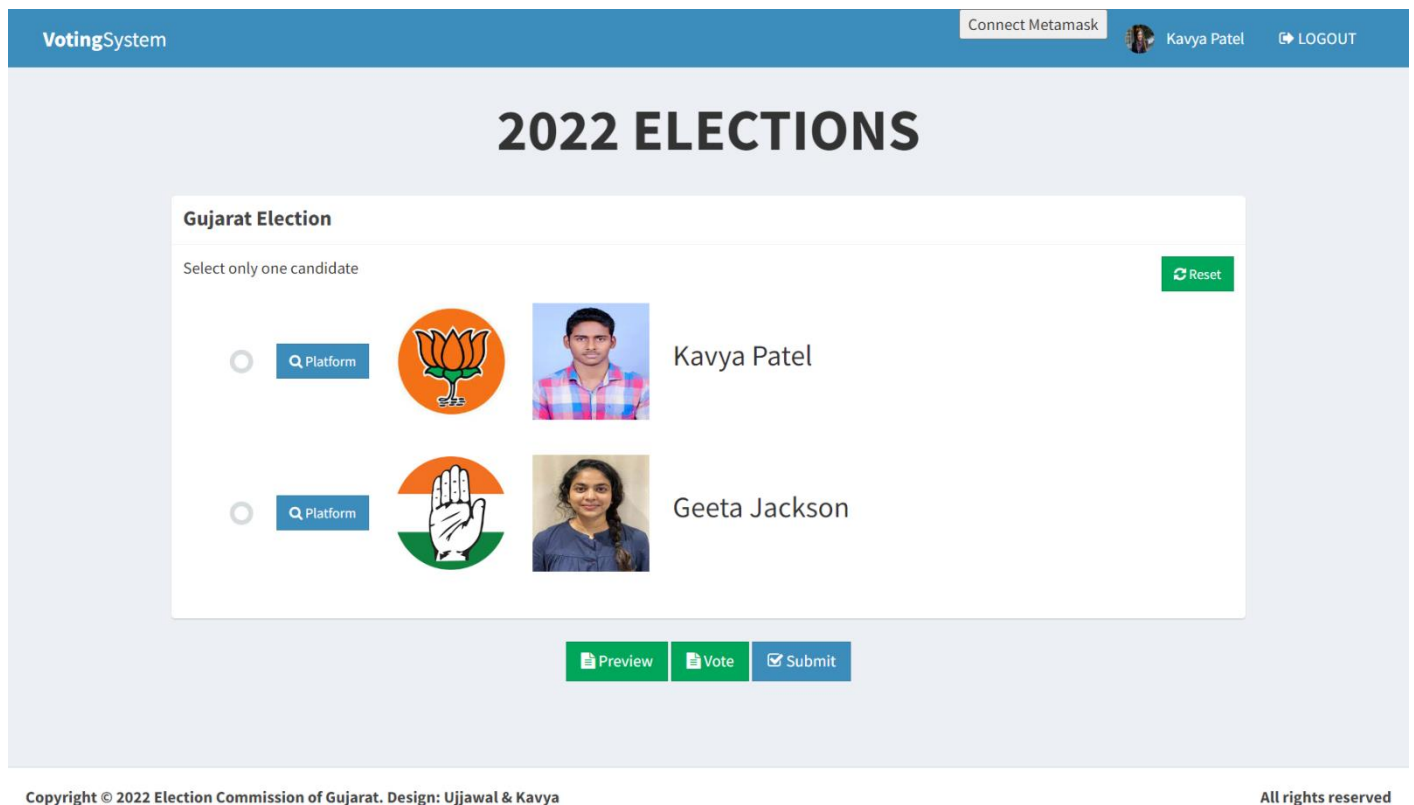
(ii) I am applying for inclusion in Electoral Roll for the first time and my name is not included in any Assembly Constituency/ Parliamentary Constituency.

(iii) I am aware that making the above statement or declaration in relation to this application which is false and which I know or believe to be false or do not believe to be true, is punishable under Section 31 of Representation of the People Act, 1950 (43 of 1950) with imprisonment for a term which may extend to one year or with fine or with both.

[Sign in if have an account for voting](#)

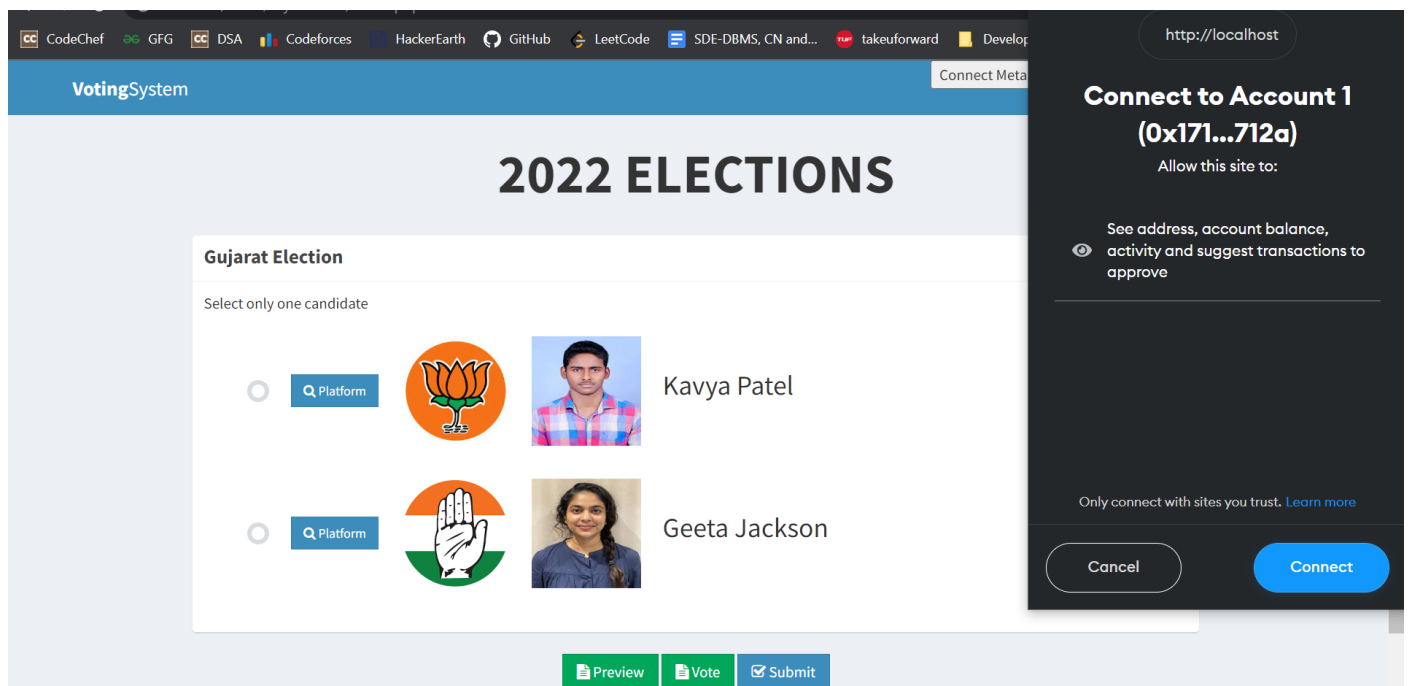
*Fig 8.3 Registration form of voting*

## 8.4 Voter's Dashboard

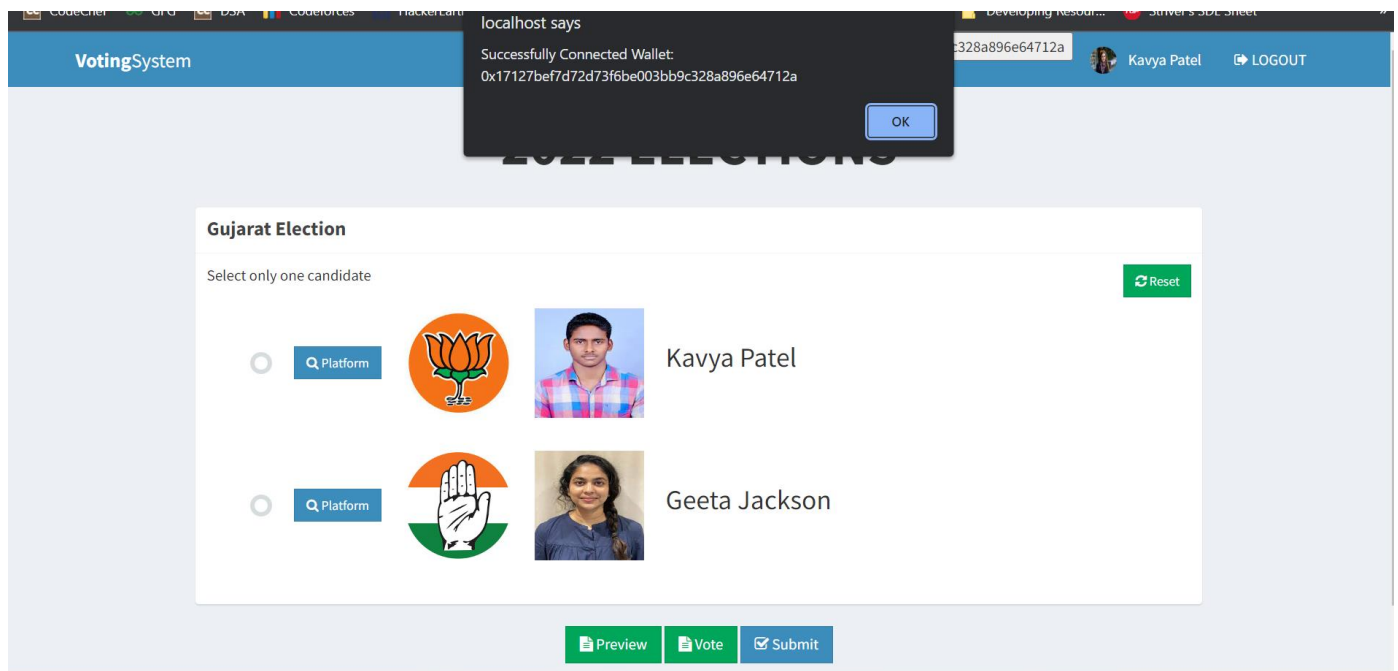


*Fig 8.4 Voter's Dashboard*

## 8.5 Connect Wallet

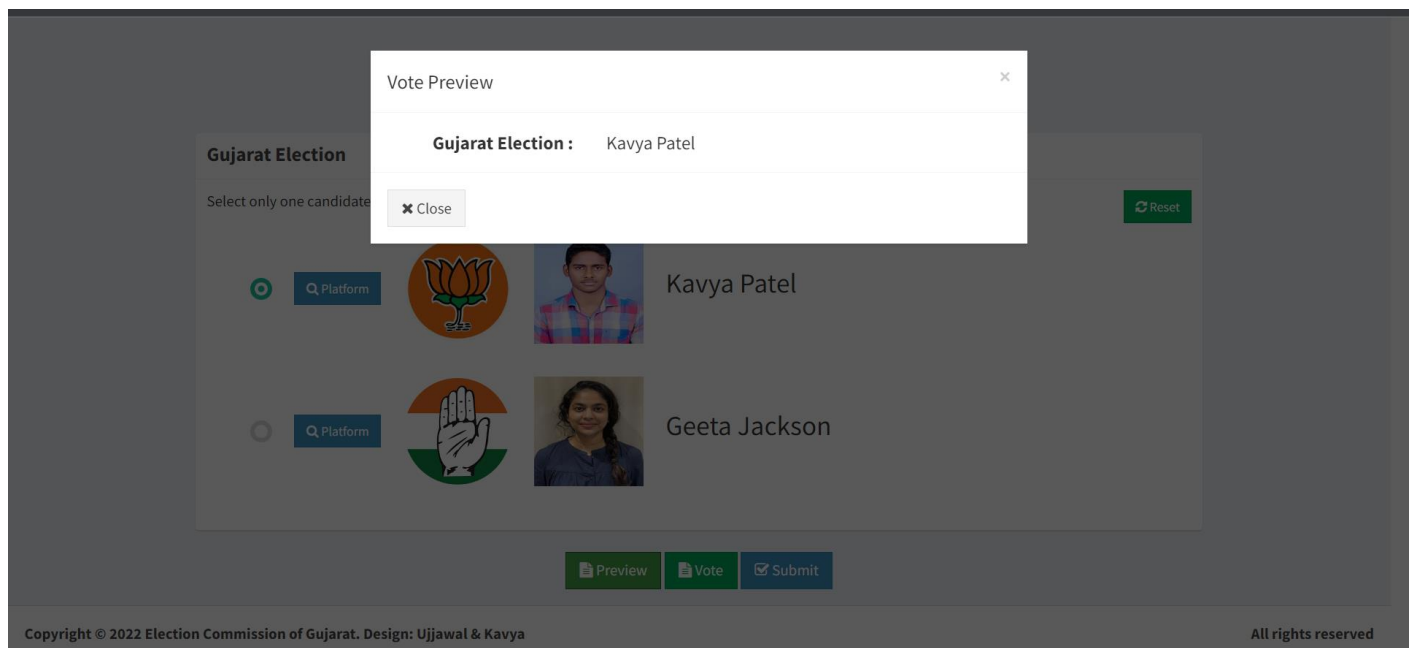






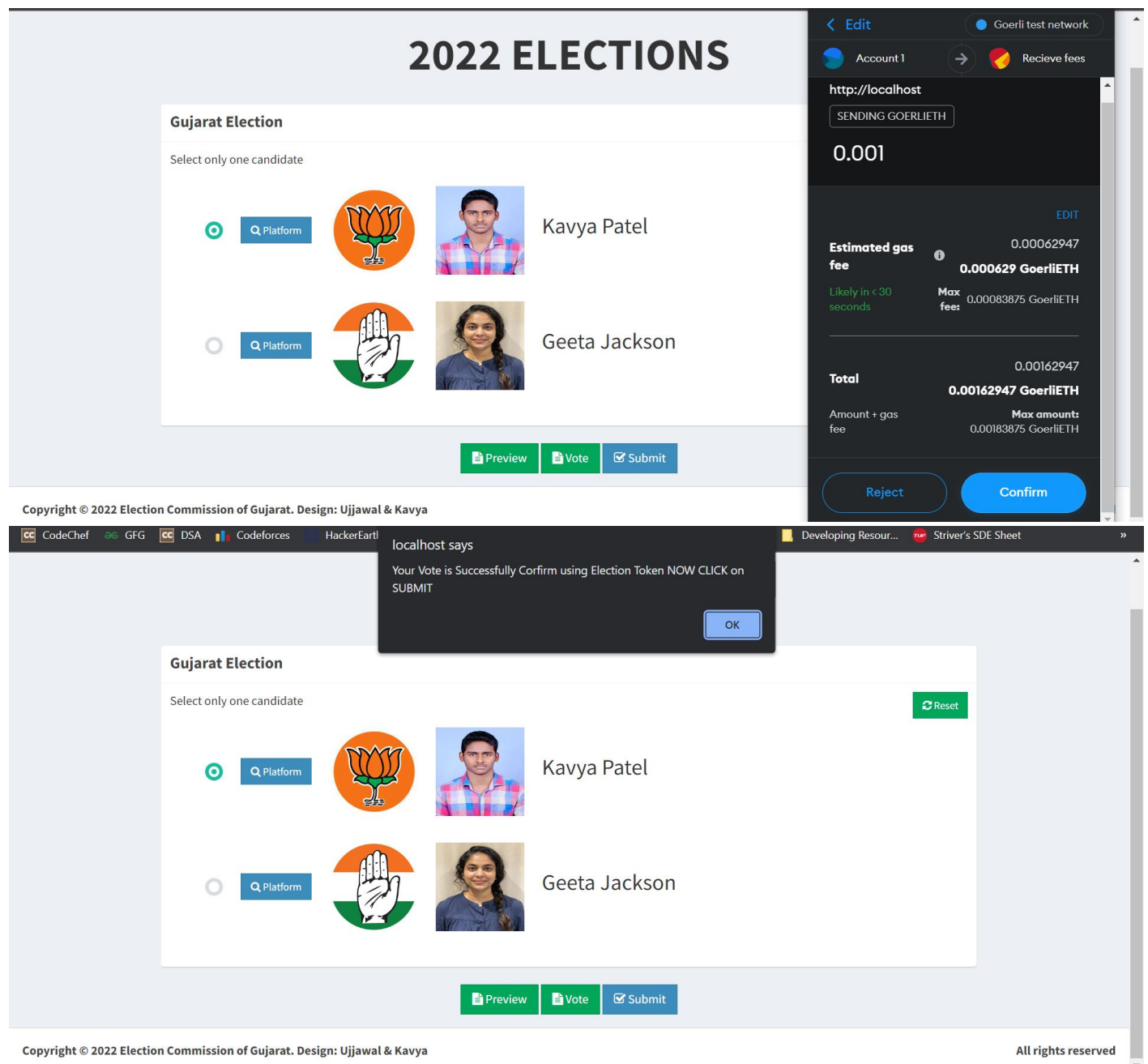
**Fig 8.5 Connect Wallet**

## 8.6 Preview ballot



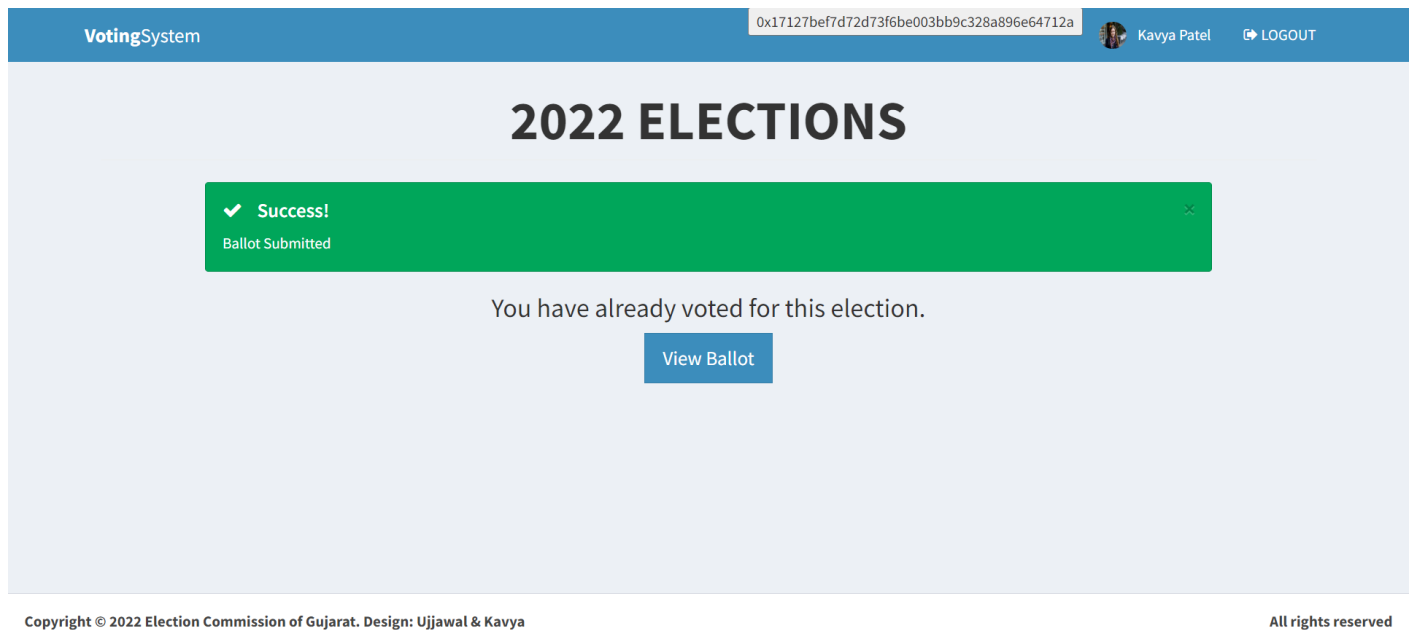
**Fig 8.6 Preview ballot**

## 8.7 Transaction for confirm vote's



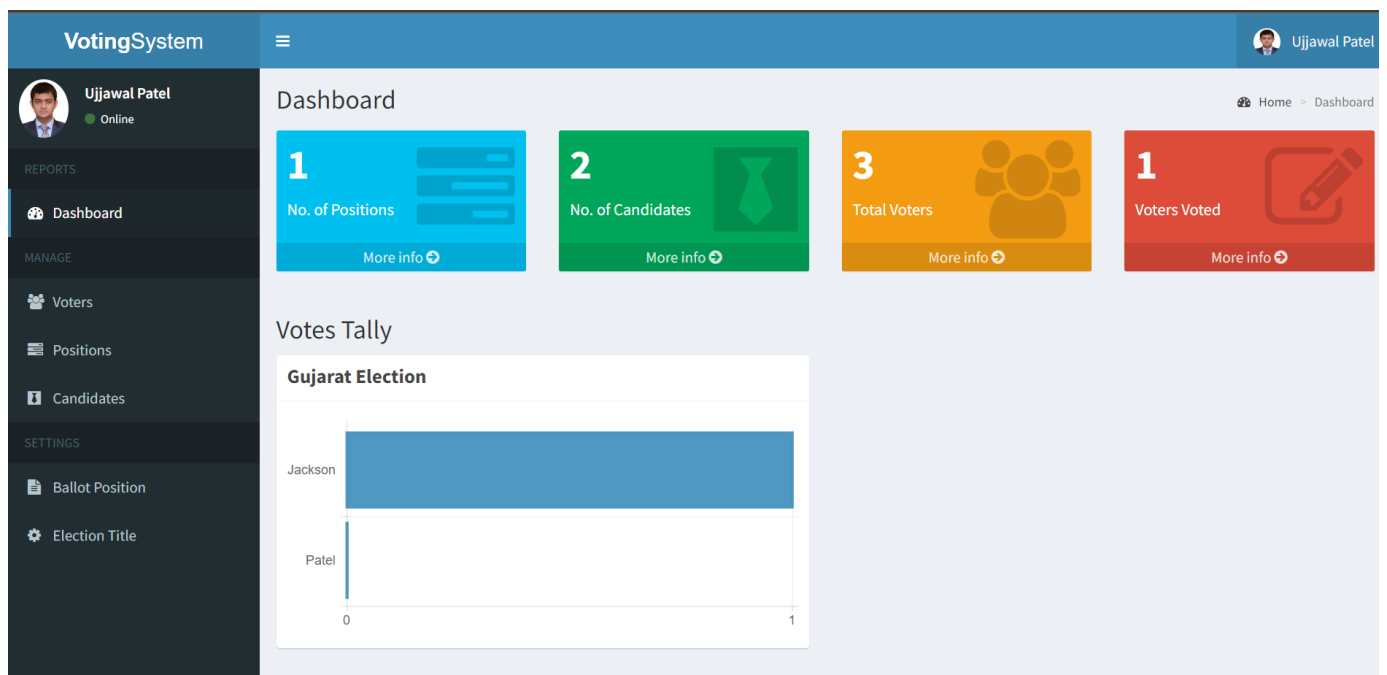
**Fig 8.7 Transaction for confirm vote's**

## 8.8 Successfully Vote Submitted



*Fig 8.8 Successfully Vote Submitted*

## 8.9 Admin Dashboard



*Fig 8.9 Admin Dashboard*

## 8.10 Voter's Verify Dashboard

**VotingSystem** Ujjawal Patel Online

**Voters List** Home > Voters

+ New

Show 10 entries Search:

Lastname	Firstname	BirthDate	Photo	Aadhar Number	MetaMask	Tools
Patel	Kavya	06-01-1964		123412341234	0x3DEbeeff31145aFbC964db690ee4Da4A47F72abe	<a href="#">Edit</a> <a href="#">Delete</a>
Patel	Shaileshkumar	06-01-1964		123412331234	0x3DEbeeff31145aFbC964db690ee4Da4A47F72abe	<a href="#">Edit</a> <a href="#">Delete</a>
Patel	Ujjawal	06-19-2001		123412341256	0x2f4dEf1ea11F8e5039F816106fe7D873e3f39141	<a href="#">Edit</a> <a href="#">Delete</a>

Showing 1 to 3 of 3 entries Previous 1 Next

Copyright © 2022 Election Commission of Gujarat. Design: Ujjawal & Kavya All rights reserved

*Fig 8.10 Voter's Verify Dashboard*

## 8.11 Candidates List

**VotingSystem** Ujjawal Patel Online

**Candidates List** Home > Candidates

+ New

Show 10 entries Search:

Position	Symbol	Photo	Firstname	Lastname	Platform	Tools
Gujarat Election			Kavya	Patel	<a href="#">View</a>	<a href="#">Edit</a> <a href="#">Delete</a>
Gujarat Election			Geeta	Jackson	<a href="#">View</a>	<a href="#">Edit</a> <a href="#">Delete</a>

Showing 1 to 2 of 2 entries Previous 1 Next

Copyright © 2022 Election Commission of Gujarat. Design: Ujjawal & Kavya All rights reserved

*Fig 8.11 Candidates List*

## 8.12 Ballot Position


The screenshot displays the 'VotingSystem' interface. On the left is a dark sidebar with a user profile for 'Ujjawal Patel' (Online) and a menu with sections: REPORTS (Dashboard), MANAGE (Voters, Positions, Candidates), and SETTINGS (Ballot Position, Election Title). The main content area is titled 'Ballot Position' and shows a 'Gujarat Election' form. The form has a header with 'Gujarat Election' and a 'Reset' button. Below the header, it says 'Select only one candidate'. There are two candidate options, each with a radio button, a 'Platform' button, a profile picture, and a name: 'Kavya Patel' and 'Geeta Jackson'. At the bottom of the page, there is a copyright notice: 'Copyright © 2022 Election Commission of Gujarat. Design: Ujjawal & Kavya' and 'All rights reserved'.


**VotingSystem** Ujjawal Patel

Ballot Position Home > Ballot Preview

**Gujarat Election** ↑ ↓

Select only one candidate [Reset](#)

☐ [Platform](#)  Kavya Patel

☐ [Platform](#)  Geeta Jackson

Copyright © 2022 Election Commission of Gujarat. Design: Ujjawal & Kavya All rights reserved

*Fig 8.12 Ballot Position*

## Chapter 9

### Conclusion

#### 9.1 Conclusion

A reliable and truthful voting system is crucial for any democratic society. Democracies depend on trusted elections and citizens should trust the election system for a strong democracy. However traditional paper-based elections do not provide trustworthiness. The idea of adapting digital voting systems to make the public electoral process cheaper, faster and easier, is a compelling one in modern society.

Making the electoral process cheap and quick, normalizes it in the eyes of the voters, removes a certain power barrier between the voter and the elected official and puts a certain amount of pressure on the elected official. It also opens the door for a more direct form of democracy, allowing voters to express their will on individual bills and propositions.

This project has been developed to a blockchain-based electronic voting system that utilizes smart contracts to enable secure and cost-efficient election while guaranteeing voters privacy. It outlines the systems architecture, the design, and a security analysis of the system.

In the next build of this application, it has been proposed to create separate client designs for various roles such as one for election commission and one for candidates registered to a certain party with the existing voting client design. Also, the current versions lack authentication as we do not have access to current Aadhar's or Voter SDK to integrate in our application. Also, it is planned that in the next build notification prompt will be given on the day of voting to all the voters to cast their vote so that the voter turnout is maximum for that election.

#### 9.2 Further Enhancements

In future work, we will investigate the possibility of implementing Paillier cryptosystem as a library in Solidity. With the system we currently have, moving the cryptography to a library in Solidity could largely improve our individual ballot verifiability. Having the Paillier library in Solidity would help us generate new private and public key for each ballot. This will help us achieve individual voter audit on different ballots without compromising the other ballots. To increase user accessibility, we will also look into integrating the Ethereum Light wallet into our system will allow users to unlock their accounts in our UI without needing to run a node or plugin. In continuation of this work, we are focused at improving the resistance of blockchain technology to 'double spending' problem which will translate as 'double voting' for e-voting systems. Although blockchain technology achieves significant success in detection of malleable change in a transaction however successful demonstration of such events have been achieved which motivates us to investigate it further. To this end, we believe an effective model to establish trustworthy provenance for e-voting systems will be crucial to achieve an end-to-end verifiable e-voting scheme. The work to achieve this is underway in the form of an additional provenance layer to aid the existing blockchain based infrastructure.

Finally, to help with voter verification, we will try to integrate an API/process that will allow us to check the validity of all e-mails used to register into our system.

## Chapter 10

### BIBLIOGRAPHY AND REFERENCES

#### 10.1 Book Used

- PHP For Dummies
- Mastering Blockchain By Code Eater (YouTube)
- Connect Wallet API by red Stapler (YouTube)
- HTML & CSS: Design and Build Web Sites
- PHP By Prof. Rachna Modi
- PHP Beginner's Guide By McGrawhill Publication
- Javascript By McGrawhill Publication
- A Beginner's Guide to Building Responsive Layouts with Bootstrap 4

#### 10.2 Reference Used

- [1] Wolchok, Scott, et al. "Security analysis of India's electronic voting machines." Proceedings of the 17th ACM conference on Computer and communications security. ACM, 2010.
- [2] Blockchain-Enabled E-Voting | IEEE Journals & Magazine | IEEE Xplore (Research Paper).
- [3] Ohlin, Jens David. "Did Russian cyber interference in the 2016 election violate international law." Tex. L. Rev. 95 (2016): 1579.
- [4] Ayed, Ahmed Ben. "A conceptual secure blockchain-based electronic voting system." International Journal of Network Security & Its Applications 9.3 (2017): 01-09.
- [5] Hanifatunnisa, Rifa, and Budi Rahardjo. "Blockchain based e-voting recording system design." 2017 11th International Conference on Telecommunication Systems Services and Applications. IEEE, 2017.
- [6] Yu, Bin, et al. "Platform-independent secure blockchain-based voting system." International Conference on Information Security. Springer, Cham, 2018.