ACKNOWLEDGEMENT

First and foremost, With great pleasure, we would like to take this opportunity to express our heartfelt gratitude to our institution, **Chhotubhai Gopalbhai Patel Institute of Technology**, for providing us an opportunity for completing our Project Work successfully.

We are especially thankful to our project guide, **Dr. Rachna Patel, a Professor in the Department of Computer Engineering,** for her invaluable technical suggestions and guidance throughout this project.

We convey our sincere thanks to **Dr. Vibha Patel, Head of the** **Department of Computer Engineering/Artificial Intelligence and Data Science, Uka Tarsadia University,** for providing essential support and timely insights during the progressive reviews.

We extend deep sense of sincere gratitude to **Prof. B.M Vadhar, Director of Chhotubhai Gopalbhai Patel Institute of Technology**, **Uka Tarsadia University,** for providing an opportunity to complete the Project Work.

We would like to acknowledge all the teaching and non-teaching staff members of the Department of Computer Engineering for their assistance in various ways throughout this journey.

Finally, we express our heartfelt thanks to our parents for their unwavering support at every stage of this project. We sincerely appreciate all those who contributed, both directly and indirectly, to the completion of this work.

**Pratham Kalra (202103103510353)**

**Purva Galani (202003103510349)**

**Vinitha Chilkamari (202103103510227)**

ABSTRACT

Electronic voting has emerged over time as a replacement for paper-based and electronic machine voting to reduce redundancies and inconsistencies. The historical perspective presented in the last two decades suggests that it has not been so successful due to the security and privacy flaws observed over time. The Decentralized Voting and Polling System is a blockchain-based application designed to address the growing concerns surrounding the integrity of centralized voting systems in government political elections. With increasing issues of trust due to claims of vote tampering and manipulation, this system provides a transparent, secure, and tamper-proof voting process. It integrates biometric verification through the Aadhaar database, and vote transactions are managed via Azure Blockchain services. This system allows voters to cast their votes from anywhere in the country by visiting government-authorized centers like banks. By leveraging blockchain, votes are made public and verifiable without compromising the security of the system. Smart contracts, developed using Solidity, ensure the security and immutability of each vote transaction. With a user-friendly interface, built using React and Tailwind CSS, the system is lightweight and accessible, even in areas with slow internet connectivity.

TABLE OF CONTENTS

[ACKNOWLEDGEMENT iii](#_Toc497505263)

[ABSTRACT iv](#_Toc497505264)

[LIST OF FIGURES vii](#_Toc497505265)

[LIST OF TABLES viii](#_Toc497505266)

[Chapter 1 Introduction 1](#_Toc497505267)

[1.1 Background 1](#_Toc497505268)

[1.2 Problem Definition 2](#_Toc497505269)

[1.3 Motivation 2](#_Toc497505270)

[1.4 Objective 2](#_Toc497505271)

[1.5 Scope 3](#_Toc497505272)

[1.6 Applications 3](#_Toc497505273)

[Chapter 2 System Planning 4](#_Toc497505274)

[2.1 Project Development Approach 4](#_Toc497505275)

[2.2 System Modules 4](#_Toc497505276)

[2.2.1 Module 1 Candidate Module 4](#_Toc497505277)

[2.2.2 Module 2 Voter Module 5](#_Toc497505278)

[2.2.3 Module 3 Graph Module 6](#_Toc497505278)

[2.3 Functional Requirements 6](#_Toc497505279)

[2.4 Non Functional Requirements 7](#_Toc497505280)

[2.5 Hardware and Software Requirements 8](#_Toc497505281)

[2.6 Timeline Chart 9](#_Toc497505282)

[Chapter 3 System Design 1](#_Toc497505283)0

[3.1 Database Schema 1](#_Toc497505284)0

[3.2 ER Diagram 1](#_Toc497505285)0

[3.3 Use Case Diagram 1](#_Toc497505286)1

[3.4 Sequence Diagram 1](#_Toc497505287)2

[3.5 Activity Diagram 1](#_Toc497505288)4

[3.6 Class Diagram 1](#_Toc497505289)5

[3.7 Data Flow Diagram 1](#_Toc497505290)6

[Chapter 4 Implementation and Testing 1](#_Toc497505291)8

[4.1 Snapshots 1](#_Toc497505292)8

[4.2 Test Cases 2](#_Toc497505293)6

[Conclusion and Future Scope 2](#_Toc497505294)8

[References 2](#_Toc497505297)9

# LIST OF FIGURES

[Figure 2.1: Timeline Chart 9](#_Toc497505240)

[Figure 3.1: ER Diagram 1](#_Toc497505241)0

[Figure 3.2: Use Case Diagram 1](#_Toc497505240)1

[Figure 3.3: Sequenece Diagram for Registration 1](#_Toc497505240)2

[Figure 3.4: Sequenece Diagram for Login 1](#_Toc497505240)3

[Figure 3.5: Sequenece Diagram for Vote 1](#_Toc497505240)3

[Figure 3.6: Activity Diagram 1](#_Toc497505240)4

[Figure 3.7: Class Diagram 1](#_Toc497505240)5

[Figure 3.8: Level-0 Data Flow Diagram 1](#_Toc497505240)6

[Figure 3.9: Level-1 Data Flow Diagram 1](#_Toc497505240)7

[Figure 4.1: Voter register Page 1](#_Toc497505240)8

[Figure 4.2: Candidate register Page 1](#_Toc497505240)8

[Figure 4.3: Login Page 1](#_Toc497505240)9

[Figure 4.4: Home Page 1](#_Toc497505240)9

[Figure 4.5: Vote Casting Page 2](#_Toc497505240)0

[Figure 4.6: Result Page 2](#_Toc497505240)0

[Figure 4.7: List Of Transaction 2](#_Toc497505240)1

[Figure 4.8: List Of Account 2](#_Toc497505240)1

[Figure 4.9: Acess Metamask 2](#_Toc497505240)2

[Figure 4.10: Connecting with metamask 2](#_Toc497505240)2

[Figure 4.11: Connecting with metamask done 2](#_Toc497505240)3

[Figure 4.12: Account gets updated 2](#_Toc497505240)3

[Figure 4.13: Deployment of contract in blockchain 2](#_Toc497505240)4

[Figure 4.14: Option to add candidates and voter by owner 2](#_Toc497505240)4

[Figure 4.15: Show all candidates and the vote option 2](#_Toc497505240)5

[Figure 4.16: List of transactions in blockchain 2](#_Toc497505240)5

# LIST OF TABLES

[Table 2.1: Functional Requirements Table 6](#_Toc84423129)

[Table 2.2: Non Functional Requirements Table 7](#_Toc84423130)

[Table 4.1: Test Cases 2](#_Toc84423130)6

# Chapter 1 Introduction

## Background

The concept of "the will of the people" is crucial for representing opinions in electoral systems, from college unions to parliaments. Voting has become a key way for people to express their choices, helping to build trust in the selection process through majority rule. This has strengthened democracy and the value of voting for electing governments.As people grow more confident in democratic systems, it’s essential that they maintain their trust in voting. A reliable voting system allows citizens to choose their representatives, who then address important issues like national security, health and education policies, international relations, and taxation. This trust in representation gives people confidence that their government will work for their benefit.

In order to make the voting process more effective the institutions like 'Election Commission' came into existence in different parliamentary democracies. The institutions, along with setting up the process and legislation for conducting the elections, formed the voting districts, electoral process, and the balloting systems to help in conduct of transparent, free, and fair elections. The concept of secret voting was introduced since the beginning of the voting system. Since the trust on democratic systems is increasing it is important to uphold that the trust on voting should not decrease. In the recent past there have been several examples where it was noted that the voting process was not completely hygienic and faced several issues including transparency and fairness, and the will of people was not observed to be effectively quantified and translated in terms of formation of the governments.

People recently trust a lot on the decision on voting, although a few trust issues have been increasing day by day where people complain about such centralized voting machines being tampered with , or the centralized database being tempered.In such cases, the trust on the process is necessary even though such cases may or may not be reported, Trust is very necessary in the process.To resolve issues, technology comes in existence where people can cast vote from where ever they are and securely cast their vote without any influence from the external sources. By taking this present issue of voting, this project is to implement secure, efficient and burden-free voting using DecentralizedVoting and Polling system in blockchain technology.

## Problem Definition

As India is the most Populated country in the world with 63.3% Adults who have the right to case a vote, but only 38.43% out of this 63% people case votes. Among all the other reasons for not participating in voting campaign, the most important is the traditional voting method is not convenient , like transport being a major reason. Aside from that, people loosing the trust in election method and politics as well.

In Such cases Decentralized voting system can help being digitization to the people not being able to vote just due to distance and travel.Upholding the trust of people will be easier and feasible in case of Decentralized system as we move more and more towards digitization.

## Motivation

The main motivation behind this project is the declining voter turnout and the loss of trust in traditional, centralized voting methods. In a country as populous as India, logistical challenges such as travel and waiting times discourage voter participation. Furthermore, concerns over vote tampering and manipulation in centralized databases call for a more transparent and secure system. A decentralized system based on blockchain technology ensures that votes are immutable, verifiable by the public, and immune to tampering, ultimately restoring trust in the election process. This system also aims to be scalable and efficient, handling large voter databases in a secure and seamless manner.

## Objective

* To provide security and trusting voting system using three step authentication and other algorithms to ensure safety and security
* To store the votes in blockchain to eliminate any chances of tempering while the voters detail should be stored in centralized database to reduce the cost of making and storing unnecessary data in blockchain nodes.
* To generate a User Interfaces over the internet using React and Tailwind for simpleruser interfaces which should be responsive and easy to use while casting the votes.
* Smart Contracts using Solidity to ensure proper vote casting and transaction.

## Scope

The decentralized voting system is designed to replace traditional voting methods in national and state political elections. Its primary scope is for government elections, ensuring transparency, security, and trust among voters. This system allows voters to cast their ballots securely from authorized centers nationwide, providing a convenient solution for individuals who face logistical barriers, such as travel and long wait times. With the potential to scale, the system could be adapted for secure voting in local government bodies as well.

## Applications

* The system can be employed in large-scale political elections, ensuring decentralized, secure, and transparent vote casting.
* This system enables people in remote areas or citizens living abroad to participate in elections by casting their votes from authorized centers.
* The system is also applicable in government-led referendums, ensuring that votes are secure and verifiable, preventing any tampering.
* By decentralizing the voting process and reducing dependence on traditional infrastructure, the system can lower election-related costs while increasing efficiency.

# Chapter 2 System Planning

## Project Development Approach

We chose **RAD (Rapid Application Development**) as our Software Development Model

**Advantages :**

* Given the time frame of 2 months we would need the rapid application within the given time frame and RAD fits best to our scenario.
* Collaborative approach with one time release fits the Rapid Application Development Model

**Disadvantage :**

* Repetitive bug fixes and Updates aren't possible in this Model as it focuses on Rapid development of an application and does not cover the Updates part.

## System Modules

In our project, we focused on creating a Decentralized Voting System made up of three main modules: Candidate, Voter, and Graph. These modules work in a hierarchical structure to ensure efficient data management and smooth user interactions.

### 2.2.1 Module 1 Candidate Module

The Leadership On boarding module manages the registration process of in dividuals who wish to contest in the election. It ensures that only verified and credible candidates are allowed to participate, maintaining the integrity of the political system.

**Page 1: Candidate Gateway**

**Description:** The Candidate Gateway is the entry point for aspiring leaders. Candidates submit their personal and political credentials, such as their name, political party affiliation, manifesto, and other required documents. This page ensures that all candidate information is stored securely on the blockchain, guaranteeing that no tampering or unauthorized changes can occur.

**Elaboration:** Once the registration is completed, the system assigns a unique candidate identifier that links the candidate’s details to the blockchain ledger. The registration process is transparent, and it includes checks to verify that each candidate meets the eligibility criteria for participation in the election. By using blockchain technology, the Candidate Gateway ensures transparency, trustworthiness, and a secure digital record of all participants.

**Page 2: Candidate Dashboard**

**Description:** The Candidate Dashboard is a personalized interface where registered candidates can monitor their registration status, update their campaign details, and check the status of their candidacy. It also displays key election updates, such as timelines and milestones.

**Elaboration:** This page is where candidates can manage and view their submitted information and track their standing in the election. Security is paramount, and the blockchain ensures that all data modifications, if allowed, are tracked and verified. This dashboard gives candidates a transparent view of their election journey, from registration to the final voting results.

### 2.2.2 Module 2 Voter Module

The Citizen’s Voice module empowers voters by enabling them to register and cast their vote securely. This module protects voter identities and prevents fraud, ensuring that each citizen has the opportunity to contribute to the democratic process in a secure, fair, and transparent way.

**Page 1: Voter Enrollment Portal**

**Description:** The Voter Enrollment Portal is a secure way for eligible voters to register for elections. Voters must submit identity verification details, which are safely stored on the blockchain to ensure only legitimate voters can participate.

**Elaboration:** The portal uses advanced blockchain technology to prevent duplicate registrations and fraud. Once registered, each voter gets a unique ID that connects them to the voting system while keeping their information private and secure. The registration process is straightforward but highly secure, protecting the election's integrity by ensuring that only registered and eligible voters can vote.

**Page 2: Vote Vault**

**Description:** The Vote Vault is a secure place for registered voters to cast their ballots. After logging in, they can choose candidates and submit their votes, which are permanently recorded on the blockchain.

**Elaboration:** Each vote is stored on the blockchain for accurate, tamper-proof counting. Once a vote is cast, the system locks access to prevent double voting. Votes are anonymous, secure, and irreversible, and voters receive confirmation that their vote has been recorded.

### 2.2.3 Module 2 Graph Module

The Victory Insights module provides a visual, transparent view of the election results in real time. It showcases which candidate is leading or has won the election, offering full transparency and confidence in the democratic process.

**Page 1: Election Pulse**

**Description:** The Election Pulse page shows real-time election results with clear graphs of each candidate's vote count. It provides voters, candidates, and officials with an overview of the election's progress and current leaders.

**Elaboration:** As votes are cast, the page updates automatically by pulling data from the blockchain, ensuring results are secure and tamper-proof. It uses charts for easy understanding and presents the final tally in a transparent, reliable format, giving everyone confidence in the election outcome.

## Functional Requirements

Table 2.1: Functional Requirements Table

|  |  |  |
| --- | --- | --- |
| **FRN** | **DESCRIPTION** | **COMMENT** |
| **FR1** | * Voters initial details have to registered using Aadhaar. * Voter can Register only if his/her Age is eligible. | Register |
| **FRN2** | Voters must be authenticated through secure methods, such as digital identity verification or biometric authentication. | Login |
| **FRN3** | * Allow voters to cast their votes securely and privately. * once a vote is cast, it cannot be altered or tampered with. | Casting Vote |
| **FRN4** | Confirm that votes have been successfully submitted and recorded on the blockchain. | Vote Confirmation |
| **FRN5** | The application shows realtime results of polling with show vote count option. | Result |
| **FRN6** | Allow for reporting and handling of errors or discrepancies in the voting process. | Error Handling |
| **FRN7** | Ensure the system has redundancy and fault tolerance to handle hardware or software failures without data loss. | Fault Tolerance |
| **FRN8** | Provide mechanisms for data backup and recovery to protect against accidental or malicious data loss. | Recovery |

## Non Functional Requirements

Table 2.2: Non Functional Requirements Table

|  |  |  |
| --- | --- | --- |
| **RN** | **DESCRIPTION** | **COMMENT** |
| **NF1** | The system must handle a high volume of votes and transactions efficiently, scaling as needed for large elections. | Scalability |
| **NF2** | * The project should be very lightweight and should work even in the very slow internet conditions. * Ensure low latency for vote submission and result retrieval to provide timely outcomes. | Performance |
| **NF3** | * The details of voter must be encrypted using smart contract and blocks. * Once a vote is cast, it must be recorded immutably on the blockchain to prevent tampering. | Security |
| **NF4** | * Protect voter anonymity while ensuring votes are accurately recorded and counted. * Ensure that vote details are kept confidential and accessible only to authorized entities. | Privacy |
| **NF5** | Design the system to be intuitive and easy to use for all voters, including those with limited technical knowledge. | Usability |
| **NF6** | System can be used for all the operating system which support Nodejs and Single page applications supported browser. | Portability |
| **NF7** | No unnecessary data should be stored in blocks. Only necessary data should be stored in blocks, rest of the data should be stored in mongodb to reduce the storage costs. | Low cost |
| **NF8** | The system can integrate with other systems or databases as needed for voter verification or result reporting. | Interoperability |

## Hardware and Software Requirements

**Hardware Requirements :**

Processor : Intel i5 / Ryzen 5

RAM : 4GB or Higher

Hard Disk : 20GB

**Software Requirements :**

Operating System : Windows/Linux

Back-end : Nodejs

Database : MongoDB

Frameworks : Blockchain and Smart Contracts

Front-End : React , Tailwind

## Timeline Chart

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ACTIVITY** | **22 AUG - 25 AUG** | | **26 AUG- 15 SEP** | **16 SEP-30 SEP** | **1 OCT-20OCT** |
| **Requirement Analysis** |  |  |  |  |  |
| **Designing** |  |  |  |  |  |
| **Implementation** |  |  |  |  |  |
| **Testing** |  |  |  |  |  |
| **Report Generation** |  |  |  |  |  |

Figure 2.1:Timeline Chart

# Chapter 3 System Design

## Database Schema

Blockchain does not store any of its information in a central location. Instead, the blockchain is copied and spread across a network of computers. Whenever a new block is added to the blockchain, every computer on the network updates its blockchain to reflect the change.**Blocks**: Each block stores transection, and when a block is full, a new block is created and linked to the previous one.

## ER Diagram

Figure (3.1) presents an entity-relationship diagram (ER diagram) that outlines the main entities in our project along with their attributes and relationships. The diagram features three primary entities: **User**, **Candidate**, and **Election**. The **User** entity includes attributes such as Aadhaar number (for verification), name, date of birth, age (derived from the date of birth), and a status indicating whether the voter has already voted. The **Candidate** entity encompasses attributes like candidate ID, name, political party, and the number of votes received. Lastly, the **Election** entity consists of attributes including election ID, start date, end date, and start and end times. The relationships illustrated in the diagram show that users cast votes in elections, while candidates participate in those elections.

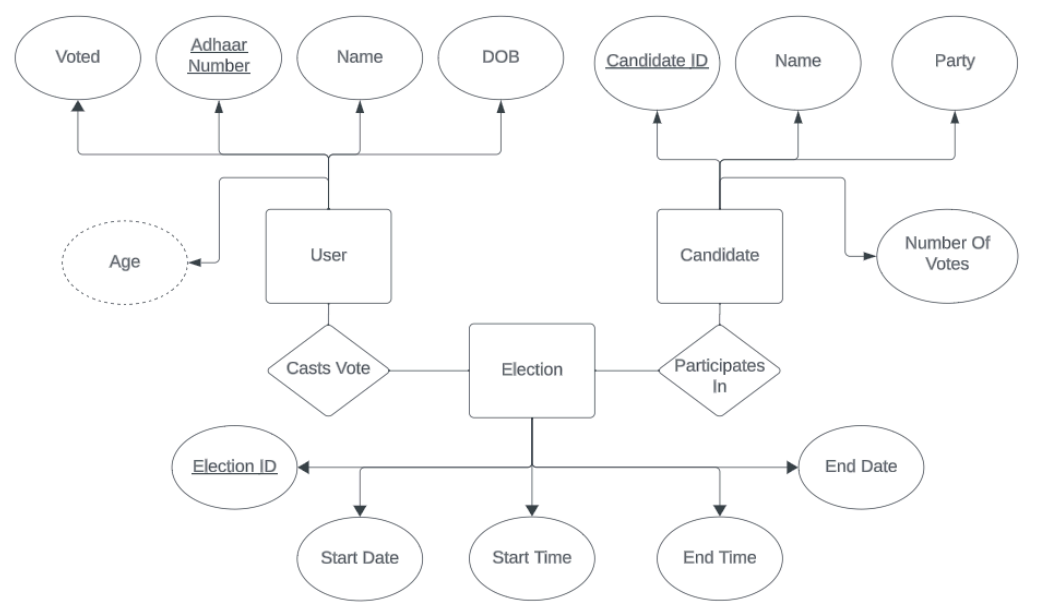


Figure 3.1: ER Diagram

## Use Case Diagram

Use case consists of user and admin, where user is allowed to login and caste his/her vote. and admin manages all the required processing like, managing candidate details, and authorizing the right voter to cast his/her vote, after the verification of the user and after the election poll ends, the admin is responsible for declaration of result in a graphical representation.

The figure (3.2) is the use case diagram which shows how the user and server manage their processes. The processes are stated as follows. It consists of user and admin registration, candidate management, Voter management, casting of vote and declaration of result. All of these processes are taken care of by the server and one of the users either the voter or the admin.

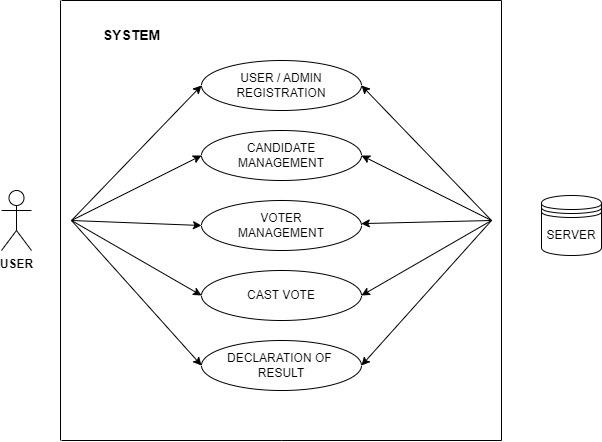


Figure 3.2: use case Diagram

## Sequence Diagram

Three diagrams are presented in the following to demonstrate the sequence of operations of a voter for authentication and vote casting. The sequence diagram presented in Figure (3.3) shows the sequence of operations for a voter to be registered with authorities. The voter requests register as user and activate their account.

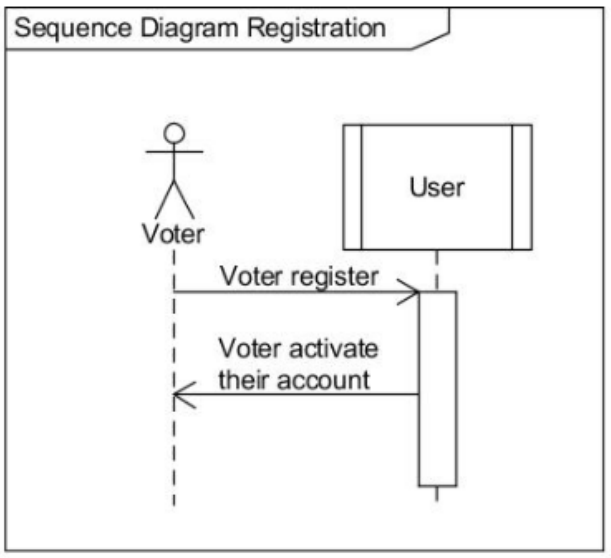


Figure 3.3: Sequence Diagram for Registration

Here we can see that voter will need to register to have an account that will be used for voting. This is handled by the User class. Then they will need to activate their account from their email address. Once that is finished, voter can login to their account, which is displayed in the sequence diagram in Figure (3.4).

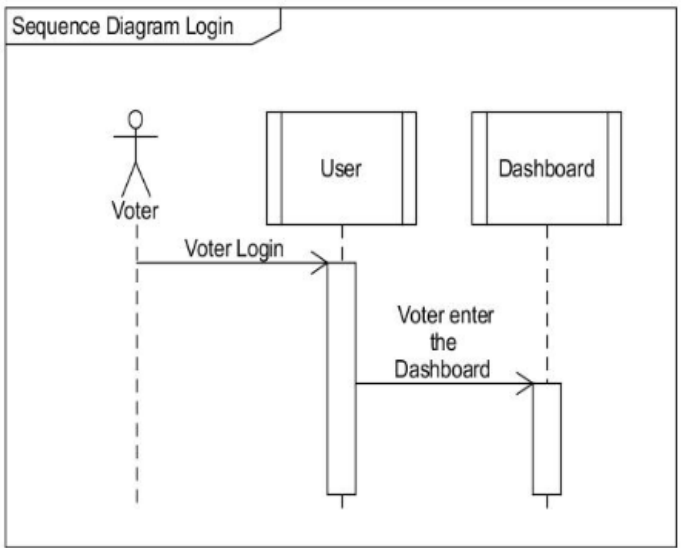


Figure 3.4: Sequence Diagram Login

From the diagram in Figure (3.4), we can see that voter will login which will be handled by the User class. Then the voter can access the dashboard to vote on their chosen candidate which will be handled by the Dashboard class. Then voter can choose their chosen candidate to vote for which will be handled by the Dashboard class. The Voting class, which is the smart contract that we have built, will store the vote for the chosen candidate on the blockchain.

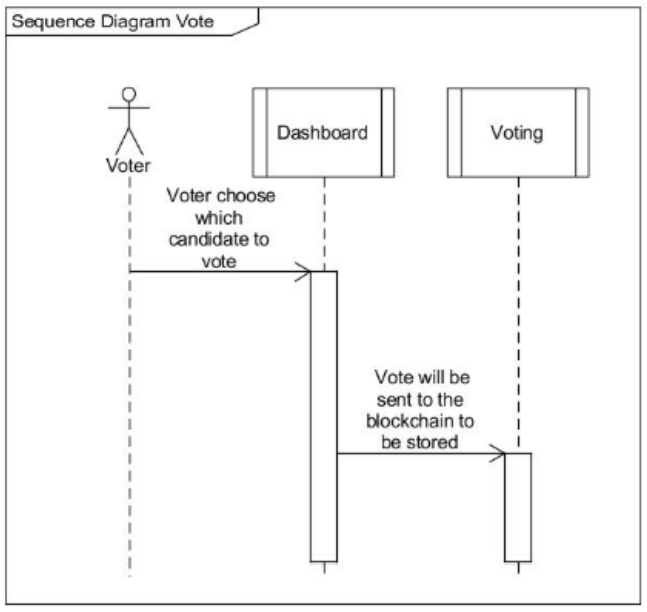


Figure 3.5: Sequence Diagram for Vote

## Activity Diagram

The voting process is briefly presented in Figure (3.6). UML activity diagrams were used to represent the flow of actions for the voting process from a starting point to a finishing point, detailing the decision paths that may occur in an activity. These diagrams illustrate alternative and concurrent action flows and separate responsibilities by using swim lanes.

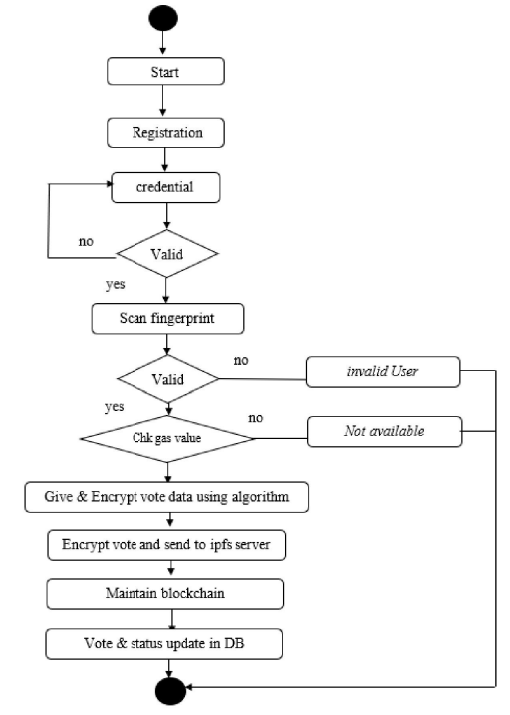


Figure 3.6: Activity Diagram for Voting process

## Class Diagram

There will be two types of users, voters and admin. Voters will be able to check who they want to vote, their vision and mission, and finally to vote for their chosen candidate. Admin will be able to open and close the period of voting, register candidates, and reset the result of past voting. The system itself is divided into numerous class that works on different parts of the system, which could be seen on the diagram in Figure (3.7). We can see from the diagram at Figure (3.7) that the system has been divided into 6 classes. This classes has their own method that ensure that the system will be usable by it's users.

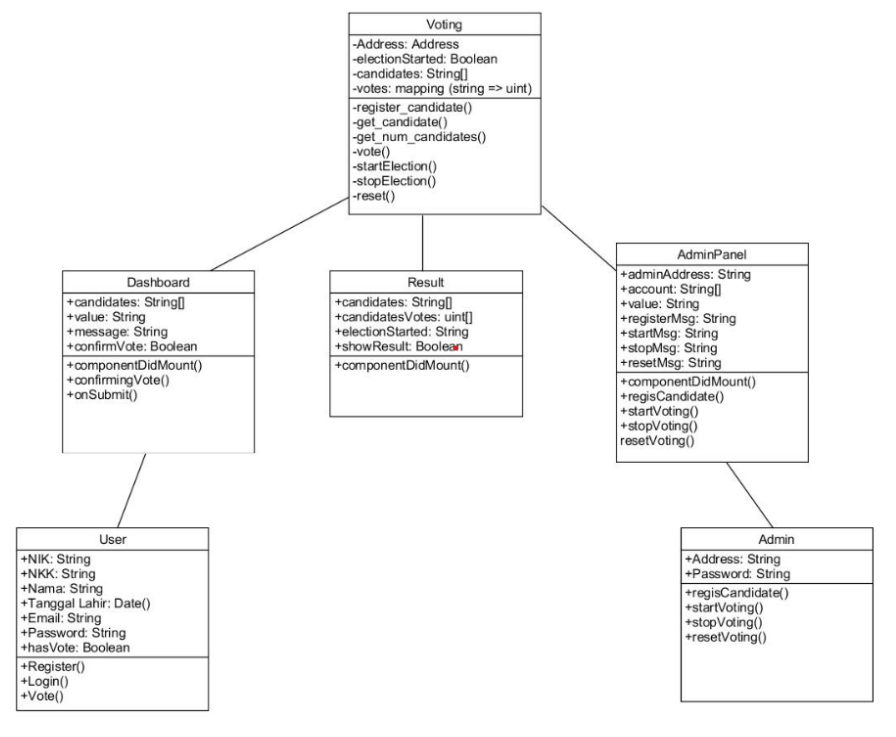


Figure 3.7: Class Diagram

## Data Flow Diagram

The diagram (3.8) describes the detailed view on the application process. It mainly shows the connection between User side architecture and the network side architecture, where the user will login using username and password. After successfully logging in and verifying identity users will vote for a candidate from the list of provided candidates. After voting the votes are given to the vote casting system in this case it is their solidity smart contract which will then hash the votes according to SHA256 algorithm and send it to the network of nodes which will then re-verify these votes and store it in their local blockchain. These votes are then sent back to the smart contracts when the command is given like for counting and showing results. The smart contact will gather all this information and will display it to the voter once the election is over. This describes our Level 0 Data flow diagram.

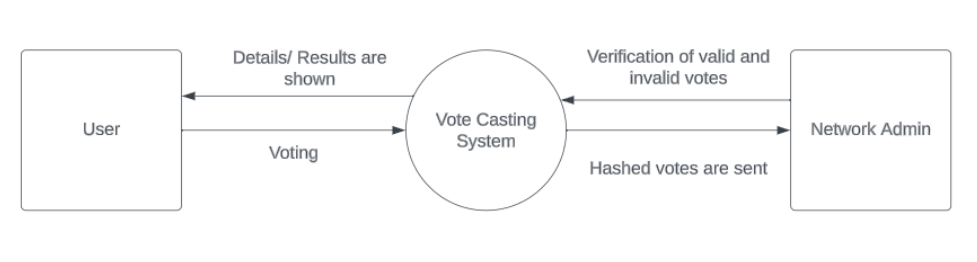


Figure 3.8: Level-0 Data Flow Diagram

The next diagram (3.9) is the level 1 data flow diagram and it explains the working of our project and the data flow in a more detailed manner than the above. The level 1 data flow diagram shows that when the user will login to the website, he will face an authentication process which will take place against the registered set of databases. The authentication process will take place according to an Aadhaar database system. The user will be needed to enter the details and they will be verified automatically through the verification system. Next when the user casts their vote, the vote will directly go through a smart contract system written in solidity language. The smart contract will verify if the voter is verified or not, if the voter is voting for a registered candidate or not, if the voter is above the age of 18 or not, and it will also check if or whether the voter has already voted or not, because no voter will be allowed to vote twice. After the smart contracts, the votes will be encrypted using a specific asymmetric key pair and will be digitally signed for the user who has voted and will then go to a network of nodes for the consensus process. In this process, the nodes which are computers basically will fight for who will add to note to the Blockchain. They will also verify if the vote has come from the source specified in the digital signature or not, and they will also verify the contents of the vote so that no tampering can occur. Other than this the smart contract will also be responsible for starting and ending elections and counting of votes.

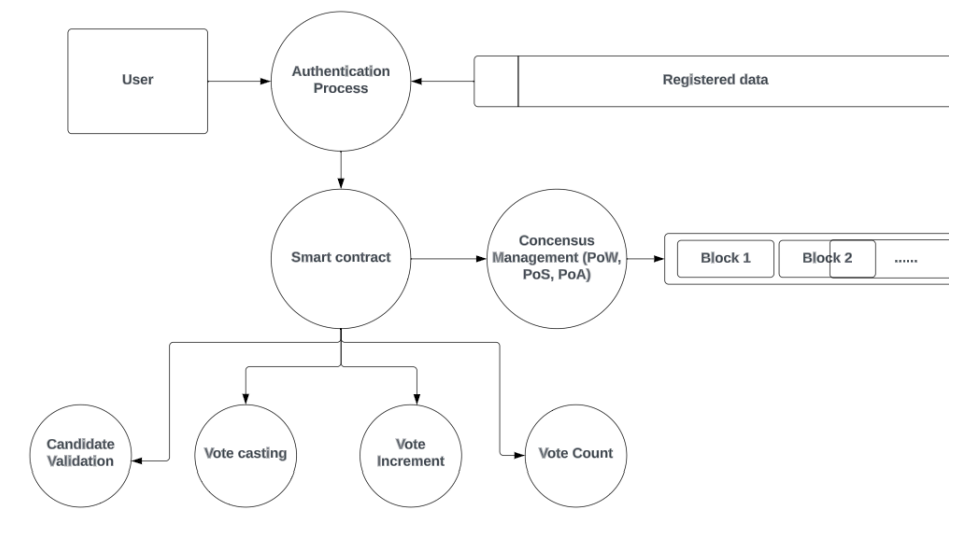


Figure 3.9: Level-1 Data Flow Diagram

# Chapter 4 Implementation and Testing

## Snapshots

* **Voter Register Page**

First step is to create an account by entering your details like name, date of birth, Aadhaar number and a voter ID.

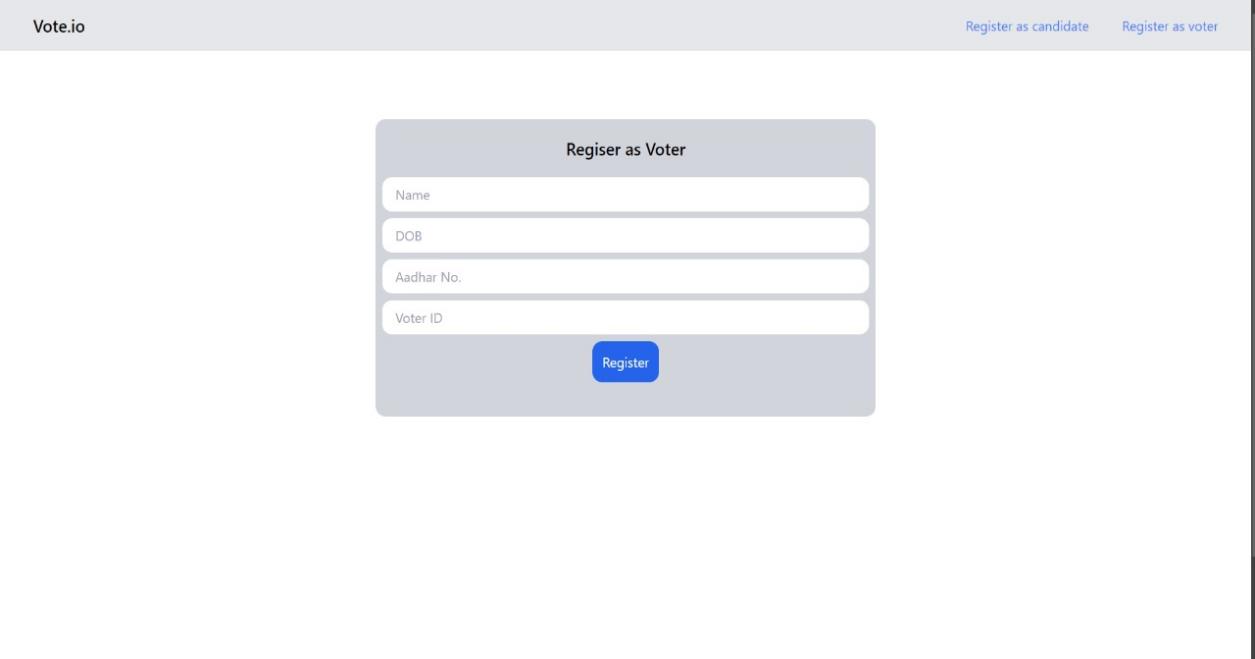
****

Figure 4.1: Voter register Page

* **Candidate Register Page**

Candidates can register for participation in election with information like candidate name, DOB, Aadhar no., Voter ID, Designation and seat of.

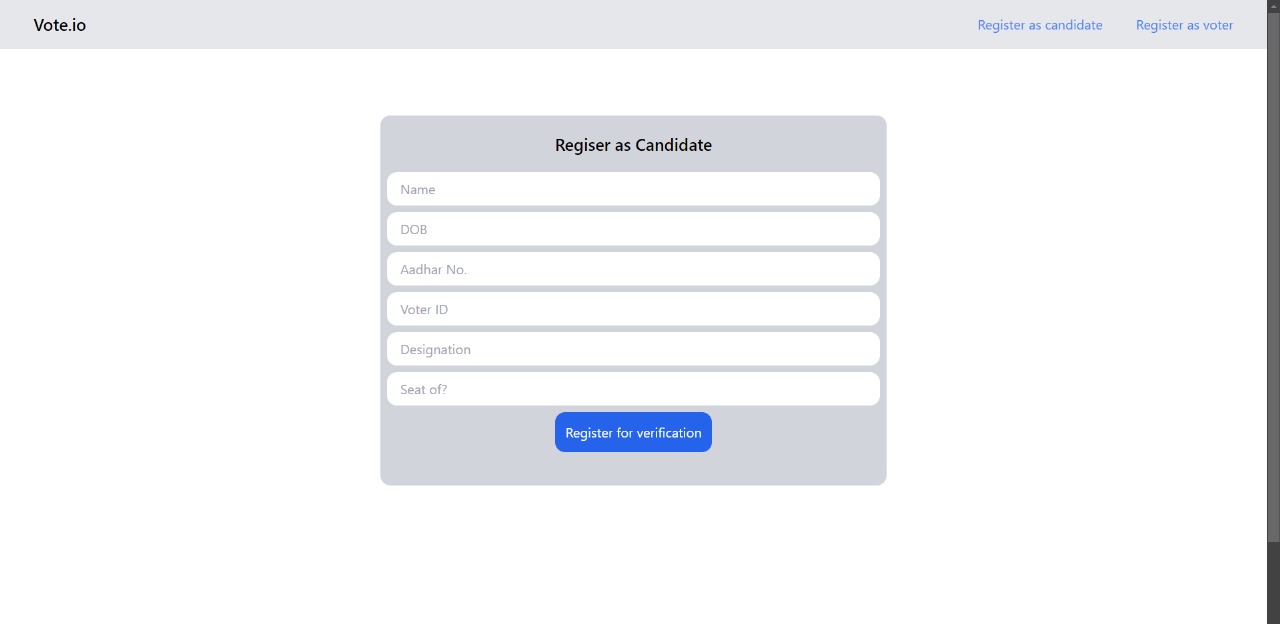
****

Figure 4.2: Candidate register Page

* **Login Page**

Voter can login the website using metamask.

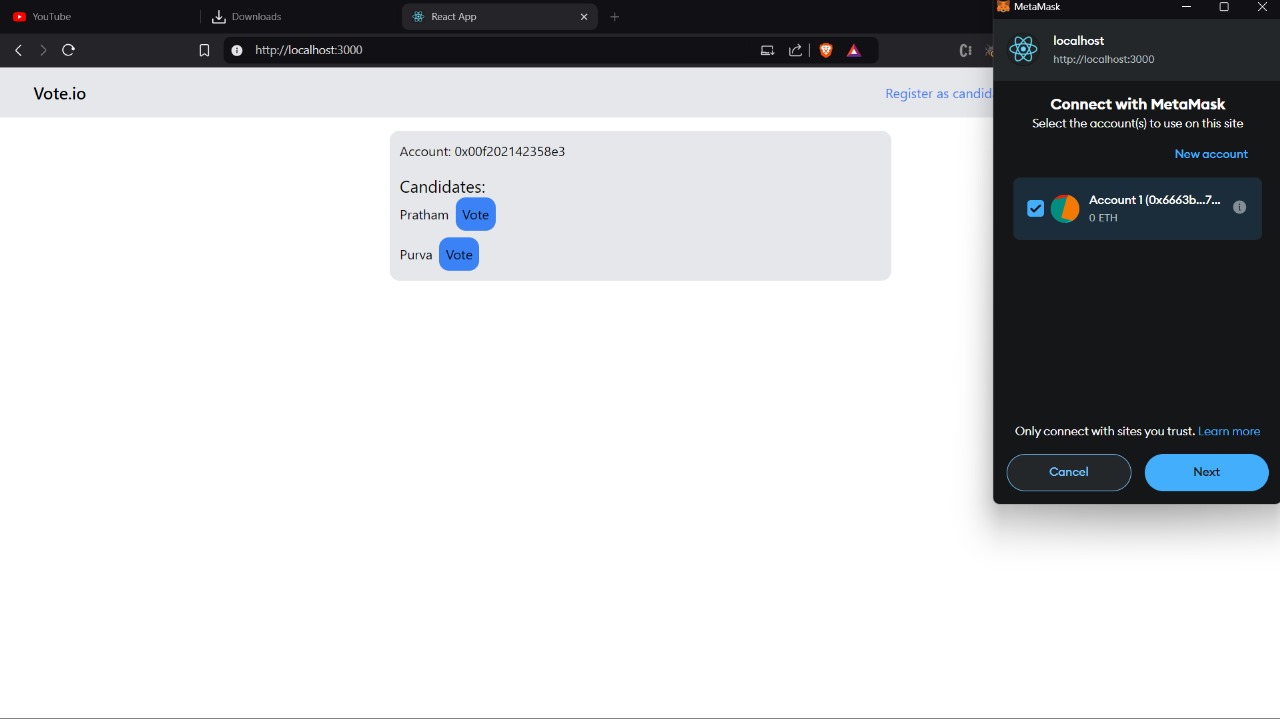


Figure 4.3:Login Page

* **Home Page**

your account is created. With your credentials login in to your account, in the home page any user can conduct elections by entering unique election name, election names can't be same.

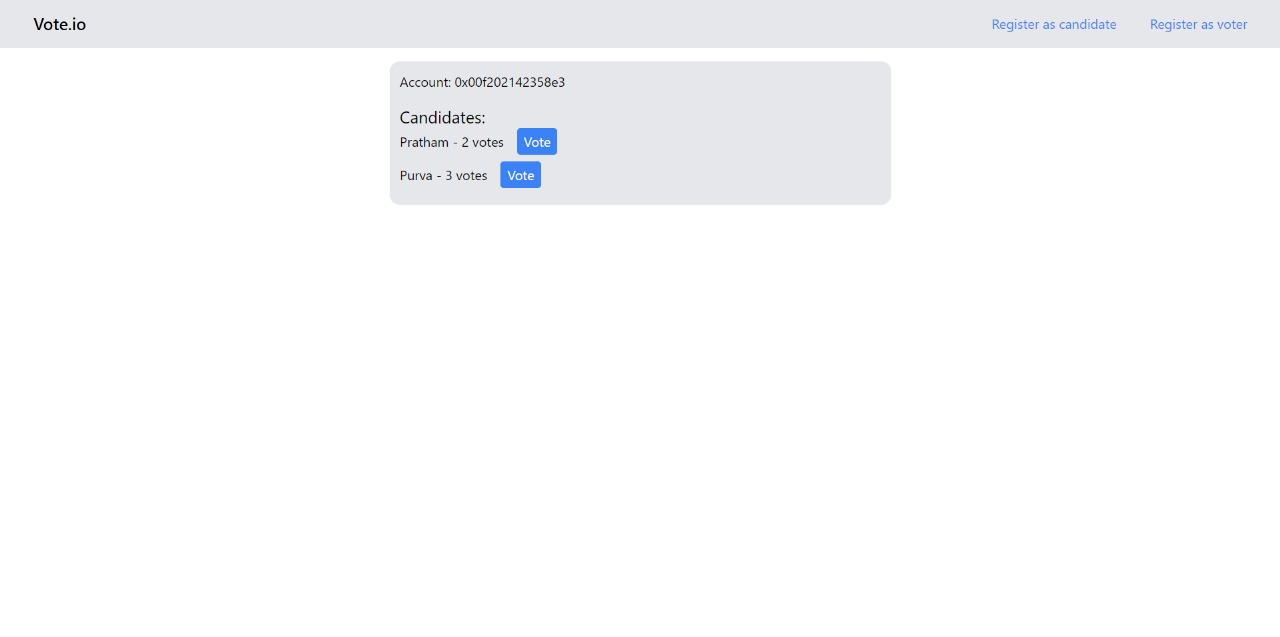
****

Figure 4.4:Home Page

* **Vote Cast**

election name, participating candidates are visible and you can cast, when we change the mode of election to inactive, no user can access to vote.

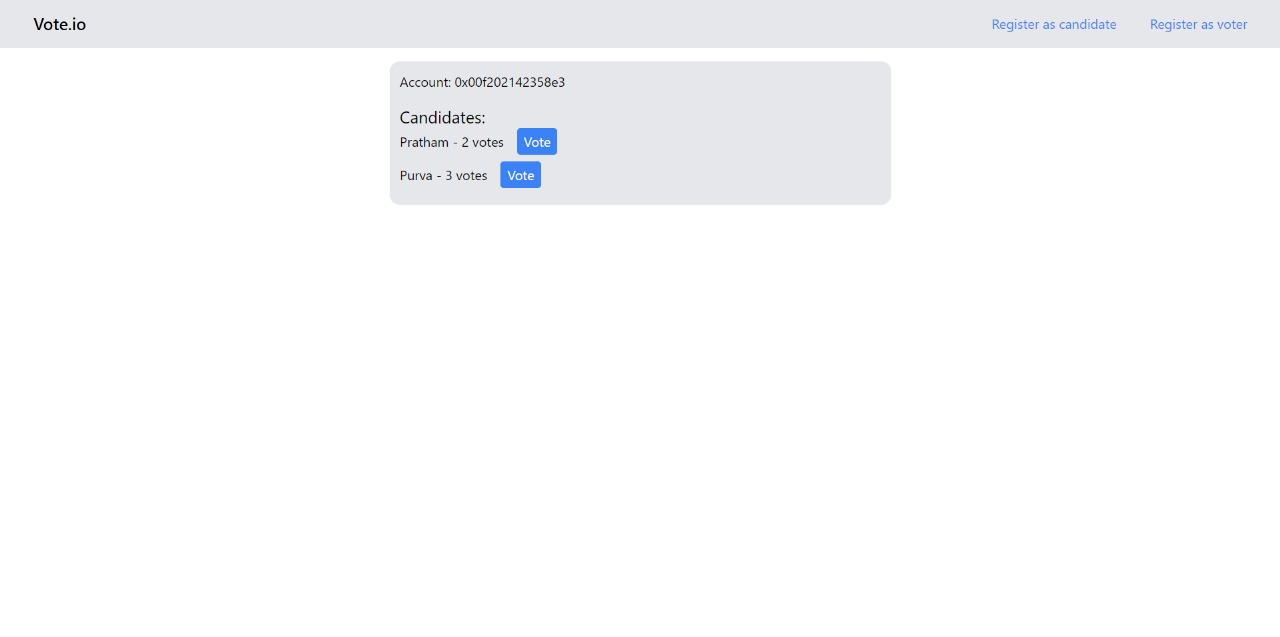
****

Figure 4.5: Vote Casting Page

* **Result**

the election administrator can declare results all the users can see the results.

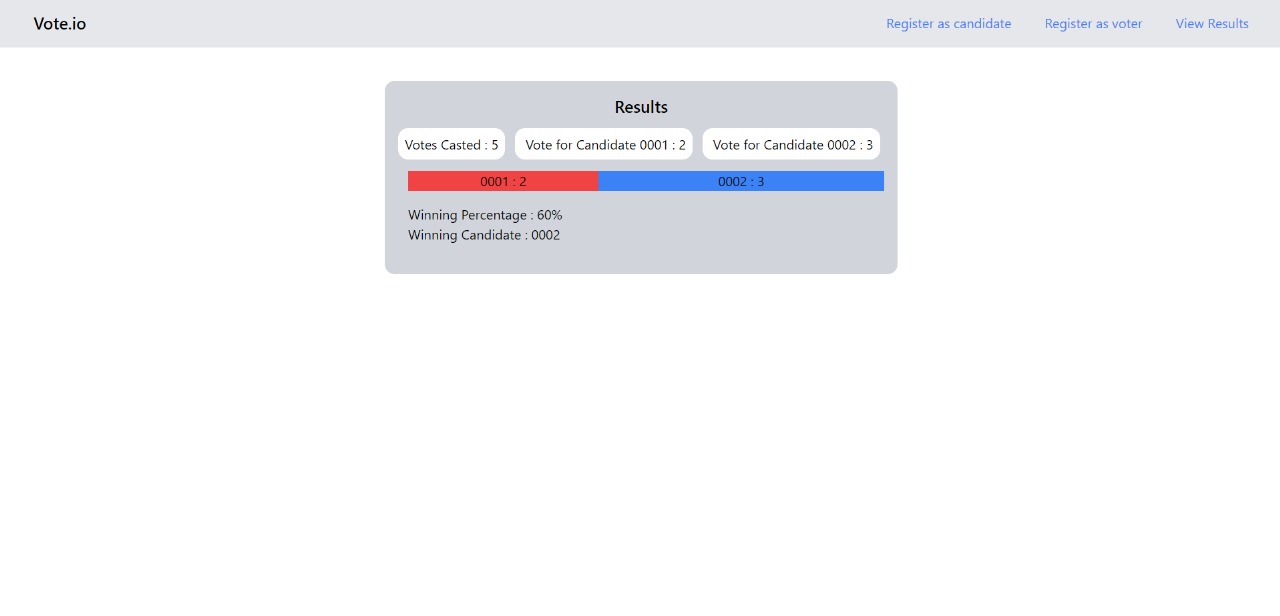


Figure 4.6: Result Page

* **List Of Transaction**

List of transection done while testing.

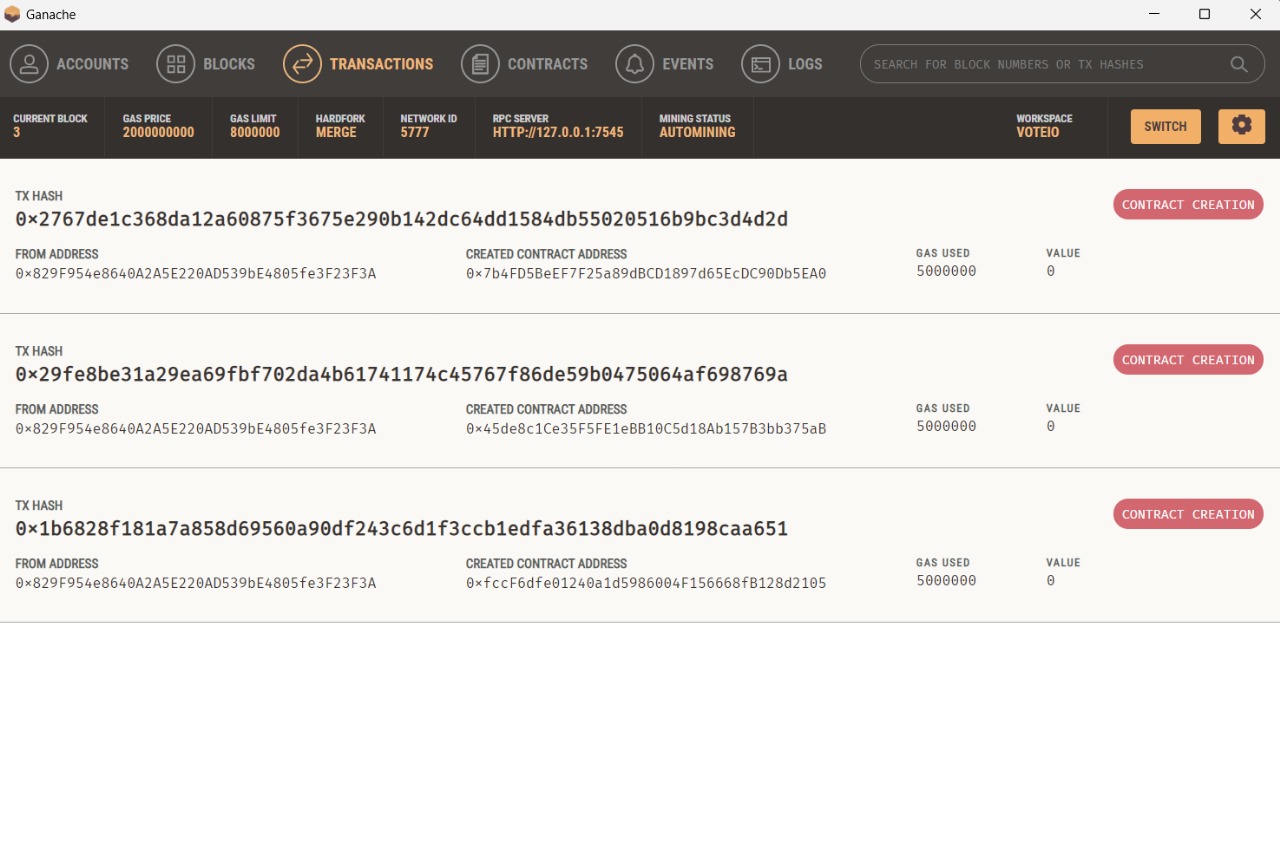


Figure 4.7: List Of Transaction

* **List Of Account**

List of account created for testing while locally hosting the blockchain.

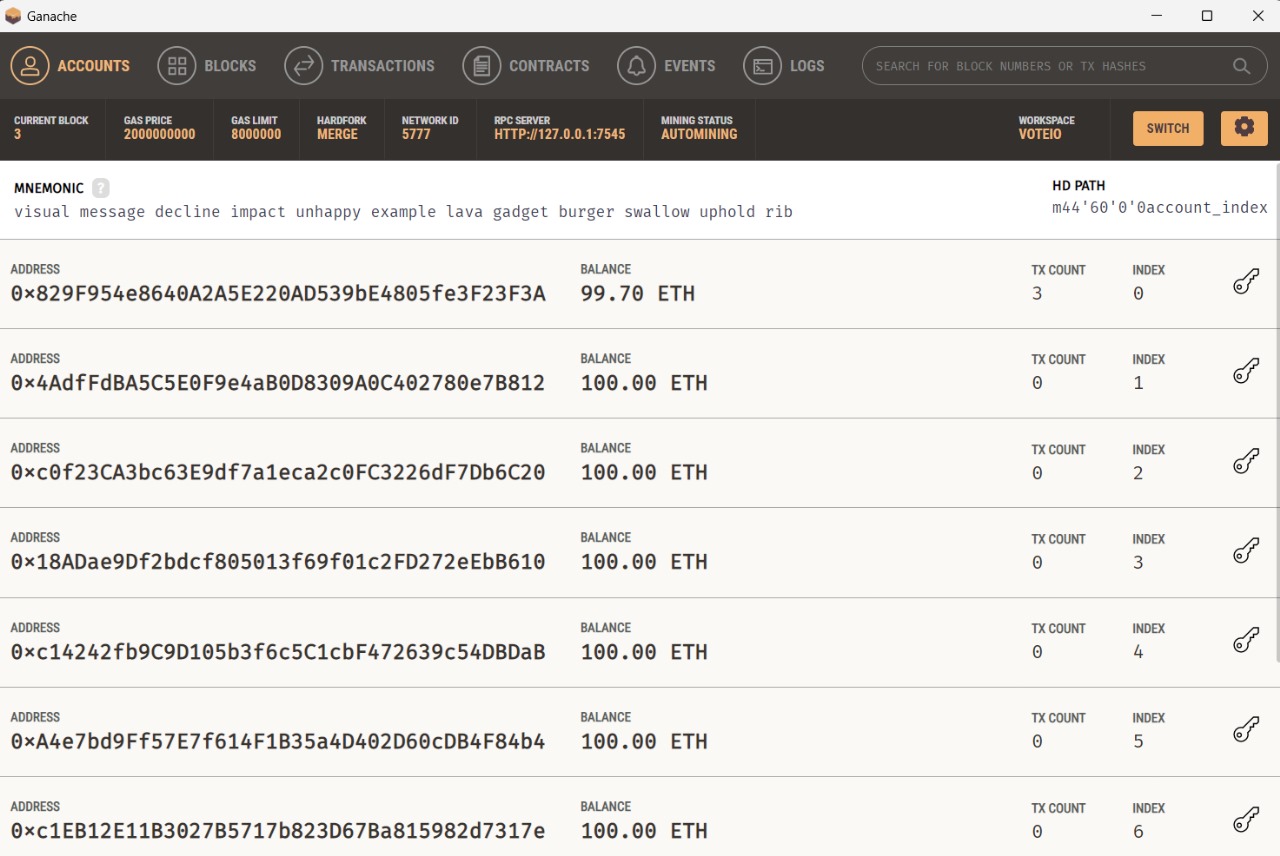


Figure 4.8:List Of Account

* **Acess Metamask**

Give permission to access metamask.

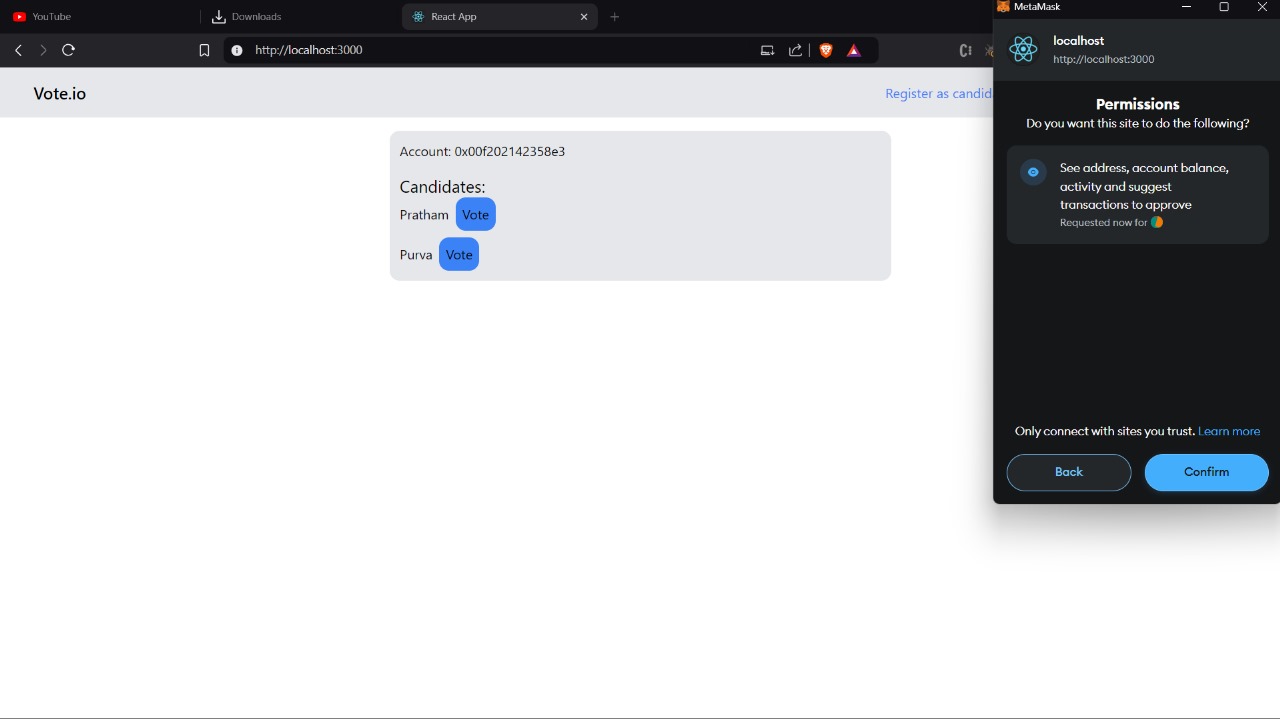


Figure 4.9:Acess Metamask

* **Connecting With Metamask**

Connecting with metamask.

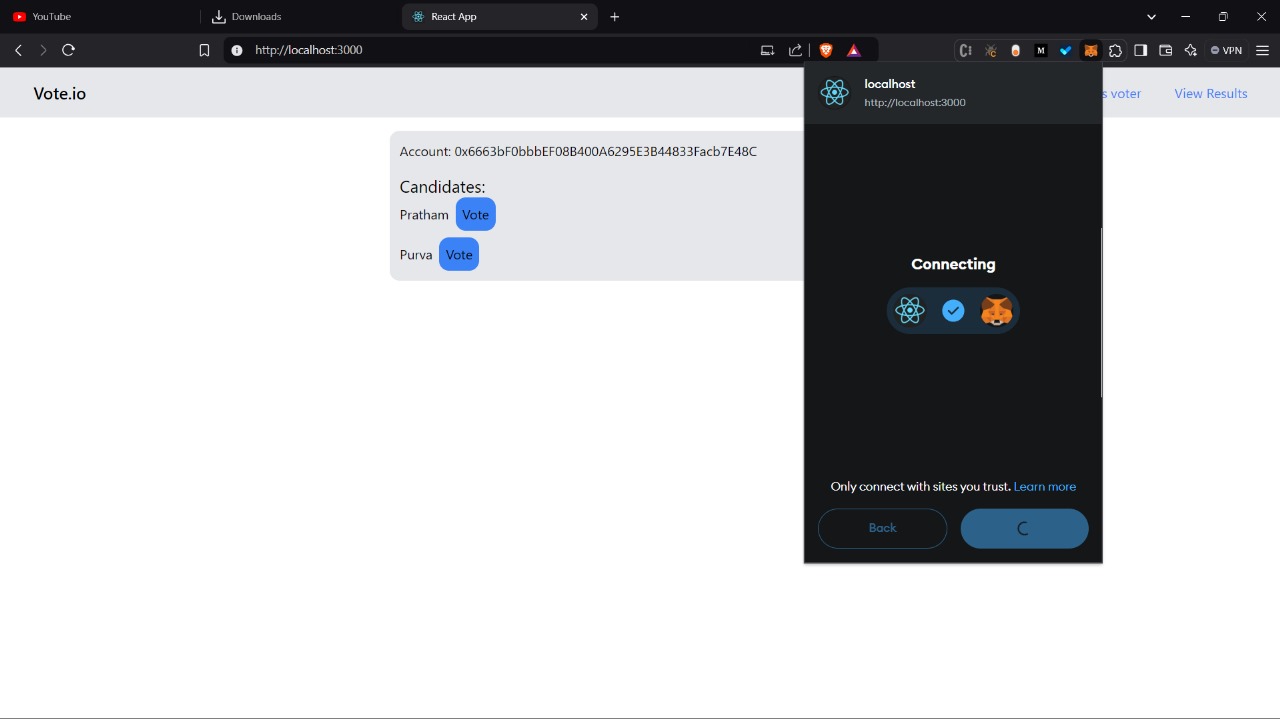


Figure 4.10:Connecting with metamask

* **Connecting With Metamask Done**

Connecting with metamask is done.

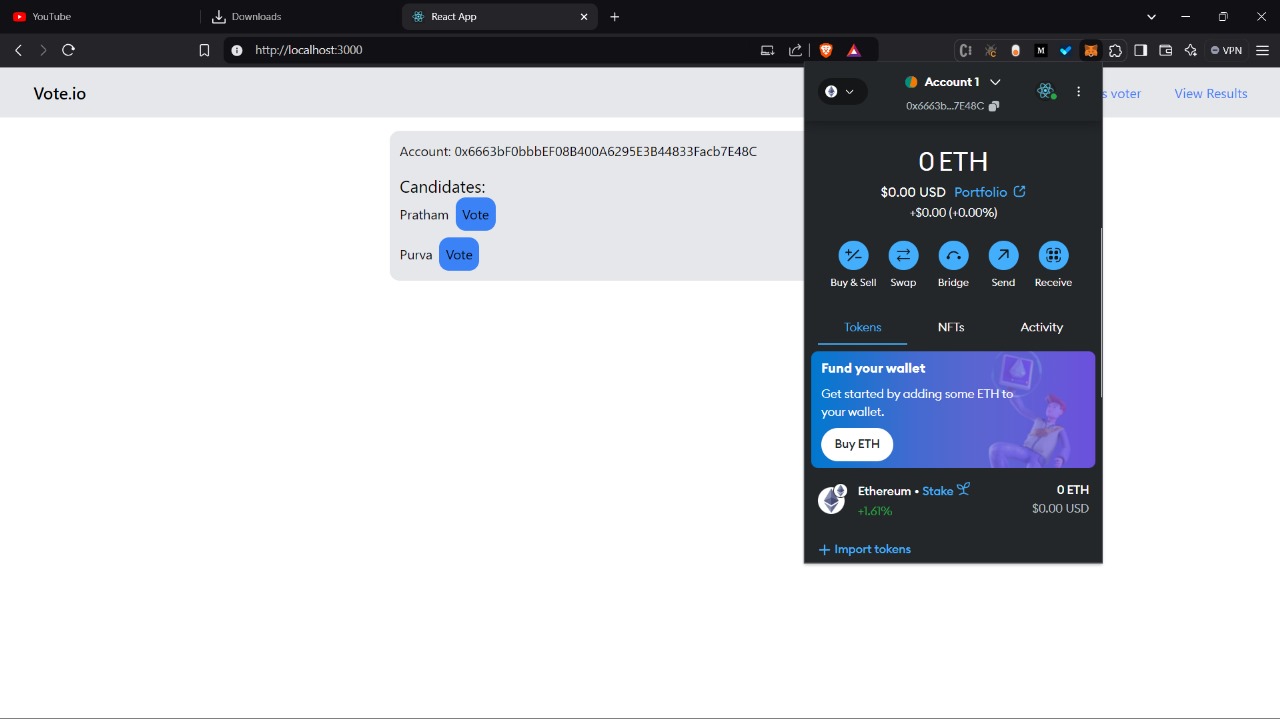


Figure 4.11:Connecting with metamask done

* **Account Gets Updated**

Account gets updated with the account address of Metamask account.

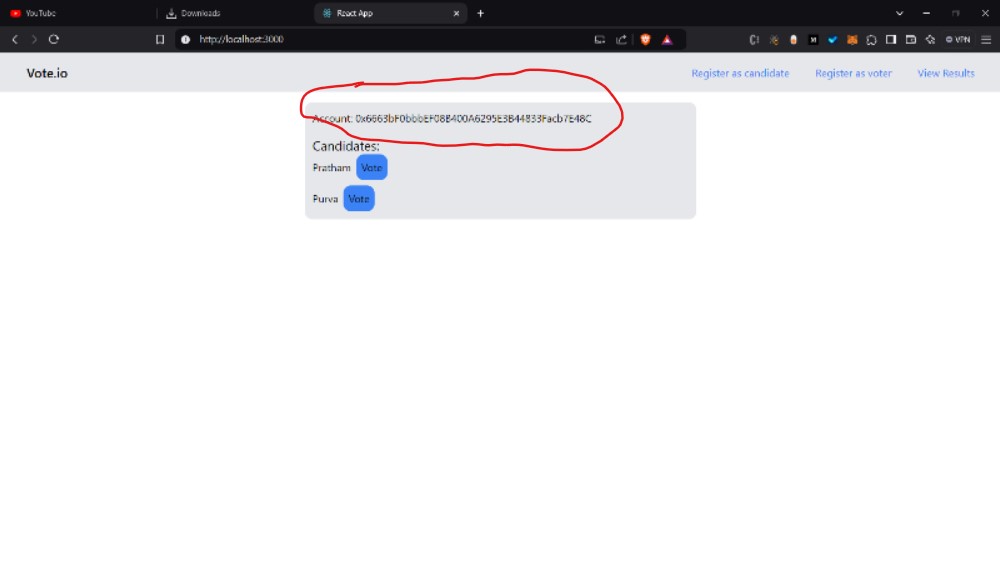
****

Figure 4.12:Account gets updated

* **Deployment of contract in blockchain**

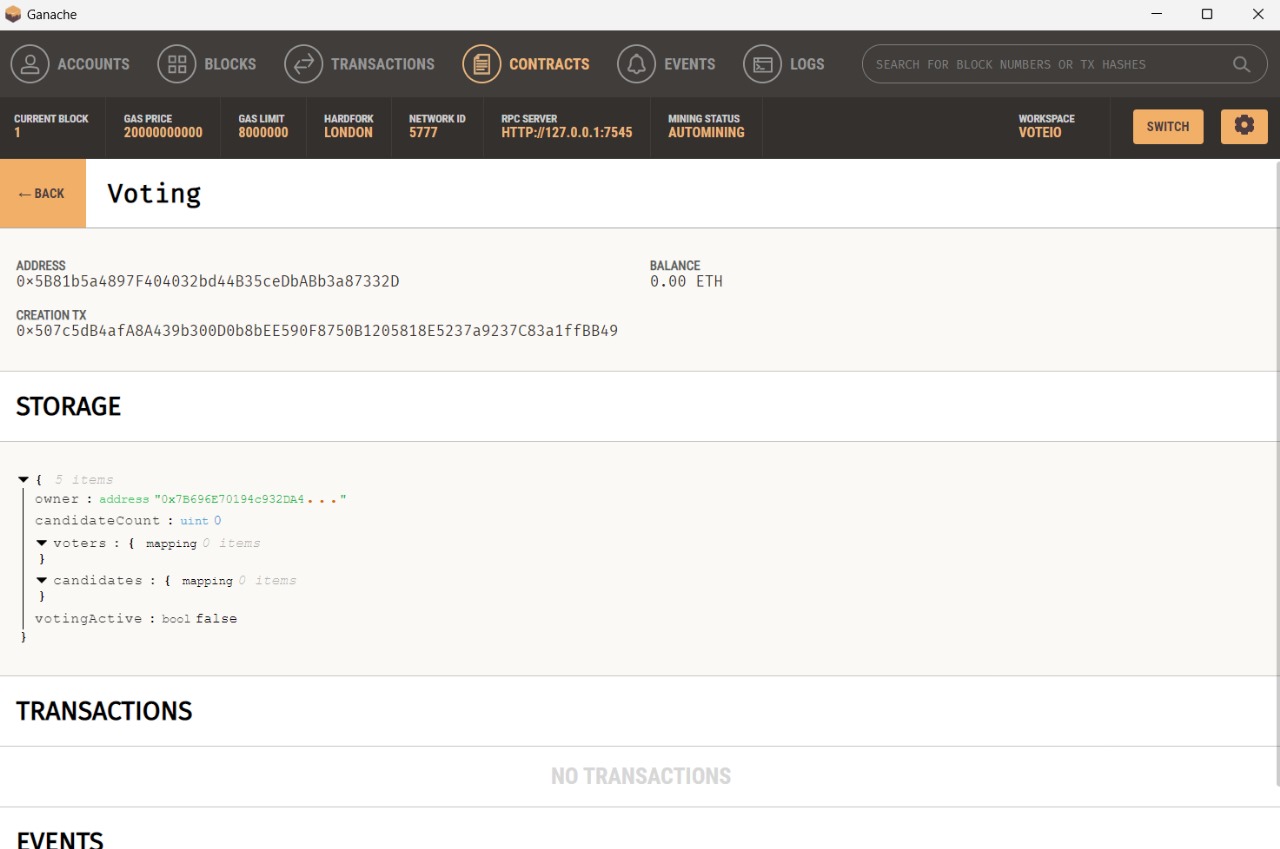
****

figure 4.13: Deployment of contract in blockchain

* **Option to add candidates and voter by owner**

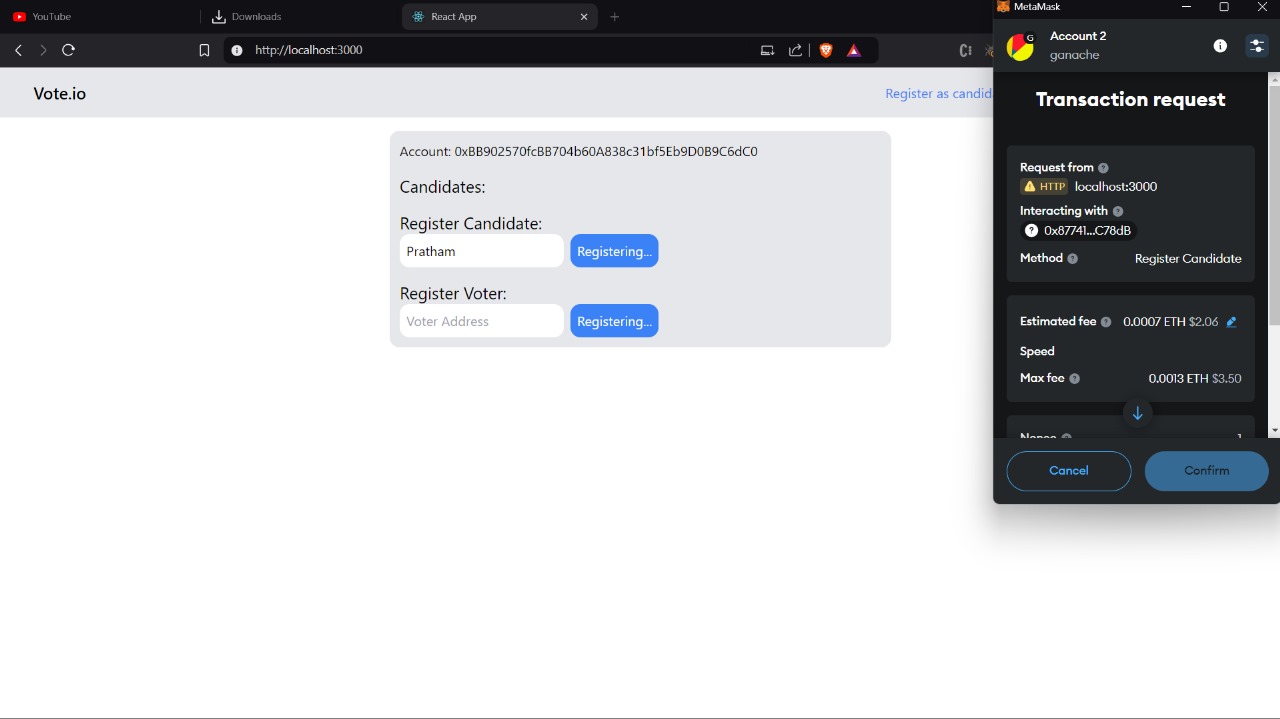


Figure 4.14: Option to add candidates and voter by owner

* **Show all candidates and the vote option**

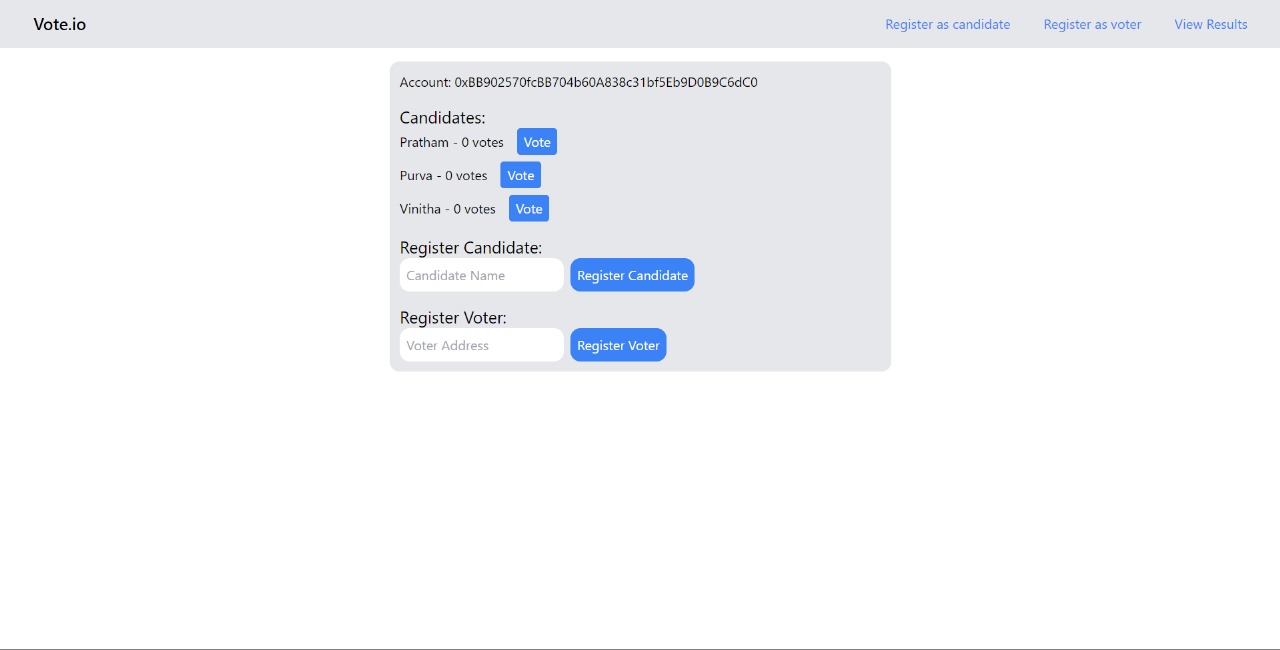


Figure 4.15: Show all candidates and the vote option

* **List of transactions in blockchain**

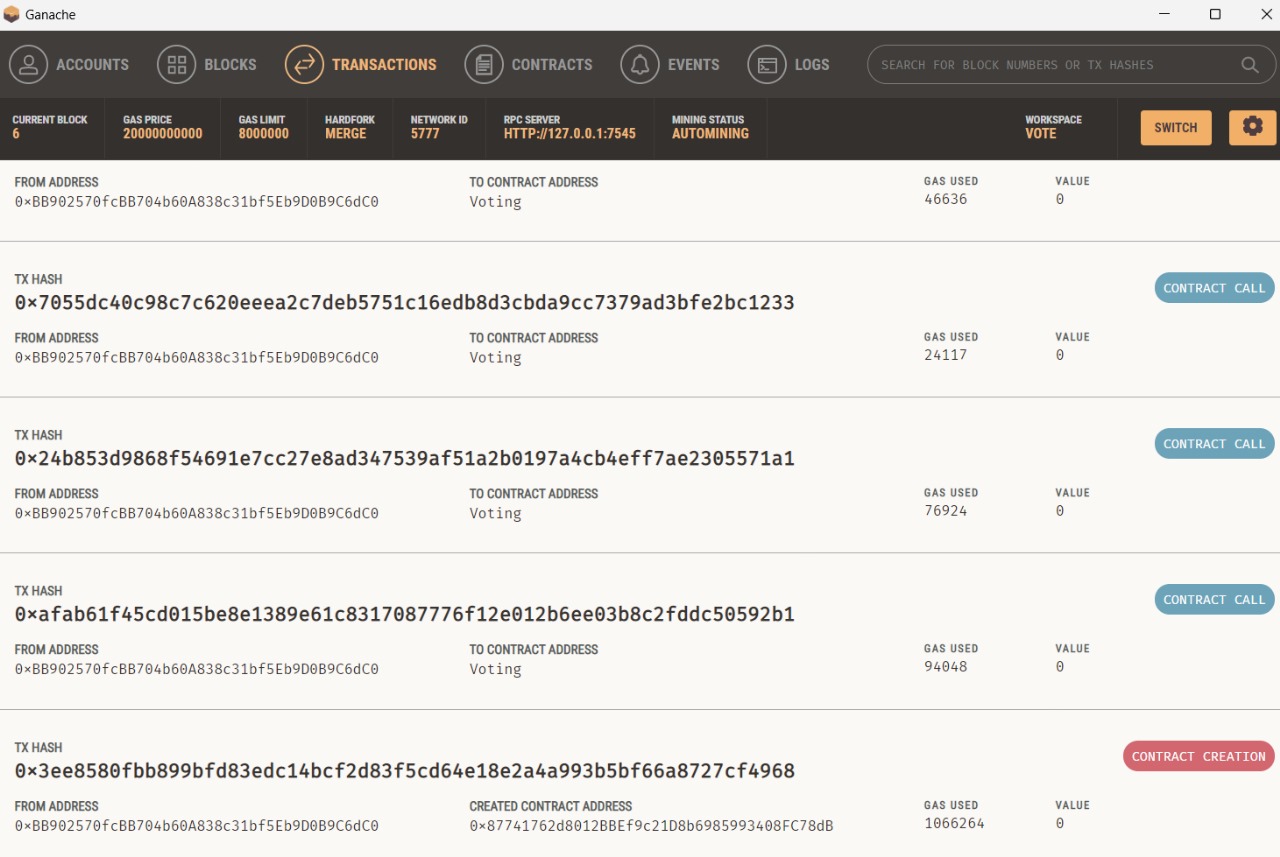


Figure 4.16: List of transactions in blockchain

## Test Cases

Sample test cases are given as below:

Table 4.1: Test Cases

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TestID** | **Case** | **Test Data** | **ExpectedResult** | **ActualResult** | **Pass/fail** |
| 1 | Home Page | metamask Login. | Login using Metamask wallet for blockchain. | Login using Metamask wallet blockchain. | Pass |
| 2 | Register as voter | Register as voter using metamask wallet. | -Voter is able to register.  - Voter is able to vote  -Number of votes is being displayed. | -Voter is able to vote.  - Voter is able to register.  - Number of votes is being shown. | Pass |
| 3 | Register as candidate | Register as candidate using metamask wallet. | -Candidate gets registered  -Candidate Count and details show. | Candidate Count and details show . | Pass |
| 4 | Results | - Number of votes casted.  - Number and details of the candidate.  - Display percentage of outcome and winning party. | - Number of votes and candidates should be displayed from blockchain.  - Display percentage of outcome and winning party in graphical format. | - Number of votes and candidates should be displayed from blockchain.  - Display percentage of outcome and winning party in graphical format. | Pass |
| 5 | Blockchain and Block | - All the transactions being registered in blockchain with addresses  - All the accounts being used visible in Locally hosted blockchain using ganache Ethereum chain. | - All the transactions visible in ganache .  - All the accounts used along with the gas must be visible in ganache. | All the transactions are visible in ganache.  - All the accounts used along with the gas must be visible in ganache. | Pass |

# Conclusion and Future Scope

The proposed framework provides complete security to the e-voting system, with the usage of Ethereum blockchain and smart contracts provide added security to the system. Blockchain implementation prevents vote manipulation and provides privacy, and integrity for voters to cast their vote. Smart contracts ensure that the voter can vote only once using his/her unique ID (Aadhar number); with the convention of different security algorithms like SHA-256, Merkel hash, and SMTP prototyping, enhances the security of the system. As a result, the voter is authorized to cast his/her vote from where ever they are; provides high-security standards to the system and convenient and easier ways to vote.

Future work:

1. To the proposed existing system, additional biometrics (fingerprint, face authentication) can be added to enhance the security of the system.

2. Three-step authentications can also be used to provide more security to the system.

References

**Paper references**

[1] FREYA SHEER HARDWICK, APOSTOLOS GIOULIS, RAJA NAEEM AKRAM and i KONSTANITINOS MARKANTONAKIS: "E-voting with Blockchain: An e-voting protocol with decentralization and voter privacy.  
[2] ALI KAAN KOC and EMRE YAVUZ: "Towards secure E-voting Using Ethereum Blockchain.  
[3] NIR KSHETRI and JEFFREY VOAS: "Blockchain-enabled E-voting".  
[4] SHEKHAR MISHRA and Y. ROJA: "electronic voting machine using biometric ginger print with Aadhar card authentication.  
[5] AMNA Qureshi "SEVEP: Verifiable, secure and privacy preserving remote polling with untrusted computing devices," in Future Network Systems and Security Feb 22(2019) ΙΕΕΕ.  
[6] RIFA HANIFATUNISA and BUDI RAHARDJO: "Blockchain based e-voting recording system design".  
[7] KANIKA GRAB, PAVI SARADWAT, SACHINE BISHT, SAHIL AGGARVAL, SAI KRISHNA KOTHURI and SAHIL GUPTA: "A comparative analysis on e-voting system using blockchain".  
[8] ISANI MANDAI: "Secure and hassle free EVM through deep learning face recognition".  
[9] S. GANESH PRABHU, RACHEL, AGENS SHINYH, and A. R. ROSHINEE. "Tracking Real Time Vehicle and Locking System Using Lab View Applications." In 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), pp. 55-57.IEEE, 2020.  
[10] ANNOSHMITHA DAS "VOT-EL: Three Tier Secured State Of-The-Art EVM DesignUsing Pragmatic Fingerprint Detection Annexed with NFC Enabled Voter -ID Card" (2016) ΙΕΕΕ.