# A BLOCKCHAIN PROJECT

## FOR

## ELECTRONIC VOTING SYSTEM

***Submitted By***


**TEAM MEMBERS:**

**JEHAN MANDAL**

**SATHISH.A**

**ANKIT KUMAR**

**SUMAN.M**


**TEAM ID: NM2023TMID11339**


**BHARATHIDASAN ENGINEERING COLLEGE**

**NATRAMPALLI- 635854**

# TABLE OF CONTENT

# 1. INTRODUCTION

# 1.Introduction

Building an Electronic voting system that offers the fairness and privacy of current voting schemes, while providing the transparency and flexibility offered by electronic systems has been a challenge for a long time. We evaluate an application of blockchain as a service to implement E-voting systems. The paper proposes an electronic voting system based on blockchain that addresses some of the limitations in existing systems and evaluates some of the popular blockchain frameworks for the purpose of constructing a blockchain based E-voting system.

## 1.1 PROJECT OVERVIEW

The implementation of a blockchain-based application, which improves the security and decreases the cost of hosting a nation wide election . The outdated paper ballot system and the widely used electronic voting devices can both be replaced by online voting. In addition to the openness of votes and the privacy of voters, an electronic voting site should provide security and integrity. This study suggests a blockchain-based electronic voting system that overcomes some of the drawbacks of the current voting methods. The report also discusses the current state of certain blockchain voting frameworks. The implementation that is currently being used is appropriate for small-scale elections held inside of offices, boardrooms, etc.

## 1.2 PURPOSE

This work proposes a very straightforward method for a fair electronic voting system that ensures anonymity, coercion resistance, correctness, ease of tallying, eligibility, fairness, high availability, integrity, and robustness. It also presents voter authentication, voter confidentiality, vote verifiability, and public verifiability. The blockchain technology is used to attain these characteristics.

Key Words: BLOCKCHAIN, E-VOTING

# 2. LITERATURE OF SURVEY

# 2. LITERATURE SURVEY

## 2.1 EXISTING SYSTEM

- ❖ Existing system is a manual one in which users and the details of the candidates are stored in books. The users have to wait a long time in queues for voting. Wrong and unwanted votes are given. Counting of votes are done manually which takes lots of time and inaccurate counting is done. It is very difficult to maintain historical data.

- ❖ In the existing system, there is compulsory need in physical presence in the time of election polling or vote counting.

## 2.2 REFERENCES


1. Archit Pandey, Mohit Bhasi and K.Chandrasekaran, “Votechain: A Blockchain Based D- Voting System”, 2019 Global Conference for Advancement in Technology (GCAT), INSPEC Accession Number: 19319487, doi: 10 .1109/ GCAT47503.2019.89 78295.
2. S.Gopi, B.Giridharan, R.Mohamed Rifoy, S. Sivachidambaram and S.Yuvaraja, “Effective In- house Voting and Implementation Using Blockchain Verification”, e-ISSN: 2395-0056, vol. 6 Issue: 03 | Mar 2019.
3. Venkata Naga Rani, Akshay S, Arun Kumar and Ishwar Kumar MA, “Decentralized E-voting System”, p-ISSN: 2395-0072, vol. 6 Issue: 03 | Mar 2019.
4. https://link.springer.com/content/pdf/10.1007%2F978-0-387-35586- 3_40.pdf
5. https://www.dappuniversity.com/articles/the-ultimate-ethereum-dapp-tutorial
6. Mrs. Harsha V Patil, Mrs. Kanchan G Rathi and Mrs. Malathi V Tribhuwan, “A
    a. Study on Decentralized E-Voting System Using Blockchain Technology”, pISSN: 2395-0072, vol. 5 Issue: 11 | Nov 2018
7. Abhishek Subhash Yadav , Yash Vandesh Urade , Ashish Uttamrao Thombare ,
    a. Abhijeet Anil Patil, 2020, E-Voting using Blockchain
    b. Technology,INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 09, Issue 07 (July 2020).

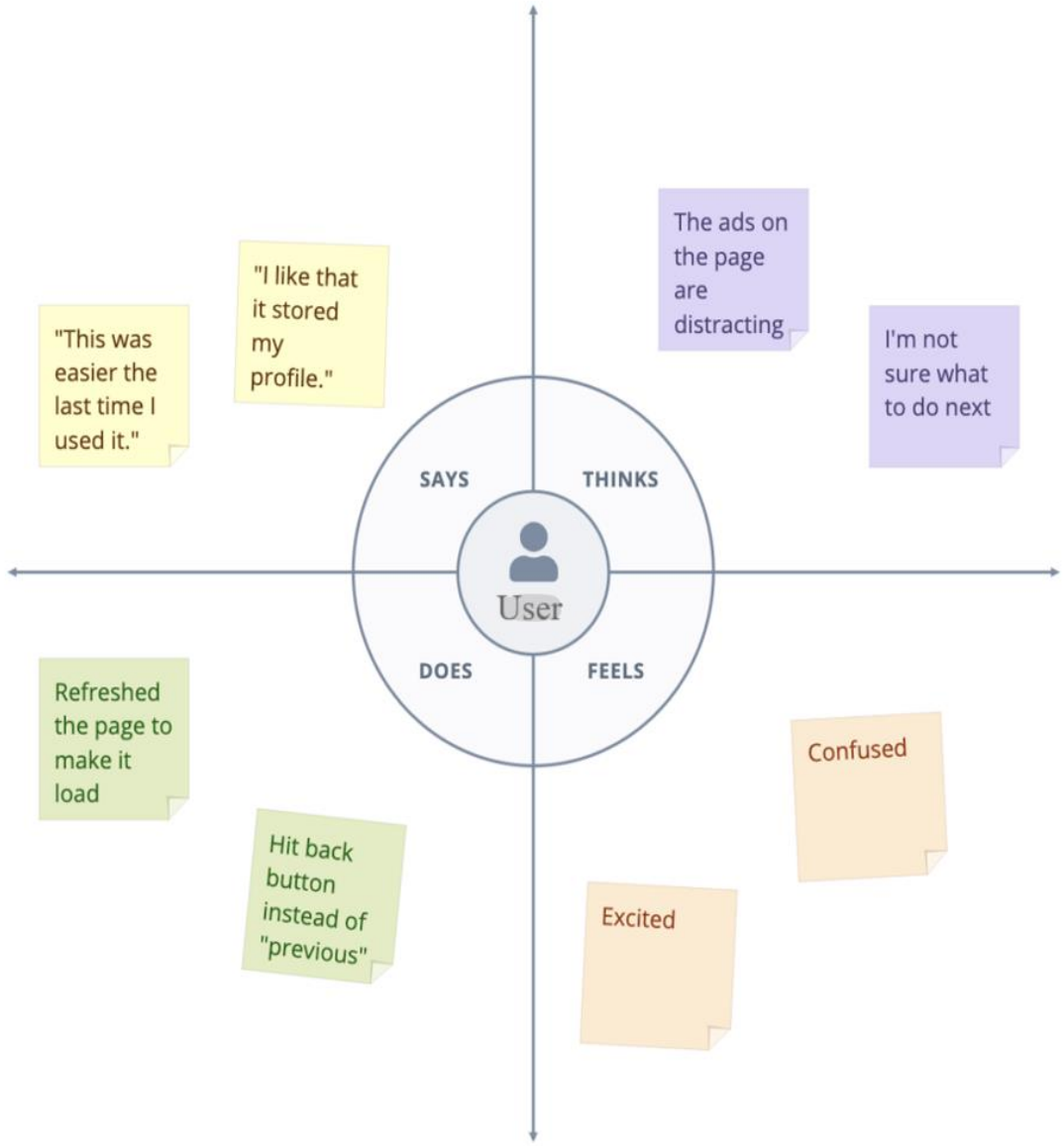
## 2.3 PROBLEM STATEMENT DEFINITION

Many theories and researches have been proposed to explain the effective working of different E-voting systems, although the literature review covers the wide variety of such theories and researches. These reviews will focus on major themes which emerged repeatedly throughout when implementing the counting and electronic voting projects. These themes are: Making a decision, build the system and implementing a system. Although this section presents these themes in a variety of context, background, related study and comparison. This section will primarily focus on working, problems and enhancement of current technologies in E-voting system.

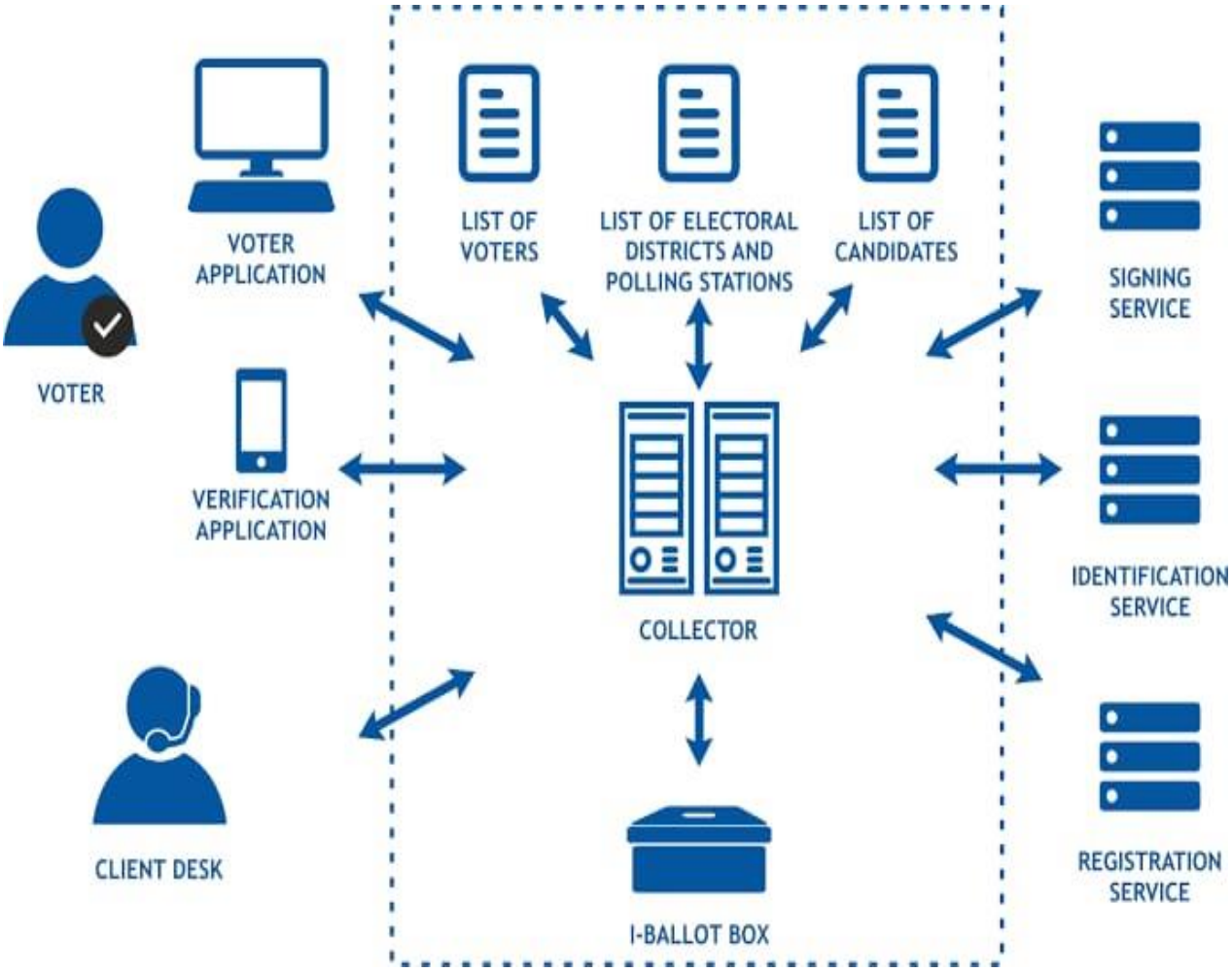
# 3.IDEATION & PROPOSED SOLUTION

# 3.IDEATION & PROPOSED SOLUTION

## 3.1 EMPATHY MAP CANVAS



## 3.2 IDEATION & BRAINSTORMING



# 4. REQUIREMENT ANALYSIS

# 4. REQUIREMENT ANALYSIS

## 4.1 FUNCTIONAL REQUIREMENTS

1.Mobility: The voter should not be restricted to cast his ballot at a single poll-site at his home precinct.

Realistic: He shall be able to vote from any poll-site within the nation.

Unrealistic/Expensive: He shall be able to vote from any county-controlled kiosk (situated at public places such as banks, shopping malls, etc.) within the nation. (Unrealistic because of logistical and cost issues).

Infeasible: He shall be able to vote from virtually anywhere using an Internet connection. (Infeasible both for technical security issues as well as social science issues).

2. Convenience: The system shall allow the voters to cast their votes quickly, in one session, and should not require many special skills or intimidate the voter (to ensure Equality of Access to Voters).

3. User-Interface: The system shall provide an easy-to-use user-interface. Also, it shall not disadvantage any candidate while displaying the choices (e.g., by requiring the user to scroll down to see the last few choices).

4. Transparency: Voters should be able to possess a general knowledge and understanding of the voting process.

5. Flexibility: The system shall be flexible in that it allows a variety of ballot question formats including open-ended questions (e.g. Write-in candidates and survey questions).

## 4.2 NON FUNCTIONAL REQUIREMENTS

1. Voter Authenticity: Ensure that the voter must identify himself (with respect to the registration database) to be entitled to vote. If voting other than at his home precinct, the voter may be asked to show some legal identification document.

2. Registration: The voter registration shall be done in person only. However, the computerized registration database shall be made available to polling-booths all around the nation.

3. Voter Anonymity: Ensure that votes must not be associated with voter identity.

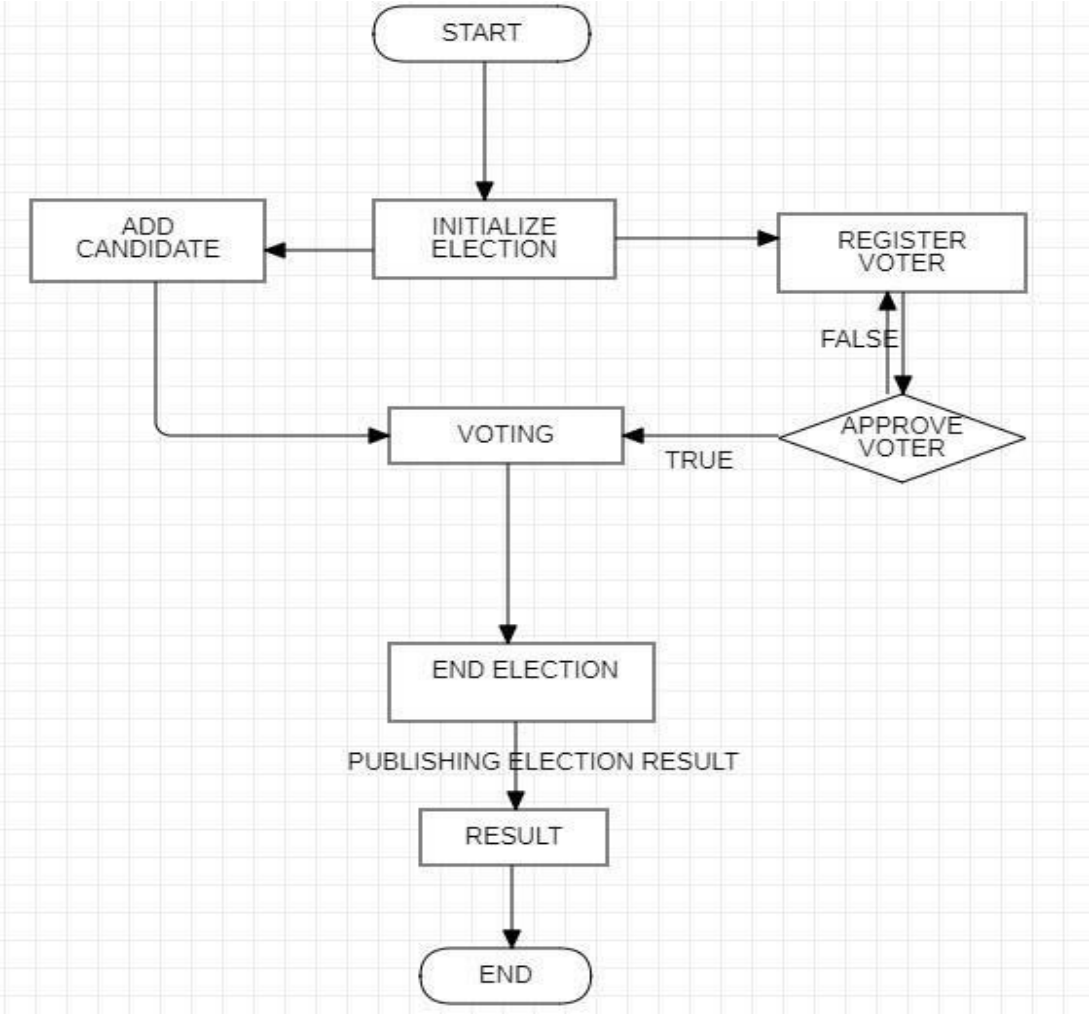
4. System Integrity: Ensure that the system cannot be re-configured during operation.

5. Data Integrity: Ensure that each vote is recorded as intended and cannot be tampered with in any manner, once recorded (i.e., votes should not be modified, forged or deleted without detection).

# 5. PROJECT DESIGN

# 5. PROJECT DESIGN

## 5.1 DATA FLOW DIAGRAM & USER STORIES

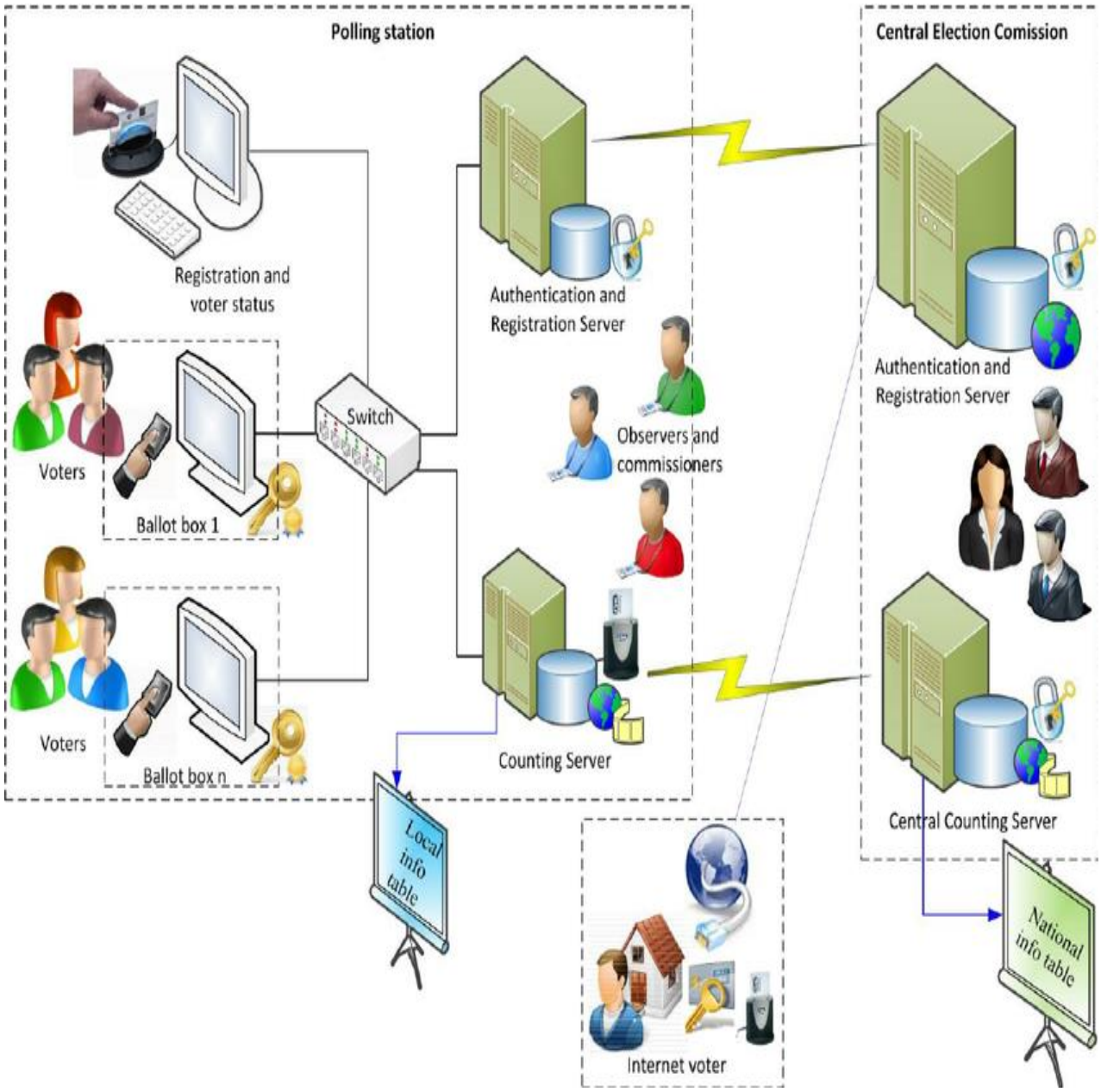



USER STORY:

The user interface is the only component of the voting system that is visible to voters. This plays a key role in ensuring that voting proceeds smoothly and voters feel comfortable with the result. The makers, particularly those officials responsible for designing and implementing e-voting systems, should insist that user interfaces for electronic voting systems are carefully designed, evaluated, and field – tested to meet agreed-upon usability objectives prior to implementation and use in elections.

As a result, user interfaces for e-voting systems must be more robust than many other user interfaces. They must provide simple and clear instructions, offer graceful recovery from user errors, provide adequate and detailed feedback regarding the effects of user actions and they must handle user and system errors and mistakes. Furthermore, these interfaces must meet all of these requirements for voters who vary widely in age, skills, physical abilities, education and cultural background, while guaranteeing confidentiality for all voters

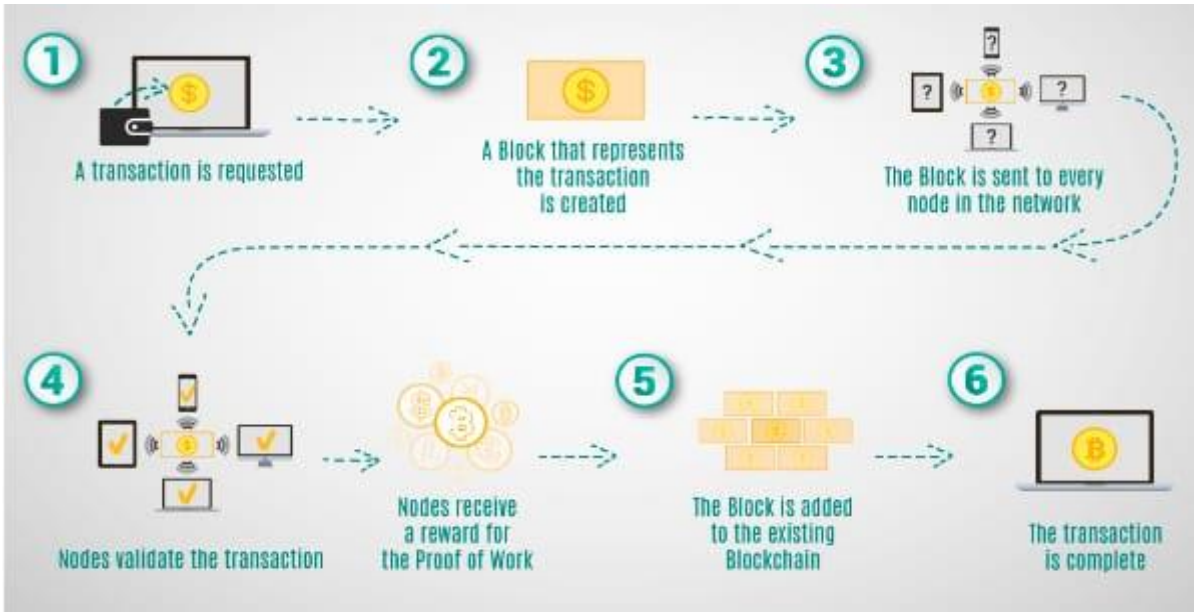
## 5.2 SOLUTION ARCHITECTURE



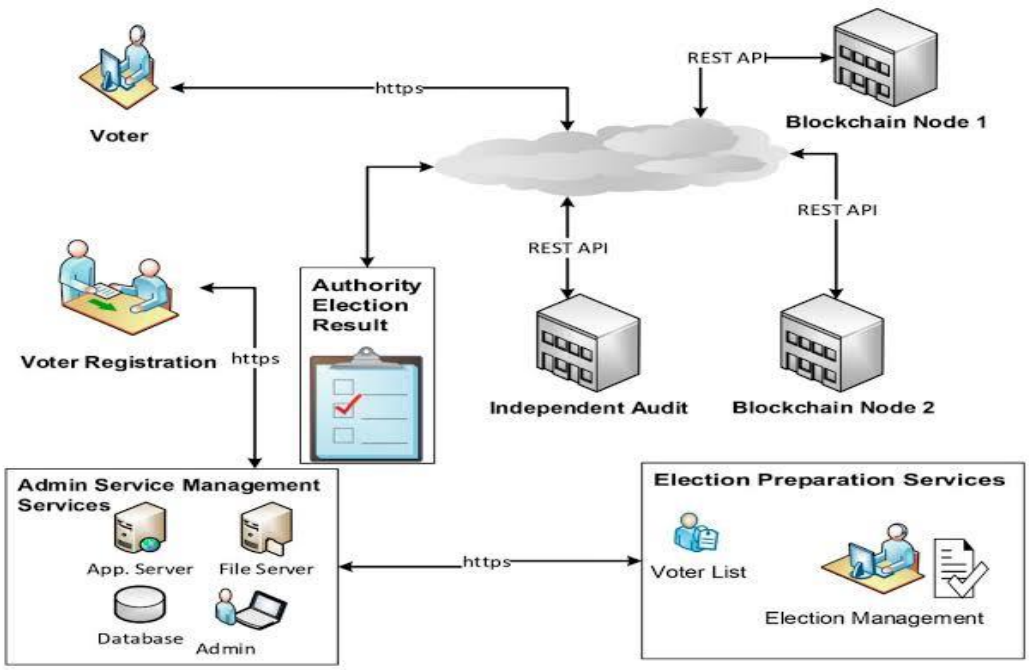
# 6. PROJECT PLANNING & SCHEDULING

# 6. PROJECT PLANNING & SCHEDULING

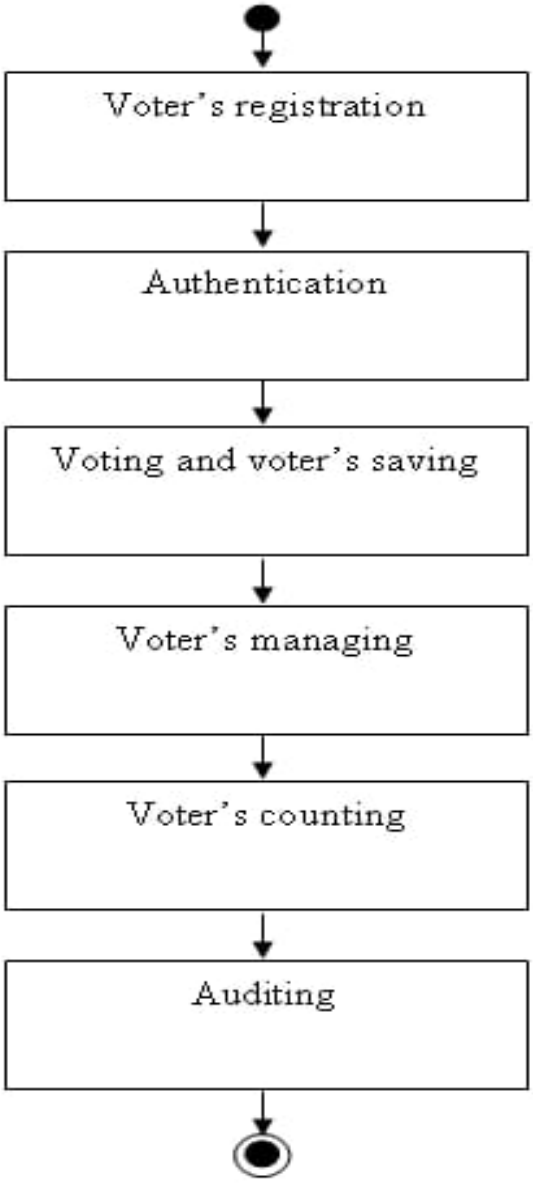
## 6.1 TECHNICAL ARCHITECTURE




## 6.2 SPRINT PLANNING & ESTIMATION



## 6.3 SPRINT DELIVERY SCHEDULE



# 7.CODING & SOLUTIONING

# 7.CODING & SOLUTIONING

## 7.1 SOURCE CODE

### Index.html

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<link rel="icon" href="%PUBLIC_URL%/favicon.ico" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<meta name="theme-color" content="#000000" />

<meta name="description" content="Web site

created using create-react-app"

/>

<link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />

<!-- manifest.json provides metadata used when your web app is installed

on a

user's mobile device or desktop. See
https://developers.google.com/web/fundamentals/webapp-manifest/

-->

<link rel="manifest" href="%PUBLIC_URL%/manifest.json" />

<!--
```

```
    Notice the use of %PUBLIC_URL% in the tags above.
    It will be replaced with the URL of the public folder during the build.
    Only files inside the public folder can be referenced from the HTML.
  Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will work
    correctly both with client-side routing and a non-root public URL.
    Learn how to configure a non-root public URL by running npm run build.
    -->
    <title>React App</title>
    </head>
    <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>
    <!--
    This HTML file is a template.
    If you open it directly in the browser, you will see an empty page.

    You can add webfonts, meta tags, or analytics to this file.
    The build step will place the bundled scripts into the <body> tag.

    To begin the development, run npm start or yarn start.
    To create a production bundle, use npm run build or yarn build.
    -->
```

```
</body>

</html>
```

## Voting Sol

```
// SPDX-License-Identifier: MIT pragma
solidity ^0.8.0;

contract VoteSystem{
    address public owner;

    constructor(){
owner= msg.sender;
    }

 struct candidate {
uint voterId;
string name;
uint age;      uint
voteCount;
 }

 mapping (uint => candidate) candidateMap;

 struct voters {
uint voterId;
string name;      uint
age;      bool
votingState;
 }

 mapping (uint => voters) votersMap;
 mapping (uint=>bool) registeredVoter;

 modifier checkVoterVoted(uint _votersVoterId){
     require (votersMap[_votersVoterId].votingState == false);
     _;
}
```

```
modifier checkRegisteredVoter(uint _votersVoterId){
require(registeredVoter[_votersVoterId]==true, "Voter is not Registered");
        _;
} uint[]
voterIdlist;
uint[] candidateIdList;


 function enrollCandidate(uint _voterId,string memory _name,uint  _age )  public {  require
(_age >= 25);
 require (candidateMap[_voterId].voterId != _voterId);

    candidateMap[_voterId].voterId = _voterId;
candidateMap[_voterId].name = _name;
    candidateMap[_voterId].age = _age;

    candidateIdList.push(_voterId);
 }

 function enrollVoter(uint _voterId,string memory _name,uint _age)  public  returns(bool){

require (_age >= 18);
require (votersMap[_voterId].voterId != _voterId);

     votersMap[_voterId].voterId = _voterId;
votersMap[_voterId].name = _name;      votersMap[_voterId].age
= _age;

     voterIdlist.push(_voterId);
    return registeredVoter[_voterId]=true;

 }

 function getCandidateDetails(uint _voterId) view public returns(uint,string memory,uint,uint)
{

     return
(candidateMap[_voterId].voterId,candidateMap[_voterId].name,candidateMap[_voterId].age,
candidateMap[_voterId].voteCount);
 }

 function getVoterDetails(uint _voterId) view public returns (uint,string memory,uint,bool){
```

```
    return
(votersMap[_voterId].voterId,votersMap[_voterId].name,votersMap[_voterId].age,votersMap
[_voterId].votingState);

 }

 function vote(uint _candidateVoterId,uint _votersVoterId) public
checkVoterVoted(_votersVoterId) checkRegisteredVoter(_votersVoterId) {
candidateMap[_candidateVoterId].voteCount += 1;
votersMap[_votersVoterId].votingState = true;  }

 function getVotecountOf(uint _voterId) view public returns(uint){
require(msg.sender== owner, "Only owner is allowed to Check Results");
return candidateMap[_voterId].voteCount;  }

 function getVoterList() view public returns (uint[] memory){

   return   voterIdlist;
   }

 function getCandidateList() view public returns(uint[] memory){


 return candidateIdList;
 }


}
```

# 8.PERFORMANCE MATRICS

## 8.PERFORMANCE MATRICS

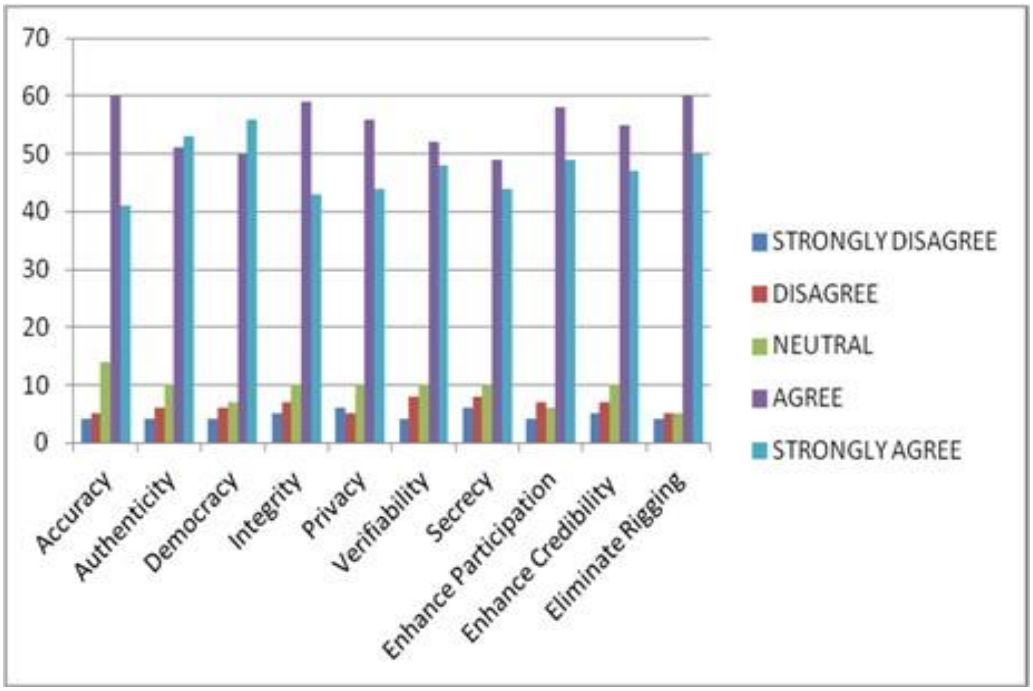



**Fig.** Results of the Performance Evaluation of the Generic and Functional Requirements of a Multifaceted Voting Framework

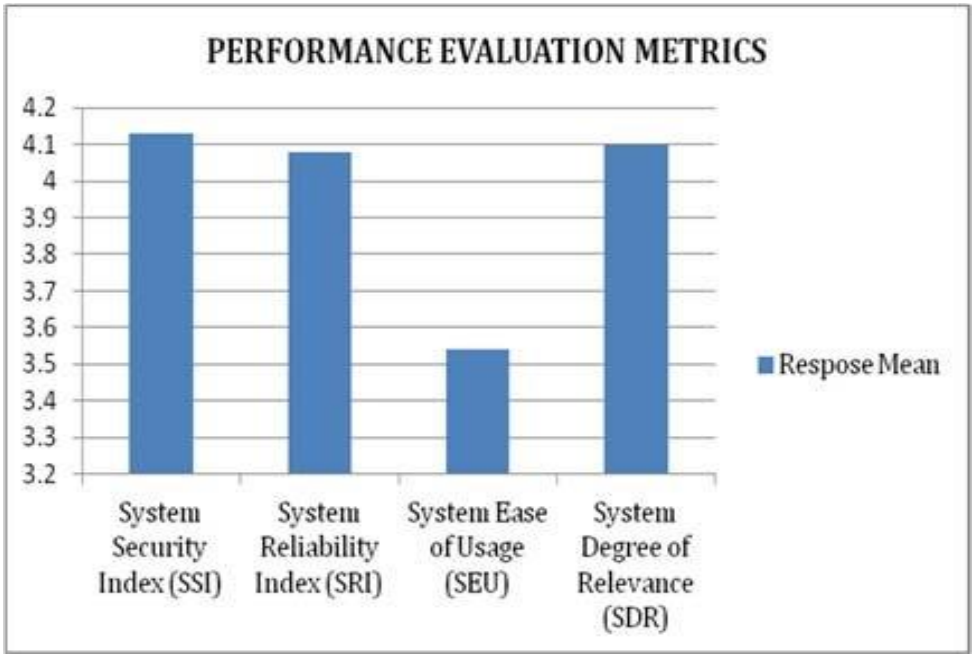



**Fig.** Graphical Representation of the Results of the Performance Evaluation Metrics

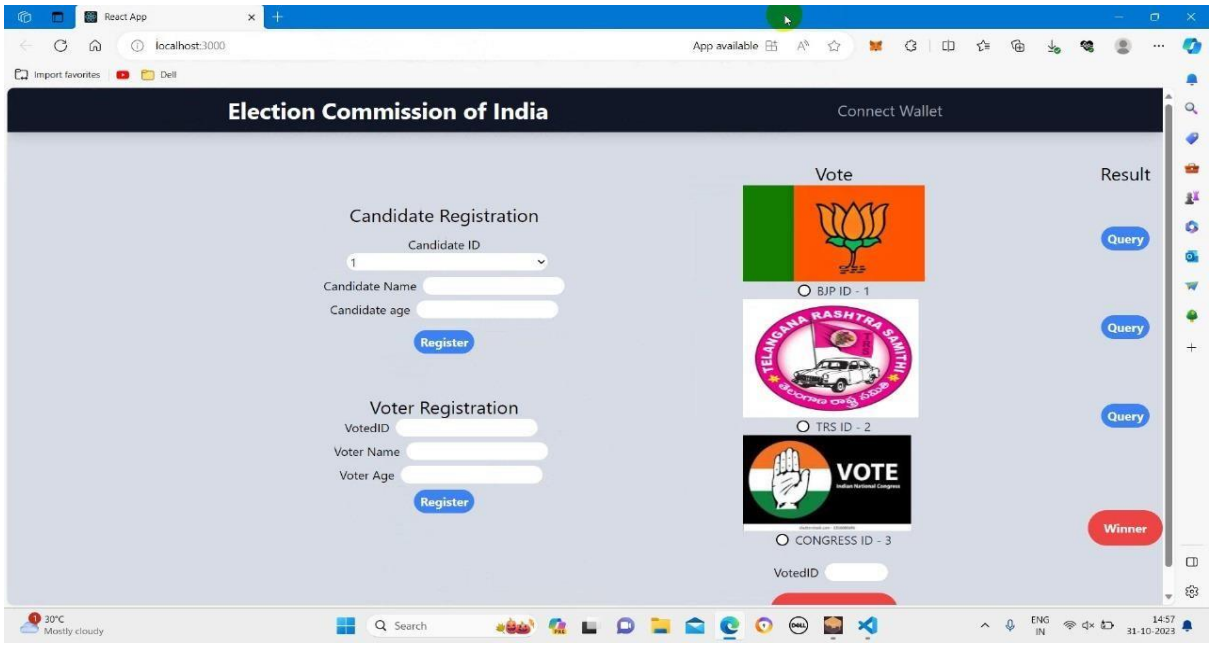
The performance assessment of the developed stegano-cryptographic e-voting model was evaluated for authentication, secrecy, confidentiality, verifiability and functional security requirements of a secured e-voting system quantitatively through perceptive assessment of these qualities by users of acceptable age range (18years and above) by administration of user perceptive form (questionnaire). The assessment form contains the established four fundamental security requirements of a secured e-voting system as well as other functional parameters such as scope for multiple registration, rigging, method of voting, technical requirements of these methods of voting, democracy, participatory effect of implementing the developed e-voting as well as of possibility of the developed model to drive a free, fair and credible secure e-democratic transition. Considering each of the elicitated security requirements in these responses, findings from the assessment of the authentication security requirement of the developed e-voting model

# 9.RESULTS

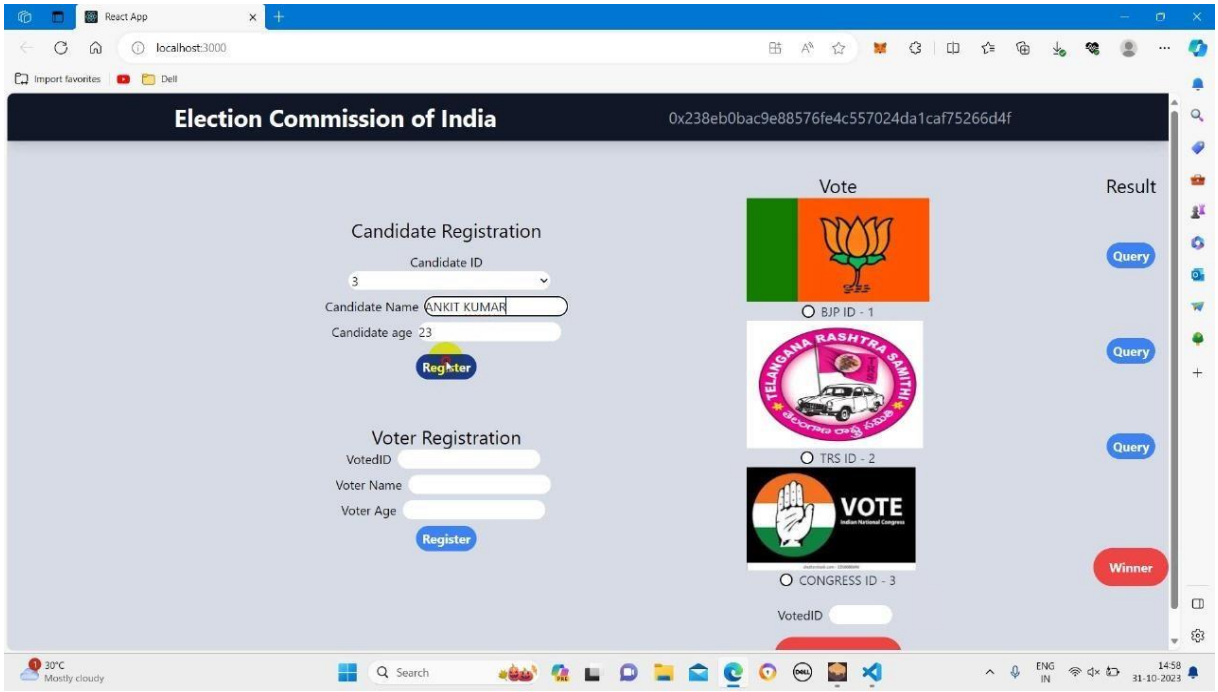
# 9.RESULTS

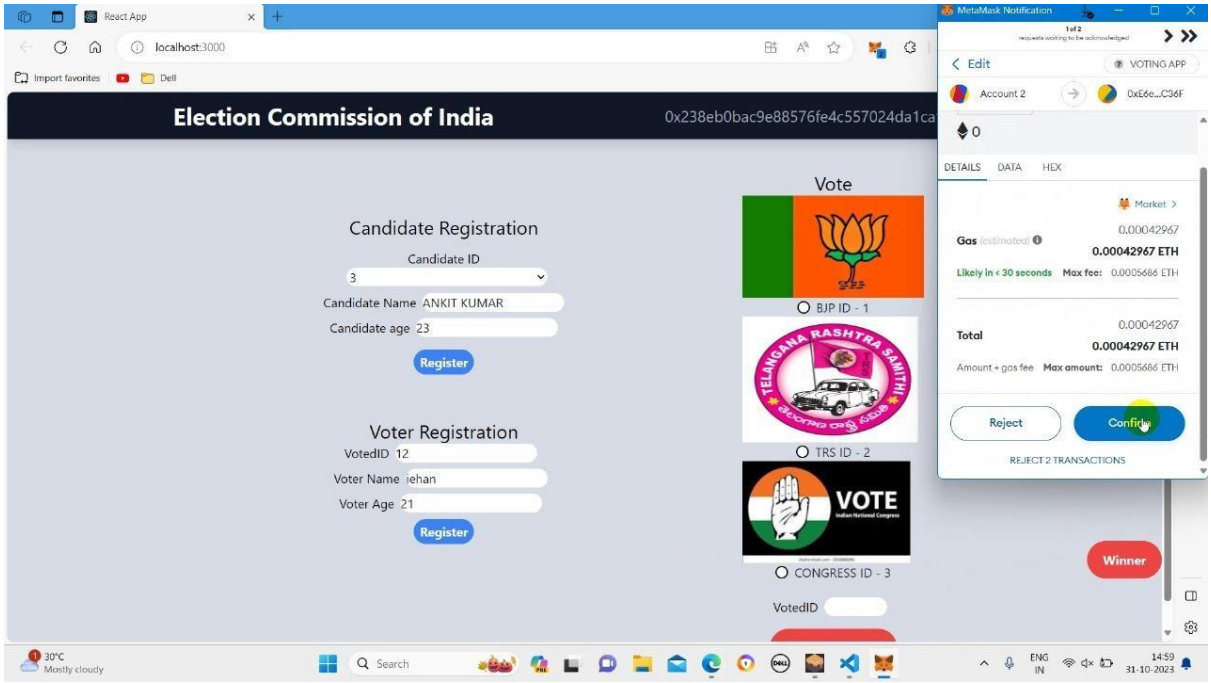
## 9.1 OUTPUT SCREENSHOTS

### INITIALIZING THE ELECTION:

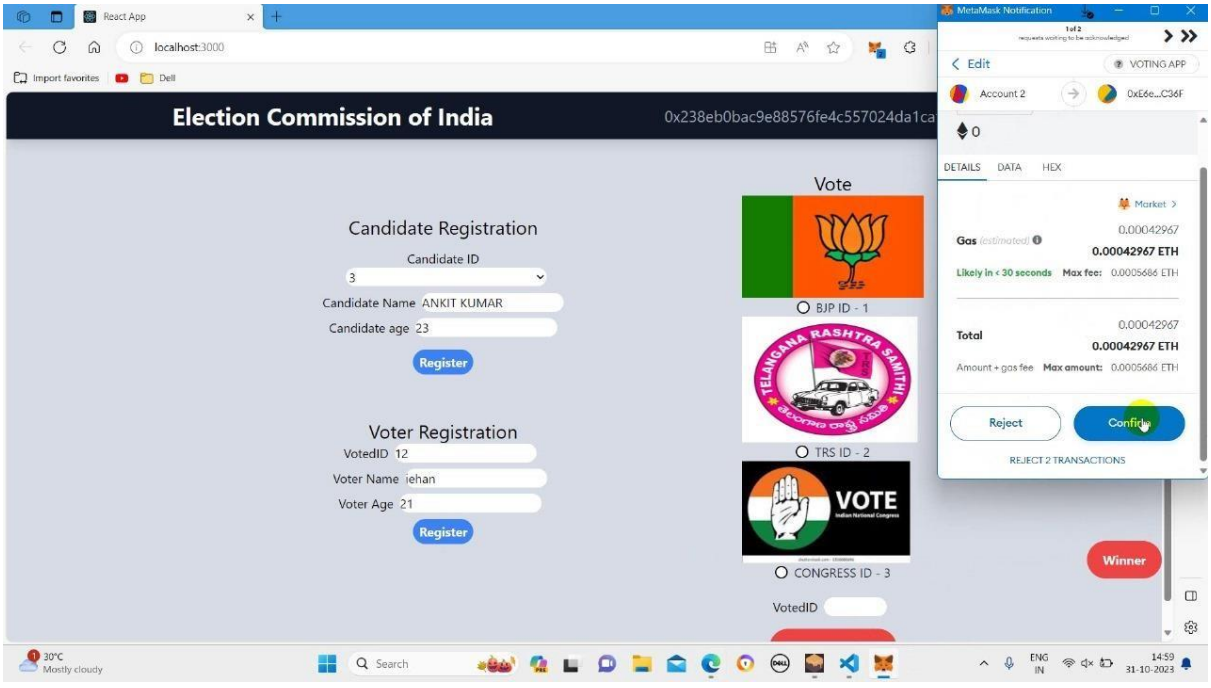


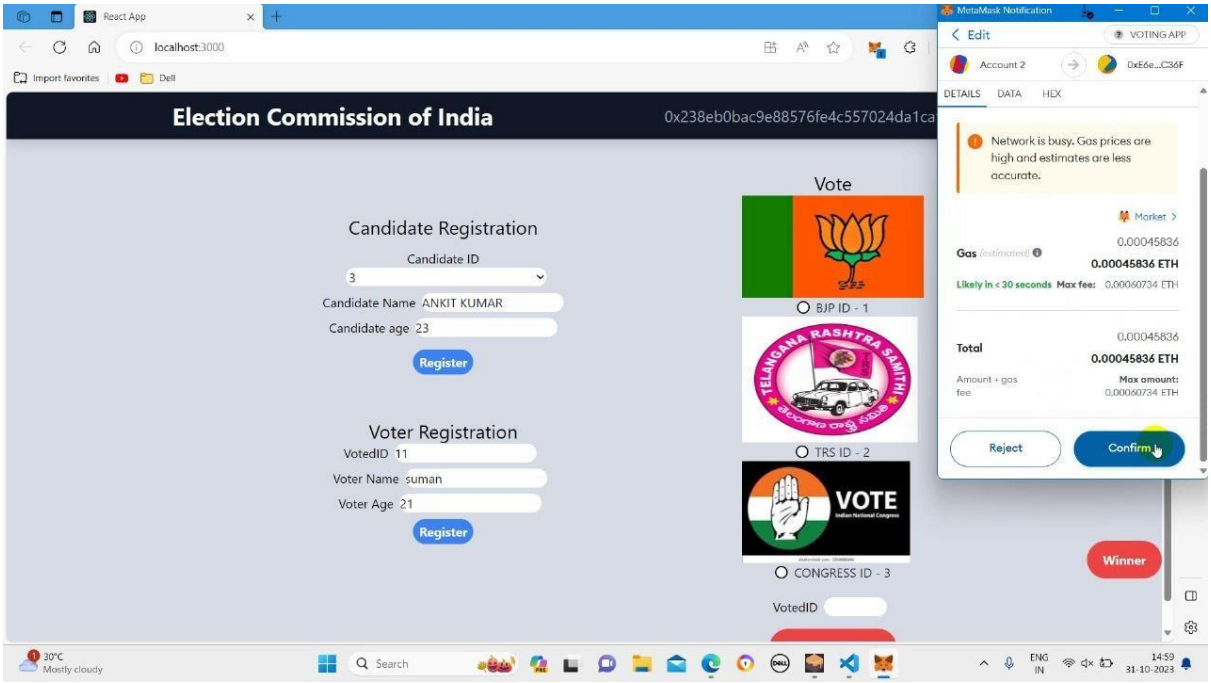

### ADD CANDIDATE:



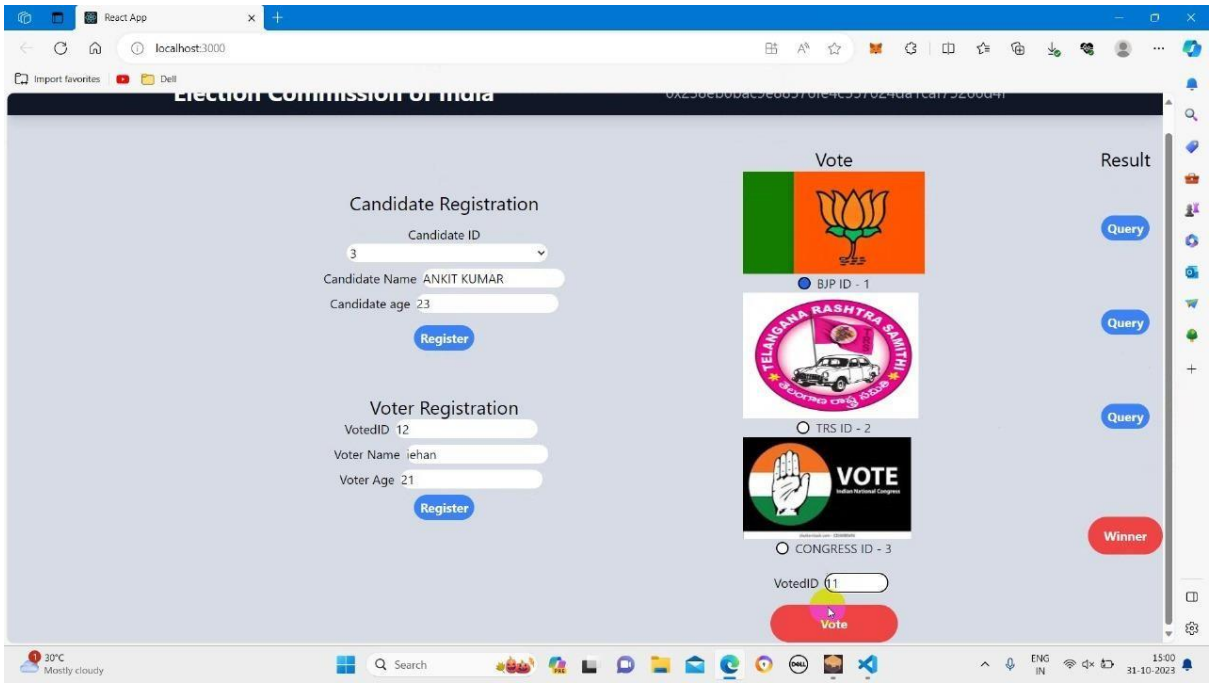



## REGISTER THE VOTER:

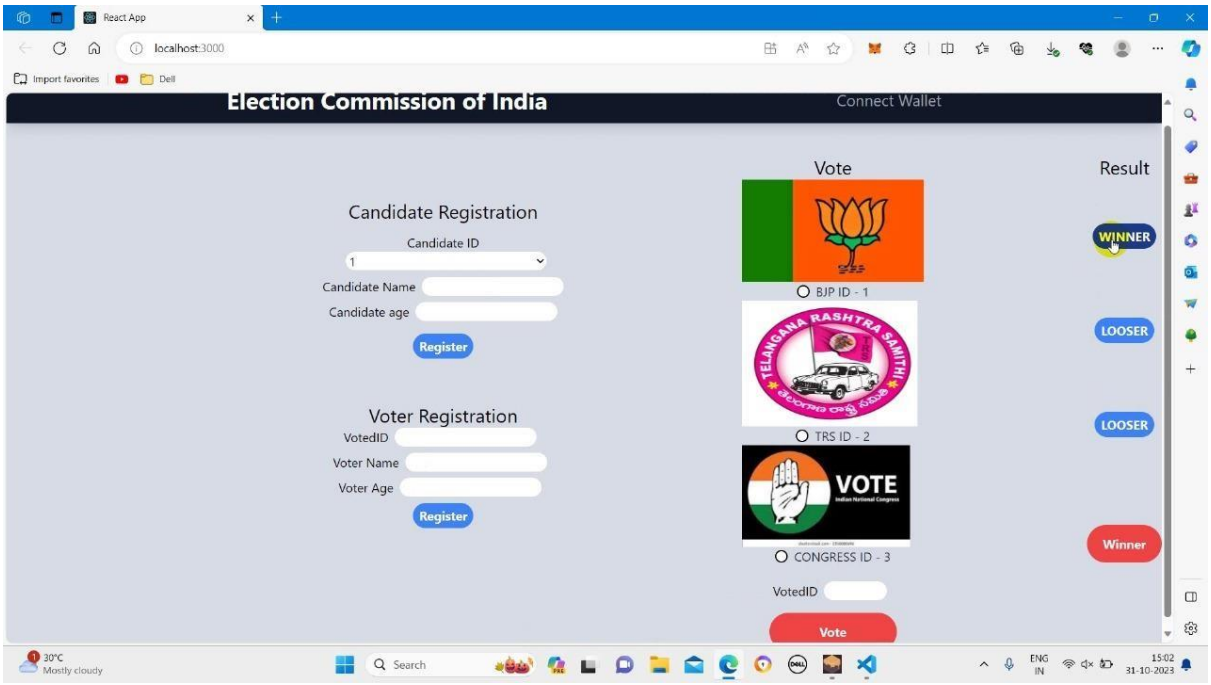





## VOTING BY VOTER:



## ELECTION RESULTS:



# 10.ADVANTAGES & DISADVANTAGES

# 10.ADVANTAGES & DISADVANTAGES

## 10.1 ADVANTAGES

Objective of the VOTING SOFTWARE is to provide better information for the users of this system easily they can vote from anywhere without facing any difficulty.
The proposed system does not require any physical presence during vote polling or counting. So it is very easy to conduct elections even during the pandemic situations without any spread of disease or human live losses.
The proposed system has good authentication so only authorized person can able to vote and also cannot vote multiple types.
Vote Counting can be made very quickly and results will be displayed in few minutes.

## 10.2 DISADVANTAGES

If elections are conducted in existing system model in the pandemic time, then there is sure spread of disease like COVID, which happened in the recent elections in India.

It is difficult to maintain important information in books.

More manual hours are needed for counting of votes.

It is tedious to manage historical data which needs much space to keep all the information regarding the voters and the candidates.

Voters have to wait in long queues for voting they have to travel long distances.

.

# 11.CONCLUSION

# 11. CONCLUSION

This paper presents a blockchain based e-voting system that runs on Ethereum. It shows that blockchain technology can overcome limitations of centralized voting systems. This implementation uses Ethereum blockchain as a network as well as database for storing voter’s accounts, candidate de- tails and votes. This implementation makes use of smart contracts. Blockchain as a technology carries a great future ahead where many real world problems of depending on third party centralized authority in day-to-day life can be resolved, people want an less ambiguous system where everything is crystal clear and at the same time making sure that their(users) data is safe and secure. Voting system using Blockchain will for sure solve all these circumstances faced by people or citizens of a country and will provide them with a system where we no longer they need to depend and follow on to these old aged traditional approaches. World is moving faster and it will move faster in terms of Technology.

# 12. FUTURE SCOPE

# 12. FUTURE ENHANCEMENT

- With the system we currently have, moving the cryptography to a library in Solidity could largely improve our individual ballot verifiability

. • Linking application with Government voting system data.

- The current project is built for small organization, but in future we would build it as a national voting system. • Increase the security from user interface prototype.
- Adding Aadhar number verification system.
- Local Languages can be included which will play a vital role for people living in rural areas
- A feedback system should also be included, that allows the people to file the complaint as well as reviews**.**