MVJ COLLEGE OF ENGINEERING BANGALORE-560067

(An Autonomous Institute)

A MINI PROJECT REPORT

ON

“E-VOTING SYSTEM USING BLOCKCHAIN”

Submitted in practical fulfillment for the award of degree of Bachelor of Engineering

in

INFORMATION SCIENCE AND ENGINEERING

SUBMITTED BY

ANSH SHARMA	**1MJ21IS015**
JISHNUBRATA GHOSH	**1MJ21IS044**
KOUSHIK RAM	**1MJ21IS050**
NAGARJUN P	**1MJ21IS064**

UNDER THE GUIDENCE OF

Prof. Bindu Madhavi.P

Assistant Professor of ISE Department

MVJ College of Engineering, Bangalore
(An Autonomous Institute)

Affiliated to VTU, Belagavi, Approved by AICTE, New Delhi, Recognized
by UGCwith 2(f)& 12(B), Accredited by NBA & NAAC
2023-24

MVJ COLLEGE OF ENGINEERING, BENGALURU-560067

(Autonomous Institution Affiliated to VTU, Belagavi)

DEPARTMENT OF INFORMATION AND SCIENCE ENGINEERING

CERTIFICATE

Certified that the mini project work titled **‘***E*-Voting System Using Blockchain**’** is carried out by **ANSH SHARMA(1MJ21IS015), JISHNUBRATA GHOSH (1MJ21IS044) , KOUSHIK RAM (1MJ21IS050) , NAGARJUN P (1MJ21IS064)** who is confide students of MVJ College of Engineering, Bengaluru, in partial fulfilment for the award of Degree of **Bachelor of Engineering in Information Science and Engineering** of the Visvesvaraya Technological University, Belagavi during the year 2023- 2024. It is certified that all corrections/suggestions indicated for the Internal Assessment have been incorporated in the mini project report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of mini project work prescribed by the institution for the said Degree.

Signature of Guide	**Signature of Head of the Department**	**Signature of Principal**
Prof. BINDU MADHAVI.P	**Dr. SHIMA RAMESH**	**Dr. V SURESH BABU**

External Viva

Name of Examiners **Signature with Date**

1.

2.

MVJ COLLEGE OF ENGINEERING, BENGALURU-560067

(Autonomous Institution Affiliated to VTU, Belagavi)

DEPARTMENT OF INFORMATION AND SCIENCE ENGINEERING

DECLARATION

We, **ANSH SHARMA, JISHNUBRATA GHOSH, KOUSHIK RAM, NAGARJUN** students of sixth semester B.E., Department of Information Science And Engineering, MVJ College of Engineering, Bengaluru, hereby declare that the mini project titled ***'E***-Voting System Using Blockchain***'*** has been carried out by us and submitted in partial fulfilment for the award of Degree of **Bachelor of Engineering** in **Information Science And Engineering** during the year 2023-2024.

Further we declare that the content of the dissertation has not been submitted previously by anybody for the award of any Degree or Diploma to any other University.

We also declare that any Intellectual Property Rights generated out of this project carried out at MVJCE will be the property of MVJ College of Engineering, Bengaluru and we will be one of the authors of the same.

Place: Bengaluru

Date:

Name	**Signature**
ANSH SHARMA(1MJ21IS015)	
JISHNUBRATA GHOSH(1MJ21IS044)	
KOUSHIK RAM(1MJ21IS050)	
NAGARJUN P(1MJ21IS064)	

ACKNOWLEDGEMENT

We are indebted to our guide**, Prof. BINDU MADHAVI.P , Dept of Information Science And Engineering, MVJ College of Engineering** for his wholehearted support, suggestions and invaluable advice throughout our project work and also for the help in the preparation of this report.

We also express our gratitude to our panel members **,** Associate Professors, and Assistant Professors, Departmen**t** of Information And Science Engineering, for their valuable comments and suggestions.

Our sincere thanks to **Dr. SHIMA RAMESH** , Associate Professor and Head, Department of Information And Science Engineering, MVJCE for his support and encouragement.

We express sincere gratitude to our beloved Principal, **Dr. V SURESH BABU** for all his support towards this project work.

Lastly, we take this opportunity to thank our **family** members and **friends** who provided all the backup support throughout the project work.

ABSTRACT

Voting is a fundamental part of democratic systems. It gives individuals in a community 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. 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,efficiency.

TABLE OF CONTENTS

List Of Figures

CHAPTER-1

INTRODUCTION

INTRODUCTION

1.1 Introduction

Internet is the greatest thing invented by 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.

1.2 Problem Statement

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

1.3 Objectives

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 short comings of the existing systems.

CHAPTER-2

THEORY AND CONCEPT

CHAPTER 2

THEORY AND CONCEPT

2.1 Purpose

The purpose of the **Online Examination System** is to modernize and improve the traditional examination process by leveraging digital technologies to create a more efficient, accessible, and user-friendly platform. This system aims to:

1. **Enhanced Learning Experience:** Provide structured tutorials that cater to various educational needs, enriching the learning process.

2. **Comprehensive Practice Opportunities:** Allow students to assess their knowledge through mock exams, ensuring they are well-prepared for final assessments.

3. **Streamlined Examination Process:** Implement an auto-submission feature that ensures timely and accurate completion of exams without manual intervention.

4. **Increased Student Engagement:** Utilize gamification elements to boost motivation and reward progress and performance.

5. **Immediate Certification:** Provide personalized certificates upon successful exam completion, fostering a sense of accomplishment among students.

6. **Centralized Management:** Serve as a centralized system for managing content delivery, examination processes, and result generation.

2.2 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 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 familiarise ourselves with current implementations, particularly Estonia.

2.3 Proposed System

The proposal is 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 possess 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. 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’sBlockchain, 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 whichthe Election Commission fetches the vote count. The limitation faced due to not using the traditional way of smart contracts is that the blockchain framework which are coded cannot run on the main net as it needs to be hosted and a separate web3 provider have to be used for interacting with it and not having a public API of voter ID creates a drawback of not havingauthentication 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 physicalform.

- Non-Resident Indian can cast their votes as it is totally online.

CHAPTER-03

METHODOLOGY

CHAPTER 03

METHODOLOGY

3.1 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 subsystems 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/ communitywho 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 Ci

•**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 Vi 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.

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 actually 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.

CHAPTER-4
SYSTEM DESIGN

CHAPTER 4

SYSTEM DESIGN

Design is a process through which requirements are translated into a representation of the software. The purpose of the designing phase is to plan a solution for the problem specified by the requirement document. This phase moves from the problem domain to the solution domain i.e. the requirements are translated into software.

The design activity often results in three separate outputs:

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

Software Requirements

- Operating System : Windows 7 (32 bit and 64 bit) and Above
- Development Environment : Solidity Programming, Web Development (CSS, HTML,JAVASCRIPT)
- Scripting Language : Solidity Proramming
- Decentralized Applications :Ethereum Framework ie. Truffle and Ganache
- Browser : Google Chrome
- Add-on in Browser :Metamask

4.1FLOW CHART:

A flowchart is a picture of the separate steps of a process in sequential order. It is ageneric tool that can be adapted for a wide variety of purposes, and can be used to describe various processes, such as a manufacturing process, an administrative or service process, or a project plan.

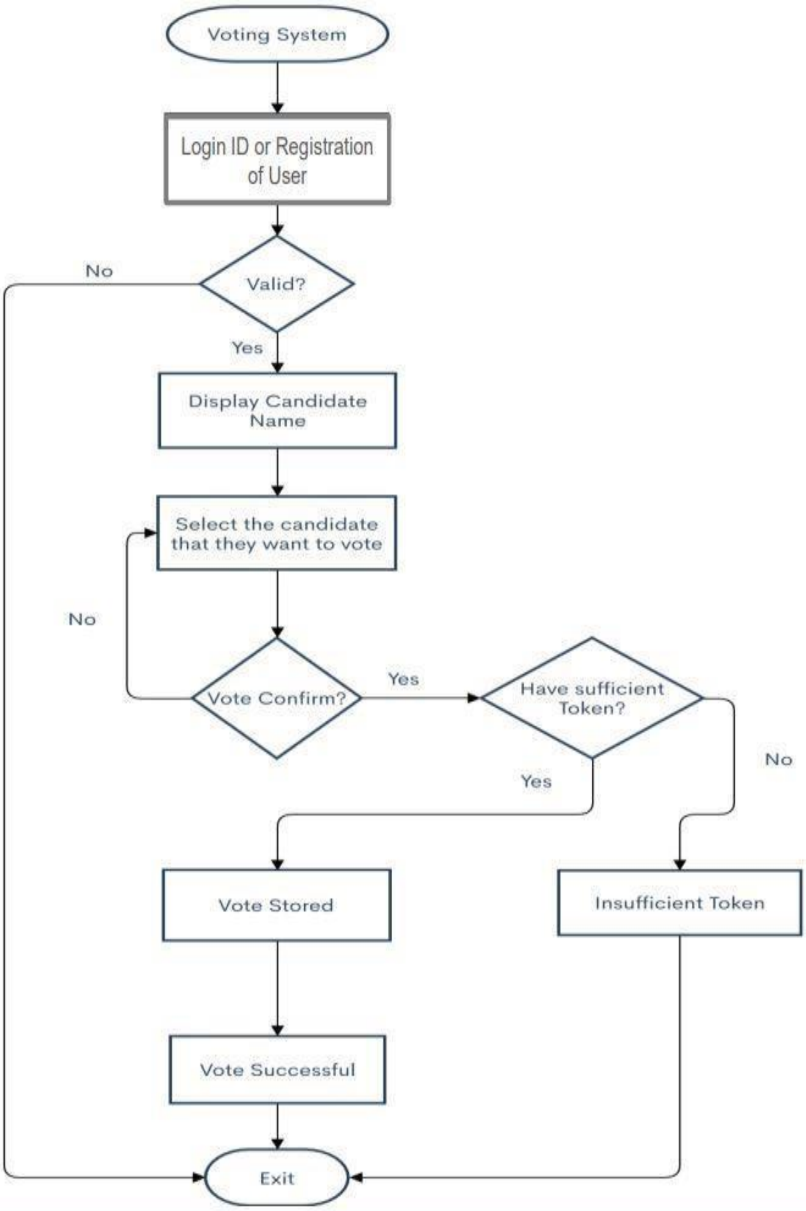


Figure 4.1 Flow Chart

4.2 ENTITY-RELATIONSHIP DIAGRAM (ER DIAGRAM):

An entity-relationship (ER) diagram is a specialized graphic that illustrates the relationships between entities in a database. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities.

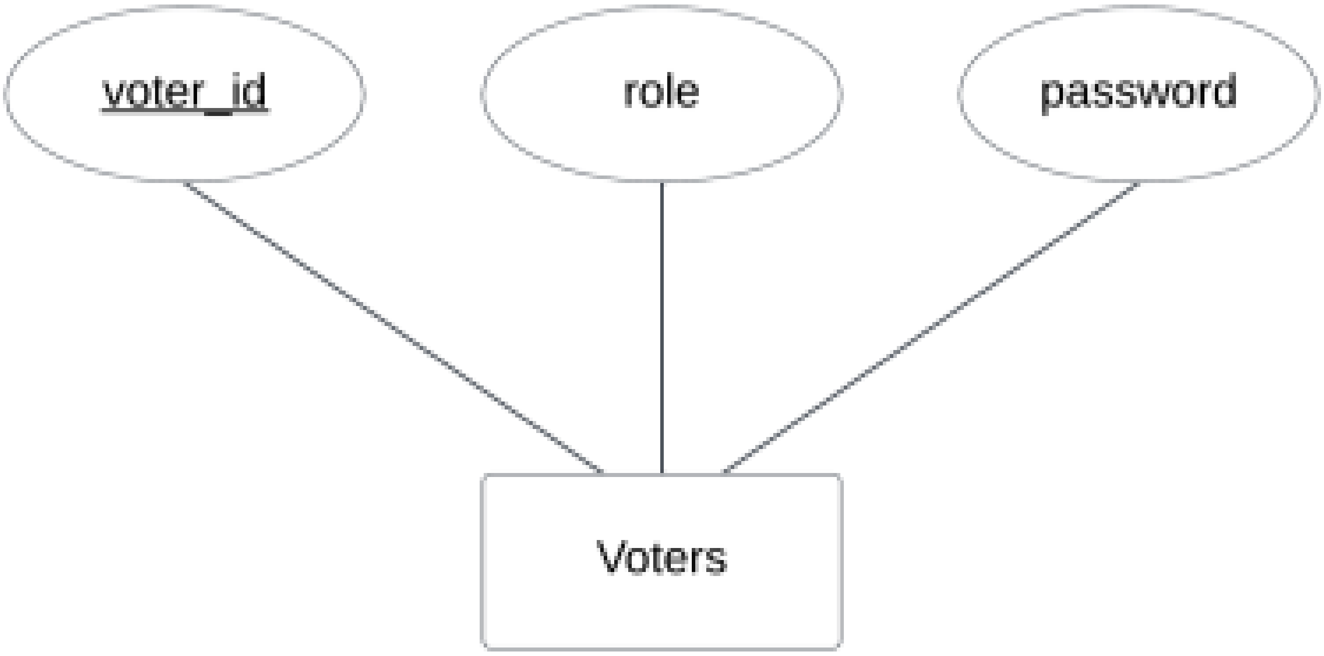


Figure 4.2 ER Diagram

4.3 CLASS DIAGRAM:

A schema contains schema objects, which could be tables, columns, data types, views, stored procedures, relationships, primary keys, foreign keys, etc. A database schema can be represented in a visual diagram, which shows the database objects and their relationship with each other.

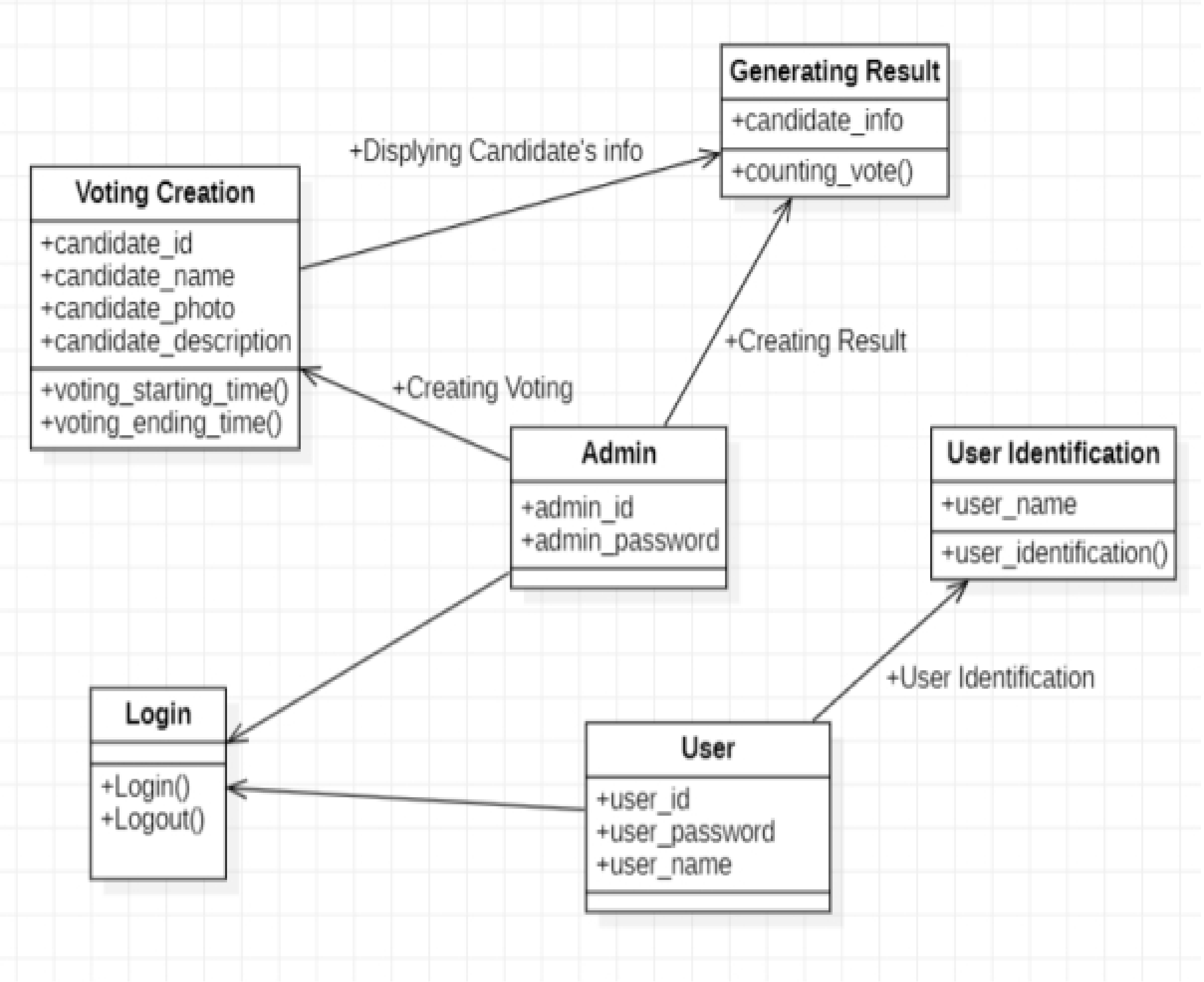


Figure 4.3 Class Diagram

4.4 Use Case Diagram

A use case is an interaction between users and a system.It captures the goal of the users and the responsibility the system to its users. Its is the functionality of the system or the service provided by the system

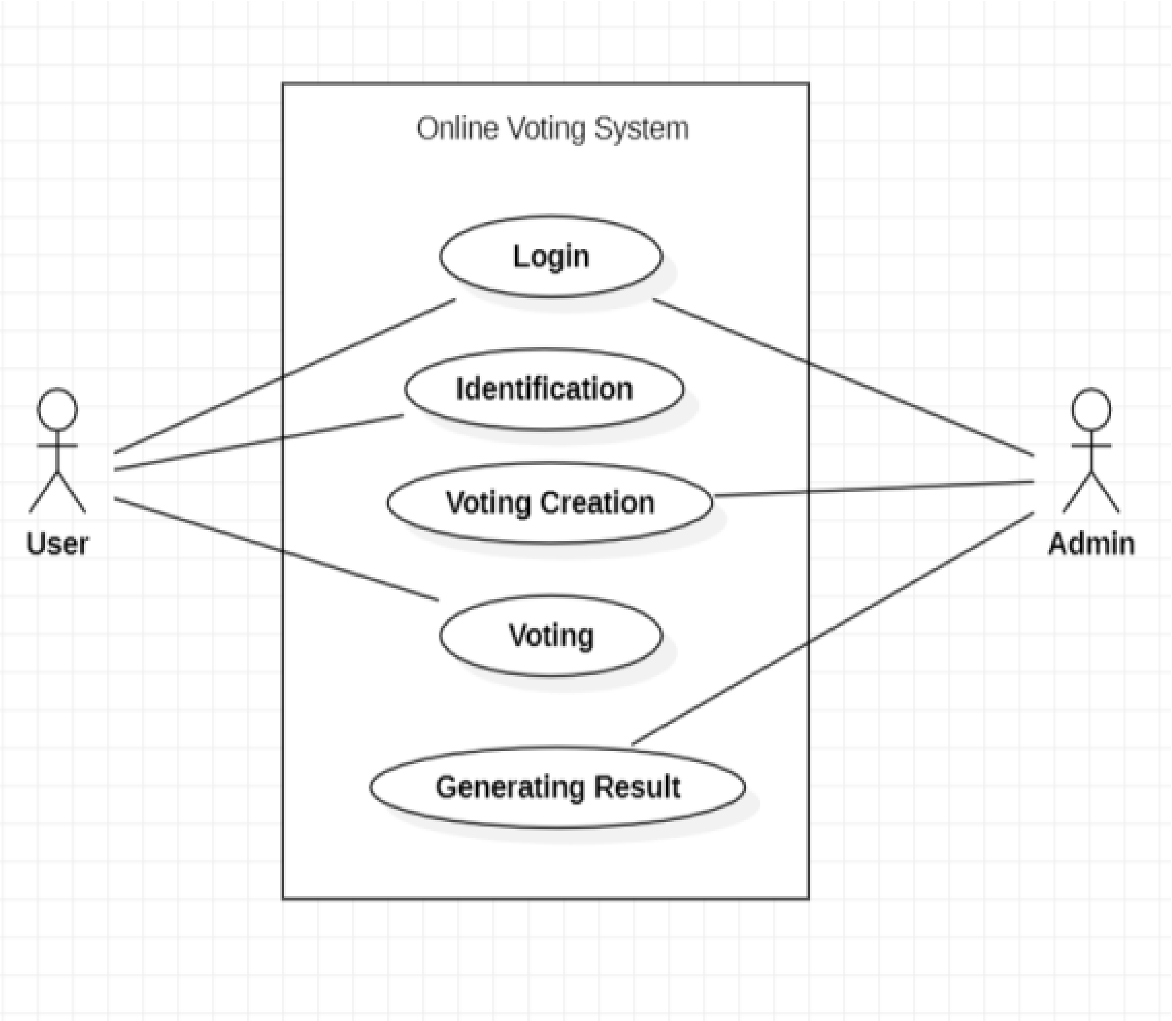


Figure 4.4 Use Case Daigram

CHAPTER-5

IMPLEMENTATION

CHAPTER 5

IMPLEMENTATION

5.1 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.

Create Ballot

If the user has permissions to create a ballot, the user is able to spawn a new voting contract by entering the required information in E-vote.html. 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

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 VotingApp Java Script sends an eth.call to the Registrar Contract to determine the validity of the ballot ID.

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.

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

5.2 Technologies used for Implementation

5.2.1 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.

5.2.2 Cascading Style Sheets

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation ofa document written in a markup language.

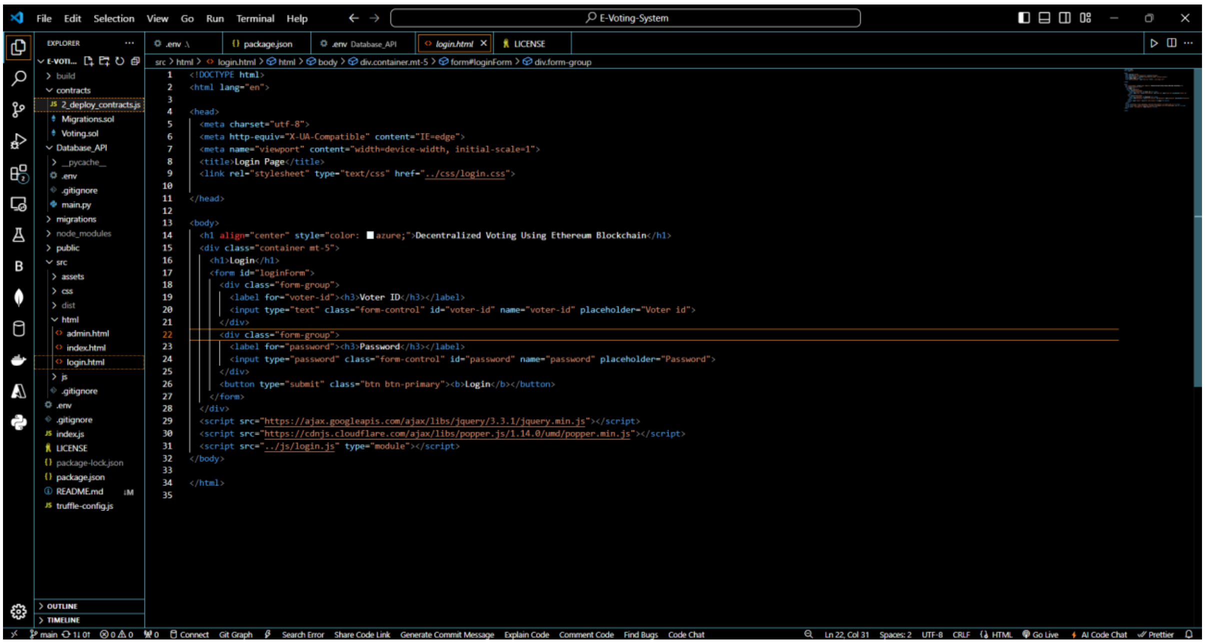
JavaScript

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.

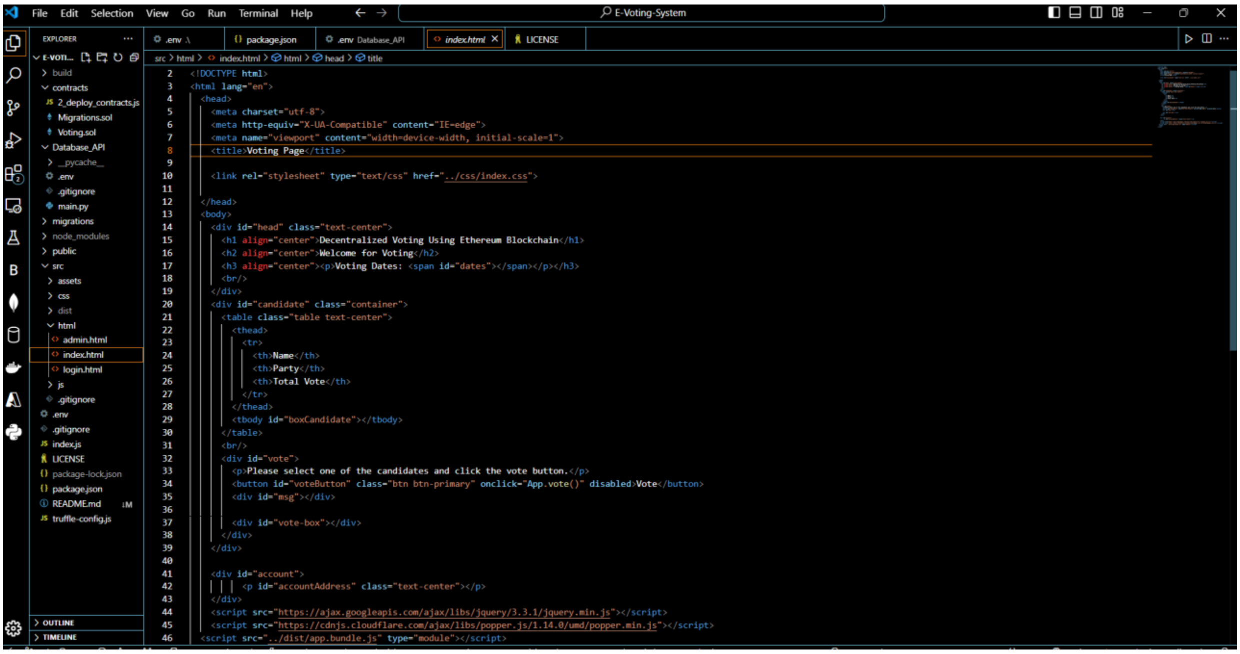
5.3 Smart Contract Concept

“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. any steps. Most smart contracts are written in one of the programming languages directly suited for such computer programs, such as Solidity.

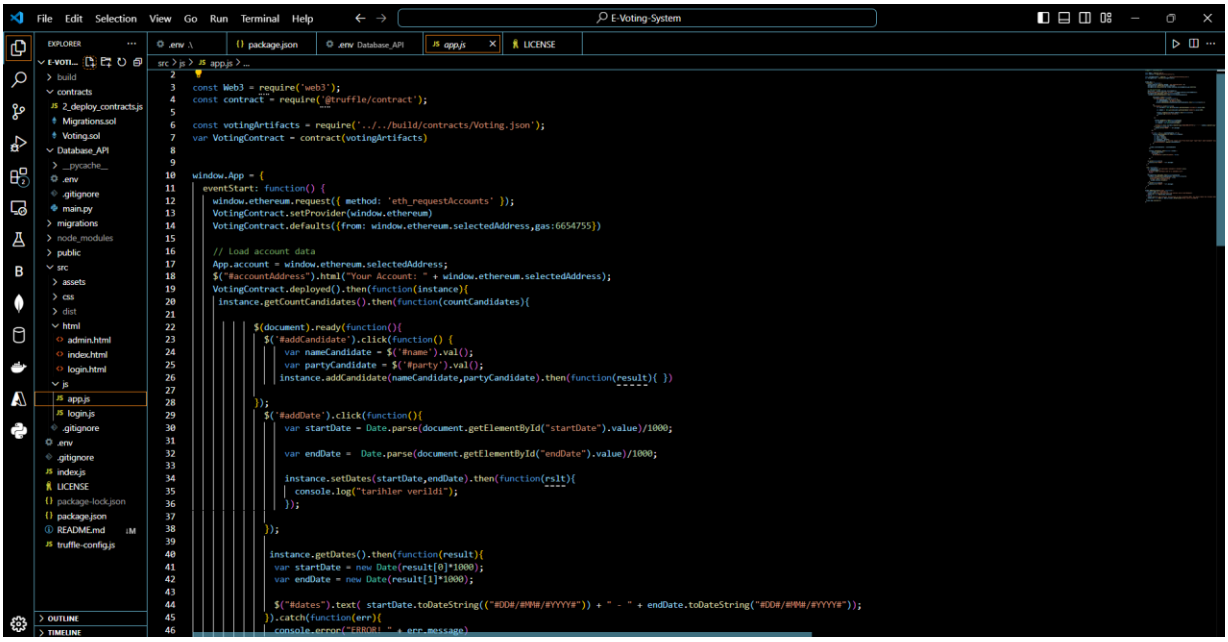
CODE SNIPPETS



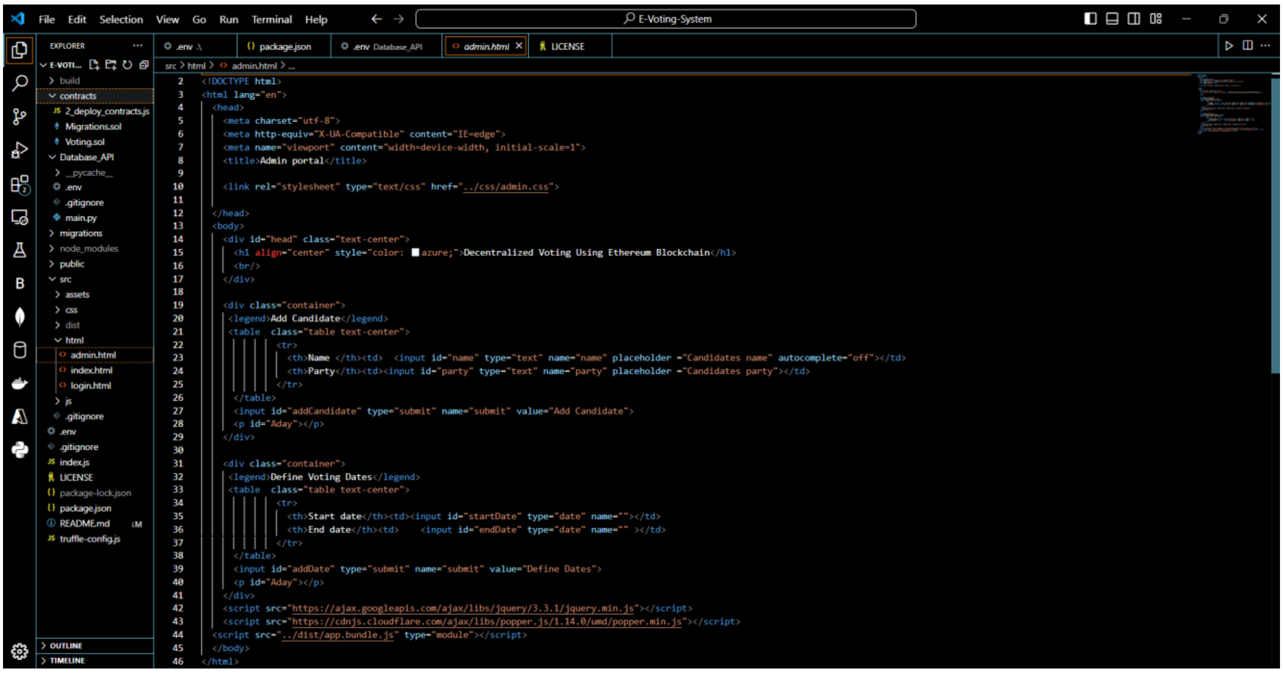
```html
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Login Page</title>
  <link rel="stylesheet" type="text/css" href="../css/login.css">

</head>

<body>
  <h1 align="center" style="color: azure;">Decentralized Voting Using Ethereum Blockchain</h1>
  <div class="container mt-5">
    <h1>Login</h1>
    <form id="loginForm">
      <div class="form-group">
        <label for="voter-id"><h3>Voter ID</h3></label>
        <input type="text" class="form-control" id="voter-id" name="voter-id" placeholder="Voter id">
      </div>
      <div class="form-group">
        <label for="password"><h3>Password</h3></label>
        <input type="password" class="form-control" id="password" name="password" placeholder="Password">
      </div>
      <button type="submit" class="btn btn-primary"><b>Login</b></button>
    </form>
  </div>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.0/umd/popper.min.js"></script>
  <script src="../js/login.js" type="module"></script>
</body>

</html>
```



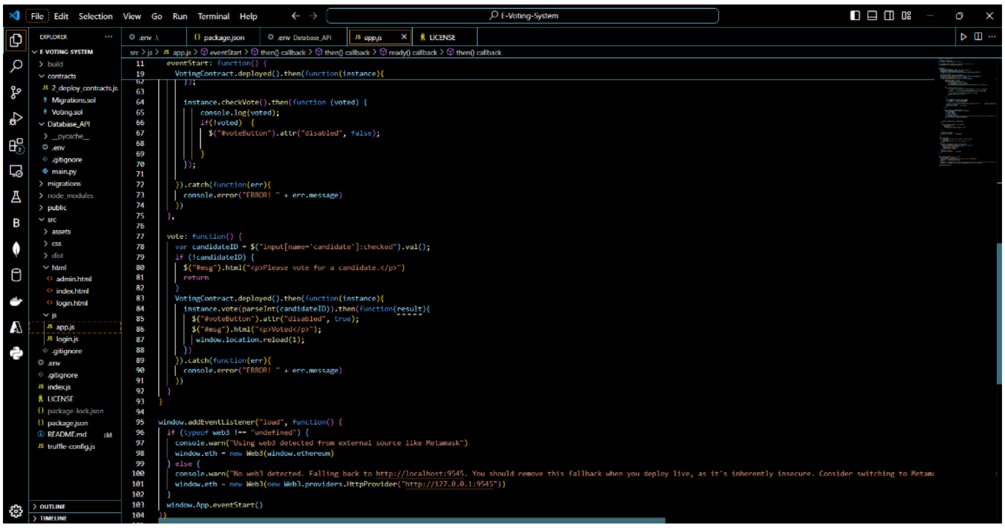
```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Voting Page</title>

    <link rel="stylesheet" type="text/css" href="../css/index.css">

  </head>
  <body>
    <div id="head" class="text-center">
      <h1 align="center">Decentralized Voting Using Ethereum Blockchain</h1>
      <h2 align="center">Welcome for Voting</h2>
      <h3 align="center"><p>Voting Dates: <span id="dates"></span></p></h3>
      <br/>
    </div>
    <div id="candidate" class="container">
      <table class="table text-center">
        <thead>
          <tr>
            <th>Name</th>
            <th>Party</th>
            <th>Total Vote</th>
          </tr>
        </thead>
        <tbody id="boxCandidate"></tbody>
      </table>
      <br/>
      <div id="vote">
        <p>Please select one of the candidates and click the vote button.</p>
        <button id="voteButton" class="btn btn-primary" onclick="App.vote()" disabled>Vote</button>
        <div id="msg"></div>

        <div id="vote-box"></div>
      </div>
    </div>

    <div id="account">
        <p id="accountAddress" class="text-center"></p>
    </div>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.0/umd/popper.min.js"></script>
  <script src="../dist/app.bundle.js" type="module"></script>
```



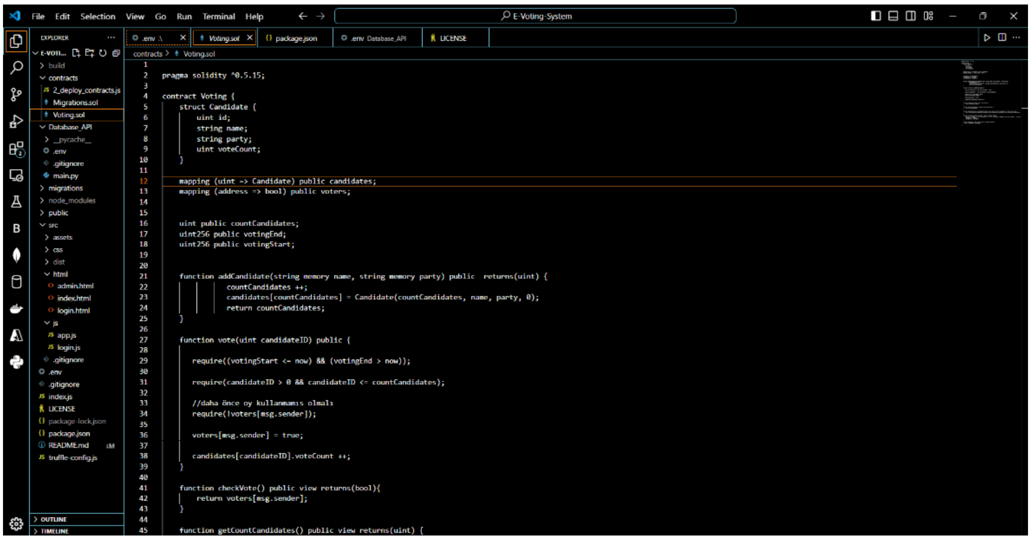
```javascript

const Web3 = require('web3');
const contract = require('@truffle/contract');

const votingArtifacts = require('../../build/contracts/Voting.json');
var VotingContract = contract(votingArtifacts)


window.App = {
  eventStart: function() {
    window.ethereum.request({ method: 'eth_requestAccounts' });
    VotingContract.setProvider(window.ethereum)
    VotingContract.defaults({from: window.ethereum.selectedAddress,gas:6654755})

    // Load account data
    App.account = window.ethereum.selectedAddress;
    $("#accountAddress").html("Your Account: " + window.ethereum.selectedAddress);
    VotingContract.deployed().then(function(instance){
     instance.getCountCandidates().then(function(countCandidates){

          $(document).ready(function(){
            $('#addCandidate').click(function() {
                var nameCandidate = $('#name').val();
                var partyCandidate = $('#party').val();
              instance.addCandidate(nameCandidate,partyCandidate).then(function(result){ })

          });
            $('#addDate').click(function(){
                var startDate = Date.parse(document.getElementById("startDate").value)/1000;

                var endDate =  Date.parse(document.getElementById("endDate").value)/1000;

                instance.setDates(startDate,endDate).then(function(rslt){
                  console.log("tarihler verildi");
                });

            });

             instance.getDates().then(function(result){
              var startDate = new Date(result[0]*1000);
              var endDate = new Date(result[1]*1000);

              $("#dates").text( startDate.toDateString(("#DD#/#MM#/#YYYY#")) + " - " + endDate.toDateString("#DD#/#MM#/#YYYY#"));
            }).catch(function(err){
              console.error("ERROR! " + err.message)
```



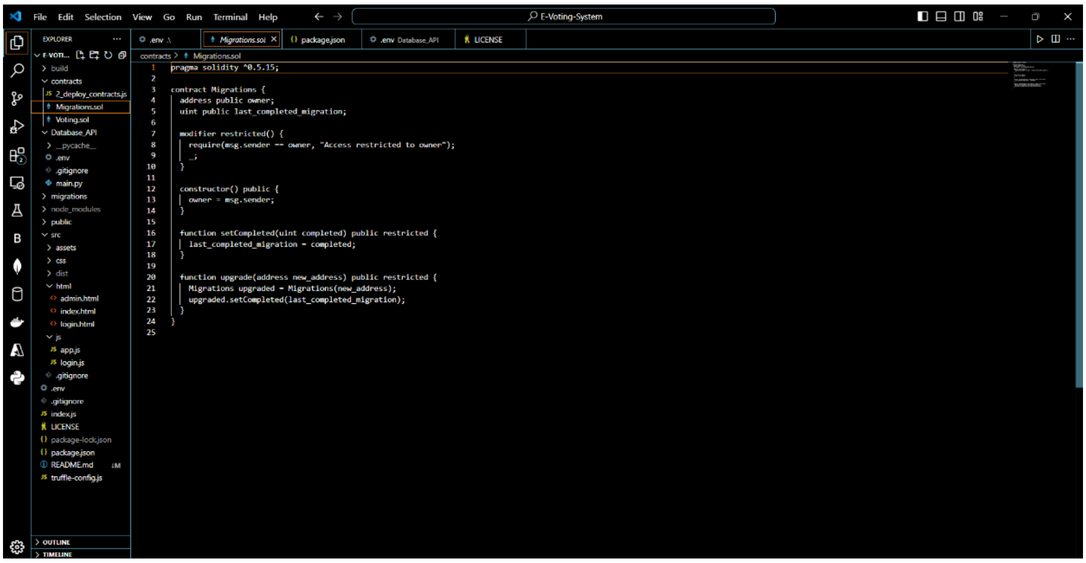
```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Admin portal</title>

    <link rel="stylesheet" type="text/css" href="../css/admin.css">

  </head>
  <body>
    <div id="head" class="text-center">
      <h1 align="center" style="color: azure;">Decentralized Voting Using Ethereum Blockchain</h1>
      <br/>
    </div>

    <div class="container">
     <legend>Add Candidate</legend>
     <table  class="table text-center">
            <tr>
              <th>Name </th><td>  <input id="name" type="text" name="name" placeholder ="Candidates name" autocomplete="off"></td>
              <th>Party</th><td><input id="party" type="text" name="party" placeholder ="Candidates party"></td>
            </tr>
      </table>
      <input id="addCandidate" type="submit" name="submit" value="Add Candidate">
      <p id="Aday"></p>
    </div>

    <div class="container">
     <legend>Define Voting Dates</legend>
     <table  class="table text-center">
            <tr>
              <th>Start date</th><td><input id="startDate" type="date" name=""></td>
              <th>End date</th><td>    <input id="endDate" type="date" name="" ></td>
            </tr>
      </table>
      <input id="addDate" type="submit" name="submit" value="Define Dates">
      <p id="Aday"></p>
    </div>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.0/umd/popper.min.js"></script>
  <script src="../dist/app.bundle.js" type="module"></script>
  </body>
</html>
```


File Edit Selection View Go Run Terminal Help
E-Voting-System
.env .\
package.json
.env Database_API
app.js
LICENSE
src > js > app.js > eventStart > then() callback > then() callback > ready() callback > then() callback
EXPLORER
E-VOTING-SYSTEM
OUTLINE
TIMELINE

```
  eventStart: function() {
    VotingContract.deployed().then(function(instance){
      });

      instance.checkVote().then(function (voted) {
          console.log(voted);
          if(!voted)  {
            $("#voteButton").attr("disabled", false);

          }
      });

    }).catch(function(err){
      console.error("ERROR! " + err.message)
    })
  },

  vote: function() {
    var candidateID = $("input[name='candidate']:checked").val();
    if (!candidateID) {
      $("#msg").html("<p>Please vote for a candidate.</p>")
      return
    }
    VotingContract.deployed().then(function(instance){
      instance.vote(parseInt(candidateID)).then(function(result){
        $("#voteButton").attr("disabled", true);
        $("#msg").html("<p>Voted</p>");
         window.location.reload(1);
      })
    }).catch(function(err){
      console.error("ERROR! " + err.message)
    })
  }
}

window.addEventListener("load", function() {
  if (typeof web3 !== "undefined") {
    console.warn("Using web3 detected from external source like Metamask")
    window.eth = new Web3(window.ethereum)
  } else {
    console.warn("No web3 detected. Falling back to http://localhost:9545. You should remove this fallback when you deploy live, as it's inherently insecure. Consider switching to Metam
    window.eth = new Web3(new Web3.providers.HttpProvider("http://127.0.0.1:9545"))
  }
  window.App.eventStart()
})
```



```

pragma solidity ^0.5.15;

contract Voting {
    struct Candidate {
        uint id;
        string name;
        string party;
        uint voteCount;
    }

    mapping (uint => Candidate) public candidates;
    mapping (address => bool) public voters;


    uint public countCandidates;
    uint256 public votingEnd;
    uint256 public votingStart;


    function addCandidate(string memory name, string memory party) public  returns(uint) {
                countCandidates ++;
                candidates[countCandidates] = Candidate(countCandidates, name, party, 0);
                return countCandidates;
    }

    function vote(uint candidateID) public {

       require((votingStart <= now) && (votingEnd > now));

       require(candidateID > 0 && candidateID <= countCandidates);

       //daha önce oy kullanmamıs olmalı
       require(!voters[msg.sender]);

       voters[msg.sender] = true;

       candidates[candidateID].voteCount ++;
    }

    function checkVote() public view returns(bool){
        return voters[msg.sender];
    }

    function getCountCandidates() public view returns(uint) {
```



```
pragma solidity ^0.5.15;

contract Migrations {
  address public owner;
  uint public last_completed_migration;

  modifier restricted() {
    require(msg.sender == owner, "Access restricted to owner");
    _;
  }

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

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

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

```

CHAPTER-6

TESTING

CHAPTER 6

TESTING

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. It includes a set of techniques and methods to identify defects, bugs, performance issues and providing a reliable and quality product

6. Types of Testing

6.1 Unit Testing

Unit testing is a type of testing that is used to evaluate the individual units or components of a software system. This type of testing helps ensure that each unit or component of the system is working correctly and is able to perform its intended function.

6.2 Integration Testing

Integration testing is a type of testing that is used to evaluate how well the different units or components of a software system work together. This type of testing helps to identify and resolve issues related to compatibility, performance, and data flow between the different units or components.

6.3 Functional Testing

Functional testing is a type of testing that is used to evaluate how well a system or software performs the specific functions or tasks that it is designed to perform. It is done by testing the system or software with various inputs and verifying that the outputs are correct. This type of testing ensures that the system or software is working as intended and is able to perform the functions it was designed to perform.

6.4 White Box Testing

White box testing, also known as structural testing or glass-box testing, is a type of testing that examines the internal structure and implementation of a software system. It involves testing the code itself and checking that it is functioning correctly and adhering to coding standards. This type of testing helps to identify and resolve issues related to logic, control flow, and data structures within the system.

6.5 Black Box Testing

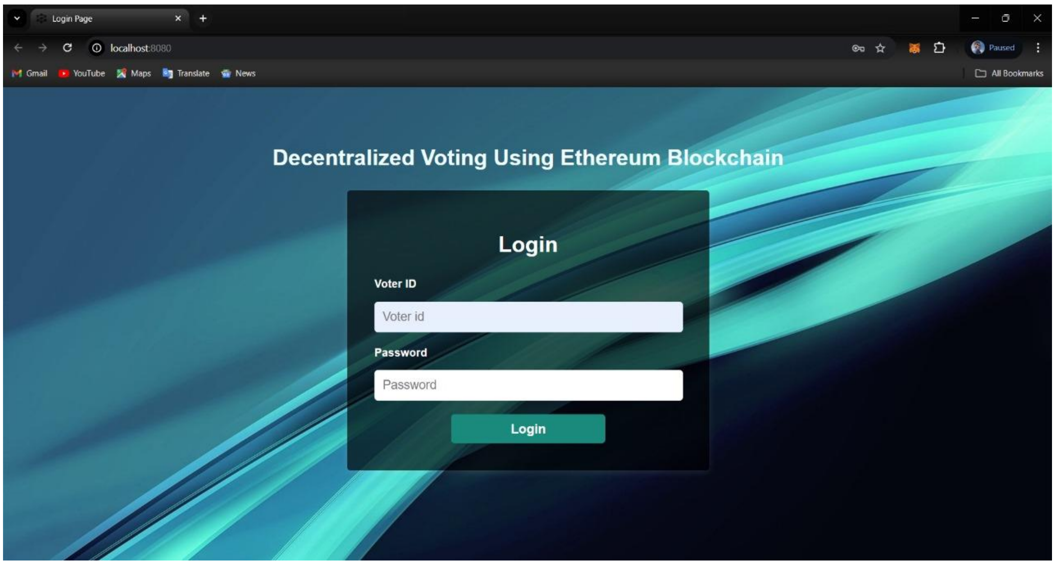
Black box testing, also known as functional testing, is a type of testing that examines the external behavior and interfaces of a software system. It involves testing the system from the user's perspective, without looking at the internal structure or implementation,and checking that it is functioning correctly and meeting the requirements.

CHAPTER-7

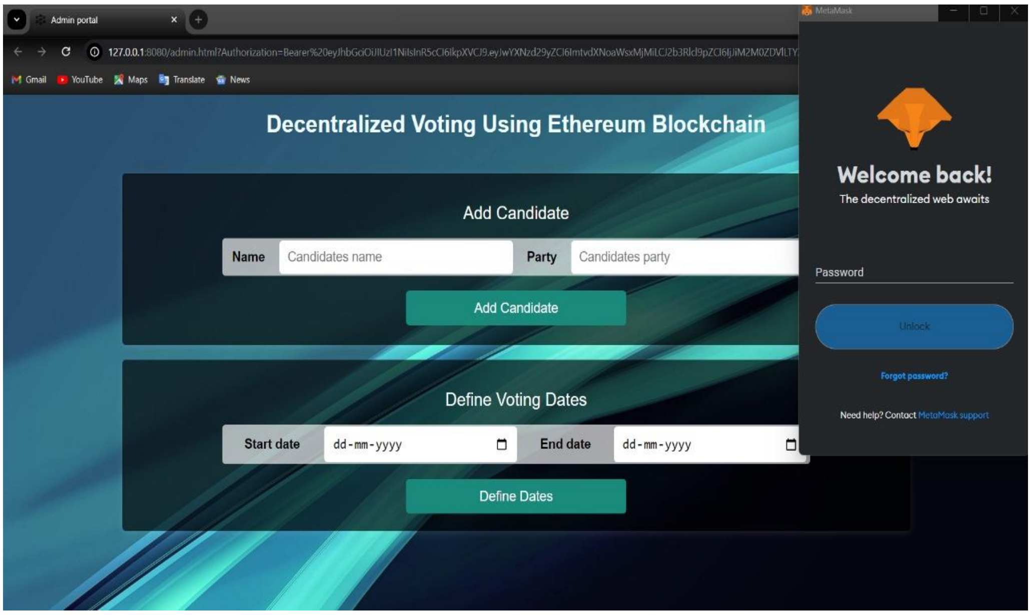
SCREENSHOTS

CHAPTER 7

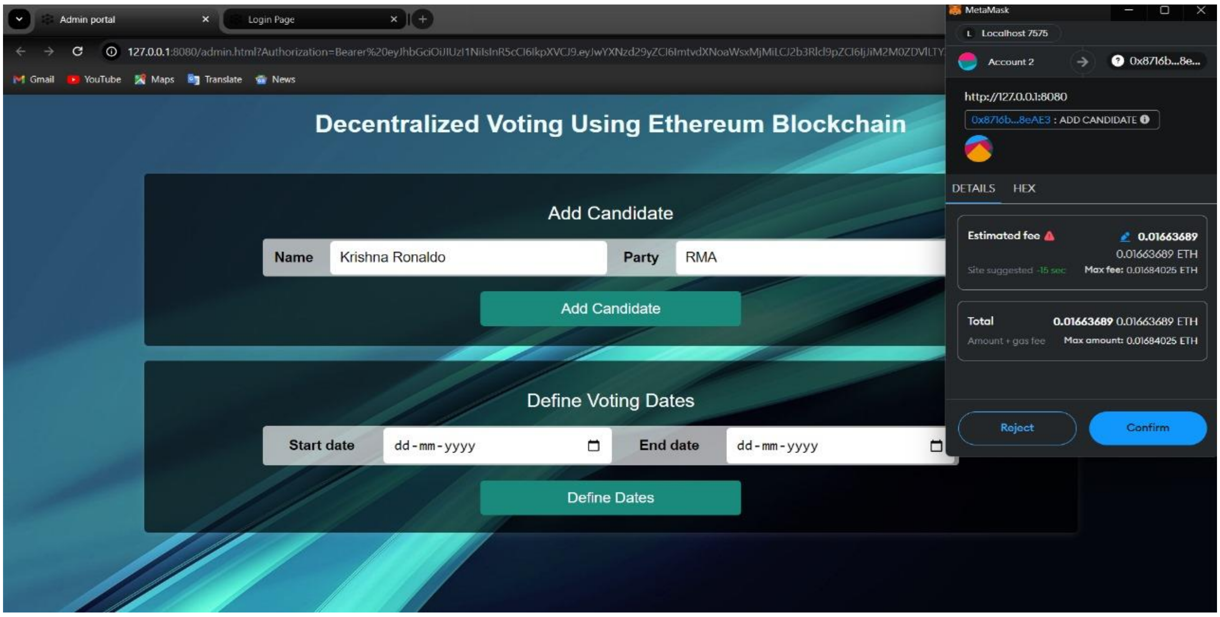
SCREENSHOTS



7.1 LOGIN PAGE


Admin portal
127.0.0.1:8080/admin.html?Authorization=Bearer%20eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJwYXNzd29yZCI6ImtvdXNoaWsxMjMiLCJ2b3Rlcl9pZCI6IjJiM2M0ZDVlLTY
Gmail
YouTube
Maps
Translate
News
Decentralized Voting Using Ethereum Blockchain
Add Candidate
Name
Candidates name
Party
Candidates party
Add Candidate
Define Voting Dates
Start date
dd-mm-yyyy
End date
dd-mm-yyyy
Define Dates
MetaMask
Welcome back!
The decentralized web awaits
Password
Unlock
Forgot password?
Need help? Contact MetaMask support

7.2 ADMIN PAGE


Admin portal
Login Page
127.0.0.1:8080/admin.html?Authorization=Bearer%20eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJwYXNzd29yZCI6ImtvdXNoaWsxMjMiLCJ2b3RlcklkIjoiJiM2MOZDVILTY
Gmail
YouTube
Maps
Translate
News
MetaMask

Decentralized Voting Using Ethereum Blockchain

Add Candidate

Name: Krishna Ronaldo | Party: RMA

Add Candidate

Define Voting Dates

Start date: dd-mm-yyyy | End date: dd-mm-yyyy

Define Dates

Localhost 7575

Account 2 → 0x8716b...8e...

http://127.0.0.1:8080

0x8716b...8eAE3 : ADD CANDIDATE

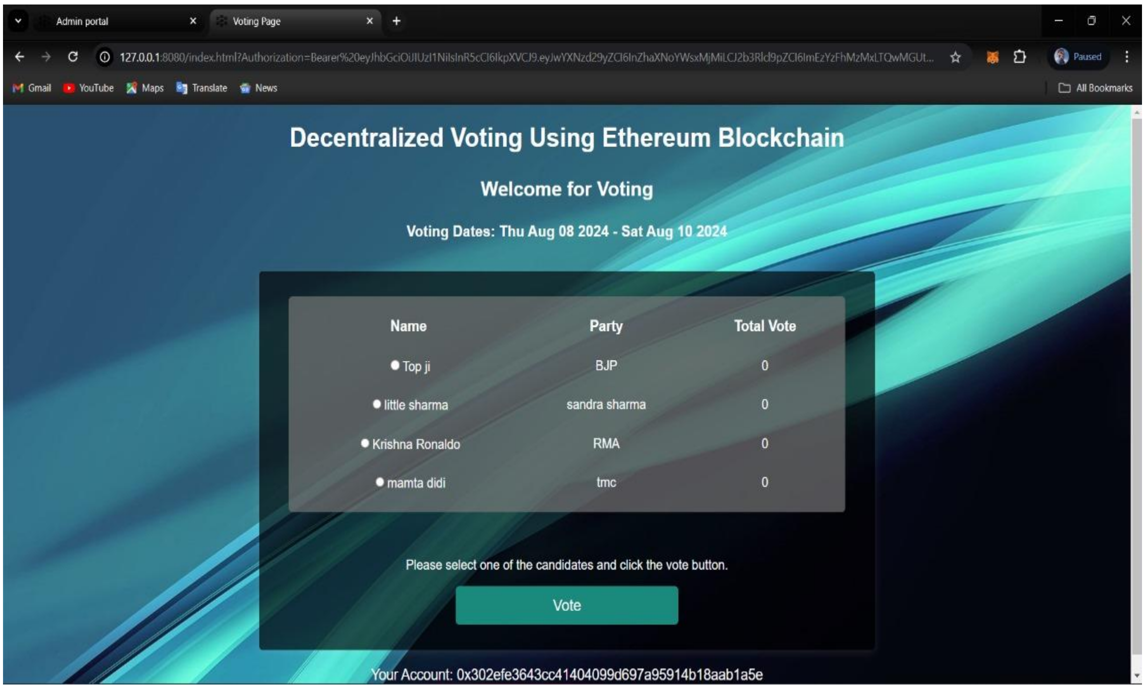
DETAILS HEX

Estimated fee 0.01663689
0.01663689 ETH
Site suggested -15 sec Max fee: 0.01684025 ETH

Total 0.01663689 0.01663689 ETH
Amount + gas fee Max amount: 0.01684025 ETH

Reject Confirm

7.3 ADDING CANDIDATE


Admin portal
Voting Page
127.0.0.1:8080/index.html?Authorization=Bearer%20eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJwYXNzd29yZCI6InZhaXNoYWwxMjMiLCJ2b3RlcklkIjoiMzEzYzFhMzMxLTQwMGUt...
Paused
Gmail
YouTube
Maps
Translate
News
All Bookmarks

Decentralized Voting Using Ethereum Blockchain

Welcome for Voting

Voting Dates: Thu Aug 08 2024 - Sat Aug 10 2024

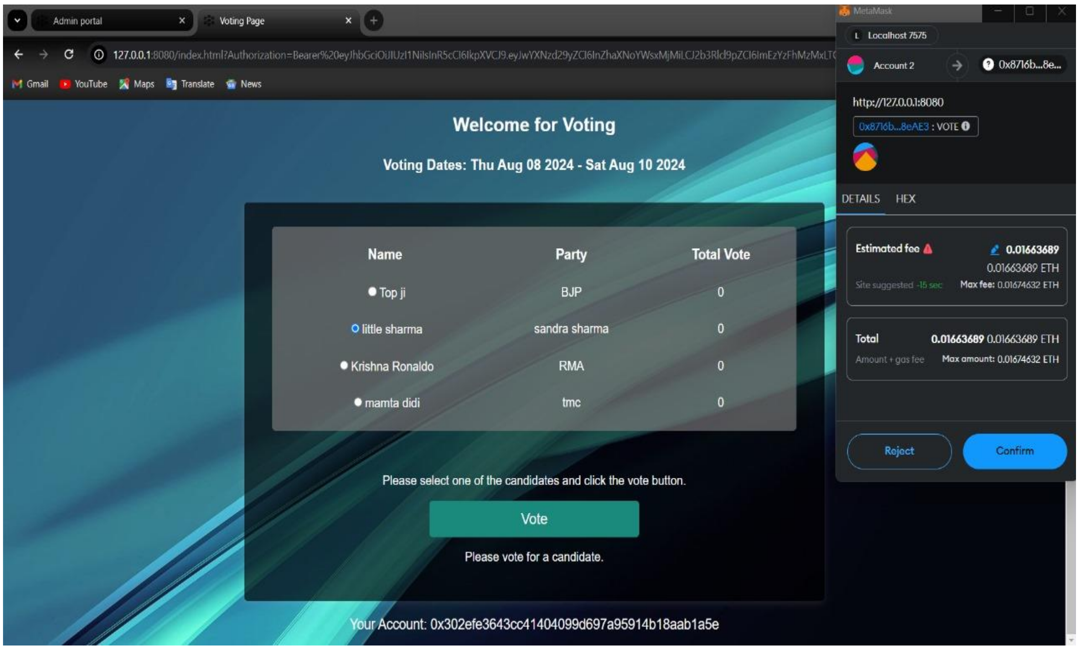
Name	Party	Total Vote
○ Top ji	BJP	0
○ little sharma	sandra sharma	0
○ Krishna Ronaldo	RMA	0
○ mamta didi	tmc	0

Please select one of the candidates and click the vote button.

Vote

Your Account: 0x302efe3643cc41404099d697a95914b18aab1a5e

7.4 LIST OF CANDIDATES



127.0.0.1:8080/index.html?Authorization=Bearer%20eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJwYXNzd29yZCI6InZhaXNoYWsxMjMiLCJ2b3Rld9pZCI6ImEzYzFhMzMxLTC

Welcome for Voting

Voting Dates: Thu Aug 08 2024 - Sat Aug 10 2024

Name	Party	Total Vote
● Top ji	BJP	0
○ little sharma	sandra sharma	0
● Krishna Ronaldo	RMA	0
● mamta didi	tmc	0

Please select one of the candidates and click the vote button.

Please vote for a candidate.

Your Account: 0x302efe3643cc41404099d697a95914b18aab1a5e

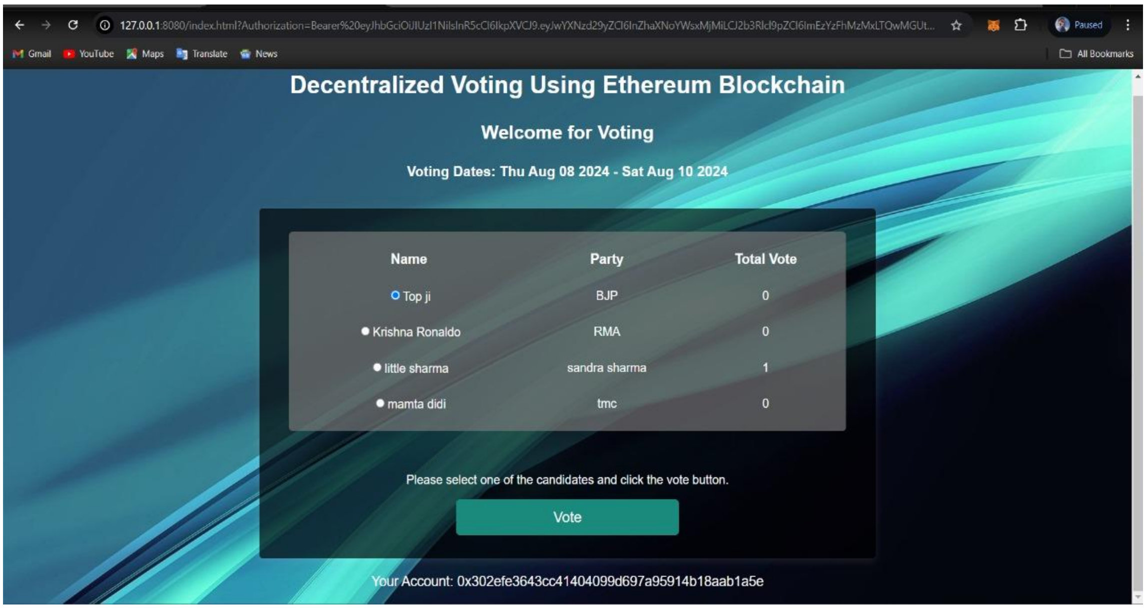
http://127.0.0.1:8080

0x8716b...8eAE3 : VOTE

Estimated fee 0.01663689
0.01663689 ETH
Site suggested -15 sec Max fee: 0.01674632 ETH

Total 0.01663689 0.01663689 ETH
Amount + gas fee Max amount: 0.01674632 ETH

7.4 CASTING VOTE



127.0.0.1:8080/index.html?Authorization=Bearer%20eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJwYXNzd29yZCI6InZhaXNoYWsxMjMiLCJ2b3Rld9pZCI6ImEzYzFhMzMxLTQwMGUt...

Decentralized Voting Using Ethereum Blockchain

Welcome for Voting

Voting Dates: Thu Aug 08 2024 - Sat Aug 10 2024

Name	Party	Total Vote
○ Top ji	BJP	0
● Krishna Ronaldo	RMA	0
● little sharma	sandra sharma	1
● mamta didi	tmc	0

Please select one of the candidates and click the vote button.

Your Account: 0x302efe3643cc41404099d697a95914b18aab1a5e

7.5 VOTING RESULT

CHAPTER -8

CONCLUSION

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 don’t have access to current Aadhar’s or Voter SDK to integrate in our application.

CHAPTER -9

REFERENCES

References

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.

1. Ohlin, Jens David. "Did Russian cyber interference in the 2016 election violate International law." Tex. L. Rev. 95 (2016): 1579.
2. Ayed, Ahmed Ben. "A conceptual secure blockchain-based electronic voting system." International Journal of Network Security & Its Applications 9.3 (2017): 01-09.
3. Hanifatunnisa, Rifa, and Budi Rahardjo. "Blockchain based e-voting recording system design." 2017 11th International Conference on Telecommunication Systems Services and Applications. IEEE, 2017.
4. Yu, Bin, et al. "Platform-independent secure blockchain-based voting system."International Conference on Information Security. Springer, Cham, 2018.
5. Krishnan, A., Blockchain Empowers Social Resistance and Terrorism ThroughDecentralized Autonomous Organizations. 1111 Journal of Strategic Security, 2020. **13**(1): p.41-58.
6. Vote, F.M. "The Secure Mobile Voting Platform Of The Future - Follow My Vote." 2020; https://followmyvote.com/.
7. Racsko, P. Blockchain and Democracy. Soc. Econ. **2019**, 41, 353–369.
8. Chan, S.; Chu, J.; Zhang, Y.; Nadarajah, S. Blockchain and Cryptocurrencies. J. Risk Financ. Manag. **2020**, 13, 227.
9. Wei, P., et al., "Blockchain data-based cloud data integrity protection mechanism"Future Generation Computer Systems, 2020. **102**: p. 1049 902-911.