

# PROJECT REPORT ON

Team ID & Title:  
2023FC12057

## Voting using Blockchain Technology

Submitted

as part of **CSI3013-Blockchain Technologies (Project)**  
by

20MID0055  
Dona Mary Jose

20MID0067  
Kiraniya N

20MIC0041  
R. Kartik Kushaal

20MID0114  
Priyanshu Prakash

20MID0119  
Damini N

To



Mr. Raparthi Yaswanth

School of Computer Science and Engineering



VIT<sup>®</sup>

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

November 2023

# Index

Chapter	Topic	Page No.
1	Importance of idea & problem statement	1
2	Objectives	2
3	Literature Review	3
3.1.	National Status	3
3.2.	International Status	4
4	Requirements	5
4.1.	Hardware	5
4.2.	Software	5
5	Architecture	6
6	Proposed model Details	7
6.1.	Objects of the proposed model	7
6.2.	Summary of proposed model	7
6.3.	Detailed Description of the Invention	8
7	Methodology	9
8	Output Screenshots	11
9	Implementation	18
10	Results and Discussion	23
11	Conclusion	24

# Chapter 1

## IMPORTANCE OF IDEA & PROBLEM STATEMENT

### IMPORTANCE OF IDEA:

- A decentralized system is less susceptible to manipulation by a single entity.
- Once a transaction is added to the ledger it cannot be edited or deleted.
- Blockchain can reduce the need for these intermediaries, streamlining the voting process and reducing the potential for human error or corruption.
- Once a vote is recorded on the blockchain, it cannot be altered or tampered with. This immutability ensures the integrity of the voting process.

### PROBLEM STATEMENT

To design a Blockchain based voting system, in order to maintain the confidentiality and security of the voting.

#### *Keywords:*

- Security Concerns: Despite its security measures, hackers can manipulate votes through malicious software on voters' devices.
- One-Time Voting: Users can only cast their votes once, and changes due to user mistakes are not possible.
- Blockchain Security: While blockchain offers security, it's not immune to attacks, requiring vigilance against evolving cyber threats.
- Privacy Concerns: Use of Shamir's Secret Sharing is mentioned for privacy, but its effectiveness and vulnerabilities need thorough evaluation.
- User Acceptance: Gaining trust and adoption of the new system by voters and authorities poses a significant challenge.

## CHAPTER 2

### OBJECTIVES

1. To prevent unauthorized changes to voting records and ensure the integrity of the election results by storing votes in an immutable and tamper-resistant ledger.
2. To use cryptographic methods to verify the identity of voters, candidates, and election authorities, reducing the risk of fraudulent activities.
3. To protect the privacy of individual voters by employing cryptographic techniques that make it difficult to link votes back to specific individuals and to ensure that the content of ballots remains confidential while being securely recorded on the blockchain.
4. To enable anyone to independently verify the election results and the integrity of the voting process by examining the blockchain ledger.
5. To provide real-time updates on the voting process, allowing citizens to track their votes and monitor the progress of the election.
6. To allow voters to participate in elections from remote locations, increasing participation rates.
7. To reduce the time required for manual tabulation, verification as well as the cost.

## CHAPTER 3

### LITERATURE REVIEW

The voting system faces security concerns, with potential for hacker manipulation despite security measures. It offers one-time voting but lacks One-Time Password (OTP) registration and may have issues with result accuracy. Blockchain's security is not foolproof, requiring vigilance, and privacy measures like Shamir's Secret Sharing need thorough evaluation. User acceptance is a significant challenge.

#### 3.1. National Status

- The Government of India has started a research project, named 'Design and Development of a Unified Blockchain Framework for Offering National Blockchain Service and Creation of Blockchain Ecosystem'.
- The objective is to enable the smooth integration and delivery of blockchain-as-a-service (BaaS) on a distributed infrastructure by establishing Open Application Programming Interfaces (Open APIs).
- As blockchain plays an important role in Web3, this project is part of the government's plan to realize Web3.
- Blockchain technology may encounter various challenges while its adoption in India. Discussed below are the same:
  - Technological Challenges – The technological infrastructure of the country and the lack of technical awareness is one of the biggest challenges
  - Storage – Data stored in the Blockchain cannot be modified. This demands very heavy resources in terms of storage and may become an issue as the chain of blocks grow
  - Skillset and Awareness Issues – Manpower who knows both Domain & Technology is required for blockchain technology management, which may be challenging to find
  - Security, Privacy and Regulation – Blockchain data is stored on every node on the network and hence privacy is not an inherent feature
  - Legal Challenges – The Reserve Bank of India has put forth restriction with respect to virtual currencies based on Blockchain technology and there is a circular to halt the usage of crypto-currency transactions in India.

### 3.2. International Status

- Blockchain-based Service Network (BSN) initiative of China aims at helping companies and individuals deploy Blockchain applications faster and cheaper
- The United Arab Emirates has “Smart Dubai” initiative, which aims to become “the first city fully powered by Blockchain
- In the US, Food and Drug inspection is using Blockchain to address the problem of lack of transparency and security in the health data processing
- Ethereum is used by various countries for various reasons:
- In Brazil government announced to move applications and popular voting onto Ethereum
- Chile uses Ethereum to track the data and finances from the energy grid to resist corruption and exploitation
- Digital IDs in Switzerland are offered and registered on Ethereum
- Samsung Blockchain Wallet powered by COSMOCHAIN Blockchain has developed CosmeeDApp for purchasing of contents using cryptocurrency

# CHAPTER 4

## REQUIREMENTS

### 4.1. Hardware Requirements

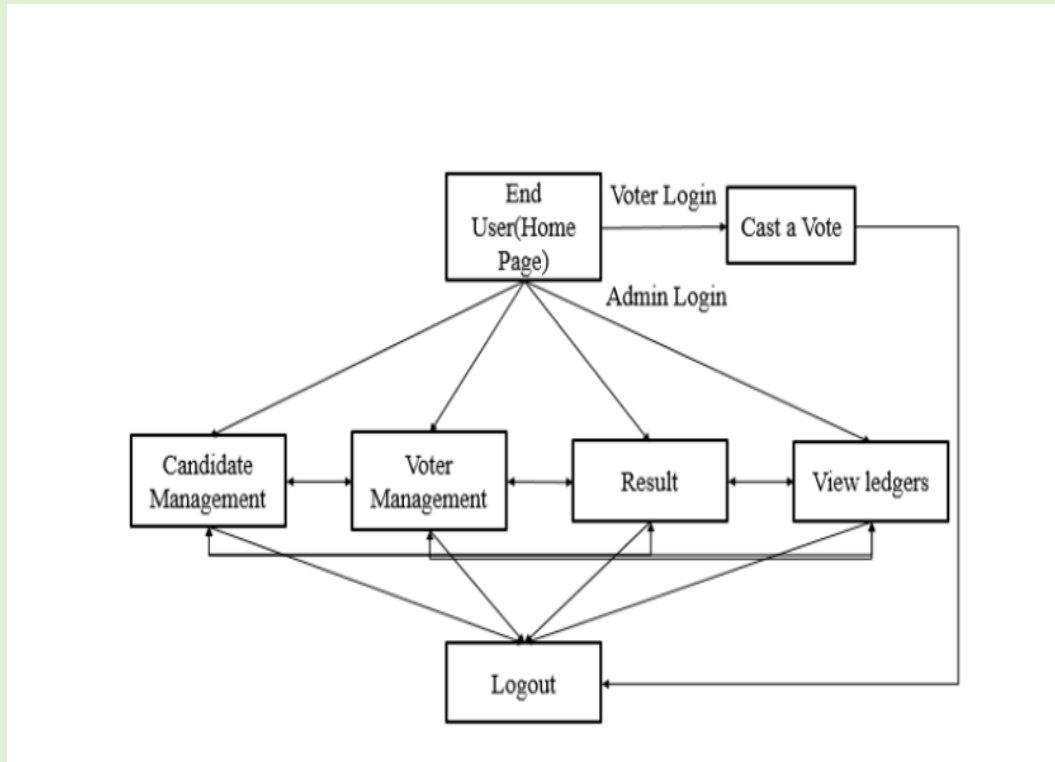
- Processor 3.5 GHz
- Intel HDD 1TB
- RAM 8 GB

### 4.2. Software Requirements

- React.js
- Solidity
- Metamask
- Ganache
- Any web browser

# CHAPTER 5

## ARCHITECTURE





## CHAPTER 6

### PROPOSED MODEL DETAILS

#### 6.1. Objectives of the proposed model

The proposed model aims to achieve the following objectives:

- **Decentralized Voting Application:** Develop a decentralized voting application using JavaScript for the frontend, ensuring a user-friendly interface for voters to interact with the system.
- **Smart Contract Development:** Write a smart contract in Solidity to handle the voting logic and candidate management. The smart contract will be deployed on the Ethereum blockchain.
- **Blockchain Integration:** Utilize the Ganache framework and Ether.js library to establish communication between the frontend and the blockchain. This integration ensures that the application can interact seamlessly with the deployed smart contract.
- **MetaMask Integration:** Integrate MetaMask, a widely used Ethereum wallet, to allow users to connect their blockchain addresses securely. MetaMask will facilitate user interactions with the decentralized voting application and the smart contract.
- **Client-Side Execution:** Design and implement the application to run entirely on the client side, within the user's browser. This ensures a decentralized architecture, providing users with control over their voting actions.
- **Voting Operations:** Implement functionalities within the application to enable users to cast votes securely through MetaMask. The application should also allow users to retrieve information about candidates and election status from the deployed smart contract.
- **Security and Privacy:** Implement secure and privacy-focused practices in the development of the decentralized voting application. This includes ensuring the confidentiality of votes and protecting user data during interactions with the blockchain.
- **User Education and Documentation:** Provide comprehensive documentation and user education materials to guide users through the process of using the decentralized voting application. This includes instructions on connecting Metamask, casting votes, and accessing election-related information.
- **Testing and Deployment:** Conduct thorough testing of the application's functionality and security features. Once testing is successful, deploy the application on a production environment, making it accessible to users.
- **Scalability Considerations:** Explore scalability considerations to accommodate potential growth in the number of users and transactions. Ensure that the application and smart contract are designed to handle increased demand while maintaining performance and responsiveness.

## 6.2. Summary of proposed model

The proposed model outlines the development of a decentralized voting application using Ganache and Solidity. The application will run on the client side, providing users with a dynamic and responsive interface for voting. A Solidity smart contract will be written to manage voting logic and candidate information, and it will be deployed on the blockchain. Communication between the frontend and the blockchain will be facilitated through the use of Ganache and Ether.js. MetaMask, an Ethereum wallet, will be integrated to securely connect users to the blockchain and interact with the smart contract. An RPC node with Ether.js will handle communication between the application and the blockchain. The model emphasizes user privacy and security, and comprehensive documentation will guide users through the application, including instructions for connecting MetaMask and casting votes. Thorough testing will precede deployment, and scalability considerations will be explored to accommodate potential growth. The model aims to deliver a decentralized, secure, and user-friendly voting solution.

## 6.3. Detailed Description of the proposed model

### 1. Frontend Development with JavaScript:

- In the context of the decentralized voting application, frontend development involves creating the user interface that voters will interact with. The goal is to ensure a user-friendly and dynamic voting interface that runs seamlessly on the client side, enhancing the overall user experience.
- We will be using HTML, CSS and JavaScript

### 2. Smart Contract Development in Solidity:

- Solidity is a programming language designed for creating smart contracts on blockchain platforms, particularly Ethereum.
- A Solidity smart contract is crafted to define the rules and operations of the decentralized voting system. This includes coding functions for tasks such as casting votes, managing candidates, and other relevant features. The smart contract acts as the backbone of the decentralized application, enforcing the logic of the voting process and securely recording votes on the blockchain.

### 3. Blockchain Deployment:

- The Solidity smart contract developed in the previous step is deployed on the blockchain. This deployment leverages the decentralized and secure nature of blockchain technology, ensuring immutability and transparency in recording votes.

### 4. Metamask Integration for User Authentication:

- Metamask is an Ethereum wallet and browser extension that enables users to interact with decentralized applications securely.
- Metamask is integrated into the voting application to securely connect users to the blockchain. It acts as a bridge between the application and the user's Ethereum address,

providing a secure authentication mechanism. Metamask enables users to interact with the deployed smart contract, such as casting votes, in a secure and user-friendly manner.

## 5. RPC Node with Ether.js for Communication:

- RPC (Remote Procedure Call) is a protocol that one program can use to request a service from a program on another computer. Ether.js is used here for Ethereum blockchain communication.
- An RPC node, coupled with Ether.js, is implemented to handle communication between the decentralized voting application and the Ethereum blockchain. This setup allows the application to query blockchain data and send transactions to the smart contract securely. Ether.js streamlines the interaction process, ensuring reliable and efficient communication.

## CHAPTER 7

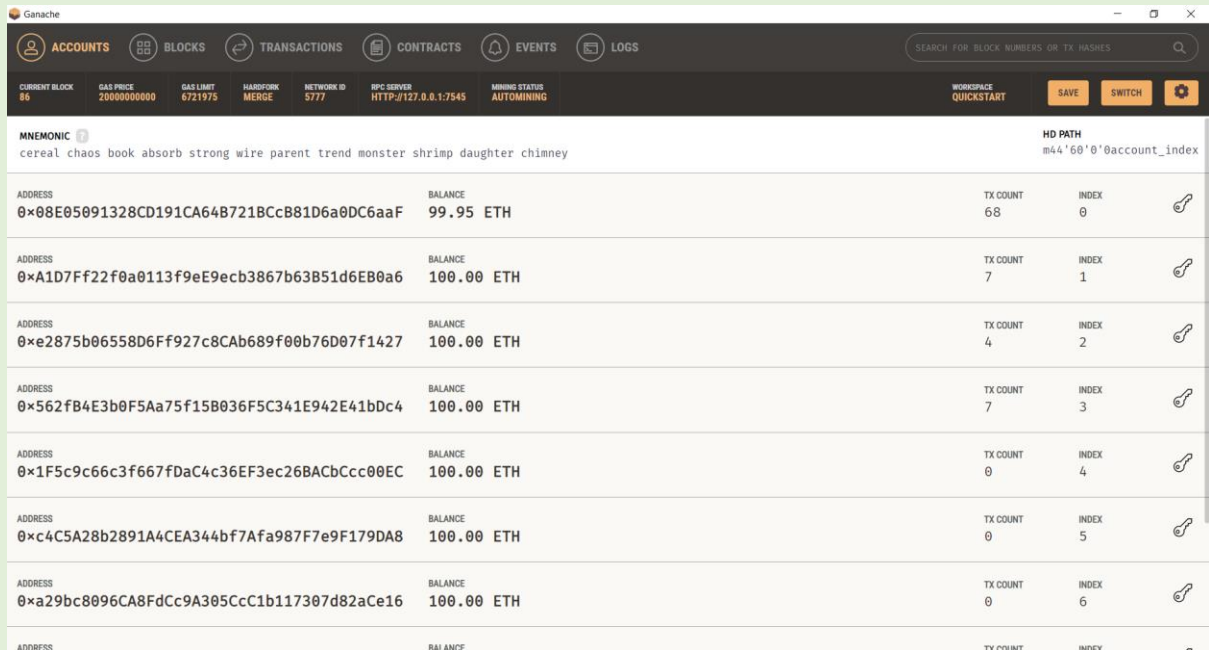
### METHODOLOGY

- In this system the voter/user must first register themselves using a registration form available within the web application and once the registration form is being submitted, an entry is made in the blockchain.
- After the registration, the user can log into the application and be a part of the polling process. The user with valid credentials can log into the system by entering their Aadhar number and password.
- Once the user is logged into their respective account the dashboard contains all the information about voting.
- When user click on the vote menu, they will see the candidates of the respective parties.
- User will select a one party and cast the vote for that party. Only one vote is casted from each account.
- Once the vote is cast that account is registered into the blockchain.
- The system has a secure login system which prevents people from casting votes on behalf of others.
- By clicking the result menu user can see the result on the dashboard.
- Dashboard also contain admin panel only accessed by the admin; the admin can add new candidates into the system and see the user's name with its valid credentials.

# CHAPTER 8

## OUTPUT SCREENSHOTS

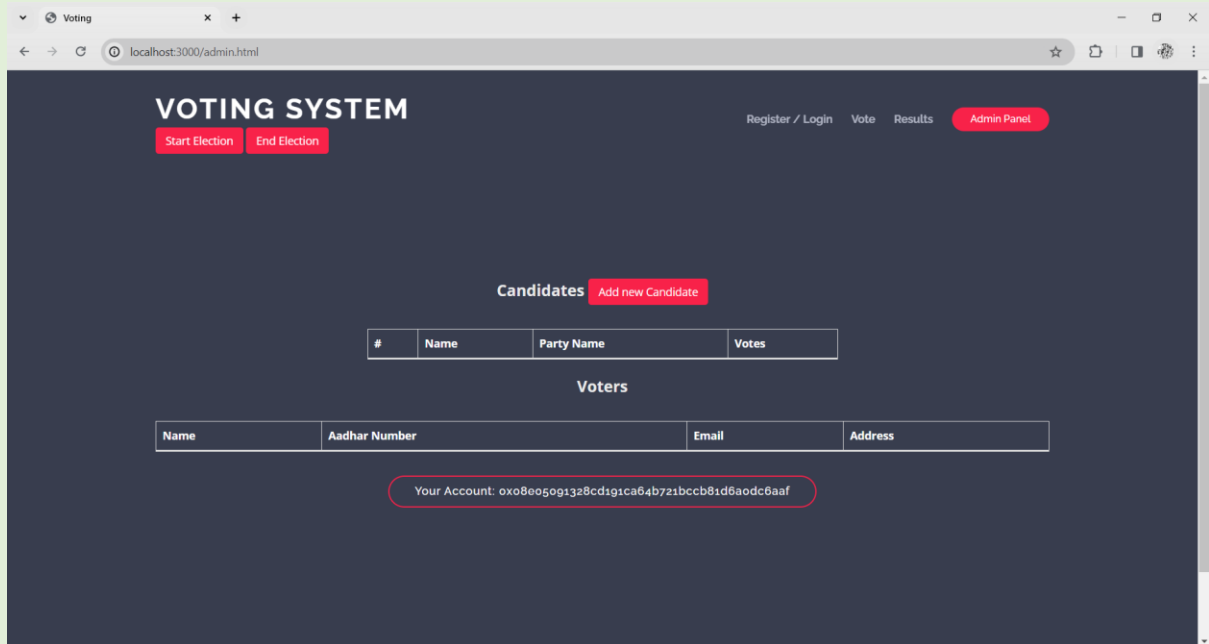
### Ganache Screenshots



The screenshot shows the Ganache desktop application. The top navigation bar includes tabs for ACCOUNTS, BLOCKS, TRANSACTIONS, CONTRACTS, EVENTS, and LOGS. Below this, a status bar displays various network metrics. The main content area shows the 'ACCOUNTS' tab with a list of accounts. Each account entry includes an address, balance, transaction count, and index. A mnemonic phrase is visible at the top left of the account list.

ADDRESS	BALANCE	TX COUNT	INDEX
0x08E05091328CD191CA64B7218CcB81D6a0DC6aaF	99.95 ETH	68	0
0xA1D7Ff22f0a0113f9eE9ecb3867b63B51d6E80a6	100.00 ETH	7	1
0xe2875b06558D6Ff927c8CAb689f00b76D07f1427	100.00 ETH	4	2
0x562fB4E3b0F5Aa75f15B036F5C341E942E41bDc4	100.00 ETH	7	3
0x1F5c9c66c3f667fDaC4c36EF3ec26BACbCcc00EC	100.00 ETH	0	4
0xc4C5A28b2891A4CEA344bf7Afa987F7e9F179DA8	100.00 ETH	0	5
0xa29bc8096CA8FdCc9A305CcC1b117307d82aCe16	100.00 ETH	0	6

### Admin Page



The screenshot shows the 'Admin Panel' of a 'VOTING SYSTEM'. The panel includes buttons for 'Start Election', 'End Election', 'Register / Login', 'Vote', 'Results', and 'Admin Panel'. Below these, there are sections for 'Candidates' and 'Voters'. The 'Candidates' section has an 'Add new Candidate' button and a table with columns for '#', 'Name', 'Party Name', and 'Votes'. The 'Voters' section has a table with columns for 'Name', 'Aadhar Number', 'Email', and 'Address'. At the bottom, a message displays the user's account ID: 'Your Account: 0xo8e05091328cd191ca64b721bccb81d6a0dc6aaf'.

#	Name	Party Name	Votes
---	------	------------	-------

Name	Aadhar Number	Email	Address
------	---------------	-------	---------

Your Account: 0xo8e05091328cd191ca64b721bccb81d6a0dc6aaf

## Add a new Candidate

Voting System [Register / Login](#) [Results](#) [Admin Panel](#)

### Add a new candidate

First Name	Last Name
<input type="text" value="Rajesh"/>	<input type="text" value="C"/>
Party Name	
<input type="text" value="ABC"/>	
<a href="#">Add Candidate</a>	
Your Account: 0xo8e05091328cd191ca64b7z1bccb81d6a0dc6aaf	

Voting System [Register / Login](#) [Results](#)

### Add a new candidate

First Name	Last Name
<input type="text" value="Rajesh"/>	<input type="text" value="C"/>
Party Name	
<input type="text" value="ABC"/>	
<a href="#">Add Candidate</a>	
Your Account: 0xo8e05091328cd191ca64b7z1bccb81d6a0dc6aaf	

MetaMask Notification

Account 6 [Ganache](#)

<https://localhost:3000>

0xd1aA...483C9 : ADD CANDIDATE

DETAILS DATA HEX

Market >

Gas (estimated) 0.00443825  
0.00443825 ETH  
Likely in < 30 seconds Max fee: 0.0058812 ETH

Total 0.00443825  
Amount + gas fee Max amount: 0.0058812 ETH

[Reject](#) [Confirm](#)

Voting

localhost:3000/admin.html

VOTING SYSTEM

Register / LoginVoteResultsAdmin Panel

Start ElectionEnd Election

Candidates

Add new Candidate

#	Name	Party Name	Votes
1	Rajesh C	ABC	0

Voters

Name	Aadhar Number	Email	Address
------	---------------	-------	---------

Your Account: 0xo8e05091328cd191ca64b721bccb81d6a0dc6aaf

Voting

localhost:3000/admin.html

VOTING SYSTEM

Register / LoginVoteResultsAdmin Panel

Start ElectionEnd Election

Candidates

Add new Candidate

#	Name	Party Name	Votes
2	Neha SV	XYZ	0
1	Rajesh C	ABC	0

Voters

Name	Aadhar Number	Email	Address
------	---------------	-------	---------

Your Account: 0xo8e05091328cd191ca64b721bccb81d6a0dc6aaf

Admin will be able to start The Election



## User Registration

Voting

localhost:3000/index.html

### VOTING SYSTEM

#### Create an account

First Name	Last Name
User1	Last Name

Aadhar Number

123123123123

Email

user1@gmail.com

Password

\*\*\*\*\*

Register

Already have an account? Click here to login

Your Account: 0x08e05091328cd191ca64b721bccb81d6a0dc6aaf

Account 7

99.9986 ETH

Buy & Sell Send Swap Bridge Portfolio

Tokens NFTs Activity

Nov 19, 2023

Vote	Confirmed	-0 ETH
Vote	Confirmed	-0 ETH
Add User	Confirmed	-0 ETH

Voting

localhost:3000/index.html

### VOTING SYSTEM

#### Create an account

First Name	Last Name
User1	One

Aadhar Number

123123123123

Email

user1@gmail.com

Password

\*\*\*\*\*

Register

Already have an account? Click here to login

Your Account: 0x08e05091328cd191ca64b721bccb81d6a0dc6aaf

MetaMask Notification

Account 6

0xd1aA...483C9

ADD USER

DETAILS DATA HEX

Gas (estimated) 0.0058308 ETH

Likely in < 30 seconds Max fee: 0.0077275 ETH

Total 0.0058308 ETH

Amount + gas fee Max amount: 0.0077275 ETH

Reject Confirm



## Voting

**VOTING SYSTEM**

Please **cast your vote**

#	Name	Party Name	Votes
2	Neha SV	XYZ	0
1	Rajesh C	ABC	0

Select Candidate

Neha SV

**Vote**

Your Account: 0xa1d7ff22f0a0113fgeegecb3867b63b51d6eboa6

MetaMask Notification

Account 7 → 0xa1d7ff22f0a0113fgeegecb3867b63b51d6eboa6

http://localhost:3000

0xa1d7ff22f0a0113fgeegecb3867b63b51d6eboa6 : VOTE

DETAILS DATA HEX

Gas (estimated) 0.00216944  
**0.00216944 ETH**  
Likely in < 30 seconds Max fee: 0.00287409 ETH

Total 0.00216944  
Amount + gas fee Max amount: 0.00287409 ETH

**Reject Confirm**

Once the user votes, they will not be allowed to vote again.

**VOTING SYSTEM**

Register / Login **Vote** Results

**Vote casted succesfully for candidate 2**

#	Name	Party Name	Votes
1	Rajesh C	ABC	0
2	Neha SV	XYZ	1

Your Account: 0xa1d7ff22f0a0113fgeegecb3867b63b51d6eboa6

## Admin ends the election

The screenshot shows the admin interface of a voting system. The browser address bar indicates the URL is localhost:3000/admin.html. The page title is 'VOTING SYSTEM'. There are two buttons at the top: 'Start Election' and 'End Election'. The 'End Election' button is highlighted. Below the buttons, there is a section for 'Candidates' with an 'Add new Candidate' button. A table lists the candidates:

#	Name	Party Name	Votes
1	Rajesh C	ABC	0
2	Neha SV	XYZ	1

Below the candidates table, there is a section for 'Voters' with a table listing the voters:

Name	Aadhar Number	Email	Address
User1 One	123123123123	user1@gmail.com	0x08e05091328cd191ca64b721bccb81d6a0dc6aaf

At the bottom, there is a red box containing the text: 'Your Account: 0xa1d7ff22f0a0113f9eegecb3867b63b51d6eboa6'.

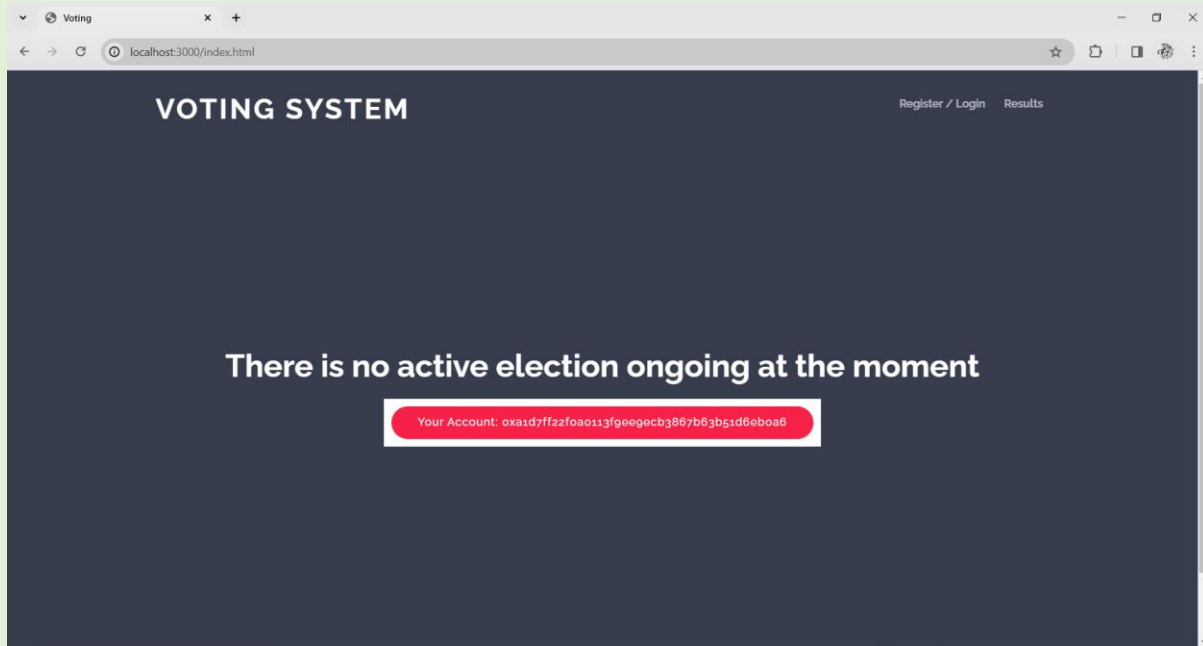
## Election Results are displayed.

The screenshot shows the results interface of the voting system. The browser address bar indicates the URL is localhost:3000/results.html. The page title is 'VOTING SYSTEM'. There are four links at the top: 'Register / Login', 'Vote', and 'Results'. The 'Results' link is highlighted. Below the links, there is a section for 'Election Results' with a table listing the results:

#	Name	Aadhar Number	Votes
1	Rajesh C	ABC	0
2	Neha SV	XYZ	1

At the bottom, there is a red box containing the text: 'Your Account: 0xa1d7ff22f0a0113f9eegecb3867b63b51d6eboa6'.

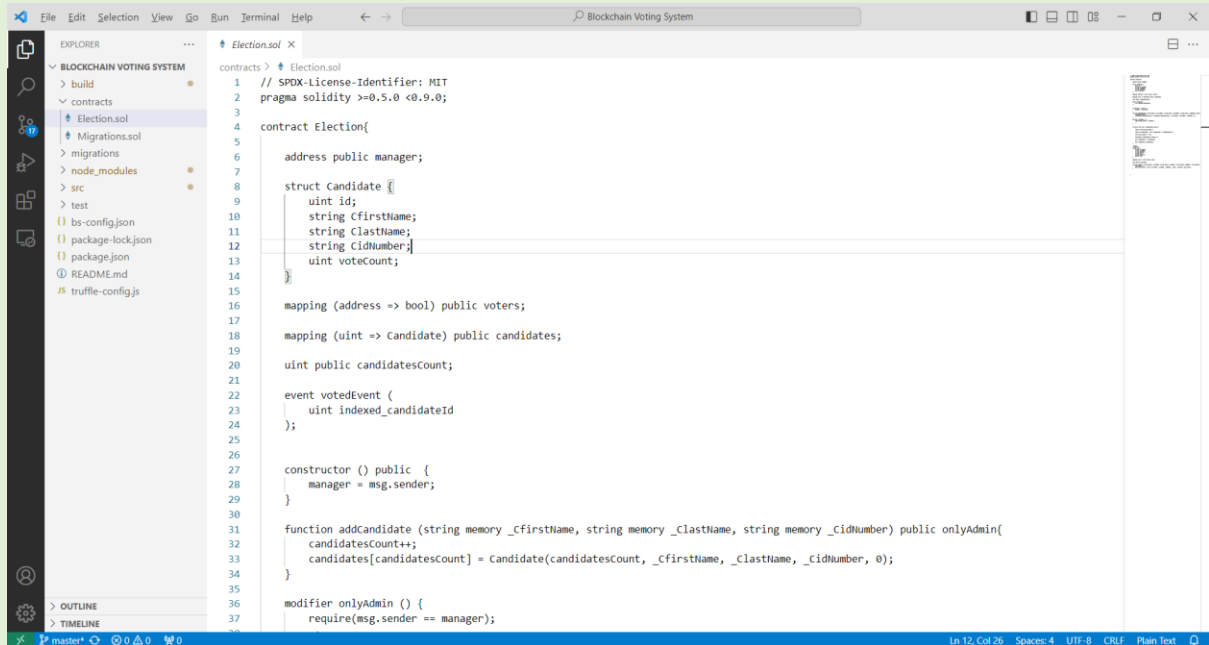
Users won't be able to login once the election has ended.



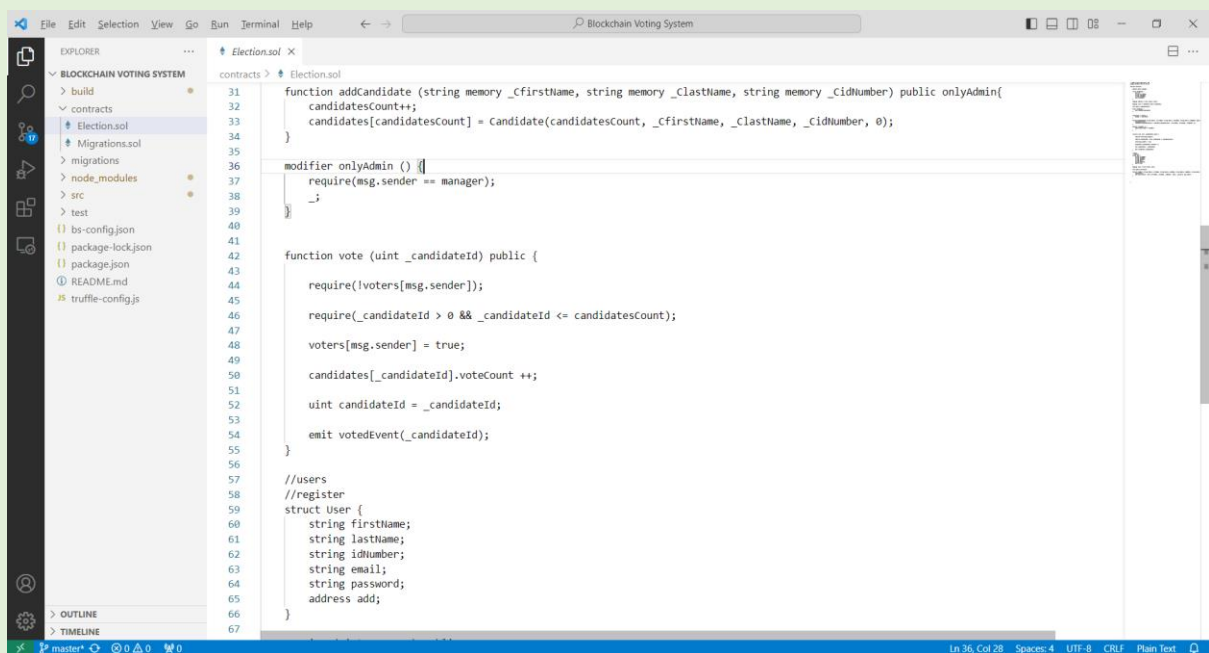
# CHAPTER 9

## IMPLEMENTATION PSEUDO CODE

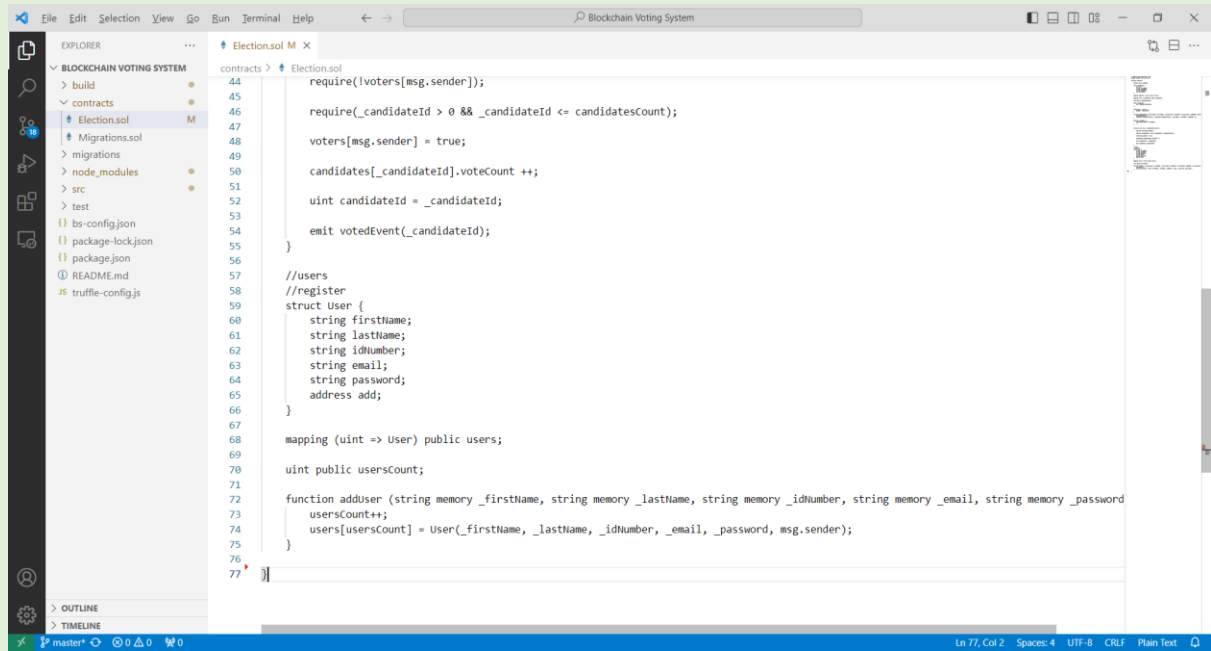
### Smart Contract – Election.sol



```
1 // SPDX-License-Identifier: MIT
2 pragma solidity >=0.5.0 <0.9.0;
3
4 contract Election{
5
6     address public manager;
7
8     struct Candidate {
9         uint id;
10        string _firstName;
11        string _lastName;
12        string _cidNumber;
13        uint voteCount;
14    }
15
16    mapping (address => bool) public voters;
17
18    mapping (uint => Candidate) public candidates;
19
20    uint public candidatesCount;
21
22    event votedEvent (
23        uint indexed_candidateId
24    );
25
26    constructor () public {
27        manager = msg.sender;
28    }
29
30    function addCandidate (string memory _cfirstName, string memory _clastName, string memory _cidNumber) public onlyAdmin{
31        candidatesCount++;
32        candidates[candidatesCount] = Candidate(candidatesCount, _cfirstName, _clastName, _cidNumber, 0);
33    }
34
35    modifier onlyAdmin () {
36        require(msg.sender == manager);
37    }
38}
```

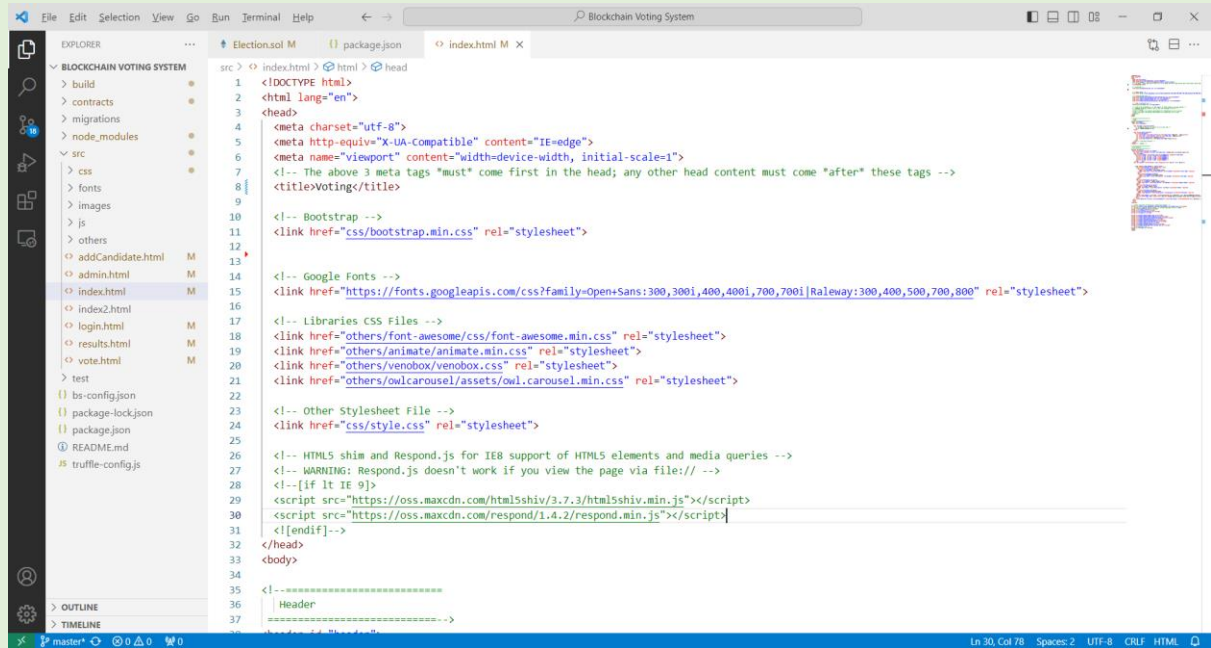


```
31 function addCandidate (string memory _cfirstName, string memory _clastName, string memory _cidNumber) public onlyAdmin{
32     candidatesCount++;
33     candidates[candidatesCount] = Candidate(candidatesCount, _cfirstName, _clastName, _cidNumber, 0);
34 }
35
36 modifier onlyAdmin () {
37     require(msg.sender == manager);
38     _;
39 }
40
41
42 function vote (uint _candidateId) public {
43     require(!voters[msg.sender]);
44
45     require(_candidateId > 0 && _candidateId <= candidatesCount);
46
47     voters[msg.sender] = true;
48
49     candidates[_candidateId].voteCount ++;
50
51     uint candidateId = _candidateId;
52
53     emit votedEvent(_candidateId);
54 }
55
56 //users
57 //register
58 struct User {
59     string firstName;
60     string lastName;
61     string idNumber;
62     string email;
63     string password;
64     address add;
65 }
66
67 }
```



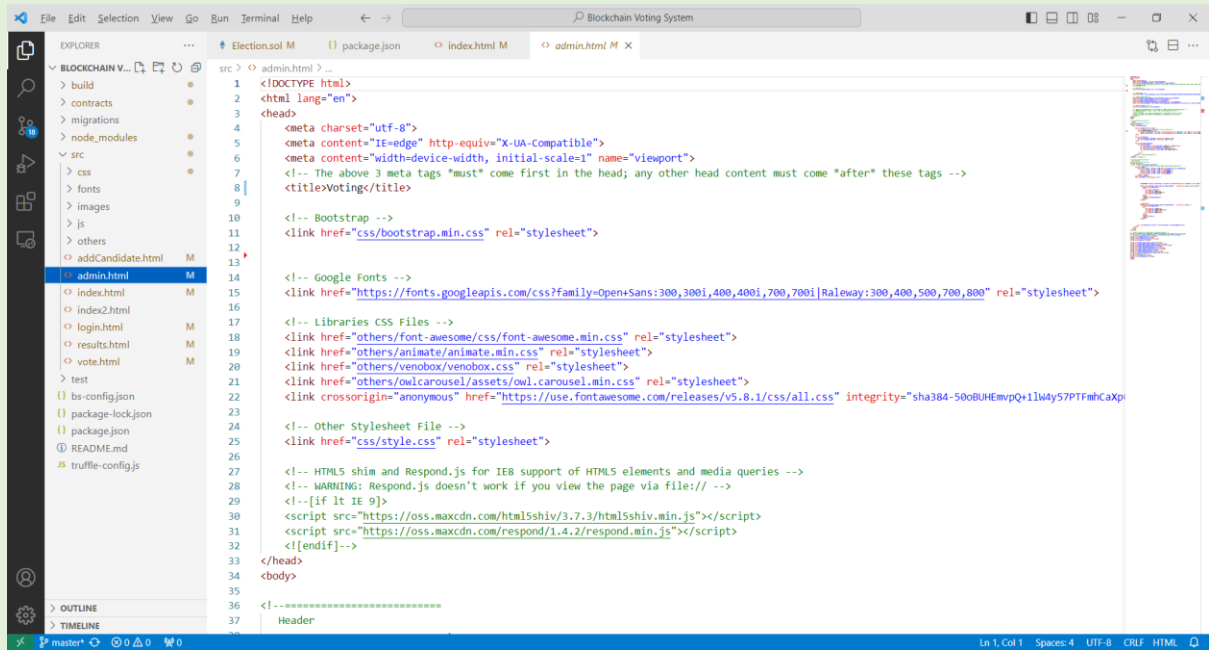
```
contracts > Election.sol
44 require(!voters[msg.sender]);
45
46 require(_candidateId > 0 && _candidateId <= candidatesCount);
47
48 voters[msg.sender] = true;
49
50 candidates[_candidateId].voteCount ++;
51
52 uint candidateId = _candidateId;
53
54 emit votedEvent(_candidateId);
55
56
57 //users
58 //register
59 struct User {
60     string firstName;
61     string lastName;
62     string idNumber;
63     string email;
64     string password;
65     address add;
66 }
67
68 mapping (uint => User) public users;
69
70 uint public usersCount;
71
72 function addUser (string memory _firstName, string memory _lastName, string memory _idNumber, string memory _email, string memory _password
73     usersCount++;
74     users[usersCount] = User(_firstName, _lastName, _idNumber, _email, _password, msg.sender);
75 }
76
77
```

## index.html



```
src > index.html > html > head
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 <meta name="viewport" content="width=device-width, initial-scale=1">
7 <!-- The above 3 meta tags *must* come first in the head; any other head content must come *after* these tags -->
8 <title>Voting</title>
9
10 <!-- Bootstrap -->
11 <link href="css/bootstrap.min.css" rel="stylesheet">
12
13
14 <!-- Google Fonts -->
15 <link href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,700,700i|Raleway:300,400,500,700,800" rel="stylesheet">
16
17 <!-- Libraries CSS Files -->
18 <link href="others/font-awesome/css/font-awesome.min.css" rel="stylesheet">
19 <link href="others/animate/animate.min.css" rel="stylesheet">
20 <link href="others/venobox/venobox.css" rel="stylesheet">
21 <link href="others/owlcarousel/assets/owl.carousel.min.css" rel="stylesheet">
22
23 <!-- Other Stylesheet File -->
24 <link href="css/style.css" rel="stylesheet">
25
26 <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
27 <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
28 <!--[if lt IE 9]>
29 <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
30 <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
31 <![endif]-->
32 </head>
33 <body>
34
35 <!-- =====
36 | Header
37 =====>
```

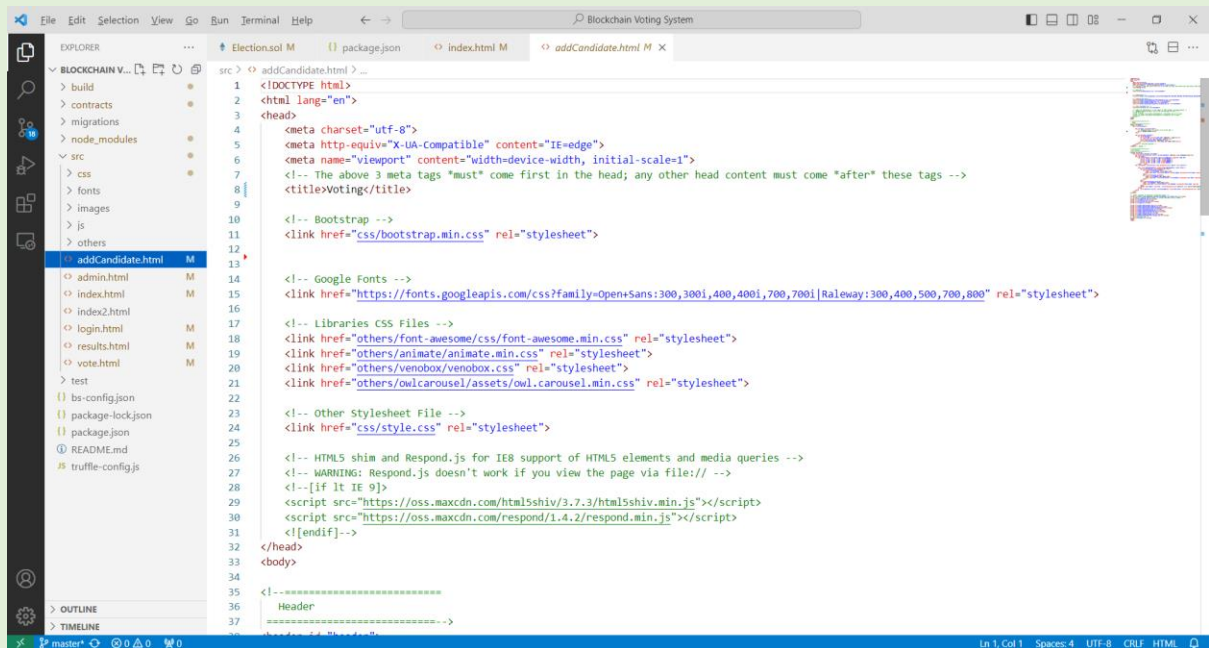
## admin.html – Admin Page



The screenshot shows the VS Code editor with the 'admin.html' file open. The Explorer sidebar on the left shows the project structure, including 'BLOCKCHAIN V...', 'build', 'contracts', 'migrations', 'node\_modules', 'src', 'css', 'fonts', 'images', 'js', 'others', 'addCandidate.html', 'admin.html', 'index.html', 'index2.html', 'login.html', 'results.html', 'vote.html', 'test', 'bs-config.json', 'package-lock.json', 'package.json', 'README.md', and 'truffle-config.js'. The main editor area displays the HTML code for 'admin.html', which includes a DOCTYPE declaration, HTML lang attribute, charset, viewport, and various CSS and JS links. The code is as follows:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <meta content="IE=edge" http-equiv="X-UA-Compatible">
6   <meta content="width=device-width, initial-scale=1" name="viewport">
7   <!-- The above 3 meta tags "must" come first in the head; any other head content must come "after" these tags -->
8   <title>Voting</title>
9
10  <!-- Bootstrap -->
11  <link href="css/bootstrap.min.css" rel="stylesheet">
12
13
14  <!-- Google Fonts -->
15  <link href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,700,700i|Raleway:300,400,500,700,800" rel="stylesheet">
16
17  <!-- Libraries CSS Files -->
18  <link href="others/font-awesome/css/font-awesome.min.css" rel="stylesheet">
19  <link href="others/animate/animate.min.css" rel="stylesheet">
20  <link href="others/venobox/venobox.css" rel="stylesheet">
21  <link href="others/owlcarousel/assets/owl.carousel.min.css" rel="stylesheet">
22  <link crossorigin="anonymous" href="https://use.fontawesome.com/releases/v5.8.1/css/all.css" integrity="sha384-500BUHEmpQ+11W4y57PTFmCaxP" rel="stylesheet">
23
24  <!-- Other Stylesheet File -->
25  <link href="css/style.css" rel="stylesheet">
26
27  <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
28  <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
29  <!--[if lt IE 9]>
30  <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
31  <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
32  <![endif]>
33 </head>
34 <body>
35
36 <!-- =====
37 Header
```

## addCandidate.html - Add Candidate Page

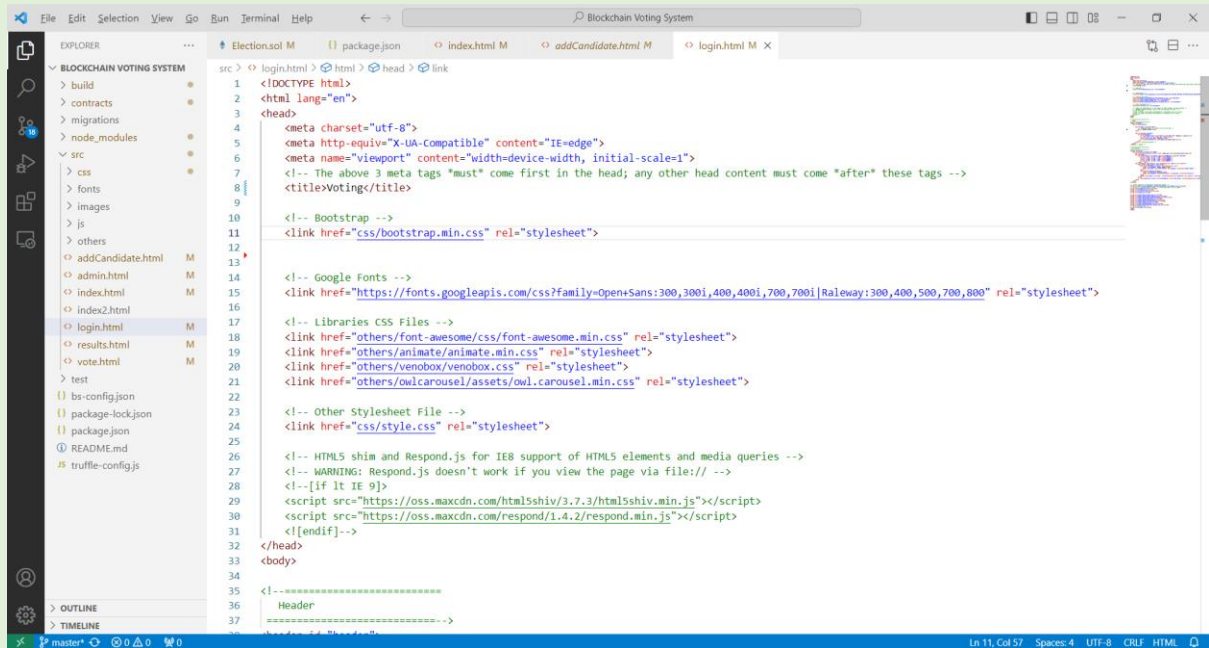


The screenshot shows the VS Code editor with the 'addCandidate.html' file open. The Explorer sidebar on the left shows the project structure, including 'BLOCKCHAIN V...', 'build', 'contracts', 'migrations', 'node\_modules', 'src', 'css', 'fonts', 'images', 'js', 'others', 'addCandidate.html', 'admin.html', 'index.html', 'index2.html', 'login.html', 'results.html', 'vote.html', 'test', 'bs-config.json', 'package-lock.json', 'package.json', 'README.md', and 'truffle-config.js'. The main editor area displays the HTML code for 'addCandidate.html', which includes a DOCTYPE declaration, HTML lang attribute, charset, viewport, and various CSS and JS links. The code is as follows:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7   <!-- The above 3 meta tags "must" come first in the head; any other head content must come "after" these tags -->
8   <title>Voting</title>
9
10  <!-- Bootstrap -->
11  <link href="css/bootstrap.min.css" rel="stylesheet">
12
13
14  <!-- Google Fonts -->
15  <link href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,700,700i|Raleway:300,400,500,700,800" rel="stylesheet">
16
17  <!-- Libraries CSS Files -->
18  <link href="others/font-awesome/css/font-awesome.min.css" rel="stylesheet">
19  <link href="others/animate/animate.min.css" rel="stylesheet">
20  <link href="others/venobox/venobox.css" rel="stylesheet">
21  <link href="others/owlcarousel/assets/owl.carousel.min.css" rel="stylesheet">
22
23  <!-- Other Stylesheet File -->
24  <link href="css/style.css" rel="stylesheet">
25
26  <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
27  <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
28  <!--[if lt IE 9]>
29  <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
30  <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
31  <![endif]>
32 </head>
33 <body>
34
35 <!-- =====
36 Header
```

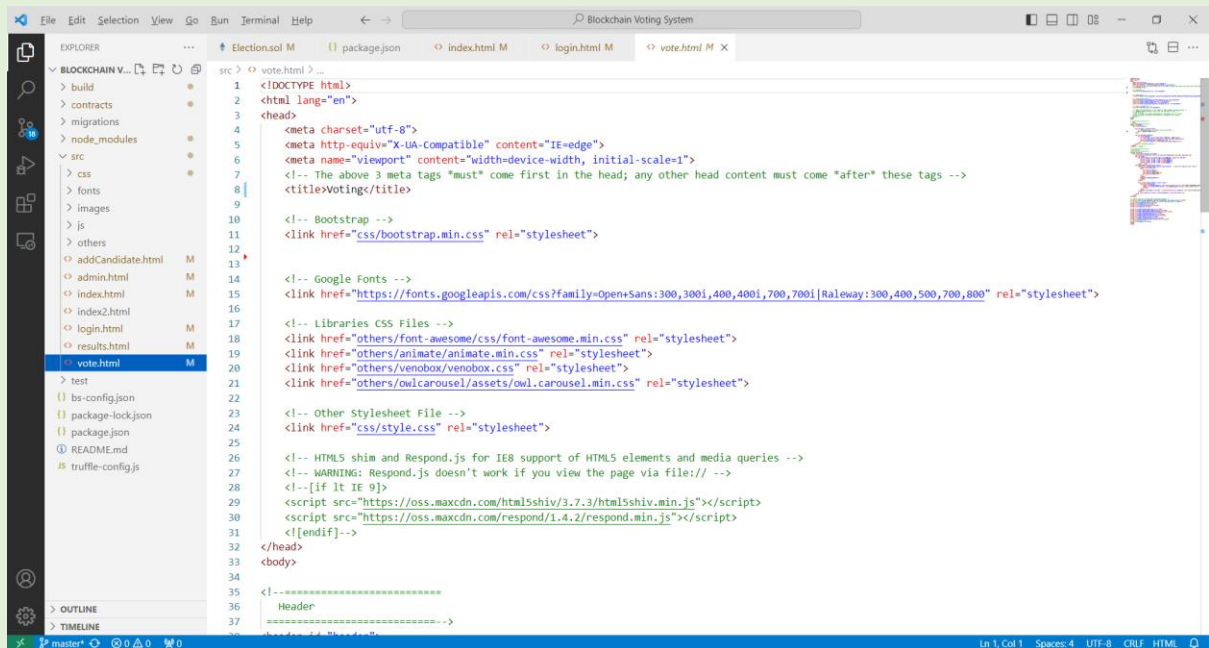


## login.html – User login page



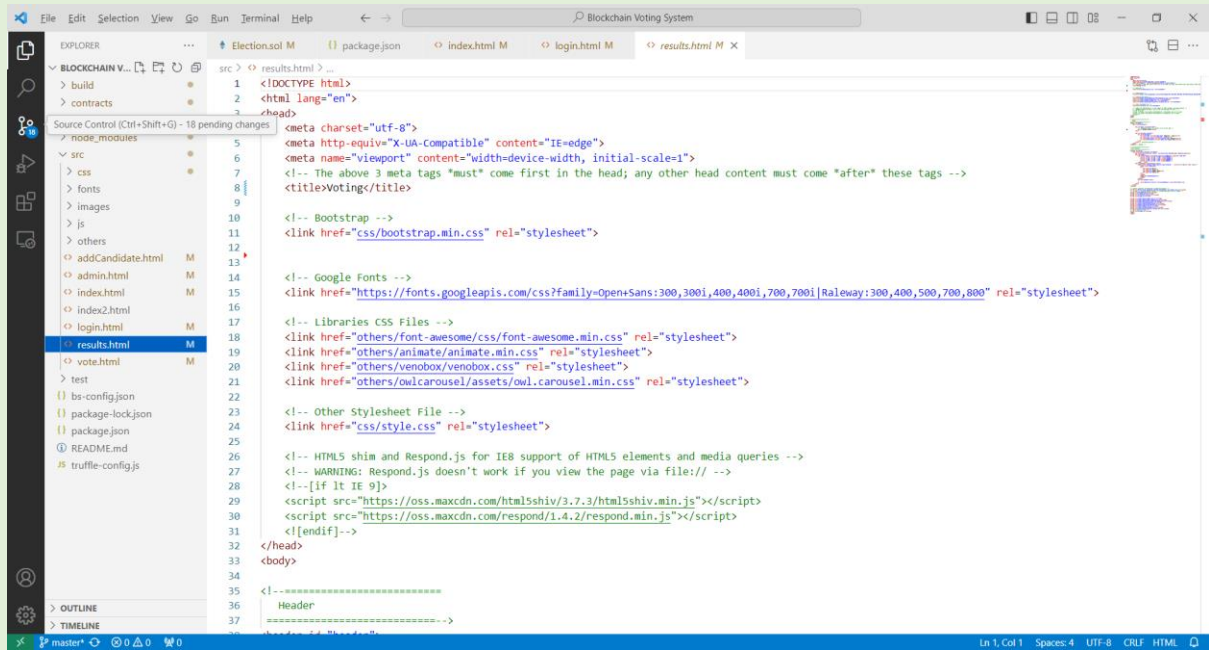
```
src > login.html > html > head > link
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1">
7      <!-- The above 3 meta tags "must" come first in the head; any other head content must come "after" these tags -->
8      <title>Voting</title>
9
10     <!-- Bootstrap -->
11     <link href="css/bootstrap.min.css" rel="stylesheet">
12
13
14
15     <!-- Google Fonts -->
16     <link href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,700,700i|Raleway:300,400,500,700,800" rel="stylesheet">
17
18     <!-- Libraries CSS Files -->
19     <link href="others/font-awesome/css/font-awesome.min.css" rel="stylesheet">
20     <link href="others/animate/animate.min.css" rel="stylesheet">
21     <link href="others/venobox/venobox.css" rel="stylesheet">
22     <link href="others/owlcarousel/assets/owl.carousel.min.css" rel="stylesheet">
23
24     <!-- Other Stylesheet File -->
25     <link href="css/style.css" rel="stylesheet">
26
27     <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
28     <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
29     <!--[if lt IE 9]>
30     <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
31     <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
32     <![endif]>
33
34 </head>
35 <body>
36
37 <!-- =====
38 Header
39 =====>
```

## vote.html – Voting Page



```
src > vote.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1">
7      <!-- The above 3 meta tags "must" come first in the head; any other head content must come "after" these tags -->
8      <title>Voting</title>
9
10     <!-- Bootstrap -->
11     <link href="css/bootstrap.min.css" rel="stylesheet">
12
13
14
15     <!-- Google Fonts -->
16     <link href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,700,700i|Raleway:300,400,500,700,800" rel="stylesheet">
17
18     <!-- Libraries CSS Files -->
19     <link href="others/font-awesome/css/font-awesome.min.css" rel="stylesheet">
20     <link href="others/animate/animate.min.css" rel="stylesheet">
21     <link href="others/venobox/venobox.css" rel="stylesheet">
22     <link href="others/owlcarousel/assets/owl.carousel.min.css" rel="stylesheet">
23
24     <!-- Other Stylesheet File -->
25     <link href="css/style.css" rel="stylesheet">
26
27     <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
28     <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
29     <!--[if lt IE 9]>
30     <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
31     <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
32     <![endif]>
33
34 </head>
35 <body>
36
37 <!-- =====
38 Header
39 =====>
```

## result.html – Result Page



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7   <!-- The above 3 meta tags "must" come first in the head; any other head content must come "after" these tags -->
8   <title>Voting</title>
9
10  <!-- Bootstrap -->
11  <link href="css/bootstrap.min.css" rel="stylesheet">
12
13  <!-- Google Fonts -->
14  <link href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,700,700i|Raleway:300,400,500,700,800" rel="stylesheet">
15
16  <!-- Libraries CSS Files -->
17  <link href="others/font-awesome/css/font-awesome.min.css" rel="stylesheet">
18  <link href="others/animate/animate.min.css" rel="stylesheet">
19  <link href="others/venobox/venobox.css" rel="stylesheet">
20  <link href="others/owlcarousel/assets/owl.carousel.min.css" rel="stylesheet">
21
22  <!-- Other Stylesheet File -->
23  <link href="css/style.css" rel="stylesheet">
24
25  <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
26  <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
27  <!--[if lt IE 9]>
28  <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
29  <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
30  <![endif]-->
31
32 </head>
33 <body>
34
35 <!-- =====
36 Header
37 ===== -->
```



## CHAPTER 10

### RESULTS AND DISCUSSION

A convincing idea in the cutting edge subculture is to modify the automated voting system to make the process of the general public voting less expensive, faster, and easier. In this work, we suggested a secure voting mechanism, which is a novel, blockchain-based electronic voting system that protects voters' privacy while enabling relaxed and cost-effective political contests. Multi component authentication is used by the proposed device to ensure that votes are legitimate. The suggested system is simple to use for all users, regardless of backgrounds, which encourages more people to voters increasing voter turnout with the added bonus of no fraudulent voters among them. As mentioned in the paper, the goal of our challenge is to develop a fully functional Dapp that is built on blockchain technology and intelligent contracts. The primary goal was to design a relaxing, trustworthy, and scalable system rather than simply a comfy instrument. A key characteristic of blockchain is the creation of an immutable system, which makes it impossible for any opponent to change the outcome of a vote after it has been cast. The issue of scalability and reliability must now be addressed because it may be crucial in a country like India where a large population may result in hundreds of thousands of servers. We have focused on developing a micro services-based architecture that will enable us to scale our server more effectively and use it in real-time scenarios.

## CHAPTER 11

### CONCLUSION

The idea of integrating online voting systems to enhance the efficiency and affordability of the public election process is a compelling notion in contemporary society. In our project, we have implemented a blockchain-based online voting framework where secure and cost-effective elections are facilitated through the use of smart contracts, ensuring voter confidentiality. Our research demonstrates that blockchain technology presents a fresh opportunity for democratic nations to transition from traditional pen-and-paper and paperless direct-recording electronic voting machines (DRE) to a more economical and time-saving election method. This shift not only strengthens the security measures of the current system but also enhances accessibility.

## REFERENCES

- [1] Ayed, A. B. (2017). A conceptual secure blockchain-based electronic voting system. *International Journal of Network Security & Its Applications*, 9(3), 01-09.
- [2] Fusco, F., Lunesu, M. I., Pani, F. E., & Pinna, A. (2018, September). Crypto-voting, a Blockchain based e-Voting System. In *KMIS* (pp. 221-225).
- [3] Alvi, S. T., Uddin, M. N., Islam, L., & Ahamed, S. (2022). DVTChain: A blockchain-based decentralized mechanism to ensure the security of digital voting system voting system. *Journal of King Saud University-Computer and Information Sciences*, 34(9), 6855-6871.
- [4] jen-ho hsiao, raylin Tso, Chien-Ming Chen, Mu-En Wu', Decentralized ETechnology-Voting Systems Based on the Blockchain
- [5] Syada Tasmia Alvi a,† , Mohammed Nasir Uddin b , Linta Islam b , Sajib Ahamed- *Journal of King Saud University – Computer and Information Sciences* 34 (2022), A blockchain-based decentralized mechanism to ensure the security of digital voting system voting system,
- [6] Micheal Pawlak, Jakub Guzuir, Aneta poniszewska Maranda, Voting Process with Blockchain Technology: Auditable Blockchain Voting System

**BRIEF BIOGRAPHY OF EACH TEAM MEMBERS**

Team\_Member-1 Name

•//Biography

Team\_Member-2 Name

•//Biography

Team\_Member-3 Name

•//Biography

Team\_Member-4 Name

•//Biography

Team\_Member-5 Name

•//Biography