#### PUNE INSTITUTE OF COMPUTER TECHNOLOGY, DHANKAWADI PUNE-43.

***A Mini-Project Report***

***On***

## D-app for E-Voting System

##### SUBMITTED BY

## NAME:

#### Devansh Mundada (41253)

#### Karishma Nair (41254)

#### Ved Patwardhan (41260)

## Yash Pawar (41261)

## Under the guidance of

### Prof. Yogesh Handge



**COMPUTER ENGINEERING DEPARTMENT**

**Academic Year: 2022-2023**

#### PUNE INSTITUTE OF COMPUTER TECHNOLOGY, DHANKAWADI PUNE-43.

***CERTIFICATE***



This is to certify that **Devansh Mundada (41253), Karishma (41254), Ved Patwardhan (41260), Yash Pawar (41261)** student of B.E. (Computer Engineering Department)

have satisfactorily completed a report on “**D-app for E- Voting System**” towards the partial fulfillment of the fourth year Computer Engineering Semester VII of SPPU.

###### Date: Place:

Dr. G. V. Kale **Head of Department*,* Computer Engineering**

## D-App for E-Voting System

**Contents**

1. [PROBLEM STATEMENT: 1](#_TOC_250007)
2. [ABSTRACT: 1](#_TOC_250006)
3. [INTRODUCTION: 2](#_TOC_250005)
4. [OBJECTIVE: 2](#_TOC_250004)
5. [MOTIVATION: 2](#_TOC_250003)
6. [THEORY: 3](#_TOC_250002)
7. [IMPLEMENTATION: 5](#_TOC_250001)
8. [RESULT: 10](#_TOC_250000)
9. CONCLUSION: 13
10. REFERENCES: 13

# 1 PROBLEM STATEMENT:

Develop a Blockchain based application D-App (de-centralized app) for e- voting

system.

# ABSTRACT:

Blockchain is a major breakthrough in the technological industry that provide immense secured platform. With the launch of Ethereum, a decentralized platform which runs decentralized applications (D-Apps) on it, a secured voting system now seems possible. Many organizations have now shifted their focus on voting through blockchain platforms. There’s a very high chance that a normal voting method won’t lead to a clear majority.

Here, we introduce the vote-trading concept where the votes can be redistributed to other candidates in case if there is no clear majority and also this ‘majority’ factor can be set by the organization according to their requirement. We discuss the design for the blockchain based preferential e-voting system using the Solidity programming language where instead of one vote per candidate, we provide the concept of giving preference to the candidates.

# INTRODUCTION:

Unlike nationwide voting where there are millions of voters, in an organization, there are very few voters, and very few nominees. As many organizations are now shifting their focus to blockchain technology because of its immense security and other reasons as mentioned in, voting through blockchain has been one of the main applications in the blockchain world. We have introduced an e-voting system based on blockchain technology.

The system utilizes smart contracts that make cost-efficient and secure election possible. We have proposed a decentralized voting platform which is based on Ethereum Technology. The future work for this system could be to develop the system further to make it more suitable for nationwide elections using some of the technologies located in the voting centers such as fingerprint. We in our system have introduced the concept of verifiable E-voting platforms and have discussed the evaluation and security concerns to the system.

# OBJECTIVE:

1. To understand and explore the working of Blockchain technology and its applications.
2. To create a blockchain application d-App for e-voting system.

.

# MOTIVATION:

Blockchain being the new and interesting emerging field has paved the path for many developers to create secure immutable applications in trustless environment. Hence as part of our curriculum we decided to create a D-App for e-voting system that could be free from manipulation and facilitate easy data storage.

# Theory:

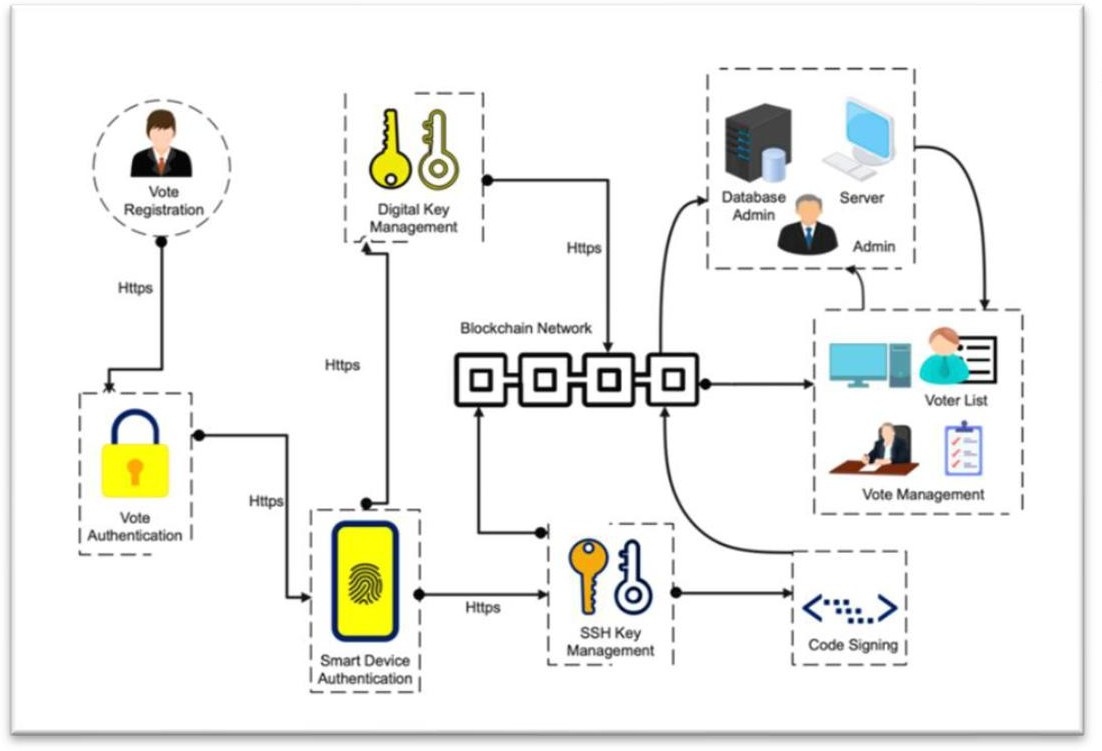
Blockchain works by creating decentralized distributed ledgers that are circulated over all devices participating in the system. It allows to share records based on peer-to-peer replication, and processing by all nodes in the network namely, transacting nodes and validating nodes. When records are placed in the ledger, all nodes in the network go through encryption procedures and are processed by all miners.

Blockchain is applied in both private and government sectors. Usually, an organization has a database where it stores information about their employees and workers. The system will use information from this database so that we can have a list of valid voters and valid candidates in our system the system flowchart for user casting the vote. The user would be directed to a login portal where the user would need to enter some credential to authenticate the user. The user interface will be a web application. Once the user enters the credentials correctly, i.e., the user is successfully able to log in into system, the user would be directed to the voting portal.

In the voting portal, there would be a list of candidates. The voter needs to give preference to the candidates with the lowest number given to the most preferred candidate and the largest number given to the least preferred candidate. In our system, each preferred list by a voter is taken as one transaction.

Thus, whenever a user or voter gives preference to the candidates and submit his list, a transaction is added to the blockchain network.

Thus, each voter would be able to cast only one vote.





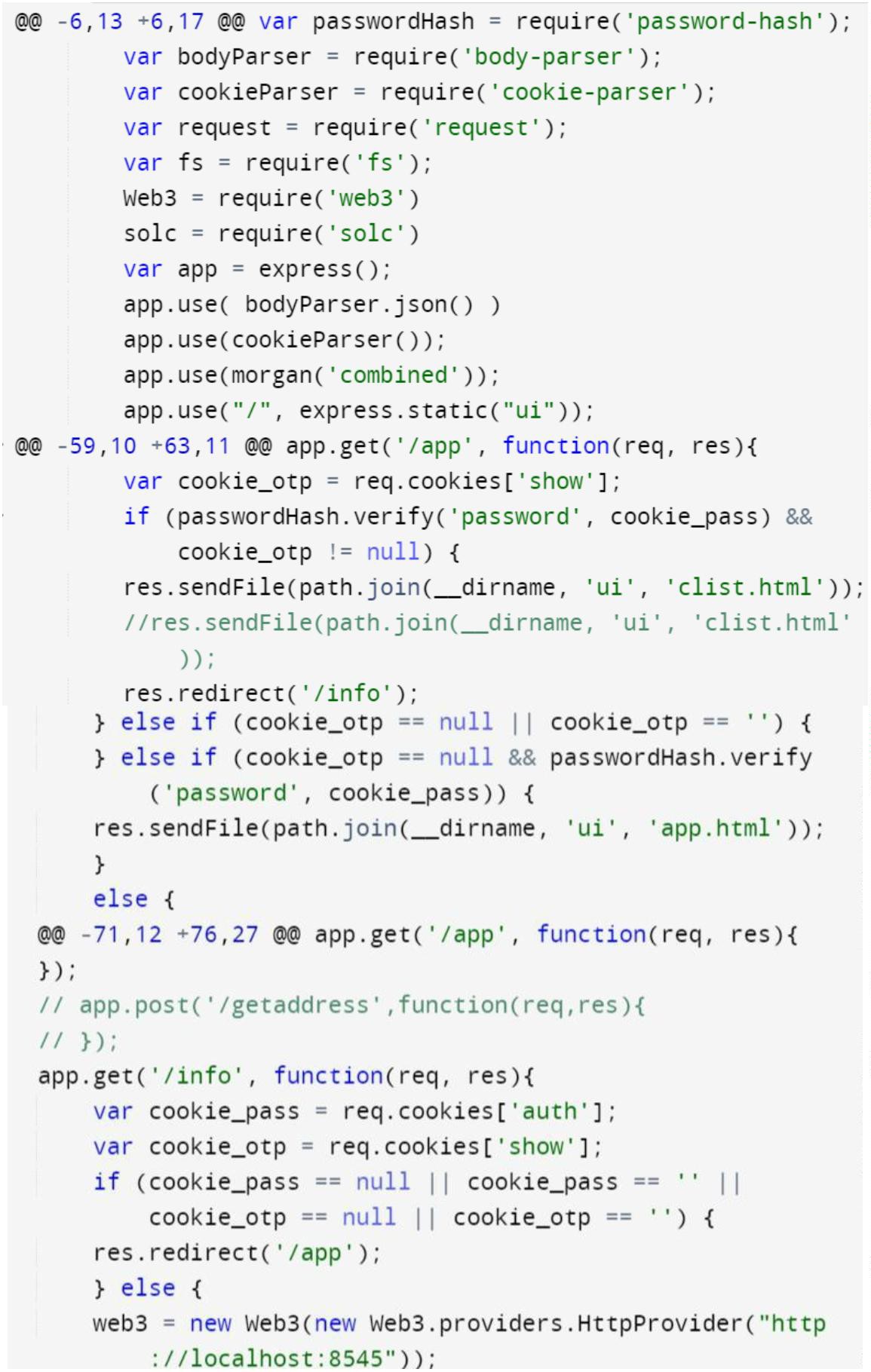
**Description:**

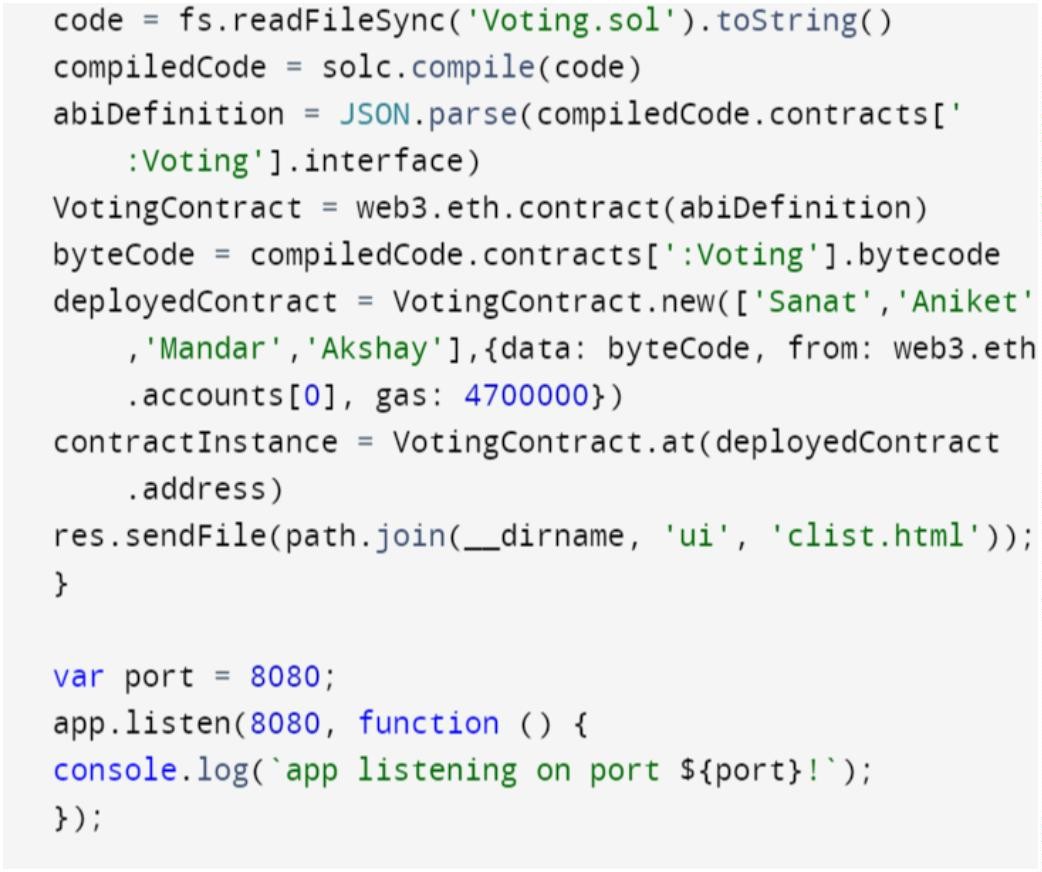


1. The authority must login first with the provided session ID.
2. The voter can now begin the process of voting with proper authentication through OTP (one time password) on the respective linked mobile number.
3. If the voter is valid then the system will check for the voters age and the address to which he can give vote.
4. The voting pallet will be opened with candidate names, their parties and logos.
5. Now the voter can give his vote by clicking vote button.
6. one voter can give his vote only once i.e., after one-time voting buttons are disabled and the vote is automatically logged out.
7. Same process continues for many more voters irrespective of their voting wards.

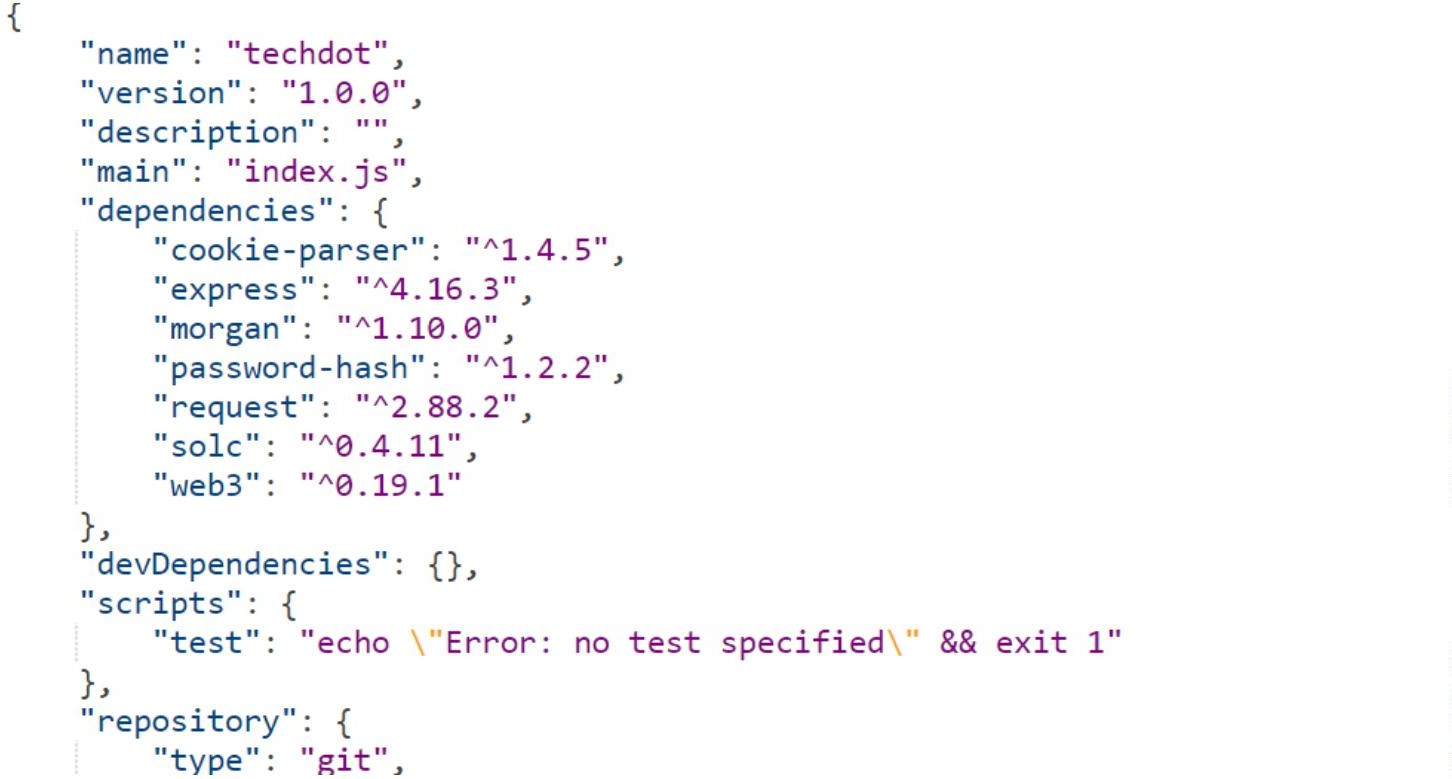
# Implementation:

###### Index.js:



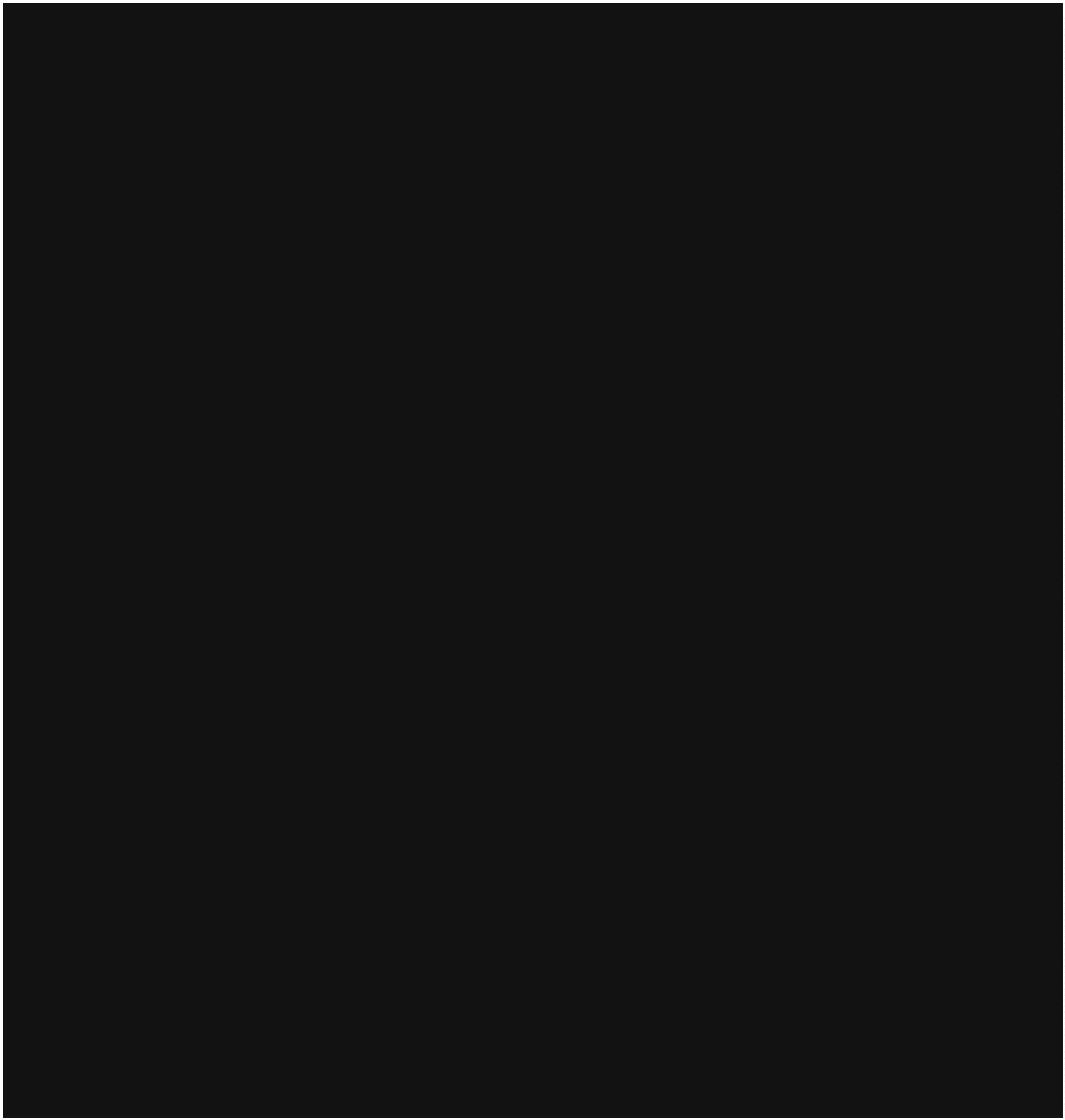


**Package.json:**





**Solidity Program:**

pragma solidity ^0.4.11;

*// We have to specify what version of compiler this code will compile with*

contract Voting {

*/\* mapping field below is equivalent to an associative array or hash.*

*The key of the mapping is candidate name stored as type bytes32 and value is an unsigned integer to store the vote count*

*\*/*

mapping (bytes32 => uint8) public votesReceived;

*/\* Solidity doesn't let you pass in an array of strings in the constructor (yet). We will use an array of bytes32 instead to store the list of candidates*

*\*/*

bytes32[] public candidateList;

*/\* This is the constructor which will be called once when you deploy the contract to the blockchain. When we deploy the contract,*

*we will pass an array of candidates who will be contesting in the election*

*\*/*

function Voting(bytes32[] *candidateNames*) { candidateList = candidateNames;

}

*// This function returns the total votes a candidate has received so far*

function totalVotesFor(bytes32 *candidate*) returns (uint8) { if (validCandidate(candidate) == false) throw;

return votesReceived[candidate];

}

*// This function increments the vote count for the specified candidate. This*

*// is equivalent to casting a vote*

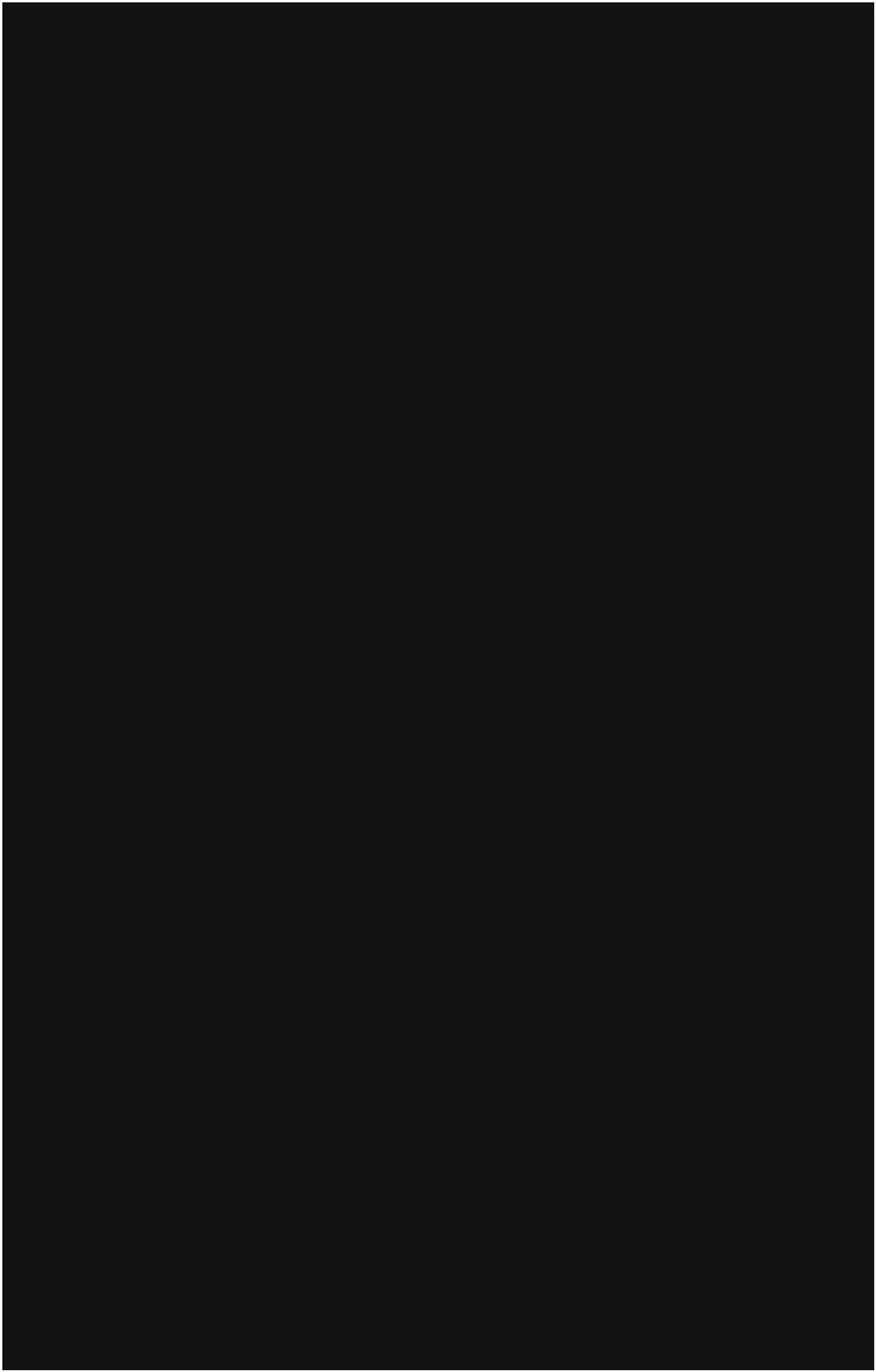
function voteForCandidate(bytes32 *candidate*) { if (validCandidate(candidate) == false) throw; votesReceived[candidate] += 1;

}

function validCandidate(bytes32 *candidate*) returns (bool) { for(uint i = 0; i < candidateList.length; i++) {

if (candidateList[i] == candidate) { return true;

}

}

return false;

}

}

ui/js/app.js :

@@ -34,8 +34,9 @@ $('#errorbox').hide()

});

var aadhaar\_no\_phone\_no = { "738253790005": "9158018030",

"300000000000": "7276478489"

"7382537xxxxx": "915801xxxx", "300000000000": "7276xxxxxx",

"<replace your aadhaar no here>": "<your phone number>",

}

function onSignInSubmit() { window.signingIn = true;

$('#errorbox').hide();

*// updateSignInButtonUI();*

var phoneNumber = "+91" + aadhaar\_no\_phone\_no[$('#aadhaar\_no').val()];

*//console.log(phoneNumber);*

var d = new Date();

d.setTime(d.getTime() + (1\*24\*60\*60\*1000)); var expires = "expires="+ d.toUTCString();

document.cookie = 'aadhaar' + "=" + $('#aadhaar\_no').val() + ";" + expires + ";path=/";

$('#verifyc').text('Enter verification code send to '+phoneNumber) var appVerifier = window.recaptchaVerifier; firebase.auth().signInWithPhoneNumber(phoneNumber, appVerifier)

.then(function (confirmationResult) {

*// SMS sent. Prompt user to type the code from the message, then sign the*

*// user in with confirmationResult.confirm(code).* window.confirmationResult = confirmationResult; window.signingIn = false;

*// updateSignInButtonUI();*

*// $('.verification-code-form').show()*

*// $('#hidepf').hide()*

$('#enter\_aadhaarno').hide()

$('#verify\_otp\_model').show() console.log('otp');

}).catch(function (error) {

*// Error; SMS not sent*

*// $('.main\_loader').hide()*

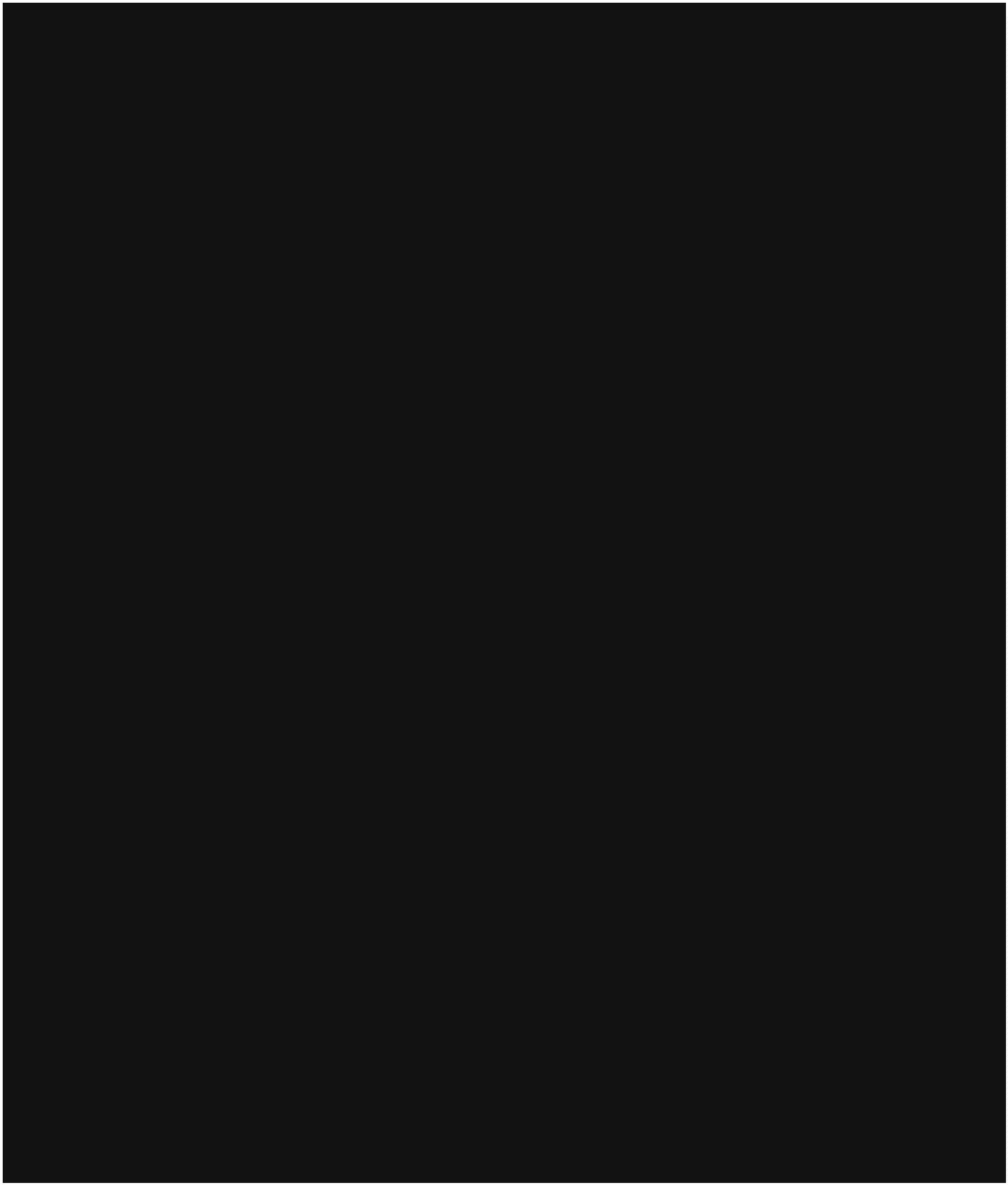
*//console.error('Error during signInWithPhoneNumber', error);*

window.alert('error\n\n'+error); window.signingIn = false;

*//updateSignInFormUI();*

*//updateSignInButtonUI();*

$('.verification-code-form').hide()

});

}

*// Phone auth end //*

$(verifyotp).click(function(){

var code = $('#verify\_otp').val() confirmationResult.confirm(code).then(function (result) {

*// User signed in successfully.* var user = result.user; window.verifyingCode = false;

*//login success* console.log(user.uid); var d = new Date();

d.setTime(d.getTime() + (1\*24\*60\*60\*1000)); var expires = "expires="+ d.toUTCString();

document.cookie = 'show' + "=" + user.uid + ";" + expires + ";path=/"; window.location = '/info'

}).catch(function (error) {

*// User couldn't sign in (bad verification code?)* console.error('Error while checking the verification code', error); window.alert('Error while checking the verification code:\n\n'

+ error.code + '\n\n' + error.message); window.verifyingCode = false;

$('#errorbox').show()

$('#error').text('Enter valid OTP')

});

});

$(getotp).click(function(){

if ($('#aadhaar\_no').val()=="") {

$('#errorbox').show()

$('#error').text('Please Enter Aadhaar No')

}

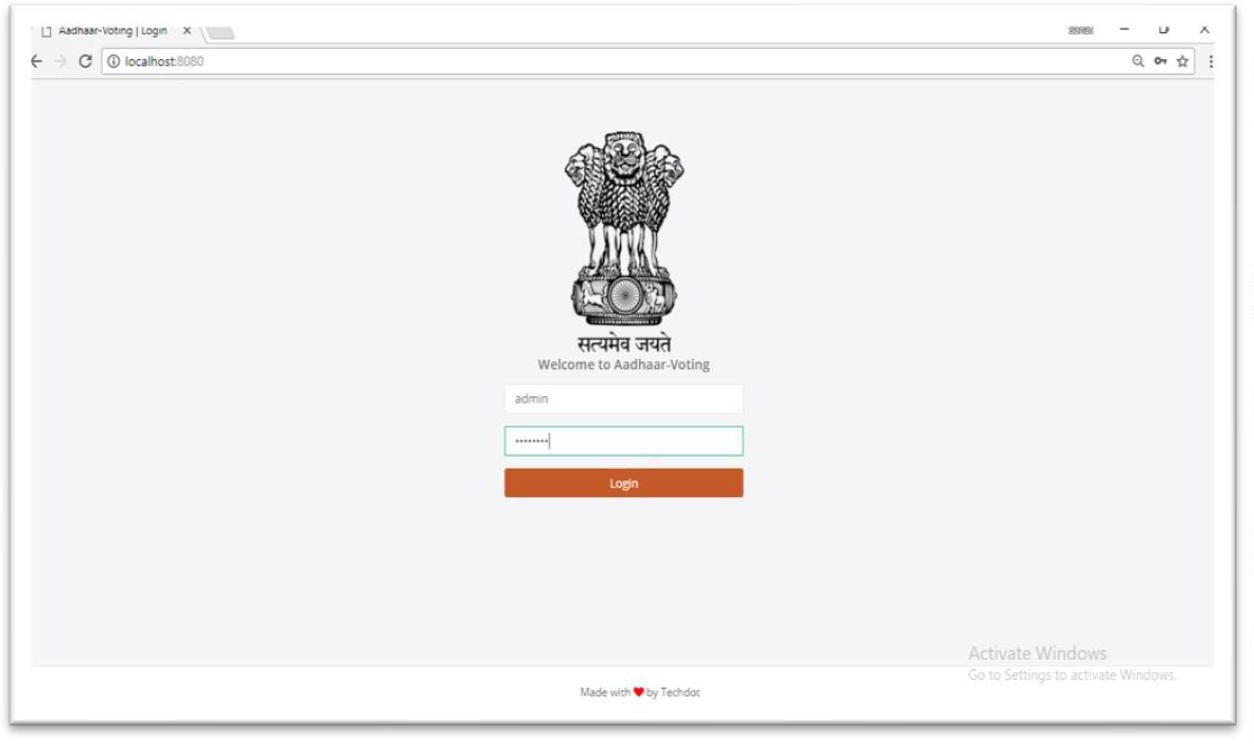
else{ onSignInSubmit();

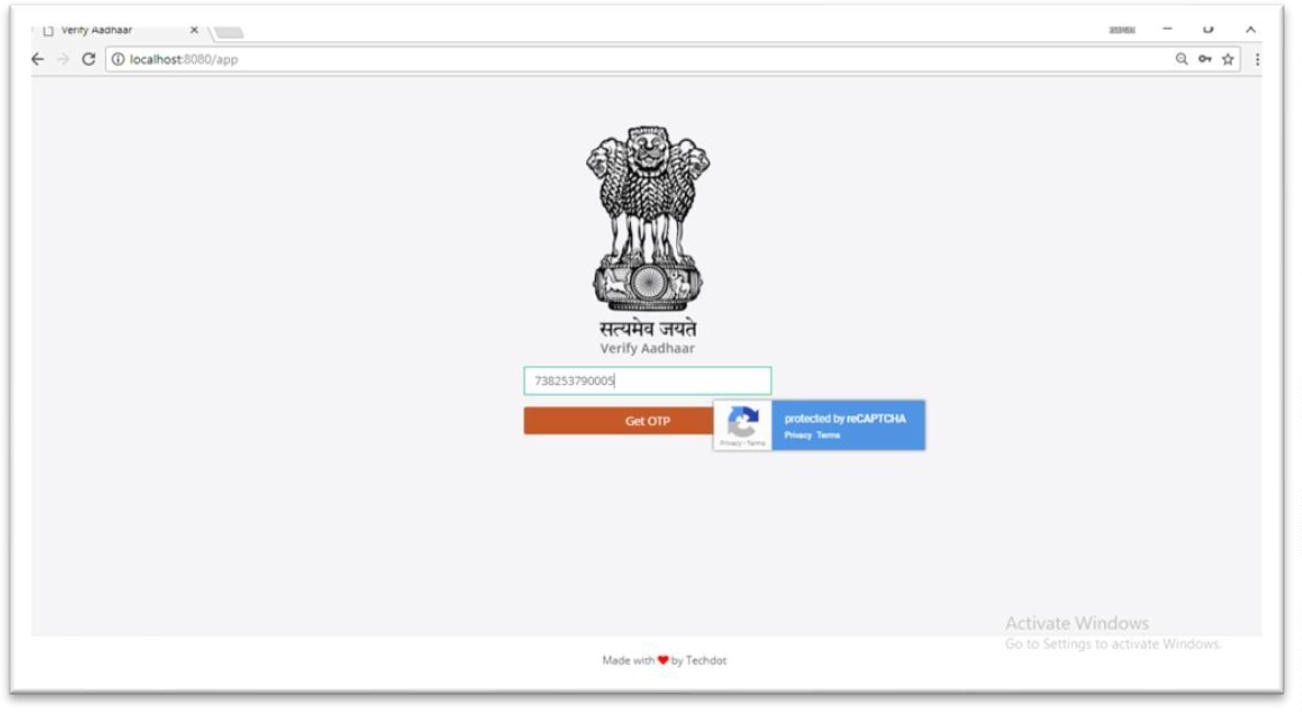
$('#errorbox').hide()

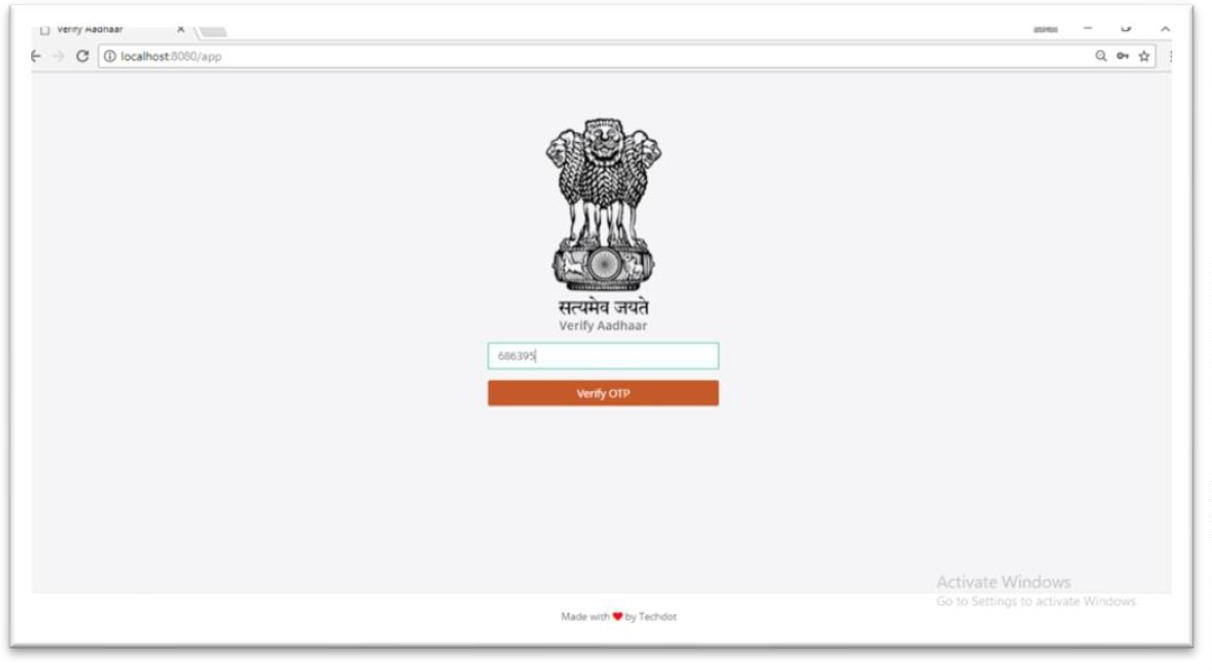
}

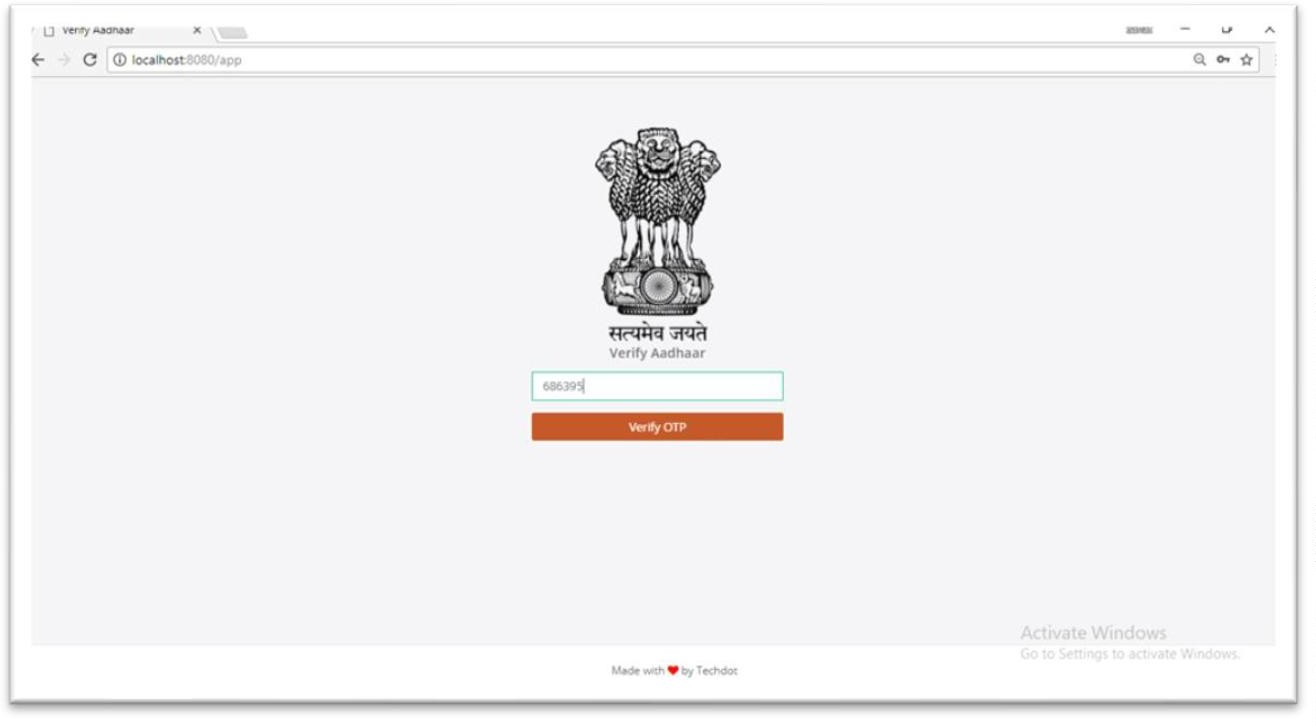
});

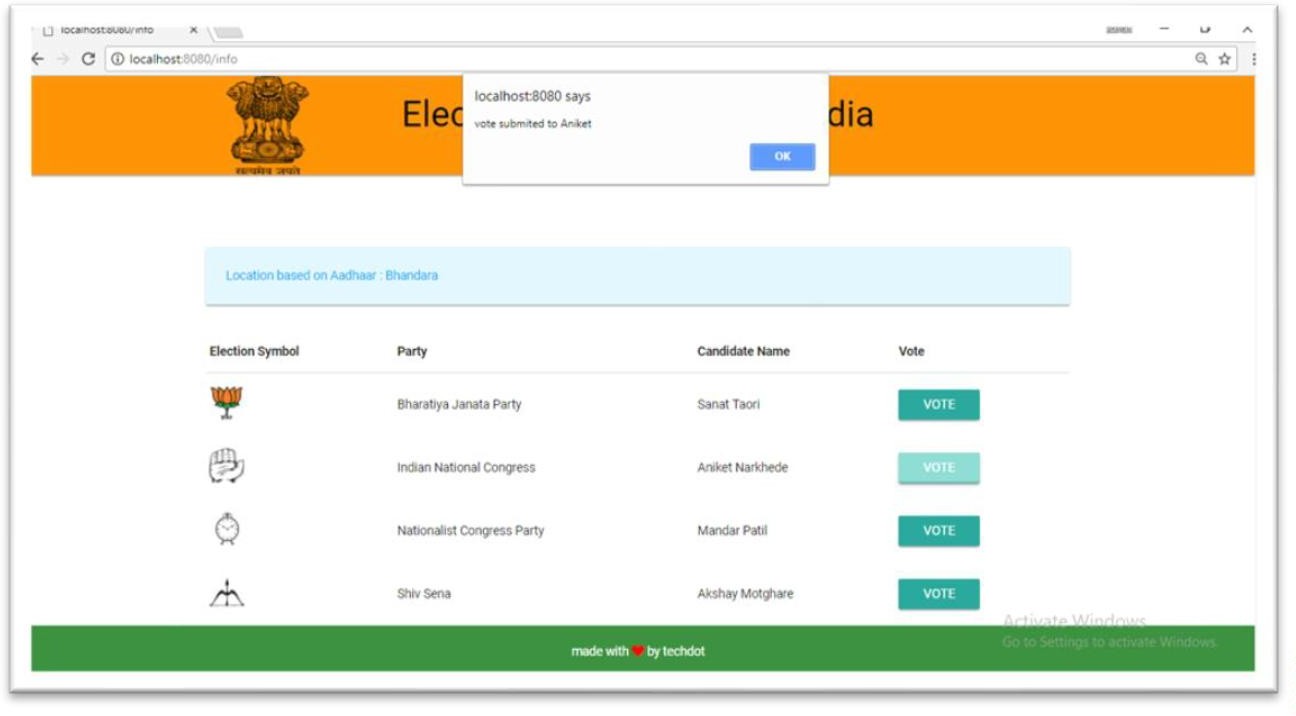
# Result:

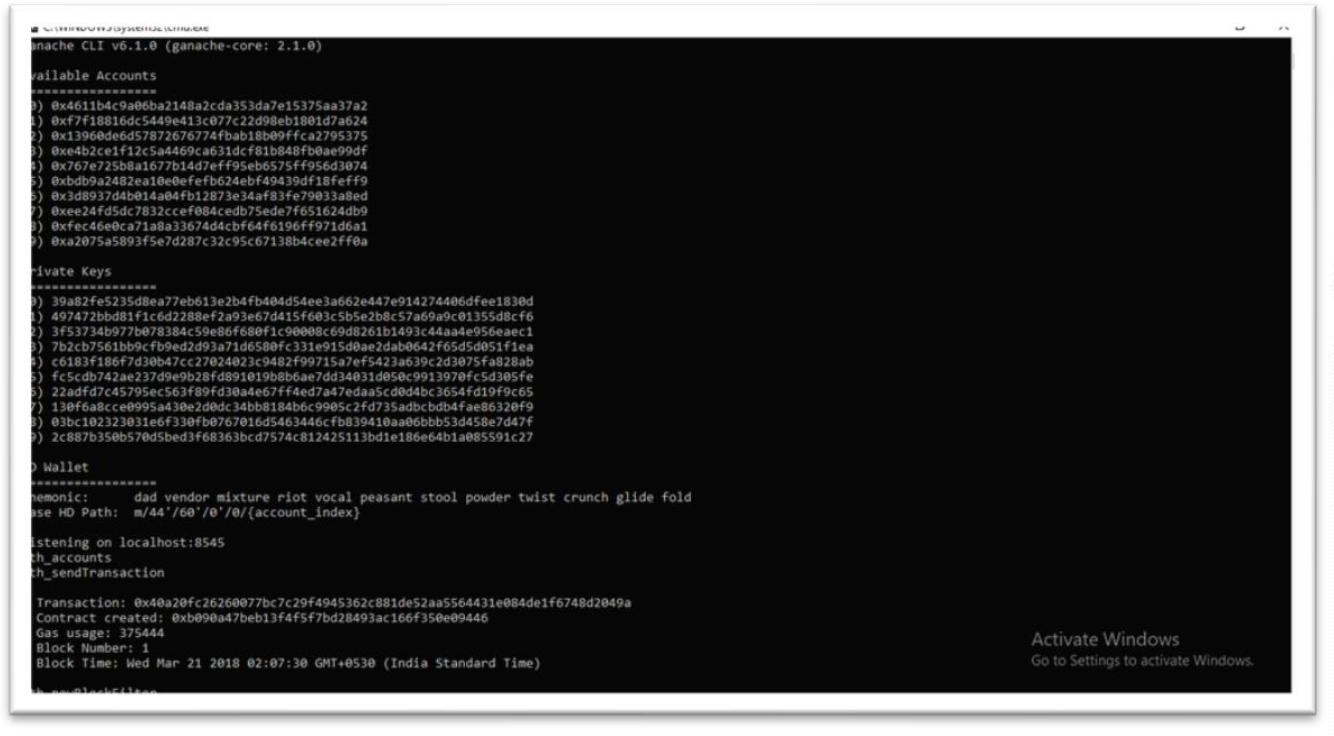












# CONCLUSION:

Our voting system uses the concept of preference-based voting, where instead of giving vote to particular candidate, the voters provide with preference for the candidates. The votes are counted, and if we have a clear majority as defined by the organization, then our system will declare the winner. But if we don’t have clear winner, the votes of the last candidate are distributed according to the preferences given by the voters for that candidate. Thus, the last candidate is removed from the competition and its votes distributed to other candidates according to the preferences given in the votes of the eliminating candidates. This process continues as long as we don’t get a clear majority winner. Ethereum platform has been used which runs smart contract on it and is known to be one of the immensely secured blockchain platform

* 1. **REFERENCES:**



1. “Blockchain Tutorial for Beginners: Learn Blockchain Technology” https://[www.guru99.com/blockchain-tutorial.html](http://www.guru99.com/blockchain-tutorial.html)
2. “Introduction to D-Apps” https://ethereum.org/en/developers/docs/dapps/



[3] “Decentralized Voting System using Blockchain” https://[www.geeksforgeeks.org/decentralized-voting-system-using-blockchain/](http://www.geeksforgeeks.org/decentralized-voting-system-using-blockchain/)

[4] “Blockchain Voting: End to End Process” https://followmyvote.com/blockchain-voting-the end-to-end-process/