A

MINI PROJECT REPORT

ON

Create a DApp (de-centralized app) for e-voting system

By

Kiran Birajdar	**405A037**
Atharv Divate	**405A038**
Pranav Rananaware	**405A039**
Sayali Garole	**405A040**

Under the Guidance of

Prof. M. A. Khade

Department of Computer Engineering
Sinhgad College of Engineering

Vadgaon (Bk), Pune 411041

Accredited by NAAC

Affiliated to Savitribai Phule Pune university, Pune
2024-25

CERTIFICATE

This is certified that the Seminar Report entitled

Create a DApp (de-centralized app) for e-voting system

Submitted by

Atharv Divate

Roll No: 405A038

It is a bonafide work carried out by them under the supervision of **Prof. M. A. Khade** and is approved for the partial fulfillment of the requirement of the **Laboratory Practices III** course in Fourth Year Computer Engineering, in the academic year 2024-2025 prescribed by Savitribai Phule Pune University, Pune.

Prof. M. A. Khade
Project Guide

Dr. M. P. Wankhade
Head, Computer Engineering Department

Dr. S. D. Lokhande
Principal
Sinhgad College of Engineering, Pune

Acknowledgement

I take this opportunity to acknowledge each and every one who contributed towards my work. I express my sincere gratitude towards guide **Prof. M.A. Khade, Assistant** Professor at Sinhgad College of Engineering, Pune, for her valuable inputs, guidance and support throughout the course.

I wish to express my thanks to **Dr. M. P. Wankhade,** Head of Computer Engineering Department, Sinhgad College of Engineering, Pune for giving me all the help and important suggestions all over the Work.

I thank all the teaching staff members, for their indispensable support and priceless suggestions.

I also thank my friends and family for their help in collecting data without which their help Database Management Laboratory report have not been completed. At the end my special thanks to **Dr. S. D. Lokhande,** Principal Sinhgad College of Engineering, Pune for providing ambience in the college, which motivate us to work.

Abstract

This project focuses on creating a decentralized application (DApp) for an e-voting system that addresses the vulnerabilities of traditional centralized voting methods. Centralized systems often face trust issues due to potential data tampering and dependence on a single authority. To overcome these challenges, the project employs blockchain technology, specifically the Ethereum platform, which ensures data integrity, transparency, and immutability through smart contracts. These contracts securely handle the voting process, making it tamper-proof and allowing only one vote per person, while anonymizing voter information to ensure privacy.

The decentralized nature of the proposed system provides a cost-efficient and secure voting method that can be accessed remotely by voters, enhancing voter turnout and participation. Blockchain's cryptographic techniques ensure that votes are securely recorded and verifiable, providing public transparency without compromising individual privacy. This system holds the potential to revolutionize traditional election processes and could be further enhanced with biometric authentication or AI for large-scale government elections.

Contents

1. INTRODUCTION

Voting has always been regarded as the primary method used by individuals to share their opinions on controversial issues and debates. It is a democratic practice, enabling people to formally express their choice against a ballot question, candidate election, political party and others . In every democracy, the security of a election is a matter of national security. With the goal of minimizing the cost of having a national election, while and increasing the security of an election, the computer system has been trying to make electronic voting system more secure . From the dawn of democratically electing candidates, the voting system has been based on pen and paper. The traditional pen and paper election system which fails to provide the voting process traceable and verifiable is replaced by the new election system. Electronic voting system has been viewed as flawed, by the security community, primarily based on physical security concerns. Anyone with physical access to such machine or server can hack or alter the votes, thereby affecting all votes cast on the machine. Blockchain technological features operate through advanced cryptography, providing a security level equal and/or greater than any previously known database. Here we evaluate the use of blockchain as a service to implement anelectronic voting system by following original contributions such as research existing blockchain frameworks suited for constructing blockchain based e-voting system, and propose a blockchain-based electronic voting system that enable liquid democracy. The main contributions of Blockchain are Enforcing voting data immutability and data integrity ensuring robustness and reliability.

2. PROBLEM DEFINITION

Create a DApp (de-centralized app) for e-voting system.

3. Literature Survey

- **Centralized vs. Decentralized Voting Systems**:
  - Traditional centralized voting systems have faced trust issues due to vulnerabilities in data integrity and the need to rely on a central authority. This has motivated a shift towards blockchain-based solutions that eliminate intermediaries and ensure transparency.
- **Blockchain Technology**:
  - Blockchain, as introduced by **Satoshi Nakamoto's Bitcoin whitepaper (2008)**, provides a decentralized, immutable, and secure system for recording transactions, making it ideal for use in e-voting. The cryptographic principles inherent in blockchain prevent tampering and ensure the integrity of the voting process.
- **Ethereum and Smart Contracts**:
  - **Vitalik Buterin's Ethereum whitepaper (2013)** introduced the concept of a decentralized platform capable of executing smart contracts. Ethereum's **smart contracts** allow for automated and secure execution of code when certain conditions are met, making them crucial for the development of secure voting applications.
- **E-voting Systems**:
  - Research into blockchain-based e-voting systems highlights their advantages in security, privacy, and data integrity. A significant contribution from **Zyskind et al. (2015)** emphasizes the decentralized protection of personal data in such systems, ensuring voter privacy while maintaining transparency.

4. Methodology

- **Problem Definition**:

  - The goal is to create a **decentralized application (DApp)** for an e-voting system using the **Ethereum blockchain**. The system aims to address vulnerabilities in traditional e-voting by ensuring data integrity, privacy, and transparency while allowing public visibility into the voting process.

- **Development Environment**:

  - **Programming Language**: Solidity, the primary language for writing smart contracts in Ethereum.
  - **Framework**: Ethereum Virtual Machine (EVM) for deploying and executing smart contracts.
  - **Tools**: Web3.js for interacting with Ethereum, HTML, CSS, and JavaScript for the user interface, and Solidity for backend blockchain logic.

- **System Components**:

  - **Smart Contract**: A Solidity contract (Voting.sol) will manage the voting logic, including adding candidates, casting votes, and displaying results. It ensures that each voter gets one vote and prevents tampering.
  - **Frontend**: A web interface (index.html) allows users to cast votes by interacting with the smart contract through the Web3.js API.
  - **Blockchain**: Ethereum blockchain is used to record votes in an immutable, decentralized manner.

- **Security and Privacy**:

  - Implement advanced cryptographic techniques to secure the voting data. Each vote is anonymized and stored securely on the blockchain, ensuring the privacy of voters while providing public visibility into the total vote counts.

5. Steps and Output

1. Voting.sol file:

```
pragma solidity ^0.6.4;
// 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 => uint256) 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
*/
constructor(bytes32[] memory candidateNames) public {
candidateList = candidateNames;
}
// This function returns the total votes a candidate has received so
far function totalVotesFor(bytes32 candidate) view public returns
(uint256) { require(validCandidate(candidate));
return votesReceived[candidate];
}
// This function increments the vote count for the specified candidate. This
// is equivalent to casting a vote
function voteForCandidate(bytes32 candidate) public
{ require(validCandidate(candidate));
```

```
votesReceived[candidate] += 1;
}
 function
 validCandidate(bytes32
 candidate) view public returns
 (bool) { for(uint i = 0; i <
 candidateList.length; i++) { if
 (candidateList[i] == candidate)
 { return true;
}
}
return false;
}
}
```

2. index.html

```
<!DOCTYPE  html>
<html>
<head>
<title>Hello World DApp</title>
<link href='https://fonts.googleapis.com/css?family=Open
Sans:400,700' rel='stylesheet'
type='text/css'>
<link
href='https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.mi
n.css'
rel='stylesheet' type='text/css'>
</head>
<body class="container">
<h1>A Simple Hello World Voting Application</h1>
<div class="table-responsive">
<table class="table table-bordered">
<thead>
<tr>
```

```
<th>Candidate</th>
<th>Votes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rama</td>
<td id="candidate-1"></td>
</tr>
<tr><td>Nick</td>
<td id="candidate-2"></td>
</tr>
<tr>
<td>Jose</td>
<td id="candidate-3"></td>
</tr>
</tbody>
</table>
</div>
<input type="text" id="candidate" />
<a href="#" onclick="voteForCandidate()" class="btn btn-primary">Vote</a>
</body>
<script src="https://cdn.jsdelivr.net/gh/ethereum/web3.js@1.0.0-
beta.37/dist/web3.min.js"></script>
<script  src="https://code.jquery.com/jquery-3.1.1.slim.min.js"></script>
<script src="./index.js"></script>
</html>
```

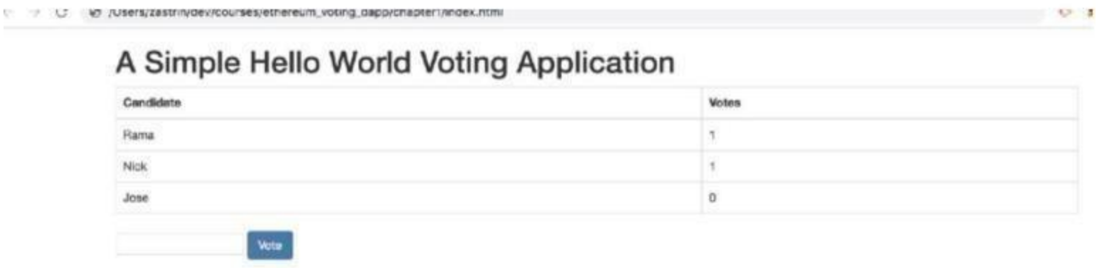
1.index.js

```
web3 = new Web3(new
Web3.providers.HttpProvider("http://localhost:8545")) var account;
web3.eth.getAccounts().then((f)
=> { account = f[0];
```

```
})
abi =
JSON.parse('[{"constant":true,"inputs":[{"name":"candidate","type":"byte
s32"}],"
name":"totalV
otesFor","outputs":[{"name":"","type":"uint8"}],"payable":false,"stateMutabilit
y": "view","typ
e":"function"},{"constant":true,"inputs":[{"name":"candidate","type":"bytes32"
}], "name":"vali
dCandidate","outputs":[{"name":"","type":"bool"}],"payable":false,"stateMutab
ilit y":"view","t
ype":"function"},{"constant":true,"inputs":[{"name":"","type":"bytes32"}],"na
me ":"votesReceived"
,"outputs":[{"name":"","type":"uint8"}],"payable":false,"stateMutability":"vie
w","type":"function"},{"constant":true,"inputs":[{"name":"","type":"uint256"}],"
name":"candidateList","outputs":[{"name":"","type":"bytes32"}],"payable":false,
"stateMutability":"view","type":"function"},{"constant":false,"inputs":[{"name":
"candidate","type":"bytes32"}],"name":"voteForCandidate","outputs":[],"payabl
e":false,"stateMutability":"nonpayable","type":"function"
},{"inputs":[{"name":"candidateNames","type":"bytes32[]"}],"payable":false,"
stateMutabilit y":"nonpayable","type":"constructor"}]')
contract = new web3.eth.Contract(abi);
contract.options.address =
"0x71789831d83d4C8325b324eA9B5fFB27525480b5";
// update this contract address with your contract address
candidates = {"Rama": "candidate-1", "Nick": "candidate-2", "Jose":
"candidate- 3"}
function
voteForCandidate(candidate) {
candidateName =
$("#candidate").val();
console.log(candidateName);
contract.methods.voteForCandidate(web3.utils.asciiToHex(candidateNam
e)).send ({from:
account}).then((f) => {
let div_id = candidates[candidateName];
contract.methods.totalVotesFor(web3.utils.asciiToHex(candidateName)).call().t
```

```
hen((f) => {
$("#" + div_id).html(f);
})
})
}
$(document).ready(function() {
candidateNames =
Object.keys(candidates); for(var i=0;
i<candidateNames.length; i++) { let
name = candidateNames[i];
contract.methods.totalVotesFor(web3.utils.asciiToHex(name)).call().then((f)
=>  {
$("#" + candidates[name]).html(f);
})
}
});
```

Output:



6. CONCLUSION

The decentralized e-voting platform based on Ethereum Blockchain promises a cost-efficient, faster, and more secure alternative to traditional voting systems. It allows voters to participate from any location via the internet while monitoring the integrity of their votes. This solution ensures **voter transparency, accountability, and security** by employing cryptographic principles inherent to blockchain technology.

The system has the potential to revolutionize democratic processes by allowing **direct voting** through a secure online platform, possibly increasing voter turnout. The platform can be enhanced by integrating **fingerprint authentication**, **artificial intelligence (AI)** for facial recognition, and other emerging technologies, making it suitable for large-scale government elections.

7. References

- **Blockchain Technology**:

Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*.
This paper introduces the blockchain concept, which is foundational for decentralized applications like your e-voting system.

- **Ethereum and Smart Contracts**:

Buterin, V. (2013). *Ethereum White Paper: A Next-Generation Smart Contract and Decentralized Application Platform*.
This is a critical resource as Ethereum is the platform you are using for your DApp.

- **E-Voting Systems**:

Zyskind, G., Nathan, O., & Pentland, A. (2015). *Decentralizing Privacy: Using Blockchain to Protect Personal Data*. In *IEEE Security and Privacy Workshops* (pp. 180-184). IEEE.
This paper is useful for understanding how blockchain can be leveraged for secure applications like e- voting.

- **Cryptography in Blockchain**:

Gennaro, R., Gentry, C., Parno, B., & Raykova, M. (2013). *Non-Interactive Verifiable Computing: Outsourcing Computation to Untrusted Workers*. In *Annual Cryptology Conference* (pp. 465-482). Springer. This paper explains cryptographic principles critical to ensuring data integrity and security, which are fundamental to your project's goals.