A

MINI PROJECT REPORT

ON

“Develop a Blockchain based application dApp (de-centralized app) for evoting system.”

In the partial fulfillment of the requirement for the Degree in

Computer Engineering

Submitted By

4133 Dighe Gauri Vishwanath

4134 Dighe Pratik Sitaram

Under the Guidance of

Guide:- Prof. Pandit S. R

Submitted for the Course of Final Year in Computer Engineering for the Practical Lab “**Laboratory Practice III (Blockchain Technology**)”

Department of Computer Engineering

AMRUTVAHINI COLLEGE OF ENGINEERING, SANGAMNER

For the academic year 2024-2025

1. **Title of Mini-Project:** Develop a Blockchain based application dApp (de-centralized app) for E voting system.

2. **Aims/Benefits of Mini-Project**: The aim of this project is to develop a decentralized application (dApp) for an e-voting system using blockchain technology. This system will ensure integrity, immutability, and confidentiality of votes while eliminating the need for a central authority. By leveraging blockchain technology, the project seeks to eliminate election fraud, ensure the integrity of votes, and improve trust in the electoral process.

3. **Course Outcomes:**
   1. Understand and Developed practical experience with dApp development of how blockchain technology works in decentralized applications.
   2. Implemented an e-voting system with a secure and immutable ledger.

4. **Actual Methodology Followed:**

1. System Setup:

- Develop a smart contract on a blockchain platform (e.g., Ethereum).
- Define functions for voter registration, vote casting, and vote tallying.

2. **Voter Registration**:

- Admin registers eligible voters by adding their unique addresses to the blockchain through the smart contract.

3. **Casting Votes**:

- Voters access the decentralized application (dApp), authenticate, and submit their vote.
- The smart contract verifies eligibility, records the vote, and updates the candidate's vote count.

4. **Vote Storage and Security**:

- Votes are stored immutably on the blockchain, ensuring transparency and preventing tampering.

5. **Vote Counting**:

- After the election ends, the smart contract automatically tallies votes, and the results are available for public verification.

6. **Results**:

- The system displays the results, and the winner is declared based on the highest vote count.

Theory:

- **dApp (decentralized application)** is an application that operates on a blockchain or peer-to-peer (P2P) network instead of relying on a central server. Unlike traditional applications (centralized apps), where a single entity manages both the backend (server) and frontend (interface), dApps distribute the backend operations across a decentralized network, making them more secure, transparent, and resistant to censorship.

  Benefits of dApps:
  Security: dApps benefit from the security of blockchain, as data is encrypted and immutable.
  Transparency: All actions and transactions are recorded on the blockchain and can be publicly verified.
  No Single Point of Failure: Decentralization ensures higher availability and resilience.
  Trustless Interactions: Users do not need to trust a central authority, as the smart contracts automatically execute operations.

- **E-Voting System:** An e-voting system is an electronic system used to facilitate voting, allowing voters to cast their votes digitally rather than through traditional paper ballots. E-voting systems can be used in various elections, including government elections, corporate decision-making, or even community votes, providing convenience, speed, and accuracy in the voting process. There are two main types of e-voting systems:

1. **On-site e-voting:** Voters cast their votes at polling stations using electronic machines (e.g., voting kiosks or Direct Recording Electronic (DRE) systems).
2. **Remote e-voting:** Voters can vote from any location via a secure web or mobile application, commonly used for absentee voting or in environments where voters are spread across wide regions.

Types of E-voting Systems

1. **Traditional E-voting Systems:** These systems involve centralized servers that handle voter authentication, vote recording, and storage. Direct-Recording Electronic (DRE) machines: Voting machines used at polling stations where voters can directly cast their votes electronically. Votes are stored on local servers or databases that can be accessed by election authorities for counting.

2. **Blockchain-based E-voting Systems:** A blockchain is a decentralized, immutable ledger, making it an ideal solution for secure e-voting. Each vote is treated as a transaction, which is recorded on a blockchain. Once recorded, it cannot be changed, ensuring transparency and security. Smart contracts automatically handle tasks like voter registration, vote counting, and result declaration. Decentralization ensures that no single entity controls the system, making it resistant to manipulation or hacking. Tamper-proof: Votes cannot be changed once submitted, preventing election fraud.

5. Program/Code:

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

contract VotingContract {
    address public owner;
    mapping(string => bytes32) public candidateId; // Candidate's name to their
    hashed ID
    mapping(string => uint) public candidateVotes; // Candidate's name to their
    vote count
    string[] public candidates; // List of candidates
    mapping(address => bool) public hasVoted; // Track if an address has voted

    constructor() {
        owner = msg.sender; // Set the contract deployer as the owner
    }

    // Function to receive Ether. msg.data must be empty
    receive() external payable {}

    // Function to represent a candidate
    function representate(string memory _candidateName, uint _age, string
    memory _candidateId) public {
        require(msg.sender == owner, "Only the owner can represent
    candidates");
        require(candidateId[_candidateName] == 0, "Candidate already exists");

        bytes32 candidateHash = keccak256(abi.encodePacked(_candidateName,
    _age, _candidateId));
        candidateId[_candidateName] = candidateHash;
        candidates.push(_candidateName);
    }
```

```
// Function to vote for a candidate
function vote(string memory _candidateName) public {
    require(!hasVoted[msg.sender], "You have already voted");
    require(candidateId[_candidateName] != 0, "Candidate does not exist");

    candidateVotes[_candidateName]++;
    hasVoted[msg.sender] = true; // Mark the voter as having voted
}

// Function to get the vote count of a candidate
function getVotes(string memory _candidateName) public view returns
(uint) {
    require(candidateId[_candidateName] != 0, "Candidate does not exist");
    return candidateVotes[_candidateName];
}

// Function to get the list of candidates
function getCandidates() public view returns (string[] memory) {
    return candidates;
}

// Function to check if an address has voted
function checkIfVoted() public view returns (bool) {
    return hasVoted[msg.sender];
}

// Function to predict the winner
function predictWinner() public view returns (string memory winnerName)
{
    uint highestVotes = 0;

    for (uint i = 0; i < candidates.length; i++) {
        string memory candidateName = candidates[i];
        uint votes = candidateVotes[candidateName];

        if (votes > highestVotes) {
            highestVotes = votes;
            winnerName = candidateName; // Update the winner's name
        }
    }
    // If no votes have been cast, return an empty string
    if (highestVotes == 0) {
        return "";
    }
}
```

6. Conclusion:

- By developing a blockchain-based decentralized application (dApp) for an e-voting system, we aim to revolutionize the traditional voting process by ensuring security, transparency, and immutability. Blockchain's decentralized nature will eliminate the need for central authorities, reducing the risk of manipulation and fraud. The use of smart contracts will automate the voting and tallying processes, enhancing trust among participants.