Blockchain Tech. Mini-Project Report

Blockchain-Based Decentralized Application

Name: Parth Ingle (41227)
Mukund Karwa (41234)
Soham Lagad (41241)
Batch: Q2

Title

Development of a Blockchain-based Decentralized Application (dApp) for E-Voting System

Problem Definition

Traditional electronic voting systems are often centralized, creating vulnerabilities such as data tampering, lack of transparency, and dependence on a single trusted authority.
This project aims to design a **Blockchain-based decentralized e-voting system** to ensure **transparency, security, immutability, and voter anonymity** using **Ethereum smart contracts**.

Learning Objective

- Understand the fundamentals of Blockchain and smart contracts.
- Learn how to design and deploy decentralized applications (dApps).
- Implement secure and transparent vote storage using Blockchain technology.
- Explore Ethereum, Solidity, and Web3.js for decentralized development.

Learning Outcome

- Ability to design and deploy a simple blockchain network for real-world applications.
- Hands-on experience with **smart contract programming** in Solidity.
- Understanding of how Blockchain ensures **immutability and trustless execution**.
- Experience in connecting smart contracts with frontend using **Web3.js or React.js**.

Theory-Related Concepts, Architecture, Syntax

Concepts Used

1. **Blockchain:** A distributed ledger that records transactions securely and immutably.
2. **Smart Contract:** Self-executing code deployed on Blockchain (written in Solidity).
3. **Ethereum Network:** Public blockchain used for deploying decentralized applications.
4. **Web3.js:** JavaScript library to interact with the Ethereum blockchain.
5. **MetaMask:** Wallet for connecting users securely to the Ethereum blockchain.

System Architecture

Frontend: HTML/CSS/React for user interface (voter registration & vote casting).
Smart Contract: Solidity code deployed on Ethereum test network.
Backend: Node.js server to interact between UI and blockchain network. **Database (Optional):** IPFS for decentralized storage of metadata.

Workflow:

1. Admin registers candidates.
2. Voters log in using MetaMask wallet.
3. Voters cast their votes (recorded on blockchain).
4. Votes are tallied automatically by the smart contract.
5. Results displayed transparently — cannot be altered.

Test Cases

Test Case ID	Input	Action / Feature	Expected Output
TC1	New voter registration	Check if wallet address already exists	Duplicate registration prevented
TC2	Casting vote	Smart contract records vote	Vote stored immutably on blockchain
TC3	Attempt to vote twice	Smart contract validation	Second vote rejected
TC4	View result before deadline	Admin-only access	Access denied

TC5	View final results	End of voting period	Transparent, verifiable results shown

Program Listing

```
// SPDX-License-Identifier: MIT pragma

solidity ^0.8.0;


contract E_Voting {

  struct Candidate {

  uint id; string

  name; uint

  voteCount;

  }


  struct Voter {

    bool voted;

    uint vote;

  }


  address public admin; mapping(address

  => Voter) public voters; Candidate[]

  public candidates;

  bool public votingOpen;


  constructor(string[] memory candidateNames) { admin =

    msg.sender; for (uint i = 0; i < candidateNames.length;
```

```
    i++) { candidates.push(Candidate(i, candidateNames[i],
    0));
   }
   votingOpen = true;
  }

  function vote(uint candidateId) public {
    require(votingOpen, "Voting closed");
    require(!voters[msg.sender].voted, "Already voted");
    voters[msg.sender].voted = true;
    voters[msg.sender].vote = candidateId;
    candidates[candidateId].voteCount++;
  }

  function endVoting() public { require(msg.sender == admin,
    "Only admin can close voting"); votingOpen = false;
  }

  function getResults() public view returns (Candidate[] memory) {
    require(!votingOpen, "Voting not ended"); return candidates;
  }
}
```

Output

- Voter successfully connects using MetaMask.
- Votes recorded securely on blockchain.
- Smart contract ensures **no double voting**.
- Final results automatically generated when voting ends.

- Verified immutability of results using blockchain explorer (Etherscan testnet).

Conclusion

The decentralized e-voting dApp demonstrates how **Blockchain** can revolutionize traditional voting systems by offering **security, transparency, and trust** without relying on a central authority.

The project showcases the integration of **Ethereum, smart contracts, and Web3.js**, building a strong foundation for secure digital governance solutions.