VOTING SYSTEM

ADVANCE BLOCKCHAIN PROJECT

Radhika Pisipati - 1RVU22CSE126

Saanvi R Prabhu - 1RVU22CSE136

Suhani Lalla- 1RVU22CSE167

Declaration By Authors

We, the undersigned authors of this report, hereby affirm the authenticity and originality of the work presented herein. By signing this declaration, we confirm the following:

1. The attached report on the topic **Decentralized Voting System** and its contents are the result of our own independent efforts and research.
2. Any sources of information utilized in this report have been appropriately acknowledged through citations.
3. The code, design, and implementation details presented in this report are original and have not been copied or reproduced from any other source.
4. We have adhered to all relevant ethical guidelines and regulations in conducting this project.

Signed:

Radhika Pisipati

Saanvi R Prabhu

Suhani Lalla

Decentralized Voting System

The Blockchain Solution

ABSTRACT

The Blockchain-Based Voting System project aims to develop a secure and transparent platform for digital voting using blockchain technology. Based on Ethereum smart contracts, the system enables voters to cast their votes securely and transparently, with all transactions recorded immutably on the blockchain. The web interface provides intuitive access to candidate information and voting functionalities. Deployment to the Sepolia testnet is facilitated using the ethers.js library. Results demonstrate the system's effectiveness in enhancing the integrity and transparency of electoral processes, showing the transformative potential of blockchain in democratizing voting systems.

TABLE OF CONTENTS

INTRODUCTION

Since the entire world is quickly moving towards digitalization, integrity and transparency in election processes are more important than ever. Conventional voting methods frequently face issues like fraud susceptibility, lack of transparency, and security flaws. Given this, there is a rising interest in using blockchain technology to improve voting procedures and increase public confidence in democracy. The Blockchain-Based Voting System project aims to address issues related to creating a decentralized, transparent, and safe digital voting platform. By incorporating blockchain technology, which is transparent, immutable, and resistant to manipulation, the initiative seeks to both protect the integrity of the electoral process and give voters greater transparency and confidence in the results of elections.

OBJECTIVE

By implementing a blockchain-based voting solution, the project seeks to achieve the following key objectives:

- Implement a secure, transparent, and decentralized platform for digital elections.
- Ensure transparency and auditability of the voting process
- Improve accessibility and inclusivity through a user-friendly web interface and integration with Ethereum wallets.
- Build trust and confidence among stakeholders by providing a verifiable and tamper-proof voting system.

METHODOLOGY

In the development of our Blockchain-Based Voting System, we followed a structured methodology to ensure the successful implementation of key functionalities. The methodology has the following stages:

1. Smart Contract Design and Development:

a. **Define Smart Contract Structure**: The initial phase involved defining the structure of the smart contract. This included creating the CandidateInfo struct to store candidate details and outlining functions for voting, retrieving candidate information, and displaying voting tallies.

b. **Implement Core Functionalities:** With the contract structure defined, the focus shifted to implementing core functionalities such as enabling voters to cast their votes securely and preventing duplicate voting.

c. **Integrate Smart Contract with Ethereum Blockchain:** The completed smart contract was then integrated with the Ethereum blockchain, ensuring that all voting transactions were recorded immutably on the blockchain.

d. **Verify Smart Contract Logic:** Rigorous testing and verification were conducted to ensure the correctness and reliability of the smart contract logic before proceeding to the next stage.

2. Web Interface Development:

a. **Design User Interface Mockups:** The web interface design phase involved creating mockups and wireframes to visualize the user interaction flow. This step ensured that the interface would be intuitive and user-friendly.

b. **Develop Frontend Components:** Following the design phase, frontend components such as buttons, forms, and data displays were developed using HTML, CSS, and JavaScript. These components were designed to seamlessly integrate with the backend smart contract.

c. **Implement MetaMask Integration:** Integration with MetaMask was implemented to allow users to connect their Ethereum wallets securely to the voting system, enabling them to cast votes directly from their accounts.

CODE FILES: DETAILED OVERVIEW

This chapter provides an overview of the five code files that are essential to our Blockchain-Based Voting System project, along with information about the purpose and functionalities of each file.

votee.sol

- ***Purpose*****:** This Solidity file contains the core logic of our voting system through a smart contract.
- ***Functionality*****:** It defines the *VotingSystem* contract, which includes a *CandidateInfo* struct to store candidate information such as ID, name, phone number, and total votes. The contract features functions for voting, retrieving candidate details, obtaining vote counts, and checking if a voter has already cast a vote. Additionally, it utilizes a mapping to associate voter addresses with their respective voter IDs, ensuring accountability and preventing duplicate votes.

deploy.js

- ***Purpose*****:** Serving as a JavaScript deployment script, this file integrates the deployment of the voting system smart contract onto the Sepolia testnet.
- ***Functionality***: Leveraging the ethers.js library, the script initializes an Ethereum provider and wallet, reads the compiled contract's ABI and bytecode, creates a contract factory, deploys the contract, and logs deployment details including the transaction status and contract address. This streamlined deployment process ensures seamless integration with the testnet environment.

index.html

- ***Purpose***: Serving as the primary HTML interface, this file provides users with access to voting system functionalities.
- ***Functionality***: It features interactive elements such as candidate selection dropdowns, buttons for voting, and displays for candidate information and vote counts. The HTML file incorporates JavaScript libraries for Ethereum interaction, enabling seamless communication with the blockchain backend. Additionally, it includes MetaMask integration for secure connection of Ethereum wallets, facilitating direct voting from user accounts.

script.js

- ***Purpose***: This JavaScript file contains client-side code responsible for orchestrating user interactions with the web interface.
- ***Functionality***: It defines functions for loading candidate names, retrieving candidate details and vote counts, casting votes, connecting wallets via MetaMask, and verifying voter participation status. The script utilizes the *Web3.js* library to interact with the Ethereum blockchain, ensuring accurate data retrieval and transaction

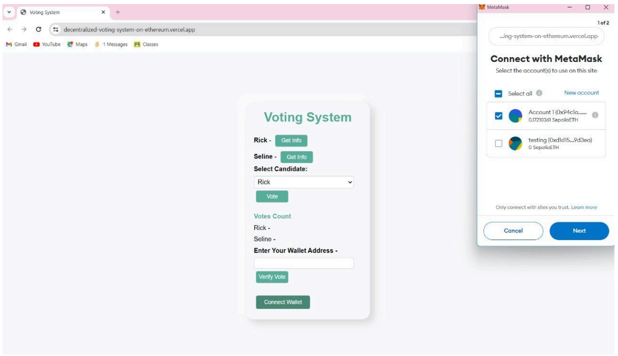
execution. Error handling mechanisms are incorporated to enhance user experience and system reliability.

styles.css

- ***Purpose***: This CSS file defines the visual styling for HTML elements of the web interface.
- ***Functionality***: It sets styles for various elements including containers, headings, buttons, input fields, and forms, ensuring a visually appealing and cohesive layout for the voting system interface. The CSS file employs responsive design techniques to adapt the interface to different screen sizes, enhancing user engagement and navigational intuitiveness.

RESULT

Connect Wallet - Click the "Connect Wallet" button to connect your Ethereum wallet securely to the voting system.



decentralized-voting-system-on-ethereum.vercel.app

Voting System

Rick -

Seline -

Select Candidate: Rick

Votes Count

Rick -

Seline -

Enter Your Wallet Address -

Connect with MetaMask

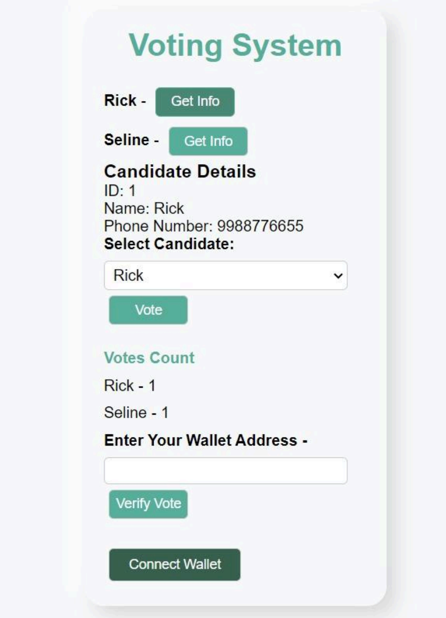
Select the account(s) to use on this site

☑ Account 1 (0x94c1a...... 0.17210361 SepoliaETH

☐ testing (0xd1d15...9d3ea) 0 SepoliaETH

Only connect with sites you trust. Learn more

Get Candidate Information - Click the "Get Info" button next to each candidate's name to retrieve detailed information about each candidate, including their ID, name, and phone number.



Voting System

Rick -

Seline -

Candidate Details
ID: 1
Name: Rick
Phone Number: 9988776655

Select Candidate: Rick

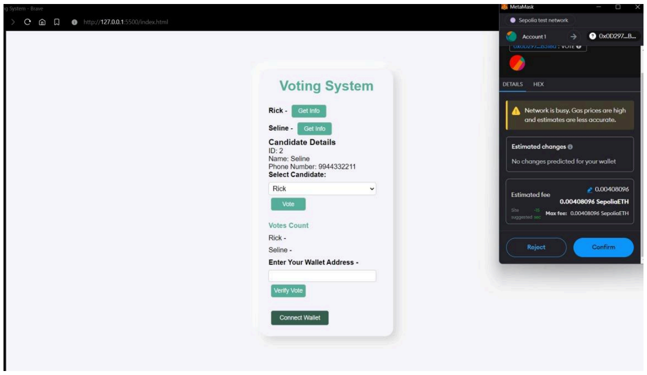
Votes Count

Rick - 1

Seline - 1

Enter Your Wallet Address -

Select Candidate and Vote - Choose a candidate from the dropdown list under "Select Candidate," then click the "Vote" button to cast your vote securely. Ensure you are connected to your Ethereum wallet before voting.



http://127.0.0.1:5500/index.html

Voting System

Rick -

Seline -

Candidate Details
ID: 2
Name: Seline
Phone Number: 9944332211

Select Candidate: Rick

Votes Count

Rick -

Seline -

Enter Your Wallet Address -

MetaMask

Sepolia test network

Account 1 → 0x0D297...B...

Network is busy. Gas prices are high and estimates are less accurate.

Estimated changes

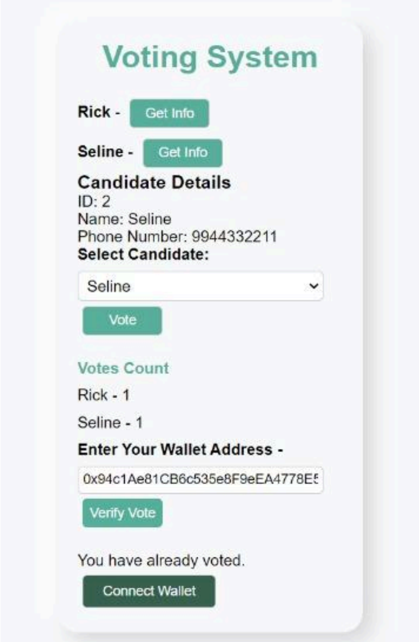
No changes predicted for your wallet

Estimated fee 0.00408096

0.00408096 SepoliaETH

Max fee: 0.00408096 SepoliaETH

Check your voting status and view the current vote counts for each candidate. Simply enter your wallet address in the designated field and click "Verify Vote" to see if you've already voted.



Voting System

Rick -

Seline -

Candidate Details
ID: 2
Name: Seline
Phone Number: 9944332211
Select Candidate:

Seline

Votes Count

Rick - 1

Seline - 1

Enter Your Wallet Address -

0x94c1Ae81CB6c535e8F9eEA4778E5

You have already voted.

Once verified, the system will display the updated vote counts for Rick and Seline under the "Votes Count" section.

APPENDICES

Appendix A: The Smart Contract

Navigate to contracts/votee.sol in the github link given below.

Appendix B: The Deployment Script

Access the deploy.js file in the GitHub repository.

Appendix C: HTML and CSS Files

Access the HTML and CSS files in the GitHub repository

- HTML Interface (index.html)
- CSS Styling (style.css)

Appendix D: The JavaScript Code

Access the script.js file in the GitHub repository

REFERENCES

1. F. Þ. Hjálmarsson, G. K. Hreiðarsson, M. Hamdaqa and G. Hjálmtýsson, "Blockchain-Based E-Voting System," 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), San Francisco, CA, USA, 2018, pp. 983-986, doi: 10.1109/CLOUD.2018.00151. keywords: {Contracts;Electronic voting;Peer-to-peer computing;Privacy;Electronic voting systems;Blockchain;E-Voting;Voting;Smart Contract;Private Blockchain}

2. Jun Huang, Debiao He, Mohammad S. Obaidat, Pandi Vijayakumar, Min Luo, and Kim-Kwang Raymond Choo. 2021. The Application of the Blockchain Technology in Voting Systems: A Review. ACM Comput. Surv. 54, 3, Article 60 (April 2022), 28 pages. https://doi.org/10.1145/3439725

3. A. Singh and K. Chatterjee, "SecEVS : Secure Electronic Voting System Using Blockchain Technology," 2018 International Conference on Computing, Power and Communication Technologies (GUCON), Greater Noida, India, 2018, pp. 863-867, doi: 10.1109/GUCON.2018.8675008. keywords: {Blockchain;Electronic voting systems;Electronic voting;Cryptography;Password;E-votin

g system;Block-chain;Hashing;Merkle hash}

4. https://github.com/smartcontractkit/full-blockchain-solidity-course-js?tab=readme-ov-file