Project Report on

Block-Chain Based Online Voting System

Software Development Project-IV

Supervised By

DR. TUSHAR KANTI SAHA

Professor

Dept. of Computer Science & Engineering

Jatiya Kabi Kazi Nazrul Islam University, Trishal, Mymensingh

Submitted by

Sakib Ahamed Shahon	*Mehedi Khan Rakib*
Roll: 19102018	Roll: 19102012
Session: 2018-2019	Session: 2018-2019

Submission Date: 25 January , 2024

The undersigned have examined the report entitled “BlockChain based Online Voting System” presented by Sakib Ahamed Shahon and Mehedi Khan Rakib are members of this project.

Advisor’s name

____

Dr. Tushar Kanti Saha

Date: 25-01-2024

CANDIDATE’S DECLARATION

We declare that this project is our own work and has not been copied from any source or web. Information & knowledge derived from web and books has been acknowledged in the text and a list of references is given.

Date: 25/01/2024

Signature of the Candidates

Sakib Ahamed Shahon

Mehedi Khan Rakib

Dedication

Firstly, we remember the Almighty who enhances our knowledge. Secondly, we dedicate our project to our parents and we are really grateful to our honorable supervisor Professor Dr. Tushar Kanti Saha.

Acknowledgments

A project is a golden opportunity for a student to learn and self-development. We consider ourselves very lucky and honored to have so many wonderful people who led us in the completion of this project.

Our thanks and gratitude to **Prof. Dr. Tushar Kanti Saha** who despite being extraordinarily busy with his duties, took time out to hear, guide, and keep us on the correct path. We do not know where we would have been without him.

We are also thankful to **Prof. Dr. Md. Sheikh Sujan Ali**, Head of the department, CSE for giving us a lab facility to accomplish this project.

Last but not least there were so many who shared valuable information that helped in the successful completion of this project.

Abstract

The Blockchain-Based Online Voting System (BOVS) stands as a pioneering software initiative with the transformative goal of revolutionizing the electoral process through the integration of blockchain technology. This innovative system offers a secure, transparent, and tamper-resistant platform for online voting, thereby elevating both accessibility and trust in the democratic process. The development of BOVS was underpinned by a thorough analysis of functional and non-functional requirements. Leveraging the Ethereum blockchain, the system employs smart contracts to efficiently manage user registration, voting procedures, and the tabulation of votes. Key elements include a user-friendly web interface, a dedicated private Ethereum blockchain network, and robust security measures such as real-time face recognition authentication. Additionally, the system boasts a feature for displaying real-time voting results. The Blockchain-Based Online Voting System represents a significant stride in modernizing electoral procedures, with the potential to enhance remote voter participation and foster trust in democratic institutions. As we continue to refine and improve the system, our unwavering commitment remains focused on delivering a robust online voting platform that empowers citizens and fortifies democratic foundations.

Table of contents

1. Introduction

1.1 Motivation

In an era characterized by rapid technological progress, safeguarding the integrity of the electoral process assumes paramount significance. The impetus behind the inception of the Blockchain-Based Online Voting System (BOVS) arises from the imperative to systematically address the inherent challenges and vulnerabilities inherent in traditional online and offline voting methodologies. Conventional approaches to casting and tallying votes frequently grapple with issues pertaining to security, transparency, and accessibility. Regrettably, instances of fraud, manipulation, and operational inefficiencies have eroded public confidence in the democratic process. The escalating demand for remote voting alternatives, particularly during times of crisis, serves to underscore the critical necessity of reenvisioning the methodologies employed in conducting elections.

1.2 Purpose of the Project

The fundamental objective of the BlockChain based online voting system project is to leverage the paradigm-shifting capabilities of blockchain technology for the development of a secure, transparent, and tamper-resistant online voting system. Our overarching goal is to reinstate and fortify public trust in the electoral process, foster increased participation in democratic decision-making, and inaugurate an era where citizens can exercise their voting rights conveniently, securely, and globally. Through this project, we aim to exemplify that technology can not only elevate the voting experience but also contribute substantially to the underpinnings of democratic principles.

1.3 Scope of the Project

The scope of the Blockchain-Based Online Voting System is extensive, covering various dimensions of the electoral process. The project encompasses:

1. **User Registration:** Establishment of a secure and user-friendly platform facilitating the registration of eligible voters through the involvement of an electoral body.

2. **Voter Verification:** Implementation of robust verification processes, ensuring voter eligibility through private-public key-based authentication and real-time facial recognition authentication.

3. **Ballot Creation:** Development of a versatile system empowering election authorities to create and tailor digital ballots for diverse elections.

4. **Voting:** Provision of a secure online platform for registered voters to cast their votes, ensuring both anonymity and integrity in the process.

5. **Vote Verification:** Empowering voters to verify the accurate recording of their votes.

6. **Vote Counting:** Designing a tamper-resistant mechanism for vote counting, elevating transparency in the electoral process.

7. **Results Publication:** Real-time display of election results while preserving the privacy of voters.

The project's scope transcends technical considerations, encompassing aspects of security, usability, scalability, and reliability.

1.4 Outline

This project report is structured into several sections to provide a comprehensive understanding of the Blockchain-Based Online Voting System (BOVS). Chapter 2 specifies the functional requirements and non-functional requirements. We discuss our methodology to develop the system in Chapter 3. In Chapter 4 we show the resulting software after development and explain its interface and usage. Finally, we conclude our report in Chapter 5 by specifying the limitations of the current software as well as future goals and scope for improvement.

2. Requirements Specifications

The project has both functional and non-functional requirements for the system. As it is designed to serve the general populace for the specialized task of voting we had to keep usability in mind as well as security and performance when setting up the requirements. We also had to ensure a practical implementation of this project that is readily usable via the end users as easily as possible.

2.1 Functional Requirements

1. User Registration:

- Users must be able to securely register on the platform, providing identification and authentication details.

- The registered information must be vetted through an electoral body in charge.

2. Voter Verification:

- The system should verify the eligibility of voters to ensure they meet the criteria for voting.

- This verification must ensure votes are non-duplicate and cast by the intended voter.

- Registration process adheres to all established standards such as biometric verification.

3. Ballot Creation:

- Election authorities should have the capability to create and customize digital ballots for different elections.

- These ballots must have separate counts and end-user and voters must only be able to use them under specified constraints such as time and number of votes casted.

4. Voting:

- Registered voters must be able to securely cast their votes online, ensuring confidentiality and accuracy.

- Individual voters votes must also be confidential while the overall result must be trackable.

5. Vote Verification:

- The system should provide voters with the ability to verify that their vote has been recorded correctly and to their intended party.

6. Vote Counting:

   - Implement a transparent and tamper-resistant vote-counting mechanism.

7. Results Publication:

   - Display election results in real-time while maintaining the anonymity of voters.

2.2 Non-Functional Requirements:

1. Security Requirements:

   - The system must implement robust security measures to prevent unauthorized access, tampering, and hacking.

2. Scalability Requirements:

   - Ensure the system can handle a large number of users and concurrent transactions, especially during peak voting times.

3. Usability Requirements:

   - Design an intuitive and user-friendly interface accessible to a wide range of users, including those with limited technical proficiency.

4. Reliability Requirements:

   - The system must be available and operational during election periods without interruptions.

These non-functional requirements define the performance and quality characteristics of the Blockchain-Based Online Voting System, ensuring that it meets the highest standards of security, scalability, usability, and reliability.

3. Methodology

3.1 Software Model

We will adopt an iterative and incremental software development model for the BOVS project. This approach allows for flexibility in responding to changing requirements and continuous improvement of the system. The development process will consist of cycles or iterations, each of which includes planning, analysis, design, implementation, testing, and deployment phases. This iterative approach will enable us to refine the system incrementally while ensuring that it meets evolving user needs.

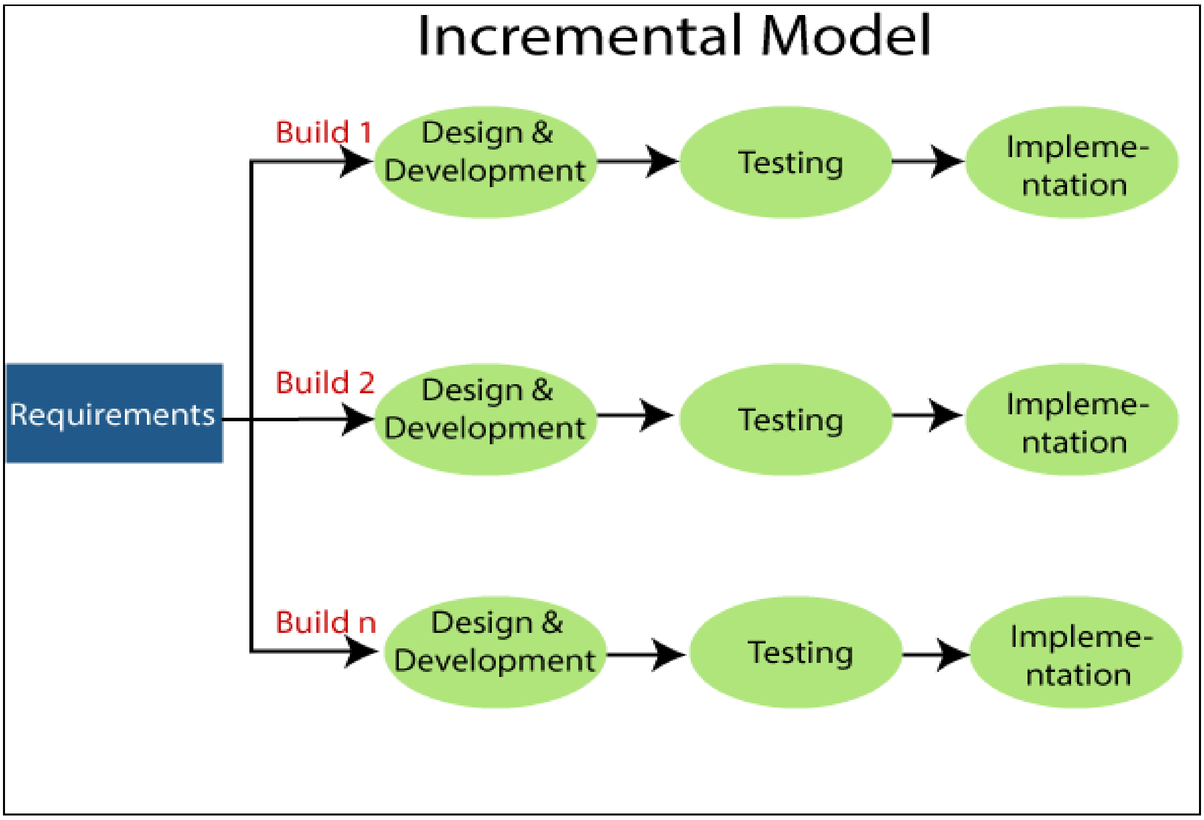

Figure 1: Incremental software development model

3.2 Software Development Methodology

We will employ the Agile methodology, specifically Scrum, to manage the development of the BOVS project. Scrum promotes collaboration, adaptability, and transparency, making it well-suited for complex software projects with changing requirements. The project will be divided into time-boxed iterations called "sprints," typically lasting two to four weeks. During each sprint, the development team will work on a set of prioritized features and functionalities, ensuring continuous integration and testing. Daily stand-up meetings will facilitate communication, and sprint reviews and retrospectives will provide opportunities for feedback and improvement.

3.3 Architectural Model

The architectural foundation of the Block-Chain based online voting system will be structured using a layered model, encompassing the following key components:

1. **User Interface (UI):** The user-centric web-based interface serves as the platform where voters engage in activities such as registration, casting votes, and result verification.

2. **Application Layer:** At the core of the system, this layer encapsulates essential functionalities, including user registration, voter verification, ballot creation, and the processing of votes. Ethereum smart contracts will play a pivotal role within this layer, ensuring the seamless execution of critical operations.

3. **Blockchain Network:** Serving as the foundational infrastructure, a private Ethereum blockchain network will uphold the ledger of votes and results. This layer is instrumental in ensuring the security and transparency integral to the system.

4. **Security Layer:** A robust security layer will be implemented, featuring encryption, authentication mechanisms such as facial, access control, and facial recognition-based authentication. These measures collectively safeguard user data and preserve the integrity of the entire system.

5. **Results Display:** The real-time results display module offers transparency to voters while meticulously preserving the anonymity of their individual choices. This component serves as a critical interface for users to access live updates on the unfolding electoral outcomes.

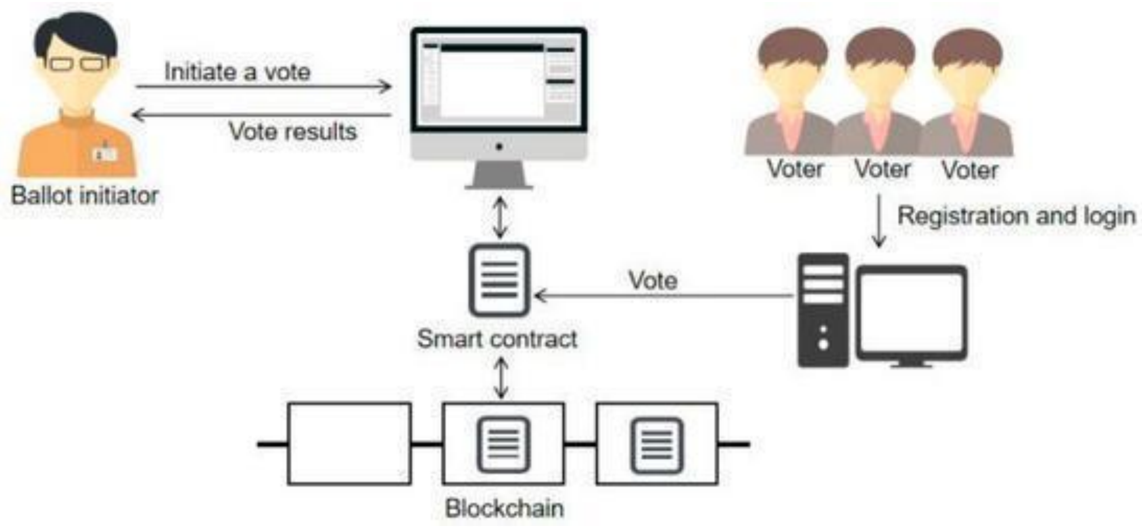

Figure 2: Block-chain-based voting system workflow.

3.4 Development Process

1. Requirements Gathering: A comprehensive collection of functional and non-functional requirements was undertaken through collaborative engagements with stakeholders.

2. System Design: The system architecture, user interfaces, and smart contracts were meticulously designed based on the elicited requirements.

3. Implementation: System components, encompassing smart contracts, user interface, and backend logic, were developed using the JavaScript programming language.

4. Testing: A rigorous testing regimen, comprising unit testing, integration testing, and user acceptance testing, was executed to identify and rectify any potential issues or vulnerabilities.

5. Deployment: The system was deployed on a secure server infrastructure with continuous monitoring for both performance and security considerations.

6. User Training: Comprehensive training sessions were conducted for election authorities, voters, and other stakeholders to ensure effective and secure utilization of the system.

7. Election Conduct: Real elections were conducted with the active participation of registered voters to assess the system's performance, security measures, and usability in a live environment.

The combination of the iterative software development model, layered architectural model, and a comprehensive development process will ensure the successful creation of the Blockchain-Based Online Voting System, meeting both functional and non-functional requirements while providing a secure, transparent, and accessible online voting platform.

3.5 Tools and Technologies Used

The development of the Blockchain-Based Online Voting System (BOVS) involved a carefully selected set of tools and technologies to ensure the successful creation of a secure and efficient online voting platform:

1. Solidity:

- Description: Solidity is a high-level, statically typed programming language designed for writing smart contracts on the Ethereum blockchain. It was used extensively to develop the core logic of BOVS, including user registration, vote verification, and ballot creation.

- Purpose: Solidity allowed us to create self-executing smart contracts that automatically enforce the rules and conditions of the voting system, ensuring the integrity of the process.

2. React.js:

- Description: React.js is a popular JavaScript library for building user interfaces. In BOVS, React.js was employed to develop the web-based user interface, providing an intuitive and responsive platform for voter registration, casting votes, and result verification.

- Purpose: React.js enabled us to create a user-friendly front-end that enhances accessibility and user engagement.

3. JavaScript:

- Description: JavaScript, alongside React.js, played a pivotal role in developing the BOVS user interface and managing client-side functionality.

- Purpose: JavaScript was used for client-side scripting, ensuring the dynamic behavior of the user interface and enhancing the interactivity of the voting platform.

4. GitHub:

- Description: GitHub is a web-based platform for version control and collaborative software development. It was used as the primary version control system for the BOVS project, allowing multiple developers to work on the codebase simultaneously, track changes, and collaborate effectively.

- Purpose: GitHub facilitated efficient code management, collaboration, and the ability to track and address issues and enhancements throughout the development lifecycle.

5. Ethereum Blockchain:

- Description: The Ethereum blockchain is a decentralized, public blockchain network that serves as the foundation of BOVS. It was used to create a private blockchain network specifically for the voting system.

- Purpose: The Ethereum blockchain ensured the security, transparency, and immutability of voting data, providing a tamper-resistant ledger for recording votes and election results.

6. Metamask:

- Description: Metamask is a browser extension and mobile app that allows users to manage Ethereum accounts and interact with Ethereum-based decentralized applications (DApps). It was integrated into BOVS to provide users with a secure and user-friendly way to interact with the blockchain.

- Purpose: Metamask facilitated secure user authentication and seamless interaction with Ethereum smart contracts, ensuring the privacy and security of voter actions.

The combination of these tools and technologies, with JavaScript as the main programming language, played a pivotal role in the successful development of the Blockchain-Based Online Voting System, providing a robust, secure, and user-friendly online voting platform that leverages the Ethereum blockchain's capabilities.

7. Express.js:

- Description: Express.js, also known as just Express, is a quick, understated, and adaptable Node.js online application framework. It is intended to make the process of creating scalable and reliable web apps and APIs easier.

- Purpose: Adding routing, middleware, HTTP utility methods, static file serving, etc. is the main goal.

8. MongoDB:

- Description: MongoDB is a popular NoSQL database management system that uses BSON (Binary JSON), a versatile format similar to JSON, to store data. It belongs to the class of document-oriented databases.

- Purpose: Offering modern applications a NoSQL database solution that is adaptable, scalable, and high-performing is the main goal of MongoDB.

4. Implementation, Result & Discussion

The project has been implemented as a website built using REACT.js Framework for the front end and Solidity.js on the back end to implement our project. We used MetaMask as our crypto wallet for connecting digital contracts to the blockchain streamlining the generation of voter and organizer accounts and tracking the blockchain for changes. We also used hugging faces API based facial recognition system for implementation of an accurate model for a secure biometric based authentication.

4.1 Facial Recognition System Implementation

We used the openCV package for implementing the entire facial recognition system. It uses the device camera for reading the voters image at runtime and checks its similarity from the database. The underlying model tracks the common facial features one by one between multiple images to accurately detect faces. The model being used is a cascade classifier built into openCV with an underlying algorithm of an haar cascade classifier.

Once a face has been identified the image of the face is run against the database to find images of the voter and run all images through a LBPH algorithm (Local Binary Patterns Histograms) which is an easier implementation of existing face recognition algorithms. It works by taking each pixel and thresholding each pixels values similarity to nearby pixels of each image . Depending on the level of similarity it gives an approximate estimation of how similar two images are . We considered a similarity of 75% to be enough to consider two images to be of the same person after much experimentation.

4.2 UI Implementation

Figure 3: Homepage

The above image shows the homepage of the voting system. It has buttons to log in as an organizer on the left and a voter on the right.

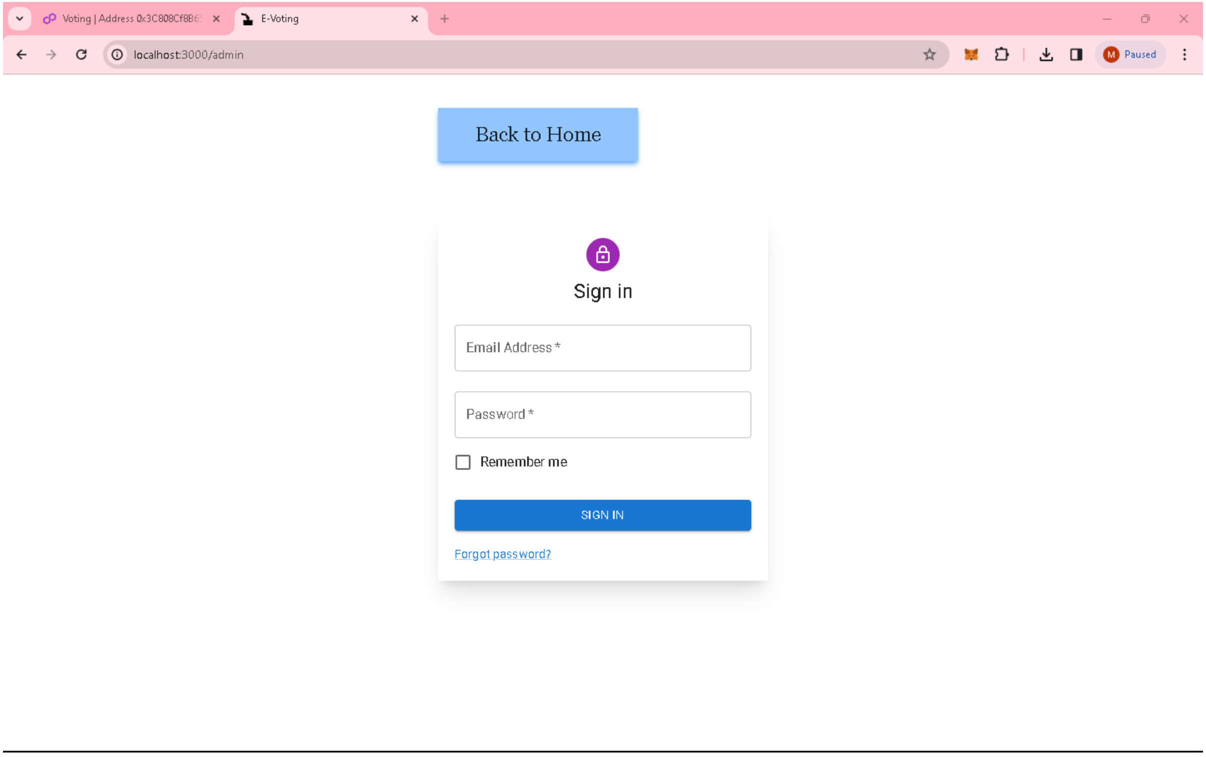

Figure 4: Admin Login Page

Here, admin can login by using the valid email and password. In this section, admin can add organizer.

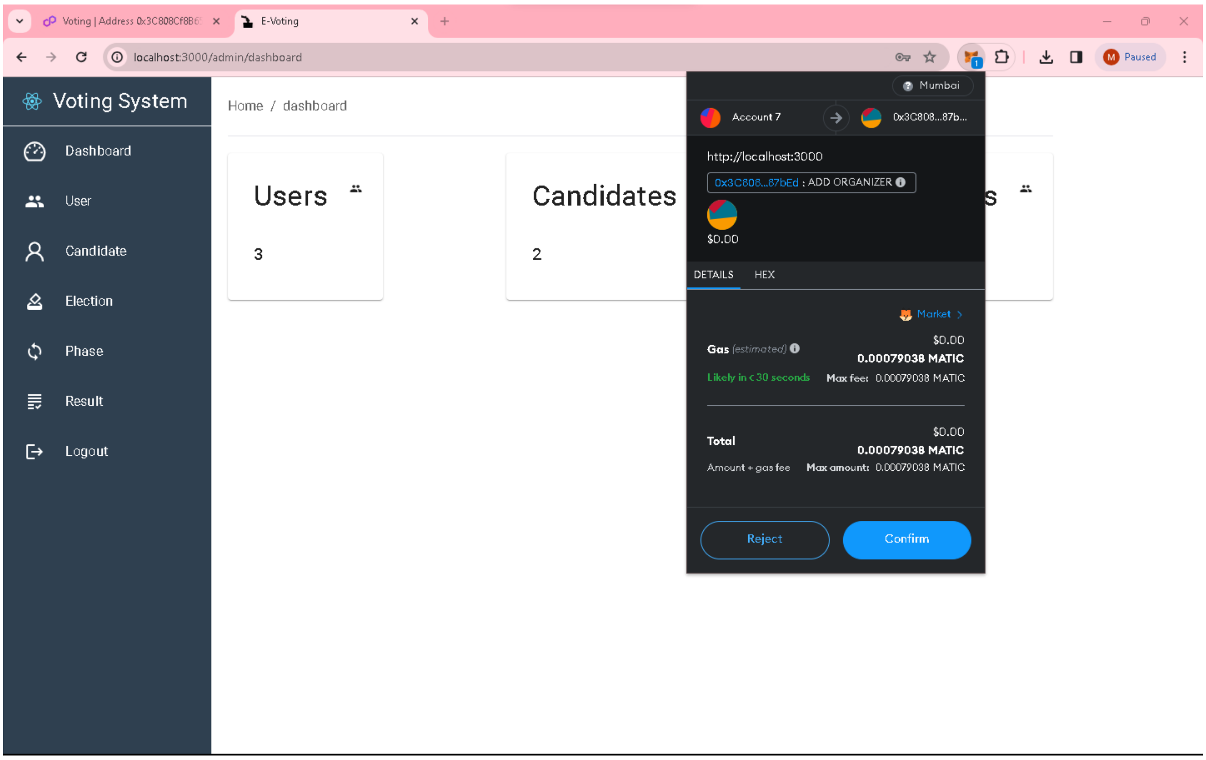

Figure 5: Organizer Selection by Admin using his/her Metamask account

The above image represents the "Add Organizer" section by providing his/her Metamask account address. The organizer can be able to add voter, candidate as well as election.

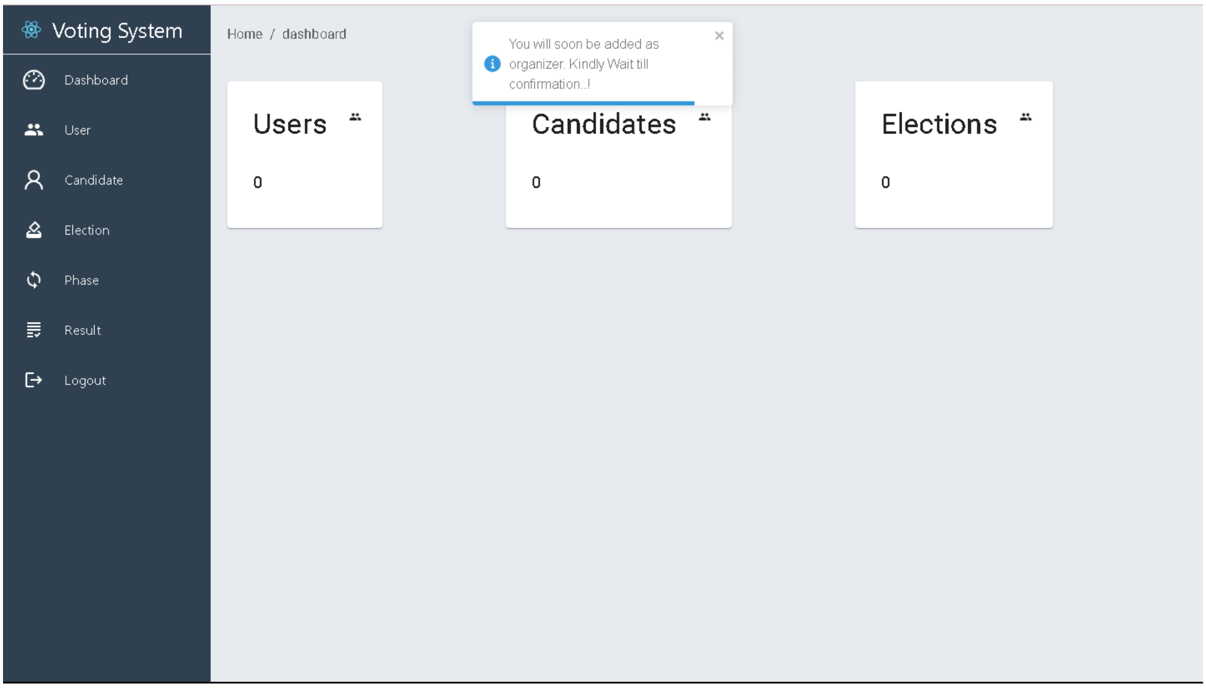

Figure 6: Organizer Added successfully to the blockchain

The above image represents a confirmation message by adding the organizer to the blockchain.

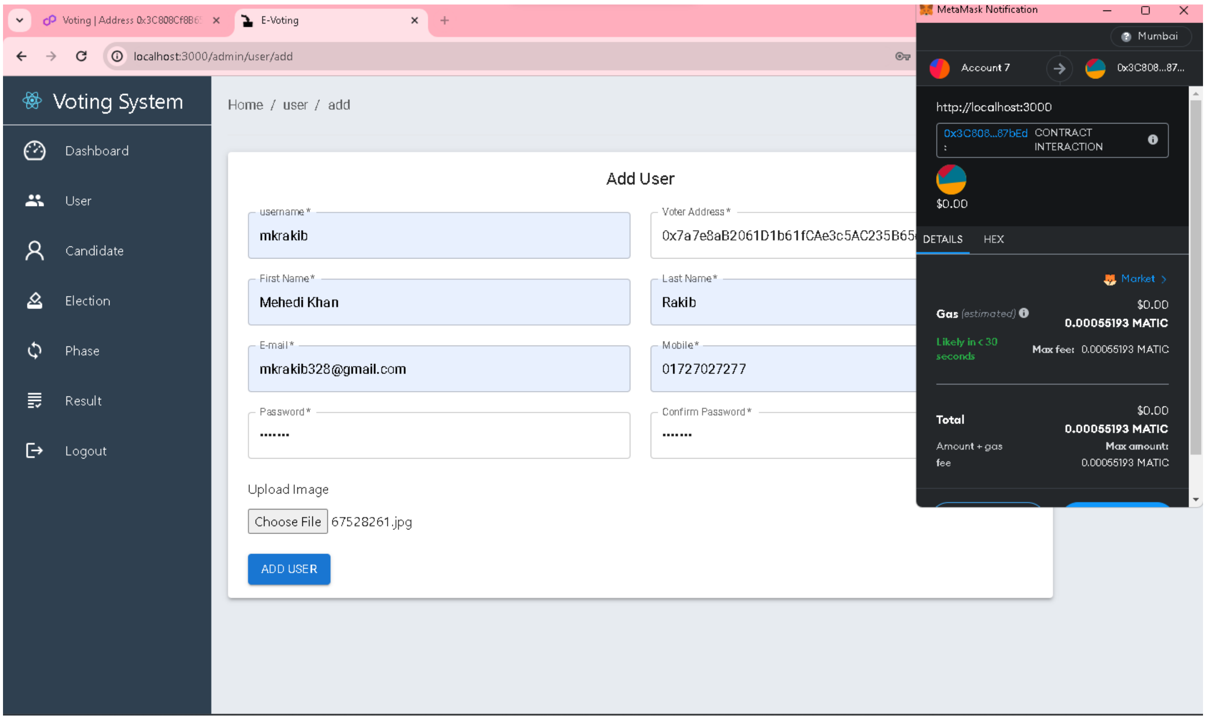

Figure 7: Add a Voter to the blockchain

By clicking on the “User” button from the left sidebar, the organizer can add voters to the blockchain.

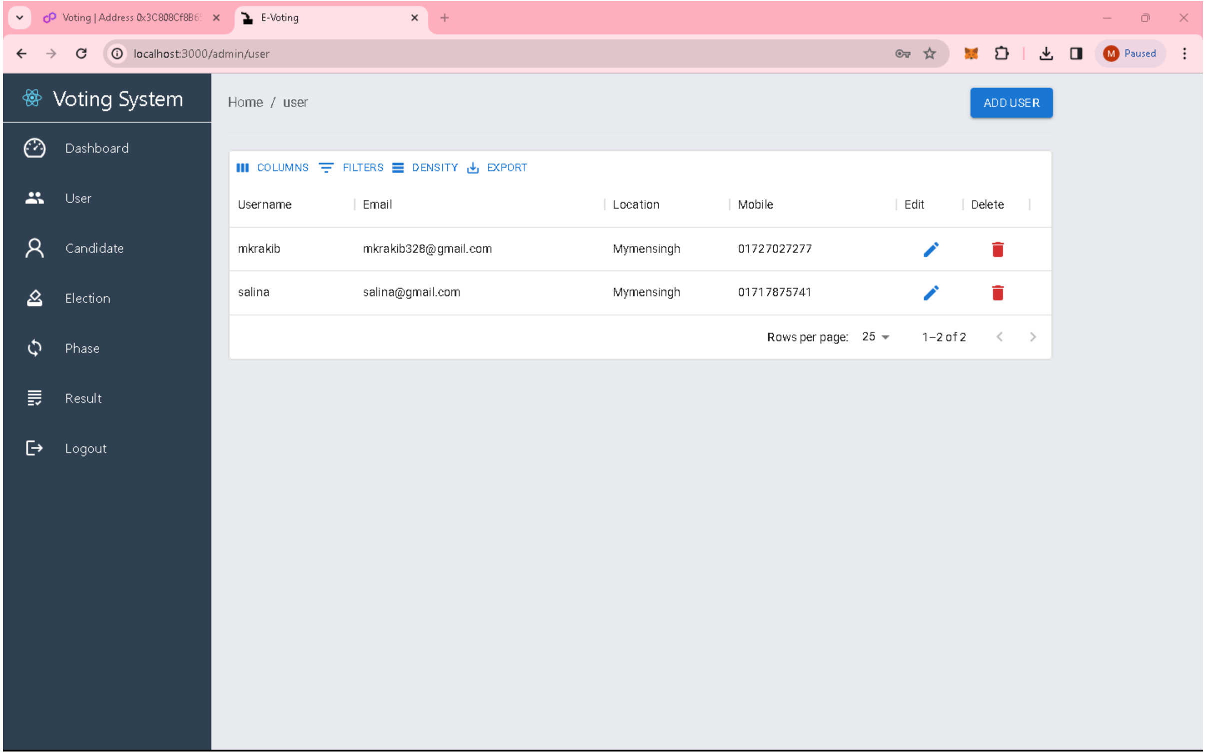

Figure 8: View all Voters

The above image shows a page where all the voters can be shown. Here, some necessary information is provided.

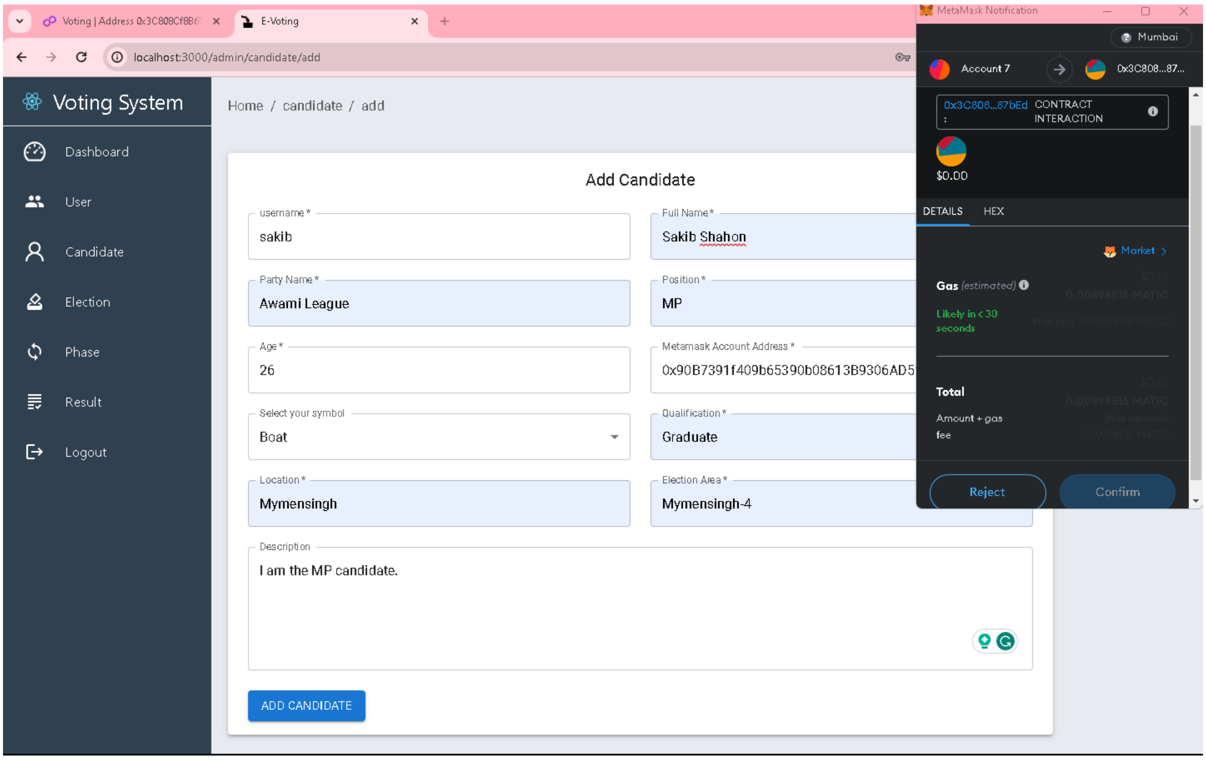

Figure 9: Add a Candidate to the blockchain

Here, the organizer can add candidates by clicking on the "Candidate" button, located at the left sidebar. The added candidates will go to the blockchain.

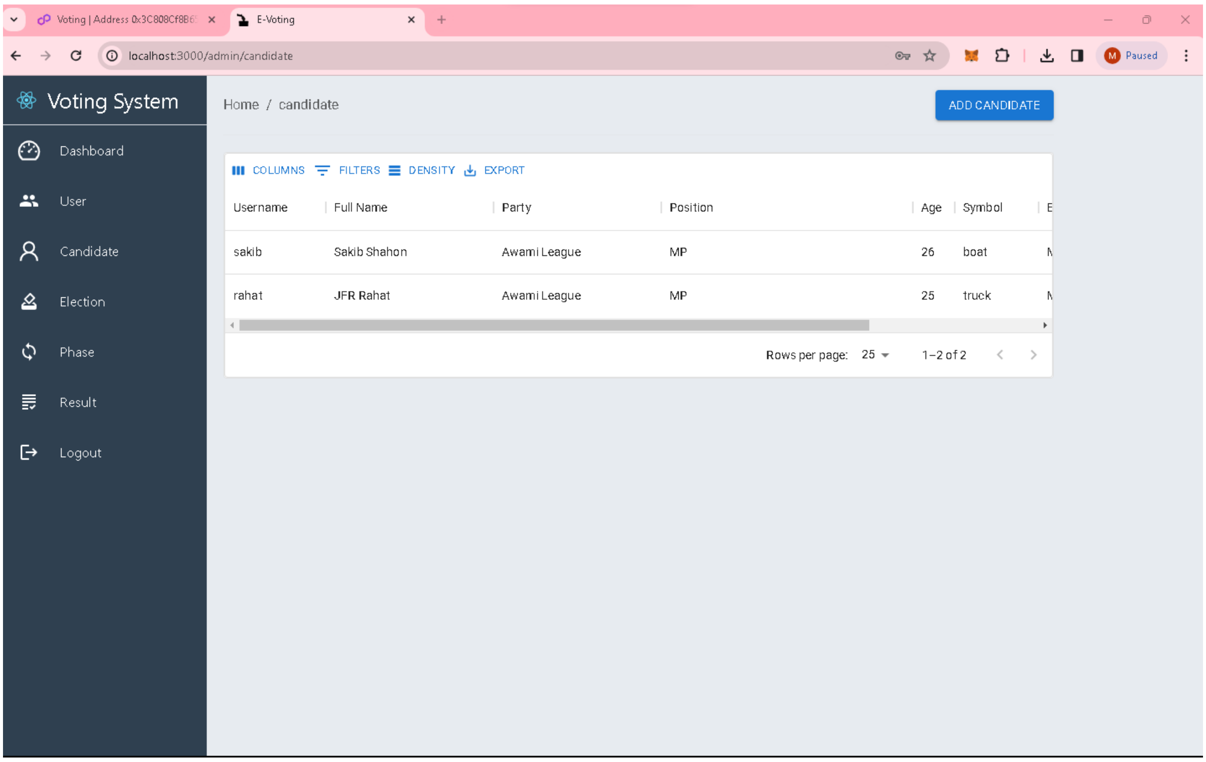

Figure 10: View all candidates

The above image shows a page where all the candidates can be shown. Here, some necessary information is provided.

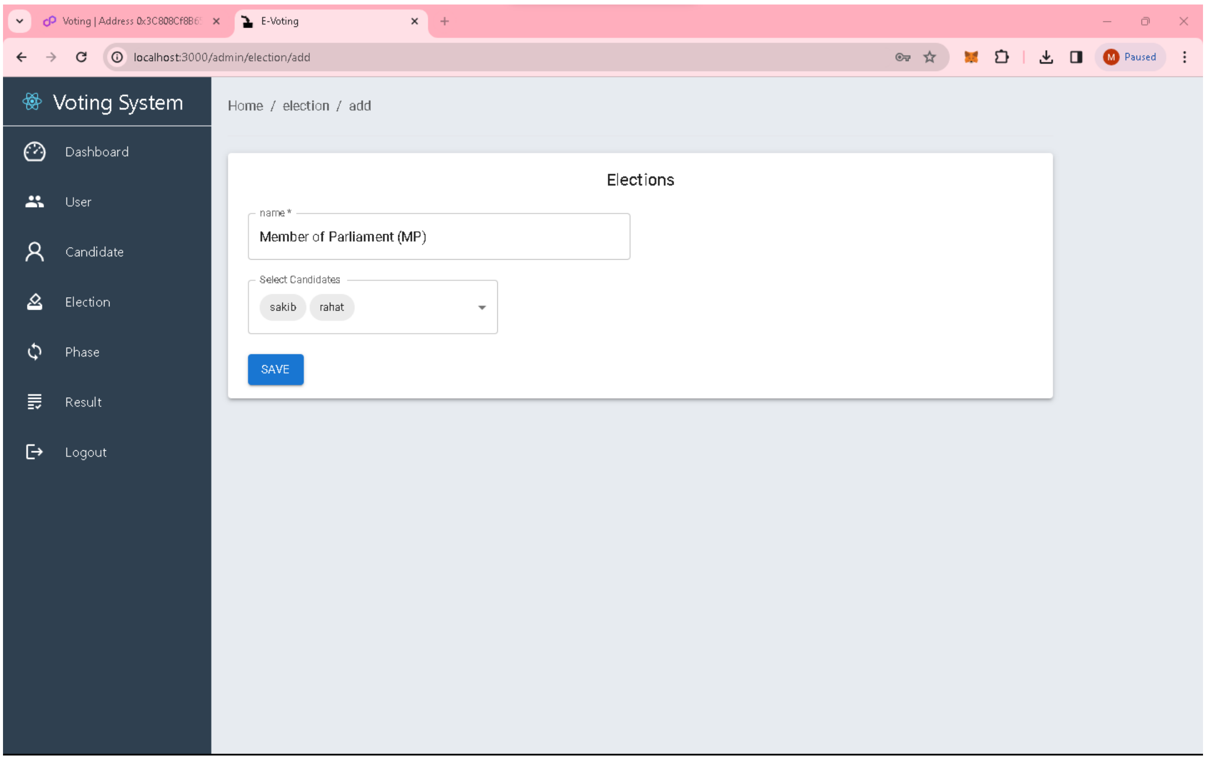

Figure 11: Add candidates to the upcoming election

Here, the organizer can add candidates to the upcoming election by clicking on the “Save” button.

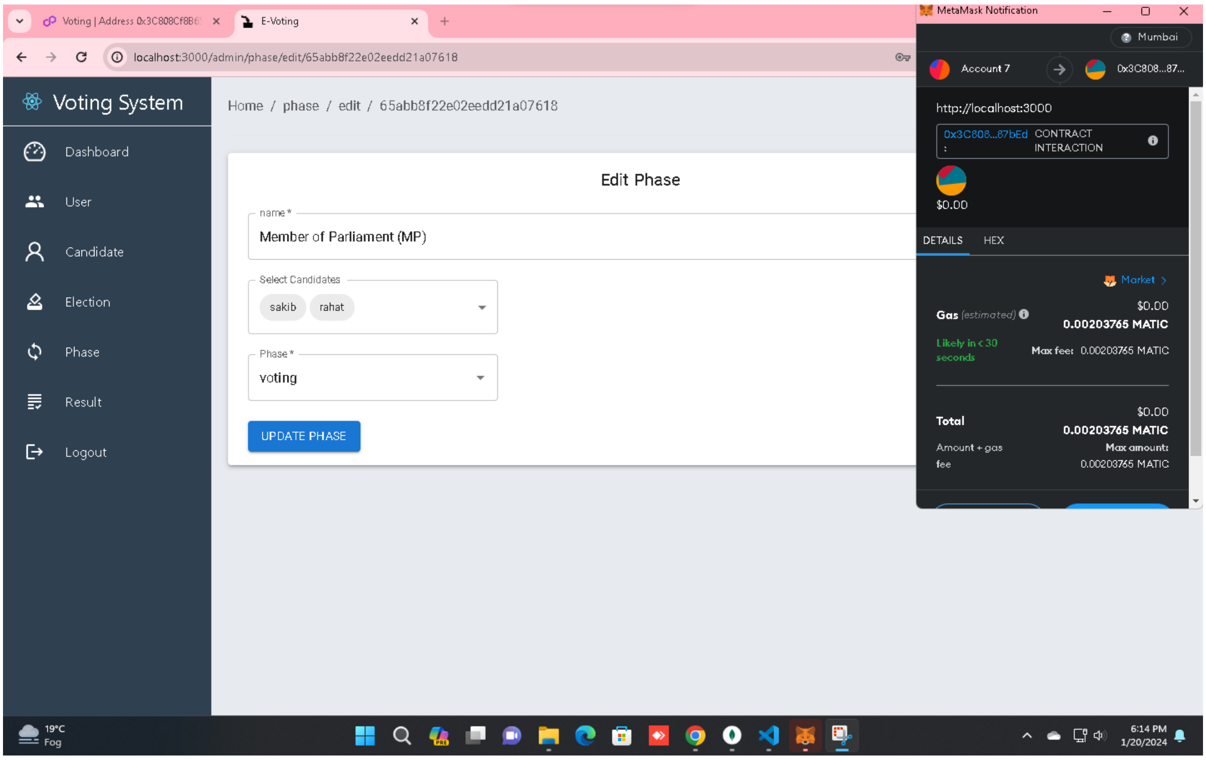

Figure 12: Edit election phase as voting

After starting the election, the organizer can edit election phase as voting. Voters can now be able to cast their votes individually.

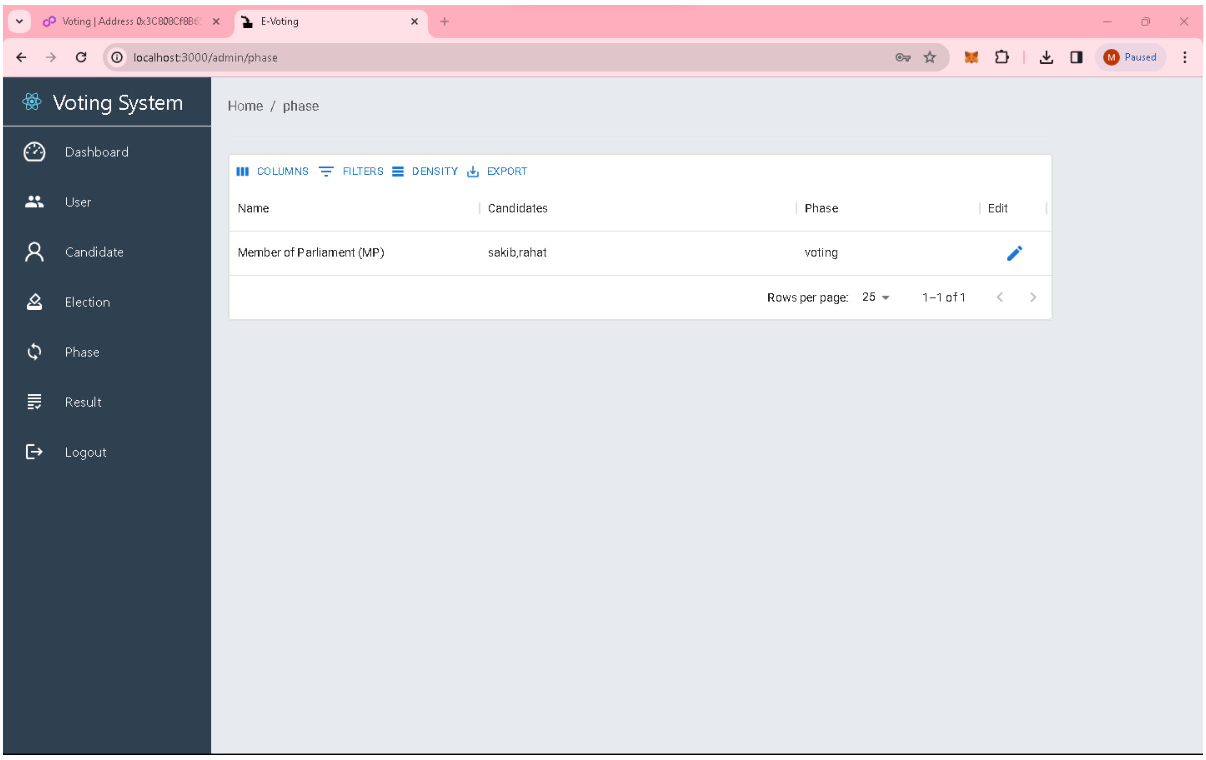

Figure 13: View election current state

The above image shows the current situation of the election right now. The election phase changes from "init" to “voting” state.

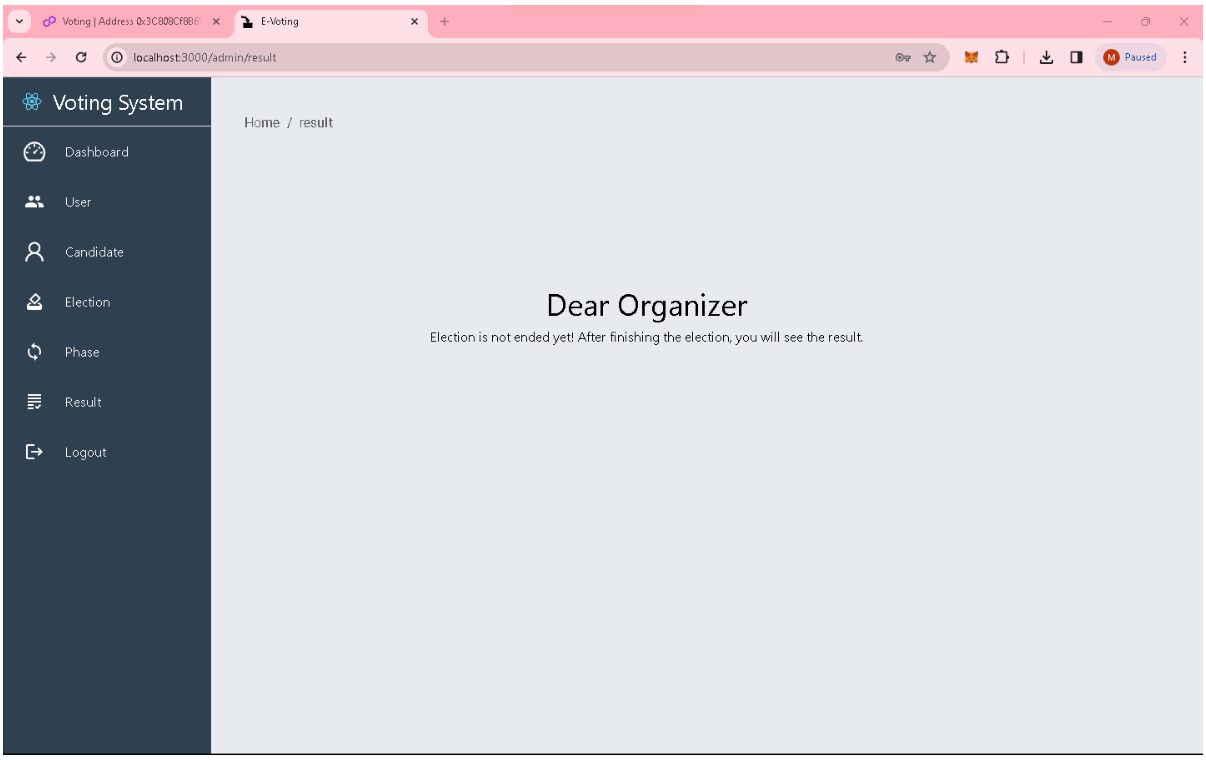

Figure 14: View result page

After starting the voting procedure of the election, the confirmation message will be displayed as the election is not ended yet.

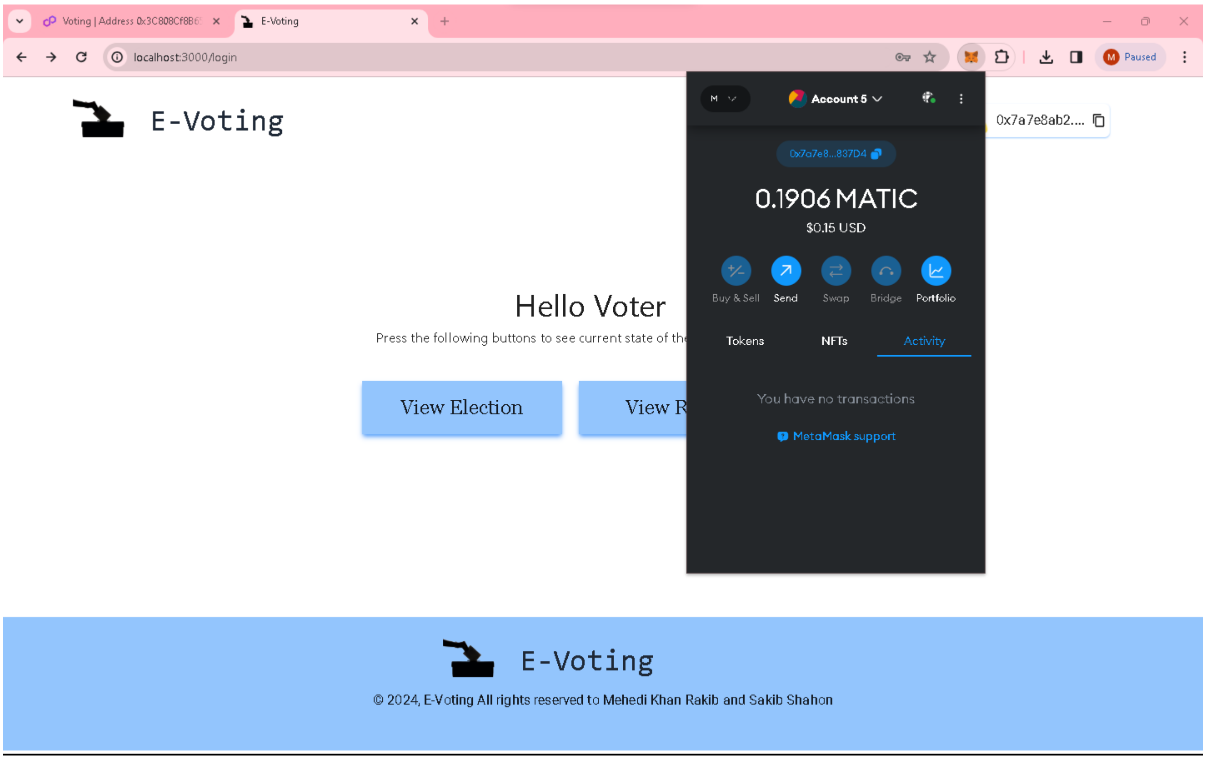

Figure 15: Voter initial login by his/her Metamask account

After starting the election by the organizer, the voter has to initially login by his/her Metamask account address. Then, he/she can notice either the election or result according to the current state of the election.

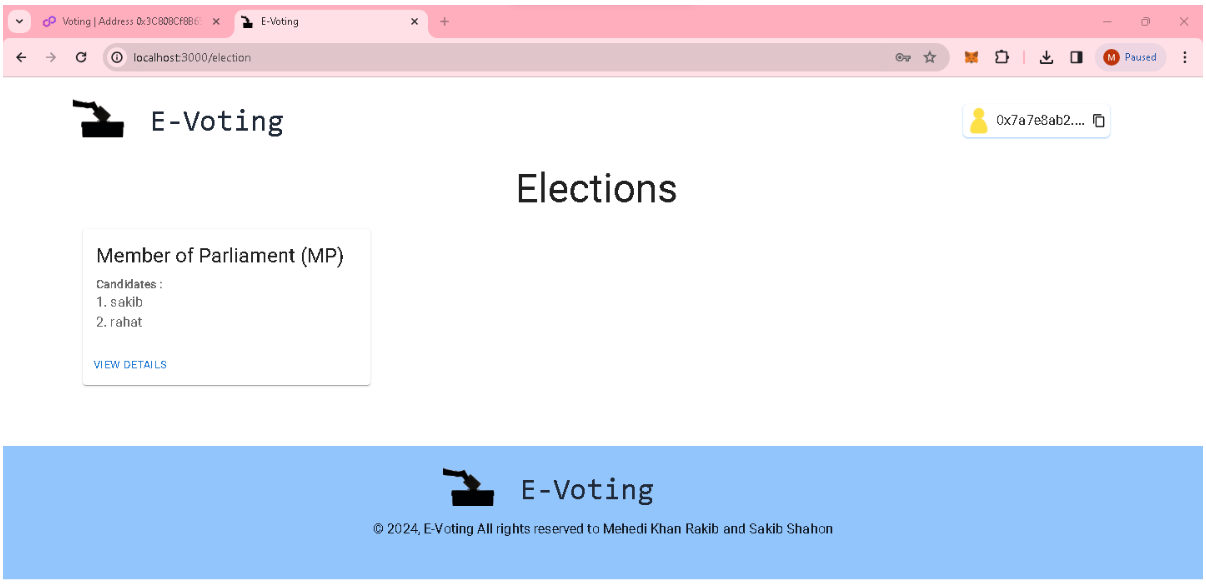

Figure 16: View election by voter

In this section, the voter can view the election. Here, he will observe the name of the election and the list of the candidates.

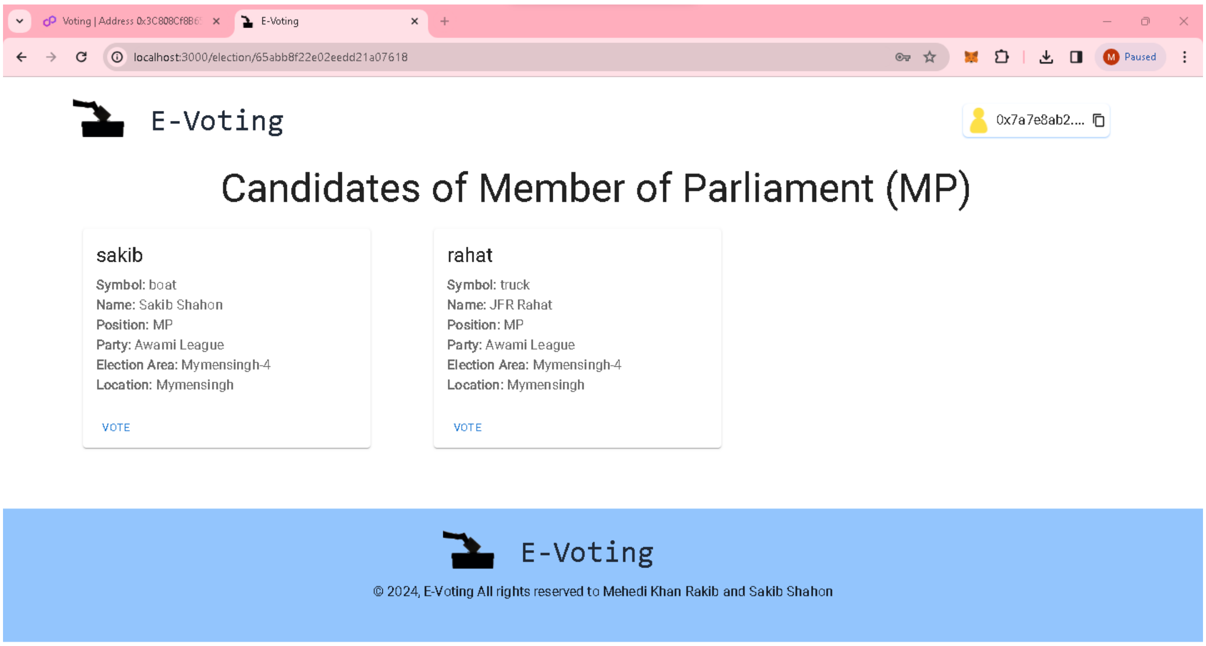

Figure 17: View candidate details by voter

After clicking on the "View Details" button, the voters will see the details of the participated candidates of the election.

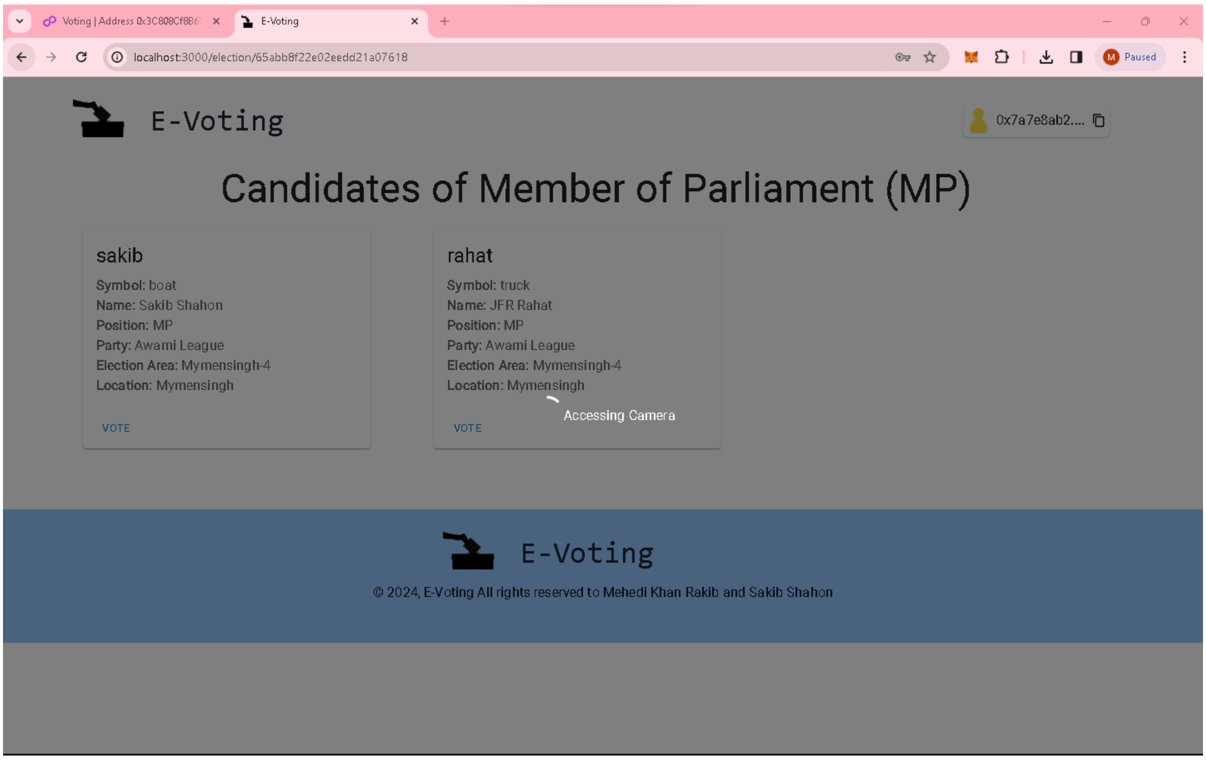

Figure 18: Camera is opening by clicking on vote button to recognize the voter using facial recognition

After clicking on the "Vote" button, voter's camera will be opened to recognize him/her for giving their vote as the authenticated voter.

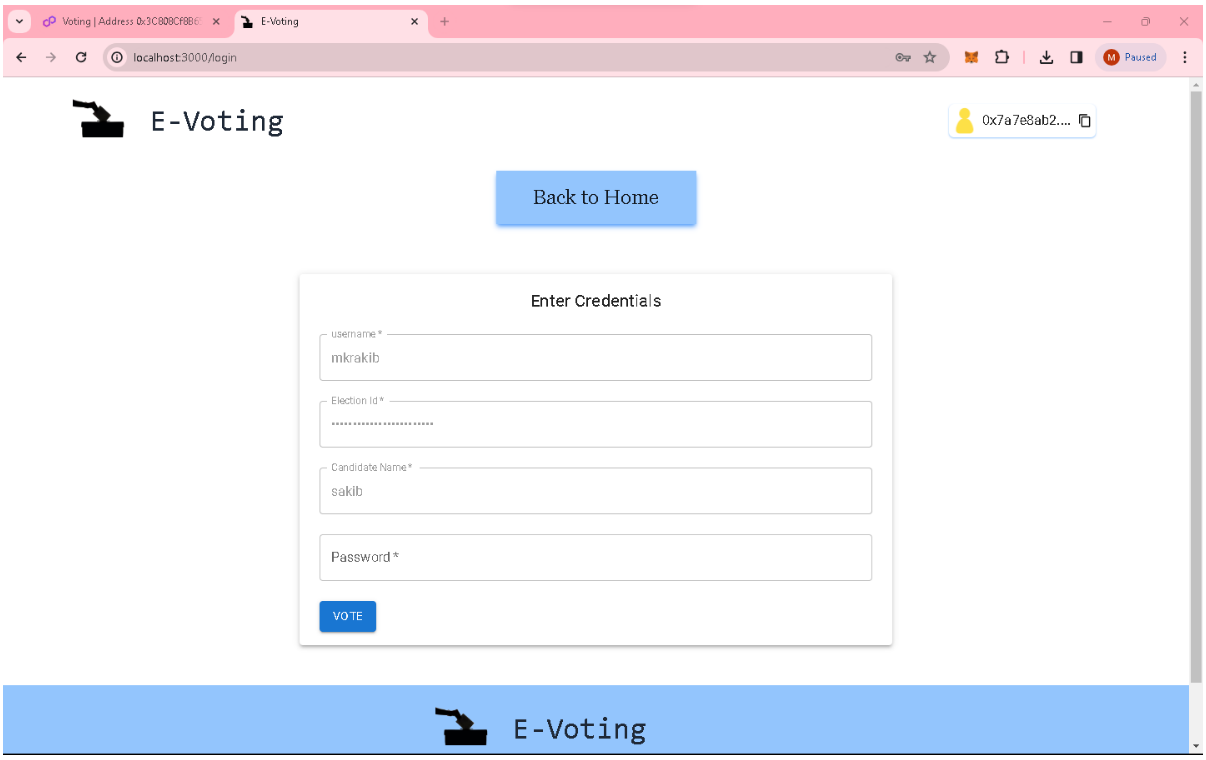

Figure 19: Voter is successfully recognized as the authenticated voter

After successfully completing the authentication of the voter, the voter can vote to the favorite candidate once.

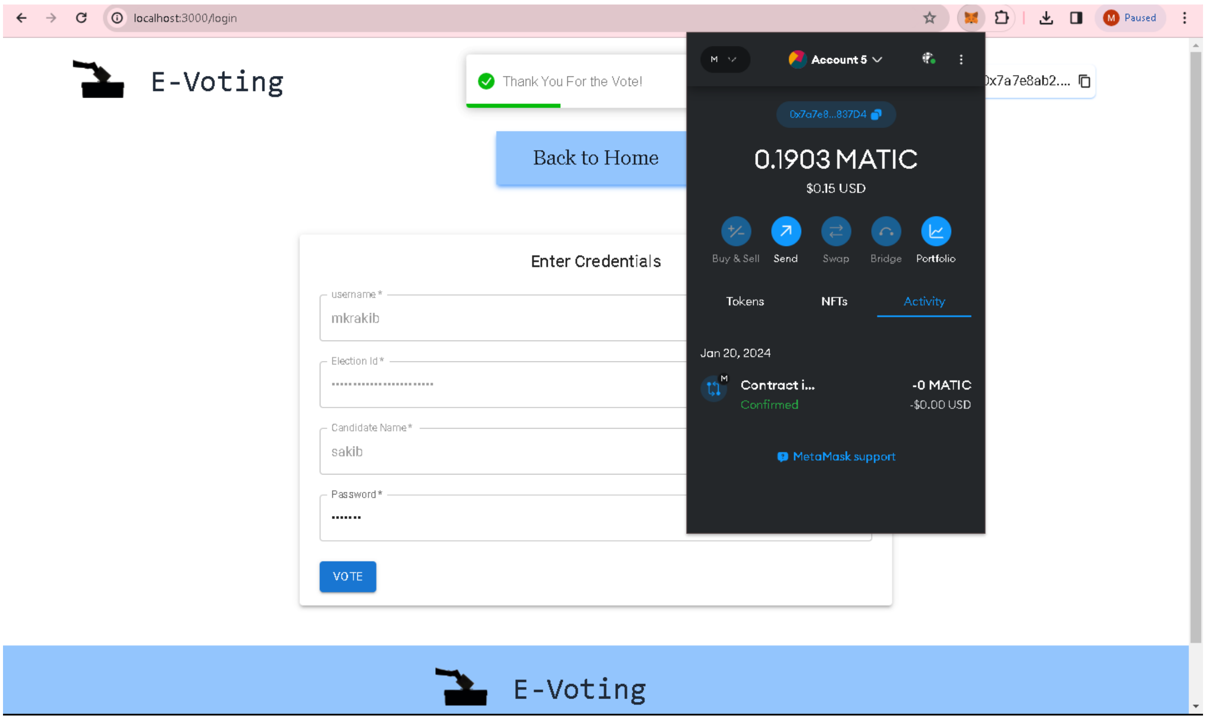

Figure 20: The Voter votes successfully and gets mail to his email address

After casting the vote, the voter can be notified by a confirmation message. He/she will also get a mail to their email address.

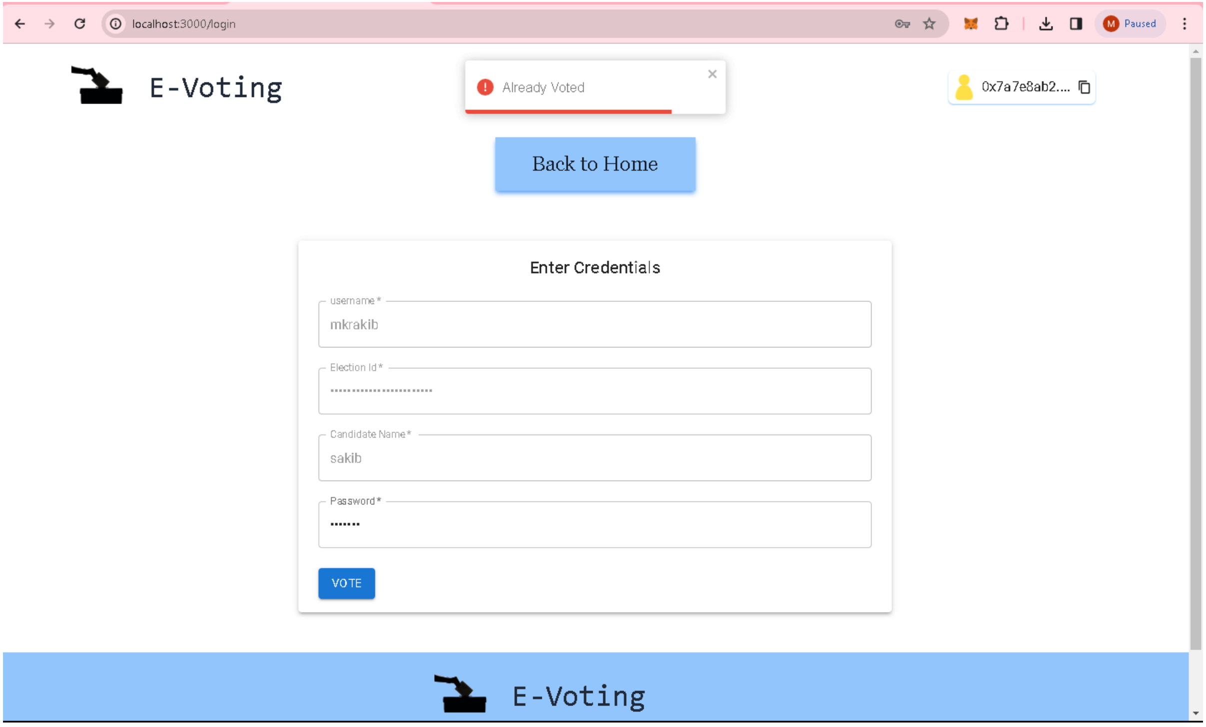

Figure 21: The Voter tries to vote twice but he/she fails

After casting the vote once, the voter tries to cast his/her vote twice. So, blockchain responds to a negative feedback by providing a confirmation message to the voter.

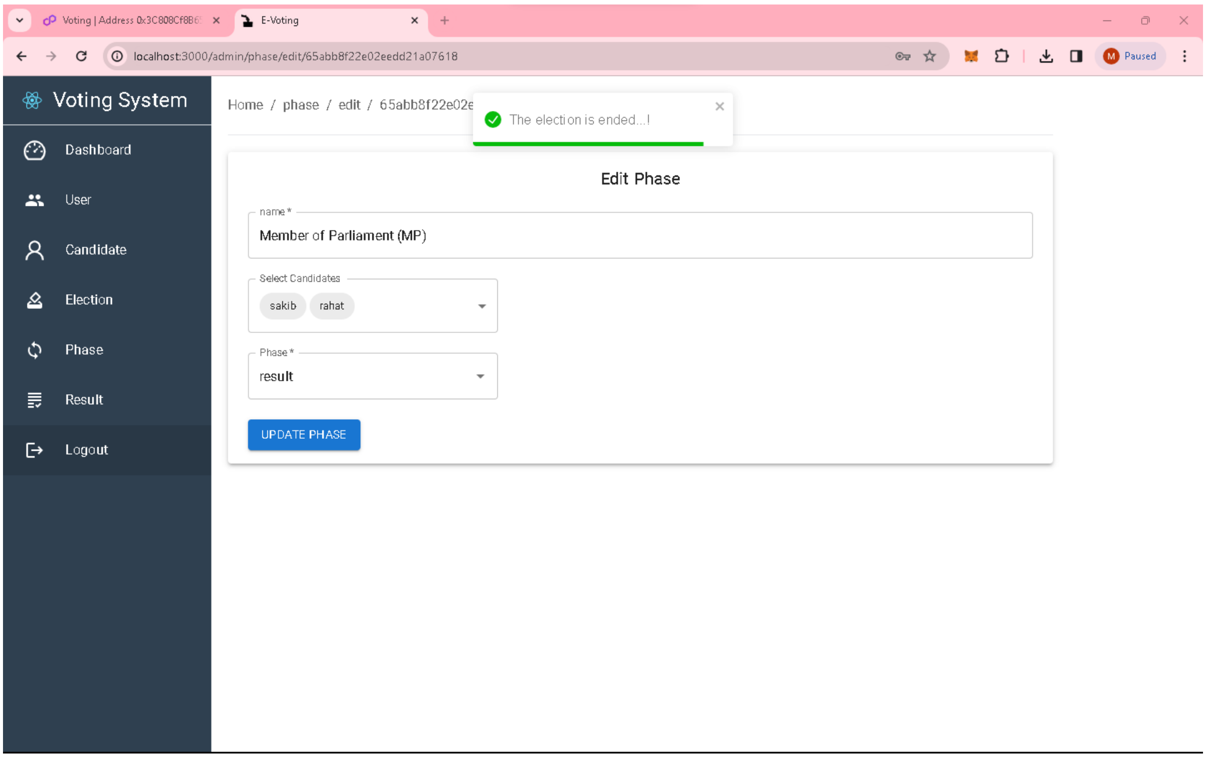

Figure 22: Edit phase by selecting result option by organizer

After completing the voting process, the organizer can now be able to end the election to the blockchain and publish the result to both the organizer himself/herself and the voters.

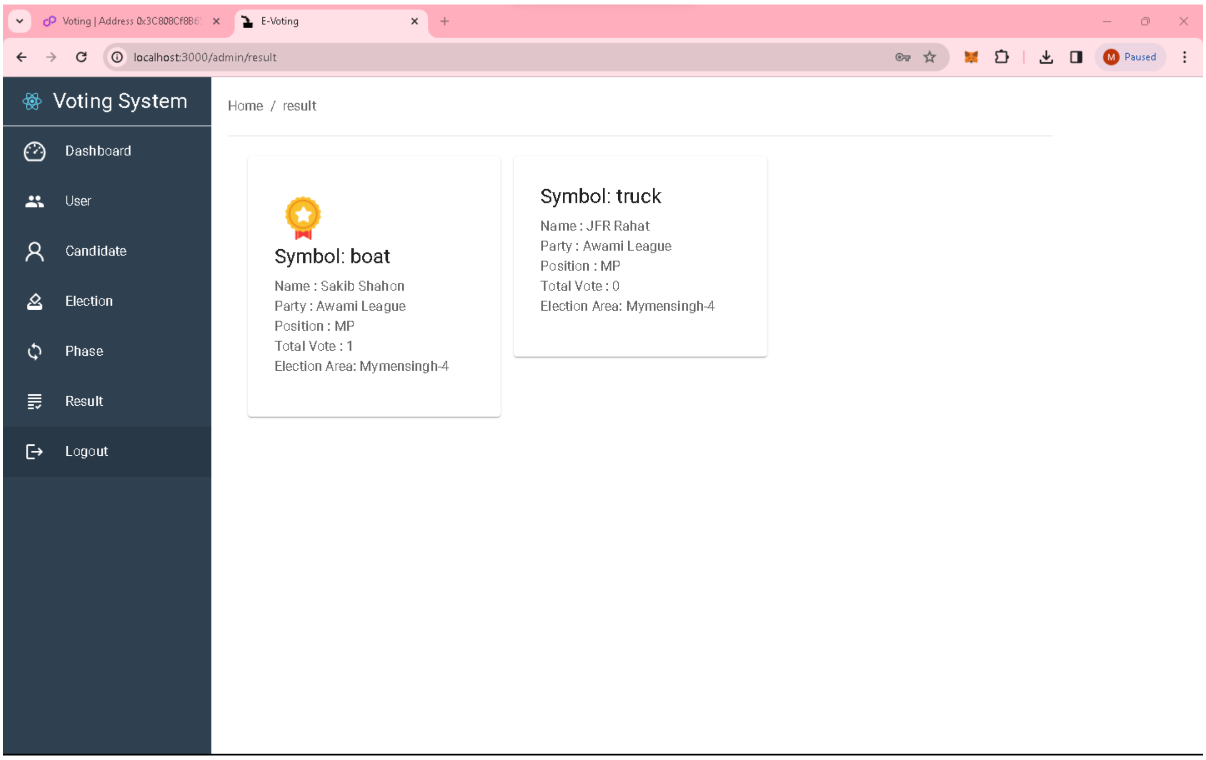

Figure 23: View the published result by organizer

By clicking on the "Result" button from the left sidebar, the organizer can view the result of the candidates. He/she will notice both the casting vote and the winner from the election.

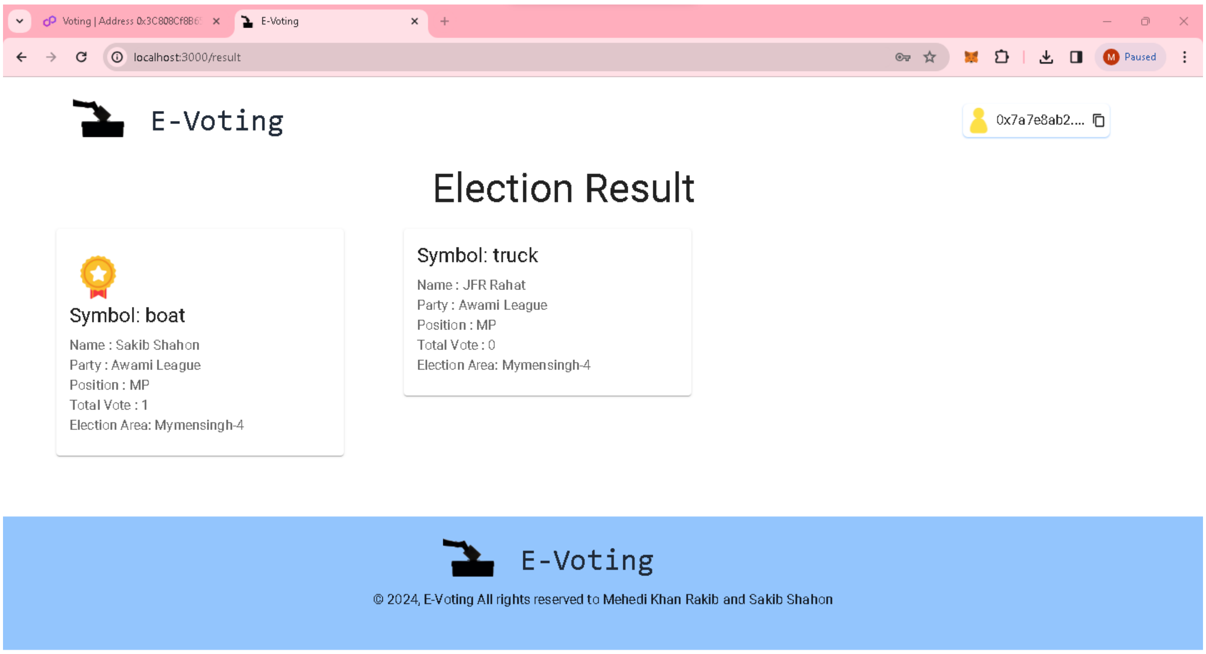

Figure 24: View the winner by the voter

By clicking on the "Result" button from the voter panel, the voter can view the result of the candidates. He/she will notice both the casting vote and the winner from the election.

5. Conclusion

In conclusion, the Blockchain-Based Online Voting System (BOVS) stands as a pivotal stride in the modernization and fortification of the electoral process. Throughout this project, our focus has been on mitigating the critical challenges inherent in traditional voting systems, addressing issues ranging from security concerns to transparency and accessibility limitations. Our journey has epitomized a commitment to exploration and innovation, guided by the overarching objective of revitalizing the democratic process.

The impetus behind the development of a Blockchain-Based Online Voting System emanates from the imperative need to restore and bolster public trust in elections. In an era marked by rapid technological advancements, it is our responsibility to harness the potential of blockchain technology, thereby creating a secure, transparent, and tamper-resistant platform for online voting.

Comprising a diverse array of functionalities, our system encompasses the entire voting process, from user registration and voter verification to ballot creation, voting, vote verification, and result publication. Every facet has been meticulously designed to ensure that the Blockchain-Based Online Voting System not only prioritizes user-friendliness but also remains impervious to security breaches and tampering, thereby upholding the integrity of the democratic process.

5.1 Limitation

Throughout the project, we have adhered to an iterative and incremental development model. Although we have incorporated the most necessary features over multiple iterations our project still faces some limitations for the current version. Including:

1. The modifiability of the structure of individual elections.
2. The dependency on third-party software for managing the accounts and transactions via the smart contract of our system.
3. Lack of methods that ensure that the digital account holder is the true voter, and the voting account isn't handed over to someone else (such as the account holder giving away his account password or voting link)

We are currently working on these pressing issues and have architected multiple possible solutions that couldn't be integrated into the current version due to the necessary time and the complexity of the solutions. Some of the constraints are due to the technology itself (Khan et al., 2020).

5.2 Future Scope

Although our system is at a usable and overall developed state, many aspects of it can still be enhanced to better serve its purpose and improve performance. The main scope for improvement in the future includes:

1. Remove all third-party dependencies for managing smart contracts and monitoring for an in built solution.
2. Make the facial recognition authentication procedure more full proof by implementing real time instruction execution based operations.

The BOVS project is a testament to our commitment to fostering democracy, enhancing voter participation, and strengthening the foundations of democratic governance.

In conclusion, BOVS is more than just a technological achievement; it is a beacon of hope for a more secure, transparent, and accessible electoral process. Through its continued development and implementation, we aspire to contribute to the evolution of democracy, ensuring that the voice of every citizen is heard and valued.

Reference

Hjalmarsson, F. P., Hreioarsson, G. K., Hamdaqa, M., & Hjalmtysson, G. (2018). Blockchain-Based E-Voting System. *IEEE 11th International Conference on Cloud Computing (CLOUD)*. https://doi.org/10.1109/cloud.2018.00151

Khan, K. M., Arshad, J., & Khan, M. M. (2020). Investigating performance constraints for blockchain based secure e-voting system. *Future Generation Computer Systems*, *105*, 13–26. https://doi.org/10.1016/j.future.2019.11.005

Solidity — Solidity 0.8.21 documentation. (n.d.). https://docs.soliditylang.org/en/v0.8.21/

React – A JavaScript library for building user interfaces. (n.d.). React. https://legacy.reactjs.org/

MetaMask developer documentation. (n.d.). https://docs.metamask.io/