Digipolls - A Blockchain-based E-voting System

Submitted In Partial Fulfillment of Requirements

For the Degree Of

Third Year

Computer Engineering

By

Omkar Boralkar

16010121027

Aditya Patil

16010121028

Nupur Chaudhari

16010121033

Vignesh Iyer

16010121066

Guide

Prof. Swapnil Pawar

Somaiya Vidyavihar University

Vidyavihar, Mumbai - 400 077

Academic Year 2023-24

Certificate

This is to certify that the TY Mini Project report entitled **Digipolls - A Blockchain-based E-voting System** submitted by Omkar Boralkar (16010121027), Aditya Patil (16010121028), Nupur Chaudhari (16010121033) and Vignesh Iyer (16010121066) at the end of semester VI of TY B. Tech is a bona fide record for partial fulfillment of requirements for the degree in **Computer Engineering** of Somaiya Vidyavihar University

Guide	Examiner

Date:

Place: Mumbai-77

Abstract

In an era where technological advancements permeate every aspect of society, the traditional methods of voting are ripe for innovation. Enter Digipolls, a groundbreaking project aimed at revolutionizing the voting experience through the integration of blockchain technology. Digipolls seeks to address the inherent challenges of conventional voting systems by leveraging the security, transparency, and decentralization capabilities of blockchain.

At the core of Digipolls is the utilization of blockchain-based smart contracts to facilitate voting procedures. These smart contracts serve as immutable records of votes cast by users, stored securely on the Ethereum blockchain. Through this decentralized ledger, Digipolls eliminates the risk of tampering or manipulation, instilling trust and confidence in the electoral process.

Moreover, Digipolls eliminates the need for intermediaries, streamlining the voting process and reducing the potential for human error or bias. By providing a direct channel for voters to cast their ballots, Digipolls enhances the efficiency and integrity of elections, fostering a more inclusive and democratic society.

In addition to its technological innovations, Digipolls prioritizes user experience, offering an intuitive and user-friendly interface for voters to participate in elections. Through seamless integration with web and mobile platforms, Digipolls ensures accessibility for users of all backgrounds, empowering individuals to exercise their democratic rights with ease.

Looking ahead, Digipolls envisions a future of fairer, more transparent, and accessible elections. Ongoing efforts focus on further enhancing security measures, scalability, and regulatory compliance to meet the evolving needs of society. With Digipolls, the future of voting is digital, decentralized, and democratic.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter presents an overview of the project, outlining its background and motivation to create a blockchain-based electronic voting system. It defines the problem statement, emphasizing the importance of protecting the anonymity of voters and ensuring that the cast votes are not tampered with or destroyed. The scope of the project is delineated, along with the objectives that the electronic voting system is expected to achieve. Additionally, it details the hardware and software requirements required for development of the project.

1.1 Background and motivation

Voting whether conducted through the traditional ballot or via electronic means forms the basis on which democracy depends. With the rise in technological impact on the youth of the country and the various anomalies faced by the current electoral process, using technology to modify the existing process is necessity of the hour. However, for any new technique to take the place of current voting system, the said system needs to satisfy certain minimum criteria. Electronic Voting has taken a central place in research with the intention of minimizing the cost associated in setting up the voting process, while ensuring the electoral integrity is maintained by fulfilling privacy, security and compliance requirements.

The current method, whether electronic or not has proved to be unsatisfactory with respect to transparency. It can be very difficult for the voters to be assured that the vote he/she has casted during the election reflects in the election result. Electronic voting using Direct Recording Electronic do not generate receipt on successful casting of votes. No record of election except vote count is made public by the government, which means that the voters are not assured of any external interference in case of government conducting the process of vote recounting. Replacing the traditional method with electronic method using Blockchain technique has the ability to prevent potential frauds that may take place during election.

Blockchain technology is a distributed network of interconnected nodes. A copy of distributed ledger is assigned to each node, each of which contains a complete history of all the transactions that have been processed by the network. Each transaction processed generated a hash. The hash created depends not only on the current transaction but also on the hash of the previous transaction. Thus, any small change on the data will impact the hash of the transaction. If a transaction is approved by a majority of nodes it is written to the block. This allows the users to remain autonomous while using the system. A basic analysis of Blockchain suggests that it provides the potential of making the voting process more secure and reliable.

The main motivation behind this project is the fact that will of people forms the basis of democracy. However, it is of utmost importance to protect the anonymity of voters and allow complete privacy to cast their votes. The current methodology may sometimes fail to protect the fundamental right of privacy of the voters. The master key to build an electronic voting system is to find out a secure underlying platform which provides the required features that overcomes the drawbacks of the current system.

1.2 Problem Statement

Design, implement and test a blockchain-based e-voting solution, utilizing the Ethereum platform and Solidity language, to ensure security, transparency, and the privacy of voters.

1.3 Scope

1. **Simple and Intuitive Design:** The e-voting system will feature a straightforward and easy-to-understand user interface to facilitate seamless participation by voters of all backgrounds. Emphasis will be placed on clarity and simplicity in the design of the voting process, minimizing the need for extensive technical knowledge or training.

2. **Blockchain Integration:** The system will utilize a blockchain for recording and storing votes in a transparent and tamper-resistant manner. Each vote will be stored as a digital record on the blockchain, ensuring that the voting data remains secure and immutable. However, the technical complexities of blockchain technology will be abstracted away from the end-users to maintain

simplicity.

3. **Basic E-Voting Functionality:** The primary functionality of the e-voting system will include voter registration, ballot creation, vote casting, and result tabulation. These features will be implemented in a straightforward manner, allowing voters to easily navigate through the voting process step by step.

4. **Centralized Management:** While the voting process will be facilitated by blockchain technology, the overall management and administration of the system will be centralized. This approach simplifies system maintenance and allows for more straightforward oversight and control by designated administrators.

5. **Web-Based Platform:** The e-voting system will be accessible through a web-based platform, allowing voters to participate in elections using standard web browsers on their computers or mobile devices. This ensures broad accessibility and eliminates the need for specialized software or hardware.

6. **Scalability Considerations:** Although the initial focus of the project is on simplicity, scalability considerations will be taken into account during the system design. The architecture will be designed to accommodate potential future growth in the number of users and votes without sacrificing performance or usability.

1.4 Objectives

1. **Security:** The proposed system aims to provide a secure platform for conducting elections, eliminating the possibility of tampering with votes, and ensuring that the election results are transparent and verifiable.

2. **Transparency:** The proposed system aims to provide complete transparency to the voters, allowing them to view the entire voting process, including the vote counting and results.

3. **Accessibility:** The proposed system aims to make the voting process more accessible to all eligible voters by eliminating the need for physical presence at a polling station, thus increasing voter turnout.

4. **Efficiency:** The system aims to increase the efficiency of the voting process by reducing the time and resources required to conduct elections. Since the system is automated and eliminates the need for intermediaries, it can significantly reduce the cost and time associated with traditional voting methods.

5. **Trust:** The proposed system aims to increase trust in the voting process by providing a transparent and tamper-proof mechanism for recording and tallying votes.

1.5 Hardware and software requirements for development

Hardware Requirements:

1 **Processor:** The e-voting system requires a processor with a minimum speed of 2 GHz or higher to ensure smooth performance during voting operations and data processing.

2 **RAM:** A minimum of 4 GB RAM is recommended to support the concurrent execution of various components and processes within the e-voting system.

3 **Disk Space:** The system should have at least 100 GB of disk space available to accommodate the installation of software components, storage of blockchain data, and logging of system activities.

Software Requirements:

1 **Node.js:** Node.js serves as the runtime environment for executing JavaScript code outside the browser, facilitating the development of server-side applications and APIs for the Digipolls system.

2 **Web3.js:** Web3.js is a JavaScript library that provides an interface for interacting with Ethereum-based smart contracts, enabling communication between the e-voting application and the Ethereum blockchain.

3 **Truffle:** Truffle is a development framework for Ethereum that streamlines the

process of building, testing, and deploying smart contracts. It provides tools for compiling Solidity code, managing project dependencies, and automating deployment tasks.

4 **Solidity:** Solidity is a programming language used for writing smart contracts on the Ethereum blockchain. It will be utilized to implement the logic and rules governing the e-voting process within the smart contracts.

5 **Ganache:** Ganache is a personal Ethereum blockchain for development and testing purposes. It allows developers to simulate a local blockchain environment, complete with accounts, transactions, and smart contracts, without the need for network connectivity.

6 **Metamask:** Metamask is a browser extension that enables users to interact with Ethereum-based decentralized applications (DApps) directly from their web browsers. It provides a convenient way for voters to securely manage their Ethereum accounts and sign transactions during the voting process.

7 **Python:** Python will be used for backend development, specifically for implementing the FastAPI framework to create RESTful APIs for communication between the frontend and backend components of the e-voting system.

8 **FastAPI:** FastAPI is a modern Python web framework for building high-performance APIs. It will be employed to develop the backend server responsible for handling user requests, processing data, and interacting with the MySQL database.

9 **MySQL Database:** MySQL will serve as the relational database management system (RDBMS) for storing and managing various types of data related to the e-voting system, including user accounts, voting records, and configuration settings.

Chapter 2

Literature Survey

This chapter presents an overview of the existing literature on blockchain-based voting systems. The focus is on understanding the technical advancements, challenges, and potential applications of these systems, as highlighted by several influential papers. This synthesis merges the conclusions of many different sources to explain the overall understanding of the topic and laying a foundation for both the problem statement and the proposed solution.

In the quest for modernizing the voting process to align with the demands of the digital age, blockchain technology has emerged as a promising solution. This literature survey aims to provide a comprehensive overview of the research conducted in the field of blockchain-based electronic voting systems.

[1] is a seminal paper provides a comprehensive review of blockchain-based electronic voting systems, outlining their advantages, limitations, and open research challenges. The authors discuss the key components of a blockchain-based e-voting system, including voter registration, voting, and result tallying. They explore various blockchain technologies suitable for electronic voting systems, such as Ethereum, Hyperledger Fabric, and Corda. The paper also Identifies several open research challenges, including scalability, interoperability, and user experience, indicating the need for further innovation in this area.

[2] proposes a secure end-to-end verifiable e-voting system that combines blockchain technology with cloud server technology. The authors demonstrate the security and efficiency of their system through a proof-of-concept implementation, highlighting the use of encryption and digital signatures to protect voter data and ensure the integrity of the voting process. The system is designed to provide end-to-end verification, offering transparency and accountability by allowing each vote to be traced back to the voter.

[3] presents a conceptual architecture for a blockchain-based e-voting system tailored for university elections. The authors propose a decentralized system where voters cast their votes using a mobile app, with the votes being stored on a private blockchain. This system aims to increase transparency, security, and voter participation. The paper discusses the use of smart contracts to automate the voting process, ensuring accurate and secure vote counting. Voter authentication is implemented through a secure mechanism to verify voter eligibility without revealing personal information.

[4] proposes an Ethereum-based e-voting system called Votereum. The system leverages smart contracts to ensure the integrity and transparency of the voting process. The authors demonstrate the feasibility of their system through a proof-of-concept implementation, showcasing the potential applications of blockchain technology in the realm of electronic voting. The system uses zero-knowledge proofs to ensure voter anonymity and prevent double voting. A decentralized application (Dapp) is created

that allows voters to interact with the voting system through a web browser, providing a user-friendly interface for casting votes.

The literature surveyed herein underscores the potential of blockchain technology to revolutionize the voting process by addressing the challenges of traditional voting methods. The papers collectively highlight the innovative solutions proposed for voter registration, voting, and result tallying, ensuring security, privacy, transparency, and user experience. However, they also identify several open research challenges, including scalability, interoperability, user experience, and voter education. Ensuring the security and integrity of blockchain-based e-voting systems in the presence of malicious actors is also an open research challenge. Further research is needed to overcome these challenges and ensure the widespread adoption of these systems.

Chapter 3
Project Design

This chapter delves into the system model and architecture of the proposed application. Various diagrams describing the different parts of the system and their interactions in order to successfully conduct a secure voting process are provided. Additionally, the timeline and distribution of project work is provided for giving an estimate of the time and resources dedicated to this project.

3.1 Proposed system model and architecture

User enters the credentials (voter id & password) and they are matched with the database. If the match is found user is either redirected to admin page or voter page as per their role corresponding to the credentials in the database. Once the admin is logged in he/she can start the voting process by adding candidates and defining dates. Voter can vote once the voting process has been started. Once the voter has voted the transaction is recorded to the blockchain and the voting page is updated with real-time votes.

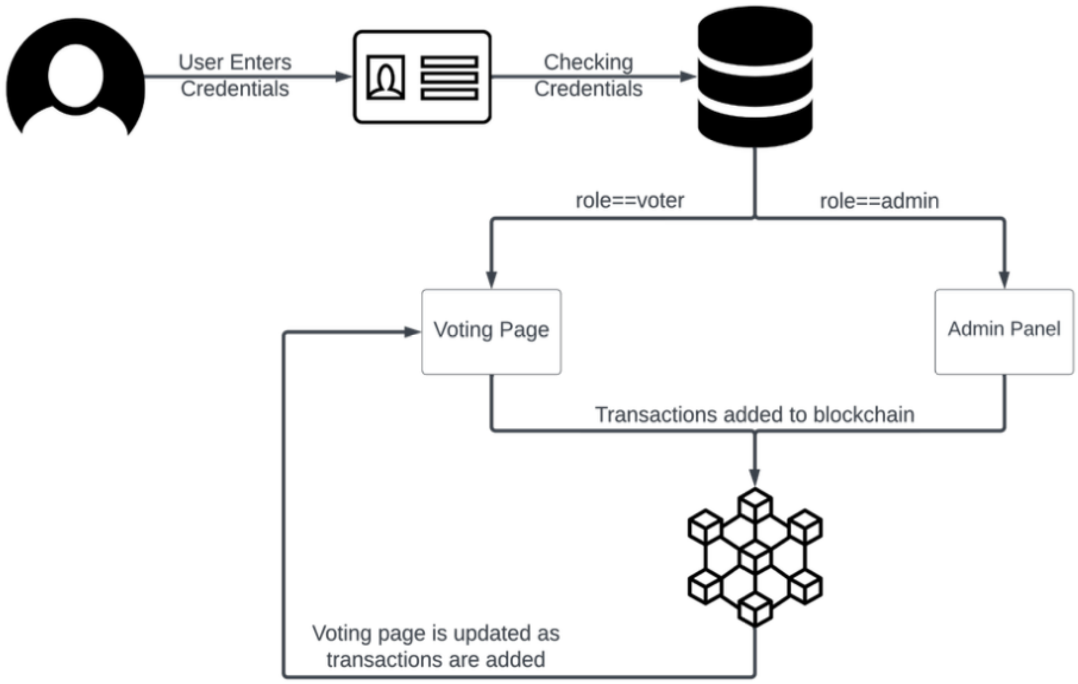


Figure 1: Digipolls system architecture

3.2 Software Project Management Plan

Activity	R1	R2	R3	R4	Mentor
1. Requirement Gathering					
1.1 Interaction with customer	C	C	C	C	A
1.2 Preparing SRS	R	C	R	C	A
2. Design					
2.1 Preparing Block diagram	C	C	R	R	A
2.2 Writing Functional Requirements	C	R	C	C	A
2.3 Writing Non-Functional Requirements	R	C	C	R	A
2.4 Developing Use Case	C	C	R	C	A
2.5 Developing Test Cases	R	C	C	C	A
3. Planning					
4. Coding					
4.1 Front End	C	C	R	R	A
4.2 ML Model	R	R	C	C	A
4.3 Integration	C	C	C	C	A
5. Testing					
5.1 Unit 1	E	A	E	A	
5.2 Unit 2	A	E	A	E	
5.3 System Testing	E	E	E	E	A

C: Creator, R: Reviewer, A: Approver, E: Executor

Table 1: Project Management Plan

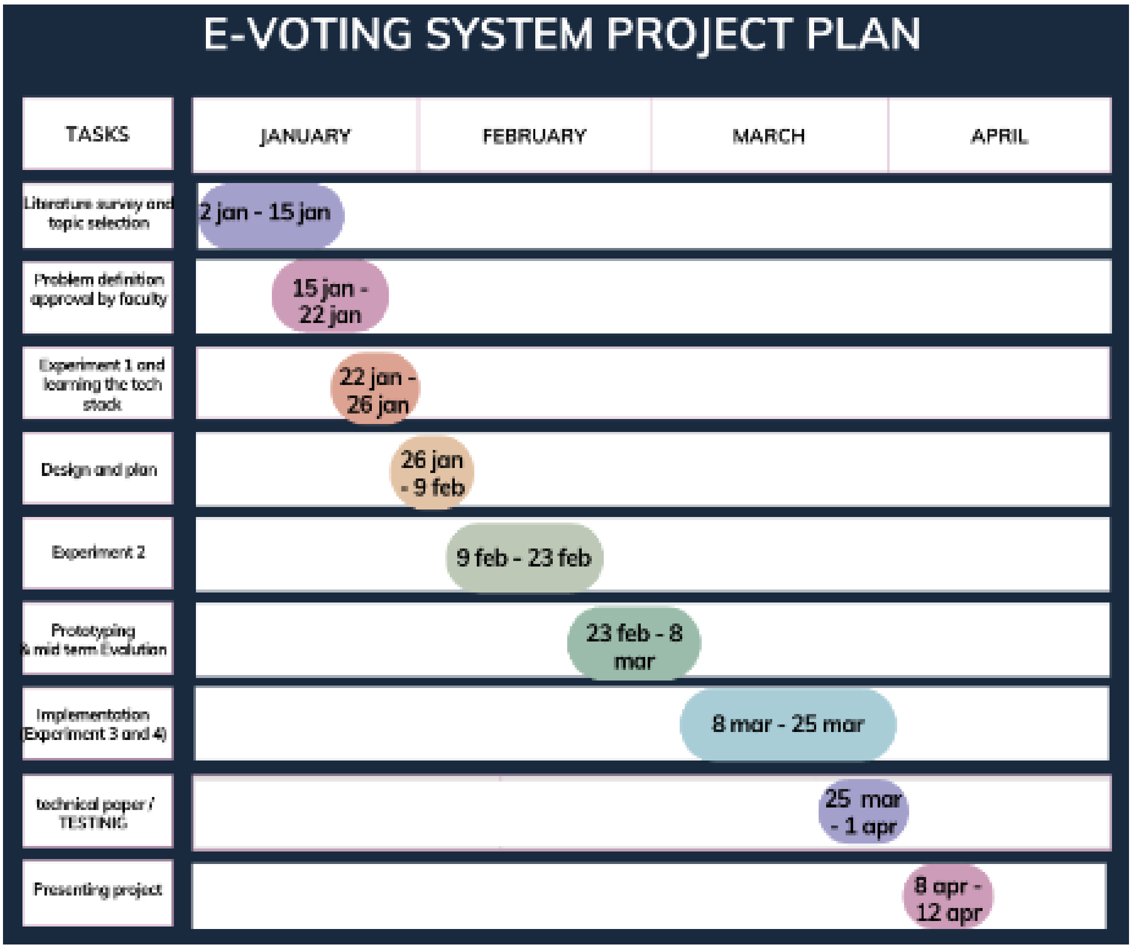


Figure 2: Project Gantt Chart

3.3 Software Design Document

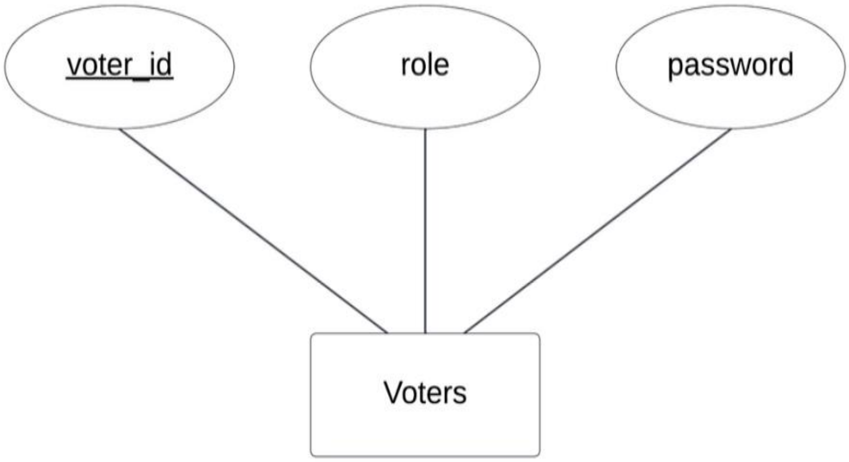
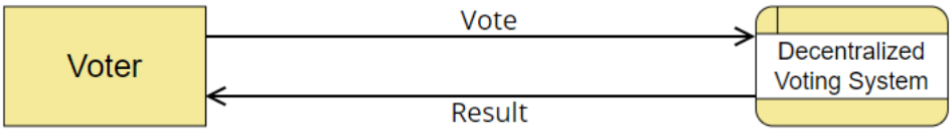
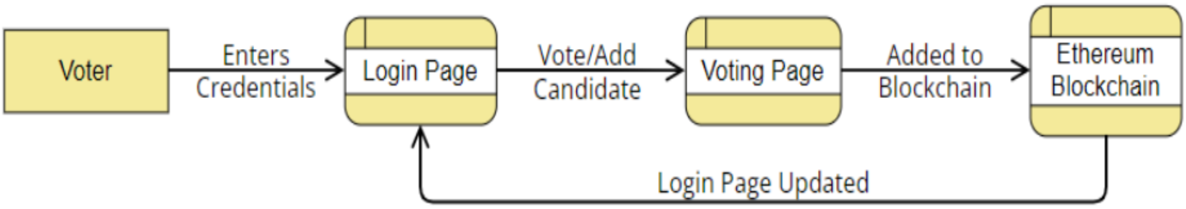


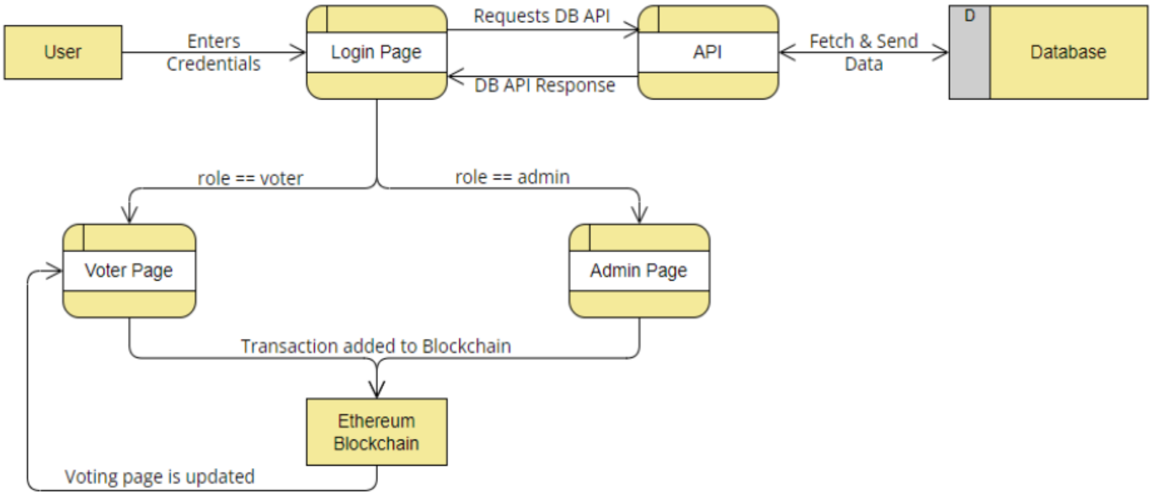
Figure 3: ER diagram



Level 0



Level 1



Level 2

Figure 4: Data flow diagram

Chapter 4

Implementation and Testing

This chapter presents the implementation of the proposed system model along with the additional details suggested during the progress seminar. It covers software testing reports at various levels, along with experimental results and their analysis, providing insights into the project's development process.

4.1 Proposed system model and implementation

User enters the credentials (voter id & password) and they are matched with the database. If the match is found user is either redirected to admin page or voter page as per their role corresponding to the credentials in the database. Once the admin is logged in he/she can start the voting process by adding candidates and defining dates. Voter can vote once the voting process has been started. Once the voter has voted the transaction is recorded to the blockchain and the voting page is updated with real-time votes.

Modules:

1. Voter

The voter module is designed for individuals who are eligible to participate in the voting process. It provides functionalities related to the voting experience and ensures the integrity and security of the votes. The main features of the voter module include:

i. Voters can securely authenticate themselves to access the voting system using their unique credentials.

ii. Voters can access information about the candidates running for various positions, such as their names, parties, and other relevant details.

iii. Voters can verify the status of their votes and ensure that their choices are accurately recorded in the blockchain.

2. Admin

The admin module is designed for administrators or election officials responsible for managing and overseeing the voting system. It provides functionalities to configure and monitor the voting process. The main features of the admin module include:

i. Admins can set up the system parameters, such as defining the start and end dates of the voting period, candidate registration, and other administrative settings.

ii. Admin can manually verify the candidate and can start the voting process.

4.2 Additional details suggested during progress seminar

Throughout the project, the project guide provided additional input and progress seminars were held to discuss and implement new ideas. This involved altering the scope and tech stack of the project for better implementation.

4.3 Software Testing

4.3.1 Unit Testing

Unit testing is a type of testing that is used to evaluate the individual units or components of a software system. This type of testing helps ensure that each unit or component of the system is working correctly and is able to perform its intended function. We had used the voters and admin module and unit testing for them was taken place.

4.3.2 Integration Testing

Integration testing is a type of testing that is used to evaluate how well the different units or components of a software system work together. This type of testing helps to identify and resolve issues related to compatibility, performance, and data flow between the different units or components.

4.3.3 Functional Testing

Functional testing is a type of testing that is used to evaluate how well a system or software performs the specific functions or tasks that it is designed to perform. It is done by testing the system or software with various inputs and verifying that the outputs are correct. This type of testing ensures that the system or software is working as intended and is able to perform the functions it was designed to perform.

4.3.4 White Box Testing

White box testing, also known as structural testing or glass-box testing, is a type of testing that examines the internal structure and implementation of a software system. It involves testing the code itself and checking that it is functioning correctly and adhering to coding standards. This type of testing helps to identify and resolve issues related to logic, control flow, and data structures within the system.

4.3.5 Black Box Testing

Black box testing, also known as functional testing, is a type of testing that examines the external behavior and interfaces of a software system. It involves testing the system from the user's perspective, without looking at the internal structure or implementation, and checking that it is functioning correctly and meeting the requirements. This type of testing helps to identify and resolve issues related to usability, compatibility, and performance.

Test Case No.	1
Test Type	Unit Test
Name of Test	Checking JWT Authorization
Test Case Description	The objective of this test case is to check jwt authorization.
Input	Login and Password
Expected Output	User should not be able to login without proper authorization.
Actual Output	User cannot access voting or admin page without authorization.
Result	Pass
Comments	Working properly.

Test Case No.	2
Test Type	Functional Test
Name of Test	Verify user login
Test Case Description	The objective of this test case is to verify that user can login to the voting portal.
Input	Voter_id and password
Expected Output	User must be able to login if credentials match the database, else unauthorized error is shown.
Actual Output	User is able to login with correct credentials only.
Result	Pass
Comments	Working properly.

Test Case No.	3
Test Type	Unit Test
Name of Test	Verify candidate registration
Test Case Description	The objective of this test case is to verify that candidate can be registered by admin.
Input	Candidate name and party.
Expected Output	Registration transaction should be successful.
Actual Output	Registration transaction is successful.
Result	Pass
Comments	Working properly.

Test Case No.	4
Test Type	Unit Test
Name of Test	Verify date registration
Test Case Description	The objective of this test case is to verify that date of voting can be specified by admin.
Input	Starting and ending date
Expected Output	Date transaction should be successful.
Actual Output	Date transaction is successful.
Result	Pass
Comments	Working properly.

Test Case No.	5
Test Type	Functional Test
Name of Test	Verify voting
Test Case Description	The objective of this test case is to verify that voter is able to cast their vote.
Input	Select a candidate and click "Vote" button.
Expected Output	Vote transaction should be successful.
Actual Output	Vote transaction is successful.
Result	Pass
Comments	Working properly.

Table 2: Test Results

4.4 Experimental Results and its Analysis

The implementation of the e-voting system yielded promising results, demonstrating the effectiveness of various features aimed at enhancing security, transparency, and usability. The following sections provide an analysis of the experimental results:

1. **Secure Voter Authentication and Authorization with JWT:** By implementing JSON Web Tokens (JWT), the e-voting system achieved robust authentication and authorization mechanisms. JWTs provided a secure means of verifying the identity of voters and authorizing access to voting functionalities. This ensured that only authenticated users could participate in the voting process, thereby mitigating the risk of unauthorized access and fraudulent voting activities.

2. **Utilization of Ethereum Blockchain for Tamper-Proof and Transparent Voting Records:** Leveraging the Ethereum blockchain proved to be instrumental in establishing tamper-proof and transparent voting records. Each vote cast by a voter was securely recorded as a transaction on the blockchain, providing an immutable and auditable trail of voting activities. This enhanced transparency and integrity, as the voting records were publicly accessible and resistant to manipulation or tampering.

3. **Removal of Intermediaries for a Trustless Voting Process:** The elimination of intermediaries from the voting process ensured a trustless environment, where voters could directly interact with the e-voting system without relying on third-party entities. Smart contracts deployed on the Ethereum blockchain autonomously executed the rules and procedures of the voting process, reducing the need for centralized authorities and enhancing the overall trustworthiness of the system.

4. **Admin Panel for Managing Candidates, Voting Dates, and Results Monitoring:** The inclusion of an admin panel provided administrators with comprehensive tools for managing various aspects of the electoral process. Administrators could effortlessly add and manage candidates, set voting dates and deadlines, and monitor real-time voting results. This streamlined the administrative

tasks associated with conducting elections and empowered administrators to oversee the voting process efficiently.

5. **Intuitive UI for Voters to Cast Votes and View Candidate Information:** The user interface (UI) of the e-voting system was designed to be intuitive and user-friendly, enabling voters to easily cast their votes and access relevant information about candidates. Clear and concise presentation of candidate profiles, along with straightforward voting procedures, facilitated a seamless voting experience for users of all backgrounds and technical abilities.

In conclusion, the experimental results demonstrate the successful implementation of key features in the e-voting system, including secure authentication, blockchain-based record-keeping, removal of intermediaries, administrative management capabilities, and user-friendly interfaces. These features collectively contribute to the security, transparency, and usability of the e-voting system, paving the way for fair and trustworthy electoral processes in the digital age.

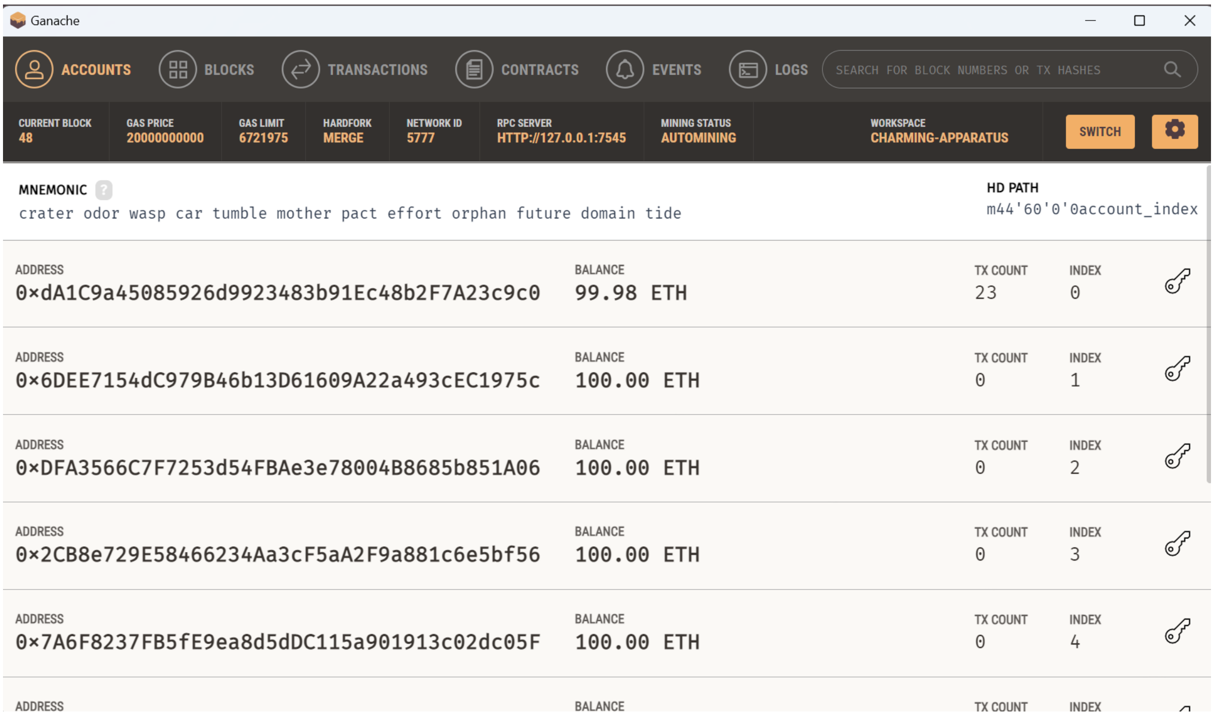


Figure 5: Ganache Interface – Accounts

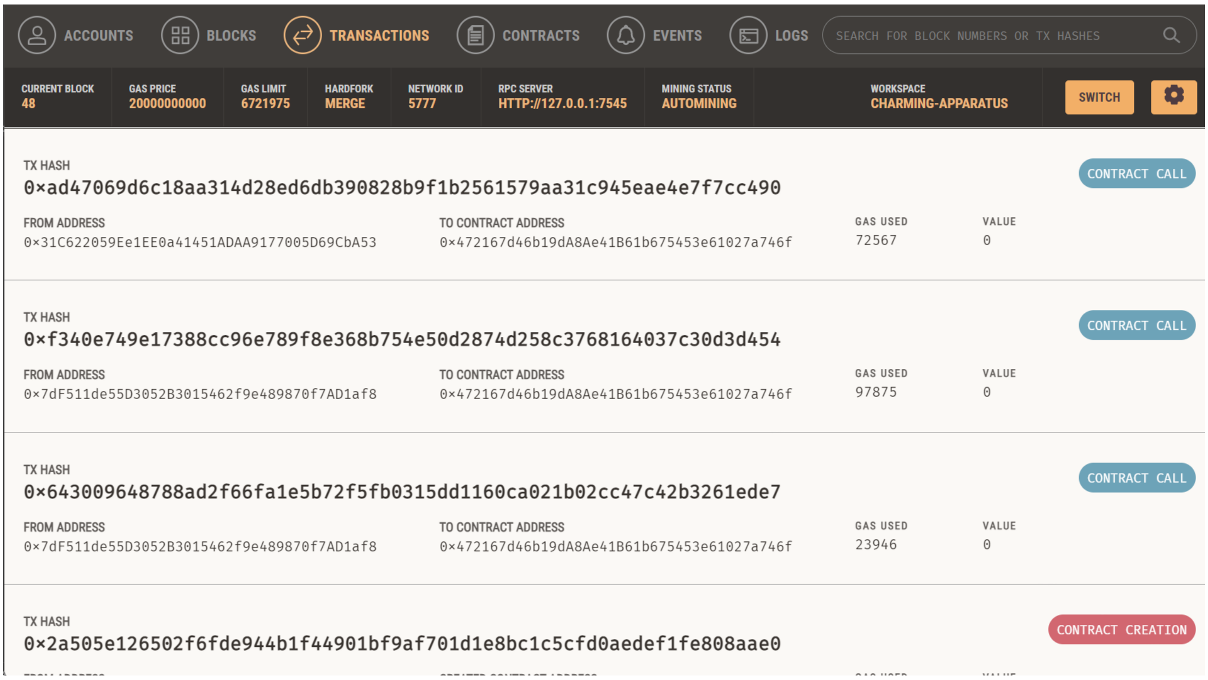


Figure 6: Ganache Interface - Transactions

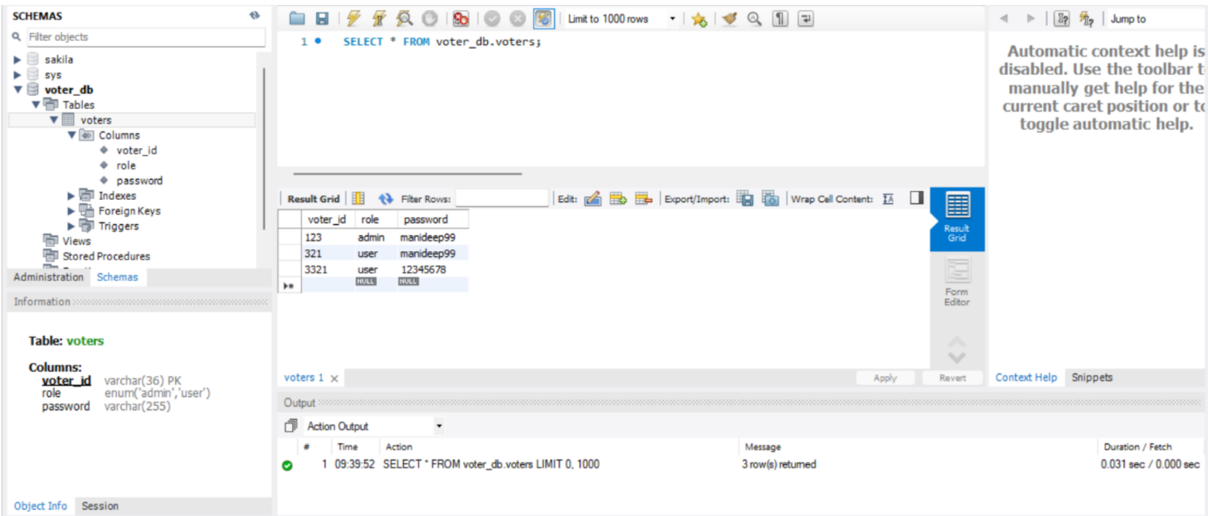


Figure 7: Database

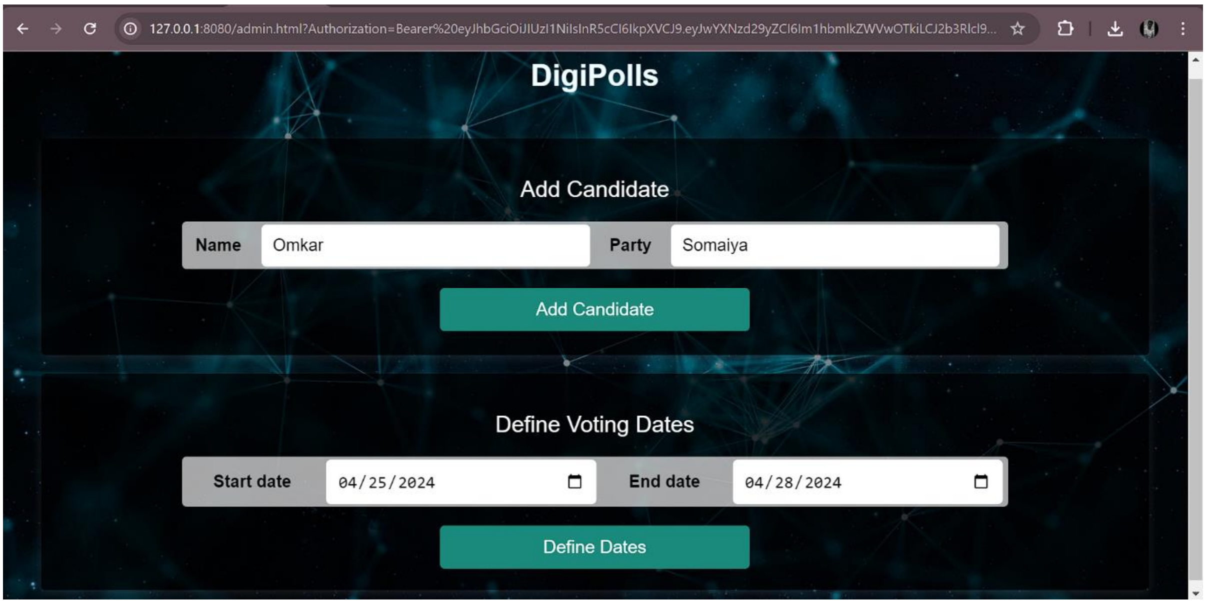


Figure 8: Digipolls - Homepage

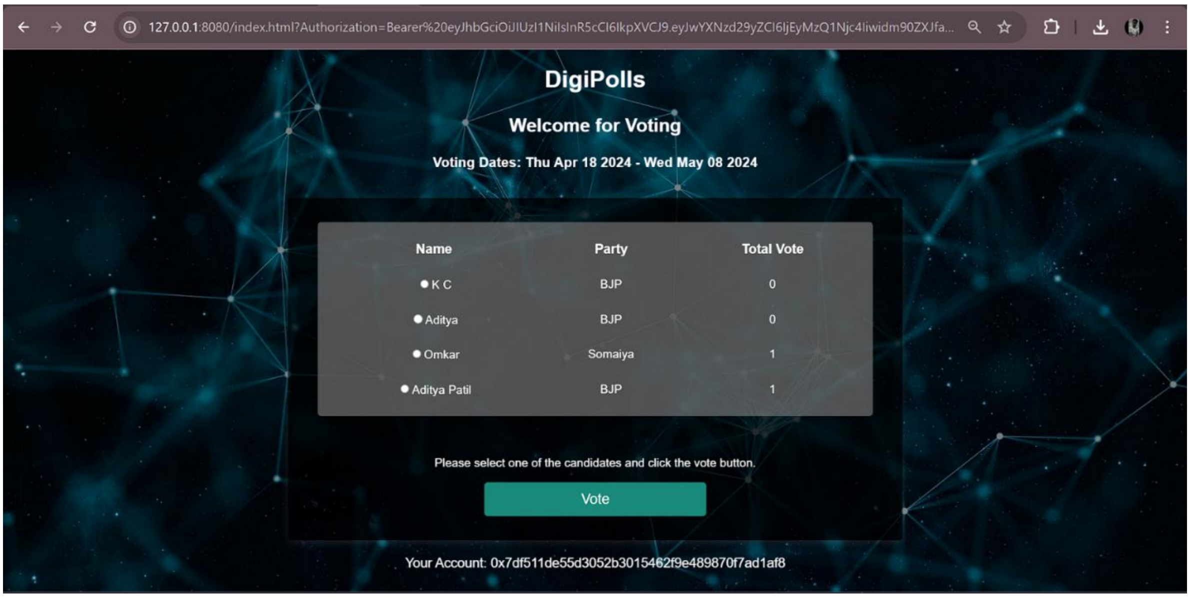


Figure 9: Digipolls - Voting

```
PS C:\Users\HZ069\OneDrive\Desktop\LAB\MP\Decentralized-Voting-System-Using-Ethereum-Blockchain> truffle migrate

Compiling your contracts...
===========================
> Compiling .\contracts\Migrations.sol
> Compiling .\contracts\Voting.sol
> Artifacts written to C:\Users\HZ069\OneDrive\Desktop\LAB\MP\Decentralized-Voting-System-Using-Ethereum-Blockchain\build\contracts
> Compiled successfully using:
   - solc: 0.5.16+commit.9c3226ce.Emscripten.clang


Starting migrations...
======================
> Network name:    'development'
> Network id:      5777
> Block gas limit: 6721975 (0x6691b7)


1_initial_migration.js
======================

   Replacing 'Voting'
   ------------------
   > transaction hash:    0x6d63a6165ef50bb93d33ea385349e6b974a156cc8fed26dab5693d2a837b20eb
   > Blocks: 0            Seconds: 0
   > contract address:    0x4A063aEe0e1801b0344D4b928609d335CE39174D
   > block number:        49
   > block timestamp:     1714108485
   > account:             0xdA1C9a45085926d9923483b91Ec48b2F7A23c9c0
```

```
   Replacing 'Voting'
   ------------------
   > transaction hash:    0x6d63a6165ef50bb93d33ea385349e6b974a156cc8fed26dab5693d2a837b20eb
   > Blocks: 0            Seconds: 0
   > contract address:    0x4A063aEe0e1801b0344D4b928609d335CE39174D
   > block number:        49
   > block timestamp:     1714108485
   > account:             0xdA1C9a45085926d9923483b91Ec48b2F7A23c9c0
   > balance:             99.978979578468560727
   > gas used:            732332 (0xb2cac)
   > gas price:           2.502036322 gwei
   > value sent:          0 ETH
   > total cost:          0.001832321263762904 ETH

   > Saving artifacts
   -------------------------------------
   > Total cost:     0.001832321263762904 ETH
Summary
=======
> Total deployments:   1
> Final cost:          0.001832321263762904 ETH


PS C:\Users\HZ069\OneDrive\Desktop\LAB\MP\Decentralized-Voting-System-Using-Ethereum-Blockchain>
```

Figure 10: Terminal Output after Voting

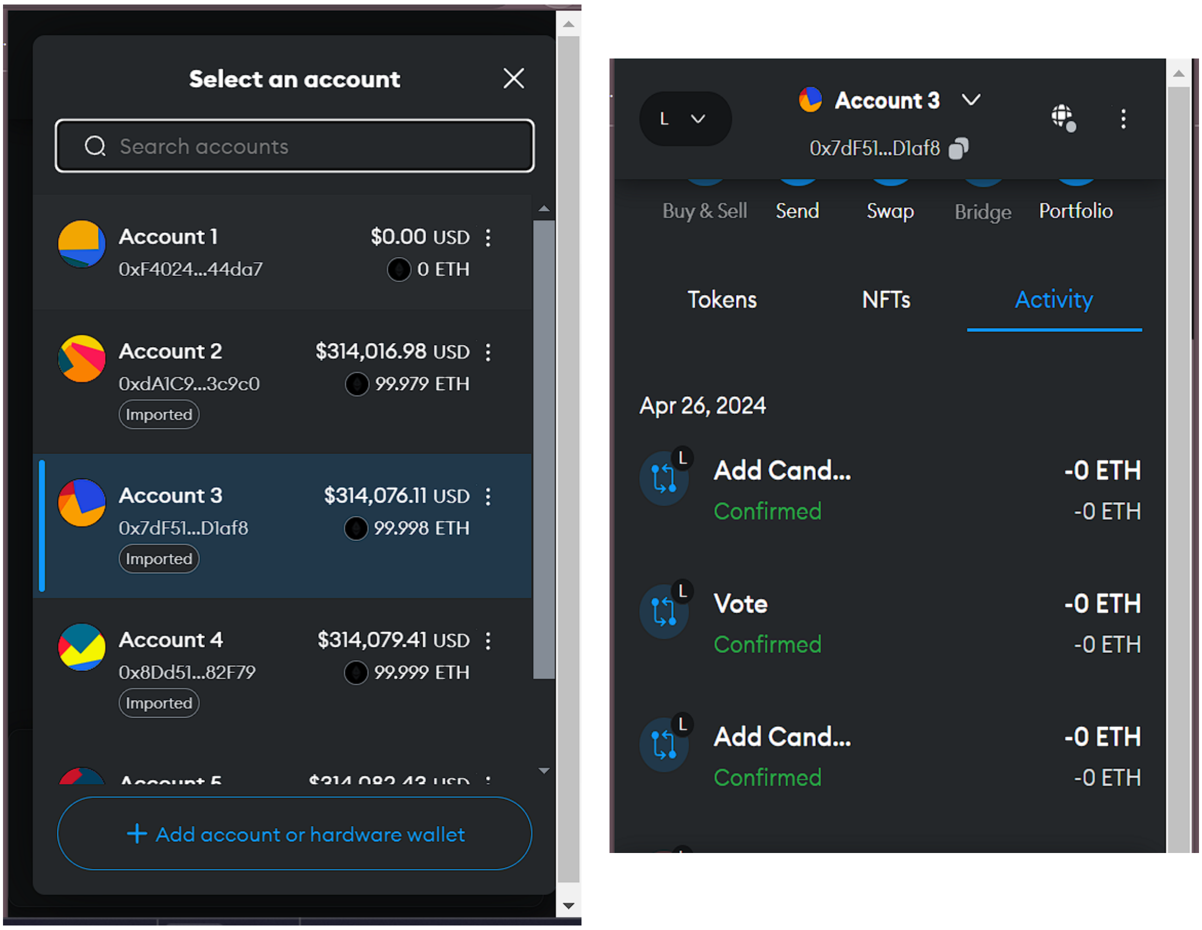


Figure 11: Metamask Interface

```
├── digipolls          # Root directory of the project.
    ├── build                    # Directory containing compiled contract artifacts.
    |   └── contracts
    |       ├── Migrations.json
    |       └── Voting.json
    ├── contracts                # Directory containing smart contract source code.
    |   ├── 2_deploy_contracts.js
    |   ├── Migrations.sol
    |   └── Voting.sol
    ├── Database_API             # API code for database communication.
    |   └── main.py
    ├── migrations               # Ethereum contract deployment scripts.
```

```
│   └── 1_initial_migration.js
├── node_modules              # Node.js modules and dependencies.
├── public                    # Public assets like favicon.
│   └── favicon.ico
├── src
│   ├── assets                # Project images.
│   │   └── eth5.jpg
│   ├── css                   # CSS stylesheets.
│   │   ├── admin.css
│   │   ├── index.css
│   │   └── login.css
│   ├── dist                  # Compiled JavaScript bundles.
│   │   ├── app.bundle.js
│   │   └── login.bundle.js
│   ├── html                  # HTML templates.
│   │   ├── admin.html
│   │   ├── index.html
│   │   └── login.html
│   └── js                    # JavaScript logic files.
│       ├── app.js
│       └── login.js
├── index.js                  # Main entry point for Node.js application.
├── package.json              # Node.js package configuration.
├── package-lock.json         # Lockfile for package dependencies.
├── README.md                 # Project documentation.
└── truffle-config.js         # Truffle configuration file.
```

Figure 12: Folder Structure

Chapter 5

Conclusion and future work

This chapter presents the conclusions derived from the development and testing of the Digipolls voting system. It highlights the results of the implementation and experimentation of the application and suggests ways to improve the existing features and also add new features for strengthening the impact of the project and bring it closer to achieving its mission.

5.1 Conclusion and discussion

In conclusion, our project aimed to create a simple and secure e-voting system using blockchain technology. We successfully added features to make sure the voting process is safe, transparent, and easy to use. We used a method called JWT to make sure only the right people can vote securely. By recording votes on the Ethereum blockchain, we made sure that nobody can change or cheat with the votes. Getting rid of middlemen means people can trust the voting process more. And with an admin panel, it's easier to manage candidates, voting dates, and results. Our easy-to-use interface lets voters cast their votes and learn about candidates without any trouble.

While our system is a big step forward, there's still room for improvement. We could make it even more secure, find ways to handle more voters, and make sure it follows all the rules. Overall, our project shows how blockchain can make voting fairer and more transparent. We're committed to making it even better to serve everyone's needs and keep democracy strong.

5.2 Scope for Future Work

Despite the successful implementation of the current Digipolls system, there are several avenues for future work and enhancements that can further improve its functionality, security, and scalability. The following are potential areas for future development:

1. **Enhanced Security Measures:**

- Implementation of advanced cryptographic techniques: Explore the use of more sophisticated cryptographic protocols to further enhance the security of voter authentication, ballot encryption, and result verification.
- Multi-factor authentication: Introduce additional layers of authentication, such

as biometric authentication or hardware-based authentication tokens, to strengthen voter identity verification.

2. **Integration of Additional Blockchain Features:**

- Smart contract optimization: Investigate techniques for optimizing smart contracts to reduce gas costs and improve the efficiency of transaction processing on the Ethereum blockchain.
- Integration with other blockchain platforms: Explore the possibility of integrating with alternative blockchain platforms to leverage their unique features and capabilities, such as scalability solutions or privacy enhancements.

3. **Expansion of Administrative Tools:**

- Automated candidate verification: Develop automated processes for verifying candidate eligibility and authenticity to streamline the candidate registration and approval process.
- Enhanced result analysis tools: Implement advanced data analytics and visualization tools to provide administrators with deeper insights into voting trends and patterns.

4. **Accessibility and Usability Improvements:**

- Mobile app development: Create dedicated mobile applications for both voters and administrators to facilitate on-the-go access to voting functionalities and administrative tools.
- Accessibility enhancements: Ensure compliance with accessibility standards and guidelines to accommodate users with disabilities and provide an inclusive voting experience for all individuals.

5. **Scalability Solutions:**

- Sharding implementation: Investigate the implementation of sharding techniques to partition the Ethereum blockchain and increase its transaction processing capacity, thereby improving scalability.

- Layer-2 scaling solutions: Explore the adoption of layer-2 scaling solutions, such as state channels or sidechains, to offload transaction processing from the main Ethereum network and reduce congestion.

6. **Adoption of Emerging Technologies:**

- Integration with decentralized identity solutions: Explore the integration of decentralized identity solutions, such as self-sovereign identity (SSI) frameworks, to enhance the security and privacy of voter authentication processes.
- Utilization of zero-knowledge proofs: Investigate the use of zero-knowledge proof protocols to enable private and verifiable voting, where voters can prove the validity of their vote without revealing their choice.

7. **Regulatory Compliance and Legal Considerations:**

- Compliance with regulatory requirements: Stay abreast of evolving legal and regulatory frameworks governing electronic voting systems and ensure compliance with relevant laws and regulations.
- Collaboration with regulatory bodies: Foster collaboration with electoral authorities and regulatory bodies to address legal and compliance challenges and ensure the adoption of the e-voting system within existing regulatory frameworks.

8. **Community Engagement and Education:**

- Voter education initiatives: Launch educational campaigns and outreach programs to increase awareness and understanding of electronic voting systems among the general public, fostering trust and confidence in the technology.
- Community feedback mechanisms: Establish feedback channels and community forums to solicit input and suggestions from stakeholders, including voters, administrators, and technologists, to inform future development efforts.

The outlined scope for future work presents numerous opportunities for further enhancing the e-voting system's functionality, security, and usability. By prioritizing these areas for development and innovation, the e-voting system can continue to evolve and adapt to meet the evolving needs and challenges of modern electoral processes.

Bibliography

[1] Uzma Jafar, Mohd Juzaiddin Ab Aziz, Zarina Shukur, **"**Blockchain for Electronic Voting System - Review and Open Research Challenges**",** https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8434614/

[2] Somnath Panja, Bimal Roy, "A secure end-to-end verifiable e-voting system using blockchain and cloud server", https://www.sciencedirect.com/science/article/abs/pii/S2214212621000557

[3] Simona-Vasilica Oprea, Adela Bâra, Anca-Ioana Andreescu and Marian Pompiliu Cristescu, "Conceptual Architecture of a Blockchain Solution for E-Voting in Elections at the University Level",

https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10049991

[4] Linh Vo-Cao-Thuy, Khoi Cao-Minh, Chuong Dang-Le-Bao, Tuan A. Nguyen, "Votereum: An Ethereum-Based E-Voting System", https://ieeexplore.ieee.org/document/8713661

[5] Sarah Al-Maaitah, Mohammad Qatawneh, Abdullah Quzmar, "E-Voting System Based on Blockchain Technology: A Survey", **https://ieeexplore.ieee.org/document/9491734**

[6] Zarif Khudoykulov, Umida Tojiakbarova, Suhrob Bozorov, Dilshoda Ourbonalieva, "Blockchain Based E-Voting System: Open Issues and Challenges", **https://ieeexplore.ieee.org/document/9670245**

[7] Shreyas Tandon, Niharika Singh, Shivani Porwal, Satiram, Ashish Kumar Maurya, "E-Matdaan: A Blockchain based Decentralized E-Voting System", **https://ieeexplore.ieee.org/document/9887759**

[8] Ashish Balti, Abhishek Prabhu, Sanskar Shahi, Shrutika Dahifale, Vrajesh Maheta, "A Decentralized and Immutable E-Voting System using Blockchain", **https://ieeexplore.ieee.org/document/10169552**

[9] Saurabh Singh, Alisha Singh, Shivam Verma, Rajendra Kumar Dwivedi, "Designing a Blockchain-Enabled Methodology for Secure Online Voting System", **https://ieeexplore.ieee.org/document/10053410**

[10] Aarti Naik, Ankit Prajapati, Shivam Pandey, Ashish Mishra, "Blockchain based e-voting system", **https://ieeexplore.ieee.org/document/10125883**

[11] Adam Hayes, "Blockchain Facts: What is it, How it works and How it can be used", **https://www.investopedia.com/terms/b/blockchain.asp**

Acknowledgements

We extend our sincere gratitude to Prof. Swapnil Pawar for his invaluable guidance and support throughout this project. His expertise and insightful feedback have been instrumental in shaping our ideas and refining our work. Prof. Pawar's mentorship has not only enriched our understanding of the subject matter but has also inspired us to strive for excellence.

We are also grateful to our esteemed colleagues for their collaborative efforts and constructive criticism during group discussions. Their diverse perspectives and valuable input have significantly contributed to the success of our project.

Furthermore, we would like to express our heartfelt appreciation to our family and friends for their unwavering encouragement and belief in our abilities. Their support has been a constant source of motivation, driving us to overcome challenges and achieve our goals.

Special thanks are extended to Prof. Vaibhav Vasani, Prof. Madhura Pednekar, and Prof. Kaustubh Kulkarni for their thorough evaluation and insightful feedback. Their constructive criticism has been instrumental in identifying areas for improvement and fostering our continuous growth and development.

Lastly, we acknowledge the support provided by K. J. Somaiya College of Engineering, which has facilitated access to resources and facilities essential for the completion of this project. The conducive academic environment has played a significant role in our academic journey.

We are sincerely thankful to everyone mentioned above for their invaluable contributions and unwavering support throughout this endeavor.