`

THE UTILIZATION OF BLOCKCHAIN IN BUILDING A DECENTRALIZED VOTING SYSTEM.

By

ANDREW SAMUEL TOBY

17010301019

A PROJECT SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE AND MATHEMATICS, COLLEGE OF BASIC AND APPLIED SCIENCES,
IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF DEGREE OF BACHELOR OF SCIENCE IN COMPUTER SCIENCE

2021

`

DECLARATION

I hereby declare that this project has been written by me and is a record of my own research work. It has not been presented in any previous application for a higher degree of this or any other University. All citations and sources of information are clearly acknowledged by means of reference.

_____

ANDREW, SAMUEL TOBY

_____

Date

`

CERTIFICATION

This is to certify that the content of this project entitled '**THE UTILIZATION OF BLOCKCHAIN IN BUILDING A DECENTRALIZED VOTING SYSTEM'** was prepared and submitted by **ANDREW SAMUEL TOBY** in partial fulfillment of the requirements for the degree of **BACHELOR OF SCIENCE IN COMPUTER SCIENCE.** The original research work was carried out by him under by supervision and is hereby accepted

_____ (Signature and Date)

Dr. F.A Kasali

Supervisor

_____ (Signature and Date)

Dr. M.O. Adewole

Coordinator, Department of Computer Science and Mathematics

`

DEDICATION

This project is dedicated to my mum, thank you for always being there.

`

ACKNOWLEDGEMENTS

I owe my profound gratitude to my Father, Mr Andrew Gabriel, for his constant support and encouragement throughout the development of this project, I also want to thank my siblings, most especially my older brother for his guidance, and regular motivation. I express gratitude to my supervisor Dr Funmilayo Kasali, for her guidance and support in ensuring the successful completion of this research. God bless you Ma.

I sincerely appreciate the Coordinator, Department of Computer Science and Mathematics, Dr. Adewole M.O., for his motivation, advice and teachings. My heart-felt gratitude goes to the members of staff throughout my undergraduate study, Late Dr. Oyetunji M.O., Dr (Mrs.) Oladejo Bola, Dr. Idowu P.A., Dr. Okunoye O.B., Dr. (Mrs.) Oladeji F.A., Mr and Mrs Taiwo.

`

TABLE OF CONTENTS

`

`

LIST OF FIGURES

`

ABSTRACT

In this era of development and technology, the need for a more advanced voting system is imminent, this study aims to develop a decentralized voting system that would be used to conduct elections by elucidating requirements needed in the systems' development, designing the system based on the elucidating requirements, developing and test running the system.

This system uses a reuse-oriented process model in development, the emancipation of blockchain and its application in building this system highlights all the benefits of this technology.

The ethereum blockchain was used in this project to create a blockchain environment that the transactions would be processed and a smart contract that defined how the architectural structure of the system would be defined. After implementation of this project, a student body presidential election was conducted to great success, showing an improved workflow and how it serves as a better alternative to the already existing voting systems.

The result of this system highlights the main benefit if using blockchain as part of its architecture, ensuring security and trust. In conclusion, blockchain technology can be used to strengthen the security of system and can be used as a data protection tool if applied toward the right domain. The system built in this project has already eliminated most of the flaws seen in traditional voting system but can be improved on. A further research into adding a better identification system would be greatly encouraged.

Keywords: Blockchain, Smart contract, Election, Decentralized System, Server.

`

CHAPTER ONE

INTRODUCTION

1.1 Background to the Study

In democratic nations, organizations, groups, and bodies, the need to select the right representative is a decisive decision, and the process surrounding this decision-making is elections. Elections are ways of choosing a representative by popular vote. In areas such as democracy and voting, Nigeria has consistently performed poorly in terms of stability, reputation, and safety (Aluaigba, 2016). The Nigerian government has struggled to keep its people safe and guarantee to hold free and fair elections over the years. From the 1999 election to the recently concluded election in 2019, the electoral administration has been flawed in several ways. Rigging, abuse, and coercion have all occurred during these elections (Agbu, 2016). The advent of fast-paced technology in the world has ignited a slew of new concepts and approaches to solving some of society's most pressing issues. Technology has advanced rapidly in the twenty-first century, not only in common sectors such as health and finance, but it can also help a nation better understand its claim to democracy. Democracy, as described by Abraham Lincoln (16th American President, 1861–1865), is "government of the people, by the people, for the people," There is no democracy without the people. This project revolves around the use of such technology in fixing some basic issues found in the current electoral system. Free and fair elections are means to demonstrate one's democratic duty.

Internet voting has been made accessible all over the world. In terms of security, confidence, privacy, and expense, governments and organizations have worked to improve the state of online voting (Suryavanshi, 2020). The disadvantages of paper ballots cannot be overemphasized. Paper voting, which has been conducted in Nigeria since 1999, has proven to be not only unsuccessful but also dangerous (Agbu, 2016). E-voting has been suggested to be implemented into the system to help minimize workload, costs, and increase citizens' confidence in the electoral body, and there is no better way to do that than by using blockchain technology as employed by (Ayed, 2017).

`

Blockchain is an in-alterable system that assures security due to its architecture, blockchain helps increase the security in the system by encrypting the information of transactions in a hash, It is easy to verify the authenticity of the system by checking the last node on the block for the results. Voting with blockchain would highlight all the limitations of voting via electronic voting machines, which have been imperfect in different ways. Electronic voting systems are not secured and anyone close to the system can manipulate the system.

The use of cryptography and the concept of decentralization in the sense of digital voting will be emancipated and expanded in this project. Blockchain will be used in conjunction with other technologies such as UI frameworks to construct a well-designed and trustworthy system that will enable voters to fulfill their democratic obligations.

1.2 Statement of the Problem

The use of ballot boxes has been a complete impediment to the full force of democracy. The biggest problem with paper-based voting has been lack of accountability, security, and confidence in the system and the results. The intervals between when the votes are counted and when the official results are announced to the public are lengthy and give room to corrupt individuals to manipulate such results. The Independent National Electoral Commission (INEC) is in charge of general elections in Nigeria, INEC's role in the democratic electoral process must ensure that the elections are considered safe for voters and voters have trust in the system they use in electing their preferred candidate. They are also responsible for the cost estimation and authenticity of the announced results. Most electoral bodies in the private or public sectors often perform the same duty, implementing elections with the old way of paper-based or electronic voting has not fully solved the issues that occur. These implementations are flawed in many ways and people often look for a solution to the problems. In the Nigerian electoral process through the years, results are not fully accepted by the declared losers of the election, and the citizens whose votes were counted are not sure if the right votes were counted, leaving the state of the election in a dilemma.

`

1.3 Aim and Objectives.

This study aims to develop a decentralized voting system that would be used to conduct elections, the following objectives would be performed to achieve this aim :

1. Elucidate system requirements needed to develop the system.
2. Define the design of the system concerning the specified requirements.
3. Develop the system.
4. Test the implemented systems against series of test cases.

1.4 Proposed Methodology

The work would be developed using a reuse-oriented process model in the development of the system. In this work, follow up to meeting the requirements of the objectives, numerous research studies would be carried out on previous works to decide how this work can be executed in the proposed environmental scope. User and System requirements would be elucidated and the use of an incremental approach to properly make updates to the system until a much-desired system is gotten. The use of a data collection system to collect information, the data collection system that would be used is a polling app that uses an actual database (no-SQL) to store polling results and details; for this, an investigation would be carried out on the system's shortcomings while emphasizing the key advantages of a decentralized server and system. The primary aim of the data collection system is to gather basic information about user reactions to the implementation and development, as well as their privacy and the validity of their vote. The user interface of this program will be implemented using a JavaScript framework called Nextjs. Due to the render time of most basic application and their various test cases, Nextjs was used due to its efficient server-side rendering capability, this would enable users to get the best experience while utilizing the voting application. Information about the blockchain used will be rendered on the server-side before reaching the user interface, this is to avoid a longer load time for the user. The app would also be subjected to series of tests, the smart contract would be tested using a javascript testing library and using other online test tools. The smart contract would also be deployed on the test network, to simplify the process as well as minimize the cost in development.

`

1.5 Scope and limitation of the Study

This project illustrates the use of blockchain technology in building a decentralized voting system. The system takes into consideration the new integration of the NIN scheme and uses it as part of its verification project. The system tries to revolve around the Nigerian ecosystem and assumes that registered users should have basic apps installed in other for the voting process to work. The system models election cases as individual elections and the ability for admins to create more than one election instance.

The system shows some basic limitations in the architecture such as the validity of users. The system assumes everyone with a valid NIN should be given the right to vote. Implementation of an artificial intelligence-powered system that monitors user status would be a valid and improved way to improve the authenticity of the users. The system also registers users individually. This is because the NIN data used in the project is a test file and not the actual NIN, thus; making the project limited. The system also assumes that all users have certain extensions installed, this might not be the case because not all the voters would have these extensions installed on their browsers.

1.6 Significance of the Study

The use of blockchain in the development of several systems in the modern world has been long encouraged and has proven essential. Since the development of the first cryptocurrency; Bitcoin (Nakamotoi, 2009), it has been widely used in various areas of the world. The cryptoeconomics of the world ranging from bitcoin to ethereum to dodgecoin, are just the brief surface on which cryptography can improve. Another sector is the cyber-security area, here, trust is also the main goal to achieve in the system. Blockchain in a voting system assures one thing that the paper ballot process fails to deliver, trust and security. Blockchain voting also offers full anonymity of the voters, this is why various research has been conducted by students and scholars to help improve the state of blockchain as a secured authentication system. This body of work is also directed at the whole Nigerian community, addressing the negative effects of our democratic process and the use of Blockchain technologies to build out the software solution.

`

Blockchain is a method of recording and verifying records that are distributed transparently by users of a specific network. In this analysis, the preferred network should be any individual who wishes to join such a network to determine the integrity of the electoral system. To ensure complete accountability, the entire source code should be hosted on GITHUB. Using the maximum power of Blockchain, any transaction in Blockchain is linked to the previous one, and discovering a way to add this functionality with an application would be beneficial.

1.7 Definition of Terms

- Blockchain: A blockchain is a public ledger of transactions. The name comes from the database's structure, which consists of individual records called blocks that are linked together in a single list called a chain.
- Smart-Contract: A smart contract is a technique for digitally validating contract agreements.
- Solidity**:** Solidity is a programming language for creating smart contracts that was created with Ethereum's Virtual Machine in mind.
- Framework: A software system is a physical or abstract platform that allows developers or users to specialize or avoid common code by using generic features. Frameworks are libraries that have a well-defined application program interface (API) and can be used anywhere in the development process.
- Election: An election is a process in which voters elect a person or a group of persons to fill a public office.
- JavaScript: JavaScript is a scripting or programming language that allows you to add advanced functionality to your web pages.
- Database: A database is a structured collection of data that is saved and retrieved electronically via a computer system.
- Decentralized-System: A decentralized structure is a knowledge network in which no single body wields supreme authority. In the context of networking and information technology, decentralized networks are often represented by networked computers.

`

- Server: A server is a software or hardware system that receives network requests and answers to them.
- Encrypt: Using a code or cipher to disguise information.
- Decrypt: To convert an encrypted or coded message back into plain text.

`

CHAPTER TWO

LITERATURE REVIEW

Elections have been a way for people to make decisions on what seems to be the right choice out of a selection of options made available to them, from counting raised hands to the use of ballot boxes, the need for people to have a say in role allocation processes has always been imminent.

Elections have been the driving force of democracy, democracy itself is defined by the people, and voting has been a means for these people to select the right candidate. Different voting mechanisms have been invented through the years and one of the oldest ways of voting is the use of paper voting using the ballot boxes. Elections can be conducted in different areas by people in different locations, from student prefects in secondary schools to the community chairman in rural areas and then to largely organized elections like local government elections or state or even federal elections.

In this section, important concept relating to this project was explained and a review into similar works was emancipated highlighting similarities and the flaws present in some of this implemented system.

2.1 Paper-Based Voting

One of the oldest ways in which people have been able to vote in elections has been to use paper ballots for voting, this method of voting can be performed in different ways assuming different security measures. Paper ballots have been used in classrooms, office voting, and even general elections held in countries. This has proven to be effective when compared to the counting of raised hands and oral voting.

In Nigeria, the use of the paper ballots to vote has been used since the first election took place, ranging from Dec. 5, 1998, local elections to Jan. 9, 1999, state election to Feb. 20, 1999, National Assembly elections; and the generally observed presidential election on Feb. 27, 1999. In research from The Carter Center and National Democratic Institute for International Affair (1999), summarized the Nigerian elections and the use of the ballots boxes, from the rigging of the elections to the people's lack of trust in the whole electoral process.

The use of paper-based voting has proven to be slightly ineffective and also risky according to (Aluaigba, 2016), elections are easily manipulated and the voting

`

grounds with these ballot boxes have been deemed unsafe for voters due to the insecurity at times at those locations.

2.1.1 Challenges of Paper-based voting

The risk associated with paper voting has led people to sought new ways to select candidates and here are some of the challenges that paper-based voting faces :

1. Untrustworthy:

Most people who vote in paper-based elections often find loopholes and irregularities in the electoral process and therefore are not fully convinced that the election was conducted right. They mostly do not trust the system, and they don't think the system is reliable. Other concerns for this paper-based election have to do with the centralization of the elections, the fact that the results can be manipulated by whoever is in charge is another concern that people have. Another loophole in the paper-based election has to do with the fact that there is no way for the user to know that his or her votes have been counted as part of the valid votes.

2. Costly:

The paper-based elections held are mostly costly, the budget allocated to these elections could be used more efficiently in other areas and other sectors of the economy.

3. Safety Risk:

If the elections are not free and fare then no one would want to participate, elections held in Nigeria are not always free and fair and sometimes people pay dearly with their lives. Security concerns always arise whenever election season comes on. According to research by (Aluaigba, 2016), the elections held in Nigeria are not safe for citizens. The paper-based elections polling grounds have to be surrounded by several security officials and even then the chances of an attack are still there.

4. Long Queues and Time Mismanagement:

There are also queues at the polling grounds and sometimes these queues are long that voters are not interested in waiting, this also has to do with the long

`

setup of voting and electoral officials, election stated to start from nine o'clock can start at eleven due to the lengthy setup and wait time, during this process new arrivals join the existing queues making the queues long.

5. Unsuitable voting environments:

The environments chosen by the electoral committee to host these elections as well as the polling times are inconvenient to the voters at times.

6. Lengthy-time for results:

Voters are made to wait a long time before the results of the votes would be announced.

2.2 Electronic Voting Systems

The use of an electronic voting system has eradicated almost all the issues associated with voting via the paper and ballot boxes processes, such as counting the votes manually and all the mistakes that could occur during such processes. The first country to use an electronic system for national elections was Estonia (Madise & Martens, 2005).

The limitations of using this system were identified and the flaws were stated, most of the organizations that use electronic voting systems mostly hide certain parts of its source code, thus; making it impossible for the users to trust the system. In the year 2013, the source code to the discontinued Switzerland electronic election was released online and it was discovered by researchers that someone can replace all the valid ballots with fraudulent ones through a cryptographic backdoor.

As part of the risk identified, the case for centralization of the said digital elections was also brought up, this indicates that there is a central source that controls what goes on in the system, according to (Taş & Tanrıöver, 2020) who further explains the effects of such voting processes.

Due to the problems faced by the use of a centralized database in election processes, the use of blockchain technology has been suggested as a solution, A Blockchain is a public transaction ledger. The term stems from the form of the database, in which individual records, known as blocks, are joined together in a single list known as a chain. This allows various information to be formed as part of the chain attributes and the use of unique hash to identify various blocks. Blockchain assures integrity in the

`

system by providing a transparent architecture as well as an almost impenetrable system that proves almost impossible to attack.

2.2.1 Challenges of Electronic voting

Even though the use of electronic voting has helped curb the problems associated with the use of traditional paper-based voting, there are still basic limitations associated with them.

Electronic voting has one major issue, the concept of centralization. The fact that most electronic voting systems have a central database, the risks associated with this are stated as follows:

1. Database can be manipulated by the allocated “admin” of the systems.
2. Most databases face the risk of being hacked.
3. Centralized databases can often reach limits of query due to the servers that manage them.

Electronic voting according to research by (Challenge of E-Voting, 2016) are generally susceptible to :

1. Hacking Online vote
2. Hacking Voting Machines
3. Hacking Election Campaigns

All these factors are why most organizations and governments try to avoid electronic voting with central databases

2.3 Blockchain

The blockchain was invented by Satoshi Nakamoto in 2008 (Nakamotoi, 2009), as a public ledger for a cryptocurrency called Bitcoin. Bitcoin is an ever-growing distributed ledger that consists of records, that are linked using cryptography. These records are called blocks. Each block contains a timestamp, a cryptographic hash of the previous block, and data of the transaction. The entire architecture of the blockchain is secured by its distribution among different nodes in a network, this is what makes it a distributed ledger and this makes it impossible for anyone to manipulate the data on the chain, and since the blockchain is not only present in one node or system, this makes it decentralized since more than one person has access to it.

`

There are three main types of blockchain, and they are grouped into the following areas :

1. Public Blockchain: In this type of blockchain, there are no restrictions to its access. Meaning, anyone, anywhere has access to the blockchain, writes to the blockchain by performing transactions, and then even becoming a validator on one of the nodes. Another name for this type of blockchain is permission-less blockchain.

2. Private Blockchain: This type of blockchain is mostly created by private organizations to serve a certain purpose, they put restrictions on who can read and write on the chain, as well as who gets to validate it.

3. Consortium Blockchain: This type of blockchain is also a private blockchain, but is after being restricted by a certain organization the ownership is divided among several organizations.

To further secure the various blocks available in the chain, a consensus mechanism is utilized by the various nodes in the network, this mechanism which is an algorithm ensures that all the transactions in the chain are valid and also ensures that all node keeps the same copy of the transactions, this is why when a transaction is being made, it is broadcast along with all the nodes in the network and everyone holds an identical copy. We have different kinds of consensus algorithms, with the Proof of Work(PoW) and the Proof of Stake(PoS) models mostly preferred as a consensus algorithm in the blockchain infrastructure. In a proof-of-work model, the transactions are verified as inimitable, secured, and reliable, the people or individuals performing a transaction can send a transaction fee for the verification of the transaction that has taken place. To illustrate this process, a blockchain will be created in javascript to show how the proof of work mechanism is used to verify the transactions that have taken place in the blockchain.

In the diagram below, blockchain is created using javascript and a reward is given to the first person to verify the current block, this is called mining. A brief mathematical problem is given and the first node to solve that problem creates a hash for the current block and the block is verified and added to the blockchain.

The proof of stake model works in different ways, in this model new chains are added to the previous block through staking, and the validators are chosen based on the amount or number of coins they want to stake, instead of the proof of work model

`

which requires computational power to verify a block. This could get expensive based on the mathematical problem given, and the energy consumed is much.

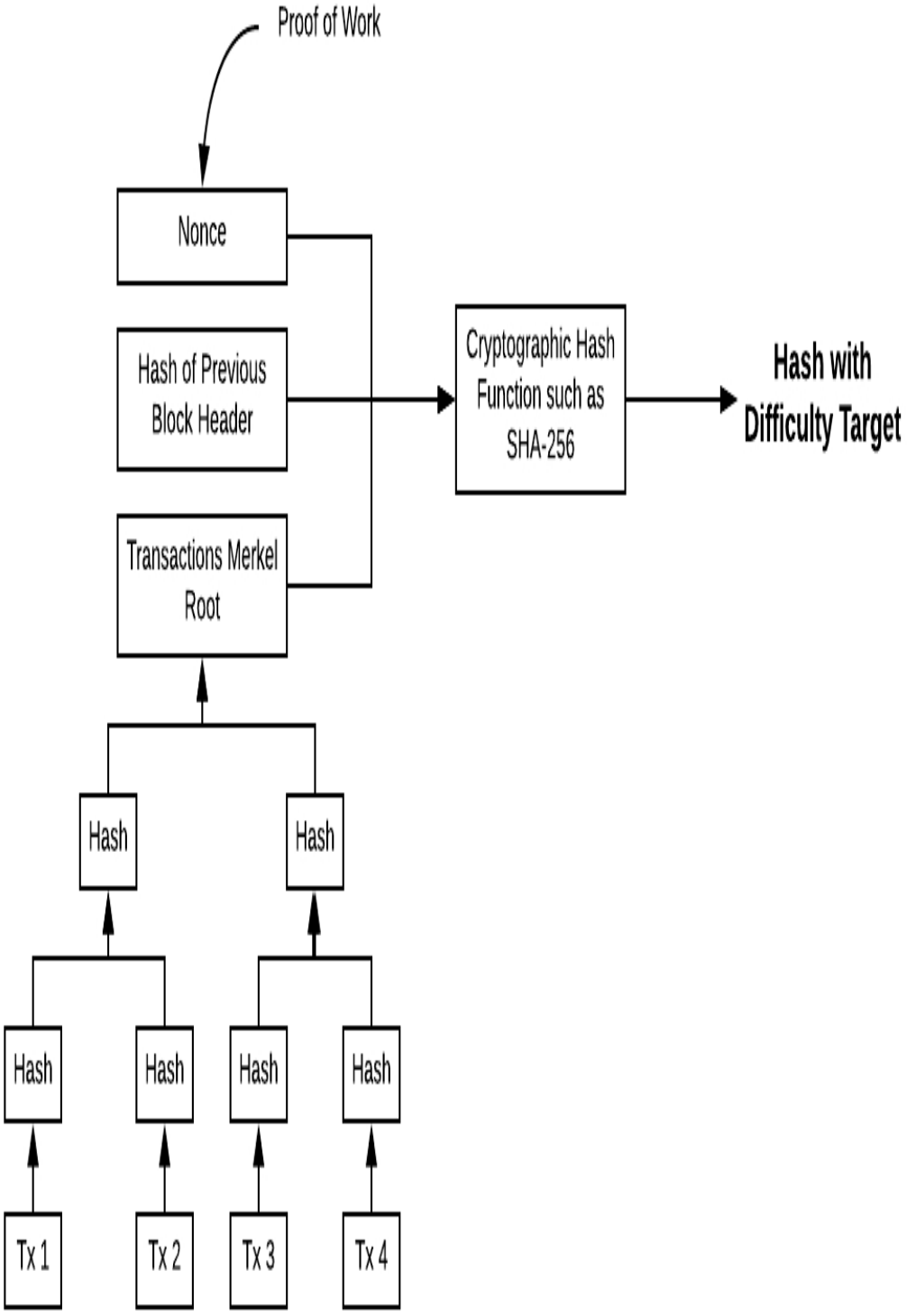

Fig 2.1: Proof of work in a blockchain (Kumar, 2018)

`

2.3.1 Application of Blockchain

Blockchain has been applied in various areas of our everyday lives now, mainly to propose transparency in whatever organization that uses it or to address most concerns on security and trust. Using blockchain we can also address identity and fraud management (Tykn, 2021). Most financial institutions can make use of blockchain to pore over clients and tackle these fraudulent activities, blockchain can also be used to facilitate payments, the use of bitcoin (Shift4Shop, 2020), and other crypto-currencies in the world of commerce has solely improved how people enable transactions. The reduced fees being paid on transactions that could take large sums in the banking system has enabled a higher interest in using crypto-currencies in transactions, for example, according to (Chandler, 2021) in April 2020, a sum of $1.1 billion (161,500 BTC) was transferred, Aside from a large amount of bitcoin involved, the transaction's cost is remarkable. It was incredibly cheap, costing only 0.00010019 BTC (about $0.68 at the time). The transaction charge when compared to that of actual banks is small and minute.

Major game changers in the industrial majorities of the world have started using blockchain and its applications in their everyday businesses. By doing this, it is easier for cooperation to process and manage their businesses more transparently. The blockchain infrastructure helps in addressing the security concerns in their systems, as well as tackling identity and fraud. Blockchain-based systems give power to financial institutions to screen clients and monitor various fraudulent activities. Dominantly, Banking institutions are transaction handlers are the suitors of the blockchain system, they are not limited to these areas. Another sector that can be impacted by blockchain is the sustainability of logistics management and B2B commerce, it does this by promoting collaboration between consumers and manufacturers, helping people adopt more sustainable existences, and helping companies improve their resources and re-utilizing process. By assuring security, transparency, and traceability, blockchain has the potential to have a significant impact on supply chain management.

Each transaction in supply chain management is carried out on a standard blockchain, without the requirement for clearance from a trustworthy center. After the delivery phase is completed, the payments are made automatically. Because the parties monitor the transactions, the blockchain has the potential to greatly improve product end-to-end monitoring and security. As a result, before purchasing a product, the

`

consumer can receive correct information regarding process transitions. Process visibility, integrity, transaction speed, and disinter-mediation are all major advantages.

Blockchain can be used in different sorts of smart services and internet-based applications, in addition to financial industries and supply chain management. The energy sector has used blockchain technology as well. Projects for local energy trade and effective charging schedule planning for electric cars are proposed. Machines can sell and buy energy using Internet of Things (IoT) devices that have been pre-defined. When IoT devices capture data in real-time, the information can be kept in a blockchain chain. It is now being utilized for real-time big data analysis. Another industry that makes use of blockchain in the insurance industry. The insurance industry today is built on trust, and it is expected that in the future, blockchain will be able to handle the occasional error or delay (Loukil, Abed, & Boukadi, 2021).

Other areas that blockchain can be applied to according to (Koksal, 2019) are:

1. Advertising
2. Stock Trading
3. Nonprofits
4. Government
5. Internet of Things
6. Real Estates

2.3.2 Blockchain Voting

As part of the different applications of blockchain, blockchain can also be applied to the voting section, using blockchain for voting, organizational bodies can leverage all the attributes of the system in implementing a secured system. A lot of research has gone into the advantages of using blockchain for voting, these advantages are transparency, the state of it being immutable, audit-able, and reliable. Voting with blockchain can be implemented using smart contracts, Zcash, custom blockchain, and cryptographic signature.

2.3.3 Voting with Smart Contracts

Due to its decentralization and each node operating on a consensus algorithm, the blockchain is considered an immutable, secure data structure. Ethereum makes use of

`

this property by expanding its blockchain with smart contracts. A smart contract is a blockchain-based application that processes incoming information.

Essentially, it is a script deployed on the blockchain that executes automatically as its functions are called. As such, it cannot be illegally removed or manipulated once written. This means that it can work transparently and autonomously without any external assistance. Many applications that would normally require a web server to function can be run through a smart contract instead. This project employs the use of smart contracts in its voting mechanism using the solidity language. The smart contract is used to structure each election process, from registering the voting candidates to registering the end-user and all the functions available in the electoral system. The smart contract created would have to be complied with a compiler, in this case, the solidity language will be used to create and compile a contract, two import meta-data would be used when interacting with a framework. The compiled contract abi, which is the contract interface that contains no code and can not be run on its own. Contract abi defines the methods and structures that are used to communicate with the binary contract, similar to API but at a lower level. The second one is the bytecode, the byte code is a series of the hexadecimal representation of the final contract that only the ethereum network can understand.

2.3.4 Voting with Zcash

Zcash is a decentralized blockchain-based payment system that aims to keep transactions anonymous. The proof-of-work mechanism, in which Zcash relies on zero-knowledge proofs, is one of the greatest distinctions between it and Bitcoin. Zcash features two sorts of addresses, unlike Bitcoin, which only has one. This allows it to handle both anonymous and transparent transactions. These addresses are z-address and t-address, where z-address maintains anonymity in transactions while t-address is structured similarly to Bitcoin addresses and enables transparent transactions. Zcash was chosen because it allows users to remain anonymous while voting by allowing them to pass both private and transparent values during a transaction. Zcash is a cryptocurrency derived from the Zerocoin protocol, which was created to hide the trail of Bitcoin transactions. Another way of voting with zcash uses Voters can pay to cast additional votes for a candidate or issue they want. This means

`

that the final result is more closely linked with the strength of voter preferences than merely following the majority vote.

2.3.4 Voting with Custom Blockchain

Another option in the blockchain voting system is to create custom voting, there are several ways to create a custom blockchain in other to conduct elections. According to research by (Khan, Arshad, & Khan, 2018), this can be implemented using Java EE while hosting the application on the Glassfish server which holds the applications EJBs and data source.

2.3.5 Voting with Cryptographic Signature.

Another option in the blockchain voting system is to vote using a cryptographic signature, there are several ways in which this can be achieved. According to research from Ibrahim, Kamat, Salleh, and Aziz (2003), voter's privacy is guaranteed by using a blind signature for confidentiality and voter's digital signature for voter's authentication, the system was implemented using Java socket technology and BouncyCastle cryptography provider.

2.4 Criteria for E-Voting and the Proposed System's Conformance

According to research from Rura, Issac, and Haldar (2011), There are certain criteria that e-Voting systems need to meet for it to be considered efficient. The criteria are:

1. Privacy - Keeping an individual's vote secret:

   The system should keep the privacy of users by protecting their choices in the elections, as the user registers on the election, a voter hash should be generated by the blockchain. If this is accomplished, it is difficult to trace which user voted for whom.

2. Eligibility - Only those registered should vote and should vote only once:

   The system should only allow the people considered eligible for the elections to vote, it is important that the users also be authorized and the electoral chairmen are responsible for identifying who is eligible for which election.

3. Receipt Freeness - Secrecy:

`

Apart from the voter and the electoral commission, no one should be told whom the voter voted for, this also helps for verification, because a hash is generated for the vote count of the user, making the vote unique and untraceable.

4. Convenience - Voting should not be stressful:

When users are voting, the voting process should not bore them into giving up on the election. Electoral commissions should create environments or user interfaces that are easy to access, with a good user experience, ensuring that human-computer interactions design principles are met.

5. Verifiability - Trust the process:

When a user votes a hash is generated for the user, the user should be able to tell that the vote was counted by checking the hash in the vote count list, but the hash should not reveal or disclose who the user voter is for.

2.5 Ethereum

Ethereum is a cryptocurrency-transfer system that allows you to send cryptocurrency to anyone for a nominal charge. It also powers open-source programs that no one can takedown. It is the first programmable blockchain in the world. With a few key distinctions, Ethereum expands on Bitcoin's innovation. Both allow you to use digital money without the usage of payment providers or institutions. However, because Ethereum is programmable, you may use it to create a variety of digital assets, including Bitcoin. This means Ethereum can be used for more than just payments. It is a financial service, gaming, and app store that won't steal your information or censor you.

2.6 Nextjs

Nextjs is a vercel-created open-source react front end framework that allows for server-side rendering and is used to develop static webpages for react apps. It is a production-ready framework that lets developers create static and dynamic websites quickly. Nextjs is one of several recommended "toolchains," all of which give a layer of abstraction to help with typical tasks. Nextjs was chosen for this project due to its server-side rendering capabilities as well as its ability to create API endpoints and serve as a full-stack application, these API endpoints will be used to further authenticate users on the frontend of the application. Nextjs also offers file-based

`

routing which defines pages and routes with files and folders instead of code, doing there will be less code written which also means less work and a highly understandable code-base.

2.6.1 Server-Side Rendering

Instead of rendering web pages in the browser, server-side rendering (SSR) renders them on a server and sends them to the browser (client-side). SSR delivers the client a completely rendered page; the client's JavaScript bundle then takes over and allows the SPA framework to function. If your application is server-side rendered, this means that the content is fetched from the server and sent to the browser to be shown to the user. Client-side rendering is distinct in that the user must travel to the page before the browser receives data from the server, implying that the user must wait a few seconds before the browser is served with the page's content. Server-side-rendered applications are applications that have SSR enabled.

`

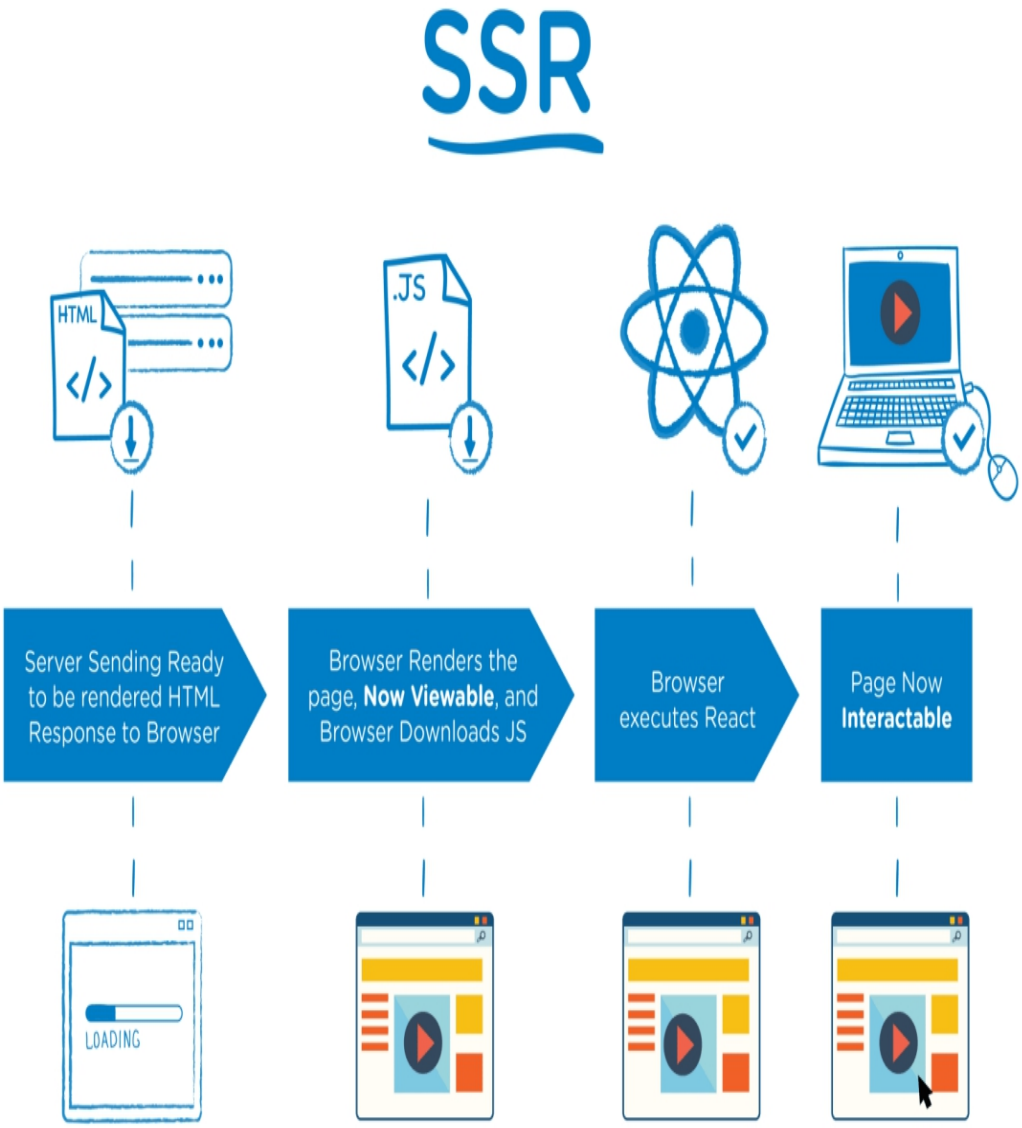

Fig 2.2: Server-side rendering (Grigoryan, 2018)

2.7 Related Works

In a clear and concise implementation of any blockchain cased voting system, security and trust are some of the elucidated requirements that the system needs to meet. In an article by (Ma, Zhou, Yang, & Liu, 2020) a system is proposed, the system uses a feedback mechanism and Wilson score ranking algorithm to implement a voting mechanism. The system uses the Wilson score ranking algorithm to modify the voter support rate of candidates and the feedback mechanism to suppress malicious votes.

`

In another work by (Marella, Milojkovic, Mohler, & Dagher, 2019), a system called Genvote was created and this system uses the standard proof of work mechanism for validating transactions. The system uses three smart contracts coded in Ethereum's Solidity language as well as two scripts written in javascript, and one HTML page. This project also assumes that users have the Metamask plugin or extension downloaded on their browser. The Genvote application allows the Administrator to deploy the application to the network, the voter who registers in the system with a valid student id or employee Id and e-mail address to vote on given ballot ID numbers, and lastly the creator, who is in charge of creating ballot permission.

The works above use the solidity language in defining the logic for the electoral processes, further; according to research by (Tso, Liu, & Hsiao, 2019), they created a system that allows voters and bidders to participate in the opening phase and improve participant anonymity, the privacy of data transmission and data reliability. They used a smart contract and privacy protection cryptography to produce a distributed electronic voting system. The proposed system has six roles; The voter (*Vi*) is voter I with voting eligibility, The Registration server (*RS*) is responsible for verifying voter identity and sending voting certificates, The Authentication server (*AS*) is responsible for verifying voter identity as well as sending ballot signature to legal voters, The Voting Website (*VWeb*) which ensures that the user can not cast more than one vote, The Record Center (*RC*); after the user cast the vote *VWeb* sends both the voting certificate and ballot signature to the *RC*, which then confirms that more than one vote has not been cast, The Smart Contract (*SC*); this is dynamic and enables voters to review their ballots and count the votes.

According to a thesis paper by (Ahlkvist et al., 2019), another system is created called DeVote, DeVote system design consists of two abstract components, which are the voterID generator that will provide eligible voters an anonymous ID, and the smart contract. The smart contract was modeled to meet the identified requirements in the system. DeVote application flows goes in this format; assume that there are two identified users or voters, One voter named Alice asked for a voter ID on DeVote and a trusted third party(TTP) generates an anonymous ID and saves it on the smart contract, Alice then receives the ID. Alice can then proceed to use the provided ID to place a vote on the system. The TTP checks if the given ID is eligible to vote and if the user is valid stores the vote on the smart contract, Alice then receives a confirmation that her vote was placed. Alice can always check that her vote was

`

placed. Bob who is another user can check if the vote is correct based on the information the smart contract provides.

Another project that uses a smart contract to implement a voting system is the project by (McCorry, Shahandashti, & Hao, 2017), they created a smart contract for boardroom voting with maximum voter privacy. The system they proposed is called the Open Vote Network, it provides maximum voter privacy as an individual vote can only be revealed by a full coalition attack that involves compromising all the other voters in the system. Also in another project proposed by (Khan, Arshad, & Khan, 2018), they base their e-voting approach on (Ryan, 2008), the system uses fingerprinting to protect against double voting. The system also sends a cryptographic hash of the transaction (ID) to the user to serve as proof that the vote has been cast.

In research from Hjalmarsson, Hreioarsson, Hamdaqa, and Hjalmtysson (2018), they developed a system that uses an external service system to generate user id, the extra voting service uses Nexus software and RFID scanners to generate and scan the id, upon registration, users will choose a pin consisting of 6 numbers. For identification, the user would have to identify themselves by scanning the ID and the chosen PINs in other to be authenticated to use the system to vote. The system they developed is on a permissioned blockchain that uses a proof-of-authority consensus algorithm.

`

CHAPTER THREE

METHODOLOGY

This section defines the systems development approach, and the stages the project passed through during development. The system was designed utilizing a reuse-oriented software development process model. To elicit requirements, various researches were carried out and a system prototype was created to identify faults in a centralized voting system, using these faults a new requirement specification was outlined and the requirements were validated according to the test cases outlined for the new system model.

3.1 System Development Process

The requirement in this system is the statement in domain-specific terms which specifies the verifiable constraint on the implementation that it should meet. Some of the requirements were gotten from developing a prototype system that uses a centralized database, with the flaws in this system highlighted and improvements made, a new system was developed using requirements gathered. The stages of development that the system would go through are:

1. Requirements Definition and Engineering.
2. Software and System Design.
3. Implementation and Unit testing.
4. Integration and System Testing.
5. Operation and Further Maintenance.

3.2 Requirements Definition and Engineering

The requirement engineering phase of this project was the phase where the system was evaluated, and the services that the proposed system requires, together with the various constraints that define the system's operation and development was defined in a detailed form. The proposed system that was developed in this project is called Smart~Vote, and this name would be used to refer to the project through the entire parts of this document. In this section, the User and System requirements were elucidated and demonstrated using the use case diagrams, and the functional and non-functional requirements were listed. Use case diagrams would be used as

`

requirement discovery techniques to further explain the interactions between entities in the system.

3.2.1 User Requirements

The user requirement of this project states the services the Smart~Vote system provides to system users and the constraints in which it operates. Here are the user requirements for the Smart~Vote system:

1. Smart~Vote should allow users to cast votes on various election instances.
2. Smart~Vote should allow only authenticated and authorized users to cast votes on authorized electoral instances.

The above requirements state what the system is required to offer to its users. In the listed requirements above, users should be able to vote for their preferred candidates. Then some constraints restrict or affect the use of the system. In this project scenario, only the users that are authenticated should be able to use the system to vote, and even if the user is authenticated, the user has to be given authorization by the officials of the elections in other to be able to vote in that election instance. This stage is the direct replica of the manual systems used, where you might be given a voter's id but if you are not from a particular state, you cannot take part in that state's governorship elections.

3.2.2 System Requirement Specifications

In this section of the project, the requirements of the system were deconstructed into a more detailed description, the detailed description covers the system’s functions, services, and operational constraints. The following are the system requirements for Smart~Vote System

1. Factory Elections should create multiple instances of other elections.
2. Election Instances should allow users to vote for their preferred candidates.
3. Users should be able to register on the system.
4. If election instances are still made available, authorized users should vote.
5. Election instances should not allow users to vote if the instance has been ended or closed.

As specified above, there should be a factory instance of the smart contract being created. The use of a factory contract is to create multiple instances of an electoral

`

process, this being the case that the one who creates the instance of this contract becomes a chairman or the official, and this office has several responsibilities. In each case of a deployed election instance, at least two candidates must be initialized with such instance in other to allow users to choose from the options made available to them. The system has a registration process that every user must follow in other to be authenticated to use the system. Election instances can be opened and closed based on the specified time limit by the creator of such instances. Basically, whoever creates the election instance will be the only person that can open an election and close the election, also the creator of an election instance is the one person that can authorize users to vote in that particular instance.

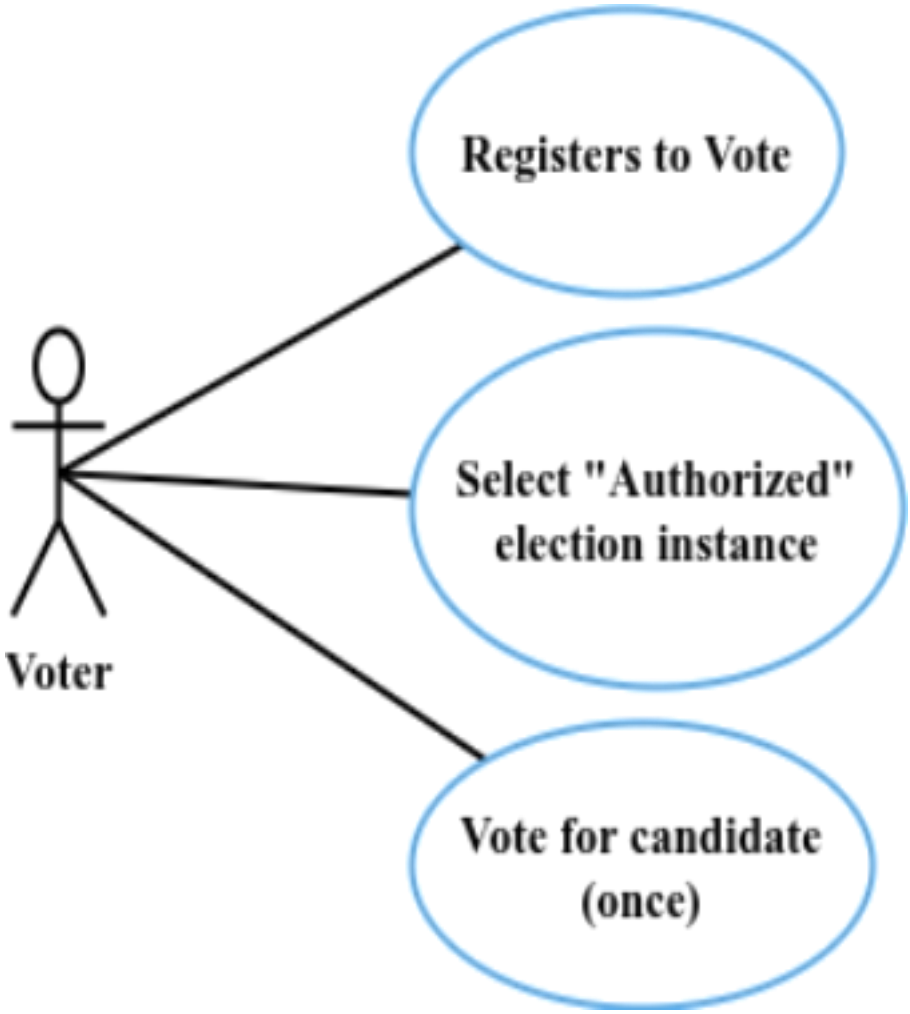

Fig 3.1: Voter Use Case Diagram

`

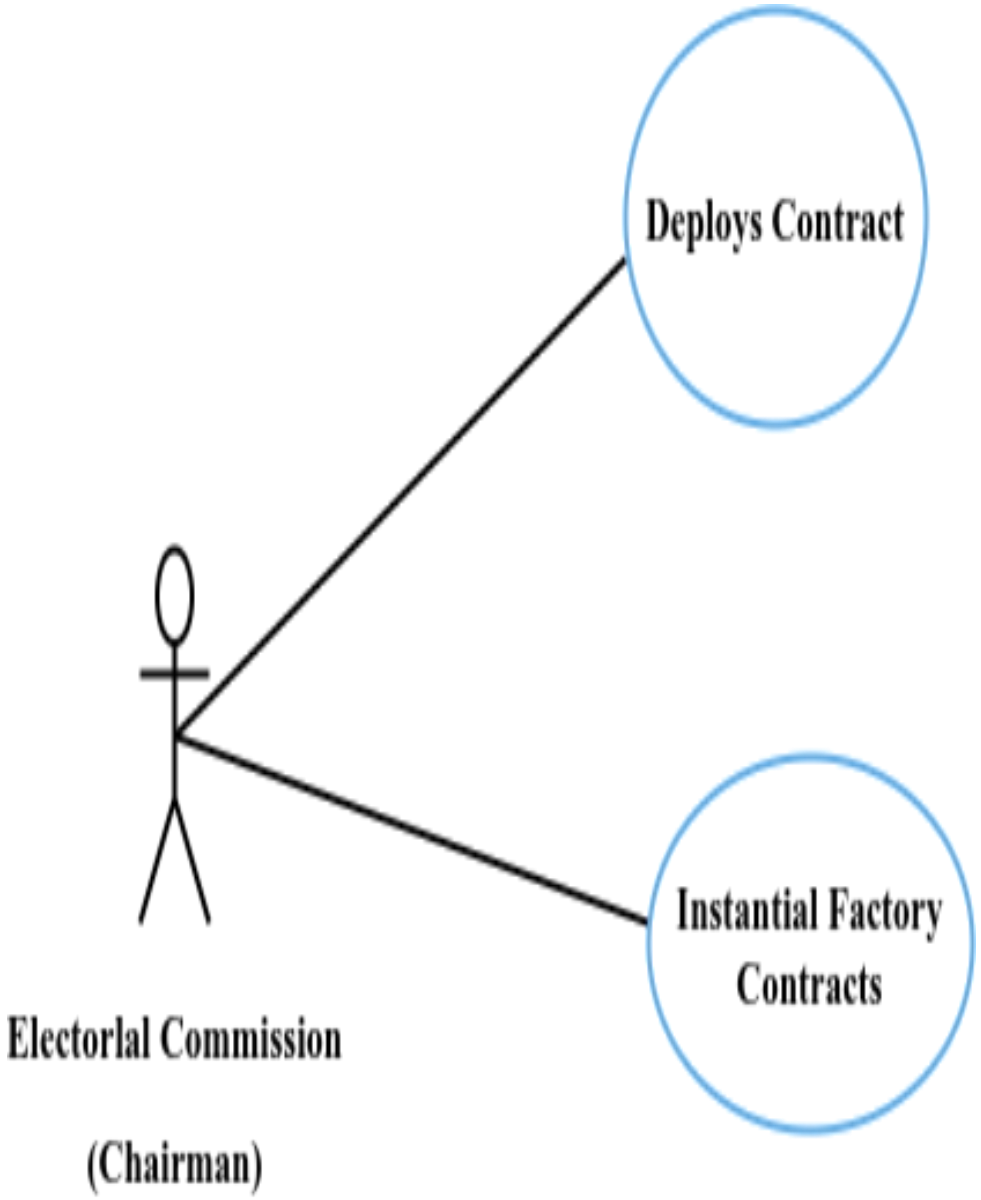

Fig 3.2: Admin Use Case Factory Contract Creation diagram

`

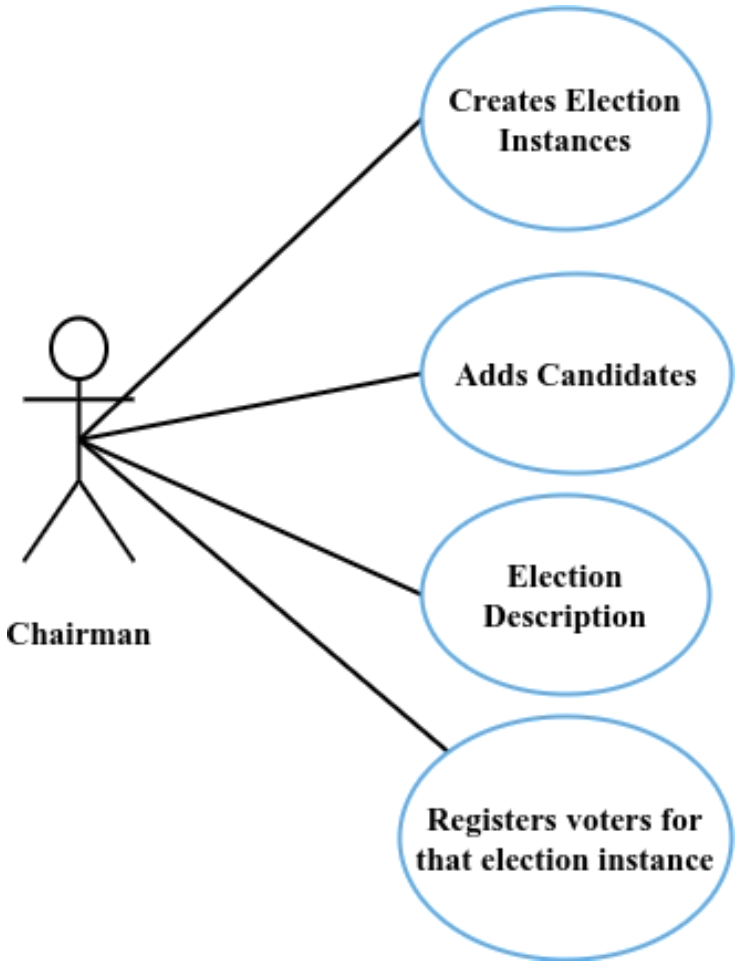

Fig 3.3: Use case diagram for chairman (creator) for the election instance

3.2.3 Functional Requirements

The Functional requirements of Smart~Vote are statements of the services that the system provides, how the system reacts to particular inputs, and how the system behaves in particular situations. The following are the functional requirements of the Smart~Vote system :

1. Voters can select between authorized election instances to vote in.
2. Voters can choose from a list of options of available candidates to vote for.
3. Each election instance should contain certain unique variables and functions that specify each electoral flow.
4. Each user should have a system-generated voter Id after registration has been made successfully.

`

3.2.4 Non-Functional Requirements

The non-functional requirements on the smart~vote system are the constraints on the system that affects how the system behaves and operates according to the input parameters, in this case, the non-functional requirements specify systems speed, size, ease of use, portability, and reliability, these are constraints on the system functionality. All these are not directly concerned with the specific services delivered by the system, several other non-functional requirements impact the behavior of the system indirectly such as availability and security. Security greatly impacts the flow of processes in the system. The following are the list of security constraints on several aspects of the system (For users - Voters):

1. Users must be registered on the systems server.
2. Users must have a system-generated voter Id.
3. Users must have one unique ethereum address when registering.

List of security constraints on several aspects of the system (For admins - Election instance creator):

1. Admins must be registered as “admin”.
2. Admins should also have a system-generated Id.
3. Admins are not allowed to vote as admins.

In terms of speed and usability, the system proposes the use of a server-side rendering client-side application in the development of the application, this would enable efficient human-computer interaction, in terms of availability the system should be hosted online for anyone to access and then the smart contract would be deployed on the general ethereum network.

3.3 Intermediate Development Stages

The components used in this stage were the required components that properly integrate with the system, the different components of this system are the ethereum network and for this test case, the system uses a local ganache server to host smart contracts and test accounts that have test balance. The second component is the web page that interacts with the ethereum smart contract. Here the use of nextjs to create the web page was utilized and the web3 library was used to perform communication between the two components. Another component used is the metamask extension

`

which plays a major role in the transactions made between web3 and library and the ethereum network.

3.4 System and Software Design

In this section, the design process for the software is identified by establishing an overall system architecture and identifying the fundamental software system abstractions and their relationships. The general system modeling stages would be followed and various diagrams would be used to illustrates various sections of the application. These designs would be generated to meet up with the requirements that have been identified for the system, the models that would be created will also conform with the use case diagrams used in the requirement elicitation process. The general design model process is in three phases, namely, the design inputs, the design activities, and then the design outputs. In the design input stage here we take the elucidated requirements, both the functional and non-functional ones, and we also take the user and the system requirements as parameters that aids in designing the model of the system. Next, we to try to design the architectural design of the system, the systems proposed interface as well as the description of several and important data needed in this activity stage. The architectural design and the data description all depend on the design of the smart contract. The different functions that the smart contract provides are what the system architecture model would present and the data descriptions also depend on the design of the smart contract.

`

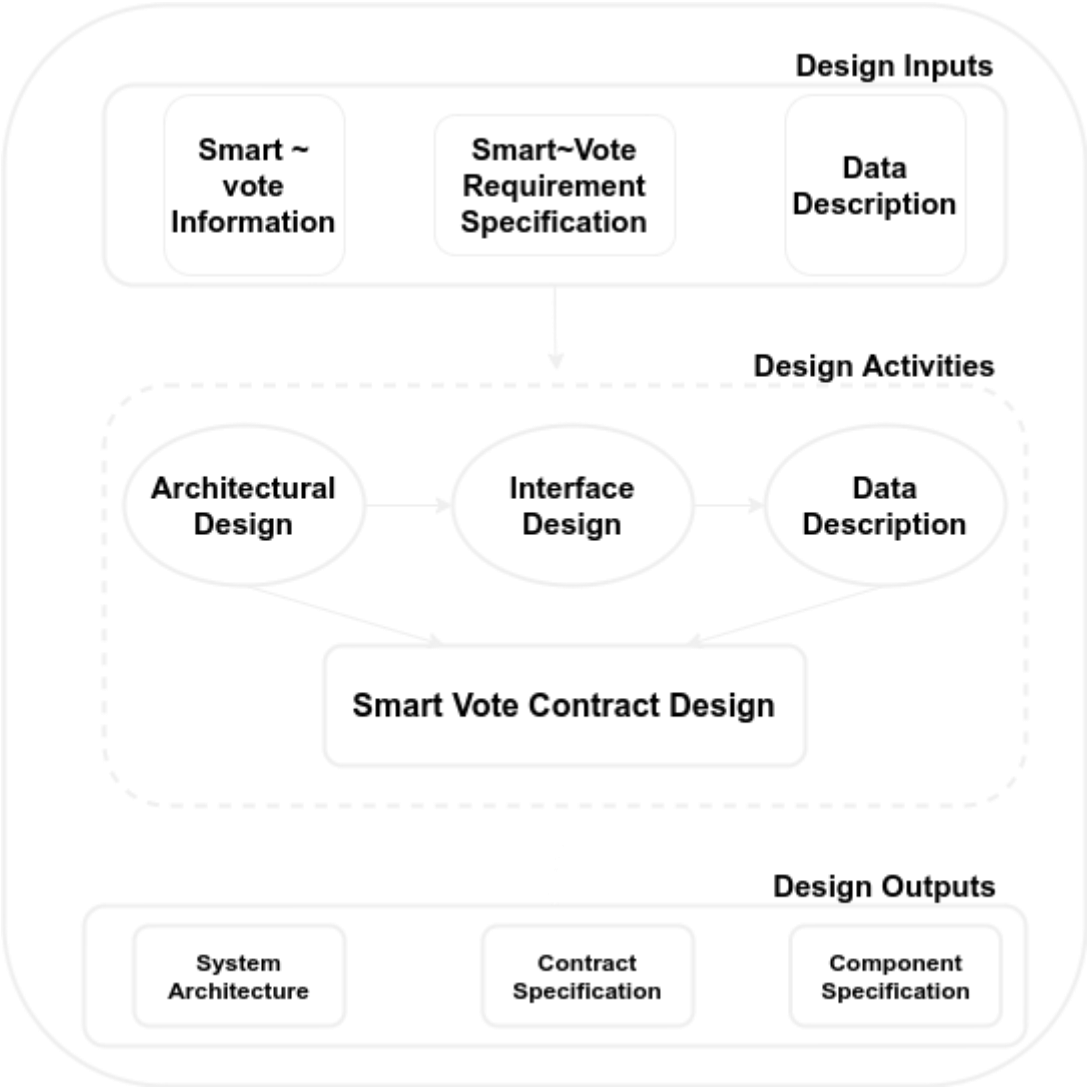

Fig 3.4: Smart~Vote General design process model

The design outputs from the activities taken would be the entire system architecture, the contract specifications after its design and modeling as well as the different components that interact with the system to enable the full working functionality.

3.4.1 System Architecture

The overall system architecture further divides the system into three sections, the smart contract design, the web or internet section, and the web applications.

`

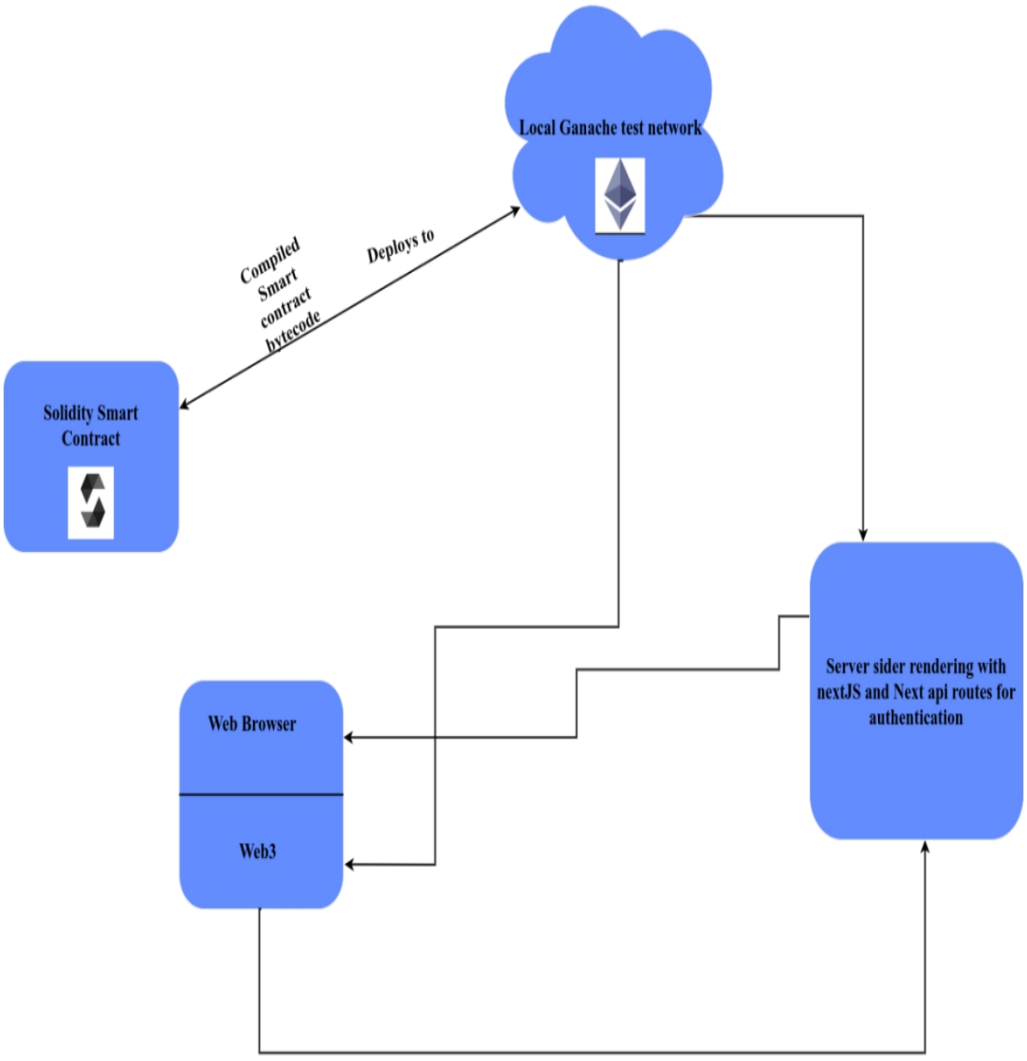

Fig 3.5: Smart~Vote Architectural Design

The overall system architecture further divides the system into three sections, the smart contract design, the web or internet section, and the web applications.

a) Smart Contract structure

The smart contract of this architecture is also divided into two, the factory contract which makes multiple instances of the election contract, and the election contract itself. The factory contract has certain variables and functions, that define how it interacts with the election contract model. An array of type ethereum address saves addresses of all the contracts deployed by the factory contract, and two special functions, namely the create election function that creates a new election instance and the getDeployedElections function which returns the deployedElections variable.

`

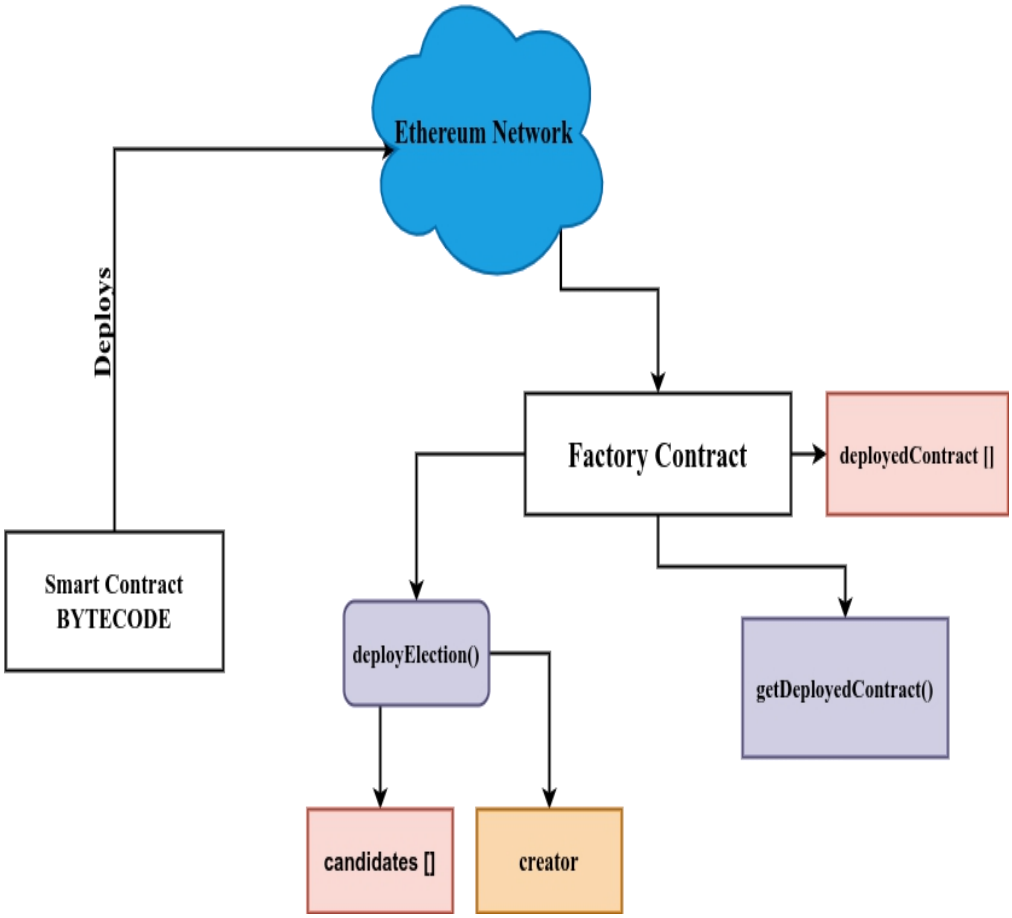

Fig 3.6: Factory Contract Architecture.

The compiled bytecode is deployed on the ethereum network and the contract's abi is used to create the multiple instances of the election.

`

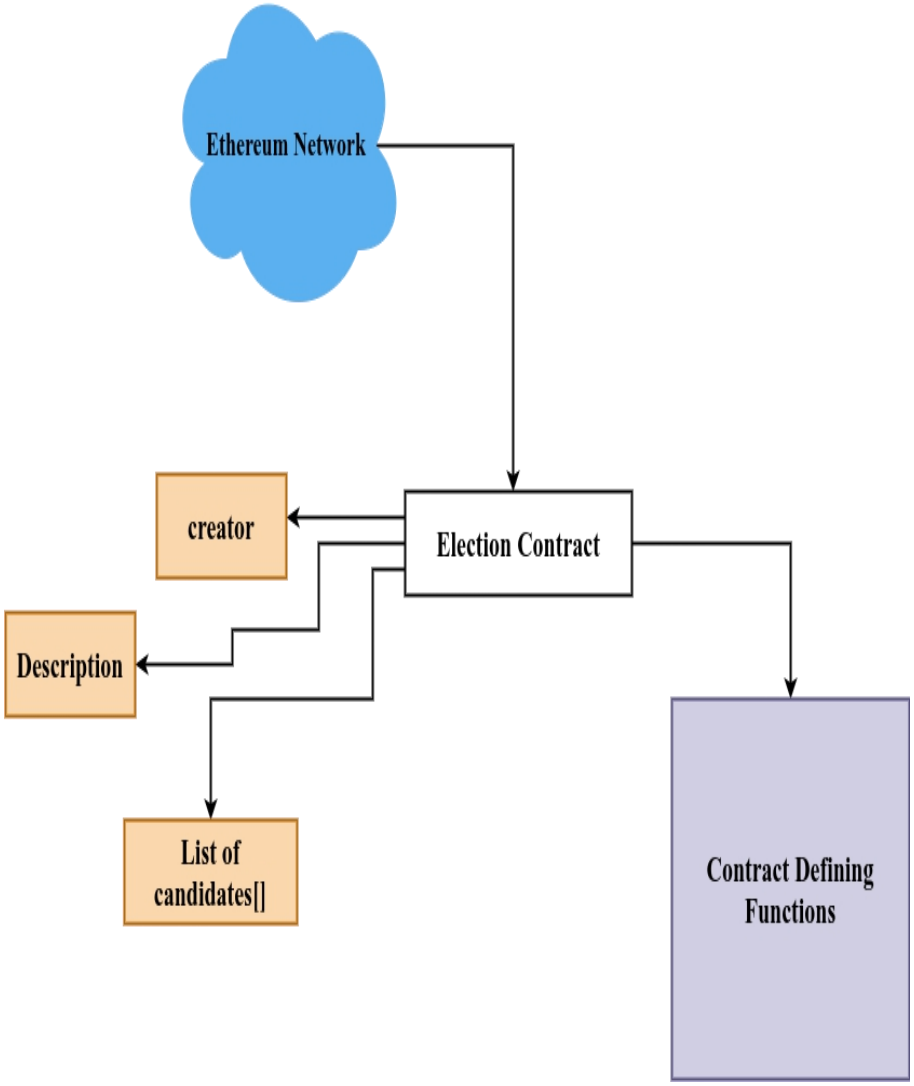

Fig 3.7: Election Contract Architecture

This architecture displays how the abi is used to create an election instance, clearly showing the parameters and the important data needed to initialize the election on the blockchain.

3.4.2 Sequence Diagram

The registration sequence for the voter requires some sort of authentication already existing on the system. In this work, it is assumed that every user has a nin used in the authentication. The nin is a 10 digit number sequence that uniquely identifies a user, and validates the user as an authorized citizen, eligible to vote in an election.

`

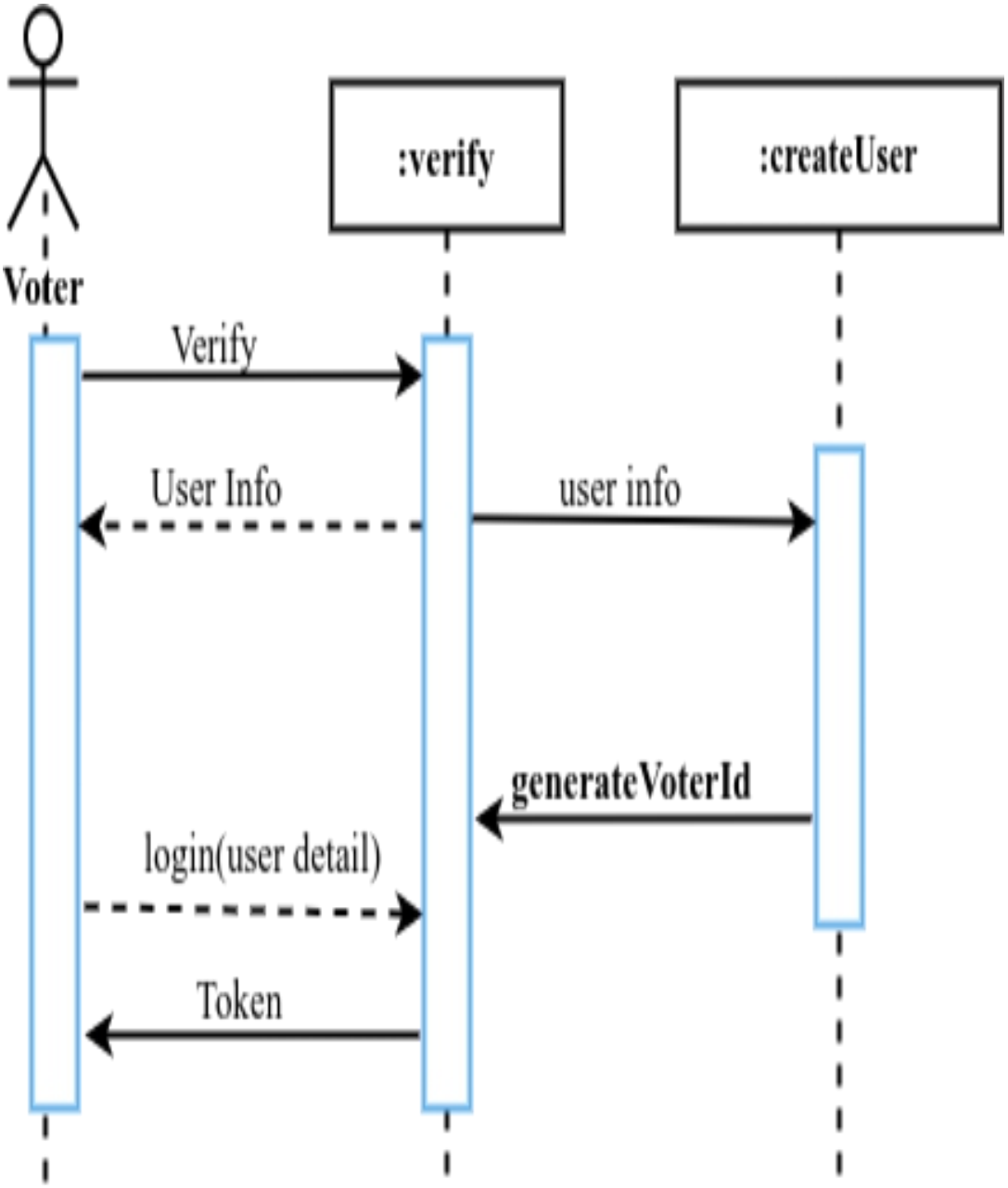

Fig 3.8: Voter Registration Sequence diagram

The admin sequence of flow concerning the creation of an election instance requires the admin to input the title of the candidate and the names of the candidates available in that election. In the sequence flow the admin inputs the important data concerning the compiled smart contract's abi.

`

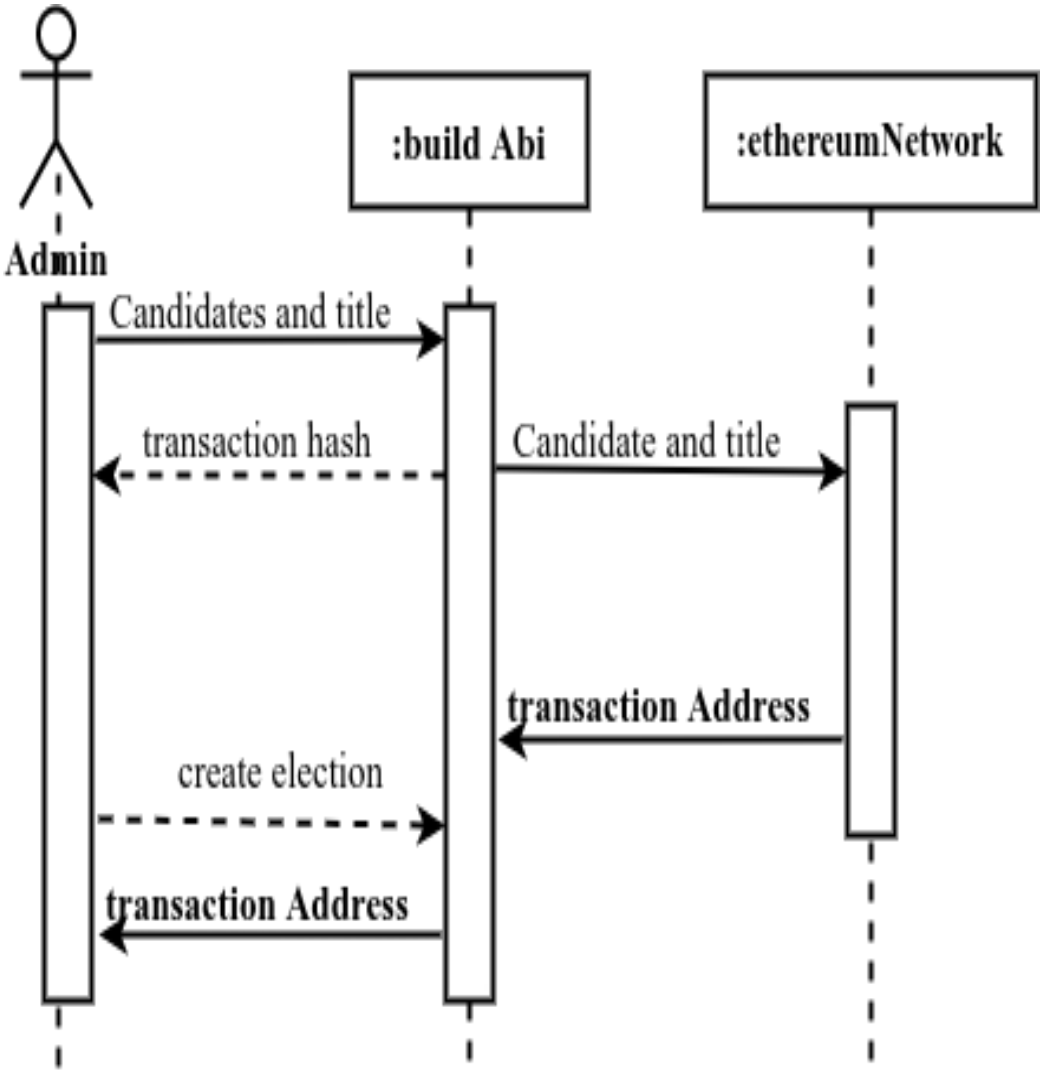

Fig 3.9: Admin election creation Sequence Diagram

The admin in the flow diagram performs some actions and interacts with components of the architecture, the admin keeps a copy of the authentication schema and uses this as a mechanism to properly identify the user. In this project we use a nin schema to represent identified users in the country, other test cases can be the matriculation number of students for an election held in a school. This project uses a nin schema to represent identified users in the country; other test cases include scenarios in a university setting where the student matriculation number can be used as a mode of identification.

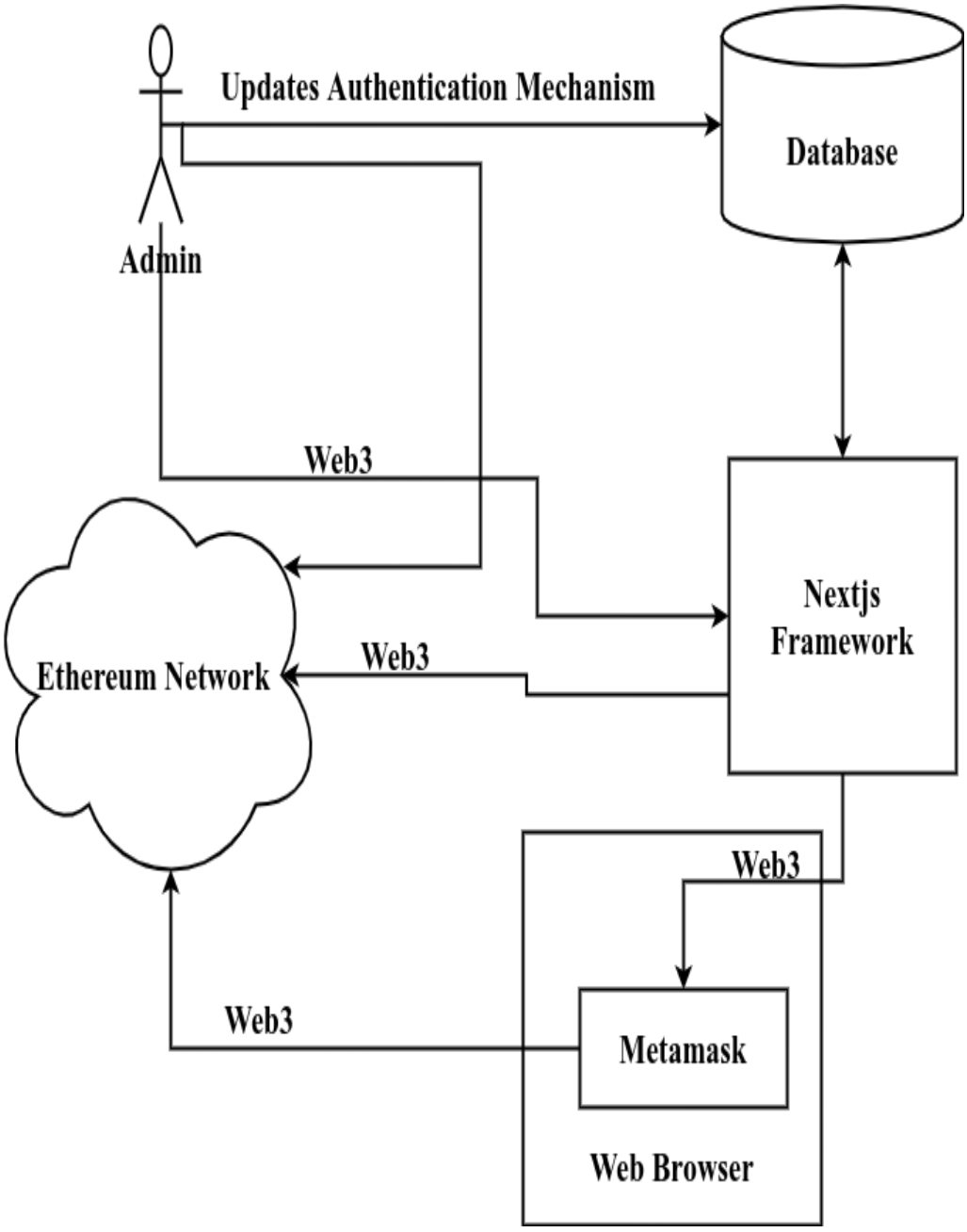

Fig 3.10: Admin Flow Diagram

`

3.4.3 Data Flow Diagram

This section depicts the data flow from the blockchain communicating with the web3 library to the nextjs framework's server-side rendering, as well as nextjs interacting with the database receiving data on the client-side.

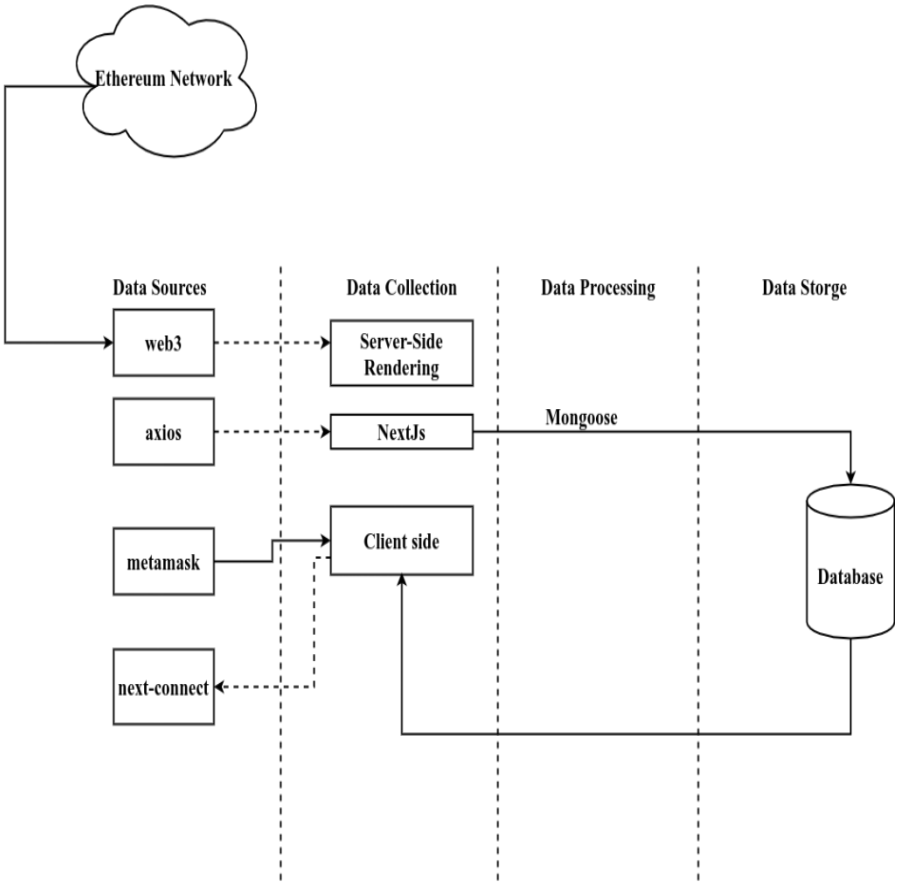

Fig 3.11: Data flow diagram

3.4.4 Flow Chart Diagram

The flowchart diagram depicts the sequential steps of the system process using the two system actors. The system takes input from the voter and the admin, representing the potential steps that each actor could take, and the end and final step of each process.

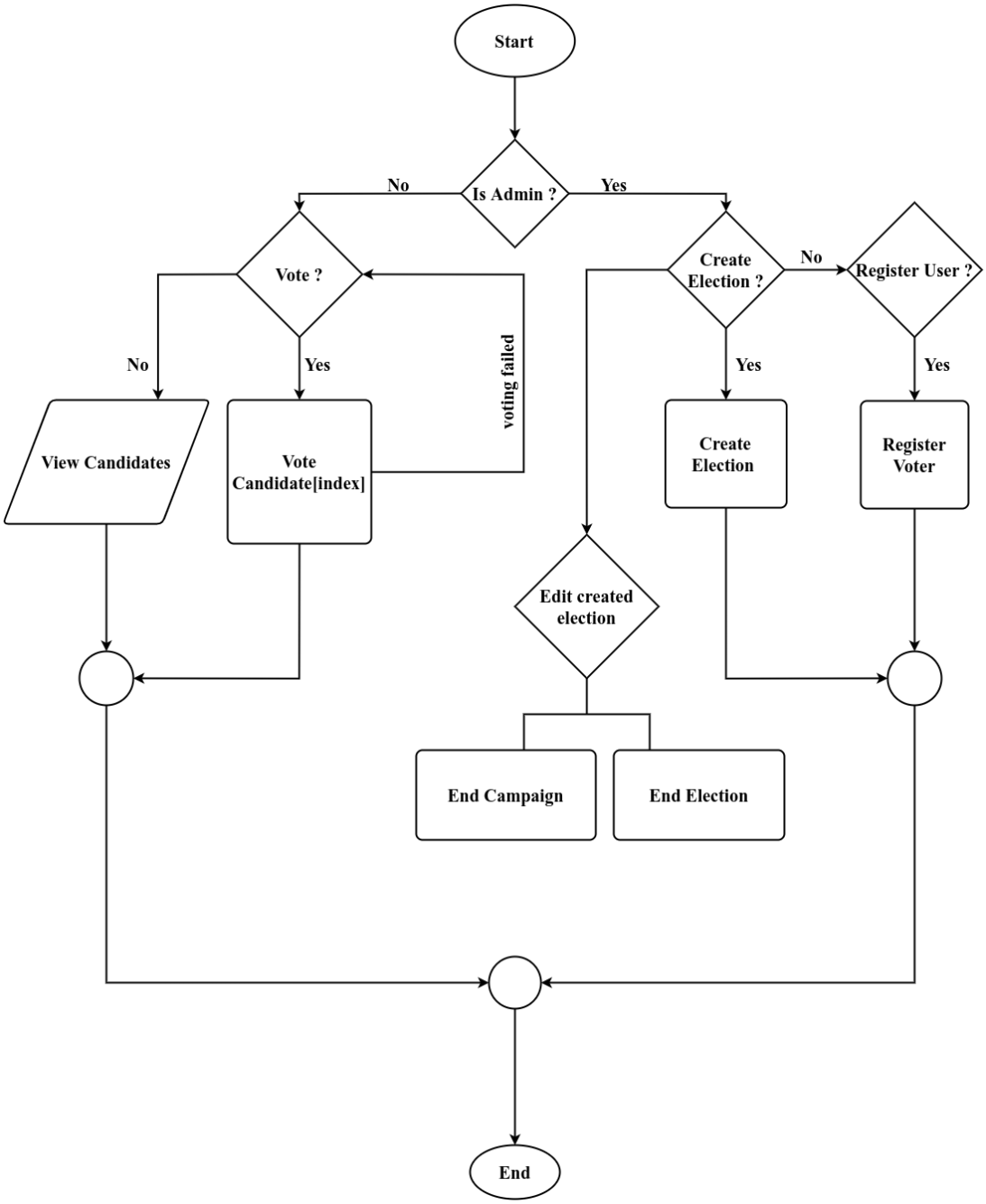

Fig 3.12: Flow Chart Diagram

`

CHAPTER FOUR

IMPLEMENTATION AND RESULT

In this chapter of the project, the system's implementation and testing were performed using the designs and the elucidated requirements. When implementing the project, a ganache local server was used as well as the test accounts and private keys to enable connection with the metamask extension on the browser. The system was developed using solidity programming language, javascript programming language, Html and CSS scripting, and styling languages. The system was taken through series of test cases, The remix online ide was used to develop and test the smart contract, and the use of javascript testing library mocha in defining more of the functionality and functions in the contract.

4.1 Software and Hardware Requirements

In other to run this application, computer systems would need the following applications and software installed on their system.

System operating system:	Windows 10, Linux distros, Mac OS
Web browser:	Mozilla-Firefox(recommended), Google-Chrome, Safari
Extensions (WEB):	Metamask

4.2 System Development

The system was developed on a Linux operating system, following the development process that was reuse-oriented, the use of other components aided the production of the project. The very first stage of the systems deployment process is the contract compilation process, the smart contract programming language solidity has a compiler that compiles the smart contract, produces a build that gets the abi and bytecode. This bytecode is then deployed to an ethereum network and the address that the contract was deployed to is used with the abi to interact with the contract on the network. This is where the web3 package library aids in the communication process between the front-end, server-side and ethereum network.

The list of tools used when developing this system are :

1. Mongo Compass: for monitoring local database

`

2. Visual Studio Code: Text Editor
3. Postman: for making requests
4. Ganache: local ethereum server: port number 8545

4.3 Application Images

The images for the application, showing the different pages in the software, for both actors in the system.

4.3.1 System Home Screen

This is the first page that both the voter and admin of the system will see. It is the welcome page of the system.

`

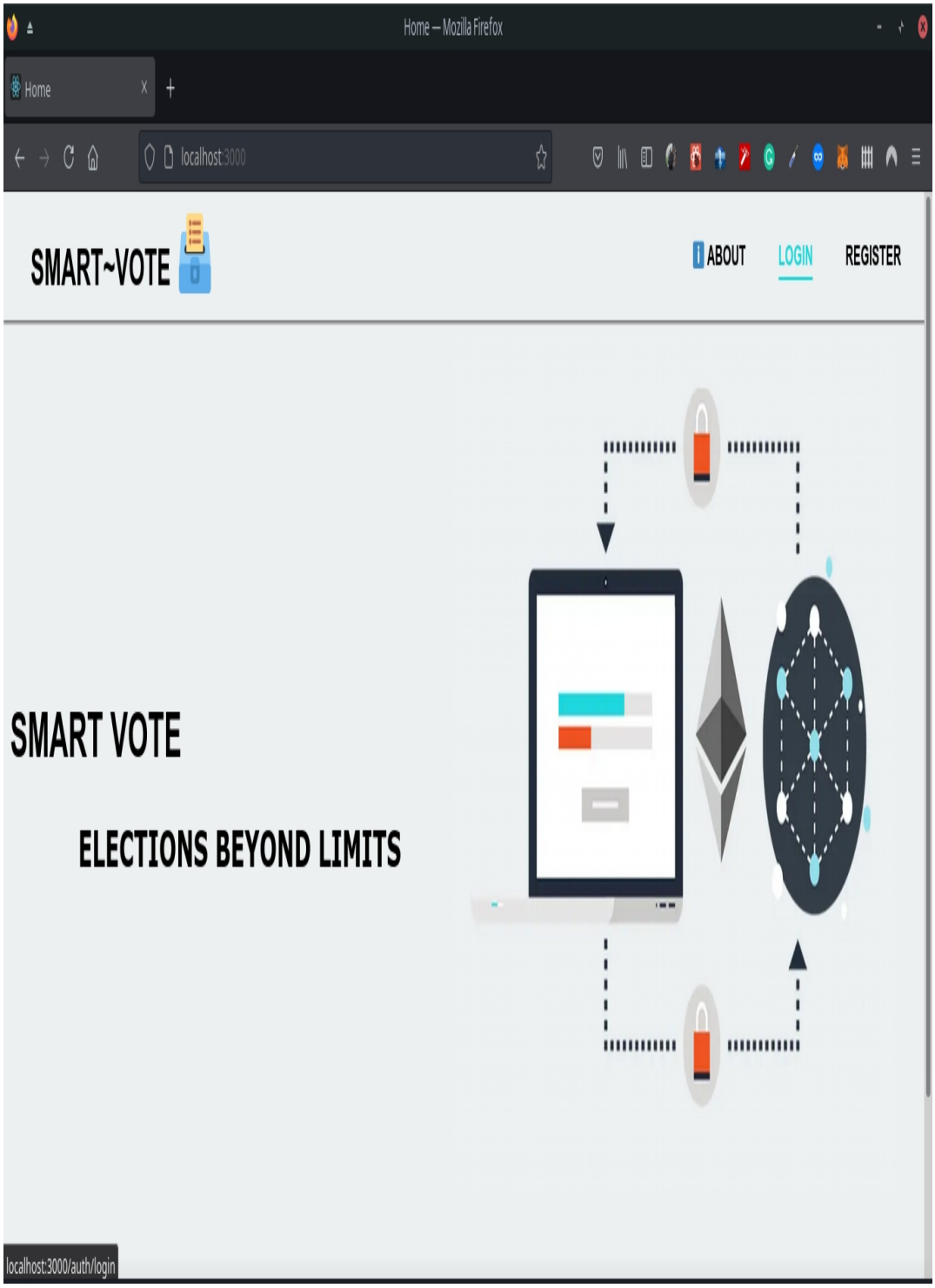

Fig 4.1: System Home Screen

`

4.3.2 System Login Screen

On this page, if the users already has an account on the system, they would be required to login with their nin and password.

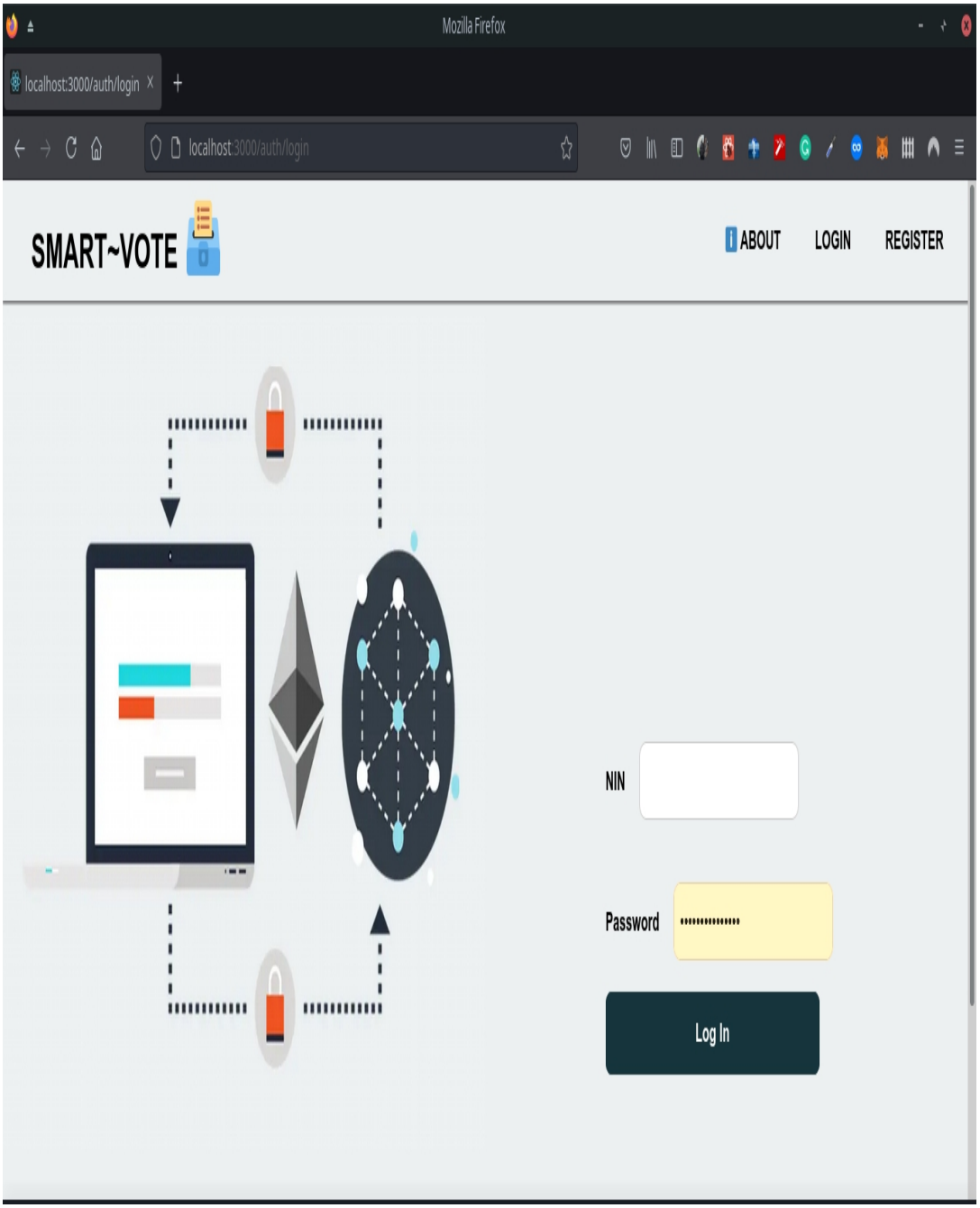

Fig 4.2: Login Screen

`

4.3.3 System Register Screen

The system's register screen takes in the users' nin and checks for verification, if it already exists on the system, it returns the user's meta-data and asks for a password that enables a user to log in. For the admin, he or she only has to log in because system administrators have already been registered the admin from the onset when the system was created.

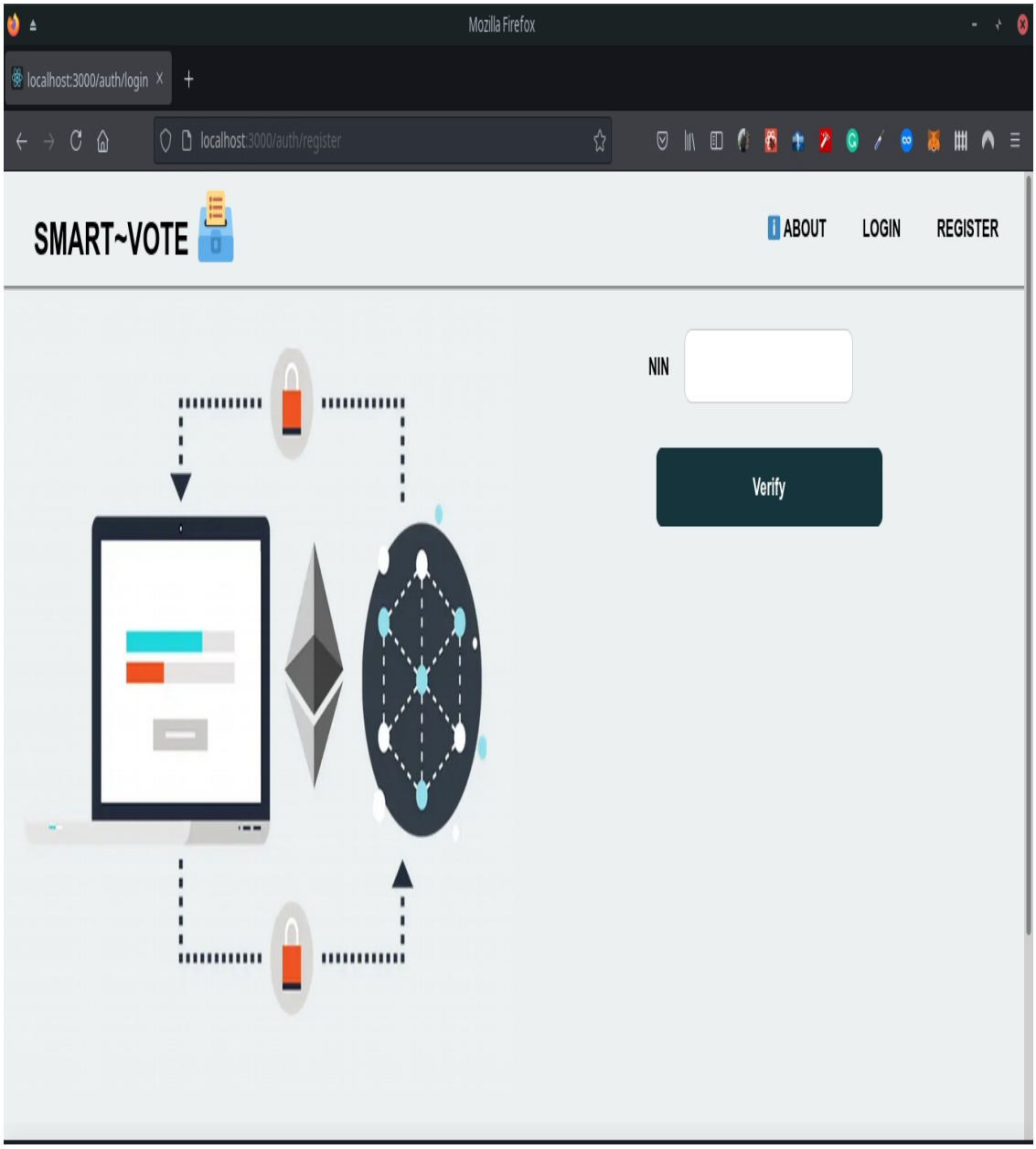

Fig 4.3: Register Screen

`

4.3.4 System Dashboard

The dashboard is divided into two based on the actors on the system, the system allows the voters to access a different dashboard from the voter.

a) Client Dashboard

The client dashboard shows the meta-data of the user and also shows available elections in the system with their ethereum address.

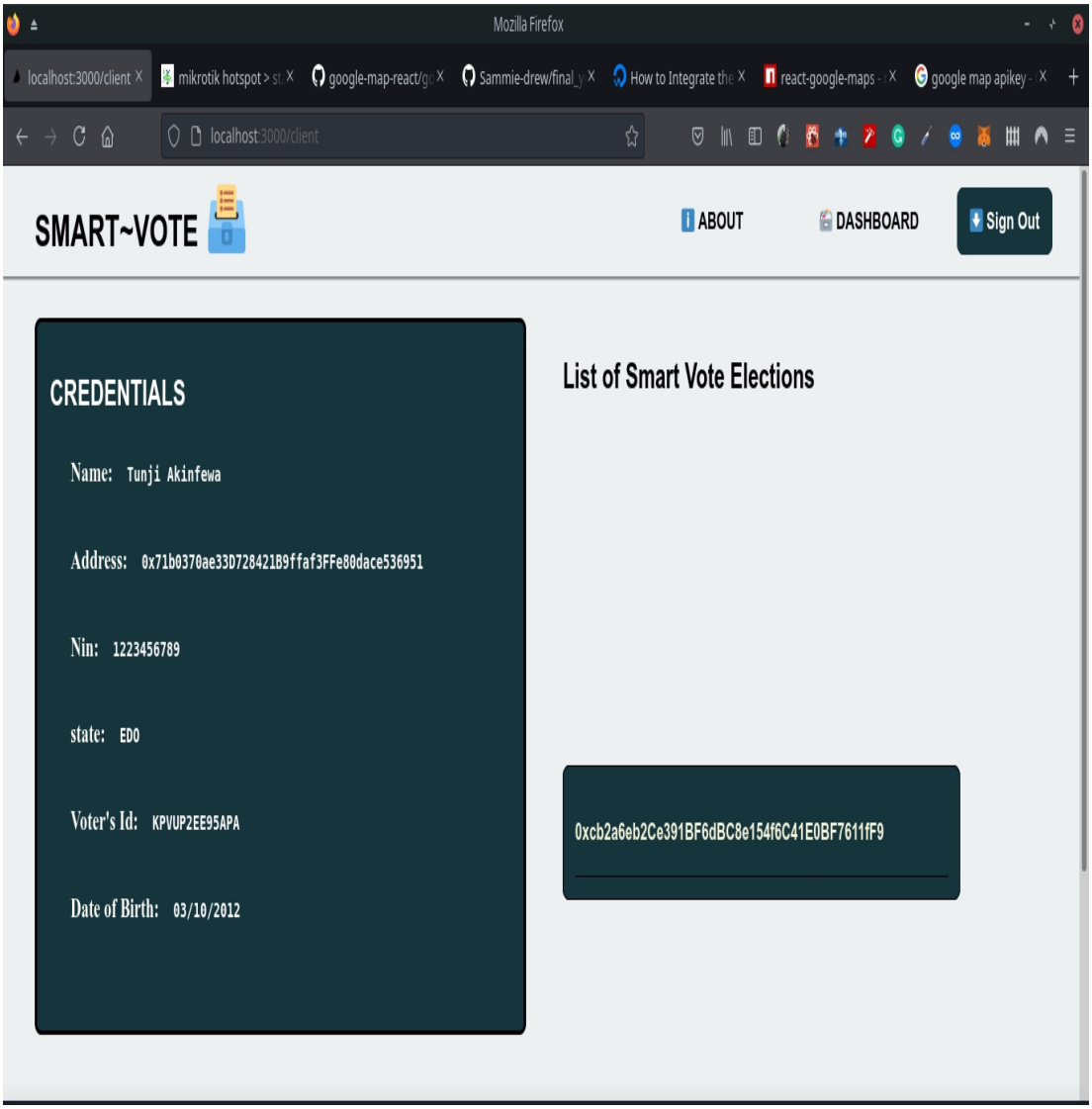

Fig 4.4: Client Dashboard

`

b) Admin Dashboard

The Admins Dashboard is more complex than the voter's dashboard, on the admin dashboard the first component rendered is the users/voters component. This component displays the list of all the users registered in the system. The second component is the create election component which allows the admin to create instances of the election using the factory contract and the compiled contract abi. The last component for the admin is the election data component which shows a list of all the elections in the system. The factory contract also routes to another page in the application. In this re-routed page, the admin performs three functions, Adds users to a particular election, End campaign for that election, and then ends the election once the time has elapsed.

`

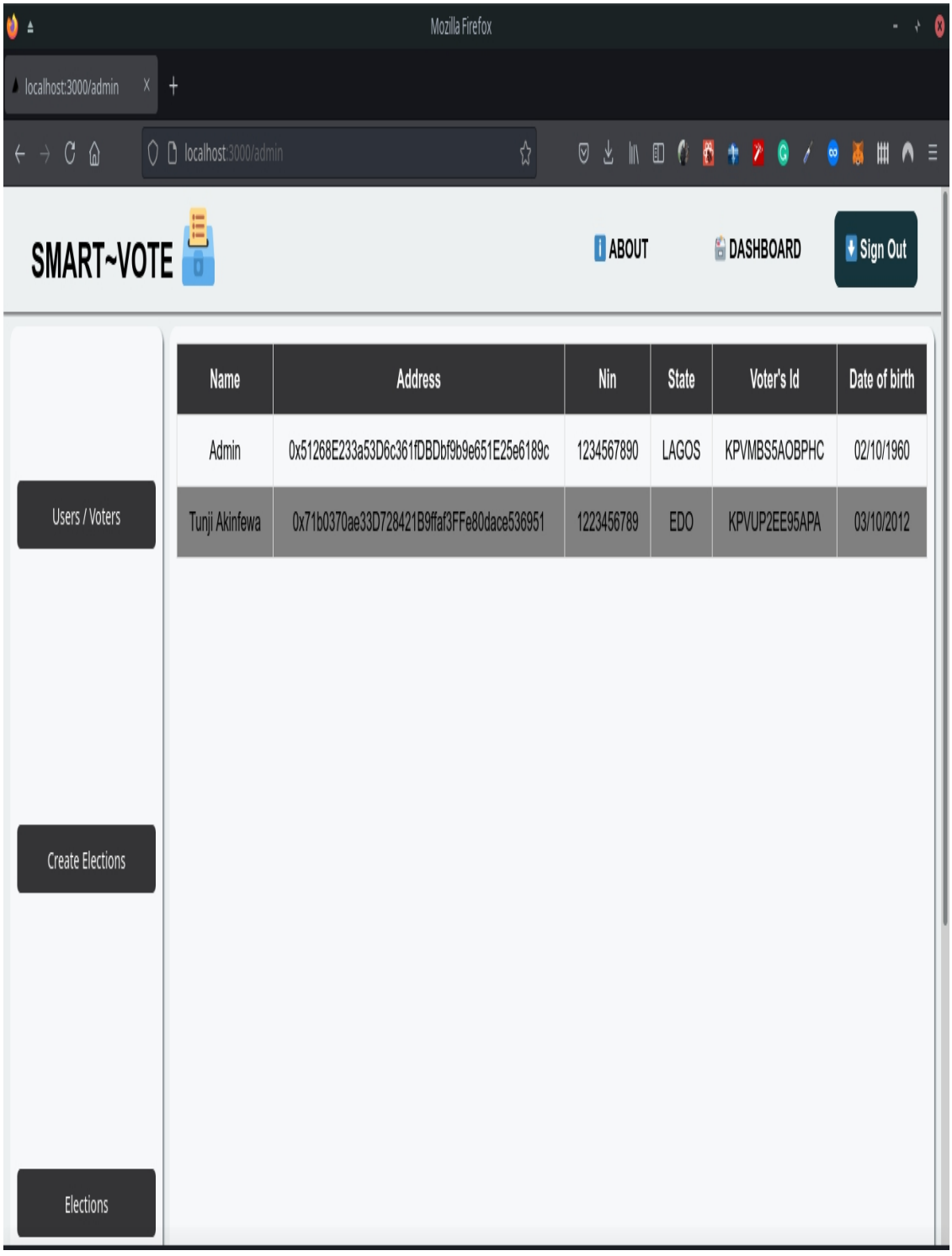

Fig 4.5: Admin First Component

`

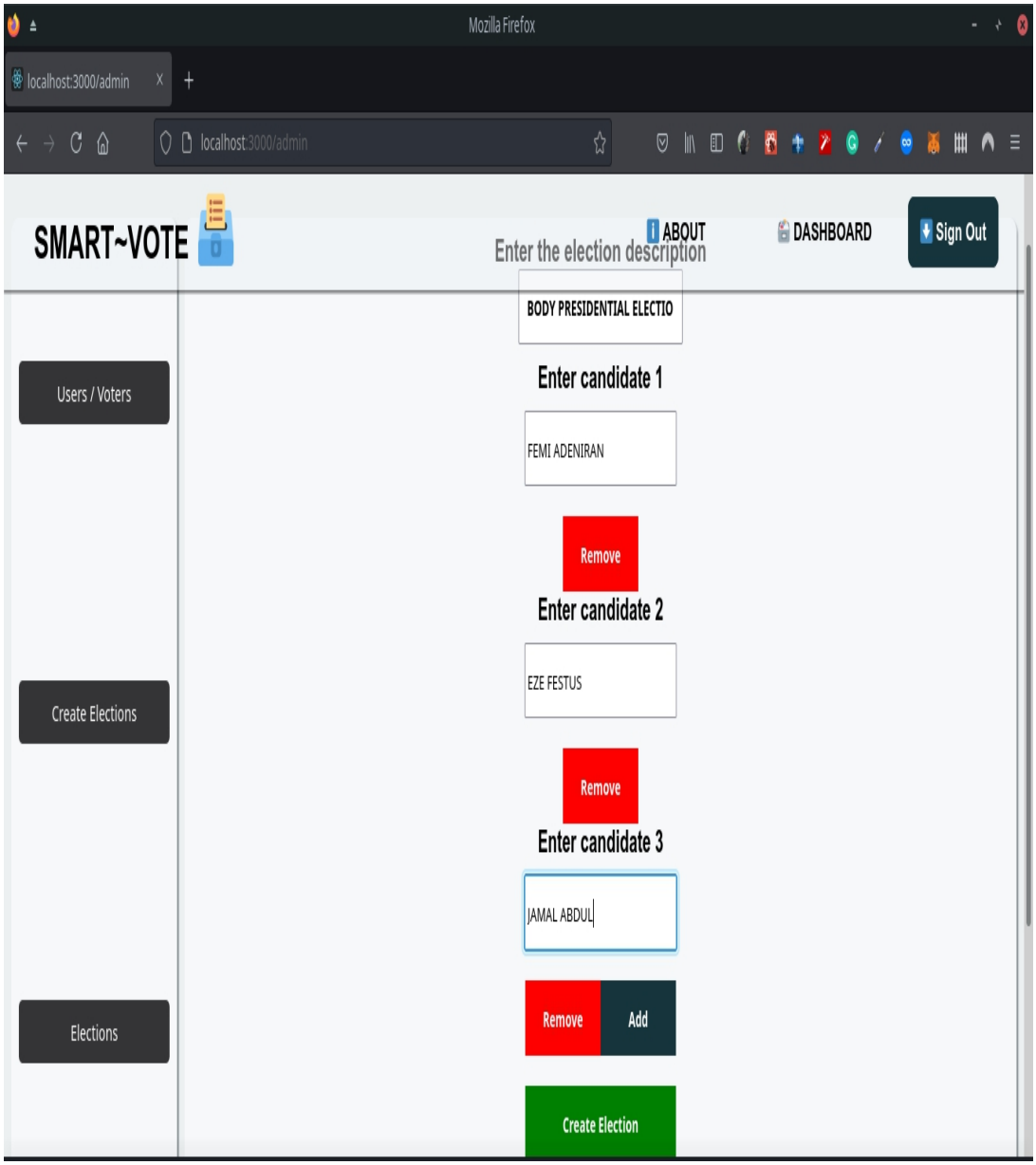

Fig 4.6: Admin Second Component

`

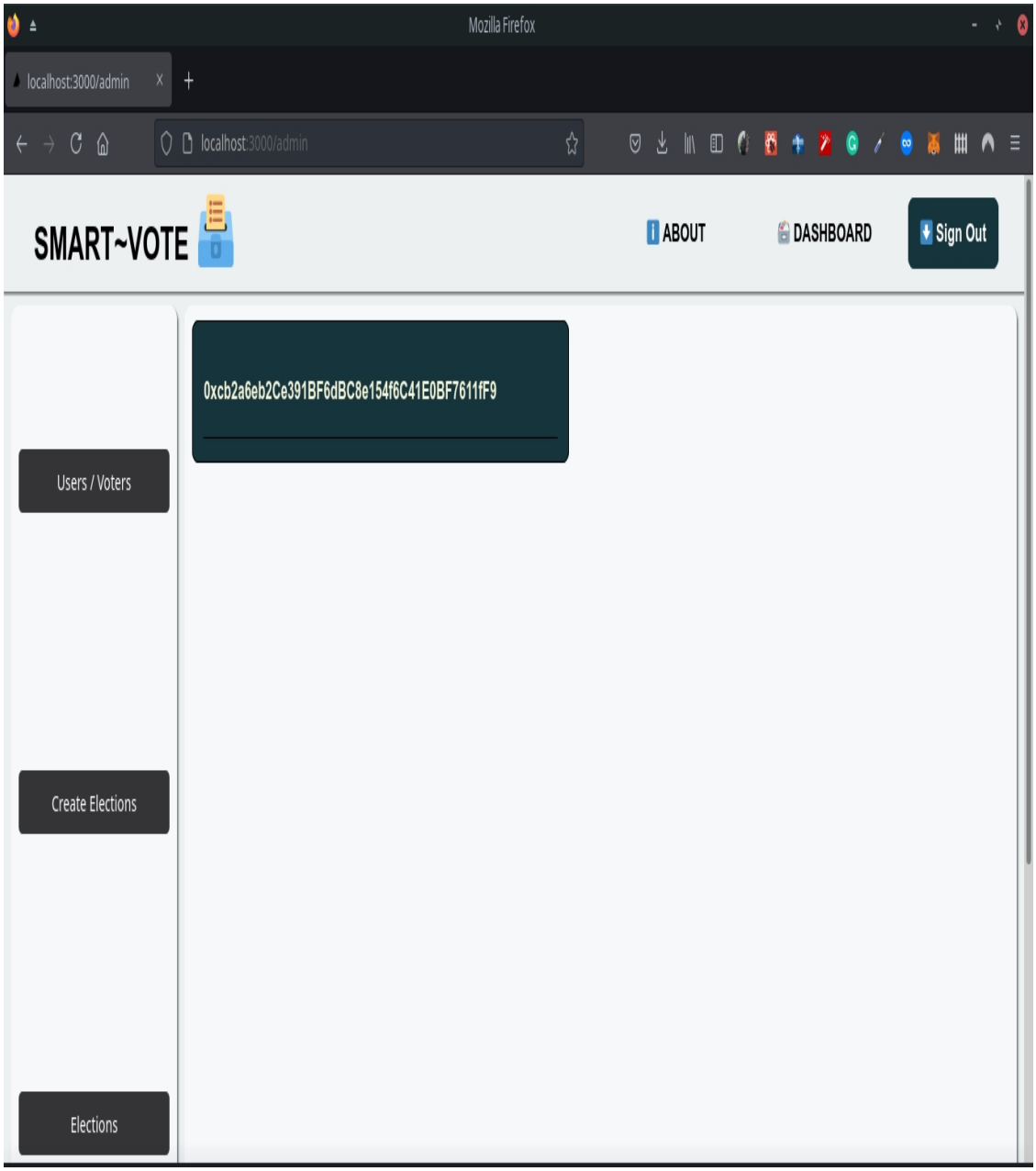

Fig 4.7: Admin Third Component

`

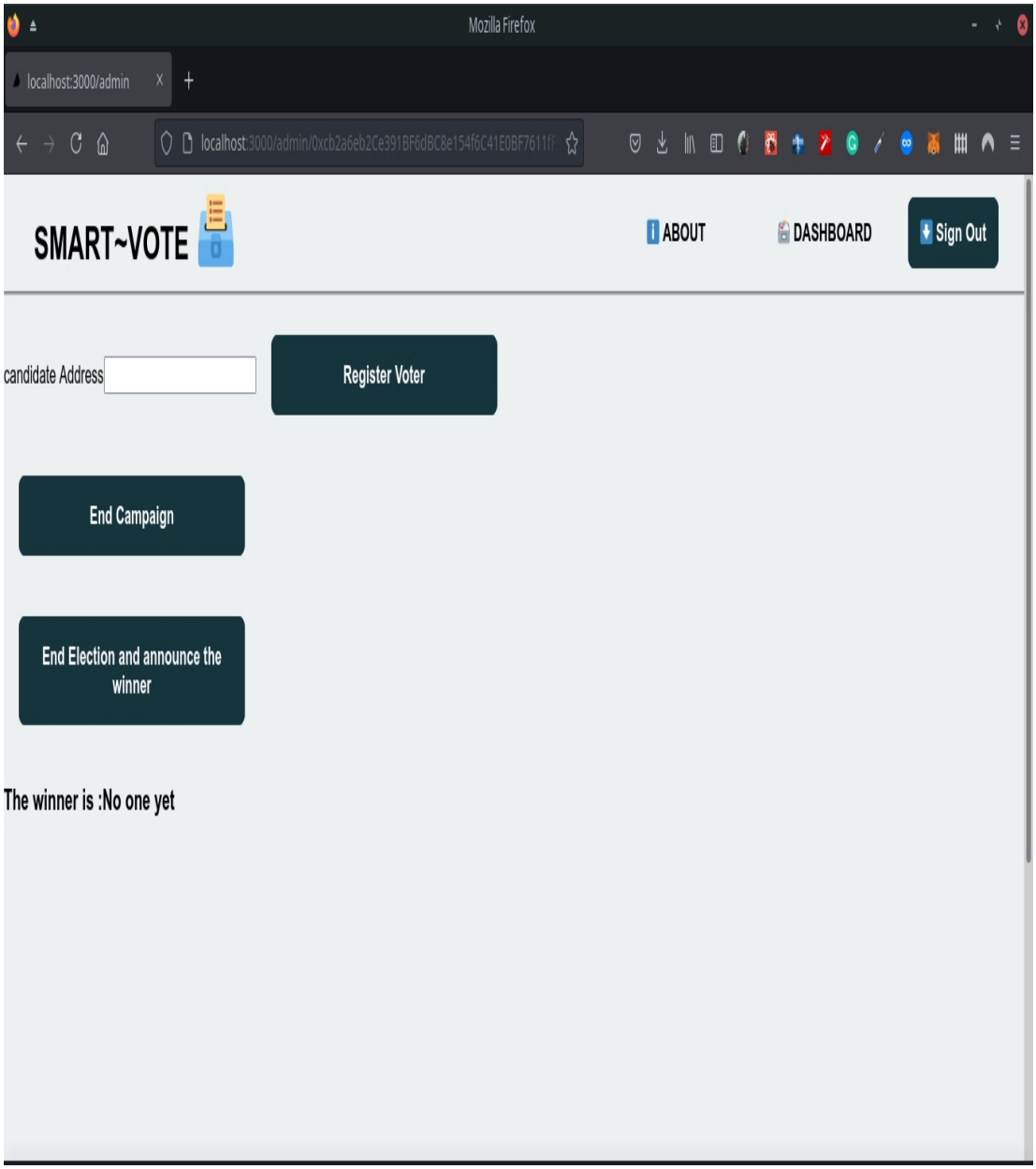

Fig 4.8: Admin Re-Routed Screen From the third Component

`

4.3.6 Vote Screen

If the user clicks on an election instance address, the system routes to the voting setup screen, this setup screen shows the information of the election clicked, the number of candidates, and the creator of the election. On the right side of the application is the vote button that also re-routes to a different screen.

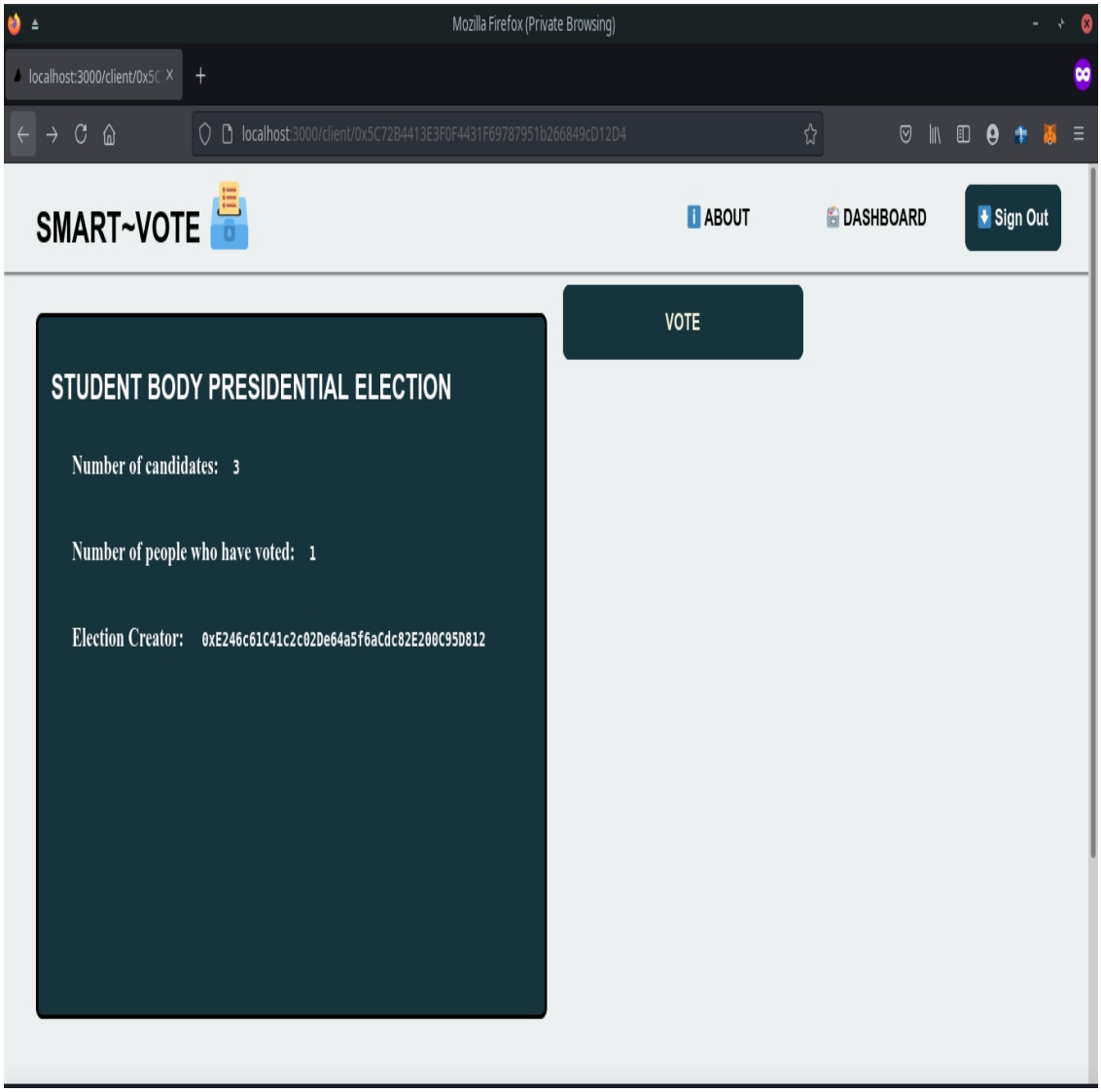

Fig 4.9: Vote Screen

`

4.3.7 Election Instance Screen

The election instance screen shows the list of all the candidates available in the election. The candidate's meta-data is also shown. The metadata for the candidates is the name of the candidate, the number of votes, and the candidate index. The system requires the candidate's index in other to vote for the candidate.

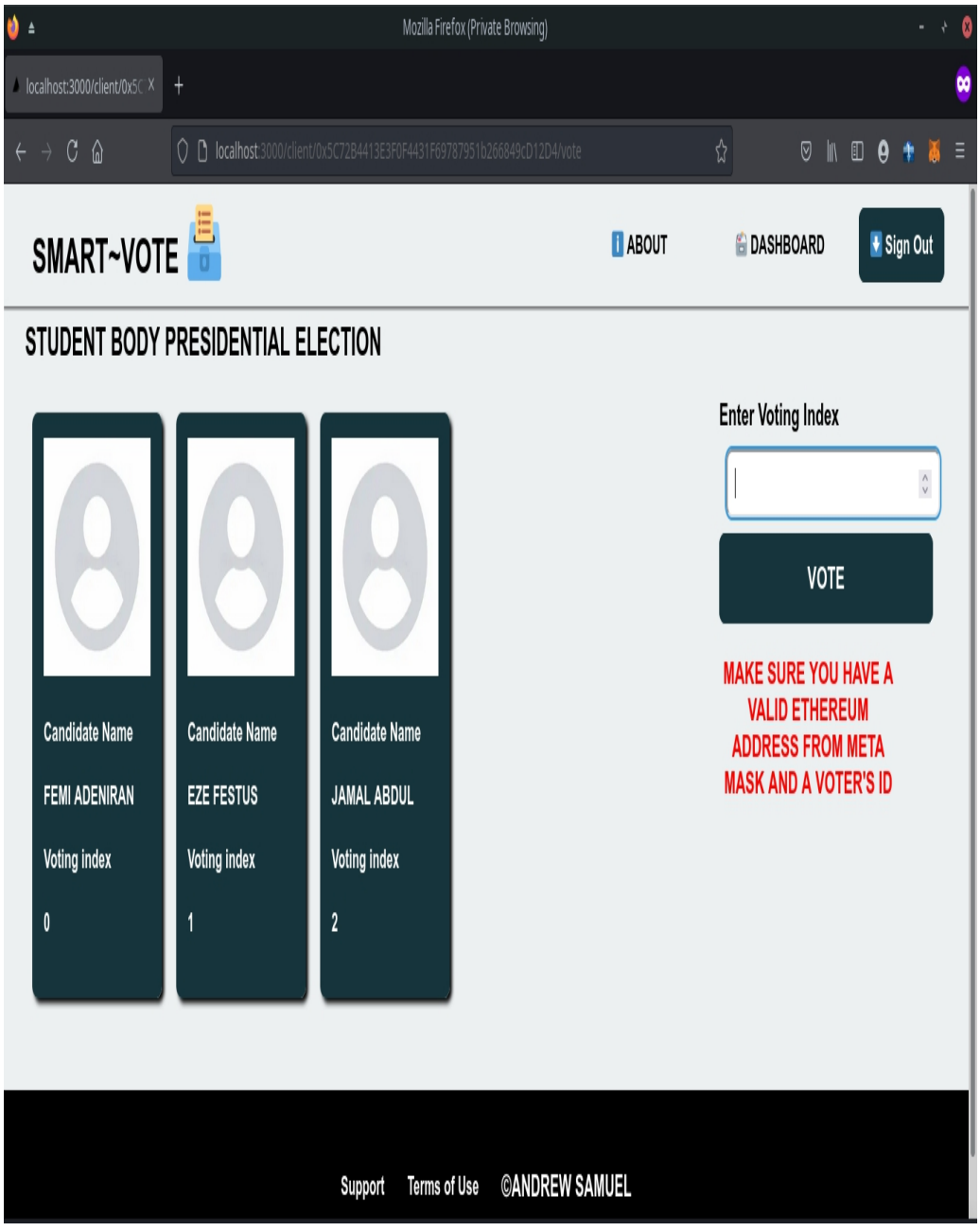

Fig 4.10: Election Instance Screen

`

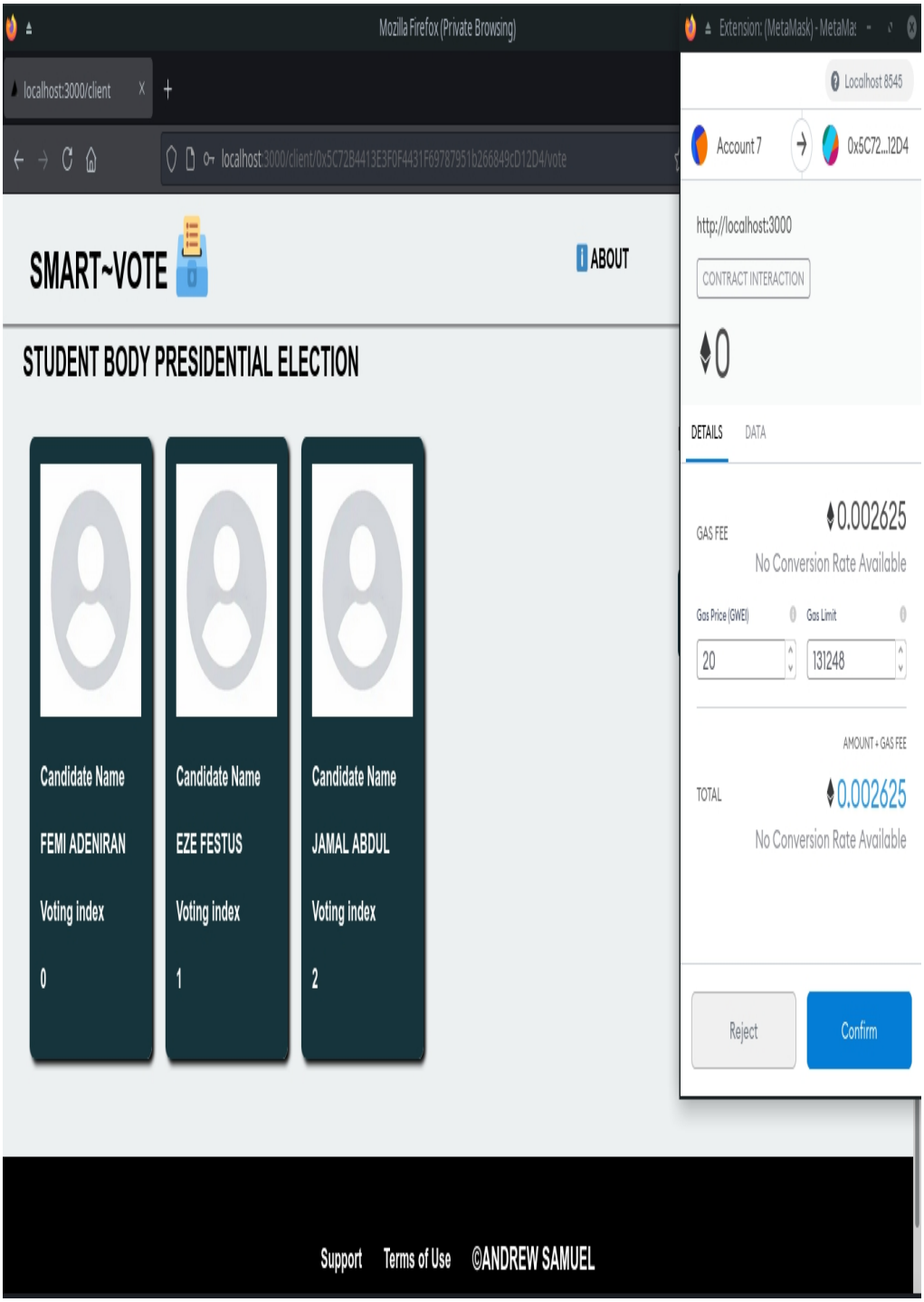

Fig 4.11: Metamask Extension pop-up transaction

On the right side of the election instance, screen voters are required to input the voting index of the candidate they want to vote for. This then enables the metamask extension to pop up on the transaction that is about to take place.

`

4.4 System Testing

During the development of the system, the system required some setup to run some basic tests. The system environment was set up to suit and mimick an ethereum network as well as connecting metamask to the localhost server.

The Ganache application was used to serve the local accounts and account balance used in the transactions, the ganache local sever served as the ethereum local network.

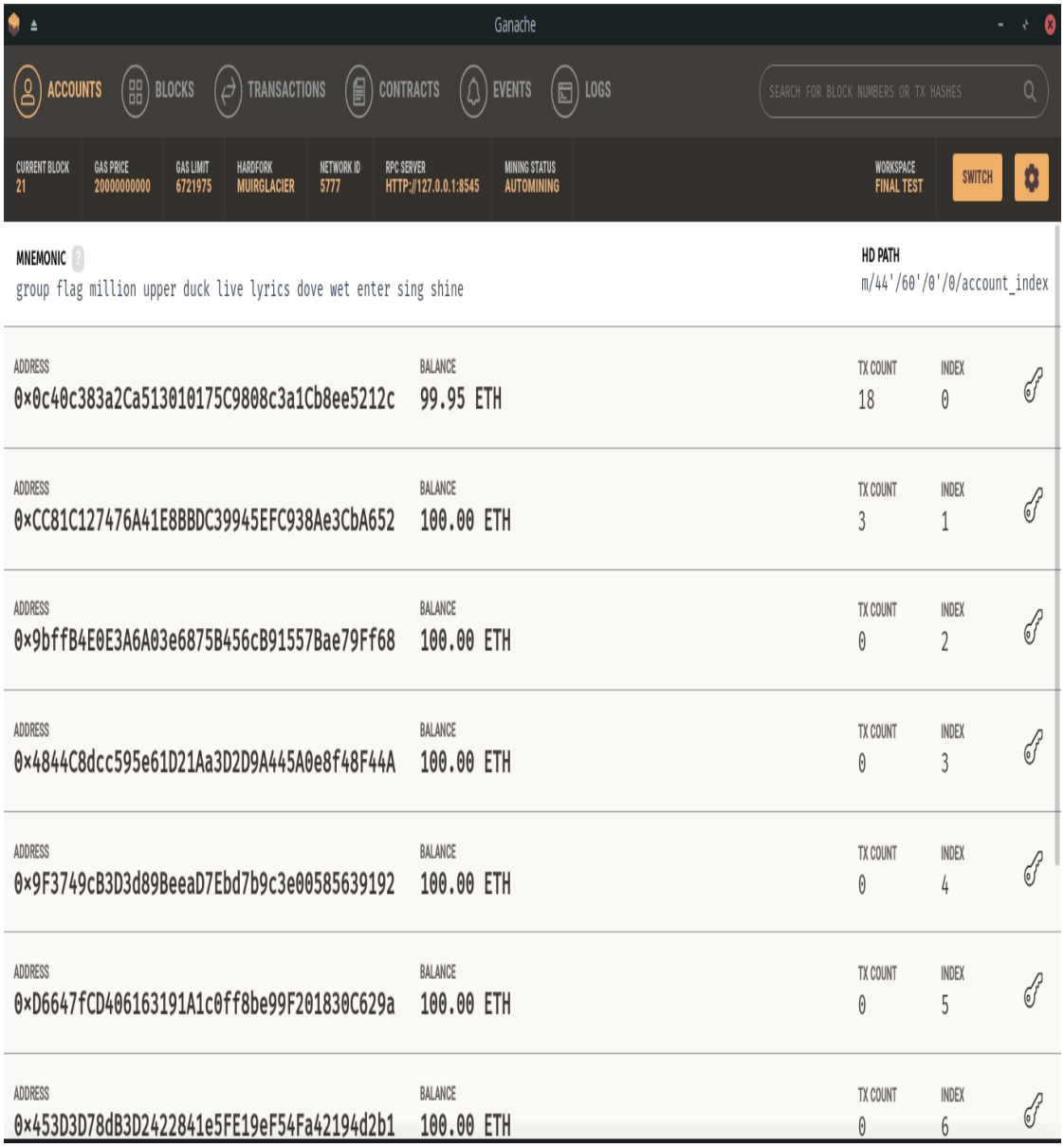

Fig 4.12: Ganache Local Server showing accounts

`

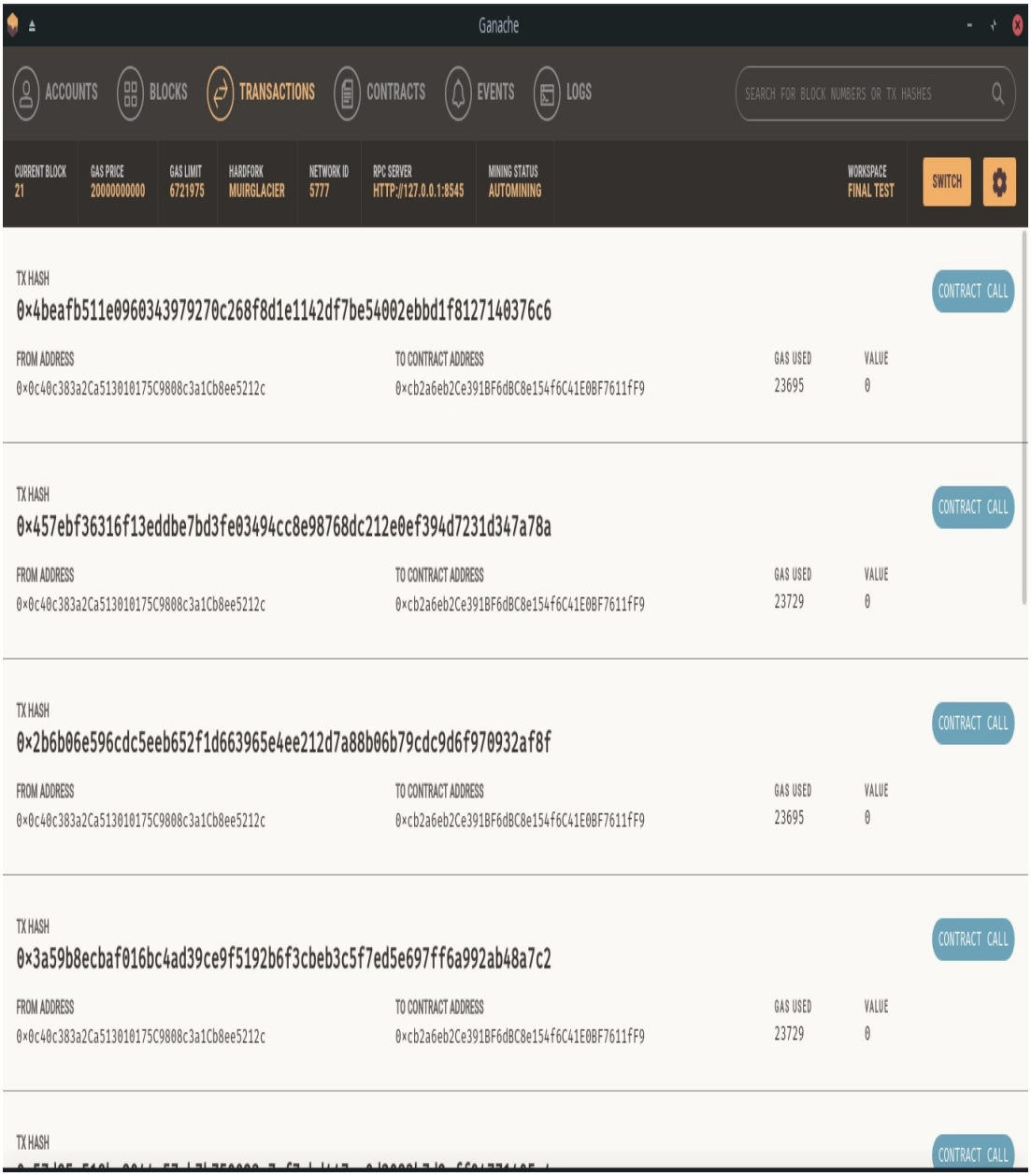

Fig 4.13: Ganache Local Server Showing Transactions

`

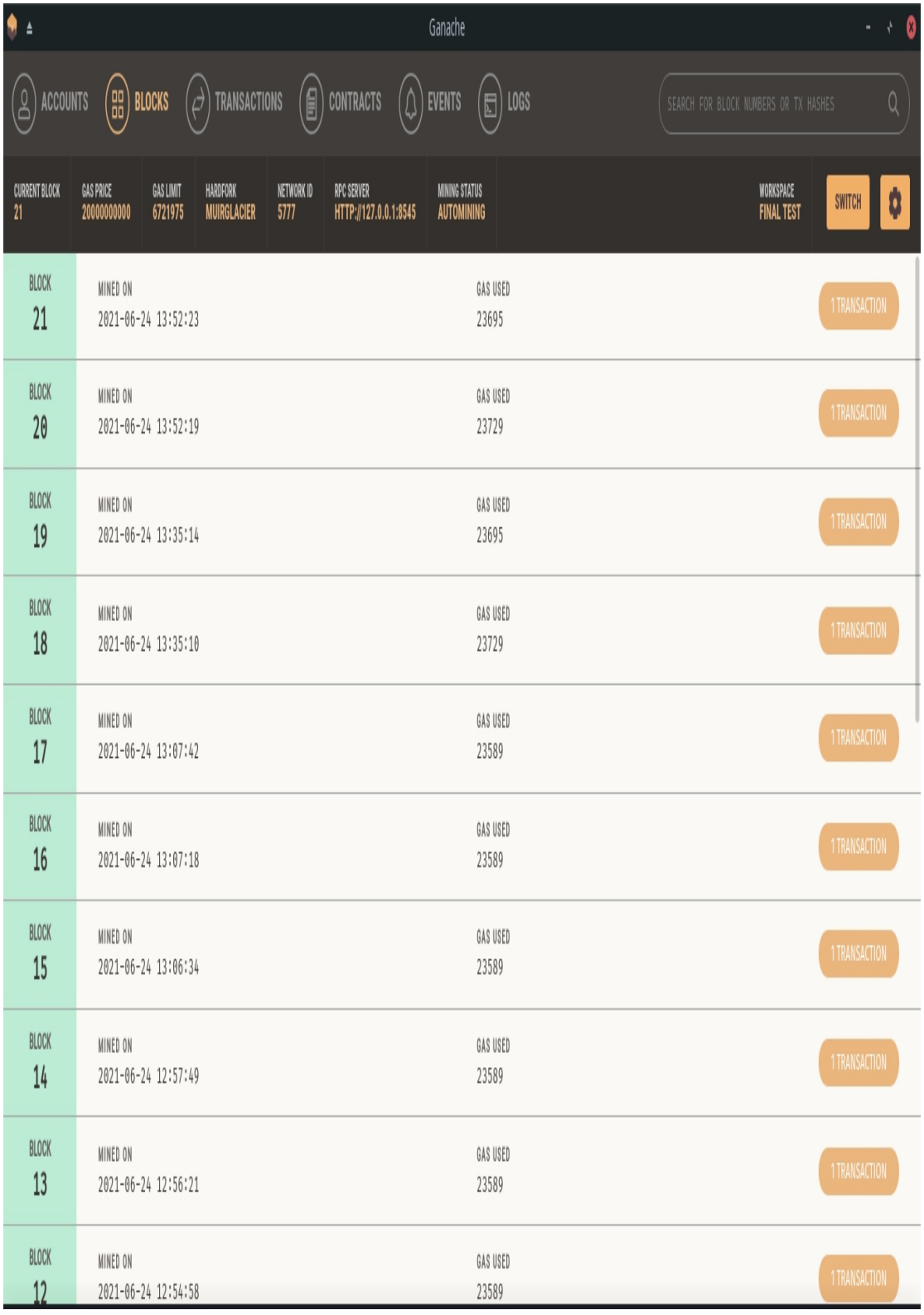

Fig 4.14: Ganache local server showing blocks in the chain

`

4.4.1 Smart Contract testing with Remix

The smart contract was tested using the remix ide, The entire flow of the application was simulated using remix, including contract deployment and interaction.

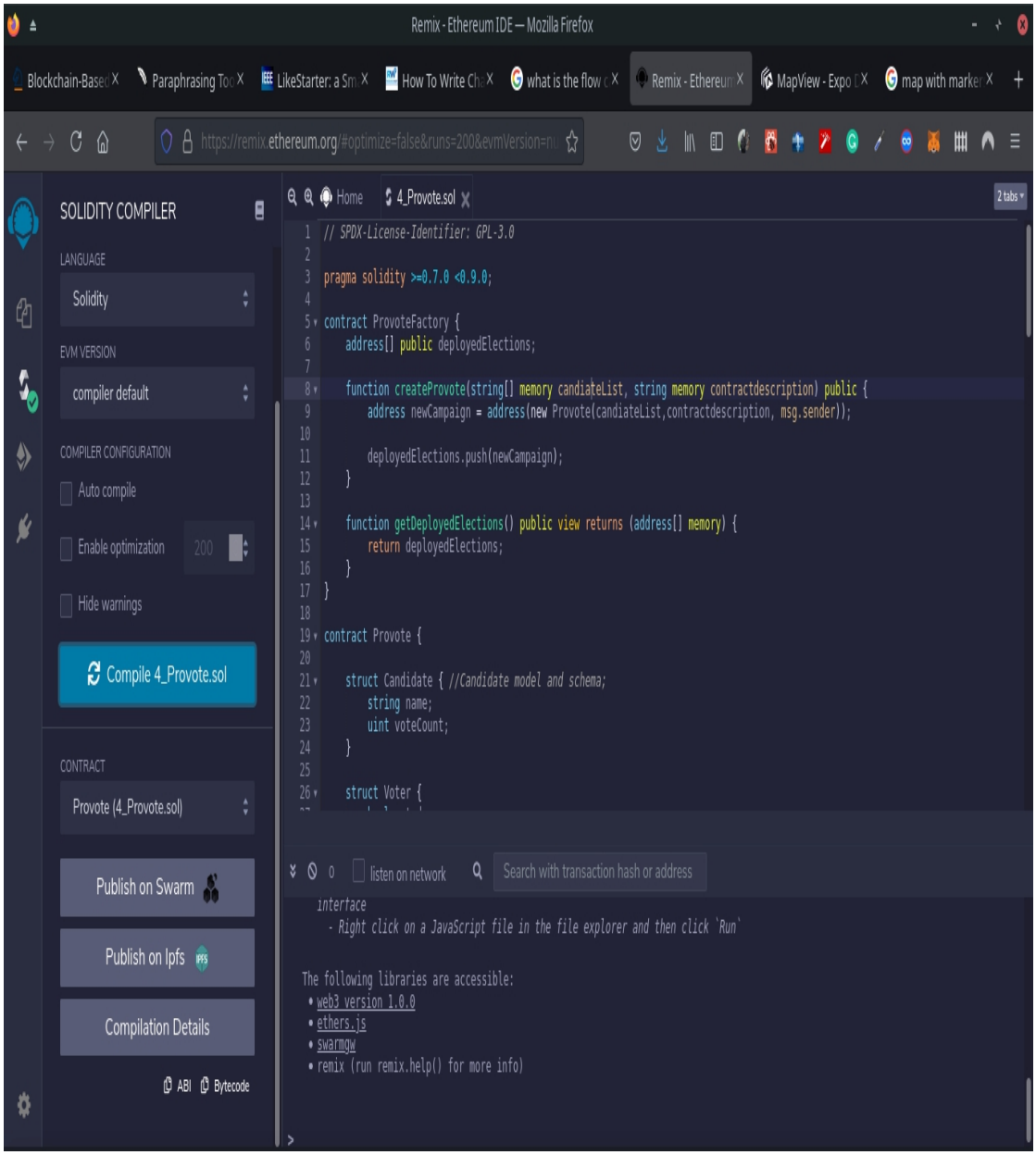

Fig 4.15 Remix Ide Compiler

`

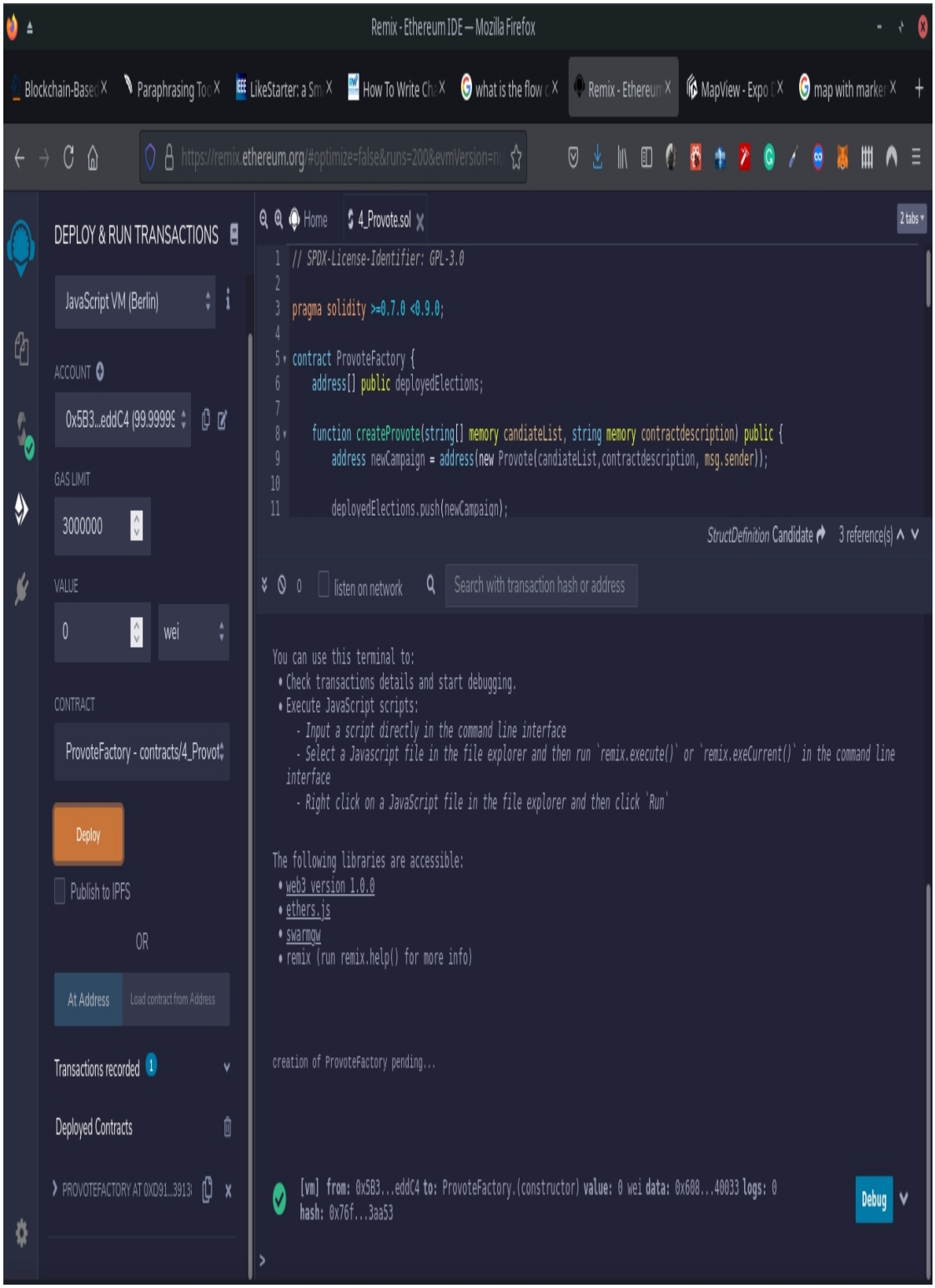

Fig 4.16: Remix Ide Deploying contract

`

4.4.2 Asynchronous Testing

A javascript testing library was used to perform an asynchronous test on the smart contract, this test was done to see how the web3 package interacts with the ethereum network, showing cases where it fails and when the test passes. Defect testing was done on some test cases to see if the smart contract handles authentication flow well and if the front-end application can utilize the output efficiently. The test cases done with mocha covers the following:

i. it should deploy a factory and an election instance.

ii. it registers the deploying address as the creator of the election.

iii. it should register a voter.

iv. it should not register a voter if the sender is not the creator.

v. it should not allow people to register if the campaign has ended.

vi. it should fail if the right to vote returns true.

vii. it should not allow users to vote on campaign day.

viii. it should not allow unregistered users to vote.

ix. it should allow registered users to vote.

x. it should return the election summary.

xi. it should end an election and announce the winner.

`

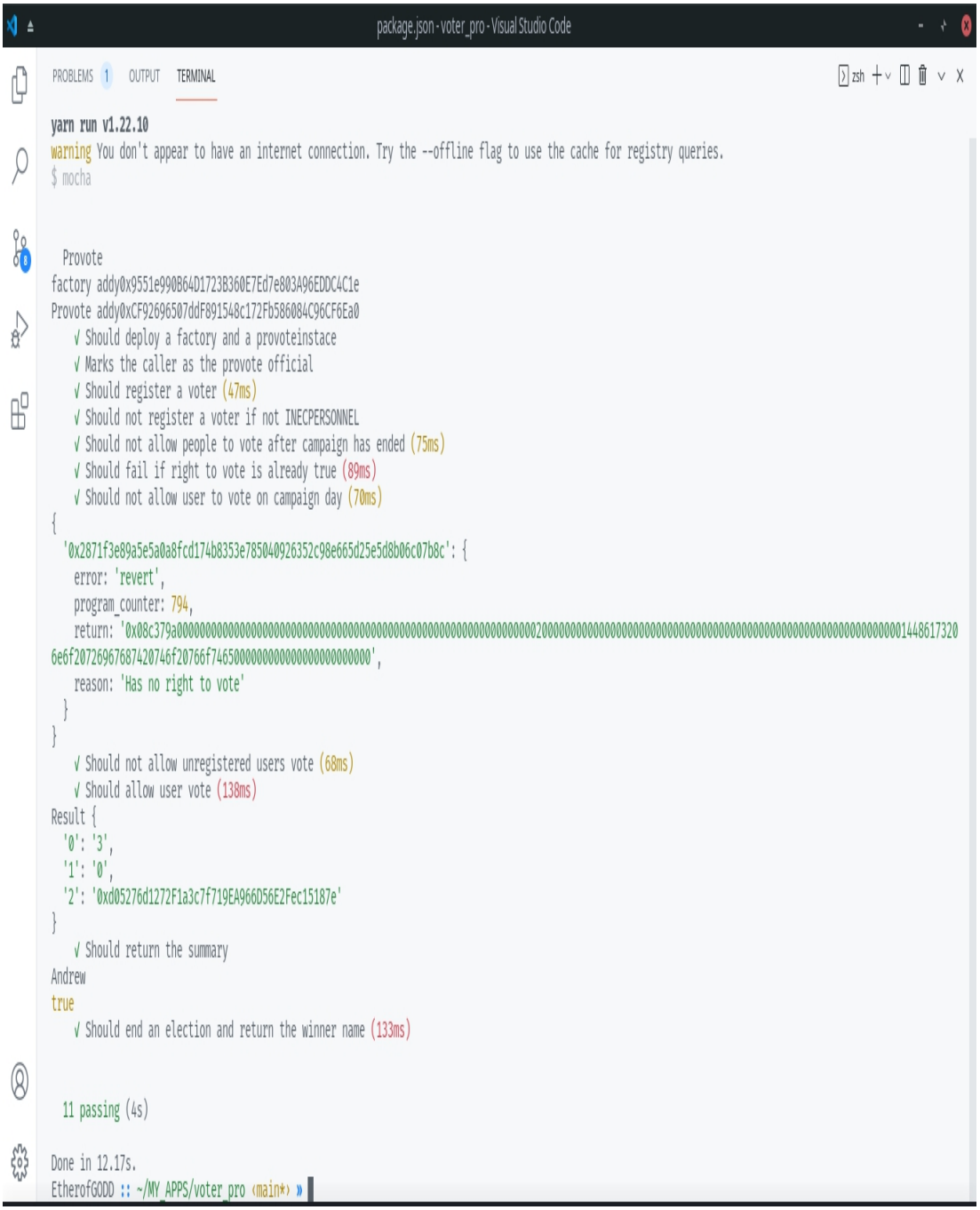

Fig 4.17: Asynchronous test with mocha.

`

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATION

5.1 Summary

This system was developed to improve the state of electronic voting, the primary aim of adding blockchain to the existing electronic voting systems is to increase trust in the system, and to aid the security process of the application.

5.2 Conclusion

In conclusion, the System highlights the faults in the paper-based method of voting as well as the limitation of a centralized voting system, this system also illustrates the use of blockchain in other aspect of everyday livelihood and how it can be applied to the voting domain. The system itself has some limitations on its' implementation, the system uses a manual way of registering voters for particular electoral instances. If the population of the users in the system increases, it would be a very hard task for admins of the system to register voters for the authenticated voting instances.

5.3 Recommendation for Further Study

This system's identification mechanism can be enhanced; the system uses a system-generated end user to register into the system, which might be invalid if anything occurs to the users. In the event that the user dies and someone else registers using the user's identification number, face recognition may be incorporated into the implementation so that each user who registers not only has their address connected to the account but also their biometric property.

`

REFERENCES

Agbu, O. (2016). Election Rigging and the Use of Technology: The Smart Card Reader as the Joker in Nigeria's 2015 Presidential Election. *Journal of African Elections*, *15*(2), 90–111. https://doi.org/10.20940/jae/2016/v15i2a5

Ahlkvist, J., Gustafsson, A., Lundborg, C., Mattsson, J. T., Sandstedt, A., & Slavnic, S. (2019). *A Decentralized Voting System*. Gothenburg, Sweden: Chalmers University of Technology.

Aluaigba, M. T. (2016). Democracy Deferred: The Effects of Electoral Malpractice on Nigeria's Path to Democratic Consolidation. *Journal of African Elections*, *15*(2), 136–158. https://doi.org/10.20940/jae/2016/v15i2a7

Ayed, A. (2017). A Conceptual Secure Blockchain Based Electronic Voting System. *International Journal of Network Security & Its Applications*, *9*(3), 01–09. https://doi.org/10.5121/ijnsa.2017.9301

Challenge of E-Voting. (2016, October 31). Retrieved June 29, 2021, from https://www.fortinet.com/blog/industry-trends/the-challenge-of-e-voting

Chandler, S. (2021, June 3). Here Are The 5 Biggest Bitcoin Transactions In History. Retrieved July 1, 2021, from https://www.cryptovantage.com/news/here-are-the-5-biggest-bitcoin-transactions-in-history/

Grigoryan, A. (2018, June 20). The Benefits of Server Side Rendering Over Client Side Rendering. Retrieved August 2, 2021, from https://medium.com/walmartglobaltech/the-benefits-of-server-side-rendering-over-client-side-rendering-5d07ff2cefe8

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

Khan, K. M., Arshad, J., & Khan, M. M. (2018). Secure Digital Voting System Based on Blockchain Technology. *International Journal of Electronic Government Research*, *14*(1), 53–62. https://doi.org/10.4018/ijegr.2018010103

Koksal, I. (2019, October 23). The Benefits Of Applying Blockchain Technology In Any Industry. Retrieved August 2, 2021, from

https://www.forbes.com/sites/ilkerkoksal/2019/10/23/the-benefits-of-applying-blockchain-technology-in-any-industry/?sh=1ff6bb7d49a5

Kumar, A. (2018, July 3). Bitcoin Blockchain - What is Proof of Work? Retrieved August 2, 2021, from https://vitalflux.com/bitcoin-blockchain-proof-work/

Loukil, F., Abed, M., & Boukadi, K. (2021). Blockchain adoption in education: a systematic literature review. *Education and Information Technologies*. Published. https://doi.org/10.1007/s10639-021-10481-8

Ma, X., Zhou, J., Yang, X., & Liu, G. (2020). A Blockchain Voting System Based on the Feedback Mechanism and Wilson Score. *Information*, *11*(12), 552. https://doi.org/10.3390/info11120552

Madise, U. & Martens, T. (2005), E-voting in Estonia 2005. The first practice of country-wide binding Internet voting in the world. Electronic voting, Bregenz, Austria.

Marella, P. B., Milojkovic, M., Mohler, J., & Dagher, G. G. (2019). GenVote: Blockchain-Based Customizable and Secure Voting Platform. *Communications in Computer and Information Science*, 152–171. https://doi.org/10.1007/978-3-030-25109-3_8

McCorry, P., Shahandashti, S. F., & Hao, F. (2017). A Smart Contract for Boardroom Voting with Maximum Voter Privacy. *Financial Cryptography and Data Security*, 357–375. https://doi.org/10.1007/978-3-319-70972-7_20

Nakamoto, S. (2008) Bitcoin: A Peer-to-Peer Electronic Cash System[White paper]. https://bitcoin.org/bitcoin.pdf

Rura, L., Issac, B., & Haldar, M. K. (2011). Secure electronic voting system based on image steganography. *2011 IEEE Conference on Open Systems*. Published. https://doi.org/10.1109/icos.2011.6079268

Ryan, P. (2008). Prêt à Voter with Paillier encryption. *Mathematical and Computer Modelling*, *48*(9–10), 1646–1662. https://doi.org/10.1016/j.mcm.2008.05.015

Shift4Shop. (2020, March 5). Understanding Bitcoins and How They're Used. Retrieved June 29, 2021, from https://blog.shift4shop.com/accept-bitcoins-with-bitpay

Suryavanshi, A. (2020). Online Voting system. *SSRN Electronic Journal*. Published. https://doi.org/10.2139/ssrn.3589075

`

Taş, R., & Tanrıöver, M. Z. (2020). A Systematic Review of Challenges and Opportunities of Blockchain for E-Voting. *Symmetry*, *12*(8), 1328. https://doi.org/10.3390/sym12081328

The Carter Center & National Democratic Institute for International Affair. (1999, June). *OBSERVING THE 1998–99 NIGERIA ELECTIONS*.

Tso, R., Liu, Z. Y., & Hsiao, J. H. (2019). Distributed E-Voting and E-Bidding Systems Based on Smart Contract. *Electronics*, *8*(4), 422. https://doi.org/10.3390/electronics8040422

Tykn. (2021, May 24). Blockchain Identity Management: The Definitive Guide (2021 Update). Retrieved August 2, 2021, from https://tykn.tech/identity-management-blockchain/