



LAKEHEAD UNIVERSITY

MASTERS OF SCIENCE

---

# E-Voting System Based on Ethereum Blockchain

---

*A Project submitted in fulfillment of the requirements  
for the degree of Masters of Science*

*in the*

Department of Computer Science  
Lakehead University,  
Thunder Bay, Ontario, Canada

## **Authors**

Bhawna Chawla

Kruthika Gururajan

Sandeep Nannamu

Department of Computer Science

Lakehead University

## **Place for Project**

Lakehead University

955 Oliver Rd,

Thunder Bay, Ontario,

Canada, P7B 5E1

## **Supervisor**

Dr. Ruizhong Wei

Chair of the Department of Computer Science

Lakehead University

# Abstract

Blockchain is one of the trending topics in the IT industry, but most people only focus on cryptocurrency. Blockchain is not only about cryptocurrency; its main feature is to provide security. Most existing e-voting systems are based on a centralized system where the voter has to trust the organizing authority for the integrity of votes. In this project, we have implemented an ethereum based e-voting system as a smart contract for the ethereum network. We have used solidity language for verifying and enforcing the constraints, and MetaMask is an ethereum wallet. User can submit their vote via a web portal, and then the ballots are handled with the consensus of every ethereum node. After the election, the ethereum blockchain will hold all the records of votes. The main of this system is to ensure data integrity and enforcing one vote per account. Ethereum Virtual Machine(EVM) is used as a blockchain runtime environment to achieve data integrity, on which smart contracts will be deployed by the organizer. This system creates an efficient and transparent environment for e-voting.

## Keywords

E-Voting, decentralized computing Blockchain, Ethereum, Ethereum Virtual Machine, Smart Contract, Navicat, MetaMask

# Acknowledgements

We would like to express our sincere gratitude to several individuals for supporting and assisting us throughout this journey of the Graduate Project. First and foremost, we offer our most profound gratitude to our supervisor, Dr.R.Wei, who has been a constant support throughout our project with his patience, knowledge, insightful comments, invaluable suggestions, and creative ideas. His extensive understanding and professional expertise in blockchain technology enabled us to understand various topics of blockchain. We are grateful to him for his valuable time in directing us.

We would like to express our genuine thanks to Lakehead University for providing us the safest and secure environment and also providing fundamental courses that helped us in building up the foundation blocks of our project. Our sincere praise for the Department of Computer Science, Lakehead University for their uniform support and timely feedback.

We would also like to thank our friends and family for their unwavering support and encouragement throughout the journey of these eight months. Finally, we thank god for his abundant blessings and for letting us through all the difficulties.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background . . . . .	1
1.2	Problem . . . . .	1
1.3	Purpose and Goals . . . . .	3
<b>2</b>	<b>Literature Review</b>	<b>5</b>
2.1	E-Voting System . . . . .	5
2.2	Blockchain-based E-Voting System . . . . .	6
<b>3</b>	<b>Architecture of Ethereum based E-voting System</b>	<b>7</b>
3.1	Overview . . . . .	7
3.2	Ethereum Blockchain . . . . .	9
3.2.1	Ether . . . . .	10
3.3	Smart Contract . . . . .	10
3.4	Ethereum Virtual Machine . . . . .	11
3.5	MetaMask . . . . .	12
3.6	Kovan Testnet . . . . .	12
3.7	Navicat Database . . . . .	13
<b>4</b>	<b>Implementation</b>	<b>15</b>
4.1	Overview . . . . .	15
4.2	Technical tools used in project . . . . .	16
4.3	Proposed Method . . . . .	17
4.3.1	Front End . . . . .	18
4.3.2	Database . . . . .	21
4.3.3	Metamask . . . . .	21
<b>5</b>	<b>Conclusion</b>	<b>24</b>

<b>6 Future Work</b>	<b>25</b>
<b>Bibliography</b>	<b>26</b>

# Chapter 1

## Introduction

### 1.1 Background

Election is an organized process to bring democracy into nations. They generally play a crucial part in the future of a country and a citizen's life. Hence, it has much significance for each single individual included in these elections. Irrespective of the organizations, elections need to be reliable. They need to guarantee people's protection and vote's security. Also, the election faculty, which is dependable for tallying votes, ought to not spend too much time on counting votes since holding up a long time increases concerns regarding manipulation of voting results. Trust is a controversial topic in any election.

Conducting honest and taut elections in any democracy is a matter of national security. For a decade, the field of computer security has been researching the possibilities of electronic voting systems, intending to reduce the cost of holding a national election while satisfying and increasing the security conditions of an election. The electoral system based on pen and paper since the advent of democratically elected candidates. Replacing the current pen and paper scheme with a modern electoral system is vital to reducing fraud and ensuring traceable and verifiable voting processes.

### 1.2 Problem

Electronic voting machines have been seen as imperfect, by the security community, basically because of the physical security concerns. Anybody with physical access to

such a device can sabotage the machine in this manner influencing all votes cast on the electronic voting machine.

Those who are doubtful of, electronic voting point to a few disadvantages and have seen dangers that are related to Web voting and telephone voting strategies. The most commonly cited problems related to security. Risks of computer virus or hacker-orchestrated 'denial of service' assaults are the most widely noted issues that might compromise an election

Inaccessible internet and phone voting give rise to more opportunities for fraud and vote-buying. Fraud happens when one person votes instead of another without their consent, and vote-buying takes place when a voter is forced by others to vote in a way that he or she wouldn't. Both show issues for vote judgment since each vote cast must be counted as the voter is expecting. There's an extra opportunity for extortion in electronic voting frameworks on the off chance that voter cards, that contain unique passwords required to cast a vote, are hacked. Privatization is additionally a concern when constituent administrators relinquish control to an unlisted firm. Contracting private companies to run the electronic operations has negative implications for some individuals. Subsequently, it has the potential to affect public confidence and belief in government and races contrarily.

There is also the danger of being susceptible to hacking. Merchants and election jurisdictions, for the most part, state that they don't transmit voting results from precincts through the web, but they may transfer them utilizing a parallel modem connection or Virtual Private Organize (VPN). Be that as it may, indeed, this approach may be subject to assault through the internet, mainly if encryption and verification are not adequate. That's because phone transmission frameworks are themselves associated with the internet and computers to which the receiving server may be associated, such as through a local area network (LAN), may have internet connections

Malicious computer program programming can also be a significant concern. Any computer program, basically generated from software programming and coding. And all these software might be tampered with by a software engineer who knows the source code. Testing electronic voting frameworks for security issues, mainly if it is intentionally presented and concealed, is mostly incomprehensible. On the off chance that software engineers embed malicious coding into a commercial program



that is activated by complex combinations of commands and keystrokes through the computer console, then election results can change totally.

Securely storing the votes is also an area of concern. The votes that are cast utilizing the electronic voting machines are stored away in a secure storage or space in the computer machine memory. Even though all electronic machines required to contain excess storage capacity, but this excess capacity isn't an autonomous record of the votes since it made by the same computer program that made the original record. As a result, the different records are of little use to check the correctness of the program.

### **1.3 Purpose and Goals**

To solve problems of both traditional and e-voting systems, e-voting is further developed by using Blockchain technology. Blockchain has impressive features to overcome inconveniences of voter's security and integrity of votes.

A blockchain is a distributed, permanent, irrefutable, open record. The record exists in numerous different locations: No single point of failure within the maintenance of the distributed ledger. There is dispersed control over who can add new transactions to the record. Any proposed "new block" to the record must reference the past version of the record, making an unchanging chain from where the blockchain gets its title, and in this way, avoiding the data tampering of previous entries. A large part of the network nodes must reach an agreement before a recently proposed new block of data gets to be a permanent part of the ledger. These features work through sophisticated cryptography, giving a security level higher than any already known databases. The blockchain technology is subsequently considered by many to be the perfect instrument, to be utilized to develop the new modern democratic voting process.

Blockchain encompasses the noteworthy potential to be an elective to general elections. It brings quick solutions to central authority problems in terms of all the blocks having all the information within the chain. It is inconceivable to alter data in a block since it is perceived by other pieces that have complete information. Thus, blockchain increments the security of data by keeping the full information in all blocks and expels the requirement for an official center to supply a secure election platform. As said before, checking the vote count and making voting results publicly accessible takes a

lot of time. Blockchain solves this issue by its nature. Since the final block on the chain keeps all data, it is sufficient to look only for the last block to check the results.

The e-voting system proposed in this paper, ethereum blockchain with the smart contracts, rises as a great candidate to be utilized for the development of a more secure, cheaper, more straightforward, and easier-to-use e-voting frameworks. Ethereum and its network is one of the most appropriate ones, due to its consistency, broad utilization, and the facility of applying smart contracts logic. An e-voting system must be secure because it ought not to permit copied votes and be entirely straightforward, whereas ensuring the protection of the voter's privacy. In this paper, an e-voting web application with a smart contract has been implemented and tested for the Ethereum network utilizing the Ethereum wallets. After an election is held, the Ethereum blockchain will maintain the records of all votes. Voters can cast their votes using the voting site, which they will be allowed access to after their identity is authenticated, and these transactions are handled with the agreement of each single Ethereum node.

# Chapter 2

## Literature Review

### 2.1 E-Voting System

E-voting is electronic-based vote casting and vote-counting system. Since the 1990s, many countries had placed various laws and regulations on e-voting. But only a few have used during the official election. The main reasons for quitting e-voting are poorly designed technology, vulnerable to cyber attacks, and security risk. Estonia is one of the first to implement an online e-voting system. It started in 2005 [1], and their system is still in use. They have used digital ID cards and personal card readers for voter's authentication. For voters to attend the election, there is a particular web portal so that anyone having a computer, internet connection, and his/her ID card can easily vote remotely. Though being successful, the Estonia model has various drawbacks too. Since it's a centralized system which creates a single point failure and therefore making it vulnerable to cyber-attacks, such attacks can harm the software database and servers. Someone can easily manipulate the valuable information during the election, and it can also lead to tampering of votes.

In 2012, France allowed its citizens, living abroad, to vote online through a portal that has been developed by a private company named as Scytl SA. Later the online voting process has been canceled by the National Cybersecurity Agency due to the high risk of cyber attacks [2].

In 2007 Netherland also banned e-voting, and in 2017, even electronic vote-counting abandoned. Everything after that comes to paper ballots and manual counting of votes due to the security concern and preventing foreign manipulation in the

election [3].

## 2.2 Blockchain-based E-Voting System

Blockchain technology owes its success from the very first cryptocurrency, Bitcoin. But their use is not limited to digital currencies.

In [4], researchers proposed an E-voting system that depends on the voter's email address. But hacking and manipulating an email address is easy; someone can register to the system by using someone else email address and cast a vote on behalf of them. Since this system does not guarantee security, data integrity, and privacy, therefore, such a system is prone to stealing elections or changing votes.

As security stands to be a significant concern in e-voting systems, in [5] blockchain was introduced into a peer to peer network to improvise the security. This system uses Distributed ledger technology or DLT to avoid forgery of votes and Elliptic-curve cryptography or ECC to provide authentication. But a drawback of using ECC is that it increases the size of encrypted messages. ECC algorithm is more complex and challenging to implement than RSA. Therefore implementation error rate increases in such a system that, in turn, reduces the security of the algorithm.

In [6], Fridrik P Hjalmarsson *et al.* aims to evaluate the framework that offers blockchain as a service and then use it to implement a distributed electronic e-voting system. Its main objective was to implement a blockchain-based voting system that improves security and decrease the cost of conducting an election.

To add on to the above systems [5] and [6], a decentralized e-voting system came into existence by combining the blockchain technology with secret sharing scheme and homomorphic encryption to eliminate third-party involvement. Still, homomorphic encryption is slow and computationally expensive.

Then to remove the third party involvement in the e-voting system, Fernando Lobato Meeser *et al.* [7] use smart contracts running on Ethereum blockchain. Ring Signature used to hide the identity of the voter and avoid multiple voting.

## Chapter 3

# Architecture of Ethereum based E-voting System

### 3.1 Overview

Voting is a critical activity that needs to be guided and carefully monitored to maintain the privacy of the voter and the vote itself. Traditional voting systems are easily tampered and leaked. Thus, introducing E-voting systems which is a website based interface for voters to cast a vote safely. These systems are implemented based on Blockchain technology for Decentralised Applications. One of the most famous Blockchain technologies standing out for its immunity towards any third party interventions is Ethereum Blockchain. We dig deep into the different aspects of Ethereum and implement its vital features. Before doing so, the main concerns of the voting system had to address. Firstly, privacy has always been an upper hand in any online application. The details of each voter have to be secured. Second, send the authentication of each voter with proper signatures and verification. Third, when the vote has been cast, it needs to enter the blockchain to avoid tampering. Below, Fig 3.1.1 shows a simple architecture of an e-voting system that we have followed for our implementation.

There are five sub-systems which regulate the flow of execution. Voter - The user who casts a vote by submitting his/her personal details such as Name, Email Address, ID Type, and ID Number. The user here creates a username and password of his/her choice to log in to the system. Voting Booth Interface – This is our website based

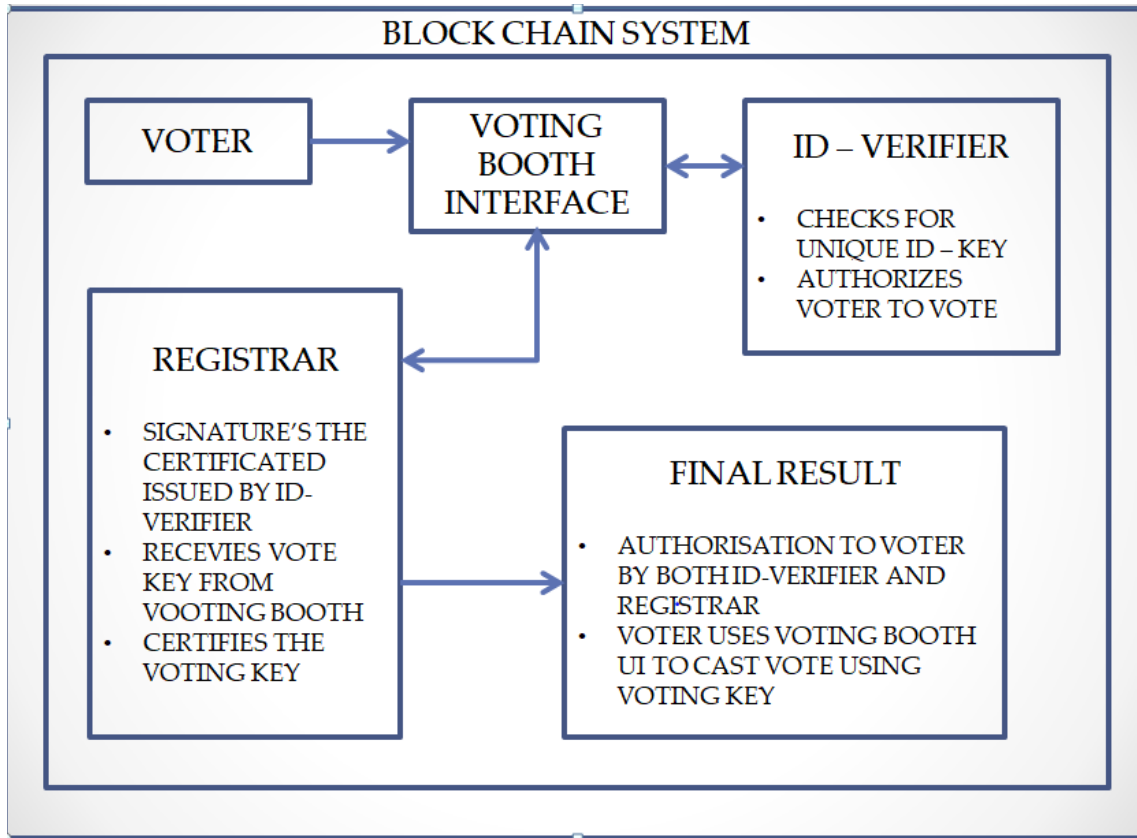


Figure 3.1.1: Architecture of Blockchain based E-Voting System.

application where the e-voting takes place. IDVerifier - A backend system that verifies the user details against the official government database and allows further processing. Registrar – Works as an admin who then activates the user upon successful verification. Final Result – a front end interface that takes in all the votes and publishes the user’s voting results.

These five sub-systems need to work in coordination to achieve a tamper-free voting process. Starting with the voter submits his/her details to the voting booth interface where the ID Verifier certifies that the ID is unique by comparing it with the official database. The uniqueness of ID authorizes the voter to vote for any ballot type. Then the ID Verifier sends a certificate with a token to the Registrar requesting a signature. The Registrar then uses the private key to sign the certificate. Once the certificate is verified and approved, then the voter is granted access to vote. Voter needs authorization from both ID Verifier and Registrar to cast a vote.

Finally, the voter then proceeds with a transaction to vote and waits for the results to be statically displayed. The voter’s voter is now sealed and can’t be removed or changed. The voter also can’t vote another time and is restricted just to view his/her results.

## 3.2 Ethereum Blockchain

Cryptocurrency has picked up a lot of interest within the past decade, particularly with the rise of Bitcoin. But as of late, a new rising star has entered the arena — Ethereum. Interest in this cryptocurrency is so massive, forecasts have indicated that the cost of graphics cards may be driven up, as miners endeavor to produce as much money as conceivable.

Ethereum is a distributed public blockchain that centers on running the programming code of any decentralized application. More basically, it is a platform for sharing data over the globe that cannot be controlled or manipulated. Ethereum can be considered as a network of computers, all running on the Ethereum blockchain and permitting individuals to trade coins. In this way, it's comparable to the prevalent cryptocurrency Bitcoin. In recent years, Ethereum has risen to end up as the second most well known digital currency in the world, with bitcoin being the most popular of all.

Before the creation of the Ethereum blockchain, applications were built to execute a restricted set of operations. There was a characterized constrain as to what they may do. For instance, Bitcoin and other comparable cryptocurrencies are created to function only as “peer-to-peer currencies,” and this was the only purpose which they served. But developers required something with more noteworthy usefulness.

The innovator of Ethereum Blockchain, Vitalik Buterin, had a thought for a completely modern approach. Not like conventional blockchain applications, which established within the fundamental concept of generating and transferring cryptocurrencies, Buterin's vision was to understand the issues within the area and create a new utilization of the blockchain technologies. One of the primary innovations of Ethereum, and a massive differentiator among other offerings, is the Ethereum Virtual Machine, which makes the method of making blockchain applications more straightforward and more proficient.

The structure of the Ethereum blockchain is comparative to Bitcoin's, in that it offers the record of whole transaction history, but what makes Ethereum distinct from Bitcoin is that it involves the new use applications, such as “smart contracts.” Ethereum utilizes blockchain innovation, not as it were for keeping up a decentralized network for payment but also for storing software code that can be used to control tamper-proof decentralized finance contracts and applications. The main components required for

implementing an application on Ethereum blockchain are explained below.

### **3.2.1 Ether**

For carrying out operations in the ethereum blockchain, the cryptocurrency used is referred to as ether. Ether is a decentralized digital currency, also cited in short as ETH. Apart from being a tradeable cryptocurrency, ether powers the Ethereum network by paying fees for transactions and other computational services. Ether and bitcoin are similar in numerous ways. For example, both of them are digital currency exchanged using online trades and stored away in different sorts of cryptocurrency wallets, and both of these tokens are decentralized.

The process of getting ether is different in each nation or at the slightest by currency. You have to search and find somebody either online or in-person who has ether and would like to exchange. There's also the alternative of meeting in-person to purchase or offer ether, mainly if they are living in a city with regular Ethereum meetups. Buying ether with another currency might be an additional step.

Bitcoin is the commonly utilized cryptocurrency, and individuals around the world are more likely in need to make a trade for it in their cash. So, on the off chance that you need to purchase ether, for instance, the easiest way may well be to buy bitcoin at an exchange and after that exchange that for ether. Once you've got ether, you'll be able to send it straightforwardly to another individual.

## **3.3 Smart Contract**

Unlike Bitcoin, which is an entirely digital currency, and generally acts as a method of payment, Ethereum takes a diverse approach, and functions as a stage through which individuals can utilize tokens to develop and run applications and make smart contracts. What could be a smart contract? The smart contract is one of the significant contrasts between Ethereum and other sorts of digital currency. It's a contract, but not in the form of a digital document; it's composed in code. The maker then transfers the contract into the blockchain. In other words, smart contracts are programs that administer the behavior of accounts inside the Ethereum state.

Proceedings on the terms of the contract are followed up and stored away within the public ledger, which makes the details encompassing that contract tamper-proof. But



how? The contract code is organized as “if-then statements.” For example, let’s say that you rent a house from a real estate company that utilizes Ethereum. A smart contract is created, and once you transfer the required amount of currency, the system consequently sends a computerized key to unlock the house. This whole process is carried out on the blockchain, and everybody can see what has been done.

Solidity is an object-oriented, high-level language mainly used for implementing smart contracts. In this project, this solidity language is used to write the contract code, which involves critical functions such as giving voting rights to users after verifying their identity, storing the vote cast by users, getting current voting status.

### **3.4 Ethereum Virtual Machine**

One of the main innovations of Ethereum, and a massive differentiator among other offerings, is the Ethereum Virtual Machine. Instead of requiring to construct a unique blockchain for each new application, this new blockchain innovation may empower the improvement of possibly thousands of distinctive applications — all within a single platform. This thought inevitably got to be the Ethereum Virtual Machine. The Ethereum Virtual Machine serves as the runtime environment for “smart contracts” inside Ethereum.

Each node on the network runs “EVM implementation” and executes based on a predefined set of directives. Each operation that’s performed inside EVM is at the same time executed by each single node inside the network. This handle is called “gas,” which is essentially an Ethereum exchange code that triggers information read and write operations and does large computations. Each action that a client completes on the network involves a cost, which is measured by “gas.” Each “gas” unit is paid for utilizing ether, which is the currency of the Ethereum network.

EVM guarantees that programs don’t have access to one another’s state, which ensures that communication is built up without the hazard of potential impedances. And since EVM is separated from the leading network, it’s the perfect testing environment. For instance, a company can produce a smart contract in EVM without contrarily influencing the primary blockchain operations. Most think of EVM as a learning environment to better understand the functioning of Ethereum, but this innovation is mainly centered on making a decentralized and independent environment.

### 3.5 MetaMask

The system proposed in this paper is a decentralized application (dApp), which is an application that runs on a decentralized network and employs its resources. Any smart contract or dApp code can be modified for EVM and executed by the decentralized ethereum computer network. To run Ehtereum dApp within the web browser, the client must introduce a browser extension that permits the browser to interact with the blockchain and oversee the user's personality. Metamask is the foremost prevalent solution for the desktop.

MetaMask gives a secure and straightforward way to connect with blockchain-based applications. The user is continuously in control when interacting with the new decentralized web application. MetaMask is a browser plugin, available as the MetaMask Chrome extension or Firefox Add-on. At its core, it serves as an Ethereum wallet: By using it, the user will get access to a one of a kind Ethereum public address, with which the user can begin sending and getting ether or tokens. But MetaMask does something more than function as an Ethereum wallet. As a browser extension, it can interact with the current webpage you're browsing. It does so by infusing a JavaScript library called web3.js. When MetaMask is installed, any front-end code can get to the smart contract functionalities, and interact with the blockchain.

### 3.6 Kovan Testnet

Although all the application are deployed on the main ethereum blockchain, there are also some testnets which simulate the main ethereum blockchain. The reason for creating these testnets is, before an application dispatches on the Ethereum blockchain or before changes are made inside the blockchain itself, a version is sent to an Ethereum Test Network ("testnet"), which mimics the leading Ethereum network. Three testnets are primarily in use as of now – Ropsten, Kovan, and Rinkeby. The designers of this kovan testnet proposed the arrangement of a public Proof-of-Authority (PoA) Ethereum testnet, named "Kovan." This modern testnet will be utilizing Parity (an Ethereum client created by Parity Technologies) to supply a steady, secure testnet environment for Ethereum designers, due to the flimsiness of the existing Ropsten testnet.

Parity support a PoA agreement engine to be utilized with Ethereum Virtual Machine

(EVM) based chains. PoA could be a substitution for Proof of Work, which can be used for both open and private chain setups. There's no mining included to secure the organize with PoA and depends on trusted 'Validators' to guarantee that substantial transactions are added to blocks, prepared and executed by the EVM steadfastly. Since mining does not happen on this public test net, malevolent miners are prevented from obtaining testnet Ether, solving the spam attack that Ropsten is as of now confronting. There's no distinction within the way that contracts are executed compared to PoW chains so that engineers can test their contracts and application interfaces dispatching the application to the mainnet in a more dependable and helpful environment.

Kovan ether has no market value; they can be obtained for free, and by visiting <https://faucet.kovan.network/> are meant for testing purposes only. Kovan network faucets provide Kovan Ether to legitimate developers willing to deploy and test contracts on the Kovan network. Benefits of Kovan testnet include shorter block times, allowing for more rapid deployment, testing and iteration, and reduced overall maintenance costs.

### **3.7 Navicat Database**

The backend database is implemented using Navicat. Navicat is a graphical database administration and development software delivered by PremiumSoft CyberTech Ltd. for MySQL, MariaDB, MongoDB, Prophet, SQLite, PostgreSQL, and Microsoft SQL Server. It has an Explorer-like graphical client interface and bolsters different database associations for the neighborhood and remote databases. Its design is made to meet the requirements of a diverse group of audiences, from database administrators and software engineers to different businesses/companies that serve clients and share data with partners.

The above-described components have been utilized to develop a decentralized e-voting web-based application which is deployed on kovan testnet. Compared to traditional centralized applications, dApps (decentralized applications) are more dependable, since they utilize the plus points of a decentralized network. A dApp stores information in a decentralized database and employs decentralized computing assets to work. Blockchain-based dApps don't deliver possession to one central specialist and, consequently, can be utilized for distinctive interfacing individuals in marketplaces; sharing assets and putting them away; keeping up cryptos; executing shrewd contracts.

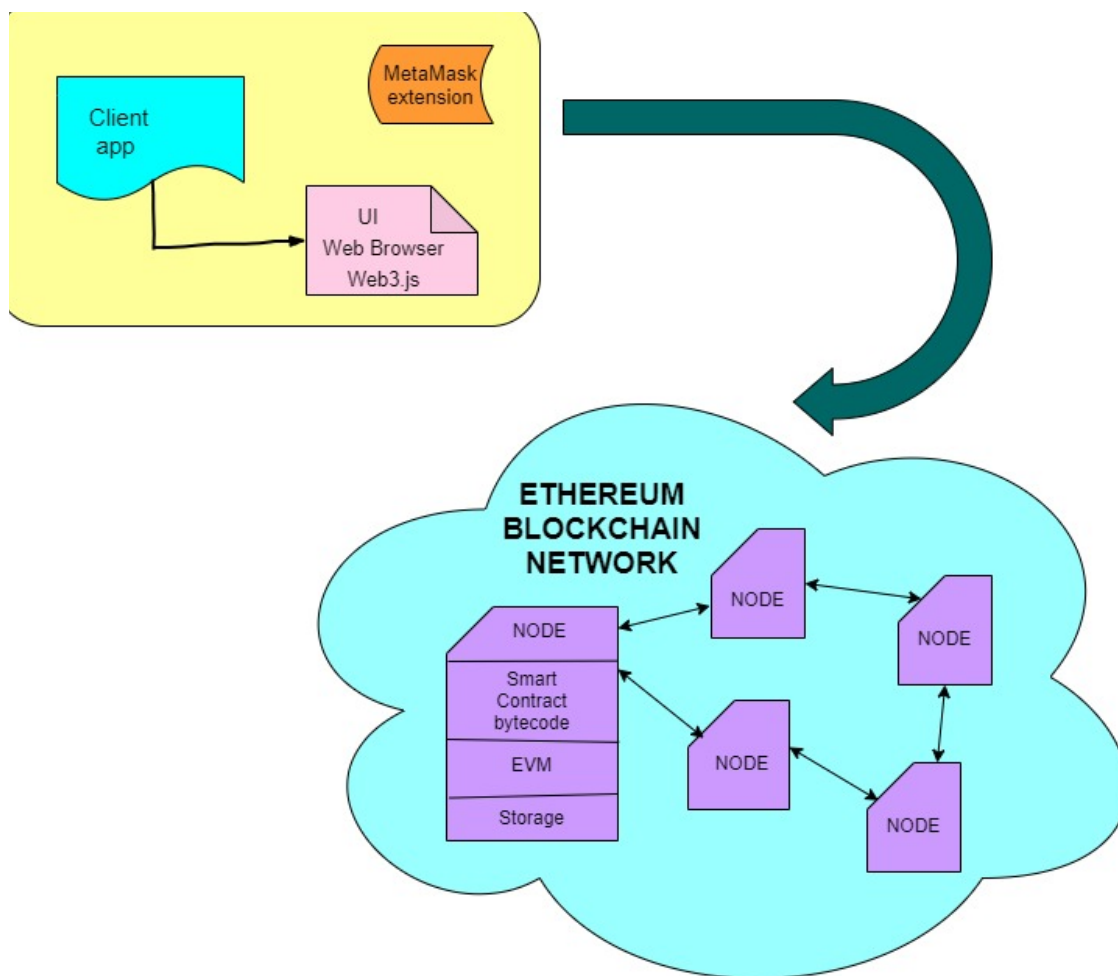


Figure 3.7.1: DApp Implementation on Ethereum Blockchain

It is necessarily outlandish to bring down a dApp since it'll require a programmer to require down all the distributed hosting nodes.

# Chapter 4

## Implementation

### 4.1 Overview

Electronic voting systems can be implemented as a mobile application or a website based system. As we use the different aspects of Ethereum Blockchain into our decentralized application, we choose to create a smooth, secure, and accessible frontend using which users can easily register to vote. Based on the core architecture of the e-voting system and the Ethereum Properties, the flow of the system is as presented below using a sequence diagram Fig 4.1.1. For a fully functional e-voting system, there are two leading ends. One is the Voter, and the other is Admin. These are two different people who access the system.

The system here is referred to as the Government Portal, which our web-based application using which voters can cast their votes. The portal consists of a Login/Sign In page for the voter to register, and it also includes an admin page, which allows the admin to verify and activate the user. The main page of the portal is the voting page where each candidate's profile who is running for the elections is present. This page gives an easily accessible interface for the voters to understand each candidate and cast their votes wisely. The authentication process uses the navicat database to provide right and secure access to the voters. Navicat maintains SQL Lite data holding all the original reliable information of each other, which we call it the dummy database as it is a small scale data. This dummy database is used to compare the data enter by each voter, verify the data, and provide the authentication. Once the authentication is approved by the admin and gives the user the right to vote. Each user can cast their

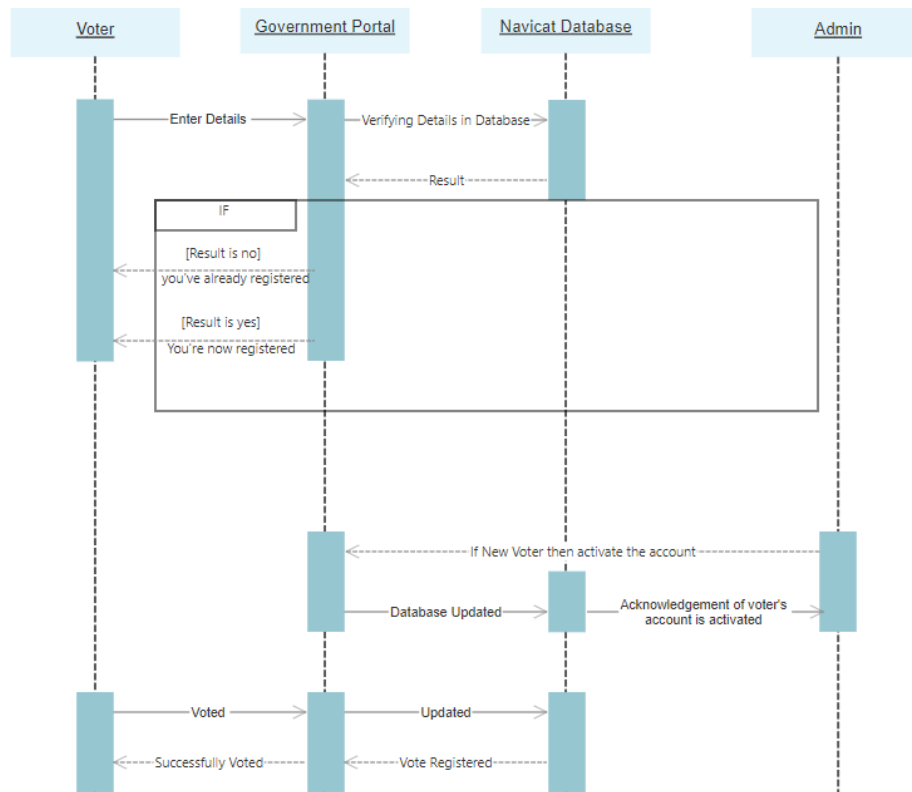


Figure 4.1.1: Sequence diagram of the proposed system.

votes by logs in to the system. If the access is not granted, then the user can't vote anymore. There are two reasons why this would happen. First, the details entered by the user might be false, or the user would have already voted before and can't vote again. Thus ensuring there is not tampering in any vote cast.

## 4.2 Technical tools used in project

To build an interactive front end and a secure database for the voters, we explore different functionalities of python in this system. As Pycharm known for its refactoring options, we used to Django framework and implemented the python-based codes in the same. Firstly, HTML and CSS was used to establish a secure interface. Now when the voter registers, the data has to be stored into a database for further retrieval and access. Thus we use the Navicat database to store all the data entered by the user securely. Database here also stores a government dummy original data, which is a list of users and their details. It is used for the verification process.

Once we had our frontend and backend ready, then we work towards implementing the Ethereum Blockchain. Metamask, which is a browser extension, is used to access the

Ethereum Wallet where ethers are used for carrying out any transaction. JavaScript files are used to forward inputs and outputs. Furthermore, as Ethereum stands out for its elimination of the third party for carrying out a seamless transaction, Smart Contracts are coded using Solidity Language with JSON inputs. Overall it is a decentralized application with active buttons and sales establishing an interactive interface.

### 4.3 Proposed Method

As described above, using the different technical tools and the flow of the system, we divide the implementation into separate parts. Fig 4.3.1 shows the flow of each essential part, such as voter admin and the MetaMask. All of these are inter-dependent and need each part is necessary to regulate the E-voting system. Without the admin's approval, the user can't be activated or authenticated to vote. To vote or approve the authentication of the user, the MetaMask is required to carry on transactions using ethers. Metamask here, however, is used in Kovan Test Network.

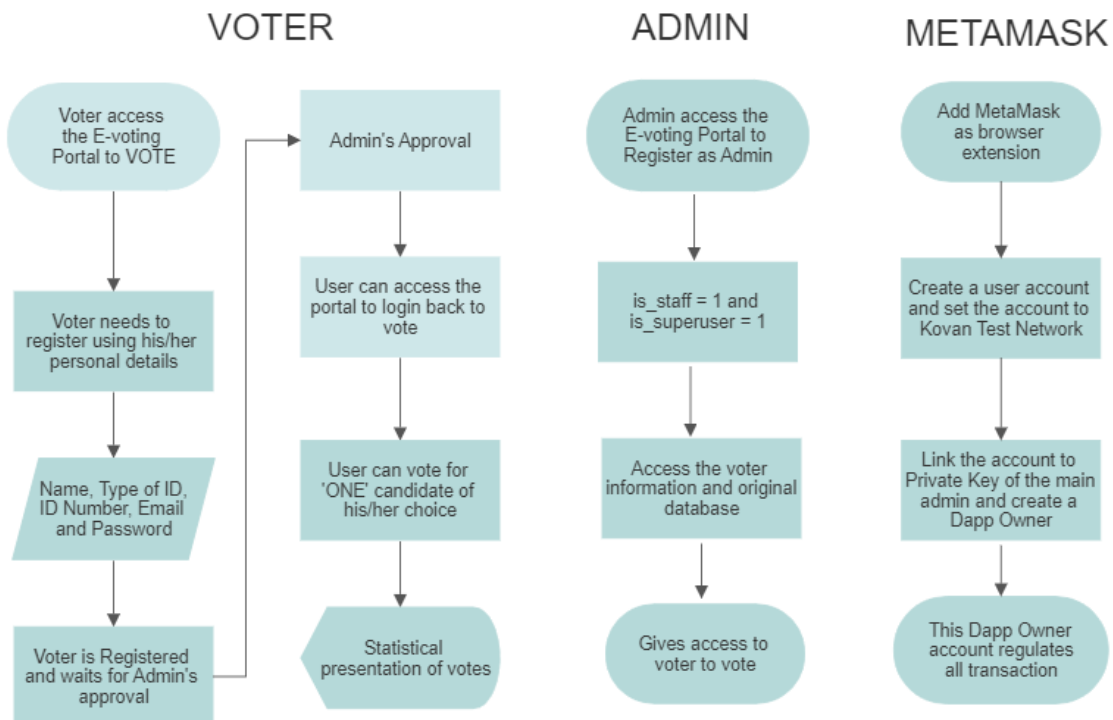
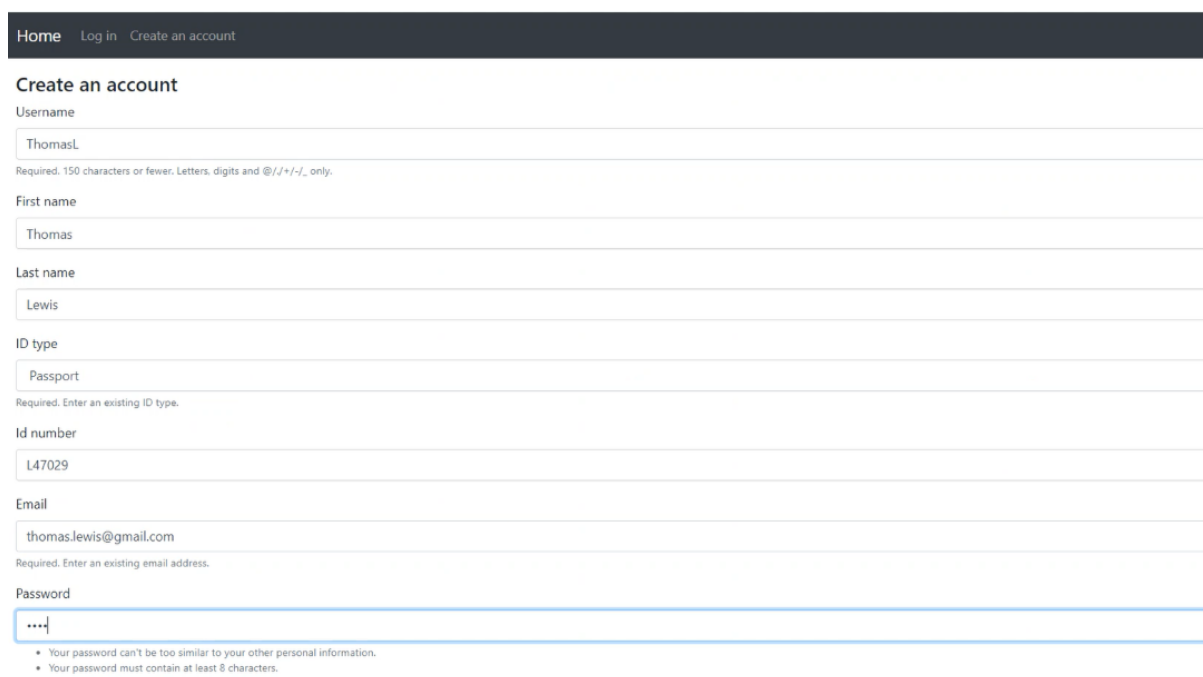


Figure 4.3.1: Flow Diagram

### 4.3.1 Front End

The front end development has three stages. Namely, the Login/Sign Up Stage where the user can register themselves by providing their details such as Name, Email ID, Type of ID, ID Number Username, and Password. The user should register with all the right information. Several checks are implemented, such as ID Number check, ID Type Check, Password Checks, etc. Only upon receiving the correct details according to the database, the user can proceed with the registration. Otherwise, the page stops the user from registering and requests the user to enter the correct details. Once the user has entered all the details, they will be prompted to wait for the admin's approval. Following this, user's also can reset their password, and even forgotten password options are available. Simple HTML and CSS scripts are used to create and navigate through the pages, as shown below. Fig 4.3.2 shows the Sign Up Page, Fig 4.3.3 shows the Request page to wait for Admin's Approval.



The screenshot displays a registration form titled "Create an account". At the top, there is a navigation bar with links: "Home", "Log in", and "Create an account". The form fields and their values are as follows:

- Username:** ThomasL. Below the field is a validation message: "Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only."
- First name:** Thomas
- Last name:** Lewis
- ID type:** Passport. Below the field is a validation message: "Required. Enter an existing ID type."
- Id number:** L47029
- Email:** thomas.lewis@gmail.com. Below the field is a validation message: "Required. Enter an existing email address."
- Password:** The field is currently empty and highlighted with a blue border. Below it are two validation messages:
  - Your password can't be too similar to your other personal information.
  - Your password must contain at least 8 characters.

Figure 4.3.2: Registration Page

Second, comes the Admin Stage where the Admin home page is shown in Fig 4.3.4. Once the Admin logs in, they can find a list of user's who have previously signed up and also who have newly signed in. Admin has a list of the original voters stored in the form of Dummy Database and a field that indicates if the details entered by the user are the same as per the database records. If this field indicates a tick mark, then the details are matched as per the record, else it indicates that the user has entered the wrong information. This comparison is made using the python codes, which are



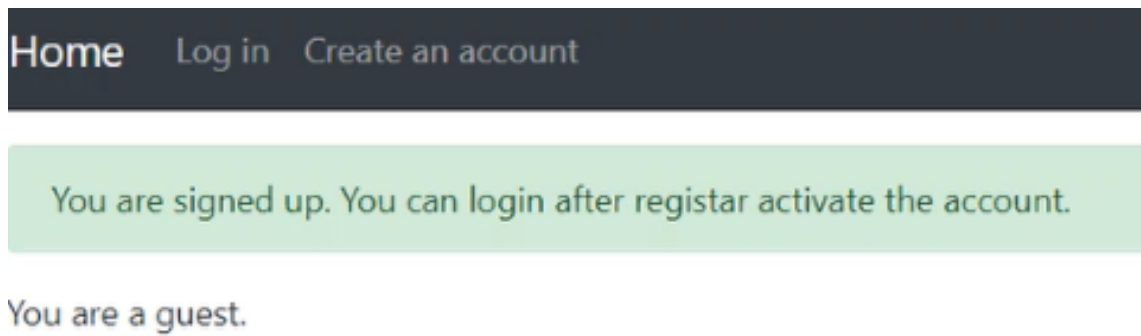


Figure 4.3.3: User Home Page Before Activation

connected to the Navicat database. The Admins use the information in this field to further decide if they need to authenticate the user or not. If found all correct, the user is activated using an activate button. Note: during this activity, the Ethereum Blockchain is activated through the MetaMask, and transactions are called in. Some ether's need to be spent to confirm each transaction.

ID	First Name	Last Name	Joined Date	Type of ID	username	Email	Privatekey	Active	Vote status	Vote Result	Dummy Exist?
942VY317	Margaret	Brooks	March 18, 2020, 12:24 a.m.	StateID	MargaretB	margaret.brooks@gmail.com	0x13df17a68c311ea	✓ activated	🗳️	candidate2	✓
L47029	Thomas	Lewis	March 30, 2020, 7:39 p.m.	Passport	ThomasL	thomas.lewis@gmail.com	0x0	✓ activated			✓
L47032	Joan	Stewart	March 28, 2020, 8:39 p.m.	Passport	JoanS	joan.stewart@yahoo.com	0x71649cac713411ea	✓ activated	🗳️	candidate2	✓
L47033	Ruby	Rogers	March 30, 2020, 7:41 p.m.	Passport	RubyR	ruby.rogers@gmail.com	0x1755994272bf11ea	✓ activated	🗳️	candidate1	✓
L47034	Carolyn	Hayes	March 30, 2020, 5:18 p.m.	Passport	CarolynH	carolyn.hayes@hotmail.co.uk	0xa0c92ee734611ea	✓ activated			✓
N045068909	Brenda	Butler	March 31, 2020, 3:03 p.m.	Driving License	BrendaB	brenda.butler@gmail.com	0x0	✓ activated			✓
N045068910	Lillian	Brown	March 31, 2020, 3:04 p.m.	Driving License	LillianB	lillian.brown@aol.com	0x0	✓ set active			✓

Figure 4.3.4: Admin Home Page

Now the user's account will be activated, turning them into voters. Fig 4.3.5 shows the dummy database page. Also, notice that there exists a field named voted, which indicates if a particular user has voted or not. If the user has already voted, then the system prohibits voting again, thus keeping it secure and untampered.

Third and the final stage is the voting page, where a voter can cast his/her vote upon successful authentication. A list of candidates who are running for the elections, with their profile, will be shown on the page. Voters can read through each of these information and then decide who he/she will cast a vote. This information helps each voter to be well educated on the candidate they choose, making it a fair election.

HomeChange passwordChange profileChange emailLog out

Change language

Your username is BrendaB and you are regitar .

Voters database

Dummy govt database

Show10entries

Search:

ID	First Name	Last Name	Birthday	Type of ID	ID Number date	Email	SIN
1	Lois	Walker	March 29, 1981	Passport	L47010	lois.walker@hotmail.com	9101304181
2	Brenda	Robinson	July 31, 1970	Passport	L47011	brenda.robinson@gmail.com	9101304182
3	Joe	Robinson	June 16, 1963	Passport	L47012	joe.robinson@gmail.com	9101304183
4	Diane	Evans	April 12, 1977	Passport	L47013	diane.evans@yahoo.com	9101304184
5	Benjamin	Russell	April 17, 1977	Passport	L47014	benjamin.russell@charter.net	9101304185
6	Patrick	Bailey	Sept. 27, 1982	Passport	L47015	patrick.bailey@aol.com	9101304186
7	Nancy	Baker	June 13, 1995	Passport	L47016	nancy.baker@bp.com	9101304187
8	Carol	Murphy	June 30, 1958	Passport	L47017	carol.murphy@gmail.com	9101304188
9	Frances	Young	Sept. 6, 1959	Passport	L47018	frances.young@gmail.com	9101304189
10	Diana	Peterson	Nov. 13, 1987	Passport	L47019	diana.peterson@hotmail.co.uk	9101304190
ID	First Name	Last Name	Birthday	Type of ID	ID Number date	Email	SIN

Showing 1 to 10 of 79 entries

Previous12345...8Next

Figure 4.3.5: Government Dummy Database

Alongside each profile, a ratio will be shown with the number of votes received to the total number of voters. Fig 4.3.6 shows the candidate profile and ratio. For better understanding and view of the votes, votes are resulted out as a statistical representation. The percentages of votes are shown in the group as in Fig 4.3.7.

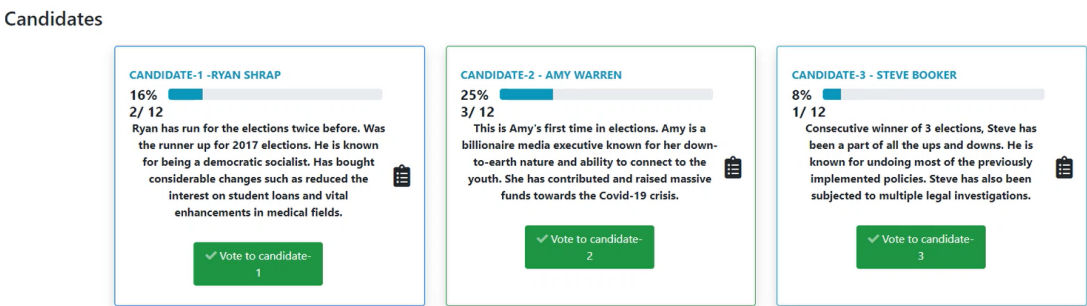


Figure 4.3.6: Voting Booth

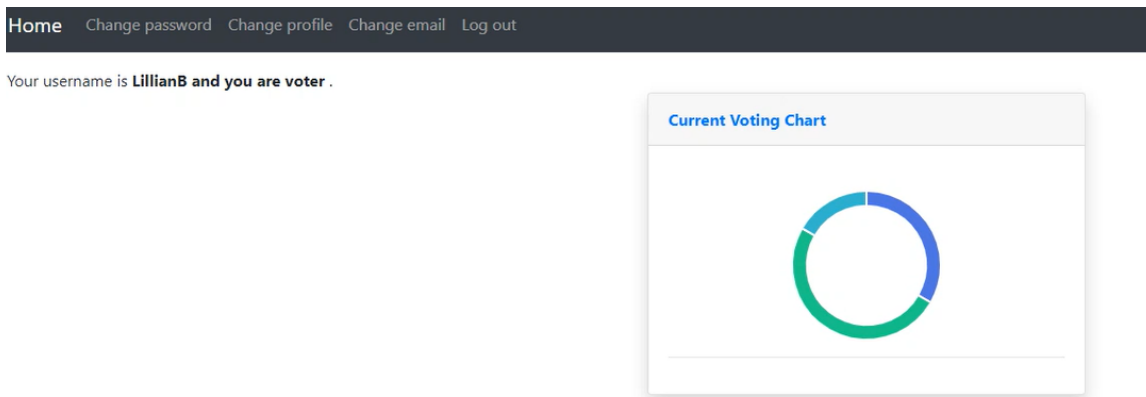


Figure 4.3.7: Statistical Representation of Votes

### 4.3.2 Database

The database used here to store all the sensitive data is Navicat Database. Navicat can hold different types of data, and we use SQL Lite. Navicat is used for high profile enterprise. Thus future expansions can be very easily applied. SQL Lite holds a relational database with easy setup and database administration. Fig 4.3.8 shows values entered while registering both of the user and admin. Below is the hierarchy of the list of database tables used.

id	password	last_login	is_superuser	username	first_name	email	is_staff	is_active	date_joined	last_name	is_voted	id_type
1	pbkdf2_sha256\$15000\$2020-03-18 00:25:42.6f	2020-03-18 00:25:42.6f	1	admin	san	kgc.dream2019@gmail	1	1	2020-03-13 03:18:05.2f	marino	0	(Null)
2	pbkdf2_sha256\$15000\$2020-03-31 11:50:29.6f	2020-03-31 11:50:29.6f	1	kdk1	Lois	lois.walker@hotmail.cc	1	1	2020-03-14 16:03:00.8f	Walker	0	(Null)
3	pbkdf2_sha256\$15000\$2020-03-26 23:53:39.2f	2020-03-26 23:53:39.2f	0	user2	Brenda	brenda.robinson@gmc	0	1	2020-03-14 16:04:20.3f	Robinson	0	(Null)
4	pbkdf2_sha256\$15000\$2020-03-17 01:48:52.1f	2020-03-17 01:48:52.1f	0	user3	Joe	joe.robinson@gmail.cc	0	1	2020-03-14 16:07:16.2f	Robinson	0	(Null)
5	pbkdf2_sha256\$15000\$2020-03-18 02:50:46.6f	2020-03-18 02:50:46.6f	0	benjaminR	benjamin	benjamin.russell@char	0	1	2020-03-17 01:15:57.5f	russell	0	(Null)
6	pbkdf2_sha256\$15000\$2020-03-28 20:42:59.5f	2020-03-28 20:42:59.5f	0	MargaretB	Margaret	margaret.brooks@gm	0	1	2020-03-18 00:24:42.2f	Brooks	0	StateID
7	pbkdf2_sha256\$15000\$2020-03-30 19:42:40.0f	2020-03-30 19:42:40.0f	0	JoanS	Joan	joan.stewart@yahoo.c	0	1	2020-03-28 20:39:46.2f	Stewart	0	Passport
8	pbkdf2_sha256\$15000\$2020-03-30 19:47:52.5f	2020-03-30 19:47:52.5f	0	CarolynH	Carolyn	carolyn.hayes@hotmail	0	1	2020-03-30 17:18:43.7f	Hayes	0	Passport
9	pbkdf2_sha256\$15000\$2020-03-30 19:47:52.5f	2020-03-30 19:47:52.5f	1	ThomasL	Thomas	thomas.lewis@gmail.c	1	1	2020-03-30 19:39:51.4f	Lewis	0	Passport
10	pbkdf2_sha256\$15000\$2020-03-31 15:09:20.1f	2020-03-31 15:09:20.1f	0	RubyR	Ruby	ruby.rogers@gmail.co	0	1	2020-03-30 19:41:48.8f	Rogers	0	Passport
11	pbkdf2_sha256\$15000\$2020-03-31 15:07:32.2f	2020-03-31 15:07:32.2f	1	BrendaB	Brenda	brenda.butler@gmail.c	1	1	2020-03-31 15:03:35.3f	Butler	0	Driving Lic
12	pbkdf2_sha256\$15000\$2020-03-31 16:41:43.2f	2020-03-31 16:41:43.2f	0	LillianB	Lillian	lillian.brown@aol.com	0	1	2020-03-31 15:04:57.0f	Brown	0	Driving Lic
13	pbkdf2_sha256\$15000\$2020-03-31 16:41:43.2f	2020-03-31 16:41:43.2f	1	FrancesY	Frances	frances.young@gmail.u	1	1	2020-03-31 16:38:56.1f	Young	0	Passport
14	pbkdf2_sha256\$15000\$2020-03-31 16:41:43.2f	2020-03-31 16:41:43.2f	0	DianaP	Diana	diana.peterson@hotm	0	0	2020-03-31 16:40:57.5f	Peterson	0	Passport

Figure 4.3.8: User and Admin Registration Table

There are three field which identifies that a user is an admin. They are is\_superuser, is\_staff, and is\_active. If these values are indicated as one, then the user is admin. Is\_active is initially for all users except for admin. When the admin verifies the user and activates them, the value turns into 1. All passwords are encrypted for safety purposes using the SHA algorithm. Superuser and staff are always an admin.

### 4.3.3 Metamask

MetaMask is discussed above is a browser extension containing access to Ethereum Wallets. We start by creating a regular account on MetaMask using a seed phrase. Next, we link this to the Kovan Test network to order to generate a testing network out of the main Ethereum network along with sample Ethers to spend per transaction. These Kovan Ethers have no real-time value associated with it. These are used only to develop and test applications. But transactions can be approved only by the owner of Decentralised Application, who is the Dapp Owner. The Private key generated from the Smart Contract is associated with the address of the account, to transform the regular

account into Dapp owner. Each transaction can be confirmed using the address. Fig 4.3.9 shown below shows the Dapp owner configuration and Fig 4.3.10 shows a sample transaction confirmation along with sample ethers.

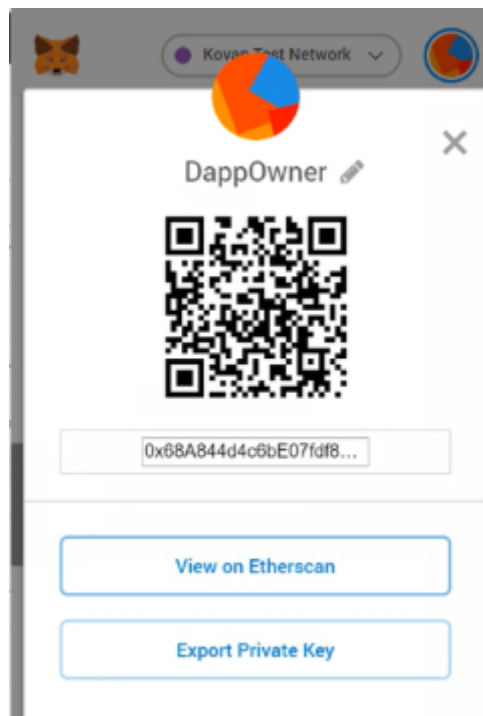


Figure 4.3.9: Dapp Owner Configuration

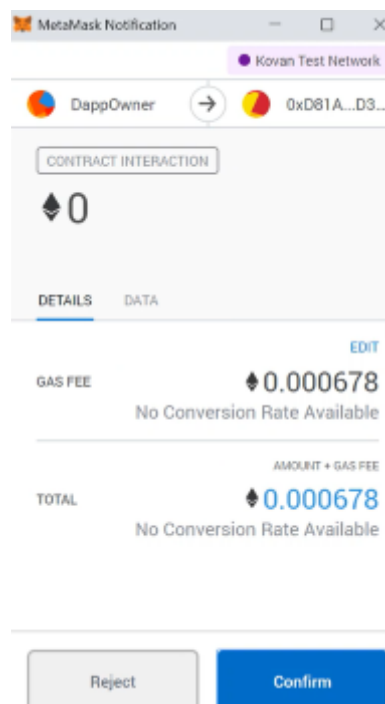


Figure 4.3.10: Transaction Confirmation

To further investigate any transaction and the value of each block of a transaction, Fig 4.3.11 shows the details of transaction. These properties are captured and stored for



# Chapter 5

## Conclusion

After all the advancement of technology, It is clear that the present e-voting system is not a secure voting system. It can easily be attacked and give results in favor of one party or candidate. By building the Ethereum based e-voting system, we have solved some of the fundamental issues of the traditional e-voting system. The traditional system requires a lot of workforce to ensure that the votes have not been tampered.

We have used blockchain in the election process to ensure transparency and anonymity. The main contribution of this system is to ensure one vote per account. The voter can be assured that his/her vote cannot be tampered. By using smart contracts, we have eliminated the need for the need of a third party. All these improvements not only safeguard the voter's vote but also bring a great turnout in election booths as now the voters know that their vote cannot tamper, and it is counted.

# Chapter 6

## Future Work

This proposed system can be applied in another country, although integration can be quite challenging since each country has different laws and election system changes between countries.

The proposed system can be improved for real-life application, by replacing the dummy government database with a real-time high volume database which has details of all the citizens in a country so that this e-voting can be implemented at a national level.

Right now, the registrar can see which candidate the user has voted to, but for better and secure voting, this option can be removed for the registrar. The voter's identity verification can be further enhanced by Incorporating advanced verification methods such as fingerprints scanning tools in personal devices. This e-voting system can be deployed in the Main Ethereum network rather than the current Kovan Test Network for real-life applications. Cloud-based options can also be explored for hosting the e-voting website.

# Bibliography

- [1] Barnes, Andrew, Brake, Christopher, and Perry, Thomas. “Digital Voting with the use of Blockchain Technology”. In: *Team Plymouth Pioneers-Plymouth University* (2016).
- [2] LePennetier, Marine, Thomas, Leigh, and Stonestreet, John. “France drops electronic voting for citizens abroad over cybersecurity fears”. In: *Reuters, June* (2017).
- [3] Lowe, Josh. *Netherlands Abandons Electronic Vote Counting Amid Hacking Fears*. 2017.
- [4] Pawlak, Michał, Poniszewska-Marańda, Aneta, and Kryvinska, Natalia. “Towards the intelligent agents for blockchain e-voting system”. In: *Procedia Computer Science* 141 (2018), pp. 239–246.
- [5] Yi, Haibo. “Securing e-voting based on blockchain in P2P network”. In: *EURASIP Journal on Wireless Communications and Networking* 2019.1 (2019), pp. 1–9.
- [6] Hjálmarsson, Friðrik Þ, Hreiðarsson, Gunnlaugur K, Hamdaqa, Mohammad, and Hjalmtýsson, Gísli. “Blockchain-based e-voting system”. In: *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*. IEEE. 2018, pp. 983–986.
- [7] Meeser, Fernando Lobato. “Decentralized, transparent, trustless voting on the ethereum blockchain”. In: (2017).