

Decentralized Voting System using Blockchain Technology

Submitted in partial fulfilment of the requirements
for the award of
Bachelor of Engineering Degree in
Computer Science and Engineering

by

Gade Yeswanth Adithya Reddy (38110146)
Kakumanu Sai Manikanta (38110489)



SATHYABAMA

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

**SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119. TAMIL NADU.**

MARCH-2022



SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY (DEMED TO BE UNIVERSITY)

(Established under Section 3 of UGC Act, 1956)

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600119

www.sathyabama.ac.in



BONAFIDE CERTIFICATE

This is to certify that this Professional Training Report is the bonafide work of **GADE YESWANTH ADITHYA REDDY (38110146)** and **KAKUMANU SAI MANIKANTA (38110489)** who carried out project entitled “Decentralized Voting System using Blockchain Technology” under my supervision from NOVEMBER 2021 to MARCH 2022

Internal Guide

Dr. S. JANCY, M.Tech., Ph.D.

Head of the Department

Dr. L. LAKSHMANAN., M.E., Ph.D.

Dr.S. VIGNESHWARI, M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **GADE YESWANTH ADITHYA REDDY** and **KAKUMANU SAI MANIKANTA** hereby declare that the Professional Training Report on "**Decentralized Voting System using Blockchain Technology**" done by me under the guidance of **Dr. S. JANCY, M.E., Ph.D.** at Sathyabama institute of science and technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. SASIKALA M.E., Ph.D., Dean, School of Computing, Dr.S. VIGNESHWARI, M.E., Ph.D., and Dr.L. LAKSHMANAN M.E., Ph.D., Heads of the Department, Dept. of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.S. JANCY**, for his valuable guidance, suggestions, and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of **Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Over the way of time, electronic Voting has created as an option for paper polling form casting a ballot to bring down redundancies and irregularities. Because of the various security and protection weaknesses talented throughout the long term, past consequences of e-casting a ballot inside the most recent thirty years show that it has as of now not been a hit. All digital forms of money are fundamentally founded on block chain. It depends on the statute of dissemination and decentralization. The worth in a Blockchain can be anything. Inside the instance of cryptographic forms of money, it takes the importance of money or cash. Blockchain can be utilized on this endeavor to sidestep votes between occasions. The citizens are one party and the applicant who procures the vote is the inverse. without having a controlling focal power body, the square chain can be carried out in an additional a steady way in mass electing casting a ballot works out. vote projecting framework that utilizes an additional a steady, sealed square chain (unchanging to casting a ballot change either by various electors or through any outsider) and is financially savvy. The citizen can make the choice from their Home, this framework will keep away from Proxy projecting and Recasting. After the client made their choice, they can likewise see their made choice.

TABLE OF CONTENTS

Chapter.	TITLE	Page No.
	ABSTRACT	5
	LIST OF FIGURES	8
1	INTRODUCTION	9
	1.1 DOMAIN INTRODUCTION	9
	1.1.1 BLOCKCHAIN	9
	1.2 FEATURES OF BLOCKCHAIN	19
	1.3 PROJECT INTRODUCTION	20
2	LITERATURE SURVEY	21
3	SYSTEM ANALYSIS AND MODULES	24
	3.1. EXISTING SYSTEM	24
	3.2. PROPOSED SYSTEM	24
	3.3. MODULES	25
4	RESULTS AND DISCUSSION	27
	4.1 ARCHITECTURE DIAGRAM	27
5	SYSTEM SPECIFICATION	29
	5.1. LANGUAGE SPECIFICATION	29
	5.1.1 SOLIDITY ETHEREUM	29
	5.1.2 NODEJS	30
	5.2 REQUIREMENT ANALYSIS	31
	5.2.1 HARDWARE ENVIRONMENT	31
	5.2.2 SOFTWARE ENVIRONMENT	32
6	CONCLUSION	33
	REFERENCES	34
	APPENDICES	35

A. SOURCE CODE	35
B. SCREENSHOTS	45

LIST OF FIGURES

Figure No.	FIGURE NAME	Page no.
1.1	BLOCKCHAIN FRAMEWORK	9
1.2	Features of Blockchain	19
4.1	Architecture Diagram	27
4.2	Blockchain Based voting System	27

CHAPTER 1

INTRODUCTION

1.1 DOMAIN INTRODUCTION:

1.1.1 BLOCK CHAIN

The blockchain is an undeniably ingenious invention the brainchild of a person or group of people known by the pseudonym, Satoshi Nakamoto. But since then, it has evolved into something greater, and the main question every single person is asking is: What is Blockchain?

By allowing digital information to be distributed but not copied, blockchain technology created the backbone of a new type of internet. Originally devised for the digital currency, Bitcoin, the tech community is now finding other potential uses for the technology.

Bitcoin has been called “digital gold,” and for a good reason. To date, the total value of the currency is close to \$112 billion US. And blockchains can make other types of digital value. Like the internet you don’t need to know how the blockchain works to use it. However, having a basic knowledge of this new technology shows why it’s considered revolutionary. So, we hope you enjoy this, What Is Blockchain Guide. And if you already know what blockchain is and want to become a blockchain developer please check out our in-depth blockchain tutorial and create your very first blockchain.

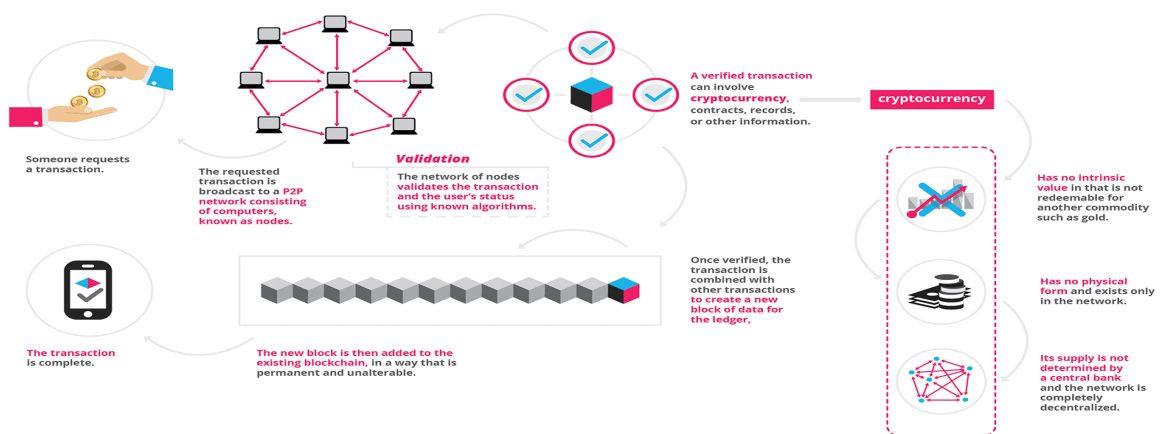


Fig:1.1 Blockchain Framework

What is Blockchain Technology?

“The blockchain is an incorruptible digital ledger of economic transactions that can be programmed to record not just financial transactions but virtually everything of value.”

How Does Blockchain Work?

Picture a spreadsheet that is duplicated thousands of times across a network of computers. Then imagine that this network is designed to regularly update this spreadsheet and you have a basic understanding of the blockchain.

Information held on a blockchain exists as a shared — and continually reconciled — database. This is a way of using the network that has obvious benefits. The blockchain database isn't stored in any single location, meaning the records it keeps are truly public and easily verifiable. No centralized version of this information exists for a hacker to corrupt. Hosted by millions of computers simultaneously, its data is accessible to anyone on the internet.

Blockchain as Google Docs

The traditional way of sharing documents with collaboration is to send a Microsoft Word document to another recipient and ask them to make revisions to it. The problem with that scenario is that you need to wait until receiving a return copy before you can see or make other changes because you are locked out of editing it until the other person is done with it. That's how databases work today. Two owners can't be messing with the same record at once. That's how banks maintain money balances and transfers; they briefly lock access (or decrease the balance) while they make a transfer, then update the other side, then re-open access (or update again). With Google Docs (or Google Sheets), both parties have access to the same document at the same time, and the single version of that document is always visible to both of them. It is like a shared ledger, but it is a shared document. The distributed part comes into play when sharing involves a number of people.

Imagine the number of legal documents that should be used that way. Instead of passing them to each other, losing track of versions, and not being in sync with the other version, why can't all business documents become shared instead of transferred back and forth? So many types of legal contracts would be ideal for that kind of workflow. You don't need a blockchain to share documents, but the shared documents analogy is a powerful one.”

Blockchain Durability and robustness

Blockchain technology is like the internet in that it has a built-in robustness. By storing blocks of information that are identical across its network, the blockchain cannot:

1. Be controlled by any single entity.

2. Has no single point of failure.

Bitcoin was invented in 2008. Since that time, the Bitcoin blockchain has operated without significant disruption. (To date, any of problems associated with Bitcoin have been due to hacking or mismanagement. In other words, these problems come from bad intention and human error, not flaws in the underlying concepts.)

The internet itself has proven to be durable for almost 30 years. It's a track record that bodes well for blockchain technology as it continues to be developed.

“As revolutionary as it sounds, Blockchain truly is a mechanism to bring everyone to the highest degree of accountability. No more missed transactions, human or machine errors, or even an exchange that was not done with the consent of the parties involved. Above anything else, the most critical area where Blockchain helps is to guarantee the validity of a transaction by recording it not only on a main register but a connected distributed system of registers, all of which are connected through a secure validation mechanism.”

Transparent and incorruptible

The blockchain network lives in a state of consensus, one that automatically checks in with itself every ten minutes. A kind of self-auditing ecosystem of a digital value, the network reconciles every transaction that happens in ten-minute intervals. Each group of these transactions is referred to as a “block”. Two important properties result from this:

1. Transparency data is embedded within the network as a whole, by definition it is public.
2. It cannot be corrupted altering any unit of information on the blockchain would mean using a huge amount of computing power to override the entire network.

In theory, this could be possible. In practice, it's unlikely to happen. Taking control of the system to capture Bitcoins, for instance, would also have the effect of destroying their value.

NODE

Computer connected to the blockchain network using a client that performs the task of validating and relaying transactions gets a copy of the blockchain, which gets downloaded automatically upon joining the blockchain network.

Together they create a powerful second-level network, a wholly different vision for how the internet can function.

Every node is an “administrator” of the blockchain, and joins the network voluntarily (in this sense, the network is decentralized). However, each one has an incentive for participating in the network: the chance of winning Bitcoins.

Nodes are said to be “mining” Bitcoin, but the term is something of a misnomer. In fact, each one is competing to win Bitcoins by solving computational puzzles. Bitcoin was the raison d'être of the blockchain as it was originally conceived. It's now recognized to be only the first of many potential applications of the technology.

There are an estimated 1600 Bitcoin-like cryptocurrencies (exchangeable value tokens) already available. As well, a range of other potential adaptations of the original blockchain concept are currently active, or in development.

THE IDEA OF DECENTRALIZATION

By design, the blockchain is a decentralized technology. Anything that happens on it is a function of the network as a whole. Some important implications stem from this. By creating a new way to verify transactions aspects of traditional commerce could become unnecessary. Stock market trades become almost simultaneous on the blockchain, for instance or it could make types of record keeping, like a land registry, fully public. And decentralization is already a reality.

A global network of computers uses blockchain technology to jointly manage the database that records Bitcoin transactions. That is, Bitcoin is managed by its network, and not any one central authority. Decentralization means the network operates on a user-to-user (or peer-to-peer) basis. The forms of mass collaboration this makes possible are just beginning to be investigated.

Who will use the blockchain?

As web infrastructure, you don't need to know about the blockchain for it to be useful in your life. Currently, finance offers the strongest use cases for the technology. International remittances, for instance. The World Bank estimates that over \$430 billion US in money transfers were sent in 2015. And at the moment there is a high demand for blockchain developers.

The blockchain potentially cuts out the middleman for these types of transactions. Personal computing became accessible to the general public with the invention of the Graphical User Interface (GUI), which took the form of a “desktop”. Similarly, the most common GUI devised for the blockchain are the so-called “wallet” applications, which people use to buy things with Bitcoin, and store it along with other cryptocurrencies.

Transactions online are closely connected to the processes of identity verification. It is easy to imagine that wallet apps will transform in the coming years to include other types of identity management.

The Blockchain & Enhanced security

By storing data across its network, the blockchain eliminates the risks that come with data being held centrally.

Its network lacks centralized points of vulnerability that computer hackers can exploit. Today's internet has security problems that are familiar to everyone. We all rely on the "username/password" system to protect our identity and assets online. Blockchain security methods use encryption technology.

The basis for this is the so-called public and private keys. A public key (a long, randomly generated string of numbers) is a users' address on the blockchain. Bitcoins sent across the network gets recorded as belonging to that address. The "private key" is like a password that gives its owner access to their Bitcoin or other digital assets. Store your data on the blockchain and it is incorruptible. This is true, although protecting your digital assets will also require safeguarding of your private key by printing it out, creating what's referred to as a paper wallet.

A SECOND-LEVEL NETWORK

With blockchain technology, the web gains a new layer of functionality. Already, users can transact directly with one another. Bitcoin transactions in 2017 averaged around \$2 billion US per day. With the added security brought by the blockchain new internet business are on track to unbundle the traditional institutions of finance.

Goldman Sachs believes that blockchain technology holds great potential specially to optimize clearing and settlements and could represent global savings of up to \$6bn per year.

THE BLOCKCHAIN A NEW WEB 3.0?

Indeed.com, one of the biggest job portals in the world, published some interesting statistics regarding the rise of Blockchain jobs. It looks like the number of blockchain jobs increased from December 2016 to December 2017 by a staggering 207%. But that's not the end of it. According to the stats, this number has increased by, a scarcely believable 631% since November 2015.

The blockchain gives internet users the ability to create value and authenticates digital information. What will new business applications result?

- *Smart contracts*

Distributed ledgers enable the coding of simple contracts that will execute when specified conditions are met. Ethereum is an open source blockchain project that was built specifically to realize this possibility. Still, in its early stages, Ethereum has the potential to leverage the usefulness of blockchains on a truly world-changing scale.

At the technology's current level of development, smart contracts can be programmed to perform simple functions. For instance, a derivative could be paid out when a financial instrument meets certain benchmark, with the use of blockchain technology and Bitcoin enabling the payout to be automated.

- *The Sharing economy*

With companies like Uber and Airbnb flourishing, the sharing economy is already a proven success. Currently, however, users who want to hail a ride-sharing service have to rely on an intermediary like Uber. By enabling peer-to-peer payments, the blockchain opens the door to direct interaction between parties a truly decentralized sharing economy result.

An early example, Open Bazaar uses the blockchain to create a peer-to-peer eBay. Download the app onto your computing device, and you can transact with OpenBazaar vendors without paying transaction fees. The “no rules” ethos of the protocol means that personal reputation will be even more important to business interactions than it currently is on eBay.

- *Crowdfunding*

Crowdfunding initiatives like Kickstarter and GoFundMe are doing the advance work for the emerging peer-to-peer economy. The popularity of these sites suggests people want to have a direct say in product development. Blockchains take this interest to the next level, potentially creating crowd-sourced venture capital funds.

In 2016, one such experiment, the Ethereum-based DAO (Decentralized Autonomous Organization), raised an astonishing \$200 million USD in just over two months. Participants purchased “DAO tokens” allowing them to vote on smart contract venture capital investments (voting power was proportionate to the number of DAO they were holding). A subsequent hack of project funds proved that the project was launched without proper due diligence, with disastrous consequences. Regardless, the DAO experiment suggests the blockchain has the potential to usher in “a new paradigm of economic cooperation.”

- *Governance*

By making the results fully transparent and publicly accessible, distributed database technology could bring full transparency to elections or any other kind of poll taking. Ethereum-based smart contracts help to automate the process.

The app, Boardroom, enables organizational decision-making to happen on the blockchain. In practice, this means company governance becomes fully transparent and verifiable when managing digital assets, equity, or information.

- *Supply chain auditing*

Consumers increasingly want to know that the ethical claims companies make about their products are real. Distributed ledgers provide an easy way to certify that the backstories of the things we buy are genuine. Transparency comes with blockchain-based timestamping of a date and location on ethical diamonds, for instance that corresponds to a product number.

The UK-based Provenance offers supply chain auditing for a range of consumer goods. Making use of the Ethereum blockchain, a Provenance pilot project ensures that fish sold in Sushi restaurants in Japan has been sustainably harvested by its suppliers in Indonesia.

- *File storage*

Decentralizing file storage on the internet brings clear benefits. Distributing data throughout the network protects files from getting hacked or lost. Inter Planetary File System (IPFS) makes it easy to conceptualize how a distributed web might operate. Like the way a bittorrent moves data around the internet, IPFS gets rid of the need for centralized client-server relationships (i.e., the current web). An internet made up of completely decentralized websites has the potential to speed up file transfer and streaming times. Such an improvement is not only convenient. It's a necessary upgrade to the web's currently overloaded content-delivery systems.

- *Prediction markets*

The crowdsourcing of predictions on event probability is proven to have a high degree of accuracy. Averaging opinions cancels out the unexamined biases that distort judgment. Prediction markets that payout according to event outcomes are already active. Blockchains are a "wisdom of the crowd" technology that will no doubt find other applications in the years to come.

The prediction market application Augur makes share offerings on the outcome of real-world events. Participants can earn money by buying into the correct prediction. The more shares purchased in the correct outcome, the higher the payout will be. With a small commitment of funds (less than a dollar), anyone can ask a question, create a market based on a predicted outcome, and collect half of all transaction fees the market generates.

- *Protection of intellectual property*

As is well known, digital information can be infinitely reproduced and distributed widely thanks to the internet. This has given web users globally a

goldmine of free content. However, copyright holders have not been so lucky, losing control over their intellectual property and suffering financially as a consequence. Smart contracts can protect copyright and automate the sale of creative works online, eliminating the risk of file copying and redistribution.

Mycelia uses the blockchain to create a peer-to-peer music distribution system. Founded by the UK singer-songwriter Imogen Heap, Mycelia enables musicians to sell songs directly to audiences, as well as license samples to producers and divvy up royalties to songwriters and musicians all of these functions being automated by smart contracts. The capacity of blockchains to issue payments in fractional cryptocurrency amounts (micropayments) suggests this use case for the blockchain has a strong chance of success.

- *Internet of Things (IoT)*

What is the IoT? The network-controlled management of certain types of electronic devices for instance, the monitoring of air temperature in a storage facility. Smart contracts make the automation of remote systems management possible. A combination of software, sensors, and the network facilitates an exchange of data between objects and mechanisms. The result increases system efficiency and improves cost monitoring.

The biggest players in manufacturing, tech and telecommunications are all vying for IoT dominance. Think Samsung, IBM, and AT&T. A natural extension of existing infrastructure controlled by incumbents, IoT applications will run the gamut from predictive maintenance of mechanical parts to data analytics, and mass-scale automated systems management.

- *Neighborhood Microgrids*

Blockchain technology enables the buying and selling of the renewable energy generated by neighborhood microgrids. When solar panels make excess energy, Ethereum-based smart contracts automatically redistribute it. Similar types of smart contract automation will have many other applications as the IoT becomes a reality.

Located in Brooklyn, Consensus is one of the foremost companies globally that is developing a range of applications for Ethereum. One project they are partnering on is Transactive Grid, working with the distributed energy outfit, LO3. A prototype project currently up and running uses Ethereum smart contracts to automate the monitoring and redistribution of microgrid energy. This so-called “intelligent grid” is an early example of IoT functionality.

- *Identity management*

There is a definite need for better identity management on the web. The ability to verify your identity is the lynchpin of financial transactions that happen online. However, remedies for the security risks that come with web commerce are imperfect at best. Distributed ledgers offer enhanced methods for proving who you are, along with the possibility to digitize personal documents. Having a secure identity will also be important for online interactions for instance, in the sharing economy. A good reputation, after all, is the most important condition for conducting transactions online.

Developing digital identity standards is proving to be a highly complex process. Technical challenges aside, a universal online identity solution requires cooperation between private entities and government. Add to that the need to navigate legal systems in different countries and the problem becomes exponentially difficult. E-Commerce on the internet currently relies on the SSL certificate (the little green lock) for secure transactions on the web. Netki is a startup that aspires to create an SSL standard for the blockchain. Having recently announced a \$3.5 million seed round, Netki expects a product launch in early 2017.

- *AML and KYC*

Anti-money laundering (AML) and know your customer (KYC) practices have a strong potential for being adapted to the blockchain. Currently, financial institutions must perform a labor-intensive multi-step process for each new customer. KYC costs could be reduced through cross-institution client verification, and at the same time increase monitoring and analysis effectiveness. StartupPolyconic has an AML/KYC solution that involves analyzing transactions. Those transactions identified as being suspicious are forwarded on to compliance officers. Another startup trade is developing an application called Trust in Motion (TiM). Characterized as an “Instagram for KYC”, TiM allows customers to take a snapshot of key documents (passport, utility bill, etc.). Once verified by the bank, this data is cryptographically stored on the blockchain.

- *Data management*

Today, in exchange for their personal data people can use social media platforms like Facebook for free. In future, users will have the ability to manage and sell the data their online activity generates. Because it can be easily

distributed in small fractional amounts, Bitcoin, or something like it will most likely be the currency that gets used for this type of transaction. The MIT project Enigma understands that user privacy is the key precondition for creating of a personal data marketplace. Enigma uses cryptographic techniques to allow individual data sets to be split between nodes, and at the same time run bulk computations over the data group as a whole. Fragmenting the data also makes Enigma scalable (unlike those blockchain solutions where data gets replicated on every node). A Beta launch is promised within the next six months.

- *Land title registration*

As Publicly accessible ledgers, blockchains can make all kinds of record-keeping more efficient. Property titles are a case in point. They tend to be susceptible to fraud, as well as costly and labour intensive to administer.

A number of countries are undertaking blockchain-based land registry projects. Honduras was the first government to announce such an initiative in 2015, although the current status of that project is unclear. This year, the Republic of Georgia cemented a deal with the Bitfury Group to develop a blockchain system for property titles. Reportedly, Hernando de Soto, the high-profile economist and property rights advocate, will be advising on the project. Most recently, Sweden announced it was experimenting with a blockchain application for property titles.

- *Stock trading*

The potential for added efficiency in share settlement makes a strong use case for blockchains in stock trading. When executed peer-to-peer, trade confirmations become almost instantaneous (as opposed to taking three days for clearance). Potentially, this means intermediaries such as the clearing house, auditors and custodians get removed from the process. Numerous stock and commodities exchanges are prototyping blockchain applications for the services they offer, including the ASX (Australian Securities Exchange), the Deutsche Börse (Frankfurt's stock exchange) and the JPX (Japan Exchange Group). Most high profile because the acknowledged first mover in the area, is the Nasdaq's Linq, a platform for private market trading (typically between pre-IPO startups and investors). A partnership with the blockchain tech company Chain, Linq announced the completion of its first share trade in 2015. More recently, Nasdaq announced the development of a trial blockchain project for proxy voting on the Estonian Stock Market.

1.2 FEATURES OF BLOCKCHAIN

The main features of blockchain technology are:



Fig: 1.2 Features of Blockchain

1.2.1 Increased Capacity

It increases the capacity of the whole network, since there are lot of computers working together which offers a great power in total then few of the devices where the things are centralized.

1.2.2 Better Security

There is no chance of shutting down of the system since Blockchain technology has a better security. The main reason is that the Blockchain network is secured by a number of computers called nodes and transaction on this network is confirmed by these nodes.

1.2.3 Immutability

One of the main values of Blockchain is creating immutable ledgers. The databases which are centralized are most likely to get hacked and to keep these databases secure they require trust in the third party.

1.2.4 Faster Settlement

A lot of settlement time is required, which usually takes days to proceed may cause the traditional banking systems to slow down. Because of this reason these banking institutes needs to upgrade their banking systems. With the help of Blockchain technology we can solve this problem as it can settle money transfer

at really fast speeds. It provides convenience to the consumers since it saves a lot of time and money from these institutions.

1.2.5 Decentralized System

The decentralized technology gives the power to store assets in a network which further accesses by the means of the internet. An asset can be like a document, a contract, etc. By this the owner can have a direct control over his account which gives the owner a power to transfer his assets to anyone he wants to by the means of a Key that is linked to his account.

1.2.6 Minting

By help of the Blockchain we can solve the various ways of minting a problem of manipulation. The role of the miner is to create a block for many new methods which has also been introduced as proof of work is the approach by which an individual can prove that he is engaged in a significant amount of computation work.

1.3PROJECT INTRODUCTION:

Electronic voting systems have been the subject of active research for decades, with the goal to minimize the cost of running an election, while ensuring the election integrity by fulfilling the security, privacy, and compliance requirements [1]. Replacing the traditional pen and paper scheme with a new election system has the potential to limit fraud while making the voting process traceable and verifiable [2]. Blockchain is a distributed, immutable, incontrovertible, public ledger. This new technology has three main features: (i) Immutability: Any proposed “new block” to the ledger must reference the previous version of the ledger.

This creates an immutable chain, which is where the blockchain gets its name from, and prevents tampering with the integrity of the previous entries. (ii) Verifiability: The ledger is decentralized, replicated, and distributed over multiple locations. This ensures high availability (by eliminating a single point of failure) and provides third-party verifiability as all nodes maintain the consensus version of the ledger. (iii) Distributed Consensus: A distributed consensus protocol to determine who can append the next new transaction to the ledger. A majority of the network nodes must reach a consensus before any new proposed block of entries becomes a permanent part of the ledger.

CHAPTER 2

LITERATURE SURVEY

Paper 1: Performance Assessment of An Imperceptible and Robust Secured E-Voting Model

Author: Olaniyi Olayemi M, Arulogun Oladiran T, Omidiora Elijah O, Okediran Oladotun

Abstract:

In this paper, we present the performance assessment of an imperceptible and robust secured stegano-cryptographic model of electronic voting. The Performance analysis was achieved based on the degree to which the model meets the generic and functional requirements of secured evoting system: authentication, integrity, confidentiality, and verifiability as well as other functional security requirements of a secured voting using fivepoint psychometric analysis. The result of the quantitative evaluation of the model assert that the model possessed capacity to guarantee and validate voters for who they said they are, guarantees the integrity of elections, ensures privacy of the voters, guarantees the confidentiality of the vote, and provide mechanism for fraud detection after the electioneering process in developing country where digital divide is significant.

Paper 2: Towards a Fraud Prevention E-Voting System

Author:

Dr. Magdi Amer ,
Dr. Hazem El-Gendy

Abstract:

Election falsification is one of the biggest problems facing third world countries as well as developed countries with respect to cost and time. In this paper, the guidelines for building a legally binding fraud-proof Electronic-Voting are presented. Also, the limitations are discussed.

Paper 3:Secure E-Voting System

Author: Monira Monir Haroon Khater

Abstract:

With the rapid growth of the internet and technologies, E- voting appears to be a reasonable alternative to conventional elections. Various Information Security and Privacy Technologies including cryptography, steganography, and combination of both have been formulated in literatures to make democratic decision through e-voting systems to be fair and credible.

In practice, different data cryptographic standards like Data Encryption Standard (DES), and Advanced Encryption Standard (AES), Rivest, Sharim and Adleman (RSA) is needed to ensure the security of the votes and maintain the confidentiality and integrity.

Homomorphic encryption scheme is used to encrypt all the votes and perform the calculation of the votes without revealing any information about them.

Current research focuses on designing and building “electronic voting protocols” such as zero knowledge authentication protocol, based on Diffie-Hellman key exchange algorithm, to ensure a mutual authentication between the election authority server and the voters. This thesis proposes a new protocol that covers and maintains the security requirements which are: (authentication, privacy, integrity, Anonymity, and non-repetition) of the voting process.

Paper 4: Online voting application using Ethereum blockchain

Author:Shalini Shukla, Shashank D O, Thasmiya A N, Dr. Mamatha H R

Abstract:

Voting is an important part of the administration of a country. Votes are still being carried out by physically going to voting booths. This process doesn't guarantee security and cases of tampering has been observed. This paper aims at removing these issues in the voting process by making it online and using the technology, Blockchain. Blockchain uses encryption and hashing to make every vote secure. In this case, one vote is considered as a transaction. A peer-to-peer network is created to create a private blockchain that share this distributed ledger having voting transaction. The application is designed in such a way so that the intricacy of the underlying architecture is hidden from the user. Each voter is uniquely identified by Government approved Aadhar number. The application makes use of this number to make sure that each voter gets only one chance to vote. When the vote gets submitted as a transaction then all the peers get synch up. Since each peer is associated with a public and private key the votes are encrypted and hashed and added to the blockchain to increase security and form a chain of blocks. Votes cannot be tracked back to the voter. In this paper, a peer-to-peer network is created having minimum three peers. Since voting is made online, it is expected that this paper will increase the voter turnouts. The scalability of the blockchain application depends on the secondary memory limit of the peer.

Paper 5: Secret Suffrage in Remote Electronic Voting Systems

Author: Adrià Rodríguez-Pérez

Abstract:

Can the principle of secret suffrage be ensured when voters are offered the possibility to cast their votes using internet voting? With the steady introduction of different forms of remote electronic voting since 2000, it has become apparent that internet voting fails at providing the privacy guarantees offered by traditional paper-based voting systems. Against this assumption, the current proposal suggests reviewing the traditional configuration of the principle of vote secrecy. With this in mind, the proposal will: (1) assess current accepted standards on voters' anonymity for traditional and internet-based voting systems; (2) evaluate the core elements of lawful relaxations to the principle of secret suffrage, and especially those traditionally associated to different forms of remote voting, and assess whether they can be applied to internet voting; and (3) study how current technical developments in the field of elections (and

more broadly, in the field of e-governance and e-democracy) may result in further relaxations of the principle of secret suffrage in the future. Overall, the goal of the proposal is to approach the principle of secret suffrage against the specificities of internet voting and, instead of evaluating electronic voting systems using traditional standards for voters' privacy and anonymity, evaluate how specific proposals aimed at ensuring voters' secrecy in internet voting comply with the very end that the principle of secret suffrage is aimed at protecting, namely: voters' freedom.

Paper 6: The Council of Europe and e-voting

Author: Robert Stein, Gregor Wenda

Abstract:

When the Council of Europe started to deal with the subject of electronic voting in 2002, the impact of its work was not foreseeable. What followed, however, was basically a "success story": The Recommendation on legal, operational, and technical standards for e-voting (Rec(2004)11), which was adopted by the Council of Ministers on 30 September 2004, has been the most relevant international document and reference regarding e-voting for a decade. Since 2010, the role of the Council of Europe with regard to e-voting has shrunk. Nevertheless, various Member States expressed the desire to further review the Recommendation in the forthcoming years. Following an informal experts' meeting in Vienna on 19 December 2013, the Committee of Ministers was confronted with the suggestion to formally update the Recommendation in order to keep up with the latest technical, legal, and political developments. The forthcoming Review Meeting on 28 October 2014 may help set the course for future e-voting activities of the Council of Europe.

Paper 7: Enhancing Electronic Voting Machines on the Example of Bingo Voting

Author: Jens-Matthias Bohli, Christian Henrich, Carmen Kempka, Jörn Müller-Quade, and Stefan Röhrich

Abstract:

The main purpose of cryptographic voting schemes is to provide transparency while protecting ballot secrecy and to enable a fast tally. In this paper, we address three major issues of cryptographic voting schemes. First, we discuss the problem of secrecy and coercion resistance in the situation of a corrupted voting machine. While hard to obtain in general, we propose and analyze a novel approach that uses encapsulated design and minimizes the information that can compromise ballot secrecy. The second issue we address is the assumption that an adversary does not know which receipts are checked and the problem of receipt stealing. Many voting schemes with receipts share this vulnerability. We provide a solution that increases protection of each vote, and which can be generalized for voting schemes that use computers to form the receipt. The last issue discussed in this paper is the question of how an election can be contested. For this, an error or a manipulation must not only be detected but also proven. While the problems and solutions are described for Bingo Voting, we argue that the problems are shared by many cryptographic voting schemes and that the solutions presented in this work give insight in the prerequisites needed for a secure election.

CHAPTER 3

SYSTEM ANALYSIS AND MODULES

3.1 EXISTING SYSTEM:

From traditional period vote fraudulent is happening on every century there are some securities is implementing but in parallel way fraudulent is increasing.

DISADVANTAGES:

- Fraudulent made on voting server
- Security level is less
- Uses Centralized Database.

3.2 PROPOSED SYSTEM:

Electronic voting system based on Blockchain that addresses some of the limitations in existing systems and evaluates some of the popular blockchain frameworks for the purpose of constructing a blockchain-based e-voting system, this system will avoid proxy casting and recasting.

ADVANTAGES:

- Security for vote using block chain
- Effective hashing techniques are implemented to ensure the security of the data.
- Stores Data in a Decentralized Way.
- The voter can cast their vote from home itself.
- Immutability.
- User Friendly.

3.3 MODULES

A modular design reduces complexity, facilitates change (a critical aspect of software maintainability), and results in easier implementation by encouraging parallel development of different parts of the system. Software with effective modularity is easier to develop because functions may be compartmentalized, and interfaces are simplified. Software architecture embodies modularity that is software is divided into separately named and addressable components called modules that are integrated to satisfy problem requirements.

Modularity is the single attribute of software that allows a program to be intellectually manageable. The five important criteria that enable us to evaluate a design method with respect to its ability to define an effective modular design are: Modular decomposability, Modular Comprehensibility, Modular Understandability, Modular continuity, Modular Protection.

The following are the modules of the project, which is planned in aid to complete the project with respect to the proposed system, while overcoming existing system and also providing the support for the future enhancement.

MODULE LIST:

1. User Registration
2. Candidate registration
3. Block chain formation
4. Voting Results

USER REGISTRATION

Once the User creates a MetaMask Account in order to Cast their Vote. Using the MetaMask's unique account id, they can cast their vote. A user Can only Cast their Vote Once.

CANDIDATE REGISTRATION:

In this module admin will register the candidate using their Aadhar number. Candidate registration will be made using name and constituency of that candidate and

their Party. If user candidate provide improper information system will discard those registration process.

BLOCKCHAIN FORMATION:

A block is a container data structure. The average size of a block seems to be 1MB. Here every certificate number will be created as a block. For every block a hash code will generate for security. Here every voting information will be stored on block chain. If we store the information on blockchain it is more secured, and every block is created based on constituency.

VOTING RESULTS:

As soon as the Voter pools their Vote they can Immediately View the Results. And also, the user will not be able to Cast the Vote again. This Removes the Double Voting Problem. Voting will be Handy in the Website. Easy to use.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 ARCHITECTURE DIAGRAM:

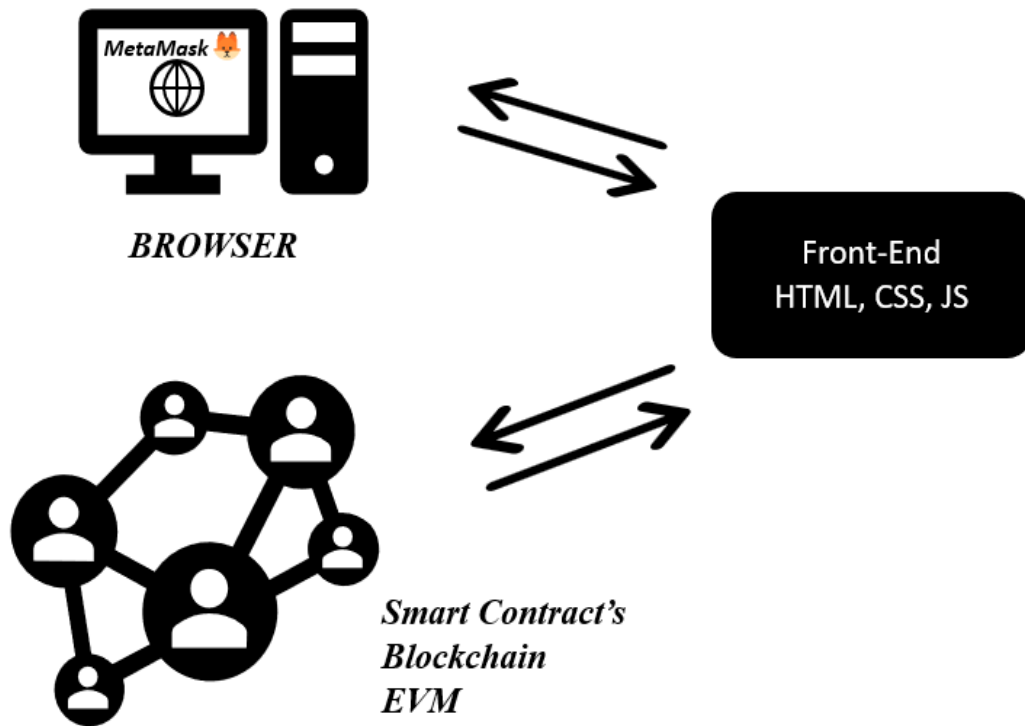


Fig4.1: Architecture Diagram

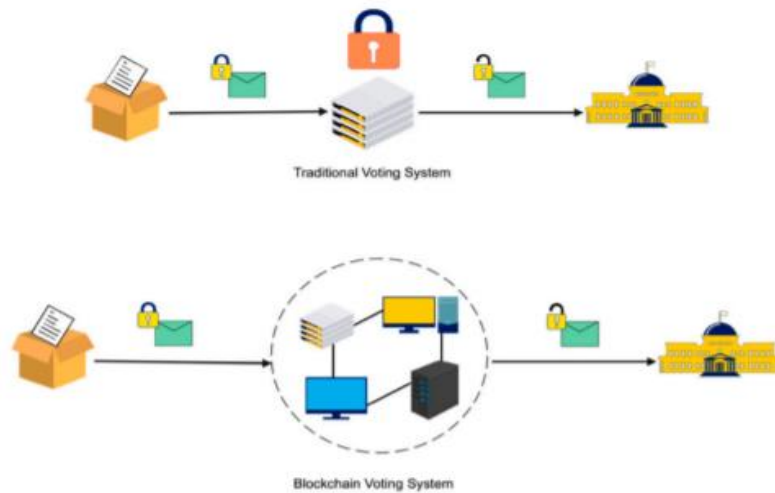


Fig 4.2: Blockchain Based Voting System

The overall architecture describes about, First the admin will register the user and candidate details with their Aadhar number on user interface. Those details will be stored on data base.

From user side they have a mobile application on their mobile. They will login into application and give their Aadhar number on that application after system will send OTP to the registered mobile number. Users have to share that OTP for confirmation of vote. Finally vote details will create as a block and stored on block chain.

CHAPTER 5

SYSTEM SPECIFICATION

5.1 LANGUAGE SPECIFICATION

5.1.1 SOLIDITY ETHEREUM:

Solidity is a brand-new programming language native to Ethereum, the second largest cryptocurrency by market capitalization, initially released in 2015. Ethereum is not only a cryptocurrency capable of storing value or making payments, but a fully-fledged platform for creating what's known as a smart contract.

“Solidity is known as a contract-based, high-level programming language. This platform has similar syntax to the scripting language of JavaScript. Solidity as a programming language is made to enhance the Ethereum Virtual Machine. Solidity is statically typed scripting language which does the process of verifying and enforcing the constraints at compile-time as opposed to run-time.”

This typed programming languages will help and do the checking at run-time as opposed to Compile-time. This platform also supports inheritance in object-oriented programming, inheritance enables new objects to take on the properties of existing objects. A class that is used as the basis for inheritance is called a superclass or base class. A class that inherits from a superclass is called a subclass or derived class. As you will see, it is possible to create contracts for voting, crowd funding, blind auctions, multi-signature wallets and more.

Ethereum Language

The source code in Ethereum language is written in Solidity version 0.4.0 which does not break functionality. The first line of code `pragma solidity contract` does not suddenly behave differently with a new compiler version. According to the docs, the keyword `pragma` is called that way because, in general, `pragmas` are instructions for the compiler about how to treat the source code. `Pragma once` is a preprocessor directive that tells the compiler to include the source code once in a single compilation.

The line `unit stored data` declares a state variable called `stored data` of type `unit`. The functions `set` and `can` be used to modify or retrieve the value of the variable.

Believe it or not, this is an Ethereum contract, albeit a rather simple deal. The functions of this contract allow you or anyone to store a single number that is accessible by anyone in the world without a feasible way to prevent you from publishing this figure. Anyone could just call set again with a different value and overwrite your number, but the number will still be stored in the history of the blockchain. The line unit stored data declares a state variable called stored data of type unit.

The example above is the simplest form of an Ethereum contract. Like learning mathematics, this contract is equivalent to adding and subtracting. Obviously, mathematics is more complicated than just adding and subtracting, but at the fundamental level, all of the mathematics can be broken down into addition/subtraction operations. For example, 5×5 is the same as adding 5 five times. A similar thought process can be used in programming to build programs of higher complexity.

5.1.2 NODEJS

Node.js is a server-side platform built on Google Chrome's JavaScript Engine (V8 Engine). Node.js was developed by Ryan Dahl in 2009 and its latest version is v15.11.0. Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux.

Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

Features of Node.js

Following are some of the important features that make Node.js the first choice of software architects.

- **Asynchronous and Event Driven** – All APIs of Node.js library is asynchronous, that is, non-blocking. It essentially means a Node.js based server never waits for an API to return data. The server moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call.

- **Very Fast** – Being built on Google Chrome's V8 JavaScript Engine, Node.js library is very fast in code execution.
- **Single Threaded but Highly Scalable** – Node.js uses a single threaded model with event looping. Event mechanism helps the server to respond in a non-blocking way and makes the server highly scalable as opposed to traditional servers which create limited threads to handle requests. Node.js uses a single threaded program and the same program can provide service to a much larger number of requests than traditional servers like Apache HTTP Server.
- **No Buffering** – Node.js applications never buffer any data. These applications simply output the data in chunks.
- **License** – Node.js is released under the MIT license.

5.2 REQUIREMENT ANALYSIS

Requirement analysis determines the requirements of a new system. This project analyses on product and resource requirement, which is required for this successful system. The product requirement includes input and output requirements it gives the wants in term of input to produce the required output. The resource requirements give in brief about the software and hardware that are needed to achieve the required functionality.

5.2.1 Hardware Environment

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the systems do and not how it should be implemented.

- Hard disk : 500 GB
- RAM : 4 GB
- PROCESSOR : Intel® Core™ i5 above/ Ryzen 5 above

5.2.2 Software Environment

The software requirements are the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks, and tracking the team's and tracking the team's progress throughout the development activity.

- Operating system : Windows 7/8.1 / Linux
- Languages : Solidity, go or Nodejs
- Web Technologies : HTML, CSS, JS, JSON
- IDE : Visual Studio Code

CHAPTER 6

CONCLUSION

We introduced a blockchain-based electronic-voting system that employs smart contracts to enable secure and cost-effective presidential races yet maintaining voter security in this endeavor. We have demonstrated that the blockchain innovation provides an extra chance to overcome the limitations and reception barriers of e-voting frameworks, ensuring the security and trustworthiness of political decisions and laying the groundwork for transparency. It is possible to transmit numerous trades per second onto an Ethereum private blockchain by employing each portion of the smart contract to support the heaps on the blockchain. A few further steps would be necessary for countries of greater size to aid with higher throughput of exchanges per second.

Because of the block-simplicity, chain's more decisions can be reviewed and comprehended. These lays claim is a part of a voting framework's requirements. These characteristics are seen in decentralized structures, and they can help bring more fair cycles to choices, especially when it comes to coordinating political based on race frameworks. A possible arrangement is to base e-voting on blockchain innovation in order for it to become more transparent, easy, and openly auditable. This project looks at the capabilities of blockchain technology and how it may be used in an e-voting system. Nobody will wish to destroy the blockchain since it will be freely irrefutable and seized.

REFERENCES:

- [1] Sos.ca.gov. (2007). *Top-to-Bottom Review* | California Secretary of State. Available at: <http://www.sos.ca.gov/elections/voting-systems/oversight/top-bottom-review/>.
- [2] Nicholas Weaver. (2016). *Secure the Vote Today* Available at: <https://www.lawfareblog.com/secure-vote-today>.
- [3] TechCrunch, (2018). *Liquid democracy uses blockchain to fix politics, and now you can vote for it*. Available at: <https://techcrunch.com/2018/02/24/liquid-democracy-uses-blockchain/>
- [4] Ajit Kulkarni, (2018), "How to Choose Between Public and Permissioned Blockchain for Your Project", Chronicled, 2018.
- [5] "What Are Smart Contracts? A Beginner's Guide to Smart Contracts", Blockgeeks, 2016. Available at: <https://blockgeeks.com/guides/smart-contracts/>
- [6] Salanfe, *Setup your own private Proof-of-Authority Ethereum network with Geth*, Hacker Noon, 2018. Available at: <https://tinyurl.com/y7g362kd>.
- [7] Geth.ethereum.org. (2018). *Go Ethereum*. Available at: <https://geth.ethereum.org/>
- [8] Vitalik Buterin. (2015). *Ethereum White Paper* Available at: <https://github.com/ethereum/wiki/wiki/White-Paper>.
- [9] Ethdocs.org. (2018). *What is Ethereum? — Ethereum Homestead 0.1 documentation*. [online] Available at: <http://ethdocs.org/en/latest/introduction/what-is-ethereum.html>
- [10] Agora (2017). *Agora: Bringing our voting systems into the 21st century* Available at: https://agora.vote/Agora_Whitepaper_v0.1.pdf
- [11] Patrick McCorry, Siamak F. Shahandashti and Feng Hao. (2017). *Smart Contract for Boardroom Voting with Maximum Voter Privacy* Available at: <https://eprint.iacr.org/2017/110.pdf>.

APPENDIX

SOURCE CODE AND SCREENSHOT

Code:

Election.sol

```
pragma solidity 0.4.25;

contract Election {
    struct Candidate {
        uint id;
        string name;
        string PartyName;
        uint voteCount;
    }

    mapping(address => bool) public voters;

    mapping(uint => Candidate) public candidates;

    uint public candidatesCount;

    event votedEvent (
        uint indexed _candidateId
    );

    constructor () public {
        addCandidate("Sivva","BJP");
        addCandidate("Mani","TRS");
        addCandidate("Krishna","Congress");
        addCandidate("Akhil","YCP");
        addCandidate("NOTA","NOTA");
    }

    function addCandidate (string _name, string _PartyName) private {
        candidatesCount ++;
        candidates[candidatesCount] = Candidate(candidatesCount, _name, _PartyName,
0);
    }

    function vote (uint _candidateId) public {
```

```

        require(!voters[msg.sender]);

        require(_candidateId > 0 && _candidateId <= candidatesCount);

        voters[msg.sender] = true;

        candidates[_candidateId].voteCount ++;

        emit votedEvent(_candidateId);
    }
}

```

Migrations.sol

```

pragma solidity ^0.4.2;

contract Migrations {
    address public owner;
    uint public last_completed_migration;

    modifier restricted() {
        if (msg.sender == owner) _;
    }

    constructor () public {
        owner = msg.sender;
    }

    function setCompleted(uint completed) public restricted {
        last_completed_migration = completed;
    }

    function upgrade(address new_address) public restricted {
        Migrations upgraded = Migrations(new_address);
        upgraded.setCompleted(last_completed_migration);
    }
}

```

```
}  
}
```

Election.js

```
var Election = artifacts.require("./Election.sol");
```

```
contract("Election", function(accounts) {  
  var electionInstance;
```

```
  it("initializes with two candidates", function() {  
    return Election.deployed().then(function(instance) {  
      return instance.candidatesCount();  
    }).then(function(count) {  
      assert.equal(count, 2);  
    });  
  });
```

```
  it("it initializes the candidates with the correct values", function() {  
    return Election.deployed().then(function(instance) {  
      electionInstance = instance;  
      return electionInstance.candidates(1);  
    }).then(function(candidate) {  
      assert.equal(candidate[0], 1, "contains the correct id");  
      assert.equal(candidate[1], "Candidate 1", "contains the correct name");  
      assert.equal(candidate[2], 0, "contains the correct votes count");  
      return electionInstance.candidates(2);  
    }).then(function(candidate) {  
      assert.equal(candidate[0], 2, "contains the correct id");  
      assert.equal(candidate[1], "Candidate 2", "contains the correct name");  
      assert.equal(candidate[2], 0, "contains the correct votes count");  
    });  
  });
```

```
});
```

```
it("allows a voter to cast a vote", function() {  
  return Election.deployed().then(function(instance) {  
    electionInstance = instance;  
    candidateId = 1;  
    return electionInstance.vote(candidateId, { from: accounts[0] });  
  }).then(function(receipt) {  
    assert.equal(receipt.logs.length, 1, "an event was triggered");  
    assert.equal(receipt.logs[0].event, "votedEvent", "the event type is correct");  
    assert.equal(receipt.logs[0].args._candidateId.toNumber(), candidateId, "the  
candidate id is correct");  
    return electionInstance.voters(accounts[0]);  
  }).then(function(voted) {  
    assert(voted, "the voter was marked as voted");  
    return electionInstance.candidates(candidateId);  
  }).then(function(candidate) {  
    var voteCount = candidate[2];  
    assert.equal(voteCount, 1, "increments the candidate's vote count");  
  })  
});
```

```
it("throws an exception for invalid candidates", function() {  
  return Election.deployed().then(function(instance) {  
    electionInstance = instance;  
    return electionInstance.vote(99, { from: accounts[1] })  
  }).then(assert.fail).catch(function(error) {  
    assert(error.message.indexOf('revert') >= 0, "error message must contain revert");  
    return electionInstance.candidates(1);  
  }).then(function(candidate1) {  
    var voteCount = candidate1[2];
```

```

    assert.equal(voteCount, 1, "candidate 1 did not receive any votes");
    return electionInstance.candidates(2);
  }).then(function(candidate2) {
    var voteCount = candidate2[2];
    assert.equal(voteCount, 0, "candidate 2 did not receive any votes");
  });
});

```

```

it("throws an exception for double voting", function() {
  return Election.deployed().then(function(instance) {
    electionInstance = instance;
    candidateId = 2;
    electionInstance.vote(candidateId, { from: accounts[1] });
    return electionInstance.candidates(candidateId);
  }).then(function(candidate) {
    var voteCount = candidate[2];
    assert.equal(voteCount, 1, "accepts first vote");
    // Try to vote again
    return electionInstance.vote(candidateId, { from: accounts[1] });
  }).then(assert.fail).catch(function(error) {
    assert(error.message.indexOf('revert') >= 0, "error message must contain revert");
    return electionInstance.candidates(1);
  }).then(function(candidate1) {
    var voteCount = candidate1[2];
    assert.equal(voteCount, 1, "candidate 1 did not receive any votes");
    return electionInstance.candidates(2);
  }).then(function(candidate2) {
    var voteCount = candidate2[2];
    assert.equal(voteCount, 1, "candidate 2 did not receive any votes");
  });
});

```

```
});
```

App.js

```
App = {
```

```
  web3Provider: null,
```

```
  contracts: {},
```

```
  account: '0x0',
```

```
  hasVoted: false,
```

```
  init: function() {
```

```
    return App.initWeb3();
```

```
  },
```

```
  initWeb3: function() {
```

```
    if (typeof web3 !== 'undefined') {
```

```
      const ethEnabled = () => {
```

```
        if (window.ethereum) {
```

```
          window.web3 = new Web3(window.ethereum);
```

```
          window.ethereum.enable();
```

```
          return true;
```

```
        }
```

```
        return false;
```

```
      }
```

```
      if (!ethEnabled()) {
```

```
        alert("Please install an Ethereum-compatible browser or extension like MetaMask
```

```
to use this dApp!");
```

```
      }
```

```
      web3 = window.web3;
```

```
      App.web3Provider = web3.currentProvider;
```

```
    } else {
```

```
      App.web3Provider = new Web3.providers.HttpProvider('http://localhost:7545');
```

```

    web3 = new Web3(App.web3Provider);
  }
  return App.initContract();
},

initContract: function() {
  $.getJSON("Election.json", function(election) {
    App.contracts.Election = TruffleContract(election);
    App.contracts.Election.setProvider(App.web3Provider);

    App.listenForEvents();

    return App.render();
  });
},

listenForEvents: function() {
  App.contracts.Election.deployed().then(function(instance) {

    instance.votedEvent({}, {
      fromBlock: 0,
      toBlock: 'latest'
    }).watch(function(error, event) {
      console.log("event triggered", event)
      App.render();
    });
  });
},

render: function() {
  var electionInstance;

```

```

var loader = $("#loader");
var content = $("#content");

loader.show();
content.hide();

web3.eth.getCoinbase(function(err, account) {
  if (err === null) {
    App.account = account;
    $("#accountAddress").html("Your Account: " + account);
  }
});

App.contracts.Election.deployed().then(function(instance) {
  electionInstance = instance;
  return electionInstance.candidatesCount();
}).then(function(candidatesCount) {
  var candidatesResults = $("#candidatesResults");
  candidatesResults.empty();

  var candidatesSelect = $('#candidatesSelect');
  candidatesSelect.empty();

  for (var i = 1; i <= candidatesCount; i++) {
    electionInstance.candidates(i).then(function(candidate) {
      var id = candidate[0];
      var name = candidate[1];
      var partyName = candidate[2];
      var voteCount = candidate[3];

```

```

        var candidateTemplate = "<tr><th>" + id + "</th><td>" + name + "</td><td>" +
partyName + "</td><td>" + voteCount + "</td></tr>"
        candidatesResults.append(candidateTemplate);

```

```

        var candidateOption = "<option value='" + id + "' >" + name + "' + " - " + "' +
partyName + "</ option>"

```

```

        candidatesSelect.append(candidateOption);
    });
}
return electionInstance.voters(App.account);
}).then(function(hasVoted) {

```

```

    if(hasVoted) {
        $('form').hide();
    }
    loader.hide();
    content.show();
}).catch(function(error) {
    console.warn(error);
});
},

```

```

castVote: function() {
    var candidateId = $('#candidatesSelect').val();
    App.contracts.Election.deployed().then(function(instance) {
        return instance.vote(candidateId, { from: App.account });
    }).then(function(result) {

```

```

        $('#content').hide();
        $('#loader').show();

```

```
    }).catch(function(err) {  
        console.error(err);  
    });  
}  
};
```

```
$(function() {  
    $(window).load(function() {  
        App.init();  
    });  
});
```

Screenshots:

```
yeswanth@yeswanth:~/Desktop/election$ truffle migrate --reset

Compiling your contracts...
=====
✓ Fetching solc version list from solc-bin. Attempt #1
✓ Fetching solc version list from solc-bin. Attempt #1
> Everything is up to date, there is nothing to compile.

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

1_initial_migration.js
=====

Replacing 'Migrations'
-----
> transaction hash: 0x38f0aa1447079892a514d2f3e4b5388e607d1103c90acb4a4d6a1f88bb661f12
> Blocks: 0        Seconds: 0
> contract address: 0x18aE1b223f5eD23b05CdED43999a869c85C21452
> block number:     1
> block timestamp:  1643015832
> account:          0x37A565F8306E64529B0699eE856505D258d476B7
> balance:          99.99522812
> gas used:         238594 (0x3a402)
> gas price:        20 gwei
> value sent:       0 ETH
> total cost:       0.00477188 ETH

> Saving migration to chain.
> Saving artifacts
-----
> Total cost:       0.00477188 ETH
```

```
2_deploy_contracts.js
=====

Replacing 'Election'
-----
> transaction hash: 0x6c6dea751b0f04c4e8257b542f9014208036dee3f24702bed83627ef3654a9eb
> Blocks: 0        Seconds: 0
> contract address: 0x558772f3809D2B1b1334EC2967eBe6F466529c60
> block number:     3
> block timestamp:  1643015835
> account:          0x37A565F8306E64529B0699eE856505D258d476B7
> balance:          99.98655466
> gas used:         391325 (0x5f89d)
> gas price:        20 gwei
> value sent:       0 ETH
> total cost:       0.0078265 ETH

> Saving migration to chain.
> Saving artifacts
-----
> Total cost:       0.0078265 ETH

Summary
=====
> Total deployments: 2
> Final cost:       0.01259838 ETH
```

ubuntu [Running] - Oracle VM VirtualBox

FileMachineViewInputDevicesHelp

ActivitiesGanacheJan 24 18:13

ACCOUNTSBLOCKSTRANSACTIONSCONTRACTSEVENTSLOGS

SEARCH FOR BLOCK NUMBERS OR TX HASHES

CURRENT BLOCK8GAS PRICE2000000000GAS LIMIT6721975HARDFORKMURGLACHERNETWORK ID5777RPC ENDPOINTHTTP://127.0.0.1:7545MINING STATUSAUTOMININGWORKSPACEDECENTRALIZED_VOTING

SWITCH

MNEMONIC

fall sing man now trend sibling script sad term dignity come pencil

HD PATHm/44'/60'/0'/0'/0/account_index

ADDRESS	BALANCE	TX COUNT	INDEX	
0x37A565F8306E64529B0699eE856505D258d476B7	99.97 ETH	8	0	
0xc9Bdd96304b19b01B0d80B002A01Ee2F94837A25	100.00 ETH	0	1	
0x43F375c29606907bB5E8F26a7699af89FECf9D5b	100.00 ETH	0	2	
0x6c9d94a57690079EEf5cAD8b354A190E180fF5eA	100.00 ETH	0	3	
0x1a6F70fc814037eBc235b966194F9b5e54250e4f	100.00 ETH	0	4	
0xE3D85eb72B3FA7Fb315B1fc1F6C57414008f9100	100.00 ETH	0	5	
0x49365B4ac32d528370eb7E914E59F00980b0f357	100.00 ETH	0	6	
0x94b9b18fe13F87471a57C87fb6D3a0a9095D2d14	100.00 ETH	0	7	
ADDRESS	BALANCE	TX COUNT	INDEX	

Right Ctrl

ubuntu [Running] - Oracle VM VirtualBox

FileMachineViewInputDevicesHelp

ActivitiesGanacheJan 24 18:27

ACCOUNTSBLOCKSTRANSACTIONSCONTRACTSEVENTSLOGS

SEARCH FOR BLOCK NUMBERS OR TX HASHES

CURRENT BLOCK8GAS PRICE2000000000GAS LIMIT6721975HARDFORKMURGLACHERNETWORK ID5777RPC ENDPOINTHTTP://127.0.0.1:7545MINING STATUSAUTOMININGWORKSPACEDECENTRALIZED_VOTING

SWITCH

BLOCK	MINED ON	GAS USED	
8	2022-01-24 16:38:24	27348	1 TRANSACTION
7	2022-01-24 16:38:24	391325	1 TRANSACTION
6	2022-01-24 16:38:22	42348	1 TRANSACTION
5	2022-01-24 16:38:21	238594	1 TRANSACTION
4	2022-01-24 14:47:15	27348	1 TRANSACTION
3	2022-01-24 14:47:15	391325	1 TRANSACTION
2	2022-01-24 14:47:13	42348	1 TRANSACTION
1	2022-01-24 14:47:12	238594	1 TRANSACTION
0	2022-01-24 14:46:30	0	NO TRANSACTIONS

Right Ctrl

ubuntu [Running] - Oracle VM VirtualBox

FileMachineViewInputDevicesHelp

Activities

Jan 24 18:29

Ganache

ACCOUNTSBLOCKSTRANSACTIONS

CONTRACTSEVENTSLOGS

SEARCH FOR BLOCK NUMBERS OR TX HASHES

CURRENT BLOCK0

GAS PRICE2000000000

GAS LIMIT6721975

HARDFORKMURGLACHER

NETWORK ID5777

RPC URLSHTTP://127.0.0.1:7545

MINING STATUSAUTOMINING

WORKSPACEDECENTRALIZED_VOTING

SWITCH

0*37A565F8386E6452988699eE856505D258d476B7

Migrations

42348

0

TX HASH

0*45Cb6aef29903314f68530ef82d00bad6597e78de4674c3c17b7583e693db694

CONTRACT CREATION

FROM ADDRESS

0*37A565F8386E6452988699eE856505D258d476B7

CREATED CONTRACT ADDRESS

0*36D7Ed6F3FCf7f878d6AA5C0b57dbD1D99458257

GAS USED

238594

VALUE

0

TX HASH

0*9241366c8e6967a54ddc9e6bd384c0e672e1ae8bd5e4e264b768253fa55e3c91

CONTRACT CALL

FROM ADDRESS

0*37A565F8386E6452988699eE856505D258d476B7

TO CONTRACT ADDRESS

0*18aE1b233F5eD23B65C6ED43999a869c85C21452

GAS USED

27348

VALUE

0

TX HASH

0*6c6dea751b0f04c4e8257b542f9014288036dee3f24702bed83627ef3654a9eb

CONTRACT CREATION

FROM ADDRESS

0*37A565F8386E6452988699eE856505D258d476B7

CREATED CONTRACT ADDRESS

0*558772f3809D2B1b1334EC2967e86F466529c68

GAS USED

391325

VALUE

0

TX HASH

0*c9468bea6a8ad6e24bd87c2adcd7a09420f6efb7894438a09d7cc09a3716529d

CONTRACT CALL

FROM ADDRESS

0*37A565F8386E6452988699eE856505D258d476B7

TO CONTRACT ADDRESS

0*18aE1b233F5eD23B65C6ED43999a869c85C21452

GAS USED

42348

VALUE

0

TX HASH

0*38f0aa1447079892a514d2f3e4b5388e607d1103c90acb4a4d6a1f88bb661f12

CONTRACT CREATION

FROM ADDRESS

0*37A565F8386E6452988699eE856505D258d476B7

CREATED CONTRACT ADDRESS

0*18aE1b233F5eD23B65C6ED43999a869c85C21452

GAS USED

238594

VALUE

0

ubuntu [Running] - Oracle VM VirtualBox

FileMachineViewInputDevicesHelp

Activities

Jan 24 18:28

Ganache

ACCOUNTSBLOCKSTRANSACTIONS

CONTRACTSEVENTSLOGS

SEARCH FOR BLOCK NUMBERS OR TX HASHES

CURRENT BLOCK0

GAS PRICE2000000000

GAS LIMIT6721975

HARDFORKMURGLACHER

NETWORK ID5777

RPC URLSHTTP://127.0.0.1:7545

MINING STATUSAUTOMINING

WORKSPACEDECENTRALIZED_VOTING

SWITCH

election /home/yeswanth/Desktop/election

NAME

Election

ADDRESS

0*f72E3DA59AB1B620473F384cE27672e1F85c7d96

TX COUNT

0

DEPLOYED

NAME

Migrations

ADDRESS

0*36D7Ed6F3FCf7f878d6AA5C0b57dbD1D99458257

TX COUNT

1

DEPLOYED

