MAHATMA GANDHI MISSION'S
COLLEGE OF ENGINEERING AND TECHNOLOGY
KAMOTHE, NAVI MUMBAI

DEPARTMENT OF INFORMATION TECHNOLOGY

Project report on

SECURE VOTING SYSTEM USING BLOCKCHAIN

Under the Guidance of

Prof. YOGESH SHAHARE

Head of Dept.

Prof. K.V.RAMAN

Prepared By:

1. SURYAWANSHI ROHIT VIJAY
2. SONAWALE DHIRAJ DATTARAY
3. RANE TANMAY SANDEEP

CERTIFICATE

This is to certify that the T.E. mini-project entitled **" Secure voting System using blockchain"** is a bonafide work of **"Suryawanshi Rohit Vijay" (117IT 1010A), "Sonawale Dhiraj Dattatray" (117IT1221A),** and **"Rane Tanmay Sandeep" (117IT1519A)** submitted to University of Mumbai in partial fulfilment of the requirement for the award of the degree of **"Information Technology Engineering"** during the academic year 2019–2020.

INTERNAL EXAMINAR **EXTERNAL EXAMINAR**

HEAD OF DEPARTMENT

T.E. Mini-Project Report Approval

This mini-project synopsis entitled ***"Secure voting System using blockchain"*** by **Suryawanshi Rohit Vijay, Sonawale Dhiraj Dattatray, Rane Tanmay Sandeep** is approved for the degree of ***Information Technology Engineering*** from ***University of Mumbai***.

Examiners

1._____

2._____

Date:

Place:

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signature

Suryawanshi Rohit (117IT1010A)

Signature

Sonawale Dhiraj (117IT1221A)

Signature

Rane Tanmay (117IT1519A)

Date:

Table of Contents

Abstract

Technology has positive impacts on various aspects of our social life. Designing a globally connected architecture enables ease of access to a variety of resources and services. Furthermore, technology like the Internet has been a fertile ground for innovation and creativity. One such innovation is blockchain – a keystone of cryptocurrencies. The blockchain technology is presented as a game-changer for many existing and emerging technologies. With its immutability property and decentralized architecture, it is taking centre stage in many services as an equalization factor to the current parity between consumers and large corporations/governments. One future application of the blockchain is in e-voting. The objective of such a scheme would be to provide a decentralized architecture to run and support a voting scheme that is open, fair, and independently verifiable. In this paper, we propose a potential new e-voting protocol that utilises the blockchain as a transparent ballot box. The protocol has been designed to achieve fundamental e-voting properties as well as offer a degree of decentralization and allow for the voter to change/update their vote (within the permissible voting period). This paper highlights the pros and cons of using blockchain for such a proposal from a practical point view in both development/deployment and usage contexts.

Keywords— Blockchain; Cryptocurrency; SHA-3; E-Voting; Decentralized.

Chapter 1. Introduction

Blockchain: A Blockchain is a growing list of records, called blocks, that are linked using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data. By design, a Blockchain is resistant to modification of the data.In this we are going to use the Operating System(Windows, Ubuntu) so that there are chances hat our code or program can be modified by some unauthorized person . To overcome this issue we can use python file system that will not over writtern or we can encrypt the program file . In this system we will use Biometric Technique and the data sets of Aadhar Card. As soon as the voter keeps his finger on fingerprint scanner, his/her data will be displayed on the screen if and only if he/she is eligible to cast a vote. Once the voter had cast his/her vote to the desired party, a light/symbol will blink or highlight indicating that the voter had successfully casted his/her vote to the desired party.The only condition required by our system is that voter should visit the polling booth to cast the vote.

1.1 Motivation

The existing system only helps the voter to cast his/her vote without much security.With the EVM, instead of issuing a ballot paper, the polling officer will press the Ballot Button which enables the voter to cast their vote. A list of candidate's names and/or symbols will be available on the machine with a blue button next to it. The voter can press the button next to the candidate's name they wish to vote for.No part of the EVM is "networked" is the most important thing .EVM machines are extremely simple machines, like pocket calculators, with no connection to the internet, no operating system and no way of being

altered without physical access to the machines.There were earlier claims regarding EVMs' temper ability and security which have not been proved.

1.2 Problem Statement

- Current Scenario:

Nowadays, EVM machine is introduced and is replaced by the pen and paper method. But EVM hardware is unsafe and any kind of misuse can be made on EVM. Also malicious software programming can be done on it.

- Our solution:

To overcome these disadvantages of the voting processes, we have proposed the idea of Secure Voting System using Blockchain Technology. Our system is easy to use, does not require much manpower and resources. It provides automatic counting of the votes.

- Advantage:

It is highly secure and also it requires less manpower. It is cost effective as it requires one time investment.

1.3 Objectives

The objective of this system would be to provide a decentralized architecture to run and support a voting scheme that is open fair and independently verifiable. Another objective of this system is to reduce the time limit which generally takes in the normal voting system without the blockchain technology. Also in this system the voters vote cannot be modified as we are using blockchain technology in which the person can only view the data but will not be able to modify it.

1.4 Scope

In our system/project we are trying to overcome the gap that are present in the existing system.

- We are using Aadhar card dataset which will contain the voters identity.
- This system is totally hardware dependent;the hardware which is used is the finger print scanner.
- Our project does not exactly accomplish the gap of voter visiting the booth. As the voter has to visit his nearest booth to cast vote because if it is not so then he/she can be pressurized to cast the vote.

1.5 Organization of the report:

• Chapter 1: Introduction

This chapter includes an introduction of our project.

• Chapter 2: Review of Literature

This chapter investigates some researches in the text summarization methods.

• Chapter 3: Requirements gathering and planning

This chapter gives information regarding the requirements of the project.

• Chapter 4: Proposed system

This chapter gives information regarding the overall flow of the system.

Implementation details

This chapter gives implementation details of the system and overview of the entire system. Technologies used

This chapter includes the different technologies that are used in the project.

• Chapter 5: Result and Discussion

This chapter includes analysis of the results and shows how the system works.

- Chapter 6: Conclusion and Future Work

This chapter concludes the dissertation by highlighting its contributions and future directions.

- Chapter 7: References

This chapter concludes the various author names whose papers we referred.

- Chapter 8: Appendix

This chapter includes the full forms of the abbreviations used in the entire report.

Chapter 2. Review of Literature

In this section, we will have a look at several similar systems that are been researched and implemented by other researchers. For further understanding on their methods and techniques, refer to the references page at the end of this report.

2.1 Secure voting system using Blockchain

In 2019, Dipali Pawar, et.al.[1] has proposed Secure Voting System Using Blockchain.In propose system as told earlier voter register himself. During registration system takes voter's unique identity number. Unique identity is for generating unique public and private key for every voter. So here problem of double voting is solved. After taking all required information from voter, if voter is eligible for voting process then only system accept registration of voter. Then system i.e election administrator generate public and private keys for voter.

- ❖ Blockchain offers an updated system which is secure and fair.
- ❖ Transparency allow votes to be followed, counted,and correlated by many different sources.
- ❖ Anonymous transactions of blockchain maintain privacy of the voters.
- ❖ As it is distributed system, data is stored on multiple machines in networks.

2.2 Multi- purpose platform independent online voting system

In 2017, Dr. Z.A. Usmani,et.al.[2] multi- purpose platform independent online voting system.The system would be like that the user won't have to register any account, they just have to provide the Aadhaar Card number i.e. National Identification Number. This would ensure the individuality of the user as the

database consists of contents such as Barcode and OTP for security verification. Therefore a user can simply just enter the Aadhar card number, pass through some security verification and get eligible for voting. The local admin i.e. the user who creates the voting ballot will be given a unique serial code by our system which the local admin broadcast to all the user's who are going to vote. The votes will be calculated automatically and the vote counts will be given to the local admin in the form of graphs and pie charts. After user cast its vote, it will be encrypted with the help of Advanced Encryption Standard (AES) or Data Encryption Standard (DES) which will be visible to the local admin only in the decrypted form.

2.3 A Comparitive Analysis on E-Voting System Using Blockchain

In 2019, Kanika Garg,et.al.[3] has proposed a Comparative Analysis on E-Voting System using Blockchain Technology.As the applications build using blockchain are decentralised and owned by multiple parties and no one can change or update data in the blockchain. If someone tries to do so it will not be accepted by stakeholders. Hence, making blockchain completely trustable. Their is no single owner of blockchain, means no single authority is controlling it and anyone can participate in the network (depending upon the type of blockchain i.e public, private and consortium). Blockchain is a read and write only database i.e once data written on blockchain cannot be altered or changed.. These features of blockchain can be very useful in building a perfect e-voting solution (i.e. building a decentralised voting platform) over traditional system, Encountering the challenges which we are our facing till now. is derived from the Greek word kryptos which means hidden or secret. It is a technique for safe communication in the presence of unsecure third party.

2.4 Secure Digital Voting System based on Blockchain Technology

In 2018,Kashif Mehboob Khan, et.al.[4] has proposed Secure Digital Voting System based on Blockchain Technology.The proposed e-voting system is based on the well-established e-voting approach. The system has been designed to support a voting application in the real world environment taking into account specific requirements such as privacy, eligibility, convenience, receipt-freeness and verifiability. The proposed system aims to achieve secure digital voting without compromising its usability. Within this context, the system is designed using a web-based interface to facilitate user engagement with measures such as finger printing to protect against double voting. With a clear need to administer the voters, constituencies and candidates for constituencies, a user-friendly administrator interface is implemented to enable ease of access. Furthermore, the system allows all voters equal rights of participation and develops a fair and healthy competition among all the candidates while keeping the anonymity of the voters preserved. The cryptographic hash of the transaction (ID) is emailed to the voter as a proof that the vote has been casted which may later on be tracked outside the premises of the constituency.

2.5 A Novel P2P based System with Blockchain for Secured Voting Scheme

In 2019, Vijayalakshmi V,et.al.[5] has proposed A Novel P2P based System with Blockchain for Secured Voting Scheme .we present various solutions for the process of secured voting by integrating e-voting and Blockchain technology. The requirements needed to poll a vote is a web application with the internet connection in the booth and a mobile application, if the user wants to verify his/her polled vote.The web application is responsible for creating and managing the new voting events.The Admin stores the information such as aadhar number, constituency and the details necessary for both the candidate

and the user through a web application. These details are then carried out to the server and maintained.Ethereum is a decentralized, open source platform and an operating system featuring smart contract functionalities.The authentication server is an application that provides network service and used to authenticate the credentials such as aadhar number of the user.Arbitration server is responsible for tracking the entire application server and it acts as an intermediary of blockchain based balloting system and a user.

Chapter 3.Requirement Analysis

3.1 Technical Feasibility

HARDWARE REQUIREMENTS:

System: i5 processor

Hard disk: 2 GB of available disk space minimum 4 GB recommended.

Screen resolution:1280 x 800 min

RAM: 4GB

Biometric : Fingerprint scanner.

SOFTWARE REQUIREMENTS:

Operating System: Windows 10

Django 2.2

Python 3.7.7

Coding Language: Python

3.2 Technologies Used

1. Django

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support.Django was initially developed between 2003 and 2005 by a web team who were responsible for creating and maintaining newspaper websites. After creating a number of sites, the team began to factor out and reuse lots of common code and design patterns. This common code evolved into a generic web development framework, which was open-sourced as the "Django" project in July 2005. Django has continued to

grow and improve, from its first milestone release (1.0) in September 2008 through to the recently-released version 2.0 (2017). Each release has added new functionality and bug fixes, ranging from support for new types of databases, template engines, and caching, through to the addition of "generic" view functions and classes (which reduce the amount of code that developers have to write for a number of programming tasks).

2. Python

Python is a general-purpose language. It has wide range of applications from Web development (like: Django and Bottle), scientific and mathematical computing (Orange, SymPy, NumPy) to desktop graphical user Interfaces (Pygame, Panda3D). The syntax of the language is clean and length of the code is relatively short. It's fun to work in Python because it allows you to think about the problem rather than focusing on the syntax. Python has a very simple and elegant syntax. It's much easier to read and write Python programs compared to other languages like: C++, Java, C#. Python makes programming fun and allows you to focus on the solution rather than syntax.

- **Free and open-source**

You can freely use and distribute Python, even for commercial use. Not only can you use and distribute software’s written in it, you can even make changes to the Python's source code. Python has a large community constantly improving it in each iteration.

- **Portability**

You can move Python programs from one platform to another and run it without any changes. It runs seamlessly on almost all platforms including Windows, Mac OS X and Linux.

- **Extensible and Embeddable**

Suppose an application requires high performance. You can easily combine pieces of C/C++ or other languages with Python code. This will give your application high performance as well as scripting capabilities which other

languages may not provide out of the box.

- **A high-level, interpreted language**

Unlike C/C++, you don't have to worry about daunting tasks like memory management, garbage collection and so on. Likewise, when you run Python code, it automatically converts your code to the language your computer understands. You don't need to worry about any lower-level operations.

3.2 UML Diagrams

3.2.1 Use Case Diagram

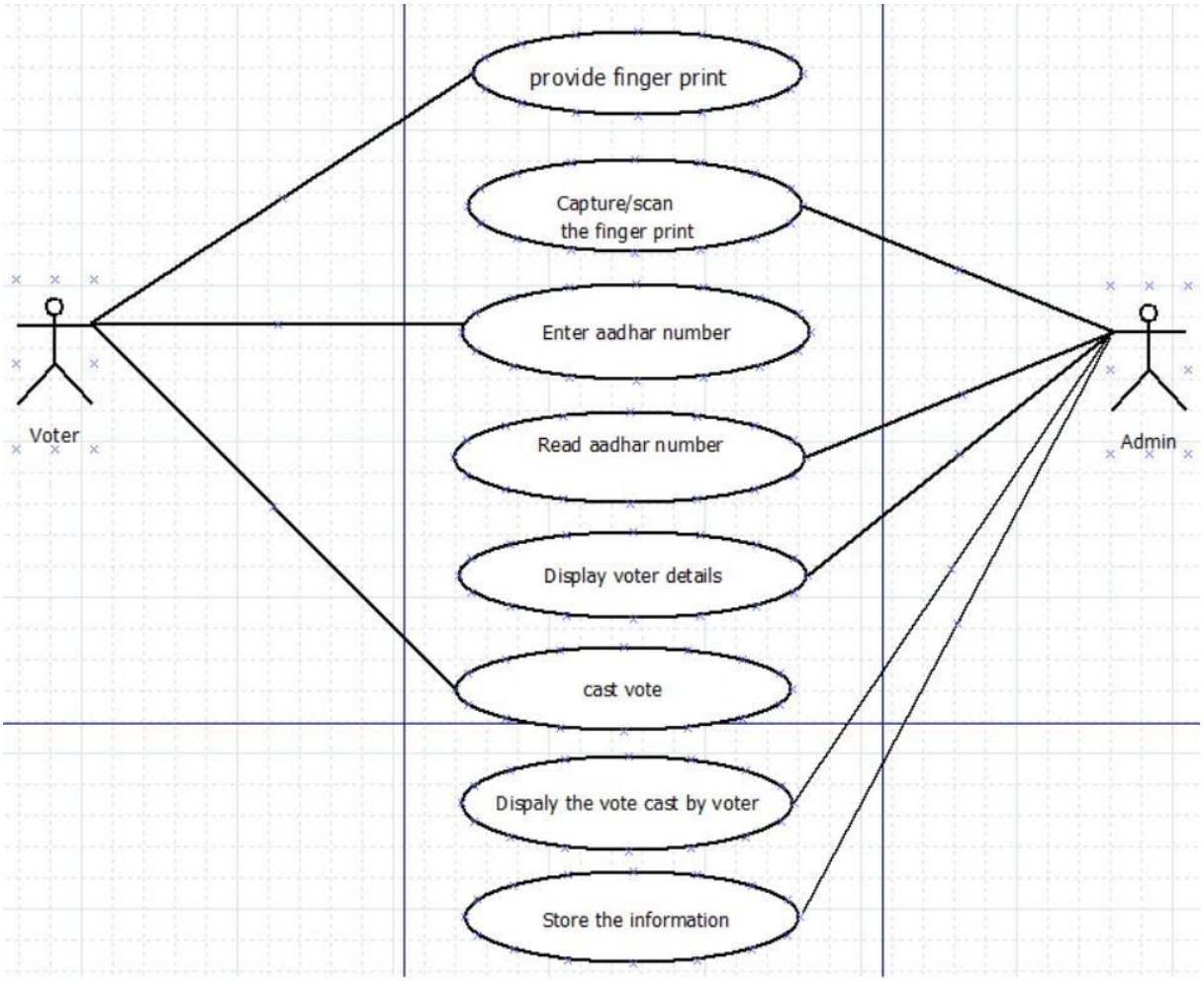

Fig no. 3.2.1

3.2.2Class Diagram

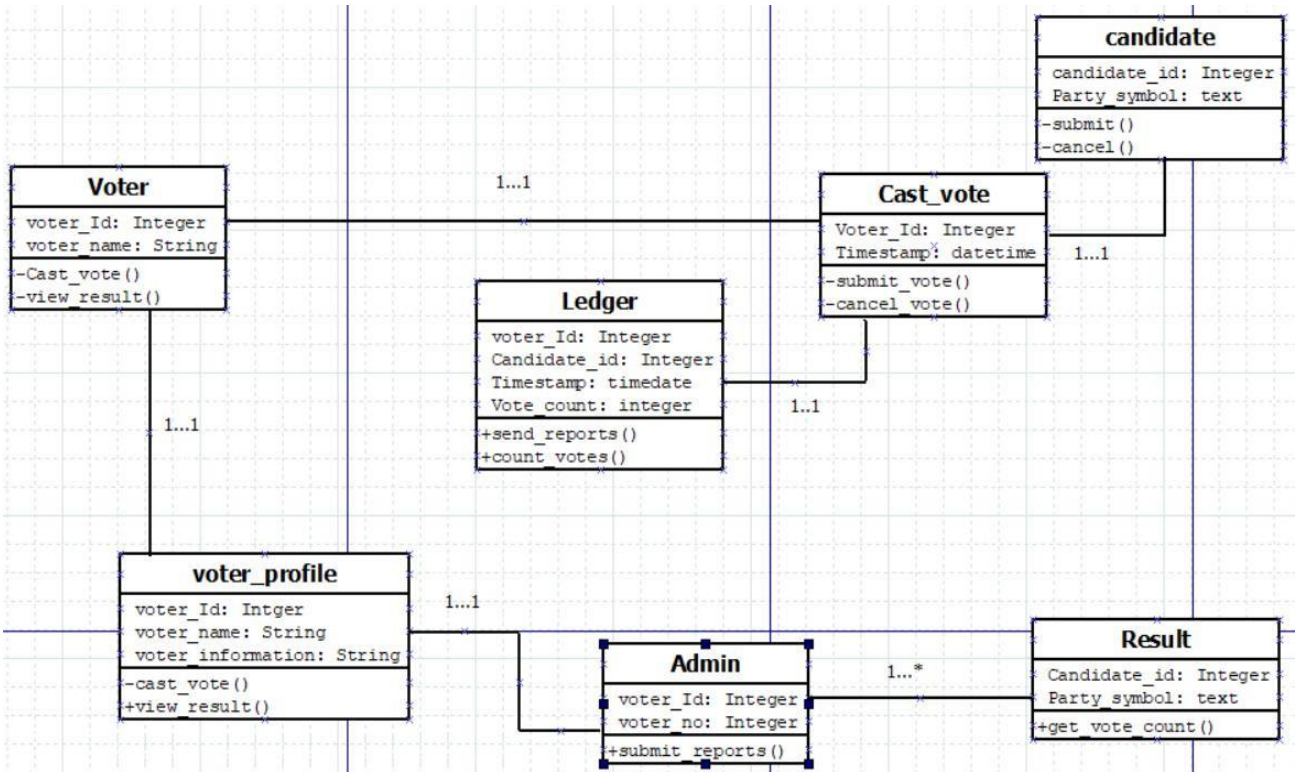

Fig no. 3.2.2

3.2.3Activity Diagram

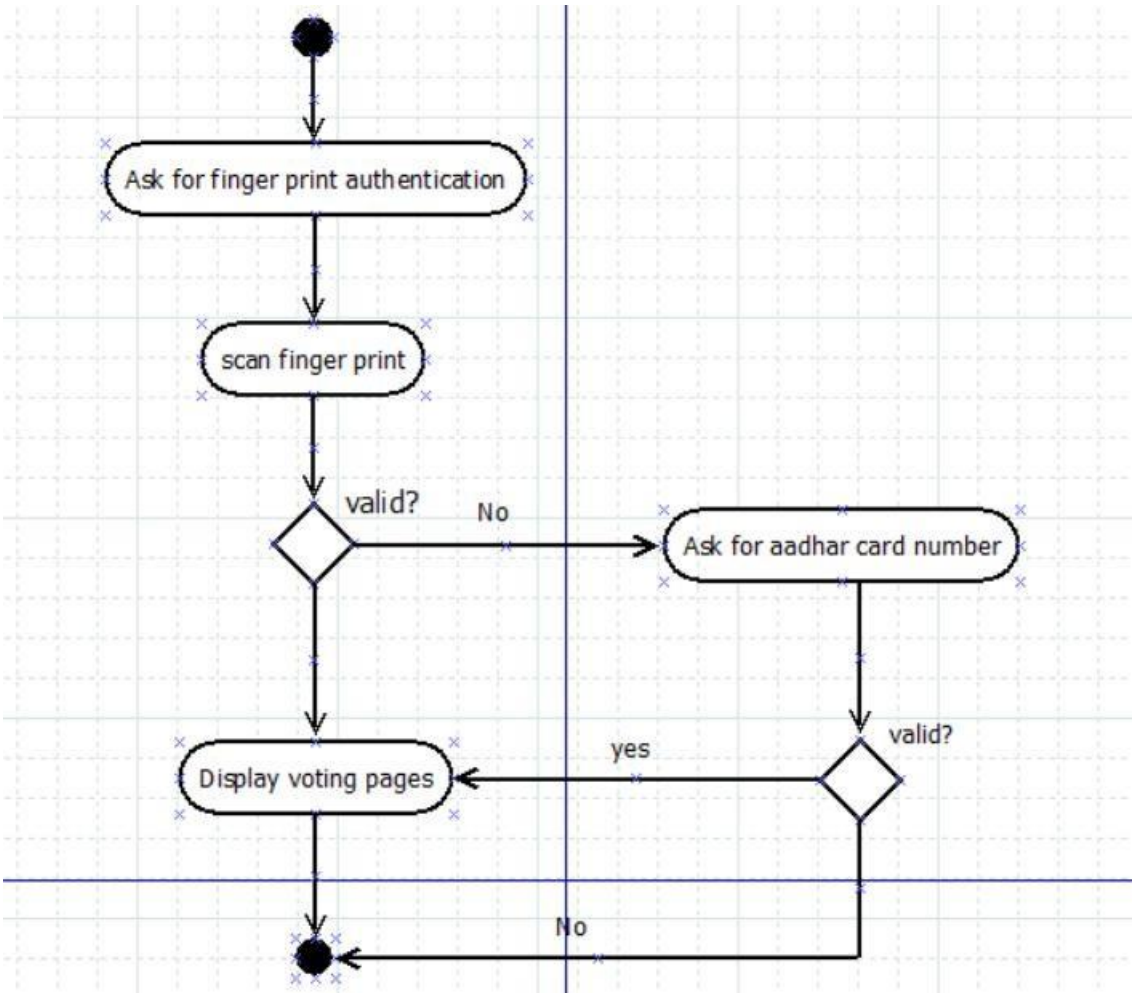

Fig no. 3.2.3

3.3 Time Line Charts

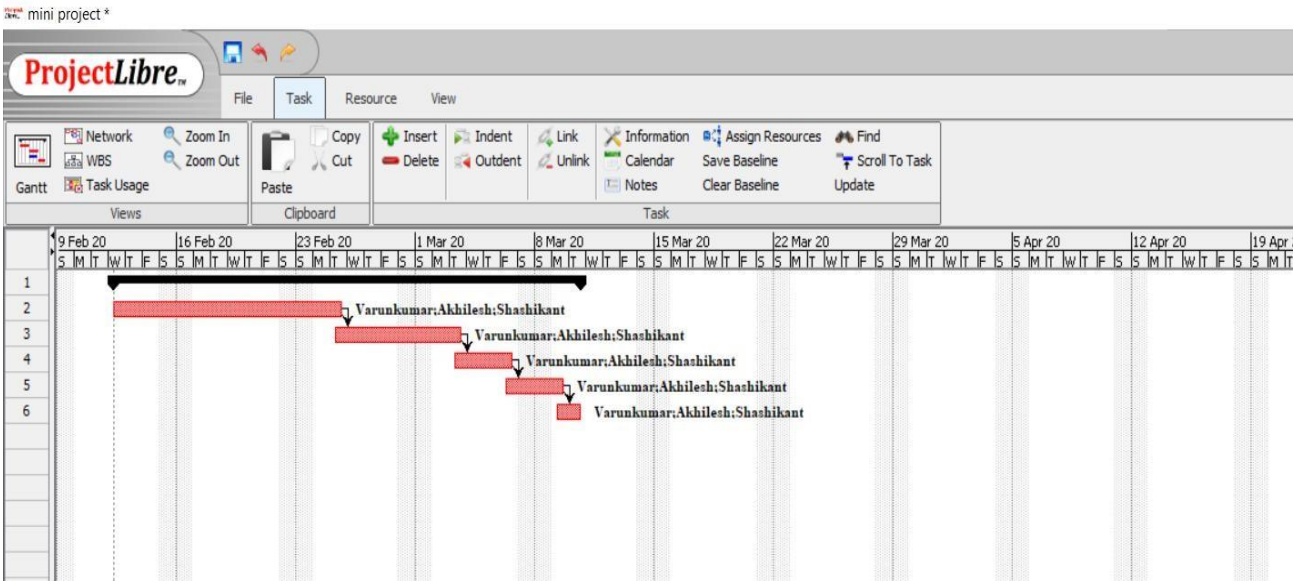

Fig no. 3.3

3.4 W.B.S. CHART

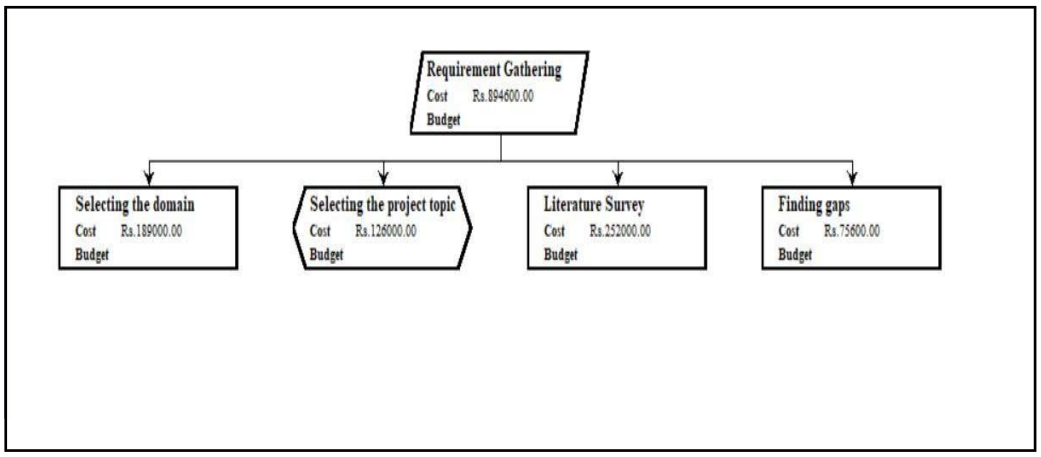

Fig no. 3.4.1

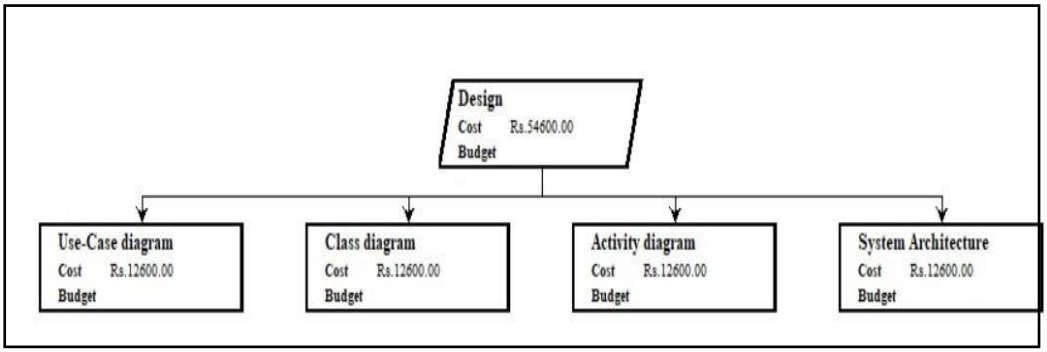

Fig no. 3.4.2

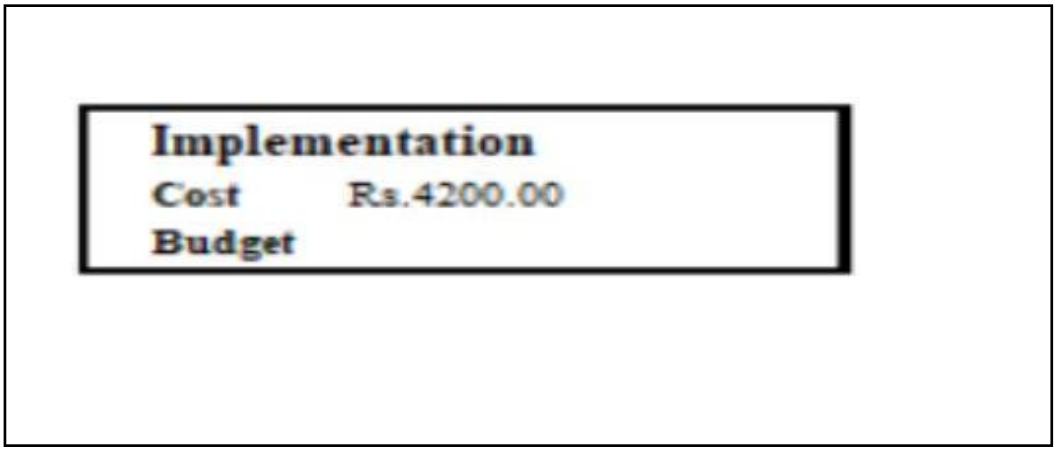

Fig no. 3.4.3

Chapter 4.Report on Present Investigation

4.1 Proposed System

Propose system is internet voting system . We provide an online platform for voting i.e a website. Propose system three parts as Voter, Election Administrator and Election Process.

A) Voter : Voter is the main part of system which participate in election process. He register himself in system by giving his personal information.

B) Election Administrator : To manage all the datacoming from voter during registration

C) and election process, election administrator has worked. Also it generate public and private keys for voters. It is nothing but python packages.

D) Election Process : In this process voter select the candidate to vote and give his vote for selected candidate.

4.1.1 Block Diagram of Proposed System

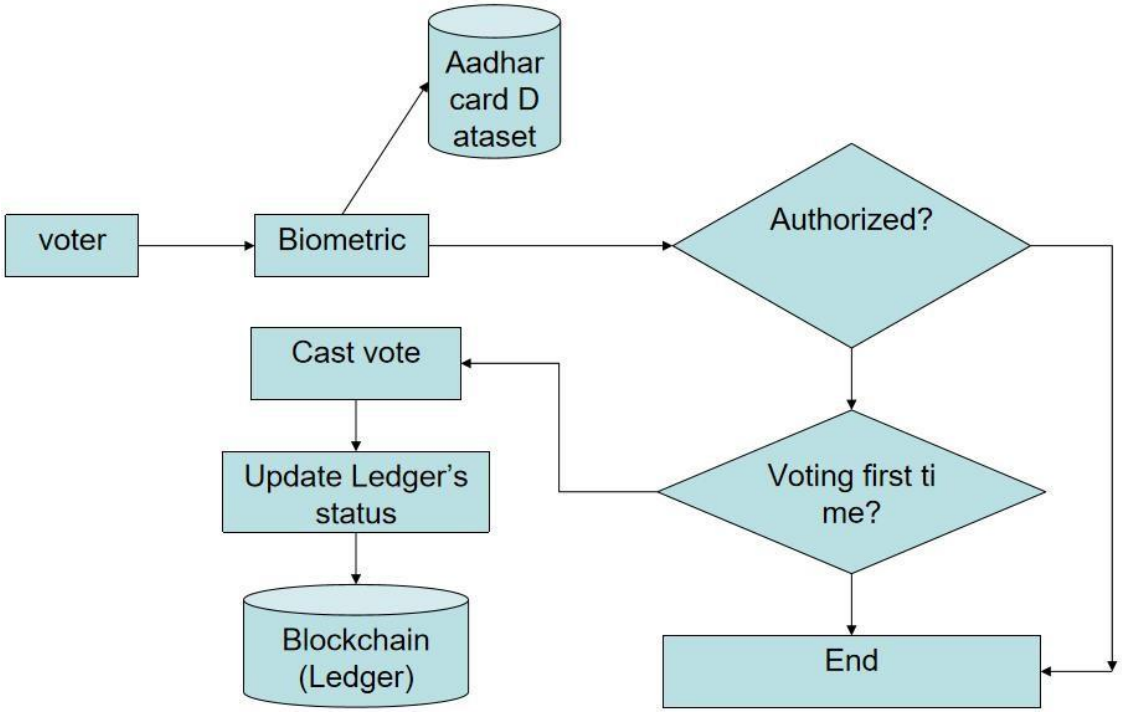

Fig no.:4.1.1

Flow of process:

1. Voter will scan his/her finger using biometric device.
2. After scanning his/her finger print will be matched in aadhar card dataset and required information will be retrived.
3. After that users authorization will be checked based on some condition.
4. once voter satisfies conditions for authorization after it will be checked whether voter is casting vote for first time.
5. If voter is appear for vote for first time then he/she can cast there vote.
6. once they cast the vote status of ledger gets update and information gets stored in it.

4.2 Implementation

Code:

1. Store transactions into blocks

We'll be storing data in our blockchain in a format that's widely used: JSON. Here's what a post stored in blockchain will look like:

```
{
  "author": "some_author_name",
  "content": "Some thoughts that author wants to share",
  "timestamp": "The time at which the content was created"
}
```

The generic term "data" is often replaced on the internet by the term "transactions." So, just to avoid confusion and maintain consistency, we'll be using the term "transaction" to refer to data in our example application.

The transactions are packed into blocks. A block can contain one or many transactions. The blocks containing the transactions are generated frequently and added to the blockchain. Because there can be multiple blocks, each block should have a unique ID:

```
class Block:
    def__init__(self, index, transactions, timestamp):
        """
        Constructor for the `Block` class.
        :param index: Unique ID of the block.
        :param transactions: List of transactions.
        :param timestamp: Time of generation of the block.
        """
        self.index = index
        self.transactions = transactions
        self.timestamp = timestamp
```

2. Add digital fingerprints to the blocks

We'd like to prevent any kind of tampering in the data stored inside the block, and detection is the first step to that. To detect if the data in the block has been tampered with, you can use cryptographic hash functions.

A **hash function** is a function that takes data of any size and produces data of a fixed size from it (a hash), which is generally used to identify the input.

We'll store the hash of the block in a field inside our Block object, and it will act like a digital fingerprint (or signature) of data contained in it:

```
from hashlib import sha256

import json

def compute_hash(block):

    """Returns the hash of the block instance by first converting it into JSON string."""

    block_string = json.dumps(self.__dict__, sort_keys=True)

    return sha256(block_string.encode()).hexdigest()
```

3. Chain the blocks

First block is called the genesis block and it can be generated either manually or through some unique logic. Let's add the previous_hash field to the Block class and implement the initial structure of our Blockchain class.

```
from hashlib import sha256
import json
import time


class Block:
    def __init__(self, index, transactions, timestamp, previous_hash):
        """
        Constructor for the `Block` class.
        :param index:         Unique ID of the block.
        :param transactions:  List of transactions.
        :param timestamp:     Time of generation of the block.
        :param previous_hash: Hash of the previous block in the chain
which this block is part of.
        """
        self.index = index
        self.transactions = transactions
        self.timestamp = timestamp
```

```
        self.previous_hash = previous_hash # Adding the previous  hash
field

    def compute_hash(self):
        """
        Returns the hash of the block instance by first converting it
        into JSON string.
        """
        block_string = json.dumps(self.__dict__, sort_keys=True) #
The string equivalent also considers the previous_hash field now
        return sha256(block_string.encode()).hexdigest()

class Blockchain:

    def__init__(self):
        """
        Constructor for the `Blockchain` class.
        """
        self.chain = []
        self.create_genesis_block()

    def create_genesis_block(self):
        """
        A function to generate genesis block and appends it to
        the chain. The block has index 0, previous_hash as 0, and
        a valid hash.
        """
        genesis_block = Block(0, [], time.time(), "0")
        genesis_block.hash = genesis_block.compute_hash()
        self.chain.append(genesis_block)

    @property
    def last_block(self):
        """
        A quick pythonic way to retrieve the most recent block in the
chain. Note that
        the chain will always consist of at least one block (i.e.,
genesis block)
        """
        return self.chain[-1]
```

4. Implement a proof of work algorithm

Proof of work is difficult to compute but very easy to verify once you figure out the nonce

```
class Blockchain:
    # difficulty of PoW algorithm
    difficulty = 2

    """
    Previous code contd..
```

```
    """

    def proof_of_work(self, block):
        """
        Function that tries different values of the nonce to get a
hash
        that satisfies our difficulty criteria.
        """
        block.nonce = 0

        computed_hash = block.compute_hash()
        while not computed_hash.startswith('0' *
Blockchain.difficulty):
            block.nonce += 1
            computed_hash = block.compute_hash()

        return computed_hash
```

5. Add blocks to the chain

To add a block to the chain, we'll first have to verify that:

- The data has not been tampered with (the proof of work provided is correct).
- The order of transactions is preserved (the previous_hash field of the block to be added points to the hash of the latest block in our chain).

```
class Blockchain:
    """
    Previous code contd..
    """

    def add_block(self, block, proof):
        """
        A function that adds the block to the chain after
verification.
        Verification includes:
        * Checking if the proof is valid.
        * The previous_hash referred in the block and the hash of a
latest block
          in the chain match.
        """
        previous_hash = self.last_block.hash

        if previous_hash != block.previous_hash:
            return False

        if not Blockchain.is_valid_proof(block, proof):
            return False

        block.hash = proof
        self.chain.append(block)
```

```
        return True

    def is_valid_proof(self, block, block_hash):
        """
        Check if block_hash is valid hash of block and satisfies
        the difficulty criteria.
        """
        return (block_hash.startswith('0' * Blockchain.difficulty)
and
                block_hash == block.compute_hash())
```

Mining

The transactions will be initially stored as a pool of unconfirmed transactions. The process of putting the unconfirmed transactions in a block and computing proof of work is known as the **mining** of blocks. Once the nonce satisfying our constraints is figured out, we can say that a block has been mined and it can be put into the blockchain.

In most of the cryptocurrencies (including Bitcoin), miners may be awarded some cryptocurrency as a reward for spending their computing power to compute a proof of work. Here's what our mining function looks like:

```
class Blockchain:

    def__init__(self):
        self.unconfirmed_transactions = [] # data yet to get into
blockchain
        self.chain = []
        self.create_genesis_block()

    """
    Previous code contd...
    """

    def add_new_transaction(self, transaction):
        self.unconfirmed_transactions.append(transaction)

    def mine(self):
        """
        This function serves as an interface to add the pending
        transactions to the blockchain by adding them to the block
        and figuring out proof of  work.
        """
        if not self.unconfirmed_transactions:
            return False

        last_block = self.last_block
```

```
        new_block = Block(index=last_block.index + 1,
                          transactions=self.unconfirmed_transactions,
                          timestamp=time.time(),
                          previous_hash=last_block.hash)

        proof = self.proof_of_work(new_block)
        self.add_block(new_block, proof)
        self.unconfirmed_transactions = []
        return new_block.index
```

Chapter 5.Results & Discussion

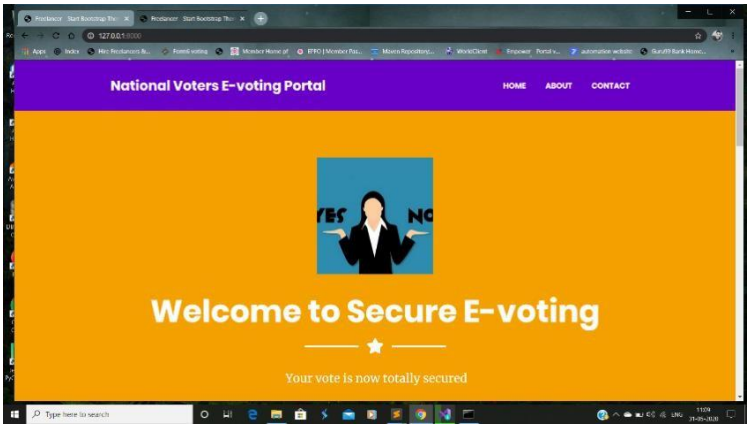

Fig no. 5.1

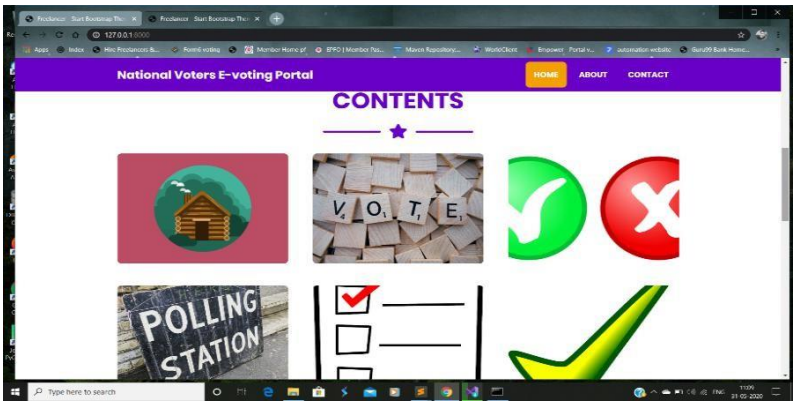

Fig no. 5.2

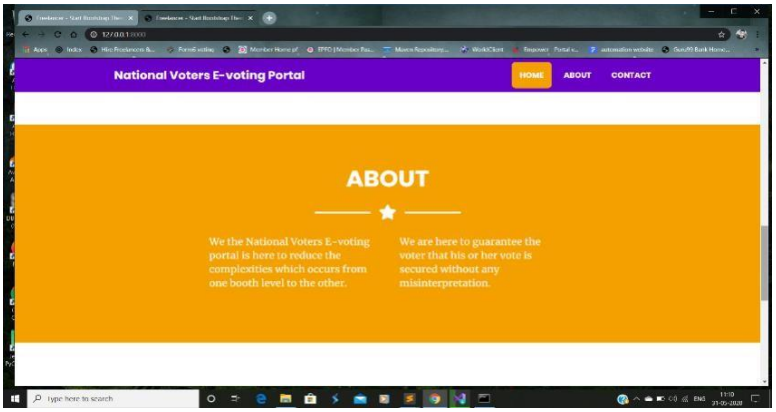

Fig no. 5.3

We have created this interface by which the voter will interact with the system and will cast a vote and vote will be stored in ledger.

Chapter 6.Conclusion & Future work

6.1 Conclusion

Blockchain Technology is gaining popularity day by day. Using blockchain in voting system will help to achieve secure and cost-efficient election while guaranteeing voter's privacy. Also, due to the encryption mechanism, it is impossible for any person to gain access to all the votes without first taking control of the entire service network.

6.2 Future Work

With the evolution of new technology everyday.our system will be further be developed and adopted on large scale;as voting plays a major role in the Indian constituency .Also to enhance the system we can replace or add the finger print scanner with the eye scanner which will not required any kind of touch; which will be a great measure looking on the current situation.

Chapter 7. References

[1] Dipali Pawar, Pooja Sarode, Shilpa Santure, Poonam Thore , “Secure Voting System Using Blockchain” , IJERT 2019

[2] Dr. Z.A. Usmani, Kaif Patanwala, Mukesh Panigrahi, Ajay Nair, “Multi purpose platform independent Online Voting System”, ICIIECS 2017

[3] Kanika Garg, Pavi Saraswat, Sachin Bisht, Sahil Kr. Aggarwal, Sai KrishnaKothari, Sahil Gupta, “A Comparative Analysis on E-Voting System using Blockchain ”, IEEE 2019

[4] Kashif Mehboob Khan, Junaid Arshad, Muhammad Mubashir Khan, “Secure Digital Voting System based on Blockchain Technology”, IEEE 2018

[5] Vijayalakshmi V, Vimal S, “A Novel P2P based System with Blockchain for Secured Voting Scheme”, ICONSTEM 2019

Chapter 8

Acknowledgement

We would like to express our deepest gratitude to all those who provided us the possibility to complete this report. We have taken efforts in this project, but, it would not have been possible without the kind support and help of many individuals. We would like to extend our sincere thanks to all of them.

We are highly indebted to Mrs. Jessica Dias(Project Guide) and Dr. Terrence Johnson(HOD) for their guidance and constant supervision as well as for providing necessary information regarding the project and also for their support in completing the project.

Thus, we have fully obliged and convey our thanks to teaching as well as non-teaching staff of the department.