

AN ONLINE VOTING SYSTEM USING VISUAL CRYPTOGRAPHY AND AADHAR

ABSTRACT

Trustworthy election is an efficient mechanism to democracy. It is a process in which people choose their representative to form a government. The key requirement for effective election process are correctness, robustness, and security. There are variety of voting schemes which are based on the traditional method.

But because of the inconvenient traditional voting system there is tremendous decrease in number of voters. Hence online voting system is introduced to overcome the drawback of traditional voting system.

The online voting system where a person can cast his vote from anywhere in the world by logging onto a website is being adopted by an increasing number of governments and companies.

Online voting system has the facility to complete voting process faster than the paper ballot voting procedure. This system itself should be intelligent to earn the trust and confidence of the user by providing enhanced security and reliability. The security is an important factor in any voting system.

However, the problem with this is that any user can log in to the website and cast votes multiple times using various identities which leads to unfair election result.

This is a frequently encountered issue with the online voting system which is deteriorating the authenticity of the decisions taken via the online mode.

So security is provided through the authentication. Authentication is a secured way to check the voter's identity. The principal objective of authentication is to prevent any adversary from copying other user. Secret sharing schemes are the powerful mechanism used for the authentication.

Secret sharing schemes are ideal for storing information that is highly sensitive and highly important. Secret sharing is also termed as secret splitting. Secret sharing is a method for allocating a secret among a group of participants. Each of whom is allocated a share of the secret.

In this paper I have implemented a two factor authentication system through which a voter can vote by passing through biometrics and VC share mechanism which facilitates authentication as well identification of a genuine voter.

CHAPTER-1

INTRODUCTION

1.1 Introduction to Online Voting system

The Online Voting system is designed to count the number of votes and thereby calculate the percentage of votes. Also the number of vote a candidate obtains is also obtained. Along with the number the percentage of votes for each candidate is calculated. The system is so designed that it can also check for duplication. It then decides the winner in every section. The project is designed with a modular approach and the number of modules is decided as per the requirements of the organization. The three modules are administrator module, the user module and Database. The administrator has total authority of the organization and maintains all the aspects. The user has the provision to view the list of all candidates and results as well as vote for the desired candidates.

1.2 Introduction to Visual Cryptography

Visual cryptography is a cryptographic technique which allows visual information (pictures, text, etc.) to be encrypted in such a way that the decrypted information appears as a visual image.

One of the best-known techniques has been credited to Moni Naor and Adi Shamir, who developed it in 1994. They demonstrated a visual secret sharing scheme, where an image was broken up into n shares so that only someone with all n shares could decrypt the image, while any $n - 1$ shares revealed no information about the original image. Each share was printed on a separate transparency, and decryption was performed by overlaying the shares. When all n shares were overlaid, the original image would appear. There are several generalizations of the basic scheme including k -out-of- n visual cryptography.

Using a similar idea, transparencies can be used to implement a one-time pad encryption, where one transparency is a shared random pad, and another transparency acts as the ciphertext. Normally, there is an expansion of space requirement in visual cryptography. But if one of the two shares are structured recursively, the efficiency of visual cryptography can be increased to 100%.

Some antecedents of visual cryptography are in patents from the 1960s. Other antecedents are in the work on perception and secure communication.

Visual cryptography can be used to protect biometric templates in which decryption does not require any complex computations.

1.3 Fingerprint Recognition

Fingerprint recognition describes the process of obtaining a digital representation of a fingerprint and comparing it to a stored digital version of a fingerprint. Electronic fingerprint scanners capture digital "pictures" of fingerprints, either based on light reflections of the finger's ridges and valleys, ultrasonic's, or the electrical properties of the finger's ridges and valleys. These pictures are then processed into digital templates that contain the unique extracted features of a finger. These digital fingerprint templates can be stored in databases and used in place of traditional passwords for secure access. Instead of typing a password, users place a finger on an electronic scanner. The scanner, or reader, compares the subsist fingerprint to the fingerprint template stored in a database to resolve the identity and validity of the person requesting access.

1.3.1 Fingerprint Identification Algorithm:

During the enrolment process, the system takes in the Voter's Aadhar and uses the fingerprint from the Aadhar and saves the persons fingerprint into database. The authentication process: It is used to authenticate the claimed person. This process consists of comparing a captured fingerprint to an Aadhar fingerprint in order to determine whether the two match. If the two finger prints match, then it allow user to cast the vote.

1.4 Existing Systems

- Online voting system using Visual Cryptography with login credentials.
- Online voting system using Visual Cryptography and login credentials along with face detection.
- Existing System also consist of electronic voting system.

1.5 Drawbacks of Existing System

- Voter's identity is not validated in existing system efficiently.
- Impossible to identify whether a voter is authentic or not
- The Existing system is vulnerable to SYBIL attack.
- Verification of votes can't be done in case of any security breach.
- Anonymity of voter is not available.

1.6 Proposed System

To rectify these issues, a two way authentication online voting system comes into picture where authenticating a voter's vote with the aid of Aadhar fingerprints mechanism as well as Visual Cryptography (VC) is suggested. Using the fingerprints authentic voter is verified, with the VC scheme, it is ensured that only the true user can cast his vote as this is only possible if he logs into authentic website using the correct aadhar credentials.

- Authenticity
- Confidentiality
- Validation
- Transparency

These are the features that are possible with the help of this system in order to identify the authentic voter. Moreover the voting process is validated end to end and there won't be any issues that can arise while the voting process is undergoing. It also helps in maintaining the transparency of all the votes that are being generated and tallied from the online voting system to the public as well as the government. Since the details of the voters is confidential it is restricted to the government itself and there won't be any leakage of data in the process.

CHAPTER 2

LITERATURE SURVEY

2.1 Cryptography

The Internet is the fastest growing communication medium and essential part of the infrastructure, nowadays. To cope with the growth of internet it has become a constant struggle to keep the secrecy of information and when profits are involved, protect the copyright of data. To provide secrecy and copyright of data, many of the steganographic techniques have been developed. But each of the technique has their respective pros and cons. Where one technique lacks in payload capacity, the other lacks in robustness. So, the main emphasis of cryptography is to overcome these shortcomings.

Cryptography is technique of securing information and communications through use of codes so that only those person for whom the information is intended can understand it and process it. Thus preventing unauthorized access to information. The prefix “crypt” means “hidden” and suffix graphy means “writing”.

In Cryptography the techniques which are used to protect information are obtained from mathematical concepts and a set of rule based calculations known as algorithms to convert messages in ways that make it hard to decode it. These algorithms are used for cryptographic key generation, digital signing, verification to protect data privacy, web browsing on internet and to protect confidential transactions such as credit card and debit card transactions.

Techniques used For Cryptography:

In today’s age of computers cryptography is often associated with the process where an ordinary plain text is converted to cipher text which is the text made such that intended receiver of the text can only decode it and hence this process is known as encryption. The process of conversion of cipher text to plain text this is known as decryption.

Features of Cryptography are as follows:

1. Confidentiality:

Information can only be accessed by the person for whom it is intended and no other person except him can access it.

2. Integrity:

Information cannot be modified in storage or transition between sender and intended receiver without any addition to information being detected.

3. Non-repudiation:

The creator/sender of information cannot deny his or her intention to send information at later stage.

4. Authentication:

The identities of sender and receiver are confirmed. As well as destination/origin of information is confirmed.

2.2 Types of Cryptography:

In general there are three types Of cryptography:

1. Symmetric Key Cryptography:

It is an encryption system where the sender and receiver of message use a single common key to encrypt and encrypt messages. Symmetric Key Systems are faster and simpler but the problem is that sender and receiver have to somehow exchange key in a secure manner. The most popular symmetric key cryptography system is Data Encryption System (DES).

2. Hash Functions:

There is no usage of any key in this algorithm. A hash value with fixed length is calculated as per the plain text which makes it impossible for contents of plain text to be recovered. Many operating systems use hash functions to encrypt passwords.

3. Asymmetric Key Cryptography:

Under this system a pair of keys is used to encrypt and decrypt information. A public key is used for encryption and a private key is used for decryption. Public key and Private Key are different. Even if the public key is known by everyone the intended receiver can only decode it because he alone knows the private key.

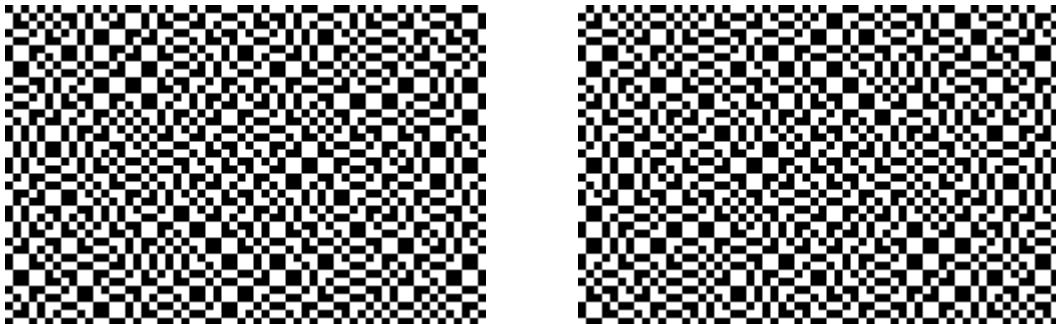
2.3 Visual Cryptography

Visual cryptography is a cryptographic technique which allows visual information (pictures, text, etc.) to be encrypted in such a way that decryption can be done just by sight reading. Visual cryptography, degree associated rising cryptography technology, uses the characteristics of human vision to rewrite encrypted photos. Visual cryptography provides secured digital transmission that is used just for merely the once.

Numerous guidance like military maps and business identifications are transmitted over the internet. Whereas pattern secret photos, security problems ought to be compelled to be taken into thought as a result of hackers may utilize weak link over the communication network to steal info that they need. To touch upon the protection problems with secret photos, varied image secret sharing schemes are developed. Anyone will use it for coding with none science information and any computations.

2.3.1 Visual Cryptography Process

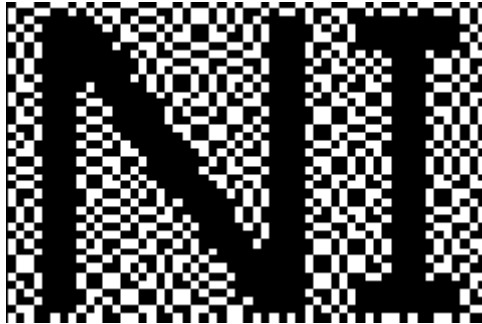
The basis of the technique is the superposition (overlying) of two semi-transparent layers. Imagine two sheets of transparency covered with a seemingly random collection of black pixels.



Individually, there is no discernible message printed on either one of the sheets. Overlapping them creates addition interference to the light passing through (mathematically the equivalent of performing a Boolean OR operation with the images), but still it just looks like a random collection of pixels.



Mysteriously, however, if the two grids are overlaid correctly, at just the right position, a message magically appears! The patterns are designed to reveal a message.



2.3.2 Authentication

Authentication is a secured way to check the voter's identity. The principal objective of authentication is to prevent any adversary from copying other user. Secret sharing schemes are the powerful mechanism used for the authentication. Secret sharing schemes are ideal for storing information that is highly sensitive and highly important. Secret sharing is also termed as secret splitting. Secret sharing is a method for allocating a secret among a group of participants. Each of whom is allocated a share of the secret. For reconstruction it uses t shares out of n no of shares. The system uses the idea to fit a unique polynomial of degree $(t-1)$ to any set of t points that lie on the polynomial. For straight line it takes two points, three points to define a quadratic, and to define cubic curve it takes four points, and so on. It means to form the polynomial of degree $t-1$ it takes t points. The method is to generate a polynomial of degree $t-1$ with the secret as the first coefficient and the remaining coefficients picked at random. It finds n points on the curve and give one to each of the participants. When at least t out of the n participants expose their points, there is sufficient information to fit a $(t-1)$ th degree polynomial to them, and the first coefficient being the secret.

2.4 Steganography

Steganography is the practice of concealing a file, message, image, or video within another file, message, image, or video. The word *steganography* combines the Greek words *steganos*, meaning "covered or concealed", and *graphe* (γραφή) meaning "writing".

The advantage of steganography over cryptography alone is that the intended secret message does not attract attention to itself as an object of scrutiny. Plainly visible encrypted messages, no matter how unbreakable they are, arouse interest and may in themselves be incriminating in countries in which encryption is illegal.

Whereas cryptography is the practice of protecting the contents of a message alone, steganography is concerned both with concealing the fact that a secret message is being sent and its contents.

Steganography includes the concealment of information within computer files. In digital steganography, electronic communications may include steganographic coding inside of a transport layer, such as a document file, image file, program or protocol. Media files are ideal for steganographic transmission because of their large size. For example, a sender might start with an innocuous image file and adjust the color of every hundredth pixel to correspond to a letter in the alphabet. The change is so subtle that someone who is not specifically looking for it is unlikely to notice the change.

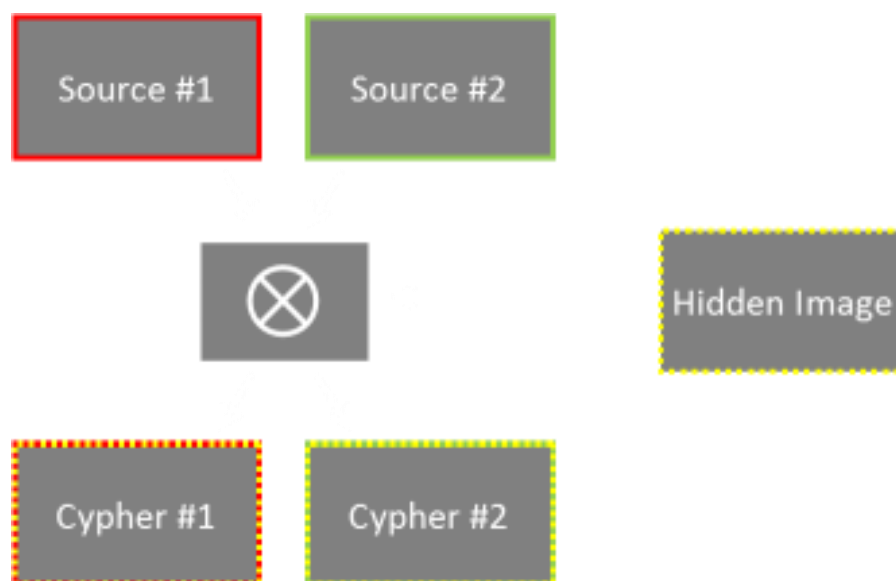
2.4.1 Visual Steganography

We can use cryptography technique to do something even cooler!

Imagine that, in addition to the two source images, we have a *third* secret image we want to encode. Let's say we want to produce two cypher images that look 'innocent', but secretly hide the third. The generated two cypher images could be printed on transparencies and made to look like legitimate images of no consequence.

However, these images, when combined in just the correct way, could be used reveal a third message.

The technology of hiding images inside other images is called Steganography.

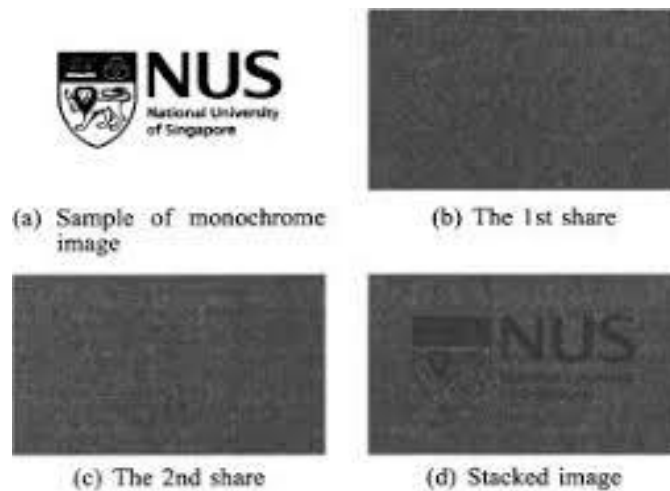


CHAPTER 3

ALGORITHM DESCRIPTION

3.1 *Two out of Two* VC Share Algorithm

2 out-of-2 visual cryptography scheme a normal scheme of (2,2) VCS generate 2 shares picture on the basis of an original image from share's at user 's end must superimpose both shares to generate original image. To secure the ratio of obtained secret image from shares in (2, 2) scheme every pixel in secret image (original image) can be replaced in share image by a 2×2 block of sub pixels. According to table 1, for white original pixel one of six combinations is created. Same process is applicable for black pixel. After superimposing the shares the original image will be obtained.



Authentication Phase

Authentication is an important mechanism to provide security to the system. Secret sharing is an efficient technique used for secured authentication. For authentication it requires both the shares. At the time of authentication voter will produce his share and authority will produce his share both the shares are necessary to reconstruct the original secret. Using authentication only valid voters will be capable to cast the vote.

Results in 2 by 2 scheme (2 subpixels)

- The secret image I is encoded into Share1 & Share2 two shares.
- D is decoded by superimposing these two shares with 50% loss of contrast. ($\alpha = 1/2$)

- The decoded image is identified, although some contrast loss is observed.
- Due to pixel expansion the width of the decoded image is twice as that of the original image.

3.1.2 Advantages

- Simple to implement
- Decryption algorithm not required (Use a human Visual System). So a person unknown to cryptography can decrypt the message.
- We can send cipher text through FAX or E-MAIL.
- Lower computational cost since the secret message is recognized only by human eyes and not cryptographically computed.

3.1.2 Disadvantages

- The contrast of the reconstructed image is not maintained.
- Perfect alignment of the transparencies is troublesome.
- Due to pixel expansion the width of the decoded image is twice as that of the original image. Leads to loss of information due to change in aspect ratio.
- Additional processing is required for colored images.

3.1.3 2 out of 2 Share Algorithm Application

- Biometric security
- Watermarking
- Steganography
- Remote electronic voting
- Bank customer identification – Bank sends customer a set of keys in advance – Bank web site displays cipher – Customer applies overlay, reads transaction key – Customer enters transaction key

CHAPTER 4

SYSTEM ANALYSIS

4.1 System Requirements

4.1.1 Software Requirements

a) JS (Front End)

JavaScript often abbreviated as **JS**, is a high-level, interpreted scripting language that conforms to the ECMAScript specification. JavaScript has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.

Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. JavaScript enables interactive web pages and is an essential part of web applications. The vast majority of websites use it, and major web browsers have a dedicated JavaScript engine to execute it.

As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative (including object-oriented and prototype-based) programming styles. It has APIs for working with text, arrays, dates, regular expressions, and the DOM, but the language itself does not include any I/O, such as networking, storage, or graphics facilities. It relies upon the host environment in which it is embedded to provide these features.



b) jQuery (Front End)

jQuery is a JavaScript library designed to simplify HTML DOM tree traversal and manipulation, as well as event handling, CSS animation, and Ajax. It is free, open-source software using the permissive MIT License. Web analysis indicates that it is the most widely deployed JavaScript library by a large margin, having 3 to 4 times more usage than any other JavaScript library. jQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications. jQuery also provides capabilities for developers to

create plug-ins on top of the JavaScript library. This enables developers to create abstractions for low-level interaction and animation, advanced effects and high-level, themeable widgets. The modular approach to the jQuery library allows the creation of powerful dynamic web pages and Web applications.

c) Scala (Back End)

Scala combines object-oriented and functional programming in one concise, high-level language. Scala's static types help avoid bugs in complex applications, and its JVM and JavaScript runtimes let you build high-performance systems with easy access to huge ecosystems of libraries.



d) XAMPP, Visual Studio (Hosting)

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible.



e) PhpMyAdmin

phpMyAdmin is a free software tool written in PHP, intended to handle the administration of MySQL over the Web. phpMyAdmin supports a wide range of operations on MySQL and MariaDB. Frequently used operations (managing databases, tables, columns, relations, indexes, users, permissions, etc) can be performed via the user interface, while you still have the ability to directly execute any SQL statement.

phpMyAdmin comes with a wide range of documentation and users are welcome to update our wiki pages to share ideas and howtos for various operations. The phpMyAdmin team will try to help you if you face any problem; you can use a variety

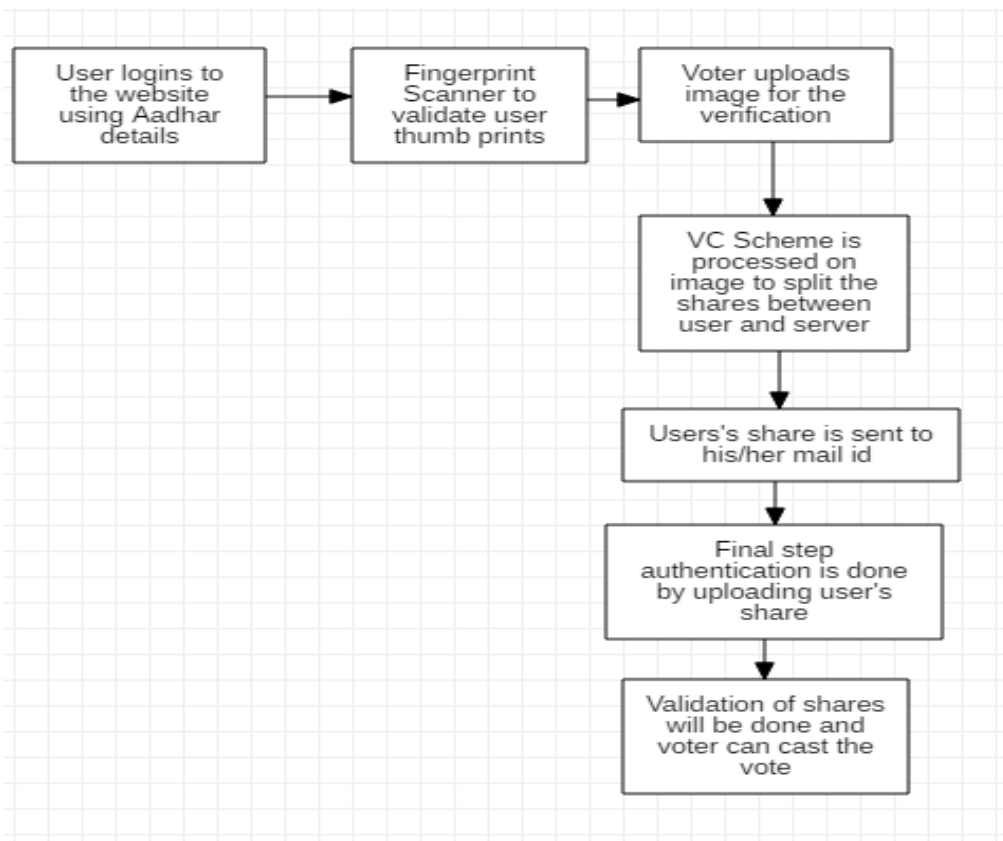
of support channels to get help. phpMyAdmin is also very deeply documented in a book written by one of the developers – Mastering phpMyAdmin for Effective MySQL Management, which is available in English and Spanish. To ease usage to a wide range of people, phpMyAdmin is being translated into 72 languages and supports both LTR and RTL languages. phpMyAdmin is a mature project with a stable and flexible code base; you can find out more about the project and its history and the awards it earned. When the project turned 15, we published a celebration page.



4.1.2 Hardware Requirements

- Processor: Intel® Xeon® processor 3500 series and Above
- HDD: Minimum 500GB Disk Space
- RAM: Minimum 4GB
- OS: Windows 8.1 or Latest, Linux, MacOSX

4.2 System Architecture



CHAPTER 5

SYSTEM DESIGN

5.1 UML Diagrams

Unified Modelling Language is a tool that helps a designer to present his ideas about the project to his client and his developer. Modelling plays a crucial role in designing a software. A poorly designed model can lead to a poorly developed software. A UML system has using five different views that help in describing systems from different perspectives. Each view has a set of diagrams that and components that represent the real time objects.

a. User Model View:

- I. It models the user behaviour in a system context.
- II. All the diagrams are drawn keeping in mind the user's response and reaction towards a system.

b. Structural Model View:

- I. This view consists of class diagram and object diagram which is used to model the static structures.
- II. It uses objects, attributes, operations and relationships.

c. Behavioural Model View:

- I. It mainly consists of the sequence diagram, collaboration diagram, state chart diagram and activity diagram. They mainly represent flow of actions between different objects involved in the system
- II. They are used to visualize various dynamic aspects of the system architecture.

d. Implementation Model View:

- I. This view consists of component diagrams and deployment diagrams. This view models the static software modules for an organization.
- II. This usually contains the data files, documentation, the executables and source code.
- III. These are the physically replaceable components of the system. They are modelled using component diagrams.

5.1.1 Use Case Diagram

The basic representation for the interaction of the user with the system is represented using the use case diagram. It involves the relationship between the user and various use cases with the actors being involved. There are different kinds of relationships that are involved between the use cases and the actors. They include:

- Association relationship
- Generalization
- Dependency
- Realizations
- Transitions

The following represents the use case diagram of the proposed system:

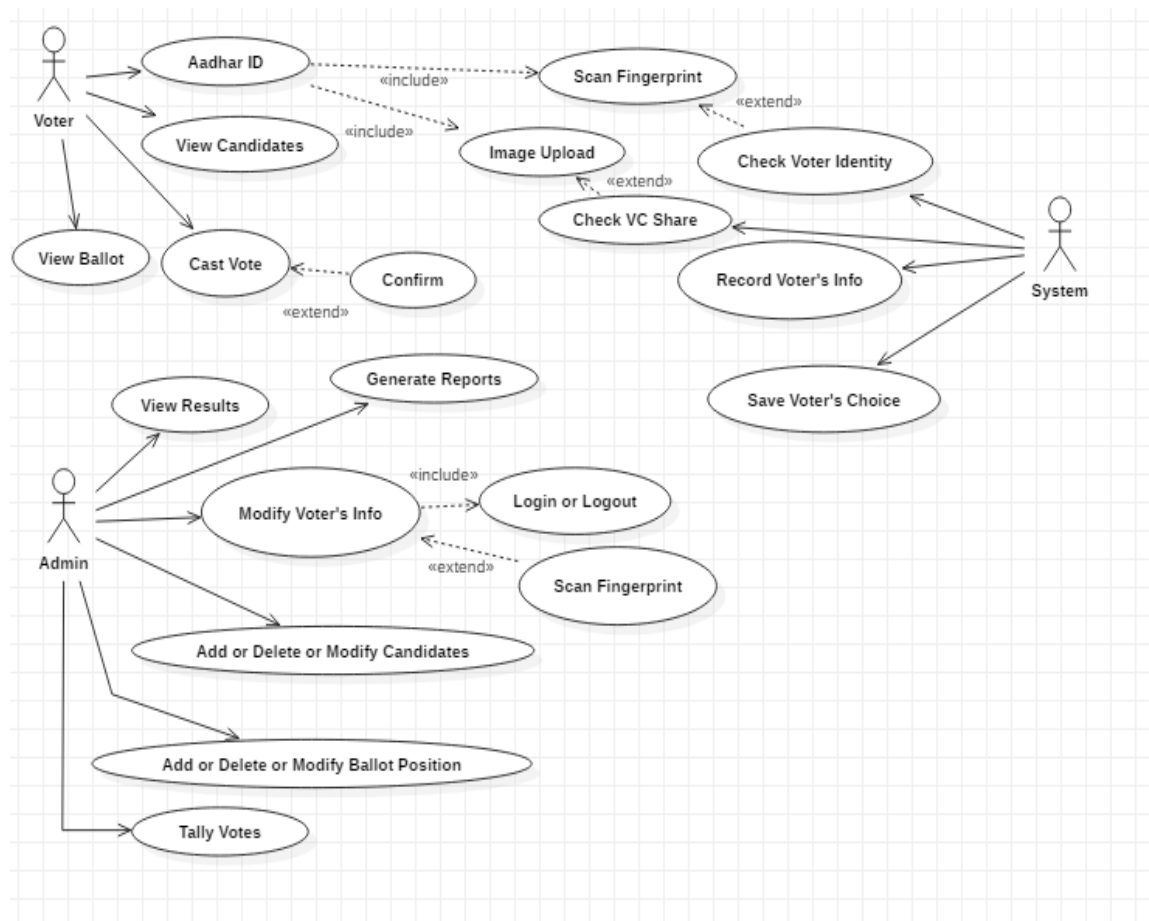


Fig 5.1: Use Case Diagram for Developed Model

5.1.2 Class Diagram

They are static representation of an application. Only the class diagrams have the capability to be directly mapped with the OOP Languages because in OOPs everything is model in the form of classes and objects. Because of this reason these diagrams are used widely at the time of construction. This is one of the most popularly used UML diagram in the designer community. A class diagram plays an essential role in forward and reverse engineering.

- a. It acts as a base for the component and deployment diagrams.
- b. It mainly describes and defines the basic responsibilities of a system's application.
- c. It implements the analysis and design view for a static application.

In a class diagram, each object is modelled as a class. Each class consists of section or compartments.

1. Class name
2. Attributes of a class or operations
3. Methods or functions
4. Documentation (optional section)

The following points ought to be recollected while drawing a class diagram:

- a. The name of the class diagram must be meaningful to portray the aspect of the framework.
- b. Each component and their connections must be distinguished ahead of time.
- c. Each class has a responsibility (attributes and methods) that must be identified clearly.
- d. Number of properties for each class must be minimum. Since pointless properties will make the diagram convoluted.
- e. At whatever point required to depict some part of the diagram use notes. Since toward the finish of the diagram it must be justifiable to the designer/coder.
- f. Before finalising the last version, the diagram must be drawn on plain paper and revise whatever number circumstances as would be prudent to make it redress.

1. **Scopes:**

The UML diagrams have two different types of scopes for class members:

- i. instance members scope and
- ii. classifier members scope

2. **Classifier members** are “static” members of a class in many programming languages. The scope is the class itself.

- i. Static attributes are common to all other objects that invoke the class.
- ii. Static methods are not instantiated.

3. **Instance members** are nothing but the members that are local to an object.

- i. The main purpose of instance members is to allow the objects to store their states.
- ii. Declarations outside the methods are usually known as instance members.

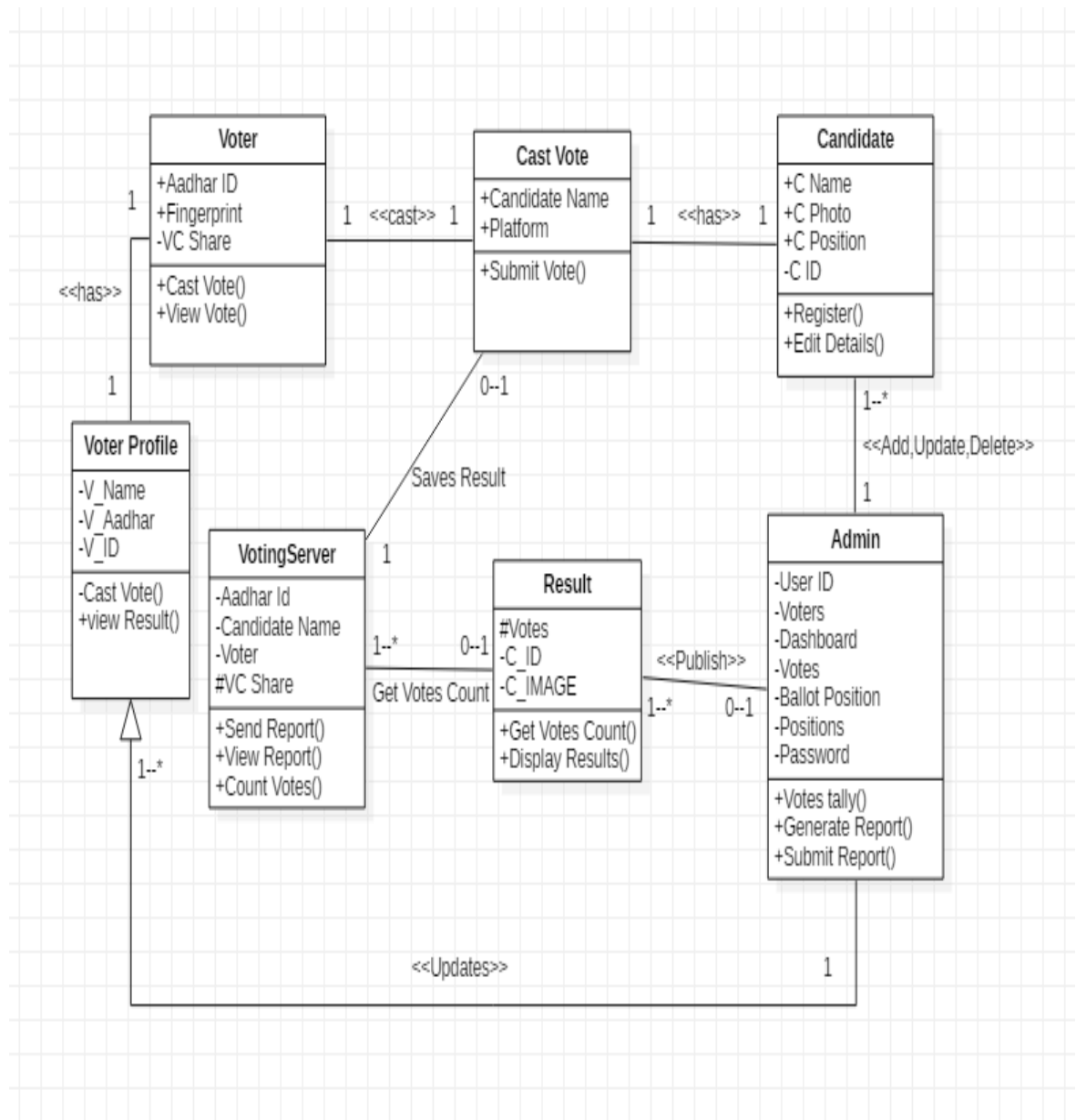


Fig 5.2: Class Diagram for Developed Model

5.1.3 Sequence Diagram

The Sequence Diagram depicts the time sequence among various objects in an application. It depicts the sequence of messages with which objects communicate with each other so that they carry out the required functionality.

It consists of the lifelines which are usually parallel vertical lines. It consists of horizontal arrows which indicates the direction of the messages that are exchanged in a proper order which makes the user easy to understand.

The lifeline for a given object represents a role. The synchronous calls are represented with the help of a solid arrow head whereas the asynchronous messages are represented with the help of open arrow heads.

All objects are represented according to their time ordering. Timing of messages plays a major role in sequence diagrams. An object is killed immediately after its use in sequence diagrams.

I). Common Properties:

An arrangement graph is much the same as unique sort of diagram and offers some indistinguishable properties from other diagrams. In any case, it varies from every single other diagram in its content.

II). Contents

Objects are normally named or unknown instances of class, however may likewise speak to occurrences of different things, for example components, collaboration and nodes. Graphically, object is represented as a rectangle by underlying its name.

III). Links

A link is a semantic association among objects i.e., an object of an affiliation is called as a connection. It is represented as a line.

IV). Messages

A message is a determination of a correspondence between objects that passes on the data with the desire that the action will follow.

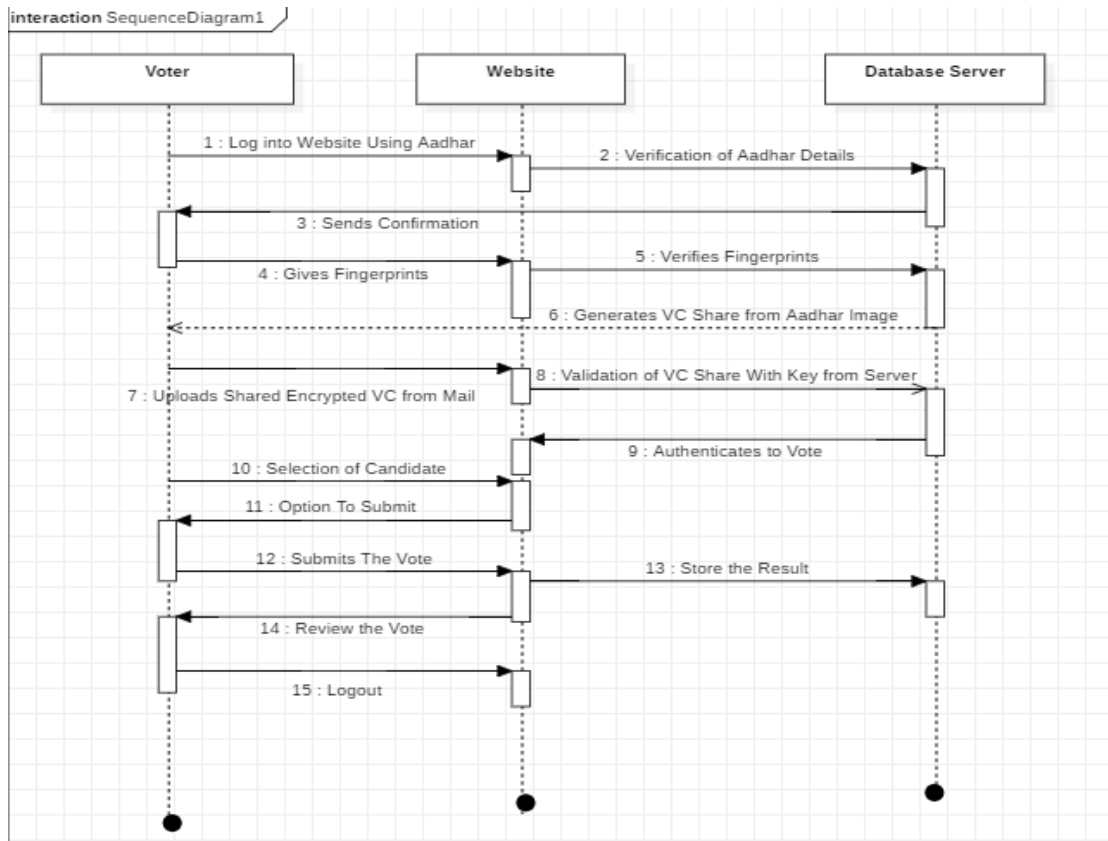


Fig 5.3: Voter's Sequence Diagram

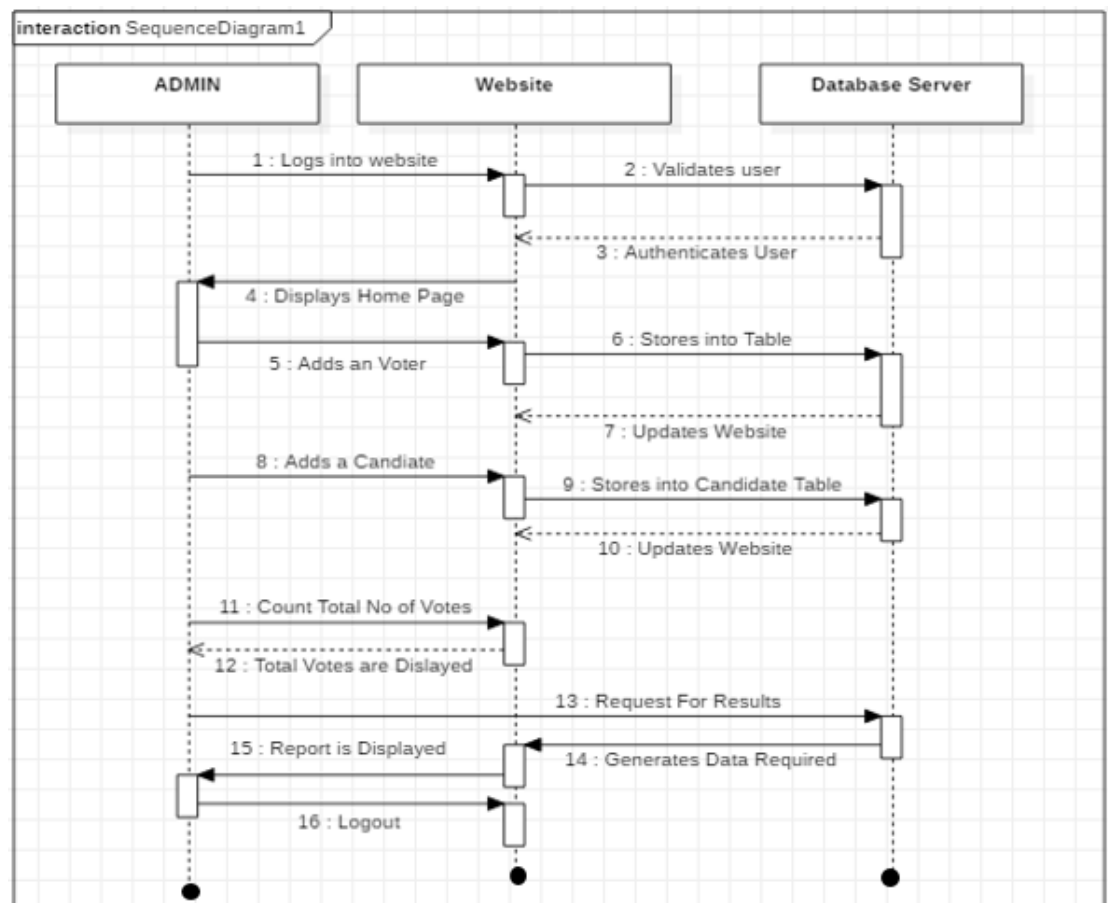


Fig 5.4: Admin's Sequence Diagram

5.4 Activity Diagram

The flow from one activity to another activity can be represented in the form of a flow chart which is usually an activity diagram. It forms a backbone for the UML diagrams.

It depicts the dynamic aspects for all the objects within the system.

The control flow from one object to another object is drawn which shows the basic operations that are to be performed.

Activity diagrams are constructed using the following:

Initial State or Start Point

A small filled circle followed by an arrow represents the initial action state or the start point for any activity diagram. For activity diagram using swimlanes, make sure the start point is placed in the top left corner of the first column.



Activity or Action State

An action state represents the non-interruptible action of objects. You can draw an action state in SmartDraw using a rectangle with rounded corners.

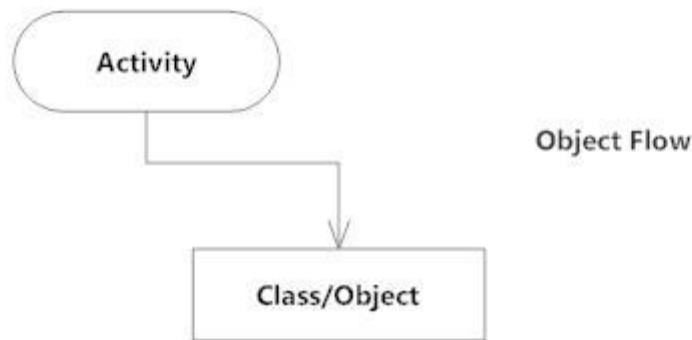


Action Flow

Action flows, also called edges and paths, illustrate the transitions from one action state to another. They are usually drawn with an arrowed line.

Object Flow

Object flow refers to the creation and modification of objects by activities. An object flow arrow from an action to an object means that the action creates or influences the object. An object flow arrow from an object to an action indicates that the action state uses the object.



Decisions and Branching

A diamond represents a decision with alternate paths. When an activity requires a decision prior to moving on to the next activity, add a diamond between the two activities. The outgoing alternates should be labeled with a condition or guard expression. You can also label one of the paths "else."



Guards

In UML, guards are a statement written next to a decision diamond that must be true before moving next to the next activity. These are not essential, but are useful when a specific answer, such as "Yes, three labels are printed," is needed before moving forward.



Final State or End Point

An arrow pointing to a filled circle nested inside another circle represents the final action state.



The basic purpose of an activity diagram is same as that of other UML diagrams.

The dynamic behaviour of the system is viewed by the activity diagram. They are used to construct a system using the backward and forward engineering mechanisms.

The purpose of an activity diagram is as follows:

- 1) For drawing the flow (i.e. activity) in a system.
- 2) For showing the flow of sequence from one activity to another activity.
- 1) For showing the concurrent and parallel flow of actions in the system. The elements which are used in an activity diagram are as follows:

- i) Association relationship
- ii) Activities
- iii) Conditions and Constraints.

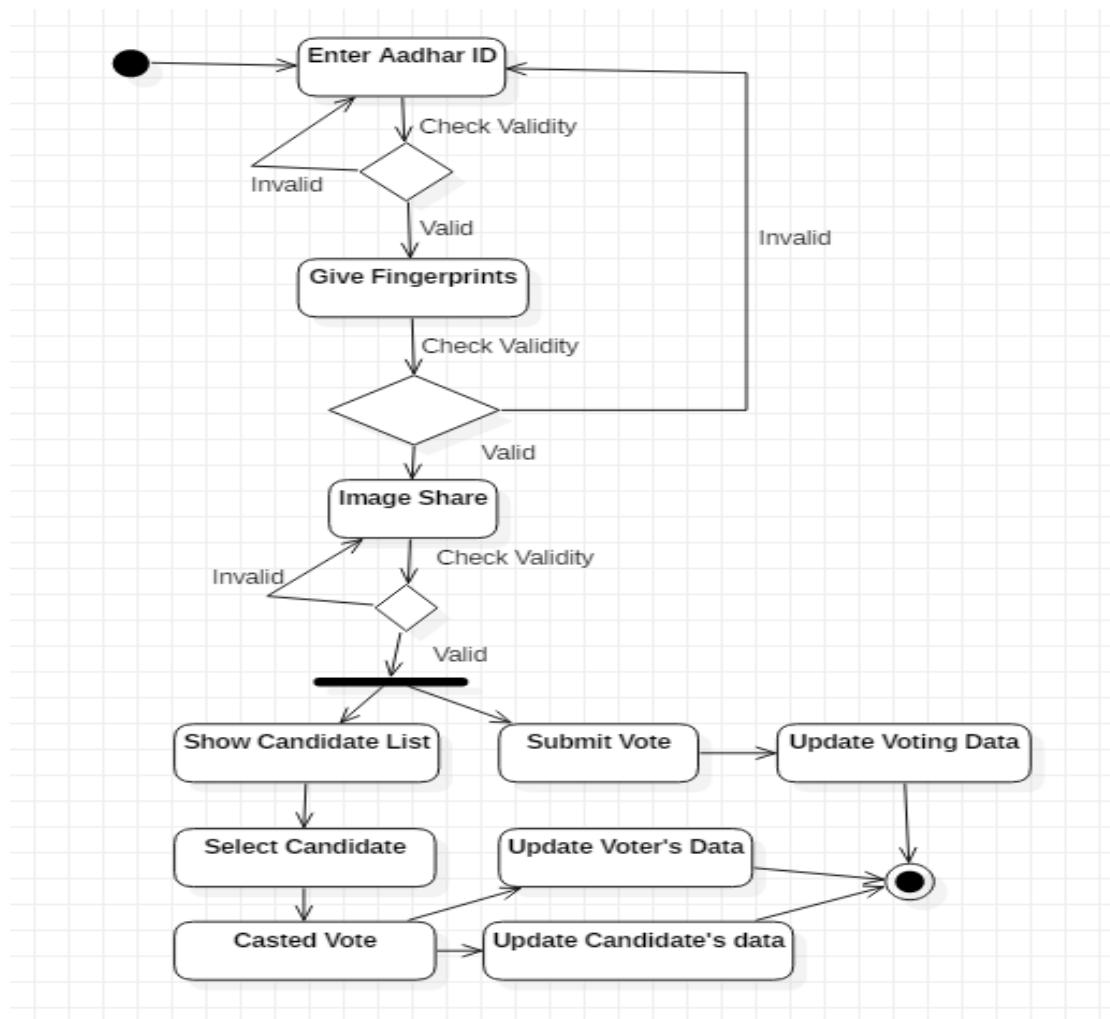


Fig 5.5: Voter's Activity Diagram

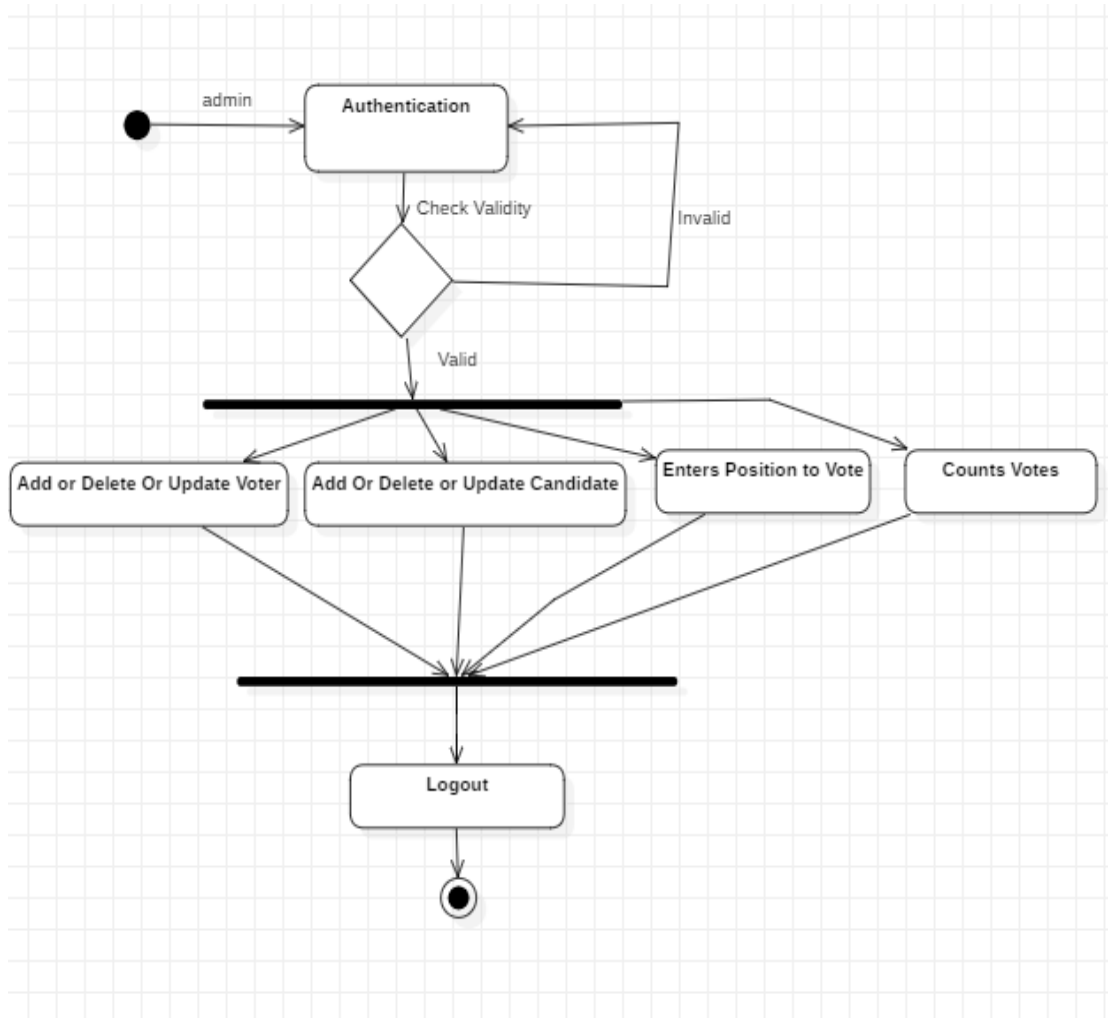


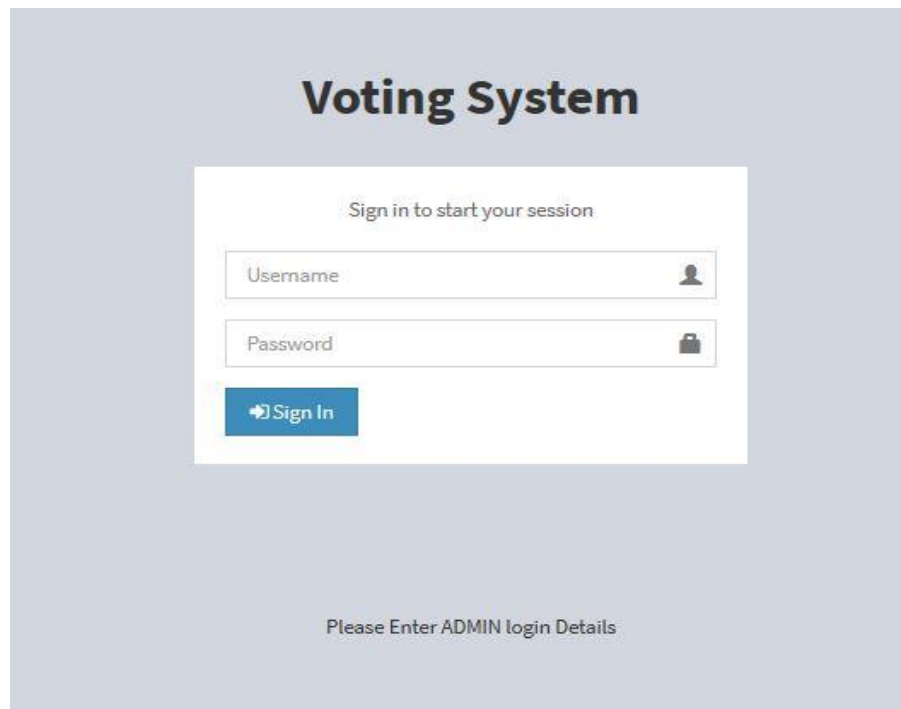
Fig 5.6: Admin's Activity Diagram

CHAPTER 6

IMPLEMENTATION

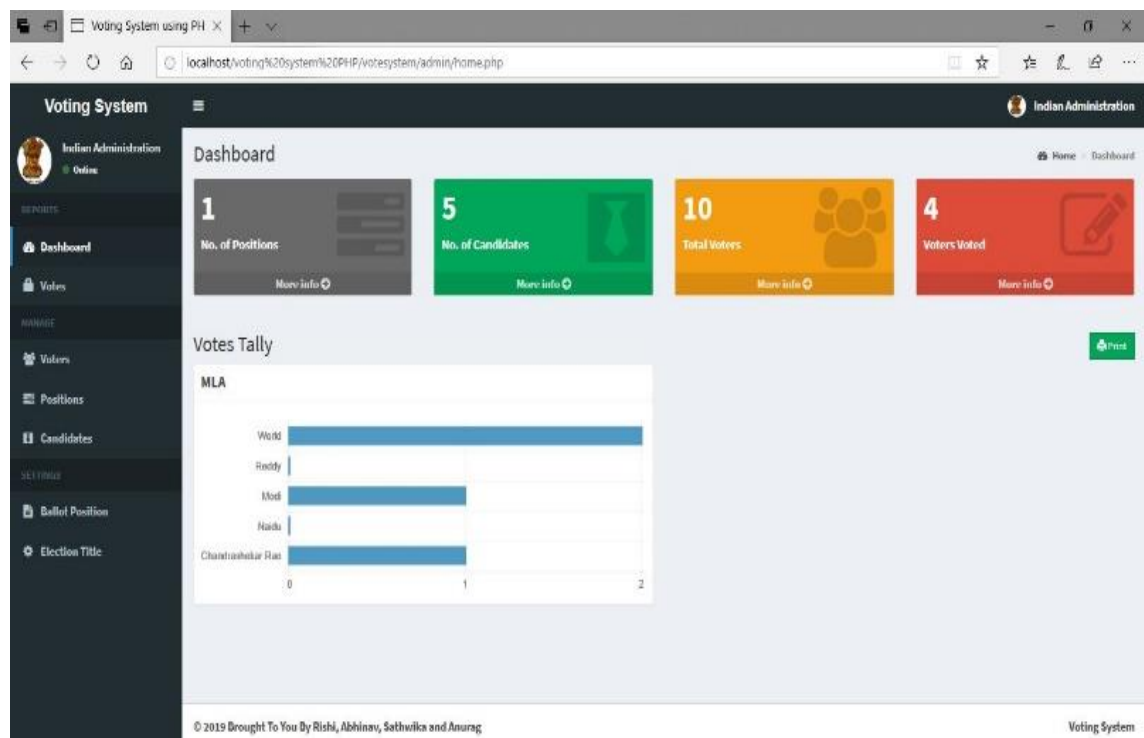
6.1 Admin's Module:

6.1.1 Entering into Admin Page



The image shows a login page for a 'Voting System'. The title 'Voting System' is at the top. Below it, a message says 'Sign in to start your session'. There are two input fields: 'Username' with a user icon and 'Password' with a lock icon. A blue 'Sign In' button is below the fields. At the bottom, a message says 'Please Enter ADMIN login Details'.

6.1.2 Dashboard to access all the required information



6.1.3 Creating the Election title to undergo election process

Configure ✕

Title

MLA Election

✕ Close

Save

6.1.4 Adding Position for a Candidate and Maximum vote to be cast

Add New Position ✕

Description

Maximum Vote

✕ Close

Save

6.1.5 Viewing the Position of a Candidate

Positions

[Home](#) > [Positions](#)

[+ New](#)

Show

10

 entries

Search:

Description	Maximum Vote	Tools
MLA	1	Edit Delete

Showing 1 to 1 of 1 entries

[Previous](#) [1](#) [Next](#)

6.1.6 Adding Required Candidates for election by adding their details as well as position of the candidate and description about their candidature

Note: Candidates information is disclosed to admin itself

×

Add New Candidate

Firstname

Lastname

Position

- Select -

▼

Photo

Browse...

Platform

✕ Close

Save

6.1.7 List of all the candidates

Candidates List

Home > Candidates

+ New

Show 10 entries

Search:

Position	Photo	Firstname	Lastname	Platform	Tools
MLA		Kalvakuntla	Chandrashekar Rao	View	Edit Delete
MLA		Chandrababu	Naidu	View	Edit Delete
MLA		Narendra	Modi	View	Edit Delete
MLA		Jagan Mohan	Reddy	View	Edit Delete
MLA		Hello	World	View	Edit Delete

Showing 1 to 5 of 5 entries

Previous

1

Next

6.1.8 Adding Voters with their Aadhar Details

×

Add New Voter

Name

Aadhar id

Password

Photo

Browse...

✕ Close

Save

6.1.9 List of all the Eligible Voters





































Voters List

Home > Voter

+ New

Show 10 entries

Search:

Aadharid	Name	Photo	Voters ID	Tools
123456789	Sathwika		 8eXipALiYZJtq2s	 Edit  Delete
234567891	Rishi Kumar		 RHj6W9tiSEIsuao	 Edit  Delete
256589556	ABCDEF		 M1wKZGiFOpVlgz9	 Edit  Delete
345678912	Pranav		 X37fKZo28rGJ9VC	 Edit  Delete
456789123	Anoushk		 r4MXlQmjREK5tko	 Edit  Delete
567891234	Rikey		 ggibQ9KBOXCN5GI	 Edit  Delete
678912345	Rohan		 eZljg75EDYGf4aH	 Edit  Delete
789123456	Al Pacino		 9ZWbzHlo64jSIGO	 Edit  Delete
890123456	Anurag		 3ofnbydH1CcAODI	 Edit  Delete

Showing 1 to 9 of 9 entries

Previous 1 Next

6.1.10 Ballot Position Verification to proceed to Election


Ballot Position


Home > Ballot Preview

MLA

Select only one candidate


☐


 Platform



Kalvakuntla Chandrashekar Rao


☐


 Platform



Chandrababu Naidu


☐


 Platform



Narendra Modi


☐


 Platform




Jagan Mohan Reddy

☐

 Platform



Hello World

 Reset

6.1.11 List of all the voters who have voted in the Election

Votes Home > Votes

[Reset](#)

Show entries Search:

Position	Candidate	Voter
MLA	Hello World	Al Pacino 789123456
MLA	Narendra Modi	Rohan 678912345
MLA	Kalvakuntla Chandrashekar Rao	Anurag 890123456
MLA	Hello World	ABCDEF 256589556
MLA	Narendra Modi	Jaffa 123456780

Showing 1 to 5 of 5 entries Previous **1** Next

6.1.12 tallying and generating Reports for the election result

MLA Election

Tally Result

MLA	
Candidates	Votes
Chandrashekar Rao, Kalvakuntla	1
Modi, Narendra	2
Naidu, Chandrababu	0
Reddy, Jagan Mohan	0
World, Hello	2

6.2 Voter's Module

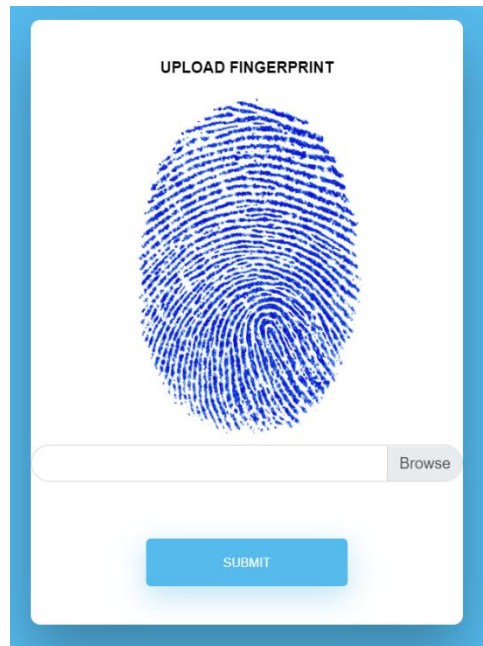
6.2.1 Login

ONLINE VOTING SYSTEM

AADHAR

LOG IN

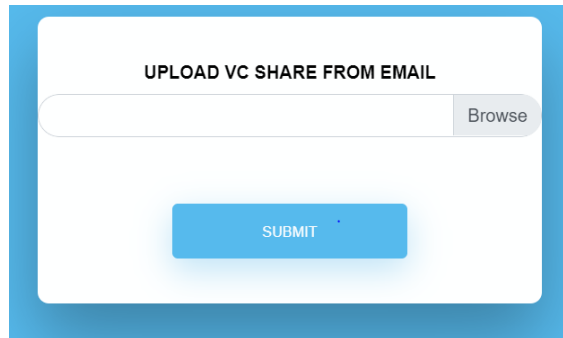
6.2.2 Fingerprint Verification for Voting (First Step Authentication)



UPLOAD FINGERPRINT

A large blue fingerprint icon is centered on the screen. Below it is a text input field with a "Browse" button to its right. At the bottom is a blue "SUBMIT" button.

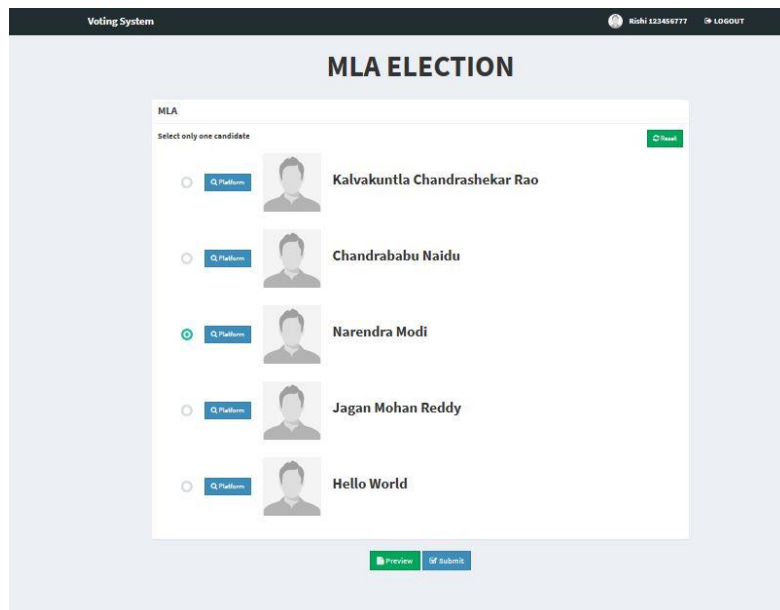
6.2.3 Upload of share from Email to authenticate voting (2 Step Verification)



UPLOAD VC SHARE FROM EMAIL

A text input field with a "Browse" button to its right. Below it is a blue "SUBMIT" button.

6.2.4 Voter allowed to cast and submit vote after successful verification








Voting System 123456777 [LOGOUT](#)

MLA ELECTION

MLA

Select only one candidate

- ☐ [Q Platform](#)  Kalvakuntla Chandrashekar Rao
- ☐ [Q Platform](#)  Chandrababu Naidu
- ☒ [Q Platform](#)  Narendra Modi
- ☐ [Q Platform](#)  Jagan Mohan Reddy
- ☐ [Q Platform](#)  Hello World

[Preview](#) [Submit](#)

6.2.5 Saved Ballot after voting the candidate the voter elected



6.2.6 Viewing the vote by entering into the ballot



6.3 Coding (Back-End)

6.3.1 Fingerprint Accessibility code

```
package com.hh.fingerprint

import java.sql.{ResultSet, Statement}

/**
 * @author ${user.name}
 */
object FingerPrintMain extends App {

    val url = "jdbc:sqlite:/G:/FingerPrintAFIS/FingerPrint.db"

    val conn = Connect.connect(url)

    val loader = LoadDB(conn)

    val sql = s"SELECT distinct id , fingerprint FROM users limit 4 "

    val stmt: Statement = conn.createStatement()
    val result: ResultSet = stmt.executeQuery(sql)

    var stack : List[String] = Nil

    while(result.next())
    {

        val user = result.getString("id")
        val fingerprint = result.getString("fingerprint")
        println(s"Check for user $user")
    }
}
```

```

        val (i1 ,i2 , i3 ) = loader.loadImageFromDBAndGenShares(user)
        println( "Check Fingerprint : " + new
FingerPrintMatch(90).matchFingerprintFromDatabase(user , fingerprint)(conn))
        println( "Check Shares : "+loader.decryptAndMatch( i1 , i2)( i3))

    }

}

```

6.3.2 Email for authentication of share

```

package com.hh.fingerprint

import javax.mail.{Authenticator, Message, PasswordAuthentication, Session,
Transport}
import javax.mail.internet.{InternetAddress, MimeMessage}

class EmailForAuthShare {

    def sendEmail( to : String , from : String )( username : String , password
: String ): Unit =
    {

        val prop = System.getProperties
        prop.setProperty("mail.smtp.host", to)
        // prop.setProperty("mail.smtp.host", "smtp.gmail.com")
        // prop.put("mail.smtp.port", "465")
        prop.put("mail.smtp.port", "25")
        // prop.put("mail.smtp.auth", "true")
        // prop.put("mail.smtp.socketFactory.port", "465")
        // prop.put("mail.smtp.socketFactory.class",
"javax.net.ssl.SSLSocketFactory")
        val session = Session.getDefaultInstance(prop , null
//      , new Authenticator() {
//          override def getPasswordAuthentication: PasswordAuthentication =
new PasswordAuthentication(username , password)
//      }
//    )
        val message = new MimeMessage(session)
        message.setFrom(new InternetAddress(from))
        // message.setRecipients(Message.RecipientType.TO , to.mkString(","))
        message.addRecipient(Message.RecipientType.TO,new InternetAddress(to))
        message.setSubject(s"Authentication - $username")
        message.setText("Hello, this email contains image of key for auth ")

        // Send message
        Transport.send(message)
        System.out.println("message sent successfully....")

    }

}

```

6.3.3 Fingerprint match verification code

```

package com.hh.fingerprint
import java.sql.Connection

import com.machinezoo.sourceafis.{FingerprintMatcher, FingerprintTemplate}

```



```

class FingerPrintMatch( threshold : Double ) {

    def matchFingerPrint( json1 : String , json2 : String ): Double = {
        val probe: FingerprintTemplate = new
FingerprintTemplate().deserialize(json1)
        val candidate = new FingerprintTemplate().deserialize(json2)
        val score = new FingerprintMatcher().index(probe).`match`(candidate)
        score
    }

    def matchFingerPrintFromDatabase(user : String , json : String )(conn :
Connection): Boolean =
    {
        val sql = s"SELECT id, email , finger_id , scan_id , fingerprint , image
FROM users where id = '$user' "

        val stmt = conn.createStatement()
        val result = stmt.executeQuery(sql)

        var stack : List[( Int , Double)] = Nil

        while(result.next())
        {

            val json_query = result.getString(5)
            val finger_id = result.getInt(3)
            stack = stack :+ (finger_id , matchFingerPrint( json , json_query ))

        }

        val possible_match = stack.maxBy[Double](_. _2)

        println(s"Possible Match for $user : $possible_match")

        possible_match._2 > threshold

    }
}

```

6.3.4 Loading the Database from backend server

```

package com.hh.fingerprint
import java.awt.image.BufferedImage
import java.io.{ByteArrayInputStream, ByteArrayOutputStream, File}
import java.nio.file.{Files, Paths}
import java.sql.Connection

import com.machinezoo.sourceafis.FingerprintTemplate
import javax.imageio.ImageIO

import scala.util._

case class LoadDB(conn : Connection) {

    def load( dir : String ): Unit =
    {

        val directory = new File(dir)
        Try{
            directory.listFiles().foreach( f => {

                val fileName = f.getName.split('.').head

```

```

println(s"load $fileName")

if(fileName == "readme")
    println(s"ignored")
else{

    val Array( id , finger_id , scan_id ) = fileName.split("_")
    val email = "BLANK"

    val fingerprint= new
FingerprintTemplate().dpi(500).create(Files.readAllBytes(Paths.get(f.getAbsolutePath))).serialize()

    val img_bmp = ImageIO.read(f)
    val baos = new ByteArrayOutputStream
    ImageIO.write( img_bmp, "bmp", baos)

    val img = baos.toByteArray
    insertIntoDB( UserDetails ( id , email , finger_id.toInt ,
scan_id.toInt ,fingerprint , img ), conn)

    }

    })

} match {
    case Success(_) => println(s"Success")
    case Failure(e) => println(s"Failed : "+ e.getMessage)
}

}

def insertIntoDB( userDetails: UserDetails , conn : Connection): Unit =
{
    val sql = "INSERT INTO
users(id,email,finger_id,scan_id,fingerprint,image) VALUES(?,?,?,?,?,?)"

    val pstmt = conn.prepareStatement(sql)
    Try{

        pstmt.setString(1, userDetails.id)
        pstmt.setString(2, userDetails.email)
        pstmt.setInt(3 , userDetails.finger_id)
        pstmt.setInt(4 , userDetails.scan_id)
        pstmt.setString(5 , userDetails.fingerprint)
        pstmt.setBytes(6 , userDetails.image)

        pstmt.executeUpdate
    } match {
        case Success(_) => println("Update Success")
        case Failure(e) => {
            println("Update Failed "+ e.getMessage)
            throw e
        }
    }

}

def loadImageFromDBAndGenShares( user : String ): (BufferedImage,
BufferedImage, BufferedImage) =

```

```

{
    val sql = s"SELECT id, email , finger_id , scan_id , fingerprint , image
FROM users where id = '$user' "

    val stmt = conn.createStatement()
    val result = stmt.executeQuery(sql)

    var count = 0
    var img : BufferedImage = null
    var imgKey : BufferedImage = null
    var imgEnc : BufferedImage = null
    while(result.next() && count == 0)
    {

        val img_bytes = result.getBytes(6)
        val inputStream = new ByteArrayInputStream(img_bytes)
        img = ImageIO.read( inputStream )

        val imgSrc: BufferedImage = Crypting.CheckSource( img , 0 , 0 , false)

        imgKey = Crypting.generateKey(imgSrc.getWidth, imgSrc.getHeight)
        imgEnc = Crypting.encryptImage(imgKey, imgSrc)

        count = count + 1
    }

    (imgKey , imgEnc , img)
}

def decryptAndMatch( key : BufferedImage , enc : BufferedImage )( img :
BufferedImage): Boolean =
{

    val imgKey = Crypting.CheckEncrImg(key)
    val imgEnc = Crypting.CheckEncrImg(enc)
    val imgOverlay = Crypting.overlayImages(imgKey, imgEnc)
    val imgClean = Crypting.decryptImage(imgOverlay)

    val width1 = img.getWidth
    val width2 = imgClean.getWidth
    val height1 = img.getHeight
    val height2 = imgClean.getHeight

    if ((width1 != width2) || (height1 != height2))
        false
    else {
        var difference = 0
        var y = 0
        while ( {
            y < height1
        }) {
            var x = 0
            while ( {
                x < width1
            }) {
                val rgbA = img.getRGB(x, y)
                val rgbB = imgClean.getRGB(x, y)
                val redA = (rgbA >> 16) & 0xff
                val greenA = (rgbA >> 8) & 0xff
                val blueA = rgbA & 0xff
                val redB = (rgbB >> 16) & 0xff
                val greenB = (rgbB >> 8) & 0xff
                val blueB = rgbB & 0xff
                difference += Math.abs(redA - redB)
                difference += Math.abs(greenA - greenB)
                difference += Math.abs(blueA - blueB)
            }
        }
    }
}

```

```

        x += 1; x - 1
    }
}

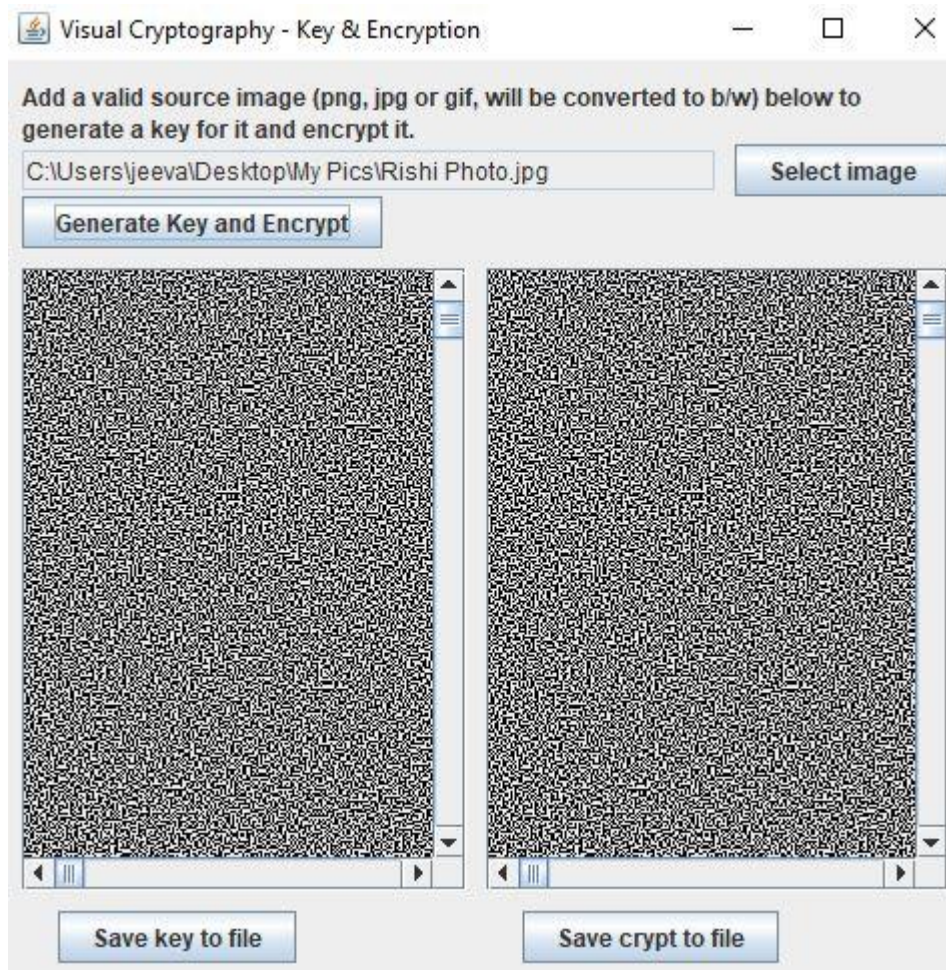
{
    y += 1; y - 1
}
}
// Total number of red pixels = width * height
// Total number of blue pixels = width * height
// Total number of green pixels = width * height
// So total number of pixels = width * height * 3
val total_pixels = width1 * height1 * 3
// Normalizing the value of different pixels
// for accuracy(average pixels per color
// component)
val avg_different_pixels = difference / total_pixels
// There are 255 values of pixels in total
val percentage: Int = (avg_different_pixels / 255) * 100

percentage == 0

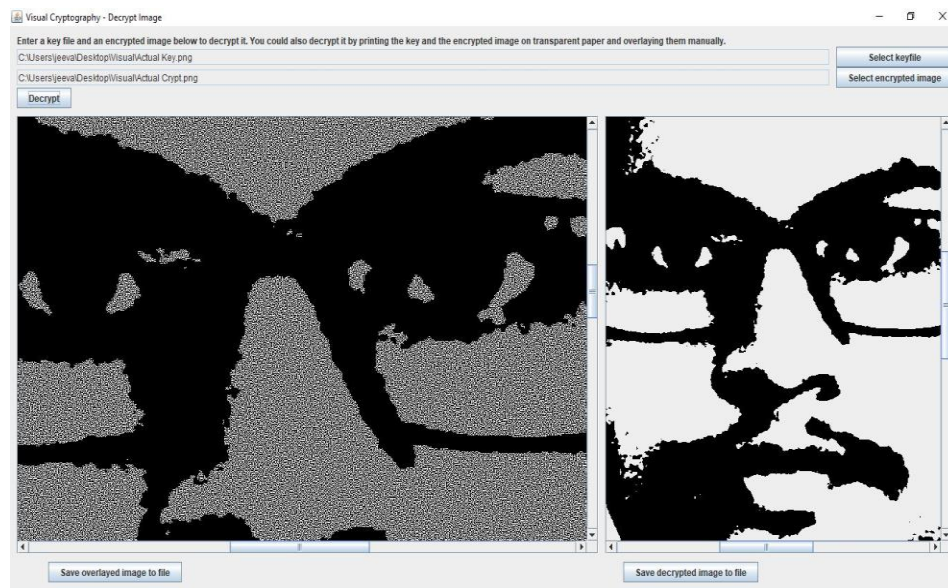
}
}

```

6.3.5 Java jar file for encryption of voter image and generation of key



6.3.6 Decryption of image by overlapping the shares



6.4. Coding (Front End)

6.4.1 Ballot.php:

```
<?php include 'includes/session.php'; ?>
<?php include 'includes/header.php'; ?>
<body class="hold-transition skin-blue sidebar-mini">
<div class="wrapper">

<?php include 'includes/navbar.php'; ?>
<?php include 'includes/menubar.php'; ?>

<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
<!-- Content Header (Page header) -->
<section class="content-header">
<h1>
Ballot Position
</h1>
<ol class="breadcrumb">
<li><a href="#"><i class="fa fa-dashboard"></i> Home</a></li>
<li class="active">Ballot Preview</li>
</ol>
</section>
<!-- Main content -->
<section class="content">
<?php
if(isset($_SESSION['error'])) {
echo "
<div class='alert alert-danger alert-dismissible'>
```

```

<button      type='button'      class='close'      data-dismiss='alert'      aria-
hidden='true'>&times;</button>
<h4><i class='icon fa fa-warning'></i> Error!</h4>
"$_SESSION['error']."
</div>
";
unset($_SESSION['error']);
}
if(isset($_SESSION['success'])) {
echo "
<div class='alert alert-success alert-dismissible'>
<button      type='button'      class='close'      data-dismiss='alert'      aria-
hidden='true'>&times;</button>
<h4><i class='icon fa fa-check'></i> Success!</h4>
"$_SESSION['success']."
</div>
";
unset($_SESSION['success']);
}
?>

<div class="row">
<div class="col-xs-10 col-xs-offset-1" id="content">
</div>
</div>

</section>

</div>

<?php include 'includes/footer.php'; ?>
</div>
<?php include 'includes/scripts.php'; ?>
<script>
$(function(){
fetch();

$(document).on('click', '.reset', function(e){
e.preventDefault();
var desc = $(this).data('desc');
$('#'+desc).iCheck('uncheck');
});

$(document).on('click', '.moveup', function(e){
e.preventDefault();
var id = $(this).data('id');
$('#'+id).animate({
'marginTop' : "-300px"
});
});

```

```

$.ajax({
type: 'POST',
url: 'ballot_up.php',
data:{id:id},
dataType: 'json',
success: function(response){
if(!response.error){
fetch();
}
}
});
});

$(document).on('click', '.movedown', function(e){
e.preventDefault();
var id = $(this).data('id');
$('#'+id).animate({
'marginTop' : "+300px"
});
$.ajax({
type: 'POST',
url: 'ballot_down.php',
data:{id:id},
dataType: 'json',
success: function(response){
if(!response.error){
fetch();
}
}
});
});

});

function fetch(){
$.ajax({
type: 'POST',
url: 'ballot_fetch.php',
dataType: 'json',
success: function(response){
$('#content').html(response).iCheck({checkboxClass:'icheckbox_flat-green',radioClass: 'iradio_flat-green'});
}
});
}
</script>
</body>
</html>

```

6.4.2 Candidates.php:

```
<?php include 'includes/session.php'; ?>
<?php include 'includes/header.php'; ?>
<body class="hold-transition skin-blue sidebar-mini">
<div class="wrapper">

    <?php include 'includes/navbar.php'; ?>
    <?php include 'includes/menubar.php'; ?>

    <!-- Content Wrapper. Contains page content -->
    <div class="content-wrapper">
        <!-- Content Header (Page header) -->
        <section class="content-header">
            <h1>
                Candidates List
            </h1>
            <ol class="breadcrumb">
                <li><a href="#"><i class="fa fa-dashboard"></i> Home</a></li>
                <li class="active">Candidates</li>
            </ol>
        </section>
        <!-- Main content -->
        <section class="content">
            <?php
                if(isset($_SESSION['error'])) {
                    echo "
                        <div class='alert alert-danger alert-dismissible'>
                            <button type='button' class='close' data-dismiss='alert' aria-
hidden='true'>&times;</button>
                            <h4><i class='icon fa fa-warning'></i> Error!</h4>
                            \"$_SESSION['error']\"
                        </div>
                    ";
                    unset($_SESSION['error']);
                }
                if(isset($_SESSION['success'])) {
                    echo "
                        <div class='alert alert-success alert-dismissible'>
                            <button type='button' class='close' data-dismiss='alert' aria-
hidden='true'>&times;</button>
                            <h4><i class='icon fa fa-check'></i> Success!</h4>
                            \"$_SESSION['success']\"
                        </div>
                    ";
                    unset($_SESSION['success']);
                }
            ?>
            <div class="row">
                <div class="col-xs-12">
```



```

<div class="box">
  <div class="box-header with-border">
    <a href="#addnew" data-toggle="modal" class="btn btn-primary btn-sm btn-
flat"><i class="fa fa-plus"></i> New</a>
  </div>
  <div class="box-body">
    <table id="example1" class="table table-bordered">
      <thead>
        <th class="hidden"></th>
        <th>Position</th>
        <th>Photo</th>
        <th>Firstname</th>
        <th>Lastname</th>
        <th>Platform</th>
        <th>Tools</th>
      </thead>
      <tbody>
        <?php
          $sql = "SELECT *, candidates.id AS canid FROM candidates LEFT JOIN
positions ON positions.id=candidates.position_id ORDER BY positions.priority
ASC";

          $query = $conn->query($sql);
          while($row = $query->fetch_assoc()){
            $image = (!empty($row['photo'])) ? '../images/'.$row['photo'] :
'../images/profile.jpg';
            echo "
              <tr>
                <td class='hidden'></td>
                <td>".$row['description'].</td>
                <td>
                  <img src='".$image.'" width='30px' height='30px'>
                  <a href='#edit_photo' data-toggle='modal' class='pull-right photo'
data-id='".$row['canid']."'><span class='fa fa-edit'></span></a>
                </td>
                <td>".$row['firstname'].</td>
                <td>".$row['lastname'].</td>
                <td><a href='#platform' data-toggle='modal' class='btn btn-info btn-
sm btn-flat platform' data-id='".$row['canid']."'><i class='fa fa-search'></i>
View</a></td>
                <td>
                  <button class='btn btn-success btn-sm edit btn-flat' data-
id='".$row['canid']."'><i class='fa fa-edit'></i> Edit</button>
                  <button class='btn btn-danger btn-sm delete btn-flat' data-
id='".$row['canid']."'><i class='fa fa-trash'></i> Delete</button>
                </td>
              </tr>
            ";
          }
        ?>
      </tbody>

```

```

        </table>
    </div>
</div>
</div>
</div>
</section>
</div>

<?php include 'includes/footer.php'; ?>
<?php include 'includes/candidates_modal.php'; ?>
</div>
<?php include 'includes/scripts.php'; ?>
<script>
$(function(){
    $(document).on('click', '.edit', function(e){
        e.preventDefault();
        $('#edit').modal('show');
        var id = $(this).data('id');
        getRow(id);
    });

    $(document).on('click', '.delete', function(e){
        e.preventDefault();
        $('#delete').modal('show');
        var id = $(this).data('id');
        getRow(id);
    });

    $(document).on('click', '.photo', function(e){
        e.preventDefault();
        var id = $(this).data('id');
        getRow(id);
    });

    $(document).on('click', '.platform', function(e){
        e.preventDefault();
        var id = $(this).data('id');
        getRow(id);
    });

    });

function getRow(id){
    $.ajax({
        type: 'POST',
        url: 'candidates_row.php',
        data: {id:id},
        dataType: 'json',
        success: function(response){
            $('#id').val(response.canid);

```

```

        $('#edit_firstname').val(response.firstname);
        $('#edit_lastname').val(response.lastname);
        $('#posselect').val(response.position_id).html(response.description);
        $('#edit_platform').val(response.platform);
        $('#fullname').html(response.firstname+' '+response.lastname);
        $('#desc').html(response.platform);
    }
});
}
</script>
</body>
</html>

```

6.4.3 Home.php

```

<?php include 'includes/session.php'; ?>
<?php include 'includes/slugify.php'; ?>
<?php include 'includes/header.php'; ?>
<body class="hold-transition skin-blue sidebar-mini">
<div class="wrapper">

    <?php include 'includes/navbar.php'; ?>
    <?php include 'includes/menubar.php'; ?>

    <!-- Content Wrapper. Contains page content -->
    <div class="content-wrapper">
        <!-- Content Header (Page header) -->
        <section class="content-header">
            <h1>
                Dashboard
            </h1>
            <ol class="breadcrumb">
                <li><a href="#"><i class="fa fa-dashboard"></i> Home</a></li>
                <li class="active">Dashboard</li>
            </ol>
        </section>

        <!-- Main content -->
        <section class="content">
            <?php
                if(isset($_SESSION['error'])){
                    echo "
                        <div class='alert alert-danger alert-dismissible'>
                            <button type='button' class='close' data-dismiss='alert' aria-
                                hidden='true'>&times;</button>
                            <h4><i class='icon fa fa-warning'></i> Error!</h4>
                            \"$_SESSION['error'].\"
                        </div>
                    ";
                    unset($_SESSION['error']);
                }
            </?php>

```

```

    }
    if(isset($_SESSION['success'])) {
        echo "
            <div class='alert alert-success alert-dismissible'>
                <button type='button' class='close' data-dismiss='alert' aria-
hidden='true'>&times;</button>
                <h4><i class='icon fa fa-check'></i> Success!</h4>
                \"$_SESSION['success'].\"
            </div>
        ";
        unset($_SESSION['success']);
    }
?>
<!-- Small boxes (Stat box) -->
<div class="row">
    <div class="col-lg-3 col-xs-6">
        <!-- small box -->
        <div class="small-box bg-aqua">
            <div class="inner">
                <?php
                    $sql = "SELECT * FROM positions";
                    $query = $conn->query($sql);

                    echo "<h3>\".$query->num_rows.\"</h3>";
                ?>

                <p>No. of Positions</p>
            </div>
            <div class="icon">
                <i class="fa fa-tasks"></i>
            </div>
            <a href="positions.php" class="small-box-footer">More info <i class="fa fa-
arrow-circle-right"></i></a>
        </div>
    </div>
    <!-- ./col -->
    <div class="col-lg-3 col-xs-6">
        <!-- small box -->
        <div class="small-box bg-green">
            <div class="inner">
                <?php
                    $sql = "SELECT * FROM candidates";
                    $query = $conn->query($sql);

                    echo "<h3>\".$query->num_rows.\"</h3>";
                ?>

                <p>No. of Candidates</p>
            </div>
            <div class="icon">

```

```

        <i class="fa fa-black-tie"></i>
    </div>
    <a href="candidates.php" class="small-box-footer">More info <i class="fa fa-
arrow-circle-right"></i></a>
</div>
</div>
<!-- ./col -->
<div class="col-lg-3 col-xs-6">
    <!-- small box -->
    <div class="small-box bg-yellow">
        <div class="inner">
            <?php
                $sql = "SELECT * FROM voters";
                $query = $conn->query($sql);

                echo "<h3>".$query->num_rows."</h3>";
            ?>

            <p>Total Voters</p>
        </div>
        <div class="icon">
            <i class="fa fa-users"></i>
        </div>
        <a href="voters.php" class="small-box-footer">More info <i class="fa fa-
arrow-circle-right"></i></a>
    </div>
</div>
<!-- ./col -->
<div class="col-lg-3 col-xs-6">
    <!-- small box -->
    <div class="small-box bg-red">
        <div class="inner">
            <?php
                $sql = "SELECT * FROM votes GROUP BY voters_id";
                $query = $conn->query($sql);

                echo "<h3>".$query->num_rows."</h3>";
            ?>

            <p>Voters Voted</p>
        </div>
        <div class="icon">
            <i class="fa fa-edit"></i>
        </div>
        <a href="votes.php" class="small-box-footer">More info <i class="fa fa-
arrow-circle-right"></i></a>
    </div>
</div>
<!-- ./col -->
</div>

```

```

<div class="row">
  <div class="col-xs-12">
    <h3>Votes Tally
    <span class="pull-right">
      <a href="print.php" class="btn btn-success btn-sm btn-flat"><span
class="glyphicon glyphicon-print"></span> Print</a>
    </span>
    </h3>
  </div>
</div>

<?php
$sql = "SELECT * FROM positions ORDER BY priority ASC";
$query = $conn->query($sql);
$inc = 2;
while($row = $query->fetch_assoc()){
  $inc = ($inc == 2) ? 1 : $inc+1;
  if($inc == 1) echo "<div class='row'>";
  echo "
    <div class='col-sm-6'>
      <div class='box box-solid'>
        <div class='box-header with-border'>
          <h4 class='box-title'><b>".$row['description']."</b></h4>
        </div>
        <div class='box-body'>
          <div class='chart'>
            <canvas
style='height:200px'></canvas>
          </div>
        </div>
      </div>
    </div>
    <div class='col-sm-6'>
      <div class='box box-solid'>
        <div class='box-header with-border'>
          <h4 class='box-title'><b>".$row['description']."</b></h4>
        </div>
        <div class='box-body'>
          <div class='chart'>
            <canvas
id='".slugify($row['description'])."'
style='height:200px'></canvas>
          </div>
        </div>
      </div>
    </div>
  ";
  if($inc == 2) echo "</div>";
}
if($inc == 1) echo "<div class='col-sm-6'></div></div>";
?>

</section>
<!-- right col -->
</div>
<?php include 'includes/footer.php'; ?>

</div>
<!-- ./wrapper -->

<?php include 'includes/scripts.php'; ?>
<?php
$sql = "SELECT * FROM positions ORDER BY priority ASC";

```

```

$query = $conn->query($sql);
while($row = $query->fetch_assoc()){
    $sql = "SELECT * FROM candidates WHERE position_id = '".$row['id']."'";
    $query = $conn->query($sql);
    $array = array();
    $varray = array();
    while($crow = $query->fetch_assoc()){
        array_push($array, $crow['lastname']);
        $sql = "SELECT * FROM votes WHERE candidate_id = '".$crow['id']."'";
        $vquery = $conn->query($sql);
        array_push($varray, $vquery->num_rows);
    }
    $array = json_encode($array);
    $varray = json_encode($varray);
    ?>
<script>
$(function(){
    var rowid = '<?php echo $row['id']; ?>';
    var description = '<?php echo slugify($row['description']); ?>';
    var barChartCanvas = $('#'+description).get(0).getContext('2d')
    var barChart = new Chart(barChartCanvas)
    var barChartData = {
        labels : <?php echo $array; ?>,
        datasets: [
            {
                label      : 'Votes',
                fillColor   : 'rgba(60,141,188,0.9)',
                strokeColor : 'rgba(60,141,188,0.8)',
                pointColor   : '#3b8bba',
                pointStrokeColor : 'rgba(60,141,188,1)',
                pointHighlightFill : '#fff',
                pointHighlightStroke: 'rgba(60,141,188,1)',
                data         : <?php echo $varray; ?>
            }
        ]
    }
    var barChartOptions = {
        //Boolean - Whether the scale should start at zero, or an order of magnitude down
        from the lowest value
        scaleBeginAtZero : true,
        //Boolean - Whether grid lines are shown across the chart
        scaleShowGridLines : true,
        //String - Colour of the grid lines
        scaleGridLineColor : 'rgba(0,0,0,.05)',
        //Number - Width of the grid lines
        scaleGridLineWidth : 1,
        //Boolean - Whether to show horizontal lines (except X axis)
        scaleShowHorizontalLines: true,
        //Boolean - Whether to show vertical lines (except Y axis)
        scaleShowVerticalLines : true,

```

```

//Boolean - If there is a stroke on each bar
barShowStroke      : true,
//Number - Pixel width of the bar stroke
barStrokeWidth     : 2,
//Number - Spacing between each of the X value sets
barValueSpacing     : 5,
//Number - Spacing between data sets within X values
barDatasetSpacing   : 1,
//String - A legend template
legendTemplate      : '<ul class="<%=name.toLowerCase()%>-legend"><% for
(var i=0; i<datasets.length; i++) { %><li><span style="background-
color:<%=datasets[i].fillColor%>"></span><%if(datasets[i].label){ %><%=datasets[i
].label%><% } %></li><% } %></ul>',
//Boolean - whether to make the chart responsive
responsive          : true,
maintainAspectRatio : true
}

barChartOptions.datasetFill = false
var myChart = barChart.HorizontalBar(barChartData, barChartOptions)
//document.getElementById('legend_'+rowid).innerHTML=
myChart.generateLegend();
});
</script>
<?php
}
?>
</body>
</html>

```

6.4.4 login.php:

```

<?php
    session_start();
    include 'includes/conn.php';

    if(isset($_POST['login'])){
        $username = $_POST['username'];
        $password = $_POST['password'];

        $sql = "SELECT * FROM admin WHERE username = '$username'";
        $query = $conn->query($sql);

        if($query->num_rows < 1){
            $_SESSION['error'] = 'Cannot find account with the username';
        }
        else{
            $row = $query->fetch_assoc();
            if(password_verify($password, $row['password'])){
                $_SESSION['admin'] = $row['id'];
            }
        }
    }
}

```



```

        }
        else{
            $_SESSION['error'] = 'Incorrect password';
        }
    }

    }
    else{
        $_SESSION['error'] = 'Input admin credentials first';
    }

    header('location: index.php');

?>

```

6.4.5 logout.php

```

<?php
    session_start();
    session_destroy();

    header('location: index.php');

?>

```

6.4.6 Voters.php

```

<?php include 'includes/session.php'; ?>
<?php include 'includes/header.php'; ?>
<body class="hold-transition skin-blue sidebar-mini">
<div class="wrapper">

    <?php include 'includes/navbar.php'; ?>
    <?php include 'includes/menubar.php'; ?>

    <!-- Content Wrapper. Contains page content -->
    <div class="content-wrapper">
        <!-- Content Header (Page header) -->
        <section class="content-header">
            <h1>
                Voters List
            </h1>
            <ol class="breadcrumb">
                <li><a href="#"><i class="fa fa-dashboard"></i> Home</a></li>
                <li class="active">Voters</li>
            </ol>
        </section>
        <!-- Main content -->
        <section class="content">
            <?php
                if(isset($_SESSION['error'])){

```

```

        echo "
        <div class='alert alert-danger alert-dismissible'>
            <button type='button' class='close' data-dismiss='alert' aria-
hidden='true'>&times;</button>
            <h4><i class='icon fa fa-warning'></i> Error!</h4>
            ".$_SESSION['error'].
        </div>
    ";
    unset($_SESSION['error']);
}
if(isset($_SESSION['success'])) {
    echo "
    <div class='alert alert-success alert-dismissible'>
        <button type='button' class='close' data-dismiss='alert' aria-
hidden='true'>&times;</button>
        <h4><i class='icon fa fa-check'></i> Success!</h4>
        ".$_SESSION['success'].
    </div>
    ";
    unset($_SESSION['success']);
}
?>
<div class="row">
    <div class="col-xs-12">
        <div class="box">
            <div class="box-header with-border">
                <a href="#addnew" data-toggle="modal" class="btn btn-primary btn-sm btn-
flat"><i class="fa fa-plus"></i> New</a>
            </div>
            <div class="box-body">
                <table id="example1" class="table table-bordered">
                    <thead>
                        <th>Lastname</th>
                        <th>Firstname</th>
                        <th>Photo</th>
                        <th>Voters ID</th>
                        <th>Tools</th>
                    </thead>
                    <tbody>
                        <?php
                            $sql = "SELECT * FROM voters";
                            $query = $conn->query($sql);
                            while($row = $query->fetch_assoc()){
                                $image = (!empty($row['photo'])) ? '../images/'.$row['photo'] :
'../images/profile.jpg';
                                echo "
                                    <tr>
                                        <td>".$row['lastname'].</td>
                                        <td>".$row['firstname'].</td>
                                        <td>

```

```

        
        <a href='#edit_photo' data-toggle='modal' class='pull-right photo'
data-id=" ".$row['id'].'"><span class='fa fa-edit'></span></a>
    </td>
    <td> ".$row['voters_id']. "</td>
    <td>
        <button class='btn btn-success btn-sm edit btn-flat' data-
id=" ".$row['id'].'"><i class='fa fa-edit'></i> Edit</button>
        <button class='btn btn-danger btn-sm delete btn-flat' data-
id=" ".$row['id'].'"><i class='fa fa-trash'></i> Delete</button>
    </td>
</tr>
";
    }
?>
</tbody>
</table>
</div>
</div>
</div>
</div>
</div>
</section>
</div>

<?php include 'includes/footer.php'; ?>
<?php include 'includes/voters_modal.php'; ?>
</div>
<?php include 'includes/scripts.php'; ?>
<script>
$(function(){
    $(document).on('click', '.edit', function(e){
        e.preventDefault();
        $('#edit').modal('show');
        var id = $(this).data('id');
        getRow(id);
    });

    $(document).on('click', '.delete', function(e){
        e.preventDefault();
        $('#delete').modal('show');
        var id = $(this).data('id');
        getRow(id);
    });

    $(document).on('click', '.photo', function(e){
        e.preventDefault();
        var id = $(this).data('id');
        getRow(id);
    });

```

```

});

function getRow(id){
$.ajax({
  type: 'POST',
  url: 'voters_row.php',
  data: {id:id},
  dataType: 'json',
  success: function(response){
    $('#id').val(response.id);
    $('#edit_firstname').val(response.firstname);
    $('#edit_lastname').val(response.lastname);
    $('#edit_password').val(response.password);
    $('#fullname').html(response.firstname+' '+response.lastname);
  }
});
}
</script>
</body>
</html>

```

6.4.7 votes.php

```

<?php include 'includes/session.php'; ?>
<?php include 'includes/header.php'; ?>
<body class="hold-transition skin-blue sidebar-mini">
<div class="wrapper">

<?php include 'includes/navbar.php'; ?>
<?php include 'includes/menubar.php'; ?>

<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
  <!-- Content Header (Page header) -->
  <section class="content-header">
    <h1>
      Votes
    </h1>
    <ol class="breadcrumb">
      <li><a href="#"><i class="fa fa-dashboard"></i> Home</a></li>
      <li class="active">Votes</li>
    </ol>
  </section>
  <!-- Main content -->
  <section class="content">
    <?php
      if(isset($_SESSION['error'])){
        echo "
          <div class='alert alert-danger alert-dismissible'>

```

```

        <button type='button' class='close' data-dismiss='alert' aria-
hidden='true'>&times;</button>
        <h4><i class='icon fa fa-warning'></i> Error!</h4>
        ".$SESSION['error'].
    </div>
    ";
    unset($_SESSION['error']);
}
if(isset($_SESSION['success'])) {
    echo "
        <div class='alert alert-success alert-dismissible'>
        <button type='button' class='close' data-dismiss='alert' aria-
hidden='true'>&times;</button>
        <h4><i class='icon fa fa-check'></i> Success!</h4>
        ".$SESSION['success'].
    </div>
    ";
    unset($_SESSION['success']);
}
?>
<div class="row">
    <div class="col-xs-12">
        <div class="box">
            <div class="box-header with-border">
                <a href="#reset" data-toggle="modal" class="btn btn-danger btn-sm btn-
flat"><i class="fa fa-refresh"></i> Reset</a>
            </div>
            <div class="box-body">
                <table id="example1" class="table table-bordered">
                    <thead>
                        <th class="hidden"></th>
                        <th>Position</th>
                        <th>Candidate</th>
                        <th>Voter</th>
                    </thead>
                    <tbody>
                        <?php
                            $sql = "SELECT *, candidates.firstname AS canfirst, candidates.lastname
AS canlast, voters.firstname AS votfirst, voters.lastname AS votlast FROM votes LEFT
JOIN positions ON positions.id=votes.position_id LEFT JOIN candidates ON
candidates.id=votes.candidate_id LEFT JOIN voters ON voters.id=votes.voters_id
ORDER BY positions.priority ASC";
                            $query = $conn->query($sql);
                            while($row = $query->fetch_assoc()){
                                echo "
                                    <tr>
                                        <td class='hidden'></td>
                                        <td>".$row['description'].</td>
                                        <td>".$row['canfirst']. ' ' . $row['canlast'].</td>
                                        <td>".$row['votfirst']. ' ' . $row['votlast'].</td>

```

```

        </tr>
        ";
    }
    ?>
</tbody>
</table>
</div>
</div>
</div>
</div>
</section>
</div>

<?php include 'includes/footer.php'; ?>
<?php include 'includes/votes_modal.php'; ?>
</div>
<?php include 'includes/scripts.php'; ?>
</body>
</html>

```

CHAPTER 7

TESTING

7.1 TESTING PLAN

Testing process starts with a test plan. This plan identifies all the testing related activities that must be performed and specifies the schedules, allocates the resources, and specified guidelines for testing. During the testing of the unit the specified test cases are executed and the actual result compared with expected output. The final output of the testing phase is the test report and the error report.

7.1.1 Test Data

Testing process begins with a test design. This arrangement recognizes all the testing related exercises that must be performed like the timetables, assigning the assets, and determining rules for testing. This testing of the unit of the predetermined experiments are executed and the genuine outcome is expected. The last part of the testing stage is the test report and the error report.

7.1.2 Unit testing

Every individual module has been tried against the necessity with some test information.

7.1.3 Test Report

The module is working appropriately given the client must enter data. All information section frames have tested with indicated test cases and all information passage shapes are working properly.

7.1.4 Error Report

On the off chance that the client does not enter information in determined request, at that point the client will be incited with error messages. Error reduction is done to deal with the normal and sudden mistakes.

CHAPTER 8

CONCLUSION

8.1 CONCLUSION

Internet-based voting offers many benefits including low cost and increased voter participation. Voting systems must consider security and human factors carefully, and in particular make sure that they provide voters with reliable and intuitive indications of the validity of the voting process. The system we propose uses visual cryptography to provide mutual authentication for voters and election servers.

8.2 Future Scope

The Future works for this system can further be added with additional features of security at the voters level by using “Block Chain Technology” to validate number of votes using Decentralization concept and “Image Recognition” using a webcam to find whether the voter is authentic or not. Furthermore on Online voting website attacks like “Sybil” can be altered with the help of authentication of unique voting details instead of login id’s, to find the valid user in the system.

CHAPTER 9

BIBLIOGRAPHY

References

- <http://103.2.233.230:8080/jspui/bitstream/123456789/305/1/06653684.pdf>
- <http://www.theijes.com/papers/v4-i3/Version-1/L0431071075.pdf>
- <https://www.irjet.net/archives/V3/i12/IRJET-V3I1230.pdf>
- <https://link.springer.com/chapter/10.1007/BFb0053419>
- <https://www.sciencedirect.com/science/article/pii/S0031320302002583>
- <https://ieeexplore.ieee.org/abstract/document/1658106/>
- <https://www.sciencedirect.com/science/article/pii/S0304397599001279>
- <https://otik.uk.zcu.cz/handle/11025/5993>
- <https://www.sciencedirect.com/science/article/pii/S0167865502002593>
- https://hrcak.srce.hr/index.php?id_clanak_jezik=264242&show=clanak
- <https://ieeexplore.ieee.org/abstract/document/6181923/>
- <https://ieeexplore.ieee.org/abstract/document/7860062/>