CHAPTER 1

INTRODUCTION

1.1 Background Study:

India is a constitutional democracy with a parliamentary system of government, and at the heart of the system is a commitment to hold regular, free and fair elections. An election helps to solve the problem of succession in leadership and thus contributes to the continuation of democracy. In terms of religion the India subcontinent is the birthplace of four of the world's major religions; namely Hinduism, Buddhism, Jainism, and Sikhism. According to the 2011 census, 79.8% of the population of India practices Hinduism, 14.2% adheres to Islam, 2.3% adheres to Christianity, and 1.7% adheres to Sikhism. In India elections are conducted by an independent and powerful autonomous election commission. It enjoys the same kind of independence that the judiciary enjoys. But once appointed the election commission is not answerable to the president of government. In India there are basically three types of elections held at a period of every five years where people of India vote to choose their leader namely, Lok Sabha or General Elections, State Legislative Elections and Local Bodies Elections . Here the PM, CM and Local Authorities like Mayors etc., are elected by the people. The Lok Sabha Elections forms the major general election consist of 542 members representing respective constituencies. In the State Assemblies Election a CM is elected and it took for each 29 states and 7 Union Territories at different times. That is to say, in any given elections, Indians vote for the PM, CMs and the Local bodies. Most of the Asian Electoral bodies not an exception right from their inception to date, even with latest advancements in technology, still use a primitive paper based methods or EVMs during voting ,this system is characterized by manual form filling to chose leaders or to be manually available to the polling booths and wait for their turn to cast their votes, this has led to an excessive number of malpractices and inconvenience for many people to vote as to be present at their constituencies at voting time making a very lower voting turnouts as expected in many areas. The main advantage of EVMs are that they are only available to government and provide ease of vote casting and counting and thought to be most secured still many questions are always been arises on the security of

EVMs and Ballot-paper's security but till now no one had been proved true. The disadvantages outweigh the advantages for instance the need to print ballot papers is a slow, expensive, inflexible, environmentally, hostile process, visual impairment, or literacy limitations and also last minute changes to the voter register are difficult to accommodate among others and also for the EVMs that it required a secured physical security every time along with transportations to different operations place and required each and every one to be present at the voting poll location of their constituencies in order to cast their poll.

Over the last few years, there have been a number of election observers who have suggested electoral organizations should introduce online voting at State and Local Government election processes. A general observation is that as more business is done using electronic mediums, it should not be difficult to carry out voting using electronic equipment like mobiles and PCs along with turning up at the polling place on voting day to use EVMs. The Online Voting System (OVS) under implementation mainly addresses the voting phase. Electronic voting using the OVS should be cheaper than the present EVMs & Ballot-Paper mode of Indian Electoral Commission (IIC). The phenomenal use of the Internet as a vehicle for improving communication, access to information and electronic commerce has led to the claim that the Internet could be used as either a replacement to attendance voting or as an additional voting option. Throughout history, election fraud has occurred in many electoral processes from which experience shows that the manual voting process is a major source of such vices and violence in many democratic countries, a case in point is the Indian Electoral Commission (IEC) that has on several occasions failed to update the India national voter's name in the voting list results in not allowing them to cast their poll.

The mechanism leading to fraud is manifested in registration places by corrupt officials on local commissions who are in a position to issue voter registration and also results in the people who be not to able to cast their votes due to non-ability of them at their constituencies location on the day of voting or due to long voting lines at voting booth.
The first online voting technique was firstly adopted by America to conduct their local bodies elections and in India the first test was conducted in 2010 with online voting process was successful due to which it then use in local bodies elections in Gujarat for 8 – districts but the result was not satisfactory as instead of increasing voting turnout it reduced

it to only upto 8.7% of total voter's available due to lack of reliability on system for security reasons, manual registration process and unawareness to the using of internet connected smart devices.

In 2005, Estonia became the first country in the world to hold nation-wide elections using this method, and in 2007, it made headlines as the first country to use i-Voting in parliamentary elections. Estonia becomes the first ever country to make their all kind of voting process online. In the case of i-Voting, the cumulative time saved in the last Estonian elections was 11,000 working days.

1.2 Introduction to i-Voting System

The i-Voting system also known as Online Voting System (OVS) is a term encompassing a technique allowing voters to poll their ballot online using their variously available electronics gadgets like mobile phones, PCs and Laptops etc The advantage of i-Voting over the common "queue method" is that the voters have the choice of voting at their own free time and there is reduced congestion. It also minimizes on errors of vote counting and thus provides instant results. The individual votes are submitted in a database which can be queried to find out who of the aspirants for a given post has the highest number of votes. This database of voter's data and their ballot is secured by using a 158-bit data encryption technique using 3-DES.This system is geared towards increasing the voting percentage in India since it has been noted that with the old voting method {the Queue System}, the voter turnout has been a wanting case. With system in place also, if high security is applied, cases of false votes shall be reduced.

Internet voting systems are appealing for several reasons which include; People are getting more used to work with computers to do all sorts of things, namely sensitive operations such as shopping and home banking and they allow people to vote far from where they usually live, helping to reduce absenteeism rate as the most secured and sensing process of banking can be done by an online mode then it could also be used for voting purpose.

"Design and implementation of Triple DES for secure i-Voting system" is an online voting technique. In this system people who have citizenship of India and whose age is

above 18 years of age can give his/her vote online without going to any physical polling station.

There is a database which is maintained in which all the names of voters with complete information is stored in an Encrypted mode with 158-bit 3DES Encryption.

In this system a voter can use his/her voting right online without any difficulty. he/she has to be registered first for him/her to vote. Registration is mainly done by the voter by using the adhaar no. for security reasons. The adhaar no. gets verified the user details and his identity .Citizens seeking registration are expected to provide their Voters ID in order to improve security. After the validity of them being citizens of India has been confirmed by the system by comparing their details submitted with those in adhaar and voter ID, the citizen is then registered as a voter.

After registration, the voter is assigned a secret Voter ID with which he/she can use to log into the system and enjoy services provided by the system such as voting.

1.3 Problem Statement

The previously use of online voting technique was first used in Gujarat in 2010 and 2015 making it become the first Indian state to allow and use online voting technique in local bodies elections. But the idea was failed both the time very badly as it was required the user to first register for the voting purpose and then need to verify their documents from the nearby EC-Office to complete their registration manually. Due to which from available 1 lakhs voters only 20,000 apply to register online in six municipal corporation and from them only approx 1,310 were make it to complete the verification procedure by verifying their documents at EC-Office to be able to eligible to cast their vote and from those voters only 806 were able to cast their votes online.

This big fails of internet voting is explained in the EC reports as in the following points.

- The voters need to more aware to use the smart devices connected to internet in order to vote online.
- The security issue was a major concern in the mind of voters about their data and

their ballot.

- The manual process of voting registration consist of manual data verification caused the voters to not complete their registrations and thus provide only about of 10% available voters to complete their registrations.
- The online voting is a good thing but need to be popularized among the youth specially.
- The EC also creates some one voting booth but it needs to create more of those or provide the booth voting techniques along with the online voting phase.

1.4 Significance of Study

The main purpose of i-Voting System includes:

- Provision of improved voting services to the voters through fast, timely and convenient voting.
- Reduction of the costs incurred by the Kenyan Electoral Commission during voting time in paying the very many clerks employed for the sake of the success of the manual system.
- Check to ensure that the members who are registered are the only ones to vote.
- Cases of "Dead People" voting are also minimized.
- I-Voting will require being very precise or cost cutting to produce an effective election management system.
- Therefore crucial points that this i-Voting emphasizes on are listed below:

i. Require less number of staff during the election.

ii. The use of 158-bit encryption of data and ballot of voter make by using Triple DES (3DES) makes it more secured and reliable for the voters to cast their ballot online.

iii. The manual process of voter's document verification of registration is changes by 1-minute Adhaar & Voter ID verification by just a One-Time-Password on registered mobile with adhaar.

iv. The ease to cast vote online due to the awareness among user for internet users through smart devices for many sensitive transactions like shopping, and banking make it more reliable for them to use this system.

v. This system is a lot easier to independently moderate the elections and subsequently reinforce its transparency and fairness.

vi. Less capital, less effort, and less labor intensive, as the primary cost and effort will focus primarily on creating, managing, and running a secure online portal.

vii. Increased number of voters as individual will find it easier and more convenient to vote, especially those abroad.

1.5 Objectives of the project

The specific objectives of the project include:

- Reviewing the existing voting process or approach used in India;
- Coming up with an automated online voting system in India with easy registration and voting process.
- Designing an enhanced security to make online voting system more secured.
- Implementing an online voting system with more secured Triple DES data Encryption technique.

1.6 Project justification

The online i-Voting system shall reduce the time spend making long queues at the polling stations during voting. It allows voters to register with their adhaar card and voter id to prove their authenticity by using the OTP (One Time Password) .It shall also enable the voters to vote from any part of the globe as explained since this is an online application available on the internet. Cases of vote miscounts shall also be solved since at the backend of this system resides a well developed database using Oracle/MySQL that can provide the correct data once it's correctly queried in an Encrypted form using 158-bits Triple DES Encryption technique.

1.7 Scope of Study

It is focused on studying the existing system of voting in India and to make sure that the peoples vote is counts, for fairness in the elective positions in and online Voting

process. This is also will produce:

- Less effort and less labor intensive, as the primary cost and focus primary on creating, managing, and running a secure web voting portal with proper data security.
- Increasing number of voters as individuals will find it easier and more convenient to vote, especially those abroad.

1.8 LIMITATION OF STUDY

Time factor was the greatest barrier to the successful completion of this exercise since it had to be done within the semester. We also had financial constraints since all the activities involved were self-sponsored.

CHAPTER 2

LITERATURE REVIEW

Review of literature is a very important part of a research project. The theory of i-Voting has gained rapidly recognition, and is often associated with statements such as glimpses into our future. Electronic Voting Machines are being used in Indian general and State Elections to implement electronic voting in part from 1999 elections. EVMs have replaced paper ballots in local, state and general elections in India. There were earlier claims regarding EVM's tamparability and security which have not been proved.

In order to overcome the security problem and to provide ease of access to voters as compare to existing system this system implement a secured way to remotely cast their vote easily by using their mobile or computer. All computer scientists who have done work in or are interested in electronic voting seem to agree that online voting does not meet the requirements for public elections and that the current widely-deployed voting systems need improvement. Voting on the Internet has disadvantages based on the areas of secrecy and protection against coercion and/or vote selling. It's such a truly bad idea that there seems to be no credible academic effort to deploy it at all.

The first use of Internet voting for a binding political election took place in the US in 2000,with more countries subsequently beginning to conduct trials of and/or use Internet voting. A total of 14 countries have now used remote Internet voting for binding political elections or referenda. Within the group of Internet voting system users, four core countries have been using Internet voting over the course of several elections/referenda: Canada, Estonia, France and Switzerland. Estonia is the only country to offer Internet voting to the entire electorate. The remaining ten countries have either just adopted it, are currently piloting Internet voting, have piloted it and not pursued its further use, or have discontinued its use. Examples of Internet voting in other countries around the world vary widely in scope and functionality. The early cases of Internet voting were less technically advanced than those being developed more recently. Many of the changes seen in Internet

voting systems have been aimed at improving the quality of elections delivered by these systems and meeting emerging standards for electronic voting.

It is fair to say that Internet voting is not a commonly used means of voting. Of the 14 countries that have so far used it in any form, only ten currently have expressed any intention of using it in the future. However, Internet voting is a relatively new voting technology and 9 have been developing significantly over the previous ten years. Internet voting seems to fit, for many countries, a niche corner of the electoral process. It is largely targeted at those who cannot attend their polling station in person on Election Day. In fact many more countries have expressed or shown an interest in the use of Internet voting, especially when they have large numbers of expatriate voters. However, the implementation of Internet voting, according to emerging standards, is a very technical exercise. It can also pose some difficult political questions if the aim is to facilitate the inclusion of large numbers of expatriate citizens in the political process.

In 2005, Estonia became the first country in the world to hold nation-wide elections using this method, and in 2007, it made headlines as the first country to use i-Voting in parliamentary elections. Estonia becomes the first ever country to make their all kind of voting process online. The idea of having electronic voting in Estonia gained popularity in 2001 with the "e-minded" coalition government. Estonia became the first nation to hold legally binding general elections over the Internet with their pilot project for the municipal elections in 2005. The electronic voting system withstood the test of reality and was declared a success by Estonian election officials. The Estonian parliamentary election in 2007 also used internet voting, another world first. In the case of i-Voting, the cumulative time saved in the last Estonian elections was 11,000 working days. In today's time Estonia is the only country to have use the online voting system as their full time voting system for approx every type of elections held there in Estonia.

In India the Indian municipal election of Gujarat for 8 states of 20015 brought national attention to the problems and first use of internet voting technique. Most people believe that the current system should need some improvements; there is much disagreement on how such improvements should be made. The IEC recommend after the failure of the online voting system in Gujarat that the system is still an effective one and have numerous

of advantages in the future .The cause of the failure is the unawareness among people about using smart devices and internet, it require to provide more ease to the process of online voting along with a high profile security to improve the reliability in the mind of people going to use it. Other researchers have done work in electronic voting; while they may not explicitly mention voting from remote poll sites, their work is nonetheless relevant to any effort at designing or implementing a remote poll site voting system. Lorrie Cranor acknowledges the problems inherent in each kind of voting apparatus, but doesn't make an overt recommendation on her site for one technology over the rest. Some other academicians like Peter Neumann focus on the immensity of the problem one faces when trying to design and implement a truly secure voting system. They often remind us of Ken Thompson's Turing acceptance speech and the fact that we really can't trust any code which we did not create ourselves. Therefore, they tend to be extremely suspicious of proprietary voting machines and their makers who insist that we should “just trust [them].”

Neumann gives a list of suggestions for "generic voting criteria" which suggests that a voti ng system should be so hard to tamper with and so resistant to failure that no commercial system is likely to ever meet the requirements, and developing a suitable custom system would be extremely difficult and prohibitively expensive. A voting machine must produce human-readable hardcopy paper results, which can be verified by the voter before the vote is cast, and manually recounted later if necessary. David Chaum presents a very interesting scheme, whereby voters could get receipts for their votes. This receipt would allow them to know if their votes were included in the final tally or not, and to prove that they voted without revealing any information about how they voted. The security of this scheme depends on visual cryptography developed by Naor and Shamir, and on voters randomly choosing one of two pieces of paper. Mercuri and Neumann advocate the use of this technique in electronic voting systems. In the recent years, voting equipments which were widely adopted in many countries may be divided into five types.

i. **Paper-based voting:** The voter gets a blank ballot and uses a pen or a marker to indicate he want to vote for which candidate. Hand-counted ballots is a time and labor consuming process, but it is easy to manufacture paper ballots and the ballot scan be retained for verifying, this type is still the most common way to vote.

ii. **Lever voting machine:** Lever machine is peculiar equipment, and each lever is assigned for a corresponding candidate. The voter pulls the lever to poll for his favorite candidate. This kind of voting machine can count up the ballots automatically. Because its interface is not user-friendly enough, giving some training to voters is necessary.

iii. **Direct recording electronic voting machine:** This type, which is abbreviated to DRE, integrates with keyboard; touch screen, or buttons for the voter press to poll. Some of them lay in voting records and counting the votes is very quickly. But the other DRE without keep voting records are doubted about its accuracy.

iv. **Punch card:** The voter uses metallic hole-punch to punch a hole on the blank ballot. It can count votes automatically, but if the voter's perforation is incomplete, the result is probably determined wrongfully.

v. **Optical voting machine:** After each voter fills a circle correspond to their favorite candidate on the blank ballot, this machine selects the darkest mark on each ballot for the vote then computes the total result. This kind of machine counts up ballots rapidly. However, if the voter fills over the circle, it will lead to the error result of optical-scan. Recent years, a considerable number of countries has adopted E-voting for their official elections. These countries include; America, Belgium, Japan and Brazil.

2.1 THE SECURITY ISSUES OF ONLINE VOTING

Foreign experience revealed that they are often confronted by security issues while the online voting system is running. The origin of the security issues was due to not only outsider (such as voters and attackers) but also insider (such as system developers and administrators), even just because the inheritance of some objects in the source code are unsuitable. These errors caused the voting system to crash. The proposed solutions were correspondingly outlined to hold back these attacks. For example, to avoid hacker making incursion into the voting system via network, we can design our system to transmit data in an encrypted form over

the network. Another example is to limit voter to input particular data, so that we can prevent the command injection from running. The other problem is to provide the voter a fast ,efficient and secured authenticity procedure for registration which is done by using the voter's adhaar number and it's data along with the voter id and the verification is done by using a One –Time –Password (OTP) on voter's registered mobile number to allow only authenticate one to to register and to create a password for further login process. The login process required the voter to login using the adhaar number as a username and the set password during the registration process along with a captcha verification to prevent any pre-defined script to be run automatically. During the voting process one need to verify OTP again send to the registered mobile number after whose verification the user's vote has been cast to their desired candidate and provide a verification.

2.2 Research Papers

- Online Elections in Romania by Vlad Costea
- Internet Voting for Expatriates: The Swiss Case by Uwe Serduit
- Online Voting in Estonia by Pricha Lechsa
- Online Voting System by Kamlakar Singh

CHAPTER 3

AIMS AND OBJECTIVES

3.1 AIM

Our aim is to promote truth and freedom by empowering individuals to communicate effectively and implement non-coercive solutions to societal problems.

As a first step towards fulfilling our aim, is taking on the challenge of improving the integrity standards of voting systems used in elections worldwide.

In order to ensure that election results are truly accurate, we are developing an online open source voting platform that provides transparency into election results by allowing voters to independently poll their ballot and enhancing the security of current i-Voting system using a Triple DES symmetric encryption. Using i-voting system we can actually accomplish this, all while protecting each voter's right to privacy. It not only improve the efficiency of voting by increasing the voting turnout but also make it most secured and reliable for the voter's by using a 158-bit data encryption and a two- factor security verification for login and one-factor verification using OTP during casting of a ballot online.

After all, in an election, it's not who votes that counts, it's who counts the votes!

3.2 OBJECTIVE

a. To study and identify secured i-Voting technique using Triple DES algorithm.
b. Design and implementation of secured i-Voting technique.
c. Implementation of Triple DES for security enhancement of i-Voting system.
d. Enhancing security of i-Voting system to provide faster paperless Registrations to increase voting turnouts.

CHAPTER 4

METHODOLOGY

In this chapter, the source of data methods of collection, the evaluation of the existing system and the organization structure of the system problem are presented. It includes specific methods which were used in order to achieve the objectives of the project, particular requirements for implementation of the project and a brief explanation of why such methods were used for implementing the proposed system, also included is a brief description of the current system of voting.

4.1 Proposed Methodology

The proposed methodology is totally based on online voting pattern in which there is an ease to poll the vote online using web-portal on computers or mobile in a more secured way as compared to the previous techniques.

The Secured i-voting which uses Triple DES Encryption uses two factor verification for user authentication and one factor verification using OTP (One Time Password) for polling after login by using user adhaar details .This methodology also include a symmetric encryption method to encrypt the data i.e., Triple Data Encryption Standard and thus provide more secured way of voting as compared to the existing method.

The proposed method is defined a way to provide user and electoral admin to easily vote remotely and get the results calculation easily and securely to be published.

4.2 Existing Methodology

The Existing method of voting in India is based firstly on ballet papers polls then EVMs and then some countries try to use i-Voting system but all the method adopted have very big loopholes in the security and also required a great amount of workers and thus money.

The EVMs fraud has been recently reported in which there is alteration with the mechanism and mapping of polls with the respected parties, which still not have been proved yet also the ballet papers also easily manipulated by the parties members and other people. Also, the form of data of i-Voting system can be easily manipulated by hackers which is loosely encrypted and required a very long process of registration to valid the pollers authenticity.

4.1.1 Problems with the Existing Voting Systems

The problems of the existing manual system of voting include among others the following:

1. **Expensive and Time consuming:** The process of collecting data and entering this data into the database takes too much time and is expensive to conduct, for example, time and money is spent in printing data capture forms, in preparing registration stations together with human resources, and there after advertising the days set for registration process including sensitizing voters on the need for registration, as well as time spent on entering this data to the database manually.

2. **Too much paper work:** The process involves too much paper work and paper storage which is difficult as papers become bulky with the population size.

3. **Errors during verifications by manual process:** Errors are human's behavior and thus it can also be caused during manual registration process at the time of document verification in previously used system.

4. **Time consuming and caused to much discomfort:** This is a very big problem since not all people have free time during the given short period of time to check and update the voter register by reaching out at EC office for document verification.

5. **Security:** Security of voter's data is the biggest point of concern as the system connected to a network is always threat to malpractices like hacking and all other illegal stuff and thus required proper security to the data of user.

Hence there is great desire to reduce official procedure in the current voter registration process along with the improvement of security of system, if the general electoral process is to improve the voting turnouts by making system more reliable to the voters.

CHAPTER 5

Triple DES (3DES) Encryption

The main and most important part of this system is to enhance the previous system security by using multiple security tactics, the triple DES is one of them and is the most important one which provide the security to the system by encrypted user data using a 158-bits encryption when it is saved or being transferred on the network to make it possibly impossible for any hacking practice to be successful.

3-DES stands for Triple Data Encryption Standards. DES is a symmetric key encryption algorithm. Same key is being used for encryption and decryption. So challenge in using symmetric key algorithm is that we need to have the same key for decryption which is used for encryption. People follow different approach to save key. Either they append key with cryptic text or physically save it somewhere. Although it's officially known as the Triple Data Encryption Algorithm (3DEA), it is most commonly referred to as 3DES. This is because the 3DES algorithm uses the Data Encryption Standard (DES) cipher three times to encrypt its data.

DES is a symmetric-key algorithm based on a Feistel network. As a symmetric key cipher, it uses the same key for both the encryption and decryption processes. The Feistel network makes both of these processes almost exactly the same which results in an algorithm that is more efficient to implement.

DES has both a 64-bit block and key size, but in practice, the key only grants 56-bits of security. 3DES was developed as a more secure alternative because of DES's small key length. In 3DES, the DES algorithm is run through three times with three keys, however it is only considered secure if three separate keys are used.

5.1 Uses of Triple DES

Once the weaknesses of normal DES became more apparent, 3DES was adopted in a wide range of applications. It was one of the more commonly used encryption schemes before the rise of AES.
Some examples of its implementations included Microsoft Office, Firefox and EMV payment systems. Many of these platforms no longer use 3DES because there are better alternatives.

The National Institute of Standards and Technology (NIST) has released a draft proposal saying that all forms of 3DES will be deprecated through 2023 and disallowed from 2024 onward. Although it's just a draft, and the above statement have not proved practically and still it have supported by most of the systems as compared to the new AES.

5.2 History of Triple DES

Since 3DES is derived from DES, it's best to introduce the earlier standard first. In the seventies, the National Bureau of Standards (NBS – it has since been renamed NIST) was searching for an algorithm that it could use as a standard for encrypting sensitive yet unclassified government information.

The NBS accepted proposals for a standard that would fit its requirements, but none of the candidates from the original round were appropriate. It invited more submissions, and this time IBM sent through an algorithm that its team developed. The submission was derived from the Lucifer cipher that Horst Feistel designed.

In 1975, the IBM algorithm was published by the NBS as the proposed Data Encryption Standard. The public was invited to comment on the design, which attracted some criticism.
Prominent cryptographers such as Whitfield Diffie and Martin Hellman, designers of the Diffie-Hellman key exchange, claimed that the key length was too short and that the S-boxes had been changed from their initial design.

At the time, many in the cryptographic community thought that the NSA had sabotaged the project and weakened the algorithm, so that it would be the only agency that could break DES.
When this was investigated by the United States Senate Select Committee on Intelligent, it was found that the "NSA convinced IBM that a reduced key size was sufficient; indirectly assisted in the development of the S-box structures; and certified that the final DES algorithm was, to the best of their knowledge, free from any statistical or mathematical weakness."

The same report went on to say that the "NSA did not tamper with the design in any way." This has been backed up by some former IBM staff who claimed that the DES algorithm was designed entirely by the IBM team.

The NSA's own declassified documentation claims that the agency "worked closely with IBM to strengthen the algorithm against all except brute-force attacks and to strengthen substitution tables…"
Suspicions of NSA tampering were eased in the nineties once differential cryptanalysis was publicly discovered. When the much-maligned S-boxes were tested with the new technique, they were found to be more resistant to attack than if they had been chosen randomly.

This indicates that the IBM team had already known about the differential cryptanalysis in the seventies, with Steven Levy claiming that the NSA asked them to keep the technique secret in order to protect national security.

Famed cryptographer Bruce Schneier once quipped, "It took the academic community two decades to figure out that the NSA 'tweaks' actually improved the security of DES."
Despite the initial questions about the algorithm's security and the NSA's involvement, the IBM algorithm went on to be approved as the Data Encryption Standard in 1976. It was published in 1977 and reaffirmed as the standard in 1983, 1988 and 1993.

When linear cryptanalysis was first published in 1994, it started to raise questions about the security of the algorithm. In 1997, NIST announced that is was looking for an

algorithm to replace DES. The need for a new algorithm was intensified as technology developed further and potential attacks grew stronger.

Various cracking attempts showed that it was less difficult to break the algorithm than previously thought. In 1998, distributed.net was able to crack DES is 39 days.
By the start of 1999, the Electronic Frontier Foundation's Deep Crack had gotten the time down to a little over 22 hours. This signaled the end of DES, since an attack of this nature was now within the reach of a well-resourced adversary.

The main issue was the small key space, and a new algorithm was sorely needed. This was a problem, because it would take several more years for NIST to settle on the algorithm which became the replacement standard, the Advanced Encryption Standard (AES).

While the cipher for AES was being decided upon, 3DES was proposed on as a stopgap measure. It involves running the DES algorithm three times, with three separate keys. In 1999, DES was reaffirmed, but with 3DES as the ideal algorithm. Normal DES was only permitted in legacy applications.

5.3 Understanding the 3-DES

Before we can talk about the details of 3DES, it's important to understand the DES algorithm that it's derived from. So let's start right at the beginning. We use encryption to turn our plaintext data into cipher text, which is information that cannot be accessed by attackers (as long as we are using appropriate algorithms). Encryption algorithms are essentially complex mathematical formulas. When it comes to encrypting something like "Let's go to the beach", many people get confused. After all, how can you apply math to things like letters and characters?

5.3.1 Encoding the text

The reality is that computers don't deal in letters and characters. Instead, they work on a system of 1s and 0s known as binary. Each 1 or 0 is known as a bit, and a collection of eight of them are known as a byte.

You can either look it up manually, or use an online converter to see that in binary, “Let’s go to the beach” becomes:

01001100 01100101 01110100 00100111 01110011 00100000 01100111 01101111 00100000 01110100 01101111 00100000 01110100 01101000 01100101 00100000 01100010 01100101 01100001 01100011 01101000

5.3.1.1 Blocks

When data is encrypted, it’s divided into separate blocks for processing. DES has a 64-bit block size, which essentially means that each block fits a mix of 64 ones and zeros. Our first block (the first 64 digits of the binary shown above) would be:

01001100 01100101 01110100 00100111 01110011 00100000 01100111 01101111

Our second would be:

00100000 01110100 01101111 00100000 01110100 01101000 01100101 00100000

And our final block would be:

01100010 01100101 01100001 01100011 01101000

5.3.1.2 Padding

You may have noticed that our third block is only 40 bits long. Before it can be encrypted, it needs to be build up to a 64-bit block size. This is done with padding, which involves adding extra information to a block in order to complete it. This can be done with a number of different schemes, and it can also serve to make encrypted information harder to crack, but we won’t get into that in this article.

.3.1.3. The DES key schedule

Encryption algorithms use keys to add in data that will alter the end result of the process. If DES only involved steps like permutation and S-boxes (permutation is explained below, while S-boxes are covered in the Substitution section), all that an attacker would have to do is uncover the details of the algorithm, then do each of the steps in reverse to reveal the initial message.

Since most of our algorithms are widely known, this wouldn’t really add much security. Instead, secret keys are added to alter the output in a way that cannot be predicted just by

knowing the algorithm (as long as a sufficiently complex algorithm is used).

DES begins with a single key, which is used to make sub keys that are applied in each round. This is a 64-bit key, which is the same size as our blocks. Let's say our key is:
01001010 10101101 11101000 10100101 01110001 01010100 10101001 11111010
Now, this key in in binary, which is the way that data is expressed when computers process it. When humans deal with keys, they will normally appear as a mix of characters, something like this: **kj329nf982bc9wn1**
In DES, the first step of deriving our round keys is to permute the key (move it around) according to the following table:

C						
57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
D						
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	13	28	20	12	4

PC1

Fig 5.1 Permutation Boxes

In permutation, each bit of our original key is shuffled to a new position as indicated by the table. Since the cell in the top left corner (of C) says 57, the first number of our permuted key will be the number in the 57th position of our old block:
01001010 10101101 11101000 10100101 01110001 01010100 10101001 11111010
The second cell says 49, which means that the second digit of our new key will be the number that is in the 49th position of our old block:
01001010 10101101 11101000 10100101 01110001 01010100 10101001 1111010
The third cell says 41, so we look for the digit at the 41st position:
01001010 10101101 11101000 10100101 01110001 01010100 10101001 1111010
So far, our key is made up of "110".
The rest of the key is arranged in the same way, according to the values of the table. We

move left to right, and once we get to the end of a row, we jump down to the next one, just like normal. Once table C is finished, we jump to table D to complete the second half of the key.

There's no easy way to transpose our entire block according to the initial permutation table. You could do the whole thing manually, or write a script for it (or even get lucky and find one in the depths of the internet), but we are going to cheat and make it up:
1100010 1010010 1010101 0101010 1010000 1111001 0001011 1000111

You may be worried that we are making up some of the numbers in this guide, but the reality is that it doesn't really matter. No one encrypts data manually any more, it's all done via programs. The most critical aspect of this tutorial is that you get a clear idea of the concepts that we are dealing with. The numbers themselves just serve to help you visualize what is going on.

Round number	Number of left shifts
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2

13	2
14	2
15	2
16	1

Fig 5.2 The round table

Some readers may have noticed that the table (and now our key), only has 56 bits rather than 64. This is because every eighth bit is skipped. This is an artifact from the older days of technology, when it was important to have parity check bits, which verified whether the key had been received correctly. These parity check bits mean that in practice, DES only has the security of a 56-bit key.

The tables C and D give us a key that has two 28-bit halves. Sometimes, the halves are referred to as C and D, but throughout this article we will refer to them as L and R, for left and right. Our left side is:

1100010 1010010 1010101 0101010

While our right is:

1010000 1111001 0001011 1000111

The next step is to shift the key by either one or two spaces to the left, depending on the round. The exact number of spaces is decided on according to the following predetermined table:

Search:

Showing 1 to 16 of 16 entries

So let’s take our left and right halves:

L 1010010 1010010 1010101 0101010

R 1010000 1111001 0001011 1000111

And shift both of them one position to the left, since the first round has a shift

of 1 according to the table (the number on the left end gets moved to the right end).

First round sub key:

L 0100101 0100101 0101010 1010101

R 0100001 1110010 0010111 0001111

In the second round, the table also says 1, so this result will again be altered by moving each number one position to the left.

Second round sub key:

L 1001010 1001010 1010101 0101010

R 1000011 1100100 0101110 0011110

In the third round, the numbers will be moved two places to the left, because the table now says 2.

Third round sub key:

L 0101010 0101010 1010101 0101010

R 0001111 0010001 0111000 1111010

In the subsequent rounds, the numbers are moved to the left according to the distances specified in the table, with each shift being applied to the result of the previous round. In the end, this gives us sixteen different sub keys, one for each round of the DES process.

The next step is another permutation according to the PC2 table shown below:

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

PC2

Fig 5.3 The PC2

By now, you should be familiar with permutations, so we won't go into the process in-depth. If you want to see how it works in more detail, refer to the explanation near the start of this section. Although the relocation positions are different, the process is the same.

Each of the 16 keys derived in the shifting process are now shuffled according to the table, with the number from the 14th position moved to the first place, the 17th to the second, the 11th to the third, etc..If you look closely at the table, you will notice that there are only 48 bits, rather than the 56 bits that we previously had. This process is known as compression permutation.

You can also see that the top half of the table features numbers between one and 28, while the bottom half contains numbers from 29 to 56. This keeps the left and right halves of our sub keys separate, and it is indicated below by the larger space in the middle of the keys.

Again, we're going to cheat and make up the numbers. Let's say that this entire process gave us the following sub keys:

Round one: 010101 010101 101010 110100 101001 100101 101010 101010
Round two: 011010 110101 101110 110010 010100 110010 111101 101101
Round three: 010100 100110 110110 101010 100110 011000 101011 011001
Round four: 011001 110101 011001 110101 000011 001011 010101 010101
Round five: 110101 001101 010101 010101 010011 001011 010111 100101
Round six: 010111 110101 011001 111001 101001 100101 101010 101010
Round seven: 110101 111010 101110 101010 100110 010110 111011 001110
Round eight: 011001 110101 010101 001001 010011 001011 010100 101010
Round nine: 111011 011010 011110 100010 100010 010110 110011 110010
Round 10: 011010 010101 101110 101001 010010 010110 111000 101010
Round 11: 110101 001101 101110 101010 100101 100101 101010 001010
Round 12: 101001 100100 101001 101010 100110 011000 101011 011001
Round 13: 010010 010010 010101 010101 010110 110001 100101 101010
Round 14: 101001 100110 010101 011101 010001 001010 110010 111110
Round 15: 011001 011010 011001 110101 001001 011001 100101 101101
Round 16: 010010 100110 010101 010101 010001 101000 110010 111010

This shifting process results in each bit from the initial key being used in about 14 of the 16 sub keys, although some bits are used slightly more than others.

Initial permutation

Once the data has been divided into blocks and padded if necessary, it's time to begin the DES encryption process. We will get back to the sub keys that we just created at a later stage. The first step is known as the initial permutation, where the data is rearranged according to the following table:

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

E

Fig 5.4 The Initial Permutation

This initial permutation process doesn't make the algorithm any more secure. This is because it doesn't involve the input of any key, and can easily be reversed. The algorithm was originally designed this way because it made implementation easier in certain contexts. Since we have covered permutations a couple of times, we'll skip any major explanation here. Head back to The DES key schedule section if you need more information on how they work.

Let's take the first block from the message "Let's go to the beach", which we derived in the Block section under Understanding the DES algorithm:

01001100 01100101 01110100 00100111 01110011 00100000 01100111 01101111

Since the first cell says 58, we would select the number from the 58th position:

01001100 01100101 01110100 00100111 01110011 00100000 01100111 01101111

Then we would take the number from the 50th position:

01001100 01100101 01110100 00100111 01110011 00100000 01100111 01101111

And the number from the 42nd position:

01001100 01100101 01110100 00100111 01110011 00100000 01100111 01101111

This gives us “110” so far. We will make up the rest of the number:

11010111 01001010 10101000 10011101 01001011 10110101 10000111 10101001

When the initial permutation is complete, the data is moved on to the next step.

Splitting the blocks

Once the data has undergone its initial permutation, it is split into two halves. We take our block that just underwent its initial permutation:

11010111 01001010 10101000 10011101 01001011 10110101 10000111 10101001

And we will separate it into two blocks, a left block (made up of the first 32 digits), and known as L0:

L0 11010111 01001010 10101000 10011101

And a right block (made up of the second 32 digits), known as R0:

R0 01001011 10110101 10000111 10101001

The F function

Now that the block has been split, it’s time for the F function to take place. In the first round, it will only be applied to the right half of the block, while the left half is kept aside until later. The right side undergoes the following four steps as part of the F function:

- Expansion permutation (E in the diagram)
- Key mixing ($\oplus$ in the diagram)
- Substitution (each S1, S2 etc. in the diagram)
- Permutation (P in the diagram)

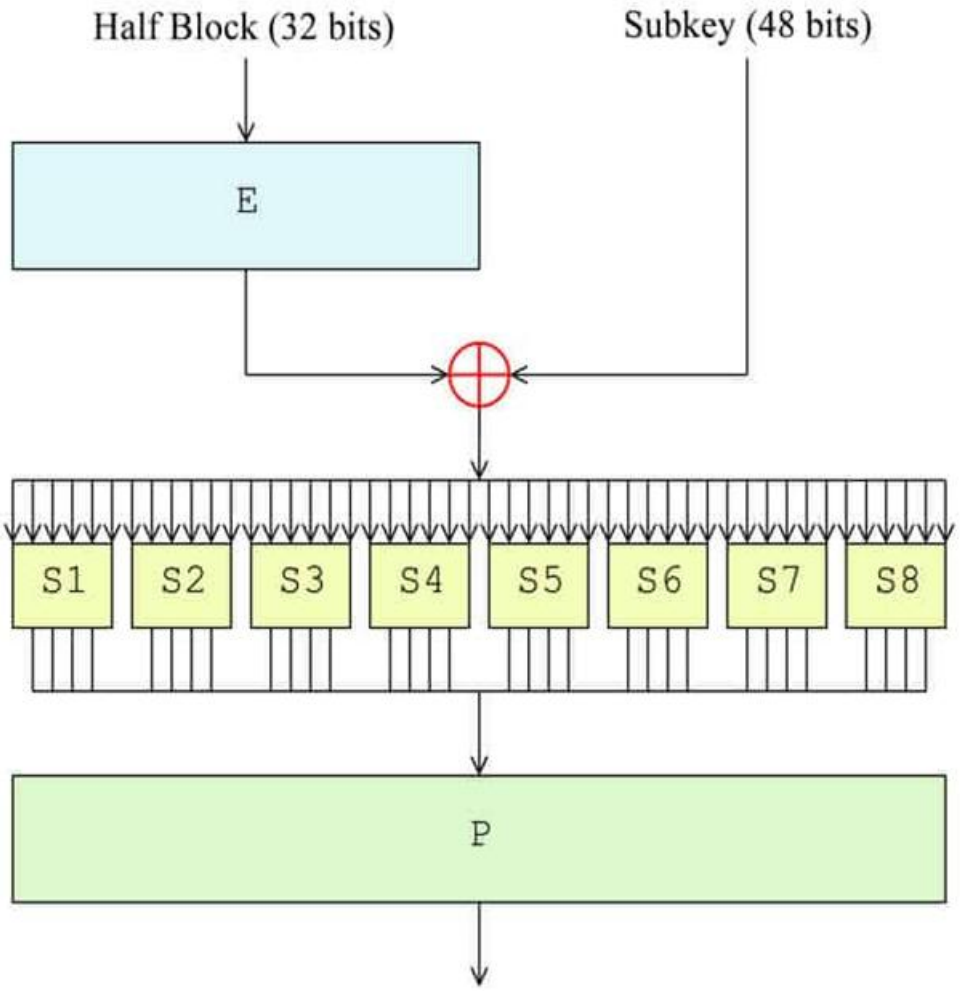


Fig 5.5 The Complete Permutation Process

Expansion permutation

The expansion permutation accomplishes three things. The most important is that it allows single bits of input data to affect the output of two other bits, causing an avalanche effect. It also makes the right half 48-bits, so that it is the same size as the sub key for the next step. The other effect of the expansion permutation is that it makes the output longer than the input. This allows it to be compressed in the substitution operation.

The bits are rearranged according to the following table. Some of the individual bits are in the table twice, which is how the block expanded from 32 to 48 bits:

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

E

Fig 5.6 The Expansion Box

Since the first cell says 32, we take our right block and select the number from the 32nd position, just like we did in the other examples of permutation listed above:

R0 01001011 10110101 10000111 10101001

We then take the numbers from the first position, the second position and so on, right up until we get to the bottom right corner of the block. Since there is a 1 in this cell, the last digit will also be the number that appears in the first position of our block.

Let's say that the expansion permutation gives us a new 48-bit block of:

101110 100110 100100 000000 001100 001110 101101 011110

Key mixing

Once the block has been expanded to 48 bits, it's time to apply the first round's sub key, which we derived in the DES key schedule section above. The block is modified by the sub key using the XOR cipher.

The XOR cipher is an addition cipher that follows a simple process, especially when compared to the other elements we have already discussed.

In an XOR cipher:

0 + 0 = 0

1 + 0 = 1

1 + 1 = 0

So let's say you have to XOR the following two numbers in binary:

1101

0101

Each digit would be added to the one below it. According to the three rules shown above, this gives a result of:

1000

To complete the key mixing step, we take the right side of our block that we just expanded to 48 bits, and the first round key. We then perform the XOR addition:

Block: 101110 100110 100100 000000 001100 001110 101101 011110

Round one key: 010101 010101 101010 110100 101001 100101 101010 101010

XOR result: 111011 110011 001110 110100 100101 101011 000111 110100

The result of the XOR operation is then passed on to the next round.

Substitution

Substitution adds confusion to data. It's normally done with lookup tables, which are also known as substitution boxes or S-boxes. DES uses eight separate tables or S-boxes, a different one for each 6 bits of data. The following table shows the eight S-boxes of DES:

שורה	מס' עמודה															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	S_1															
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	3	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	13	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
	S_2															
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
	S_3															
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
	S_4															
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
	S_5															
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
	S_6															
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
	S_7															
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
	S_8															
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Fig 5.7 The s-boxes

The eight separate S-boxes are used to translate each 6-bit input into a 4-bit output. The first step in the process is to take the digits at the beginning and end of a 6-bit segment, then convert that binary value to decimal.

Let's take the data that we just finished XORing in the previous step:

111011 110011 001110 110100 100101 101011 000111 110100

We will look at the first 6-bit segment to show you how the substitution process works:

111011

Since the first number and last number are both 1, this gives us a value of 11. We then convert 11 from binary to decimal, which gives us 3. These are just equivalent values, written in different ways. Think of it as converting computer language to human language. You can check out the conversion for yourself with an online calculator if you want.

We then take the four middle digits of the first 6-bit segment:

111011

And convert them from binary to decimal. 1101 translates to number 13.

Now, we take these two numbers and look them up in the S1 table:

שורה	מס' עמודה															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	S_1															
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	3	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	13	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Fig 5.8 The Substitution Box S1

Our first number, 3, tells us to look in the third row, while our second number, 13 tells us to look in the 13th column. The value in the third row of the 13th column is 0.

Now that we have looked up our number in the table, we convert it back to four digit binary. Zero is normally written as 0 in binary, but 0000 is the same, and this is the format that is most suitable for our purposes.

Following this process, the S-box converts our first 6-bit section of data (111011) into a different 4-bit value (0000). It seems convoluted, but this technique helps to further obscure the relationship between the cipher text and the plaintext that it is linked to.

The next 6-bit section of data then goes through the same process, but instead it uses the S2 box shown above. The third section uses the S3 table and so on, up until the final section undergoes the substitution through the S8 table.

Again, we're going to cheat for the rest of the values. Let's say that the substitution boxes give us a result of:

0000 1010 1100 1001 0100 1001 0111 0001

Once each section of the data has gone through its S-box, it moves onto the next step.

Permutation

The last stage of the F function is another permutation, using the following table:

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

P

Fig 5.9

The last Permutation Table

By now, you should have a decent understanding of how permutations shift digits from the old block to a different position in the new block, so we won't go into it again.

Let's say that this permutation takes our previous result:

0000 1010 1100 1001 0100 1001 0111 0001

and gives us an output of:

0101 0110 1001 0101 0010 0100 0101 0010

Now that the permutation has been completed, we have finished with the four steps of the F function in this round. In mathematical notation, this value is known as f(R0, K1). This means that the result is the function (f) of the initial right side of the block (R0) and the first round's sub key (K1).

XOR with the left block

Remember how we split the block in half just before we began the steps of the F function? We set aside the left side of the block (L0), while the right side underwent each of these processes. Well, now it's time for L0 to come back into action.

We take the right side that we have just processed f(R0, K1) and add it to the old left side (L0) using the XOR cipher. This gives us R1, the result of our first round:

f(R0, K1): 0101 0110 1001 0101 0010 0100 0101 0010

L0: 1101 0111 0100 1010 1010 1000 1001 1101

XOR result (R1): 1000 0001 1101 1111 1000 1100 1100 1111

Refer to the Key mixing section above if you need a reminder of how the XOR cipher works.

15 more rounds…

If you've gotten this far, then DES probably seems like an arduous process. But it's not even close to being finished yet. The data goes though the four steps of the F function, followed by the XOR, another 15 times, for a total of 16 rounds.

In the second round, we take the original, untouched version of the right side of the block (R0) and make it the new left side (L1). Meanwhile, we take the result of our first round and send it through the F function. Everything happens the same as last time, however this time the subkey for round two is used instead. Let's say that this process gives us a result of:

f(R1, K2): 1011 0111 1000 1011 1001 1101 1001 1110

We then XOR the result with L1, which is actually R0 (we derived this in the Splitting blocks section). This gives us the result of the second round, R2:

f(R1, K2): 1011 0111 1000 1011 1001 1101 1001 1110

L1: 0100 1011 1011 0101 1000 0111 1010 1001

R2: 1111 1100 0011 1110 0001 1010 0011 0111

This step can seem a bit confusing, but under the Feistel scheme, the old right side

becomes the new left, while the result of the operation becomes the new right side. The following diagram gives you a visual representation of what is happening. IP represents the initial permutation, F is a stand-in for the entire F function, the ⊕ symbolizes the XOR function and the arrows indicate each side of the block moving between left and right:

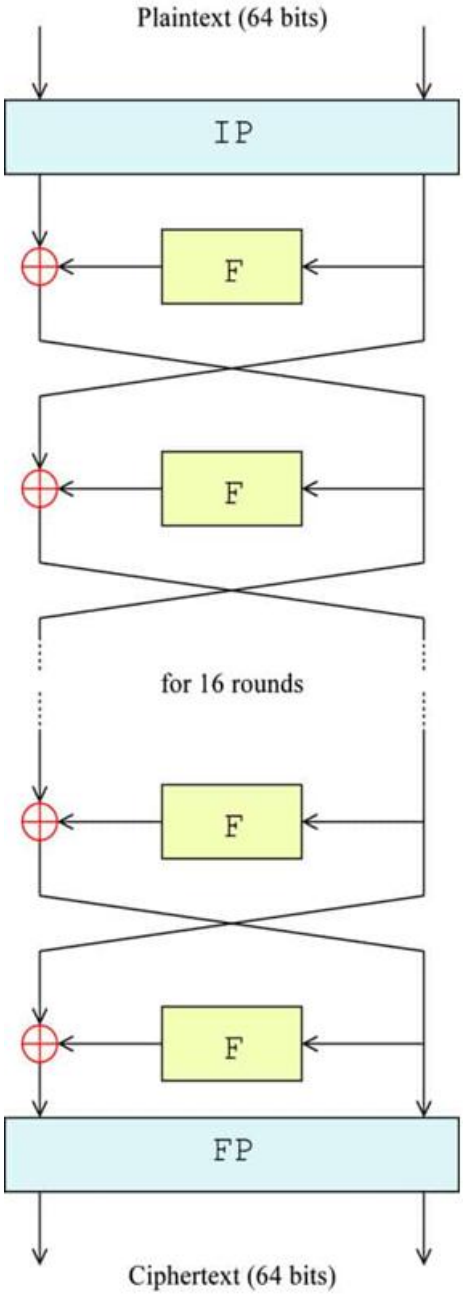


Fig 5.10 Fiestel Cipher Structure

The exact formula for each step is:

Ln = Rn-1

Rn = Ln-1 + f(Rn-1,Kn)

Where:

L = The left half of the block (starting with L0 when the block was initially split)

R = The right half of the block (starting with R0 when the block was initially split)

n = The round number (beginning with 0, when the block was initially split)

f = The F function

Kn = The subkey for round n

According to the formula and the diagram, in the third round, R1 becomes the new left half (L2), while R2 is processed through the F function. Let's say that it gives us a result of:

f(R2, K3) 1001 0111 0000 1011 1101 0111 1011 1011

We then calculate the result of our third round (R3), using the XOR cipher, just like before:

f(R2, K3): 1011 0111 1000 1011 1001 1101 1001 1110

L2: 0100 1011 1011 0101 1000 0111 1010 1001

R3: 1111 1100 0011 1110 0001 1010 0011 0111

The same process continues up until the fifteenth round, with the blocks switching over and the next subkey being used in each round. In the 16th and final round, the blocks are not switched over. Instead, they are combined to form a 64-bit block. Refraining from swapping the blocks in this last stage allows the algorithm to be used for both encryption and decryption.

Let's say that the final round gives us a result of:

1010 0101 0100 1011 1001 0001 0100 1000 0101 1010 1101 0001 1101 1001 1001 1101

Final permutation

This permutation is the inverse of the initial permutation, and again, it adds no extra security value. It rearranges the data according to the following table:

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

IP^{-1}

Fig 5.11 Inverse Permutation Table

This permutation table works the same as the previous ones. Since it's the final step of the encryption process, the result will be the cipher text for the first block of "Let's go to the beach". Let's say that the encrypted block is:
0100 1001 0011 0010 1001 0101 0111 0100 1011 1010 0111 0101 0111 1010 0101 0101

Now, if you wanted the real cipher text for "Let's go to the beach", you could have just skipped the whole learning process and gone straight to an online DES encryption tool. If we enter our sentence alongside a key (let's say kj329nf982bc9wn1) the tool gives us an encrypted text of:
U2FsdGVkX19Pienyu3w3q4zCd2IPKEPUWBzu3AeyVu2H3FeimZe6hA

If you want to, you can then convert the key and the cipher text to binary and then compare how the first block's cipher text lines up with the entire process that has been outlined.

5.4 DES decryption

In DES, the decryption process is incredibly straightforward. The algorithm's Feistel structure allows it to easily be reversed. The process is run almost exactly the same to decrypt information. The only difference is that the sub keys are applied in reverse. This is an efficient setup, because it means that the same software and hardware can be used in both the encryption and decryption processes.

To decrypt the data, it first goes through an initial permutation, then the block is split and the right half goes through the F function. The difference is that in the first round of decryption, the 16th sub key is applied. Everything else proceeds as normal. Once the F function is complete, it is XORed with the left side of the block.

The blocks are switched over and the result goes through the same process for the second round, with the only exception that the 15th sub key is applied. This process continues up until the 16th round, when the 1st sub key is used.

Just like in the encryption process, the blocks aren't swapped in the final stage, and then the data undergoes a final permutation. This finishes the decryption process, resulting in the original plaintext of the message.

5.5 3DES

As the security weaknesses of DES became more apparent, 3DES was proposed as a way of extending its key size without having to build an entirely new algorithm. Rather than using a single key as in DES, 3DES runs the DES algorithm three times, with three 56-bit keys:

- Key one is used to encrypt the plaintext.
- Key two is used to decrypt the text that had been encrypted by key one.
- Key three is used to encrypt the text that was decrypted by key three.

In each stage, the complete DES process is followed as outlined above.

It uses a separate key. If the first key was also used to decrypt the data in the second step, then the data would be right back where it started.

However, since it uses a different key, the decryption process doesn't actually serve to decrypt the data. It may seem logically perverse, but decrypting with a separate key only serves to jumble up the data even further.

Once the second key has “decrypted” the data, the third key is applied to encrypt it again. The result is the 3DES cipher text.

3DES is structured this way because it allows implementations to be compatible with single key DES, two key DES and three key DES (these are covered in the following section). This would not work if encryption was used in all three steps.

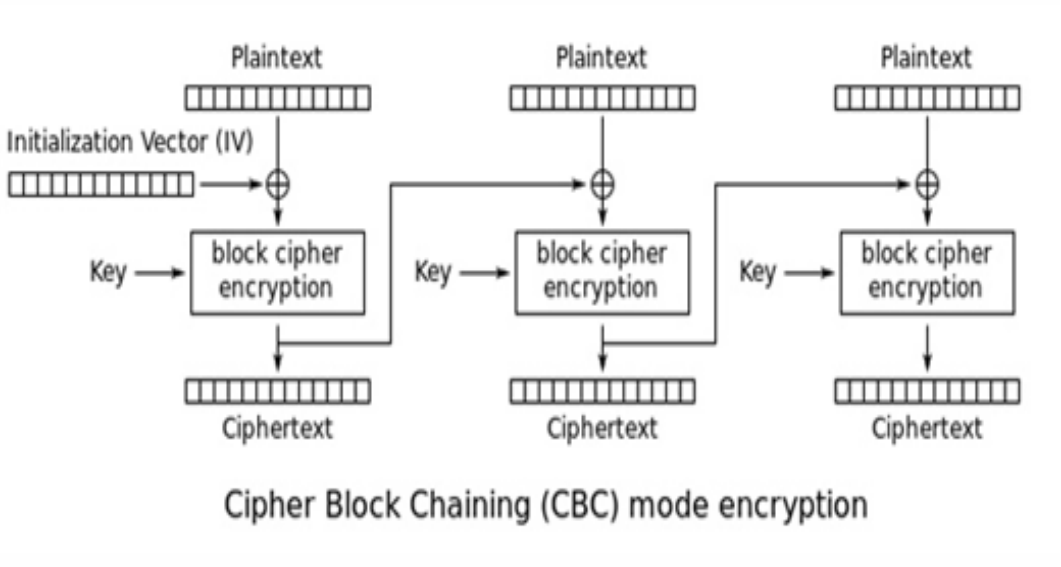


Fig 5.12 Cipher Block Chaining mode in Triple DES

5.5.1 3DES keying options

Technically, 3DES can be implemented with three different key configurations. Despite this, the second and third option is insecure and should never be implemented.

1. Keying option one – This option uses three independent keys and is the most secure.
2. Keying option two – In this configuration, the first and third keys are the same.
3. Keying options three – This uses three identical keys. When identical keys are used, the decryption process in the second stage cancels out the first encryption, leaving only the final encryption to alter the data. This makes the result the same as ordinary DES.

5.6 The security of 3DES

The security of 3DES depends on which keying option is being used. Keying option one involves three different 56-bit keys, which gives it a total key length of 168 bits. The effective length is reduced considerably by meet-in-the-middle attacks, which bring its real-world security down to 112 bits.

Meet-in-the-middle attacks are useful against encryption schemes that repeat the same algorithm several times. The technique stores the immediate values from each encryption stage, and then uses this information to radically improve the time that it would take to brute force the algorithm.
Options two and three have significantly smaller keys and are vulnerable to known-plaintext, and chosen-plaintext attacks, as well as others.

Known-plaintext attacks are possible when an adversary has access to both the plaintext and cipher text of a message. If an algorithm is susceptible to these attacks, the attacker can use this information to deduce the key, which allows them to crack all of the other data that has been encrypted by the same key.

A chosen-plaintext attack is similar, but it involves the attacker uncovering the key by comparing cipher texts to arbitrary plaintexts. Because of these vulnerabilities and the overall small key-sizes involved, keying options two and three are insecure and should not be implemented.

CHAPTER 6

DIAGRAMS

6.1 Project Flow Diagram

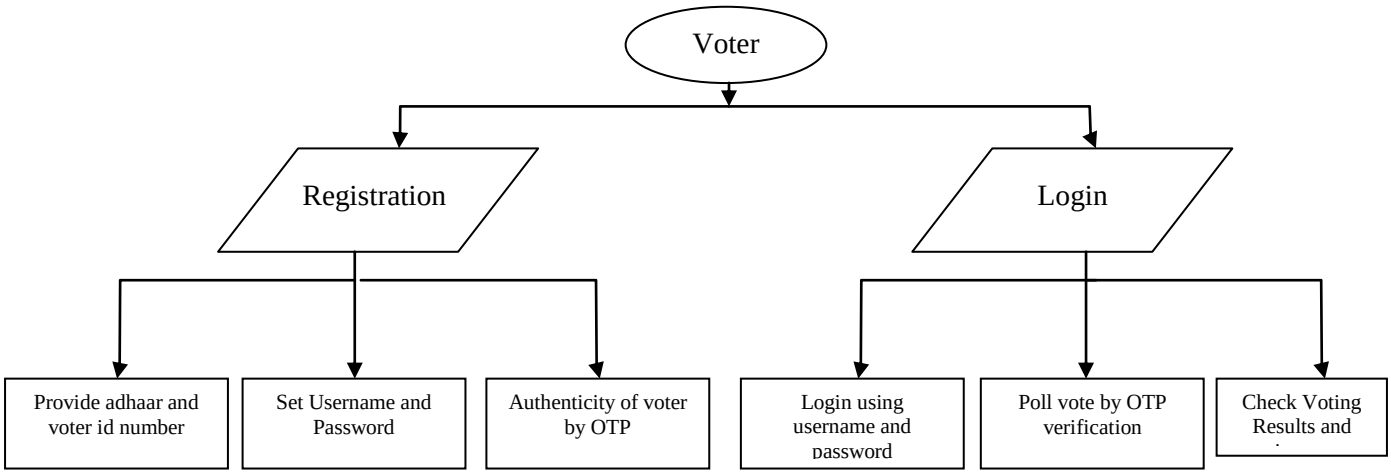


Fig 6.1 : The Voter Flowchart

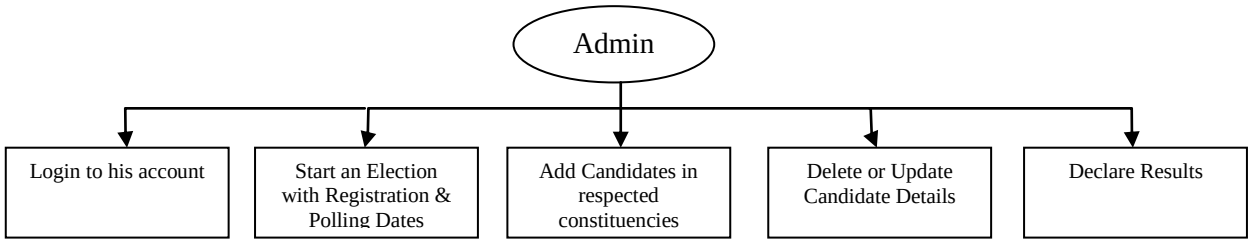


Fig 6.2 : The Admin Flowchar

6.2 E-R Diagram

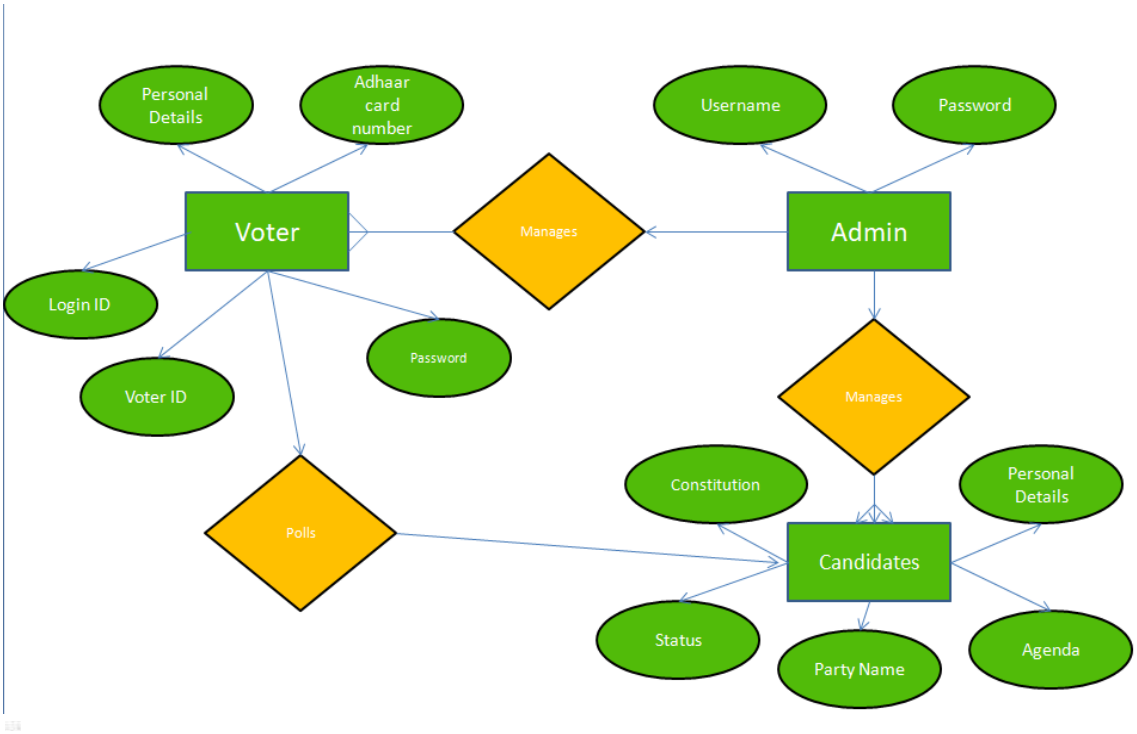


Fig 6.3: ER Diagram of i-Voting System

6.3 UML Diagram

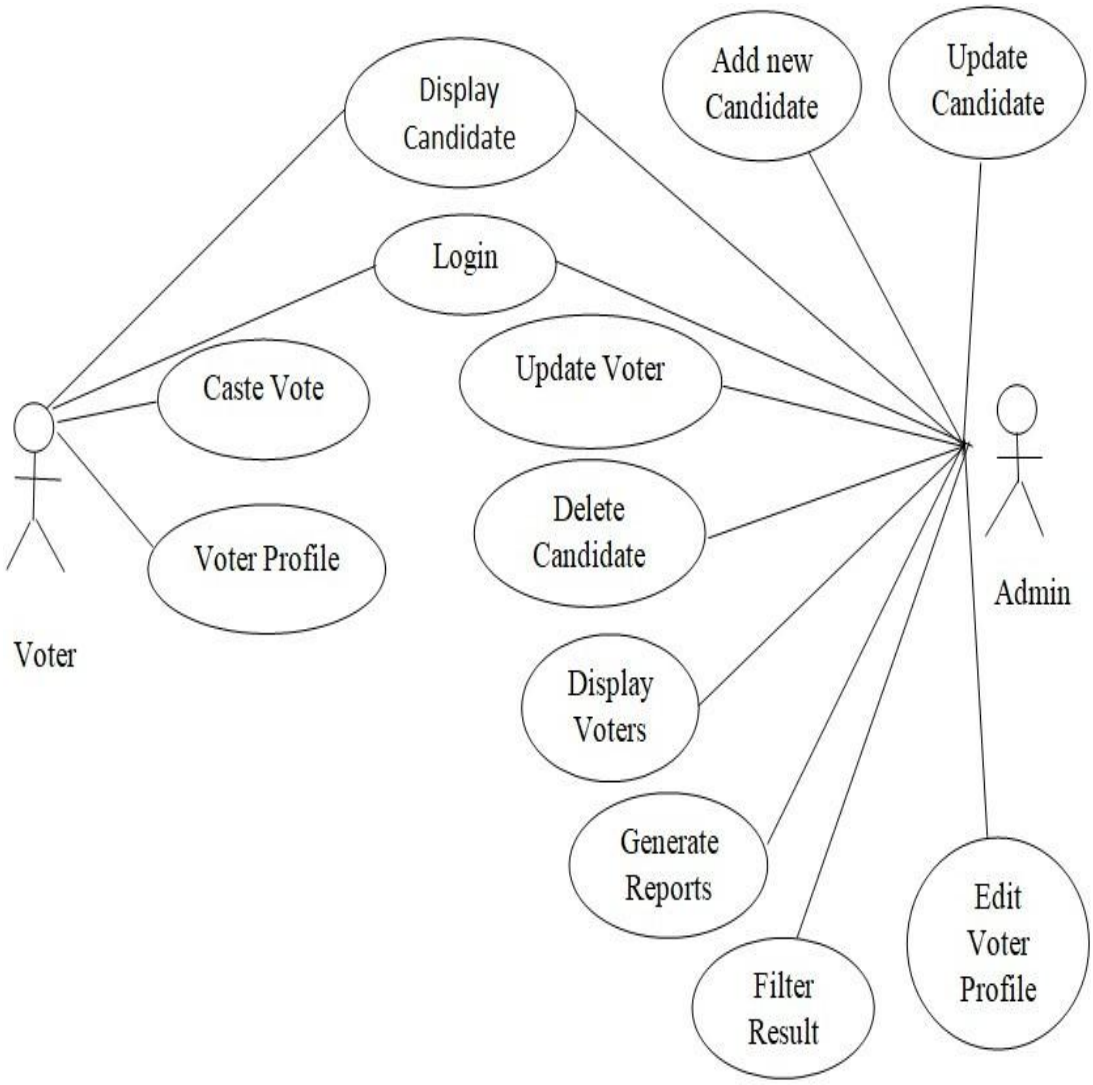


Fig 6.4: UML of i-Voting System

6.4. Data Flow Diagrams

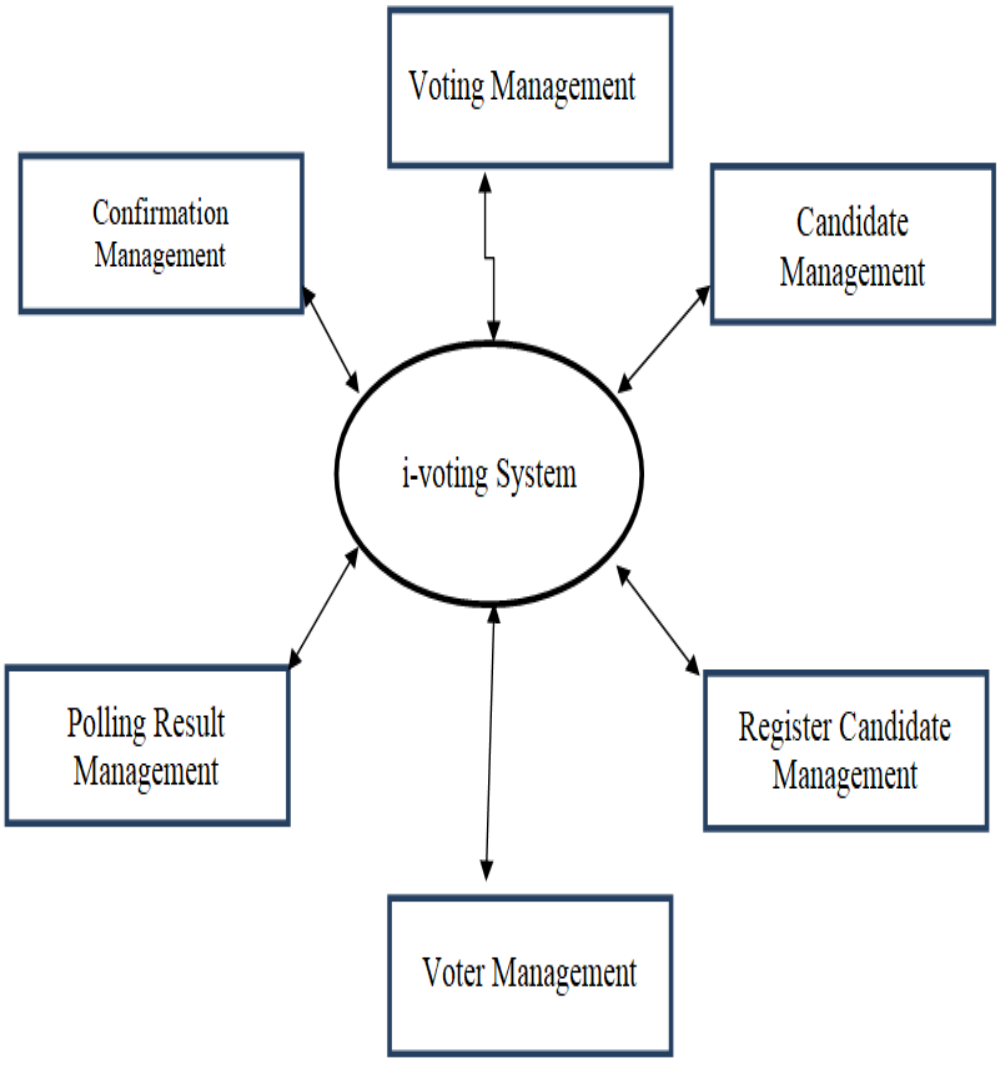


Fig 6.5: 0 level DFD of i-Voting System

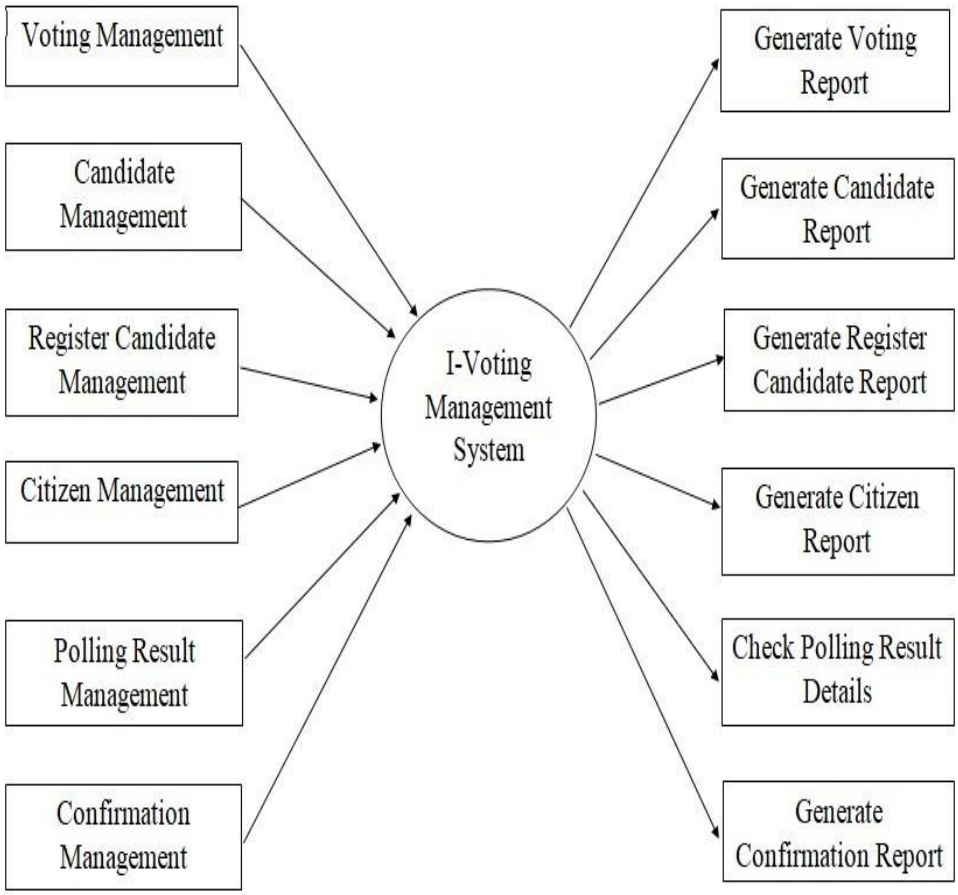


Fig 6.6: 1 Level DFD of i-Voting System

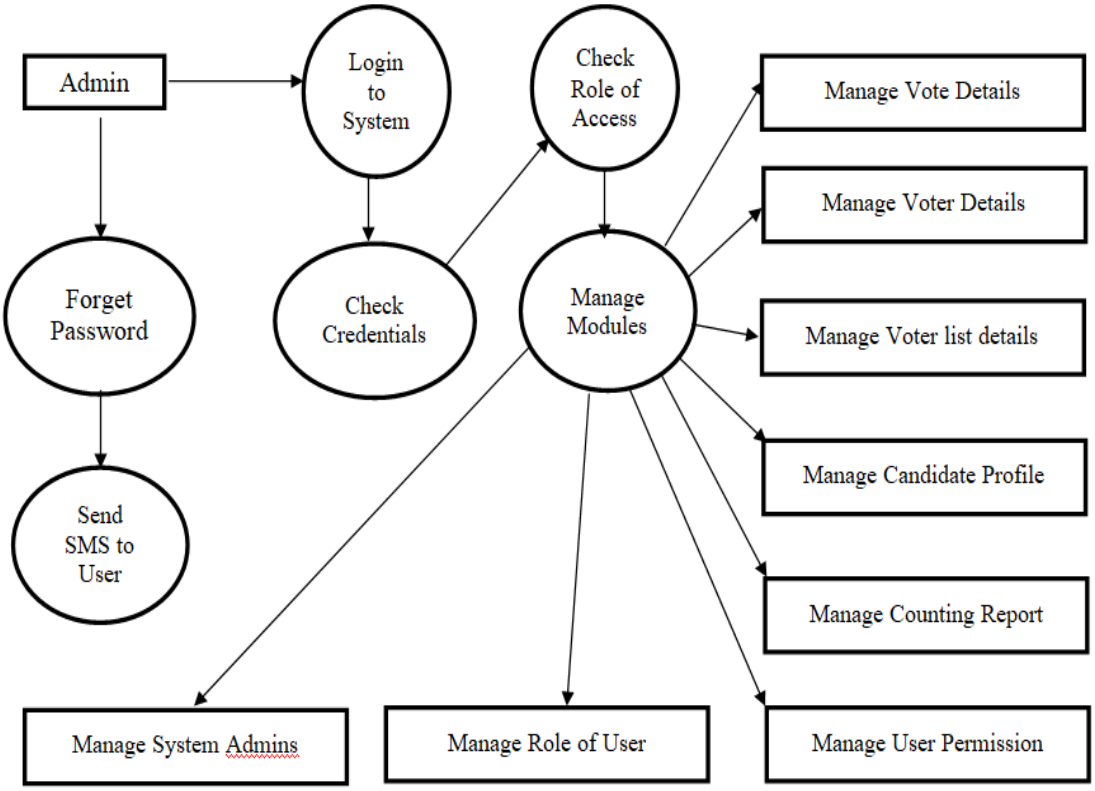


Fig 6.7: 2 level DFD of i-Voting System (Admin)

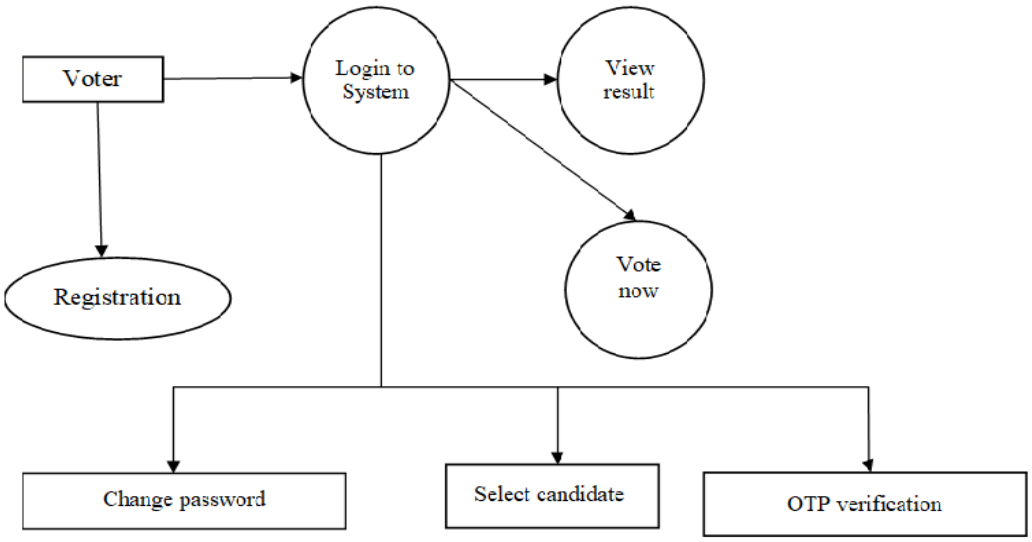


Fig 6.8: 2 level DFD of i-Voting System (Voter)

CHAPTER 7

SYSTEM SPECIFICATIONS AND DESIGN

7.1 User Requirements for the Proposed System

The OVS should:

- Be able to display all registered voters in the database to the SYSTEM ADMIN(s) as per their access rights and privileges.
- Have a user-friendly interface and user guides understandable by people of average computer skills.
- Be robust enough so that users do not corrupt it in the event of voting.
- Be able to handle multiple users at the same time and with the same efficiency, this will cater for the large and ever growing population of voters.
- Be able to provide faster registration as well as polling process.
- Be able to provide full security to the user's data.

7.2 REQUIREMENT SPECIFICATION

A system should meet the following requirements for it to run the i-Voting system:

- Web browsers like Mozilla Firefox, Google chrome, Opera and Internet Explorer,
- Web Servers Connectivity like MYSQLDBMS, WampServer, Macromedia Dreamweaver 8
- Programming language such as JAVA, JSP, Servlets, HTML, CSS, JavaScript.
- Operating Systems: Windows Operating Systems
- At least 2.0 GHz Processor speed,
- At least 40 GB Hard Disk Capacity and 1 GB RAM

7.3 Functional Requirements

- Secure storage and retrieval of voters‟ details from the database.
- Enable secure login of voters, that is to say non- legitimate voters should never be allowed to Login to the tool, these include the under aged and non nationals.

- Maintaining and manipulating records in database through functions like edit, delete, and view.
- Validate and verify input and output data

CHAPTER 8

PROJECT MODULES

This i-Voting project is consist of total two main modules, first is the voter module where user can registered themselves for online voting process and login to do so with a valid username i.e. Adhaar number and password. The user can poll their ballot after login or can check for the voting results and voting status.

The second module is for the admin of this system. This includes many options for the admin which it can access after getting login to the system. It includes starting an election, adding candidate, deleting or updating the details of the candidates, check voting status, declare results etc.

Here are the details explanations of each screen in a module with brief description about each of them:-

8.1 The Voter Module

As stated earlier the voting module is fully concerned with the voter accessible page where user first reach whenever he/she logon to the given DNS. It consist of many main and extra options which can be accessible by any one logged in or not. The main features like voting option, change password , check results, check voting status can only be accessible to the one who have a valid login credentials. The extra options include party's name, Election Period, Voting Rights, Forgot Password etc.

8.1.1 Main Page of Voter's Module

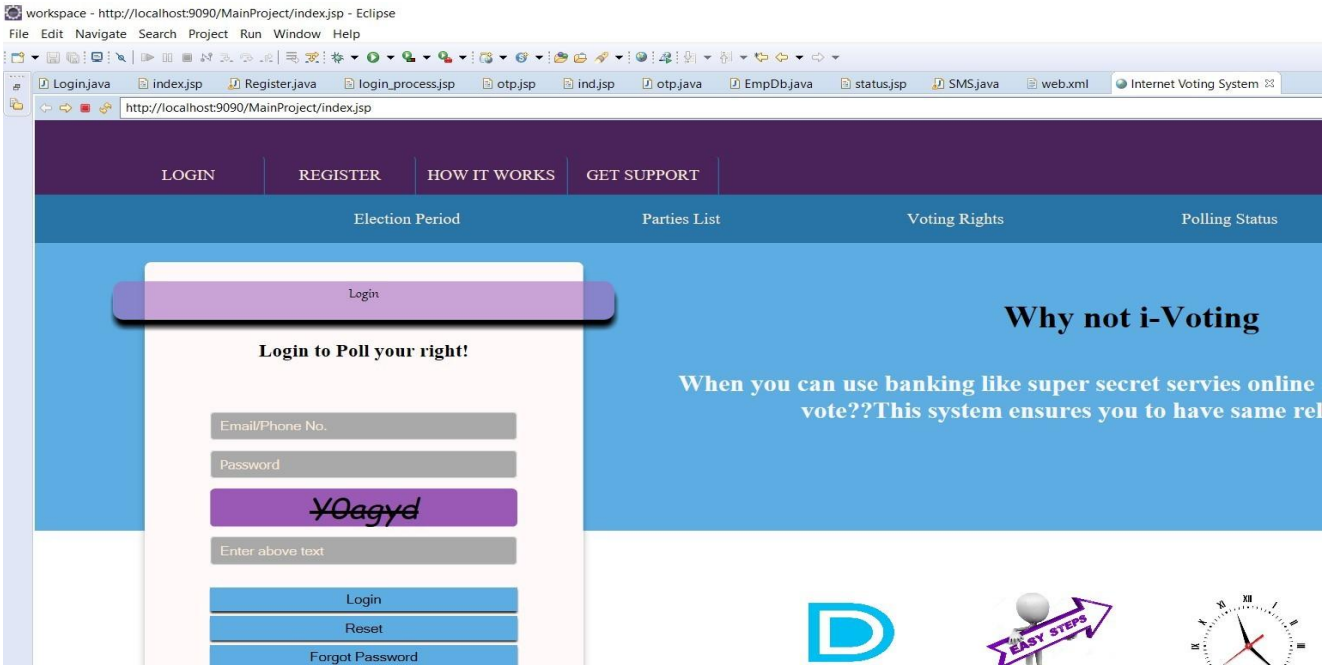


Fig 8.1 The main Page with Login Form

This page is the main page where the voter first visits. Here he have the option to Register themselves to poll there in the specified registration period and have login option to login to explore further option.

The Login process here required the user to have valid username which is the adhaar number and a password which he/she set during registration period and to enter the captcha as shown in captcha figure to authenticate and authorize access to the system.

There are other option which includes:

8.1.1.1 **How It Works:** This option as the name suggest will provide an step by step guide to the voter to let them to understand the voting process so as to make them easy for them to use the system without any error.

8.1.1.2 **Get Support: This** option provide the voter to get any type of support by posting their queries directly to the system maintenance staff.

8.1.1.3 **Election Period:** This option provides the user to get known about the date and

periods of registration and voting along with the details that which election is going on.

8.1.1.4 **Parties List:** The option is concerned with the details of different parties participating in the elections along with their brief description and their manifesto.

8.1.1.5 **Voting Rights:** This option provides the details about the rights of voter.

8.1.1.6 **Polling Status:** This option provide the polling status currently going on with three percentage of voter vote in each constituency.

8.1.2 The Registration Option

The registration process is the main process where an voter need to register with the adhaar number and voter id in order to get his/her identity along with its constituency of voting. The authenticity of user's data is done by one time password sent to the voter to his/her registered mobile number. It is a only one minute process which is 100% secured and on successful registration the user's data is saved into database in an encrypted form.

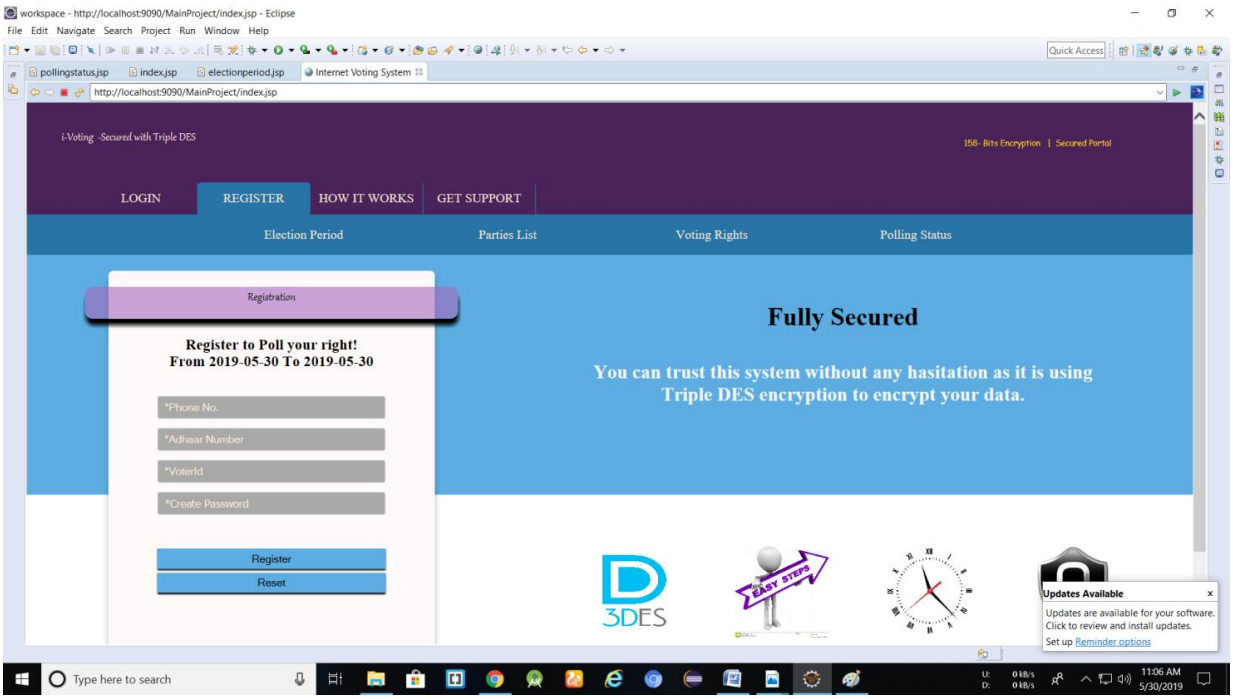


Fig 8.2 The Registration Page

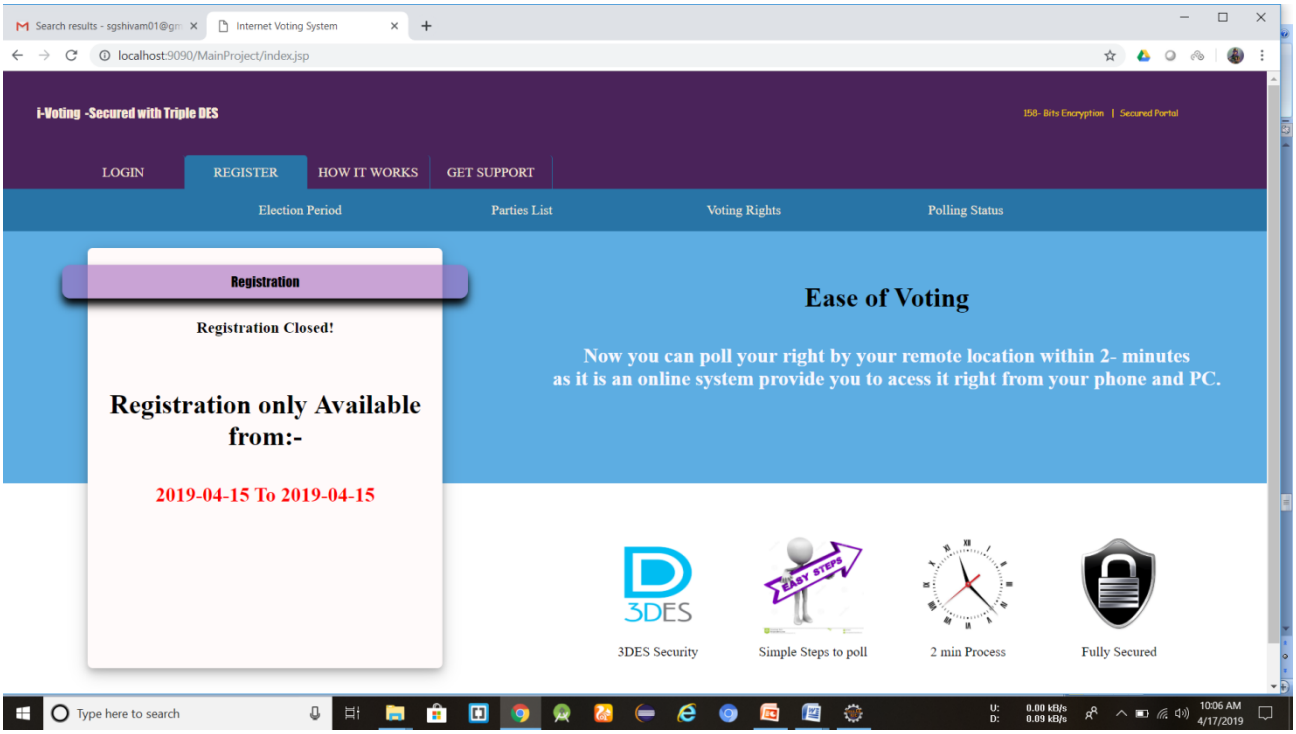


Fig 8.3 Registration Closed Error

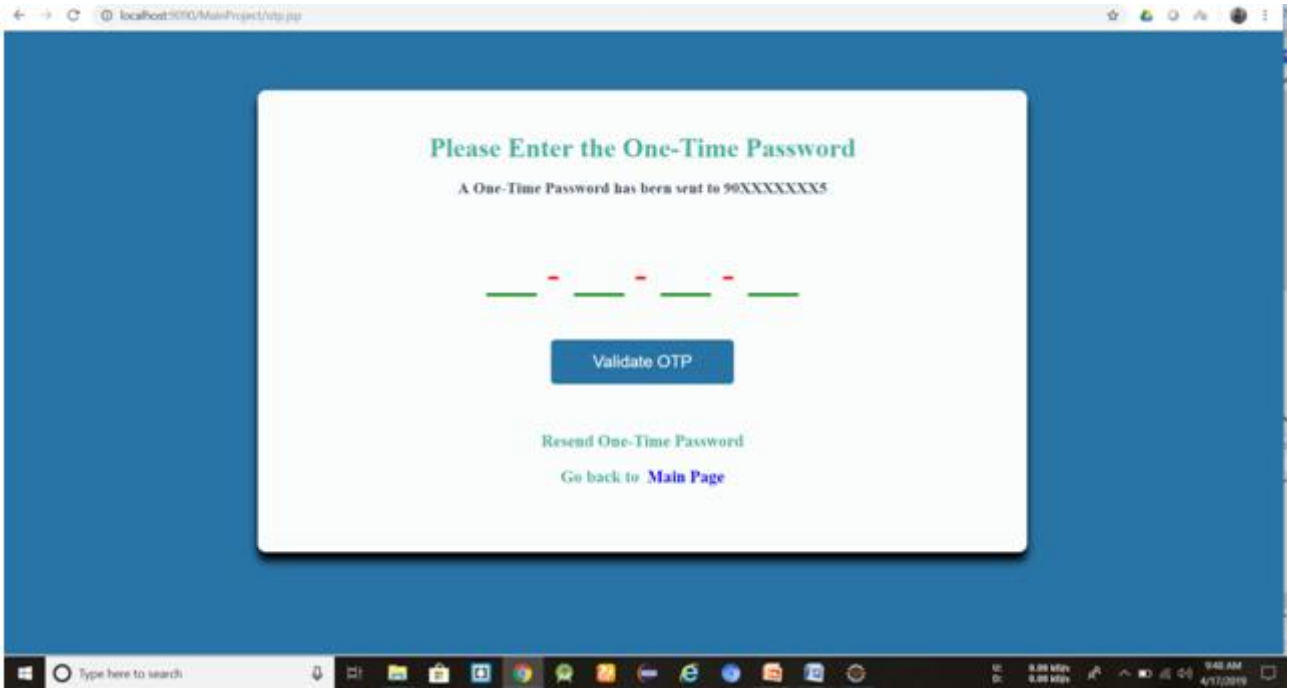


Fig 8.4 OTP Verification

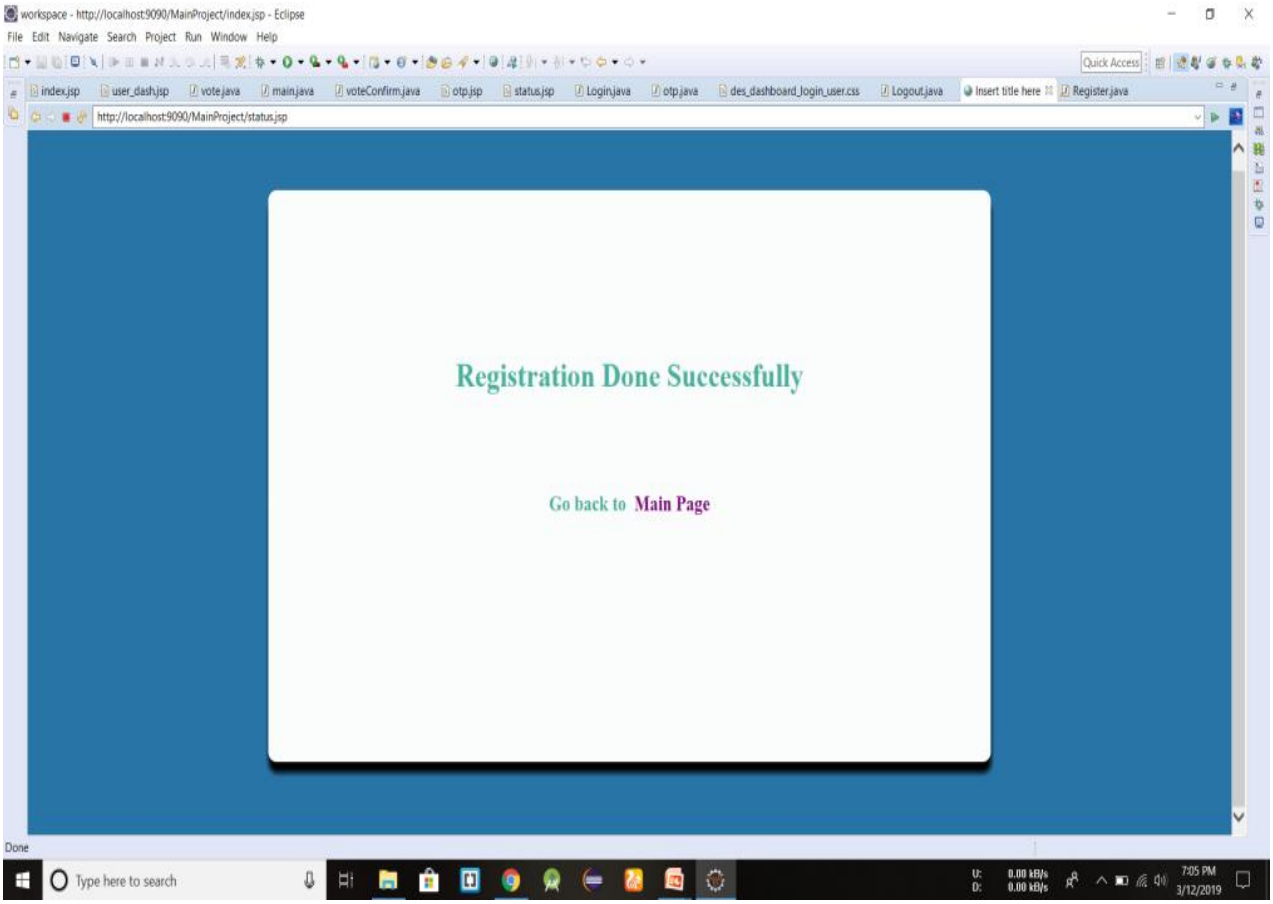


Fig 8.5: Registration Successful status

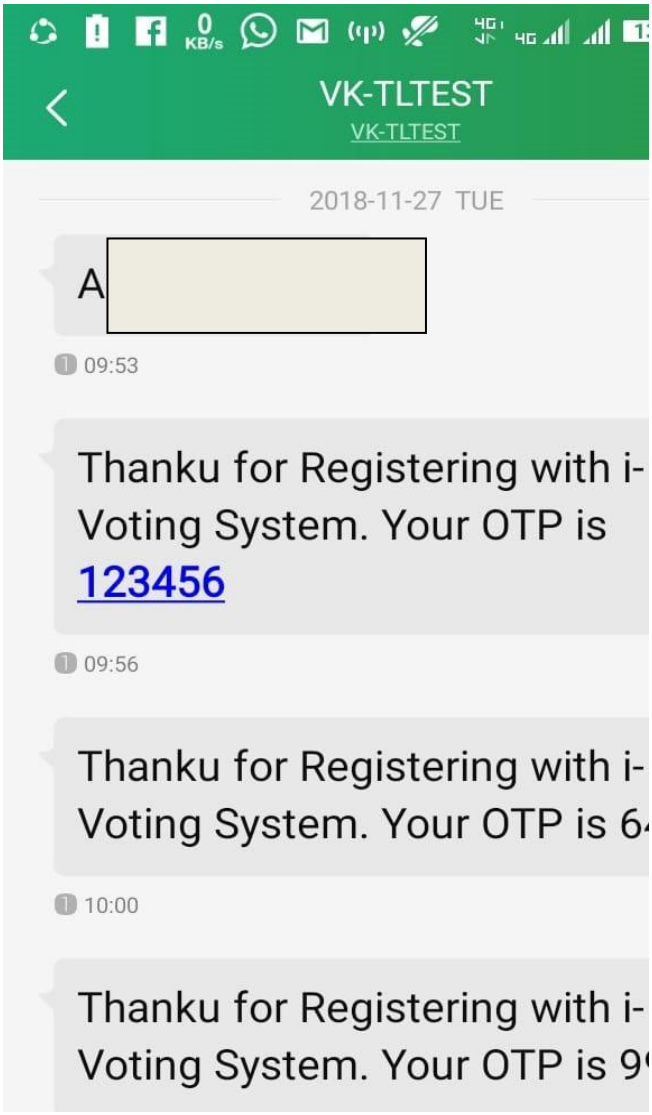


Fig 8.6 OTP Received

8.1.3 The Voter's Dashboard

This is the dashboard where a voter will reach after the successful login with its username and password type valid credentials. Here there are mainly two main options:

8.1.3.1 **Vote Now:** This option will take user to the voting page for the currently going election where user cast his/her votes after authencation with a One-Time-Password (OTP) only once.

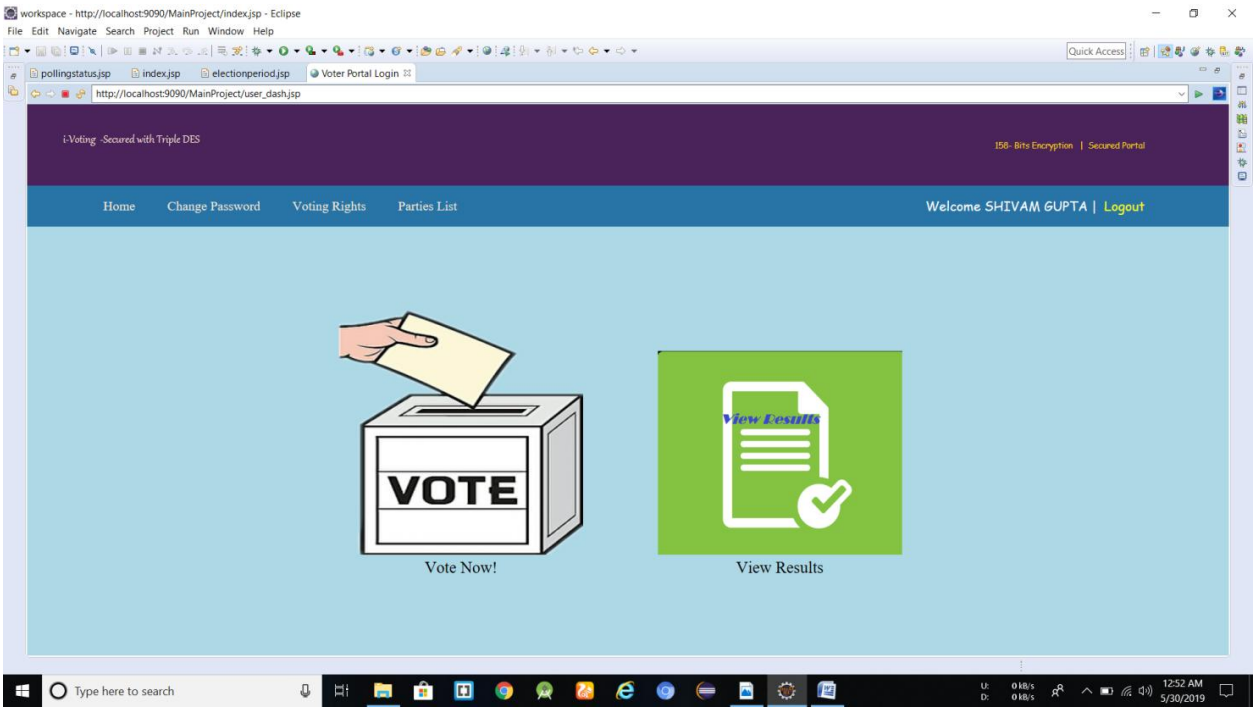


Fig 8.7: Voter's Dashboard

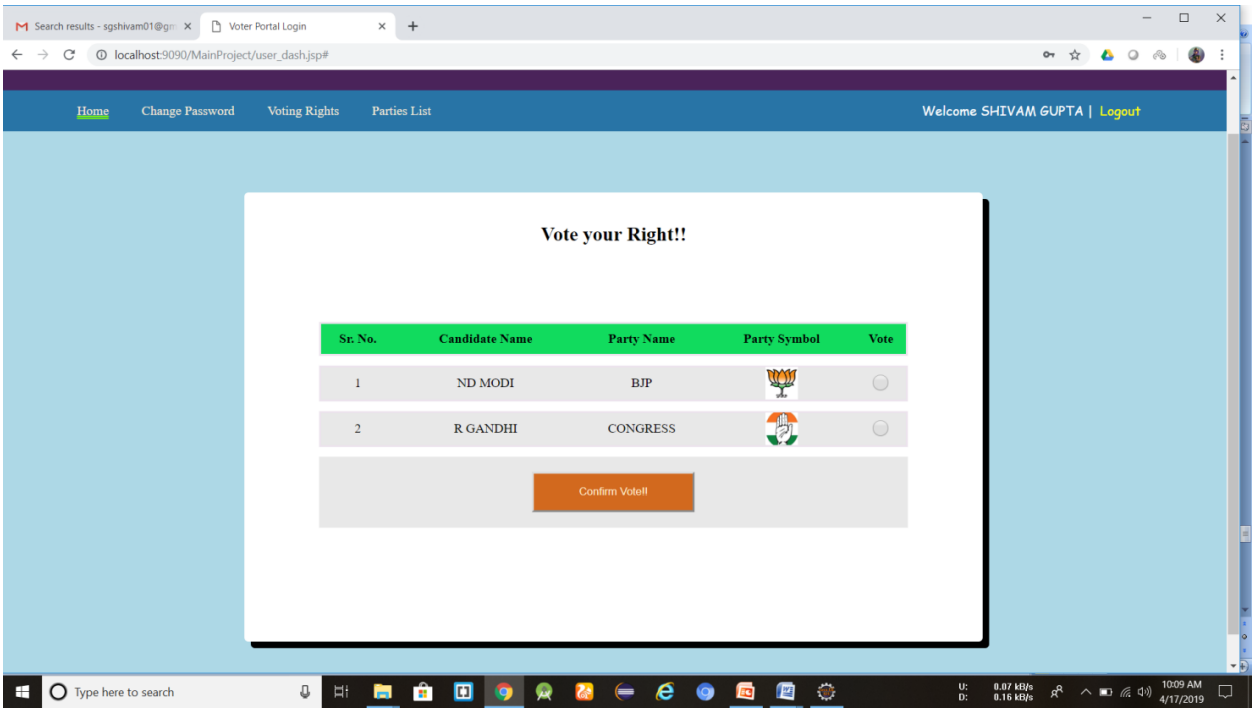


Fig 8.8: Voting Dashboard

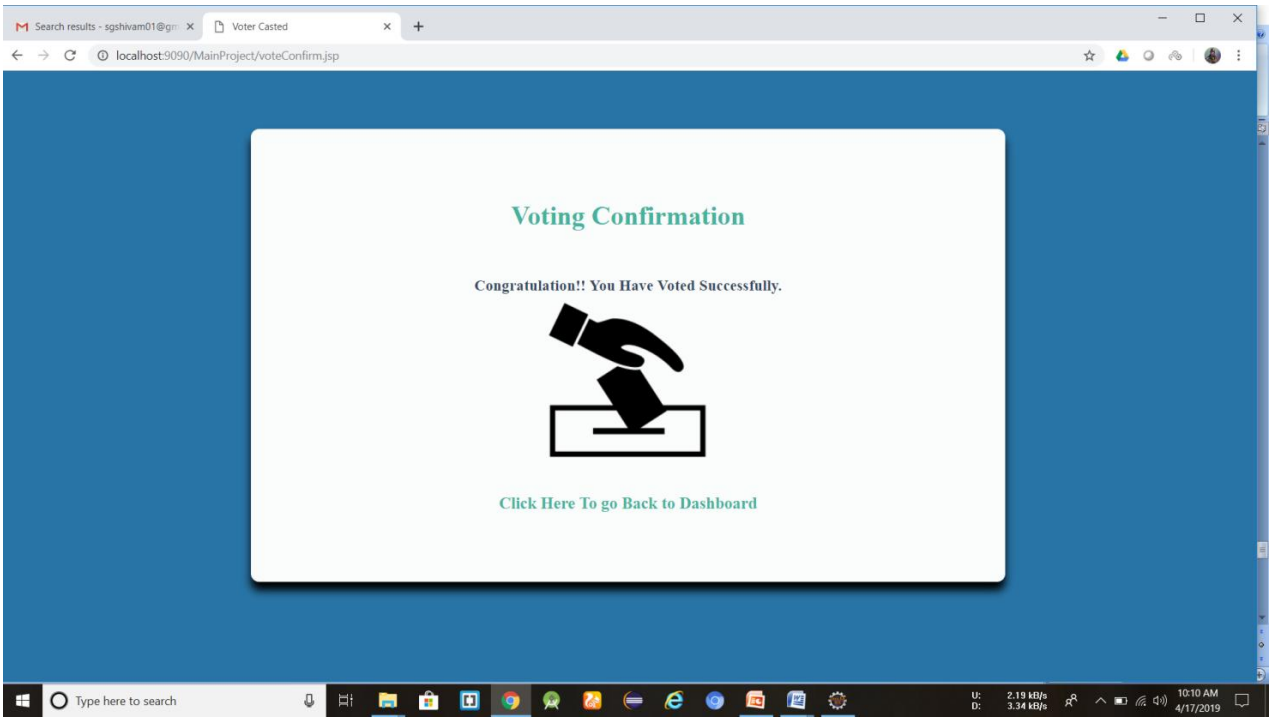


Fig 8.9: Voting Confirmation

8.1.3.2 Change Password: This option allows a logged in user to set a new password for his/her account.

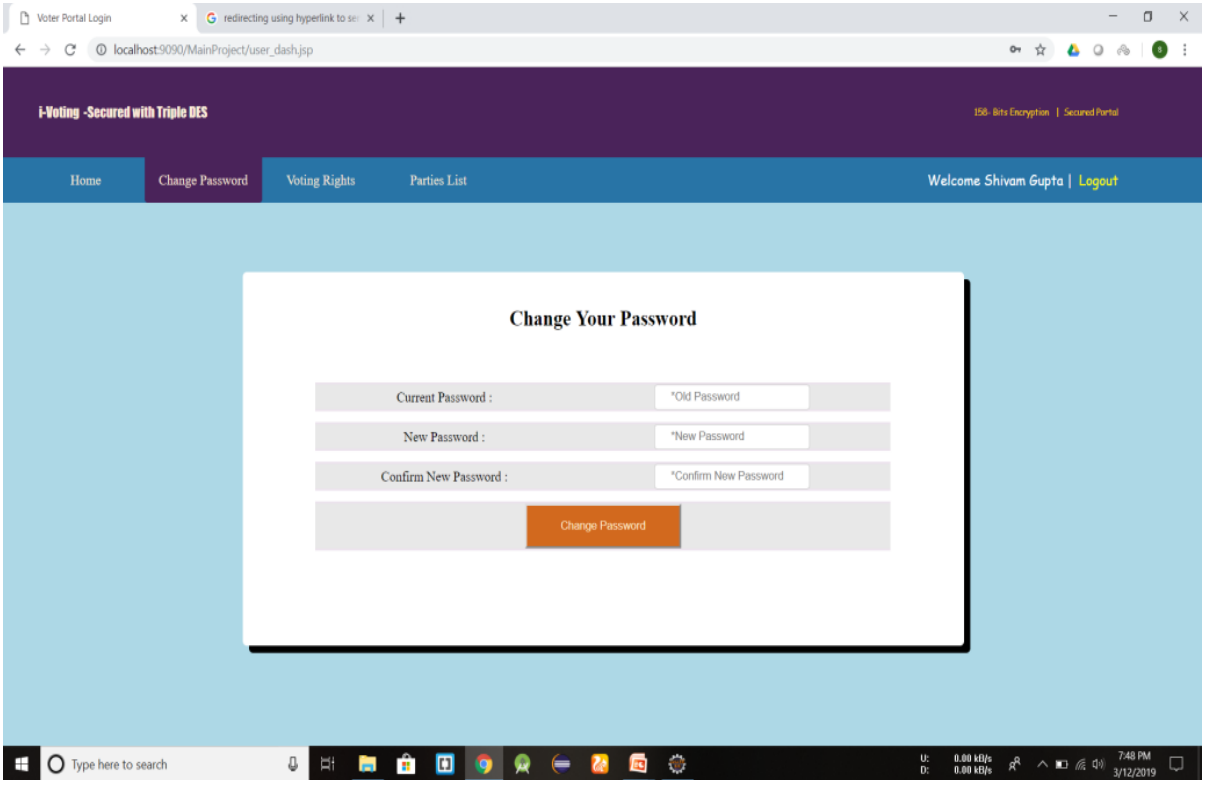


Fig 8.10: Change Password Option Screen

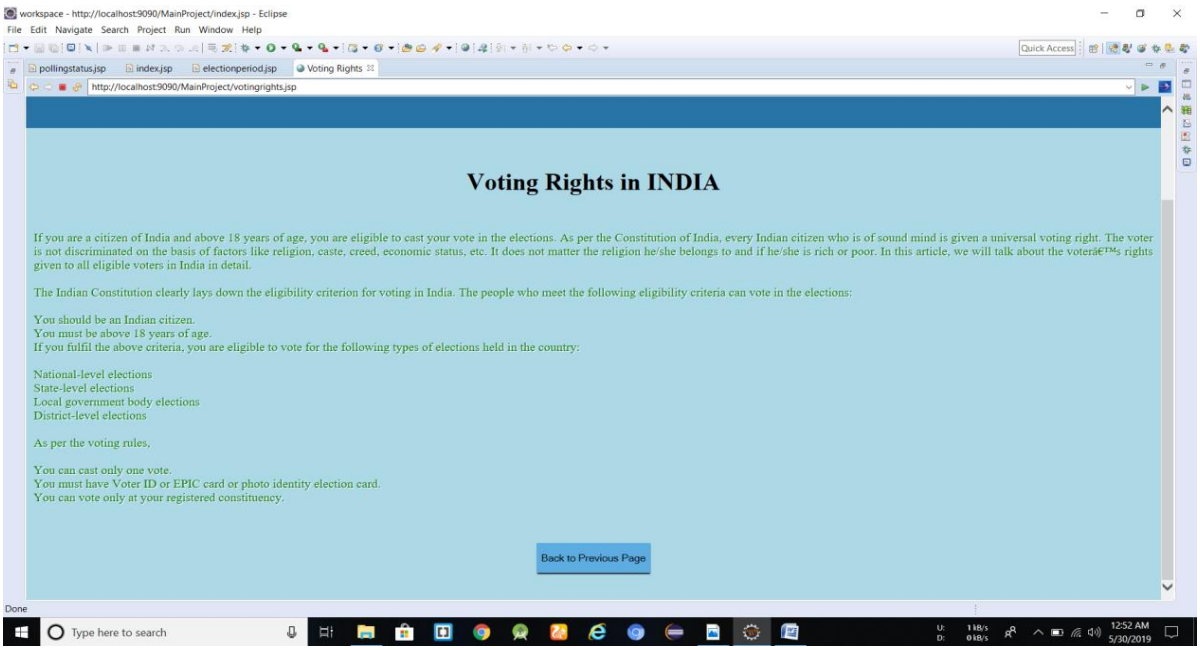


Fig 8.11: Voting Rights Option

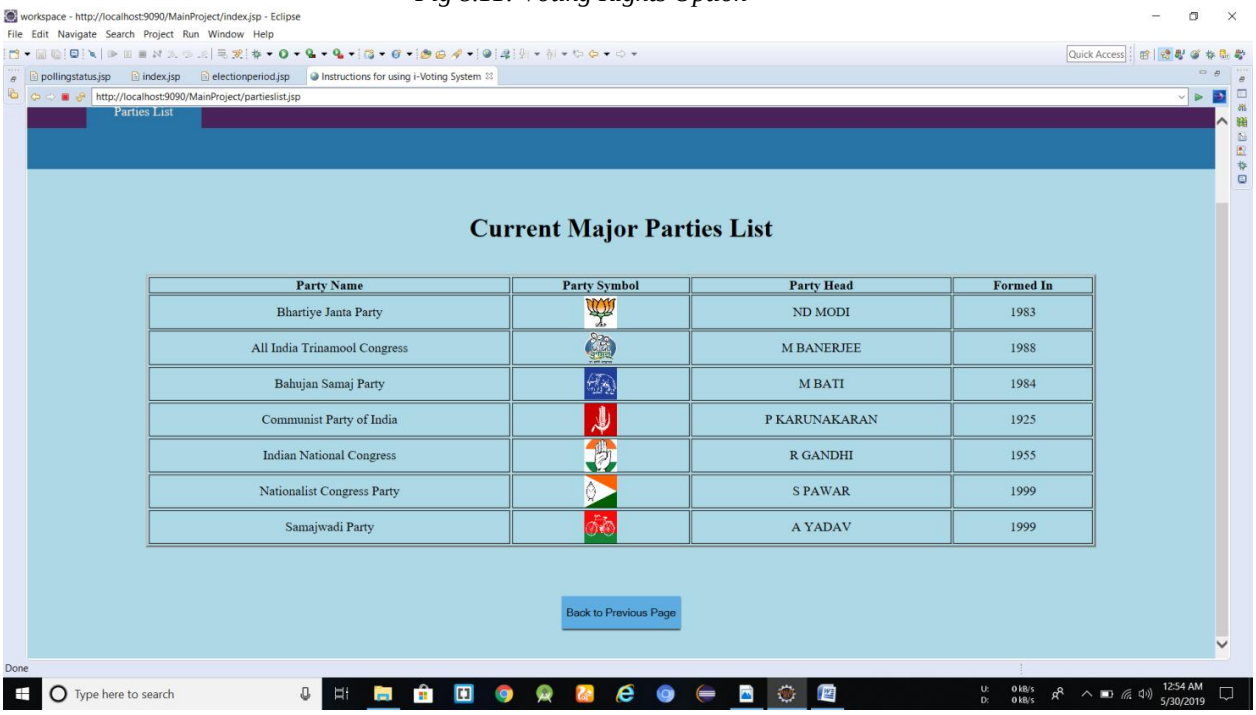


Fig 8.12: Parties List

8.2 The Admin Module

The administrator module is contained by the options used by the admin to manage the voting process like conducting elections, declaring results etc. It is only be accessible by the admin with its credentials.

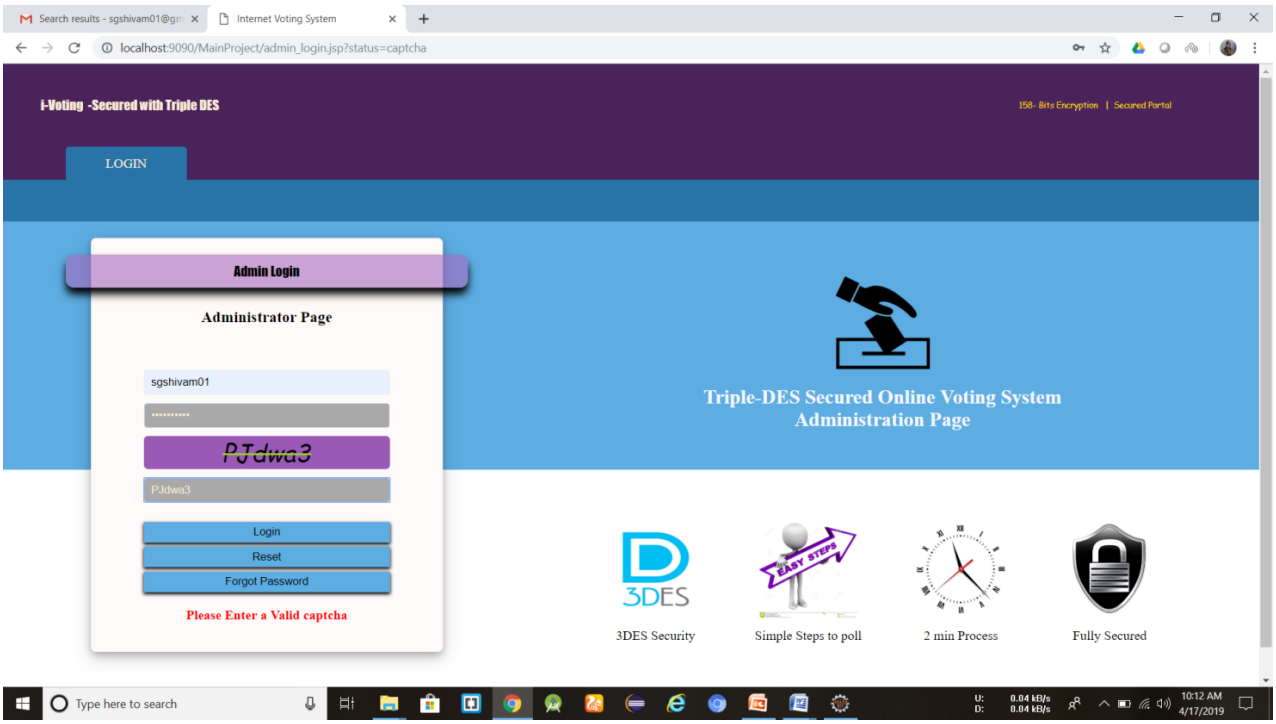


Fig 8.13: Admin Login Page

8.2.1 Admin dashboard

This is the page from where the admin manages the online voting process with the multiple available options after successful login.

8.2.1.1 Start New Election

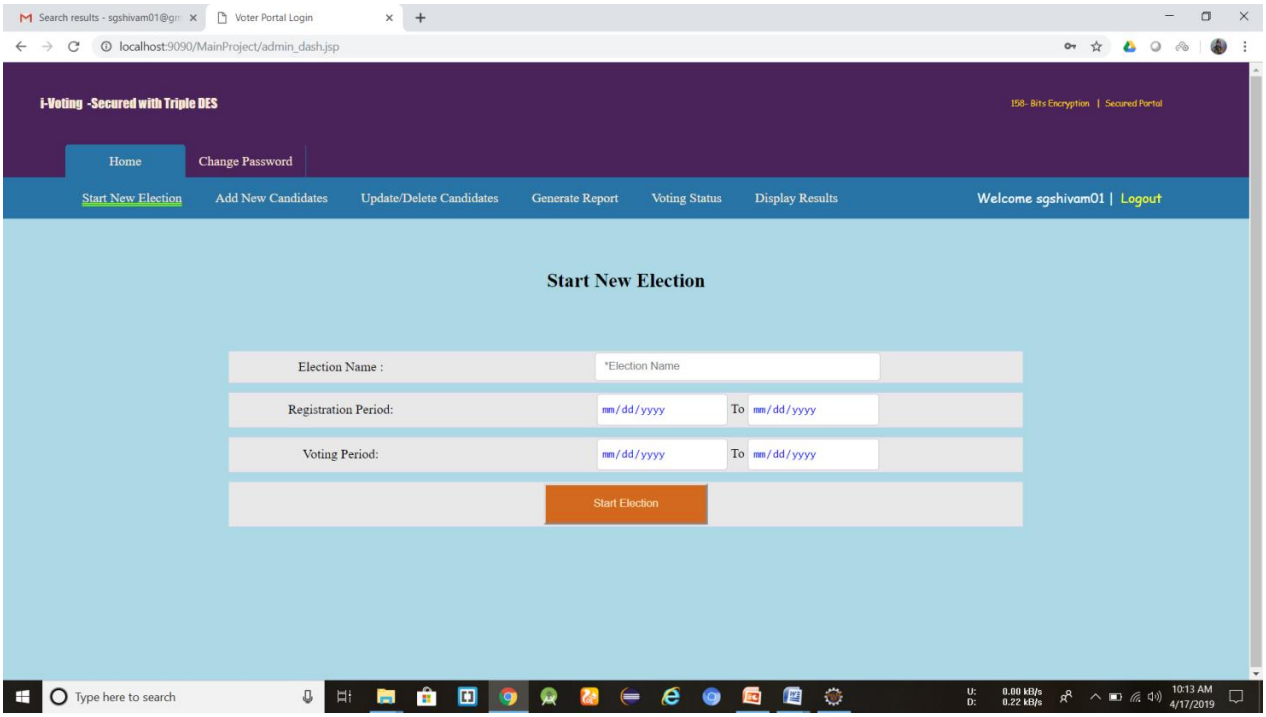


Fig 8.14: Start New Elections

8.2.1.2 Add New Candidate

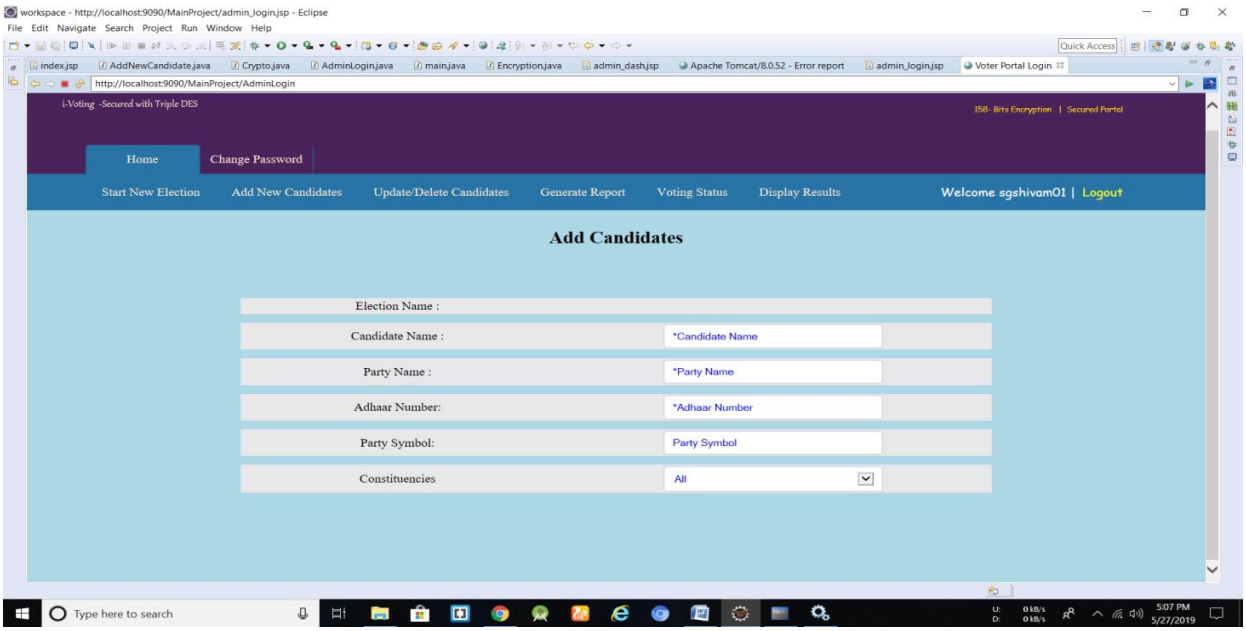


Fig 8.15: Add new candidate

8.2.1.3 Voting Status

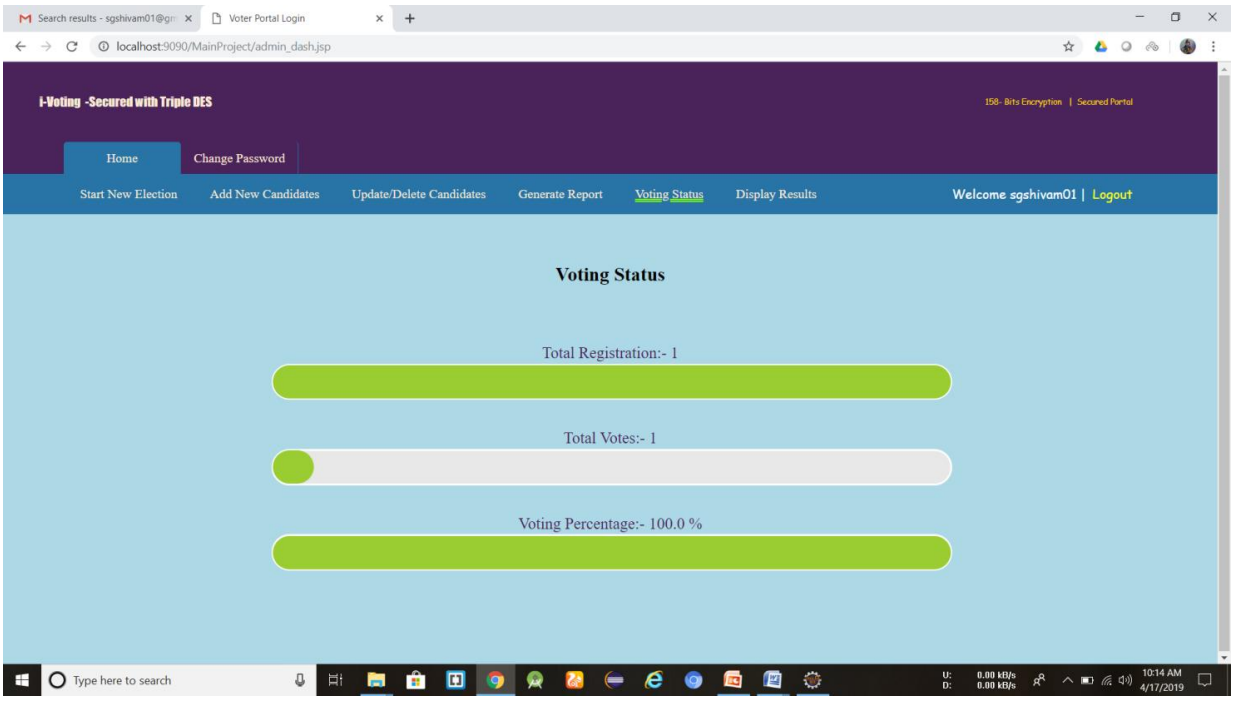


Fig 8.16: Admin Voting Status Option

8.3 Data Bases Tables

ADHAAR	NAME	FNAME	ADDR	MBNO	PASS
T7ak6TlmNo8ISW6WusJ1bQ==	WFCBn5jVhO6yVeAH+QvalQ==	wN2eNkiQaERCP8wyYepbfg==	dW9+3Ez4cMMaVAajr1brbNtQWp/vNTpB	cR1cSuyNRQ1u2AMD6ZRJ6w==	NaWmTNL4z0xoA7T+YYvqnw==

Fig 8.17: Encrypted DB table of Registered Voters

PARTY_NAME	COUNT
XFAYIIPb3j4=	+8sUU3Q0+KI=
B1UhkVChH00=	LvHw2ygMZv4=
b428oqQwtg4=	LvHw2ygMZv4=
w9223xpdx+0=	LvHw2ygMZv4=
KFcUvqZtGl4=	LvHw2ygMZv4=

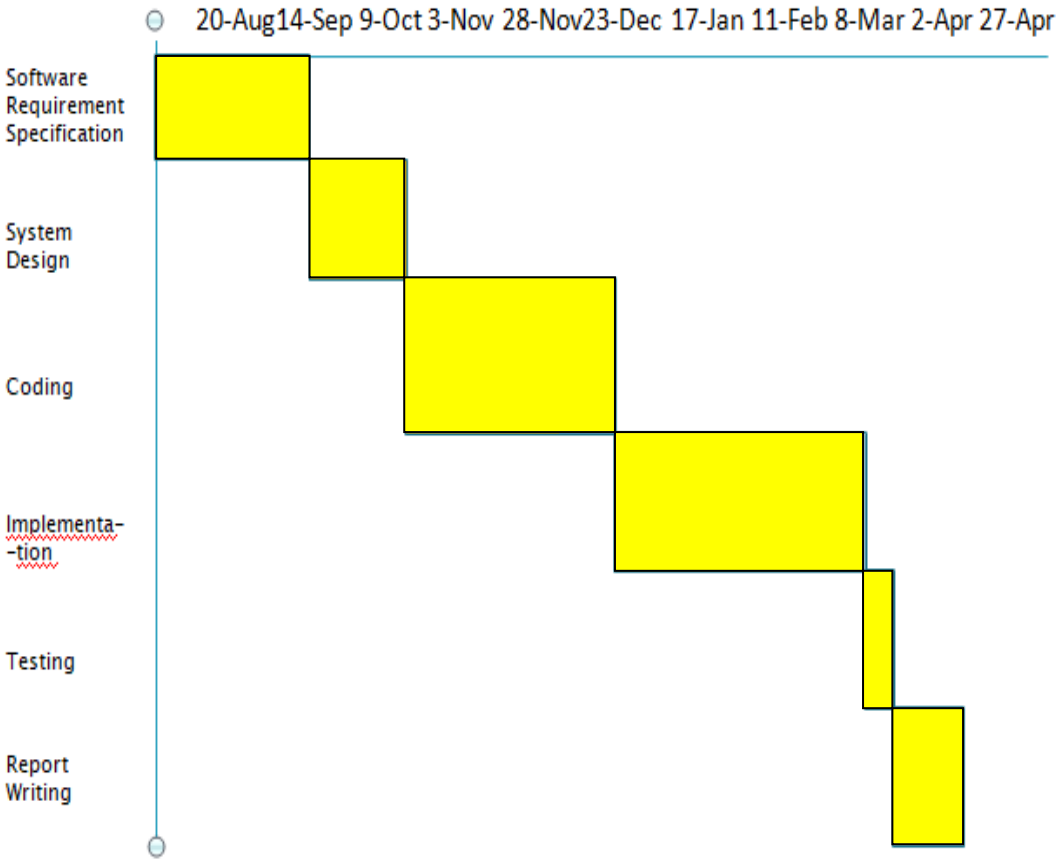
Fig 8.18: Encrypted DB table of Party's Vote Counts

SRNO	CANDIDATE_NAME	PARTY_NAME	PARTY_SYMBOL	IMAGE_ADD	IMG_ADD
1	N D MODI	Bhartiye Janta Party	KAMAL	BJP	symbols/BJP.jpg
2	R GANDHI	CONGRESS	PANJA	CNG	symbols/CNG.jpg
3	A YADAV	SAMAJVADI PARTY	CYCLE	SP	symbols/SP.jpg
4	M BATI	BAHUJAN SAMAJVADI PARTY	HATHI	BSP	symbols/BSP.jpg

Fig 8.19: DB table of Party and their candidates

CHAPTER 9

PROJECT PROGRESS



REFERENCES

[1] JAVA core and Advanced by Oracle

[2] Oracle PL/SQL Programming by Bill Pribyl and Steven Feurstein

[3] Servlet and JSP by Budi Kurniawan

[4] Applied Cryptography by Bruce Schneier

[5] Online Elections in Romania by Vlad Costea (Research Paper)

[6] Internet Voting for Expatriates: The Swiss Case by Uwe Serduit (Research Paper)

[7] Online Voting in Estonia by Pricha Lechsa (Research Paper)

[8] Online Voting System by Kamlakar Singh (Research Paper)

[9] www.ieexplore.org.in

[10] www.quora.com/online%voting%al%allowed%errors

[11] www.stackoverflow.com

[12] www.jsptutorial.com