

Chapter 1

Introduction (Online Voting System using MongoDB)

1.1. Motivation

As we know that Election procedure in India is a very hectic one. So many people stand in lines for hours, waiting to exercise their right to vote. Reaching the polling booth is also not a very easy feat. During the election days, the traffic reaches its peak and there are tonnes of traffic jams, which might cause a person to reach late and miss their voting chance. Also, it is not possible for everyone to go to a polling booth to cast their vote, the reason being that the person might be elderly or disabled. It is also possible that someone lives at a very remote location where there is no possibility of creating a polling booth. While all these situations have existed for a very long time, some very new problems have arrived in the last year, as we know that the Coronavirus Pandemic is at its peak, and the social distancing is of utmost importance during these days. Using the same voting machine by everyone is not at all sanitary. Also, during the lockdown period, it is not possible to conduct elections on a large scale, without risking the lives of people.

The solution to all these problems can be an Aadhar based Online Voting System for Online Elections. The Internet reach has increased very widely due to the Digital India Program and one of its best uses would be to conduct online elections. Everyone can vote from the ease of their home and without risking their health in the process. The unique encryption system provides the safety that duplication of the votes is not possible.

All the above-mentioned reasons have been our motivation to create this software.

1.2. Aim of the Proposed System

Our system provides the end-user (people who are voting) an online registration form with details of the available candidates for their constituencies and collects their vote for their preferred candidate. It allows System administrators to Add / Remove users from the Database of eligible votes, Add / Remove candidates from various parties who are standing for elections along with their respective details.

1.3. Objectives

The objectives of the Online Voting System are:

- The system must be robust without vulnerabilities or crashes.
- Have a user-friendly interface to simplify the election process.
- Have a user validation system to prevent illegitimate votes for candidates.

- Must be secure from external attacks.

Chapter 3

Software Requirement Specification

3.1 Hardware Requirements

The following hardware requirements should be met for the software to work in a computing environment:

- Intel Pentium or processor with minimum speed 2.10GHz
- Minimum 1GB RAM
- A minimum capacity of 500MB (Hard Disk)
- A displaying monitor
- Accelerated graphics card
- Better performance with 250MB and above

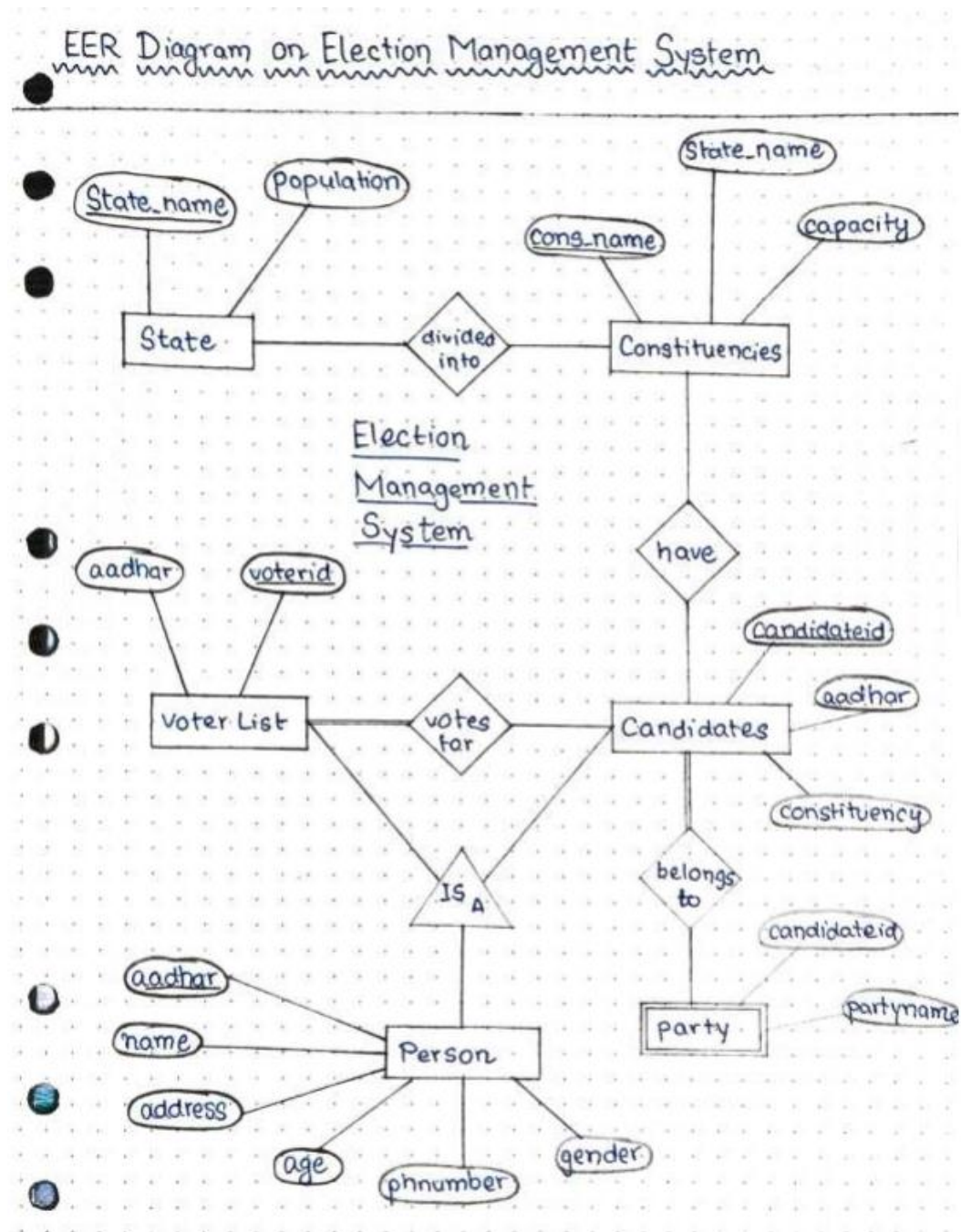
3.2 Software Requirements

The following software requirements need to be met for actuating the application within the system:

- Any operating system with GUI (Graphical User Interface) compatible
- Browser for accessing the website
- DirectX 9 for better display
- Frontend – HTML 5, CSS 3 and JavaScript
- Backend – NodeJS and Express framework • Database- mongo DB 4.2.8

Chapter 5

Entity Relationship Diagram



Chapter 6

Tables

Voter Details:

Table 1.1. Voter Details.

Attributes	Data Type
Name	String
User Name	String
Password	String
Email	String
Phone Number	Number
Aadhar Number	Number
Time Created	Date
Hash	String
Candidate Voted	String

```
var UserSchema = new mongoose.Schema({
  username: {type:String, unique:true},
  password:String,   email:String,
  voterid:{type:String, unique:true},
  phoneno:Number,   candidate:String,
  adhaar:{type:Number,unique:true},
  created:{type: Date,default: Date.now}
});
```

Candidate Details:

Table 1.2. Candidate Details

Attributes	Data Type
Name	String
Age	Number
Party	String
Constituency	String
State	String
ID	Number
Vote Count	Number
Information	String
Photo	Image

```
var CandidateSchema = new mongoose.Schema({
  cname: String,   cage: Number,   cparty:
String,   cinfo: String,   cimage: String,
  candidateid: { type: String, unique: true },
  constituency: String,   cstate: String,
  ccount: { type: Number, default: 0 },
  created: { type: Date, default: Date.now },
});
```

Admin Details:

Table 1.3. Admin details

Attributes	Data Type
------------	-----------

Username	String
Password	String

```
var AdminSchema = new mongoose.Schema({  
  user: String,   password: String,  
});
```

Chapter 7

Forms and Reports

GUI forms:

The System will work on a Web Browser like Google Chrome or Mozilla Firefox.

- **Home Page:** This page will be the main page that appears in front of the voter when they open the voting panel.

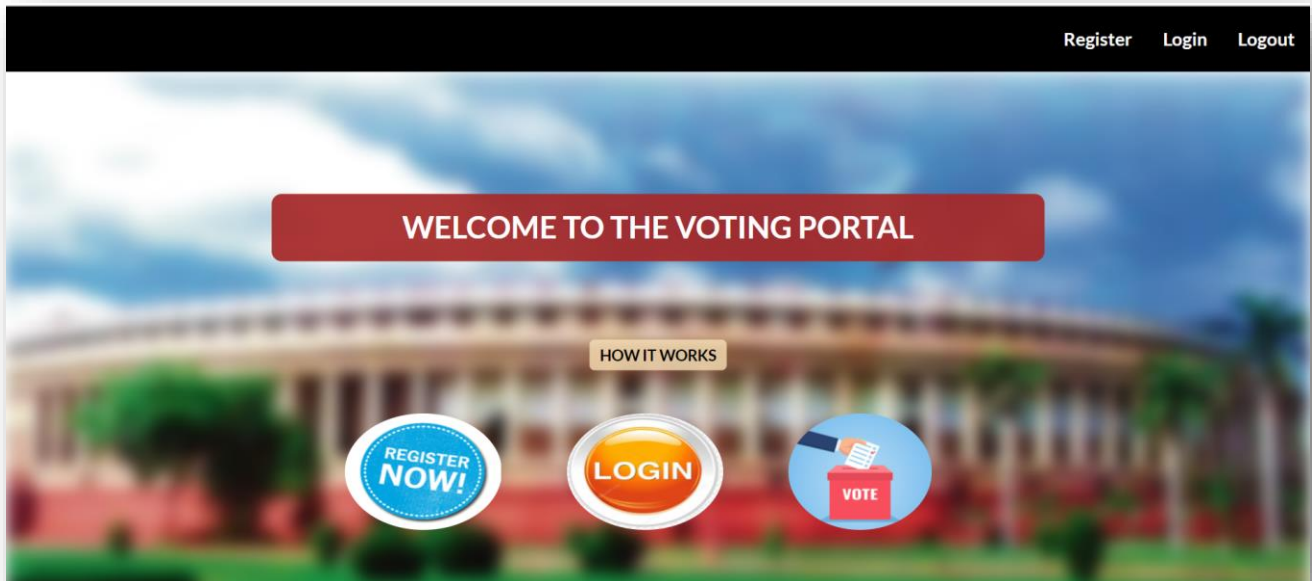


Fig 7.1: Home Page

-

Voter Registration Page: This page will contain the registration form to sign up as a voter.

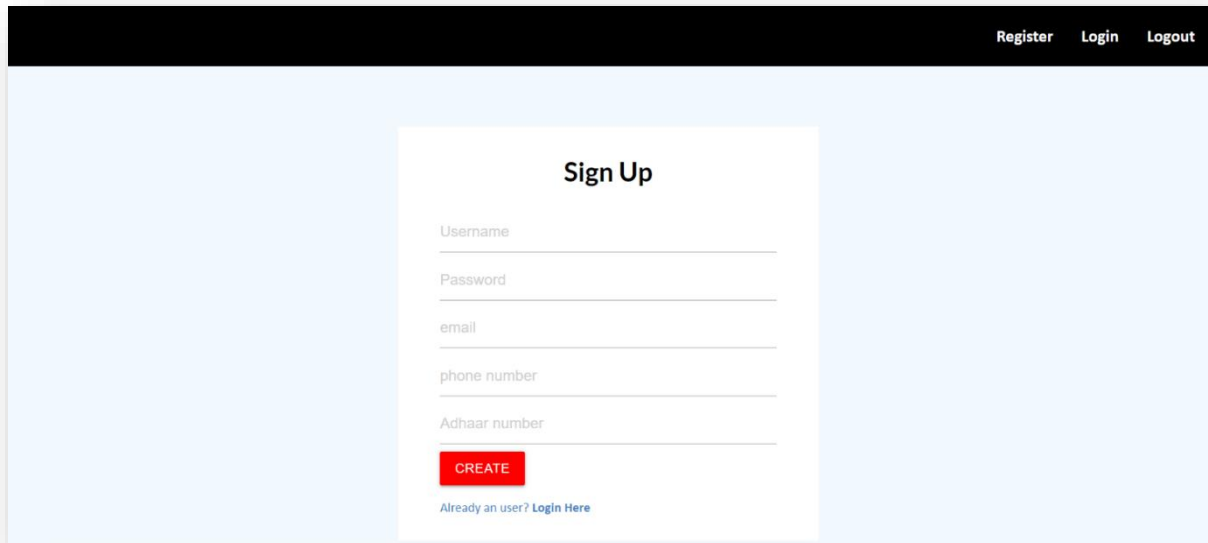


Fig 7.2: Sign up page

- **Voter Login Page:** This module will contain a login page after the user has successfully registered as a voter and is ready to cast their vote.

Register Login Logout

Login

Only for authorized voter! ✕

Username

Password

Unique Voter Key

LOGIN

Fig 7.3: Login page

Add Candidate Page: This module will contain the Nomination Form for a person to fill as a candidate.

New Candidate

Candidate Name

Name

Candidate Age

Age

Party name

Party Name

Constituency

Constituency Name

State

State Name

Candidate ID

Candidate ID

Vote count

Vote count

Candidate Information

Candidate Photo

Candidate photo

Submit

Fig 7.4: Add Candidate page

-
- **Voter Authorization Page**: This page allows for the administration to authorize a person as a valid voter and create their unique credentials for casting the vote.

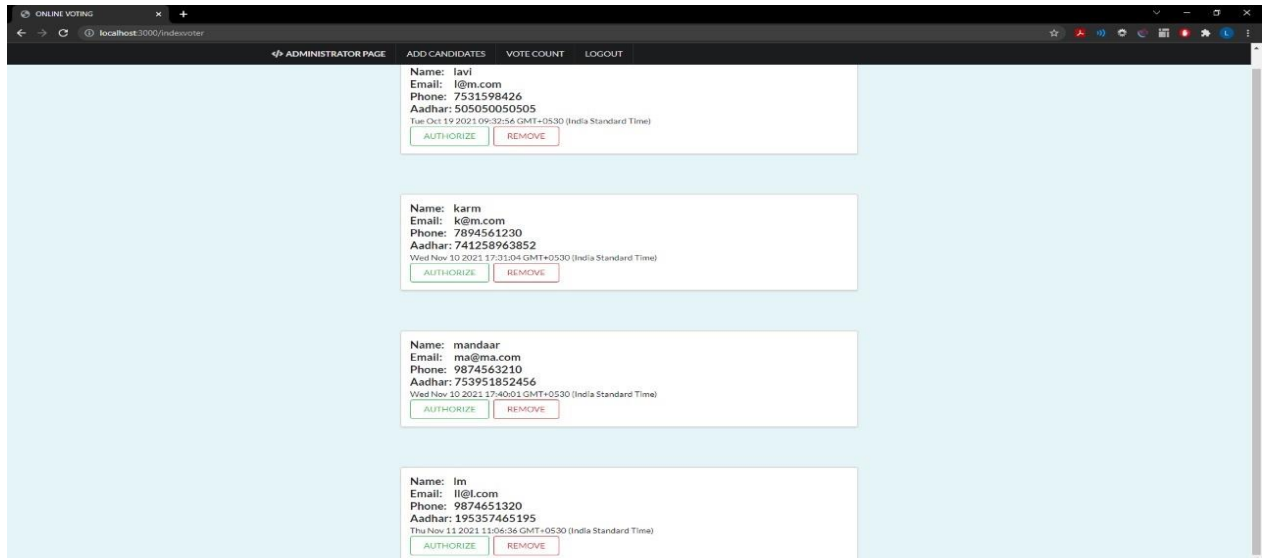


Fig 7.5: Authorize Voter page

Voting Page: This page will show the list of candidates that a voter can vote. Along with each candidate, their photo, party name, age and description will be provided.

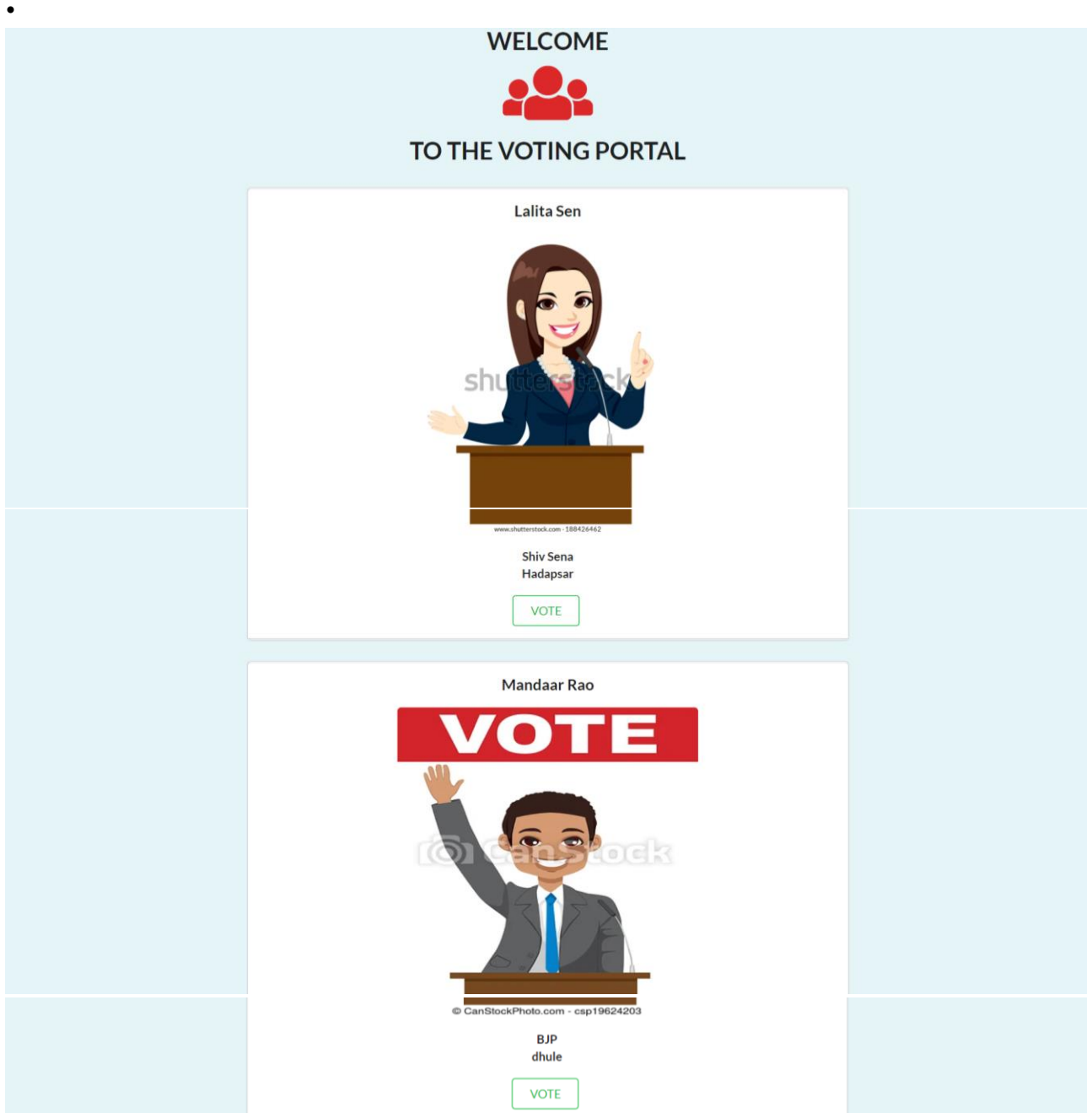


Fig 7.6: Select a candidate page

- **Vote Count Page:** This page will contain the status of votes casted to each candidate.

Here the admin will be able to see the number of votes, each candidate has received.

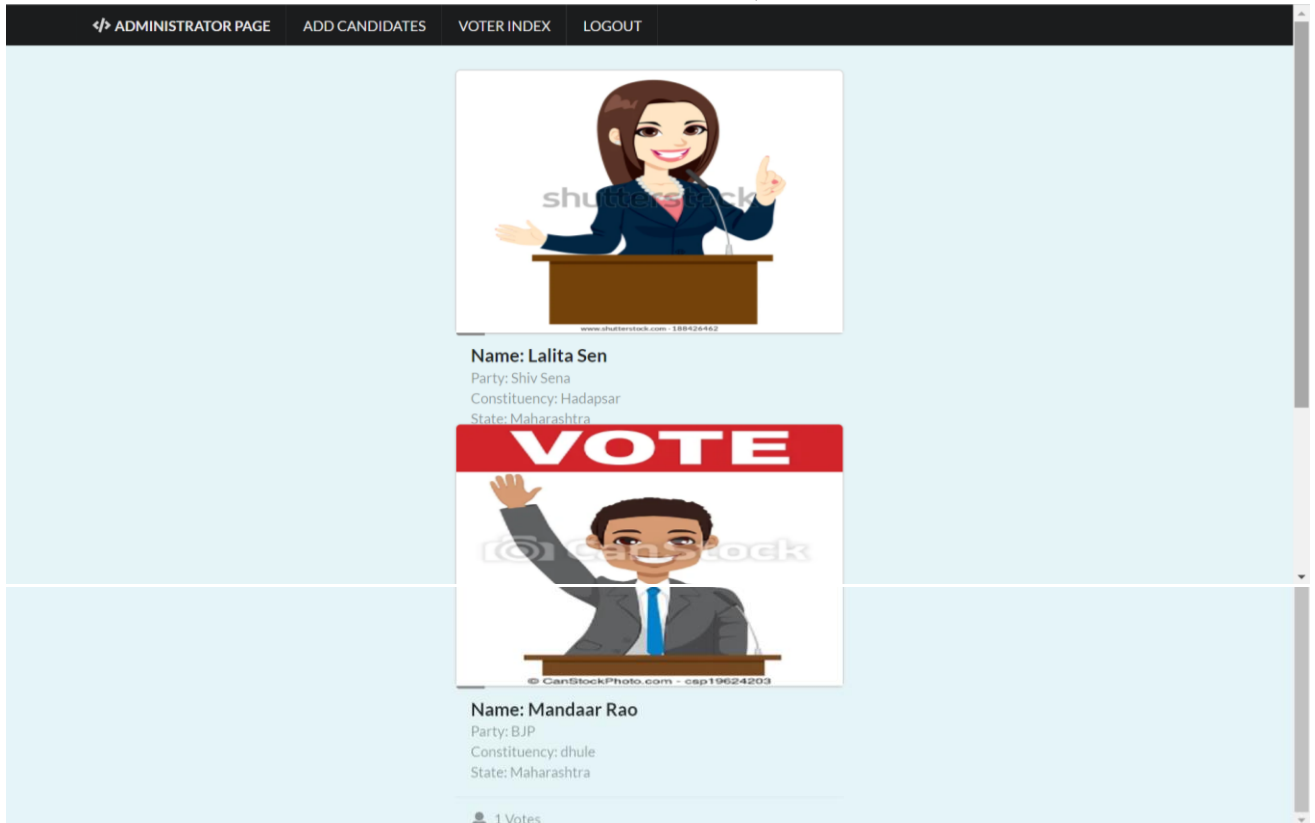


Fig 7.7: Candidate vote tally page

Conclusion

Through this implementation, we have created a system which allows users to skip the tedious process of standing in long lines or travel long distances to reach common voting centres. Citizens can now vote from the ease of their home and thus can increase the voter turnout. This becomes increasingly relevant during the COVID-19 pandemic as large public gatherings can be avoided thus preventing the transmission of the disease. Compared to the existing alternative where voters have to fill mail ballots and send them in, our system does not depend on manual counting. It also offers increased security and transparency with the help of blockchain technology. All these features make our system easy to use, secure and transparent.

Chapter 10 Future Scope

The Following future expansions are expected to be carried out in the system:

- Automatic Mailing system to deliver the Unique voting credentials to the emails of the registered voters automatically.
- Automatic SMS/Texting System to deliver the Unique voting credentials to the Phones of the registered voters automatically via text messages.
- Expansion of databases to be able to support the huge citizen and candidate database, during the real elections.
- Distributed and Parallel Processing to be introduced into the system, so that the system can take the advantage of multicore processing and the data among all the running instances of the software to be synchronised into a single database for ease of calculating the votes.
- Addition of an automatic result declaration and display system that allows the Election Commission to declare results without any human interference or calculations like tallying up votes from different constituencies. And results could be directly displayed on the software portal itself.