A Real Time Research Project Report on

**Employee Management System**

Submitted in partial fulfillment for award of the degree of Bachelor of Technology in

### Computer Science and Engineering

by

**K. Kalpana 22321A0537**

**P. Lavanya 22321A0542**

**K. Mounika 22321A0557**

Under the esteemed Guidance of Coordinator

G. Divya Vani Assistant Professor, CSE



Bhoj Reddy Engineering College for Women

## Department of Computer Science and Engineering

(Sponsored by Sangam Laxmibai Vidyapeet, approved by AICTE & affiliated o JNTUH)

### Vinaynagar, IS Sadan Crossroads, Hyderabad – 500 059

Ph: +91-40-2453 7282; Website: [www.brecw.ac.in](http://www.brecw.ac.in/); Email: [principal@brecw.ac.in](mailto:principal@brecw.ac.in)

2023- 2024



Bhoj Reddy Engineering College for Women

(Sponsored by Sangam Laxmibai Vidyapeet, approved by AICTE & affiliated to JNTUH)

Vinaynagar, I S Sadan Crossroads, Hyderabad 500 059, Telangana Telephone: 040-2453 7282, Website: [www.brecw.ac.in,](http://www.brecw.ac.in/) Email: [principal@brecw.ac.in](mailto:principal@brecw.ac.in)

Date:05.07.2024

**CERTIFICATE**

This is to certify that the Project Report entitled **“Employee Management System”** is a bonafide work carried out by

**K. Kalpana 22321A0537**

**P. Lavanya 22321A0542**

**K. Mounika 22321A0557**

in partial fulfillment for award of the degree of Bachelor of Technology in Computer Science and Engineering department from Bhoj Reddy Engineering College for Women, Hyderabad, affiliated to Jawaharlal Nehru Technological University Hyderabad (JNTUH).

**Internal Guide Head of the Department**

**Dr B Raveendranadh Singh**

**External Examiner**

Sangam Laxmibai Vidyapeet is an educational society for promotion of education among girls and women.

It is established in 1952 and registered under the Telangana Societies Registration Act.

#### ACKNOWLEDGMENT

It is our pleasure to express our whole-hearted thanks to Dr B Raveendranadh Singh, Professor & Head of Computer Science and Engineering Department for providing the necessary facilities to undergo this project.

We are thankful to our project coordinator G. Divya vani, Assistant Professor, Computer Science and Engineering Department, for giving us this opportunity to do this project.

We are also thankful to our project reviewer P. Deepthi, Associate Professor, Computer Science and Engineering Department for his/her extreme guidance and support in completing this project successfully.

We are also thankful to the staff members of Computer Science and Engineering Department, my friends and to our parents who helped us in completing this project successfully.

By

K..Kalpana (22321A0537)

P. Lavanya (22321A0542)

K.Mounika (22321A0557)

**Contents**

|  |  |  |
| --- | --- | --- |
|  |  | Page No |
| List of Figures | | i |
| List of Tables | | ii |
| Abbreviations Abstract | | iii iv |
| **1.** | **Introduction** | **1-3** |
|  | 1.1 Introduction of the project | 2 |
|  | 1.2 Existing system | 2 |
|  | 1.3 Problems in existing system | 2 |
|  | 1.4 Proposed system | 2 |
|  | 1.5 Advantages of proposed system | 3 |
| **2.** | **Requirements Analysis** | **4-9** |
|  | 2.1 Functional requirements | 5-6 |
|  | 2.2 Non-functional requirements | 7-8 |
|  | 2.3 Computational resource requirements | 7-8 |
|  | 2.3.1 Hardware resources | 7 |
|  | 2.3.2 Software resources | 8 |
|  | 2.4 Life cycle model | 8-9 |
| **3.** | **Architecture** | **10-19** |
|  | * 1. Architectures      1. Software Architecture      2. Technical Architecture | 11-12 |
|  | 11 |
|  | 12 |
|  | 3.2 Use case diagram | 12-13 |
|  | 3.3 Class diagram | 14 |
|  | 3.4 Sequence diagram  3.5 Activity diagram | 15-16  17-18 |
|  | 3.6 Database | 19 |

|  |  |  |
| --- | --- | --- |
| **4.** | **Implementation** | **20-24** |
|  | * 1. Technologies      1. HTML      2. CSS      3. Java Script      4. PHP      5. MySQL | 21-22  21  21  21  22  22 |
|  | 4.2 Pseudocode | 23-24 |
| **5.** | **Testing** | **25-30** |
|  | 5.1 Overview | 26 |
|  | 5.2 Dimensions of Testing | 26 |
|  | * 1. Stages of Testing      1. Unit Testing      2. Integration Testing      3. System Testing      4. Acceptance Testing | 27-28  27  27  27  28 |
|  | * 1. Types of Testing   2. Test Cases | 28  29-30 |
| **6.** | **Screenshots** | **31-37** |
| **7.** | **Conclusion and Future Scope**   * 1. Conclusion   2. Future Scope | **38-39**  39  39 |
| **8.** | **References** | **40-41** |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Fig.No** | **Figure Name** | **Page.No** |
| 2.1 | Spiral Model | 9 |
| 3.1 | Software Architecture | 11 |
| 3.2 | Technical Architecture | 12 |
| 3.3 | Use Case Diagram for User | 13 |
| 3.4 | Use Case Diagram for Admin | 13 |
| 3.5 | Class Diagram | 14 |
| 3.6 | Sequence Diagram for User | 15 |
| 3.7 | Sequence Diagram for Admin | 16 |
| 3.8 | Activity Diagram for User | 17 |
| 3.9 | Activity Diagram for Admin | 18 |
| 3.10 | Database | 19 |

|  |  |  |
| --- | --- | --- |
| 6 | Screenshots   * 1. Home page   2. Home Page   3. Election Page   4. Guidelines Page   5. Services Page   6. Login Page   7. Vote Page   8. Vote page   9. Admin Panel   10. Admin Panel   11. Locally storing encrypted votes | 31-37  32  32  33  33  34  34  35  35  36  36  37 |

**List of Tables**

|  |  |  |
| --- | --- | --- |
| **Table.No** | **Table Name** | **Page.No** |
| 5.1 | User | 29 |
| 5.2 | Admin | 29 |
| 5.3 | Vote | 30 |

**Abbreviations**

IDE: Integrated Development Environment OS: Operating System RAM: Random Access

UI : User Interface

UML: Unified Modelling Language

## Abstract

The Employee Management System (EMS) is a comprehensive solution designed to streamline and enhance the management of employee information and organizational workflows. The system integrates various functionalities, including employee record keeping, attendance tracking, payroll processing, performance evaluation, and leave management. By leveraging advanced technologies such as cloud computing and database management systems, the EMS provides a centralized platform that ensures data accuracy, security, and accessibility. This system aims to reduce administrative burdens, improve operational efficiency, and foster a more productive work environment. Additionally, the EMS offers robust reporting and analytics capabilities, enabling management to make informed decisions based on real-time data insights. The implementation of the EMS is expected to contribute significantly to the optimization of human resource management practices and overall organizational effectiveness.

Top of Form

Bottom of Form

# INTRODUCTION

* 1. Introduction of the project

## Introduction

In today’s fast-paced and competitive business environment, effective management of human resources is crucial for organizational success. An Employee Management System (EMS) serves as an essential tool in achieving this goal by providing a unified platform to manage various aspects of employee information and organizational processes. The EMS encompasses a wide range of functionalities, including maintaining employee records, tracking attendance, processing payroll, managing leaves, and conducting performance evaluations. Employee Management System project is designed to keep track of employee information in any company. It is a distributed application developed to maintain the details of employees working in any organization. Employee Management System project gives managers a better idea of their employees and helps them plan and manage their work hours to cut costs and boost productivity. It gives appropriate directions and supervisions for employees.

### Existing System

### The existing system, characterized by manual processes and isolated digital tools, presents significant challenges in terms of efficiency, accuracy, and security. There is a clear need for a more integrated and automated approach to employee management, which an advanced Employee Management System can provide. The transition to a comprehensive EMS promises to address these challenges by centralizing data, automating processes, and enhancing accessibility and security, ultimately leading to a more efficient and productive organizational environment. The company uses an application which is a single user system to find the employee information. The employee information are stored in the application which is like a excel sheet.

### Problems in Existing System

* The important and the most significant drawback is that the system is manual. There are errors due to carelessness or oversight that may result in loss to the data and as to the organization.
* For an organization, time is very important factor. Since the data is stored in the form of excel sheets etc, this makes it impossible to search for a company information in such a long table manually and to add a new query if the searched query is not available

### Proposed System

The proposed EMS aims to resolve these issues by introducing an advanced, digital platform that integrates time tracking, attendance management, and broader HR functionalities into a cohesive system.

### Advantages of Proposed System

* Easy attendance marking.
* Easy calculation of various leave in categories.
* Increased Productivity.
* Create new users to the system accordingly.
* Huge data storage with less computer memory.
* Improved Decision Making
* Reduce maintenance Costs

# REQUIREMENT ANALYSIS

## Requirement Analysis

### Functional Requirements

These functional requirements provide a comprehensive overview of the features that an Employee Management System should include to effectively manage employee-related processes and information.

Employee Management System consists of the following modules:

1. Admin Module

2. Employees Module

#### Admin Module:

* Add Employee
* View Employee Details
* Remove Employee
* Update employee details
* Manage finance
* Manage Attendance

#### Employees Module:

Employees module consist of Employee and HR or Project Manager

Employee

* View employee details
* View Financial details
* View project details

HR or Project Manager

* View Employee details
* View financial details
* Assign projects
* View projects
* Give performance rating

### Non - Functional Requirements

**Performance:** The form load quickly and the update operation is performed efficiently without noticeable delay.**Maintainability :** The code should be well-organized and commented to ensure it is easy to maintain and update in the future.**Reliability:** The application handle unexpected errors gracefully and provide meaningful feedback to the user.

### Computational Resource Requirements

#### Hardware Requirements

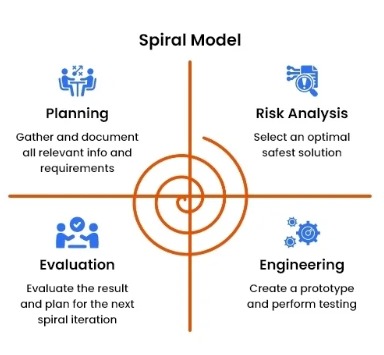
* + - * Processor : Intel i4
      * Hard Disk : 50GB
      * RAM : 4GB

#### Software Requirements

* + - * Operating System : Windows 10
      * IDE : Visual Studio Code
      * User Interface : HTML, CSS
      * Scripting language : JS
      * DataBase : Xampp Server
      * Server Deployment : Github
  1. Life Cycle Model

We use Spiral Model, it is a risk-driven model, meaning that the focus is on managing risk through multiple iterations of the software development process. It consists of the following phases:

* **Planning:** The first phase of the Spiral Model is the planning phase, where the scope of the project is determined and a plan is created for the next iteration of the spiral.
* **Risk Analysis:** In the risk analysis phase, the risks associated with the project are identified and evaluated.
* **Engineering:** In the engineering phase, the software is developed based on the requirements gathered in the previous iteration.
* **Construction:** During this phase, the software is actually built based on the specifications outlined in the previous phases. It involves coding, testing, integration of components, and system testing. The emphasis is on implementing the requirements gathered in the earlier stages.
* **Release:** Once a version of the software is deemed stable and functional, it is released to the customer or stakeholders. The release may be a prototype, an alpha version, or a beta version, depending on the progress and the agreement with the customer. Feedback from users and stakeholders gathered during this phase is vital for refining the software.
* **Customer Evaluation:** In the evaluation phase, the software is evaluated to determine if it meets the customer’s requirements and if it is of high quality.
* **Customer Communication:** Communication with the customer is continuous throughout the Spiral model. It involves regular interactions to gather requirements, discuss progress, share prototypes or interim deliverables, and collect feedback. This ongoing communication allows for adjustments and improvements based on the customer's evolving needs.



2.1 Spiral Model

# ARCHITECTURE

## Architecture

Project architecture represents number of components we are using as a part of our project and the flow of request processing i.e. what components in processing the request and in which order. An architecture description is a formal description and representation of a system organized in a way that supports reasoning about the structure of the system. Architecture is of two types. They are

1. Software Architecture
2. Technical Architecture

#### Software Architecture

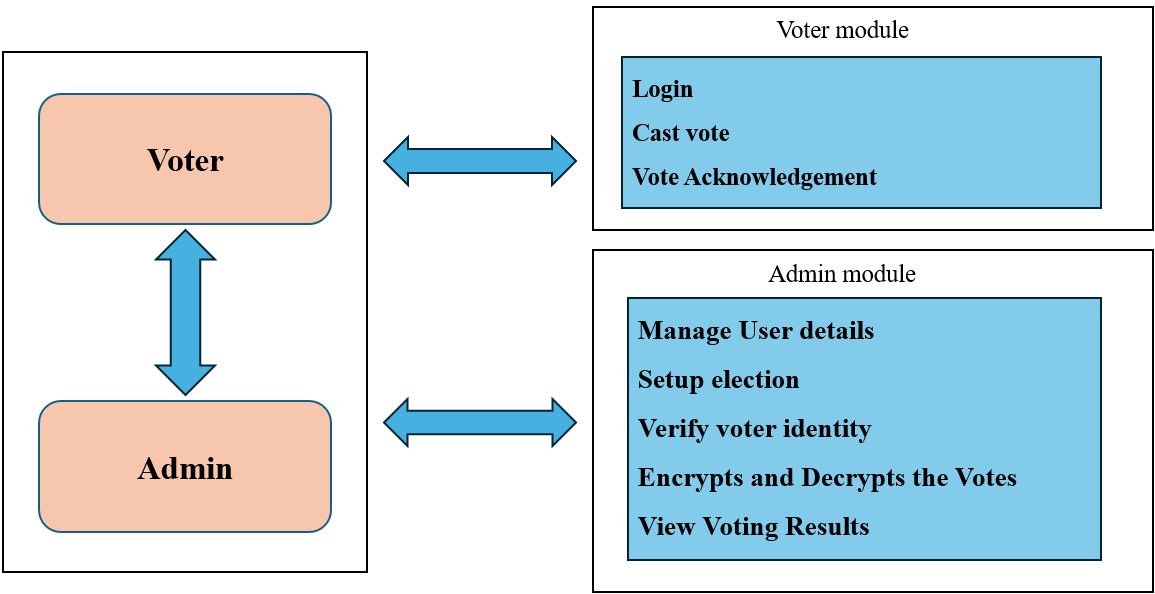
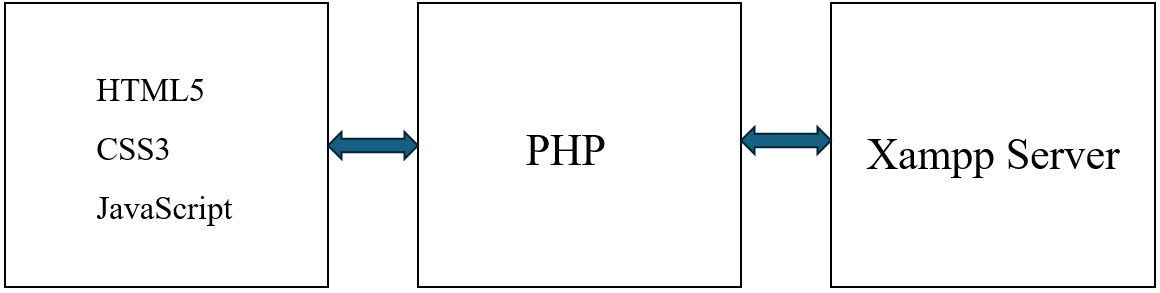
****

Fig 3.1 Software architecture

* + 1. **Technical Architecture:**

****

### UML Diagrams

Fig 3.2 Technical architecture

Design engineering deals with the Unified Modelling Language (UML) which is a standard language for writing software blue prints. The UML is a language for

* + - * Visualizing
      * Specifying
      * Constructing
      * Documenting the artifacts of a software intensive system

The UML is a language which provides vocabulary and the rules for the rules combing words in that vocabulary for the purpose of communication. A modelling language is a language whose vocabulary and the rules focus on the conceptual and physical representation of a system. Modelling yields an understanding of a system.

### Use Case Diagram:

Use case diagram are used to gather the requirements of a system including internal and external influences. A use case represents a functionality of a system. So, use case diagrams are used to describes the relationships among the functionalities and their internal/external controllers. These controllers are known as actors.

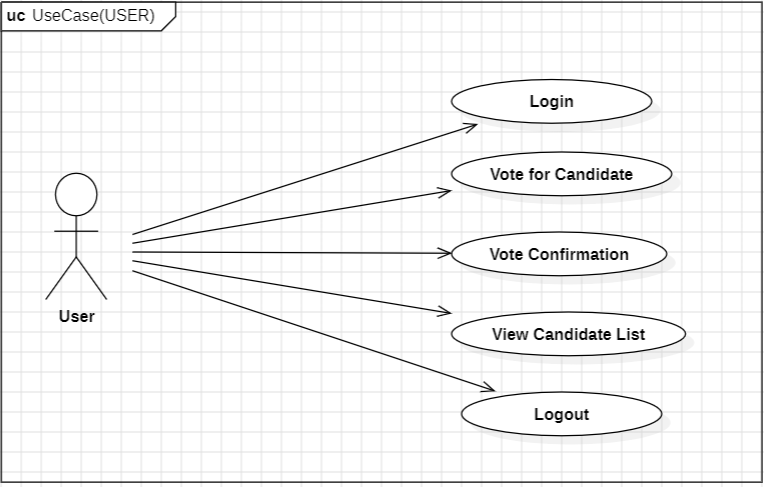


Fig 3.3 Use case diagram for User

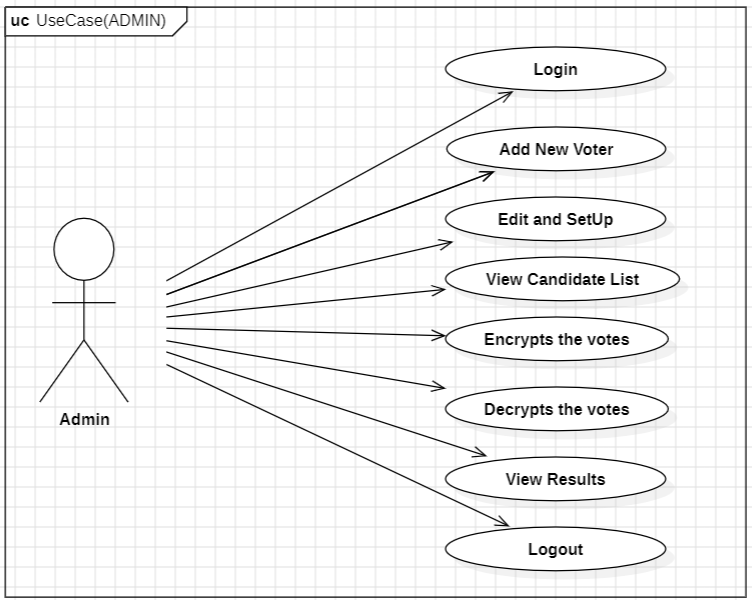


Fig 3.4 Use case diagram for admin

### Class Diagram:

A class diagram in UML represents the static structure of a system by showing its classes, attributes, operations, and relationships. Key components include classes (depicted as rectangles divided into sections for name, attributes, and operations), attributes (properties of a class), operations (methods of a class), and various relationships like associations, aggregations, compositions, generalizations, and dependencies. It is used for designing and modeling object-oriented systems, serving as a blueprint for software development and facilitating communication among developers and stakeholders. Class diagrams can be used for both forward and reverse engineering of systems.

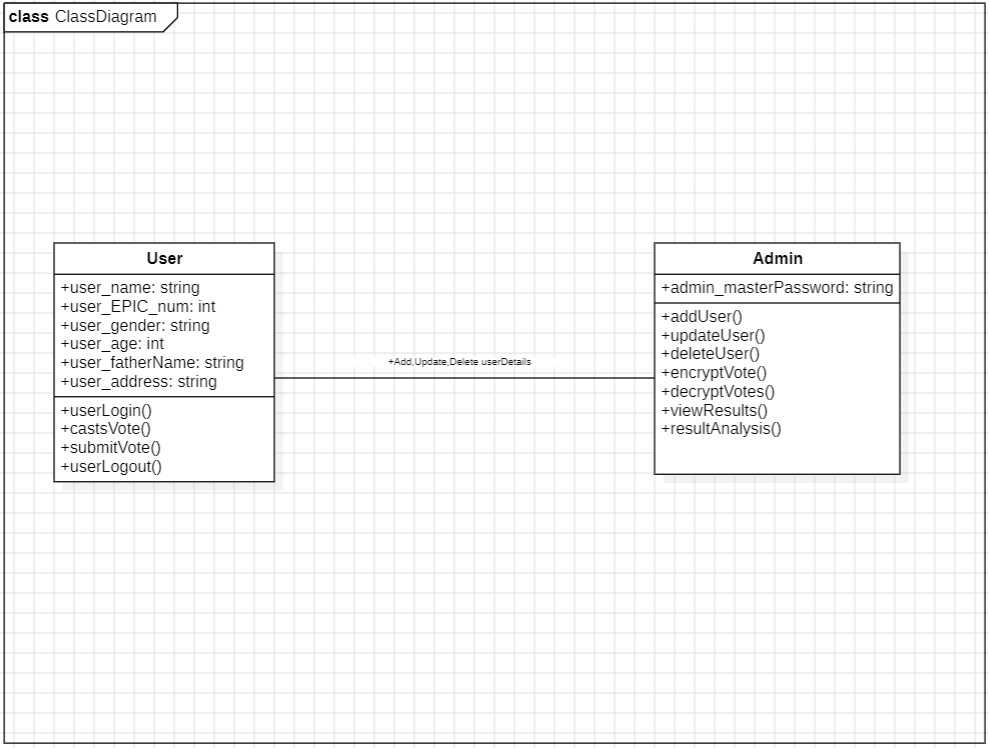


Fig 3.5 Class Diagram

### Sequence Diagram:

A sequence diagram illustrates the sequence of messages between objects in an interaction. A sequence diagram consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction.

**Purpose of Sequence Diagram:**

* Model high-level interaction between active objects in a system.
* Model the interaction between object instances within a collaboration that realizes a use case.
* Model the interaction between objects within a collaboration that realizes an operation.

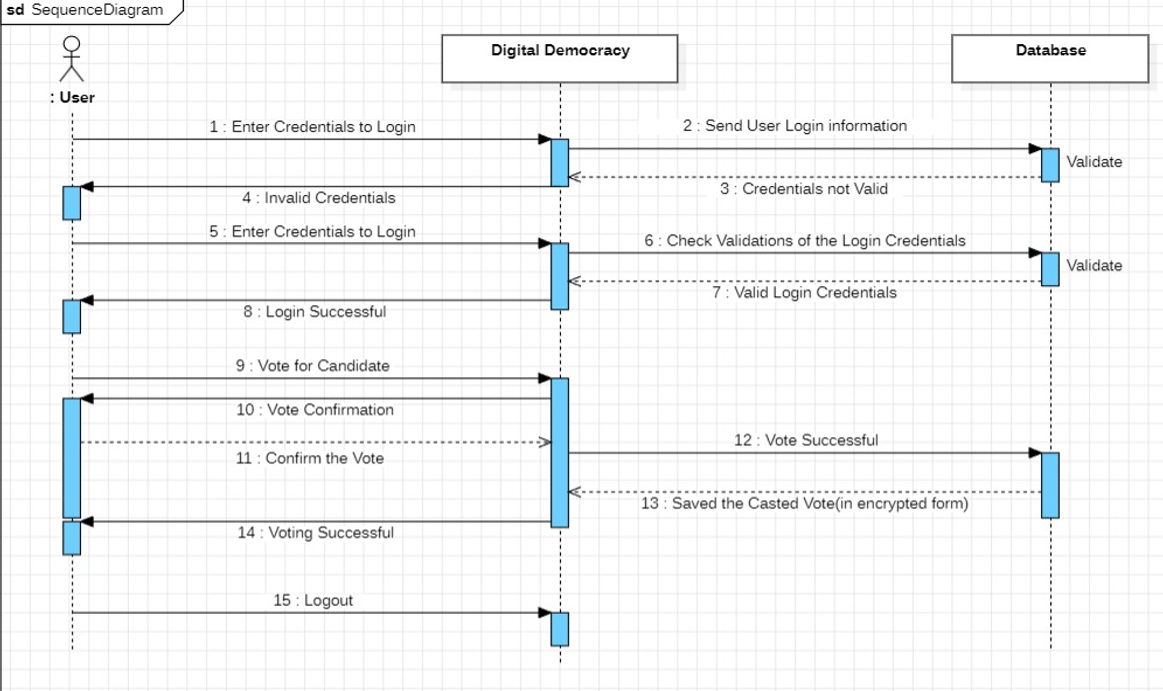


Fig 3.6 Sequence diagram for User

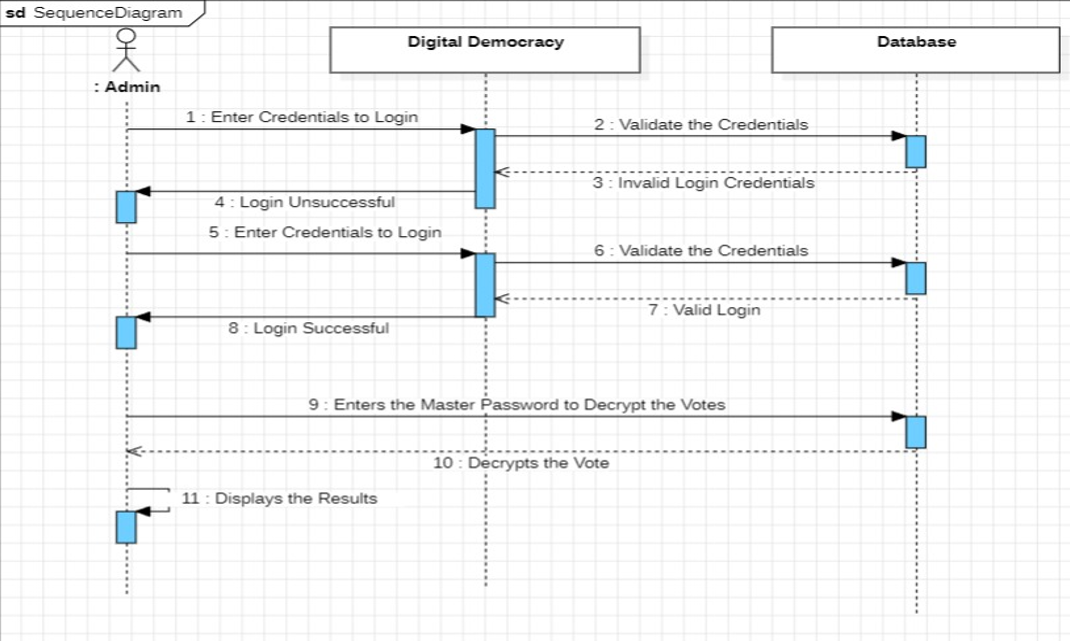


Fig 3.7 Sequence diagram for Admin

### Activity Diagram:

Activity Diagram describe how activities are coordinated to provide a service which can be at different levels of abstraction. Typically, an event needs to be achieved by some operations, particularly where the operation is intended to achieve several different things that require coordination, or how the events in a single use case relate to one another use cases where activities may overlap and require coordination. It is also suitable for modelling how a collection of use cases coordinate to represent business workflows.

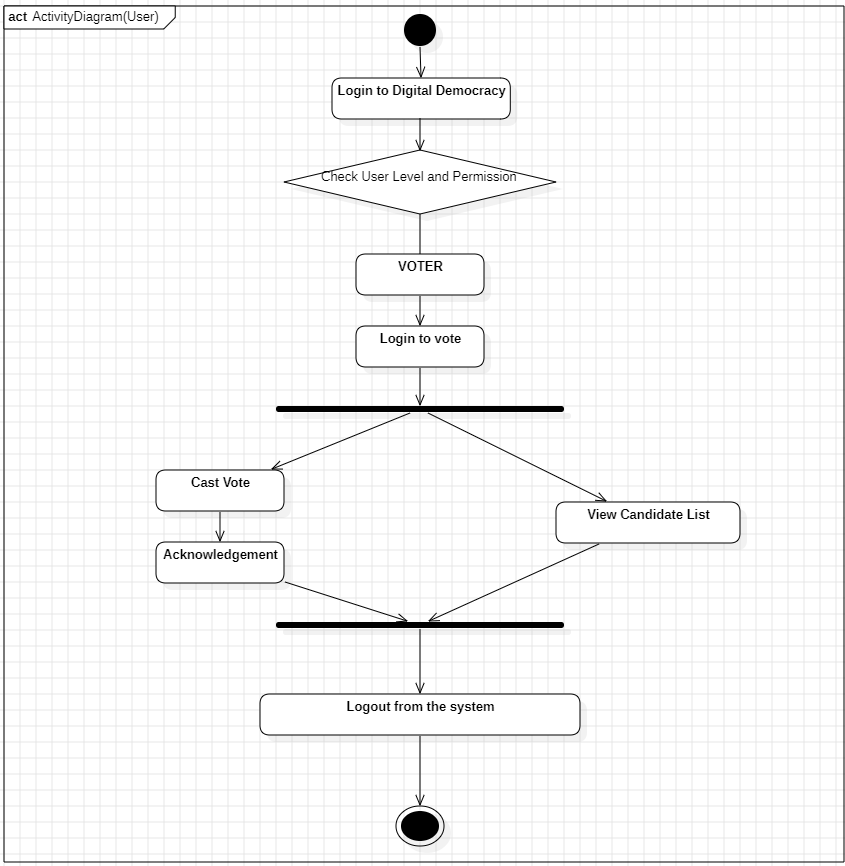


Fig 3.8 Activity diagram for User

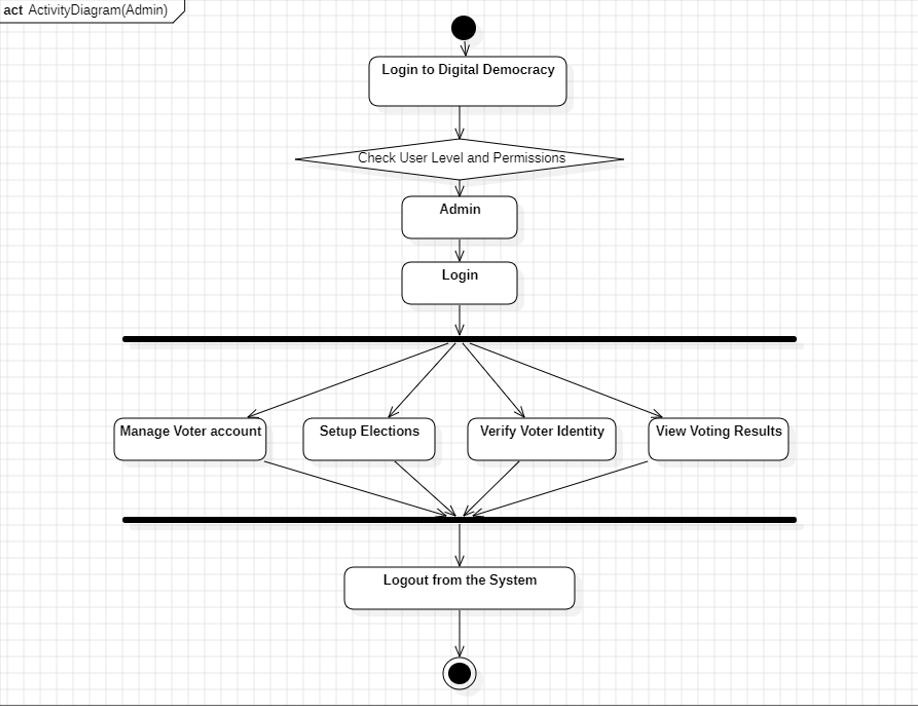


Fig 3.9 Activity diagram for Admin

### DataBase



Fig 3.10 User details

# IMPLEMENTATION

* 1. Technologies

## Implementation

This system is developed using Java programming language using APACHE Tomcat .

#### HTML

HTML (HyperText Markup Language) is the standard markup language used to create web pages. It forms the backbone of the web application, providing the structure and layout of the content displayed on the online voting platform. HTML is used to define the elements of the voting system, including the login forms, voting ballots, and user interface components.

#### CSS

CSS (Cascading Style Sheets) is used to control the presentation and layout of the web pages. It allows for the styling of HTML elements, making the online voting system visually appealing and easy to navigate. CSS is used to define the look and feel of the platform, ensuring a consistent and professional appearance across all pages.

#### JavaScript

JavaScript is a versatile scripting language that enables dynamic and interactive features on web pages. In the online voting system, JavaScript is used to enhance user interaction, validate input data, and manage client-side operations. It plays a crucial role in providing real-time feedback to users, such as confirming vote submissions and updating election results without requiring page reloads.

* + 1. **PHP**

PHP (Hypertext Preprocessor) is a server-side scripting language used to handle backend operations. It is responsible for processing user input, managing sessions, and interacting with the database. PHP scripts handle the authentication of users, record votes, and generate reports on election results. It ensures the secure transmission and storage of sensitive data, such as voter information and vote records.

* + 1. **MySQL**

MySQL is a robust and widely used relational database management system. It is used to store and manage all the data associated with the online voting system. This includes user registration details, voter credentials, candidate information, and vote counts. MySQL ensures that data is stored securely and can be retrieved efficiently, supporting the smooth operation of the voting process.

**Integration and Security:**

The integration of these technologies ensures a cohesive and functional online voting platform. The combination of HTML, CSS, JavaScript, PHP, and MySQL provides a robust foundation for developing a secure, reliable, and user-friendly system. Security measures such as encryption, secure authentication, and data validation are implemented to protect the integrity of the voting process and safeguard user information.

By leveraging these web technologies, the online voting system offers a modern and efficient solution for conducting elections, enhancing voter participation, and ensuring the accuracy and transparency of the electoral process.

### Pseudocode

### **Encryption:**

FUNCTION encryptVote(vote: STRING) RETURNS STRING

// Define a constant key for XOR encryption

CONSTANT key = "PraharshaSathvikaVaishnavi"

INITIALIZE encryptedVote TO EMPTY STRING

// Trim leading and trailing whitespace from the input vote

TRIM leading and trailing whitespace from vote

FOR each character i in vote

// Compute the encrypted character using XOR operation

encryptedChar = ASCII value of character at position i in vote XOR ASCII value of character at position (i MOD length of key) in key

// Append the encrypted character to the encryptedVote string

APPEND encryptedChar to encryptedVote AS CHARACTER

END FOR

// Encode the encryptedVote string in Base64

Base64Encode encryptedVote TO base64EncryptedVote

RETURN base64EncryptedVote

END FUNCTION

**Decryption:**

FUNCTION decryptVote(encryptedVote: STRING) RETURNS STRING

// Define a constant key for XOR decryption

CONSTANT key = "PraharshaSathvikaVaishnavi"

INITIALIZE decryptedVote TO EMPTY STRING

// Decode the input Base64 encoded string

Base64Decode encryptedVote TO decodedEncryptedVote

// Loop through each character in the decoded encrypted vote

FOR each character i in decodedEncryptedVote

// Compute the decrypted character using XOR operation

decryptedChar = ASCII value of character at position i in decodedEncryptedVote XOR ASCII value of character at position (i MOD length of key) in key

// Append the decrypted character to the decryptedVote string

APPEND decryptedChar to decryptedVote AS CHARACTER

END FOR

TRIM leading and trailing whitespace from decryptedVote

RETURN decryptedVote

END FUNCTION

# TESTING

### Testing

* 1. Overview

Software testing is a process, to evaluate the functionality of a software application with an intent to find whether the developed software met the specified requirements or not and to identify the defects to ensure that the product is defect free in order to produce the quality product.

As per the current trend, due to constant change and development in digitization, our lives are improving in all areas. The way we work is also changed. We access our bank online, we do shop online; we order food online and many more. We rely on software’s and systems. What if these systems turnout to be defective? We all know that one small bug shows huge impact on business in terms of financial loss and goodwill. To deliver a quality product, we need to have Software Testing in the Software Development Process.

Some of the reasons why software testing becomes very significant and integral part in the field of information technology are as follows.

1. Cost effectiveness
2. Customer Satisfaction
3. Security
4. Product Quality

### Dimensions of Testing

There are many different dimensions to consider:

1. Layers of the application (DataBase, APIs, UI)
2. Scale of testing (unit, module, integration, scenario)
3. Type of testing (functional, performance, security, etc.)
4. Methodology (exploratory, scripted manual, automated)

### Stages of Testing

#### Unit Testing

During This first round of testing, the program is submitted to assessments that focus on specific units or components of the software to determine whether each one is fully functional. In this phase, a unit can refer to a function, individual program or even a procedure, and White box testing method is usually used to get the job done. One of the biggest benefits of this testing phase is that it can be run every time a piece of code is changed, allowing issues to be resolved as quickly as possible. It quite common for software developers to perform unit tests before delivering software to testers for formal testing.

#### Integration Testing

Integration testing allows individuals the opportunity to combine all of the units within a program and test them as a group. This testing level is designed to find interface defects between the modules/functions. This is particularly beneficial because it determines how efficiently the units are running together. Keep in mind that no matter how efficiently each unit is running, if they properly integrated, it will affect the functionality of the software program. In order to run these types of tests, individuals can make use of various testing methods, but the specific method that will be used to get the job done will depend greatly on the way in which the units are defined.

#### System Testing

System testing is the first level in which the complete application is tested as a whole. The goal at this level is to evaluate whether the system has complied with all of the outlined requirements and to see that it meets Quality Standards. System testing is undertaken by independent testers who haven’t played a role in developing the program. This testing is performed in an environment that closely mirrors production. System Testing is very important because it verifies that the application meets the technical, functional, and business requirements that were set by the customer.

#### Acceptance Testing

The final level, Acceptance testing (or User Acceptance Testing), is conducted to determine whether the system is ready for release. During the Software development life cycle, requirements changes can sometimes be misinterpreted in a fashion that does not

meet the intended needs of the users. During this final phase, the user will test the system to find out whether the application meets their business needs. Once this process has been completed and the software has passed, the program will then be delivered to production. The extensiveness of these tests is just another reason why bringing software testers in early is important. When a program is more thoroughly tested, a greater number of bugs will be detected; this ultimately results in higher quality software.

### Types of testing

#### Black box testing

It is also called as Behavioral/Specification-Based/Input-Output Testing. Black Box Testing is a software testing method in which testers evaluate the functionality of the software under test without looking at the internal code structure. This can be applied to every level of software testing such as Unit, Integration, System and Acceptance Testing.

#### White box testing

It is also called as Glass Box, Clear Box, Structural Testing. White Box Testing is based on applications internal code structure. In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. This testing usually done at the unit level.

White Box Testing Techniques:

1. Statement Coverage
2. Branch Coverage
3. Path Coverage

### Test Cases

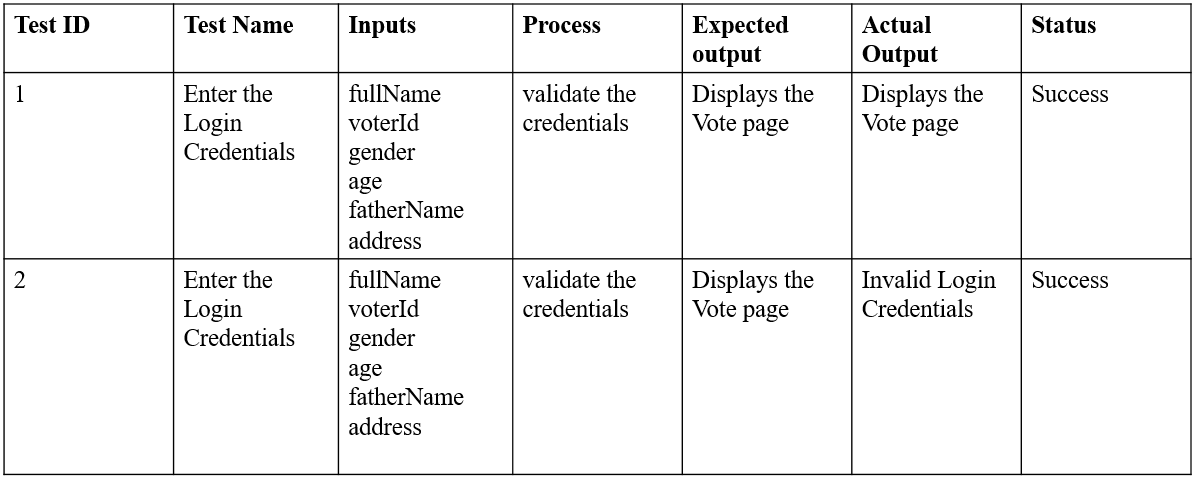


Table 5.1 User

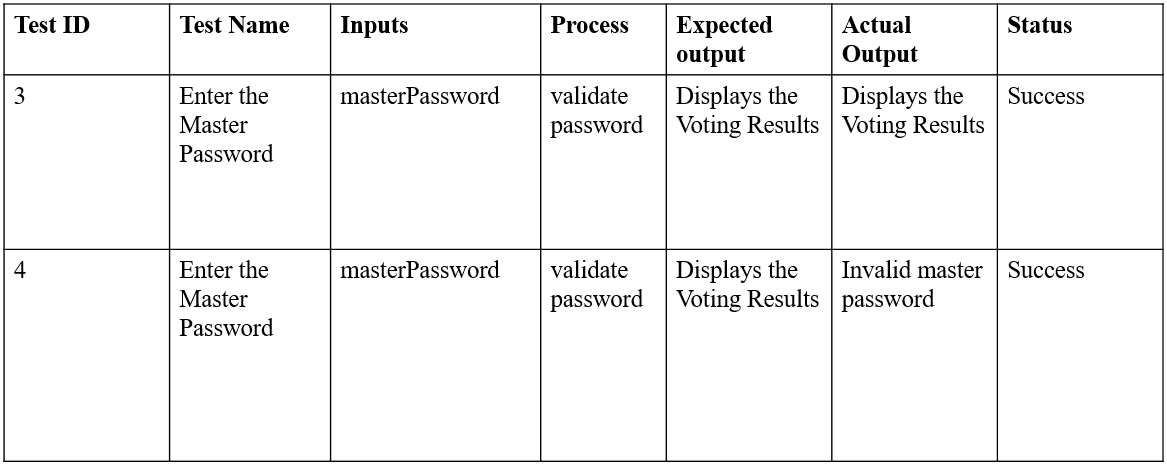


Table 5.2 Admin

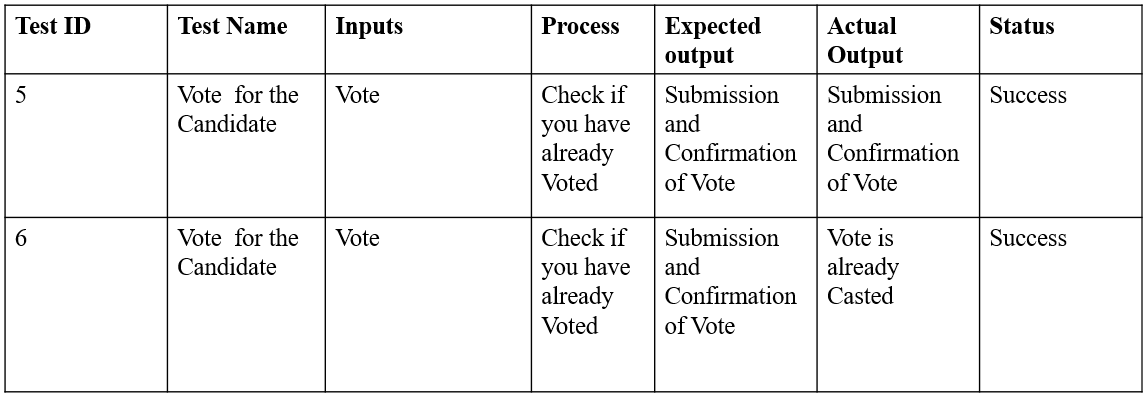
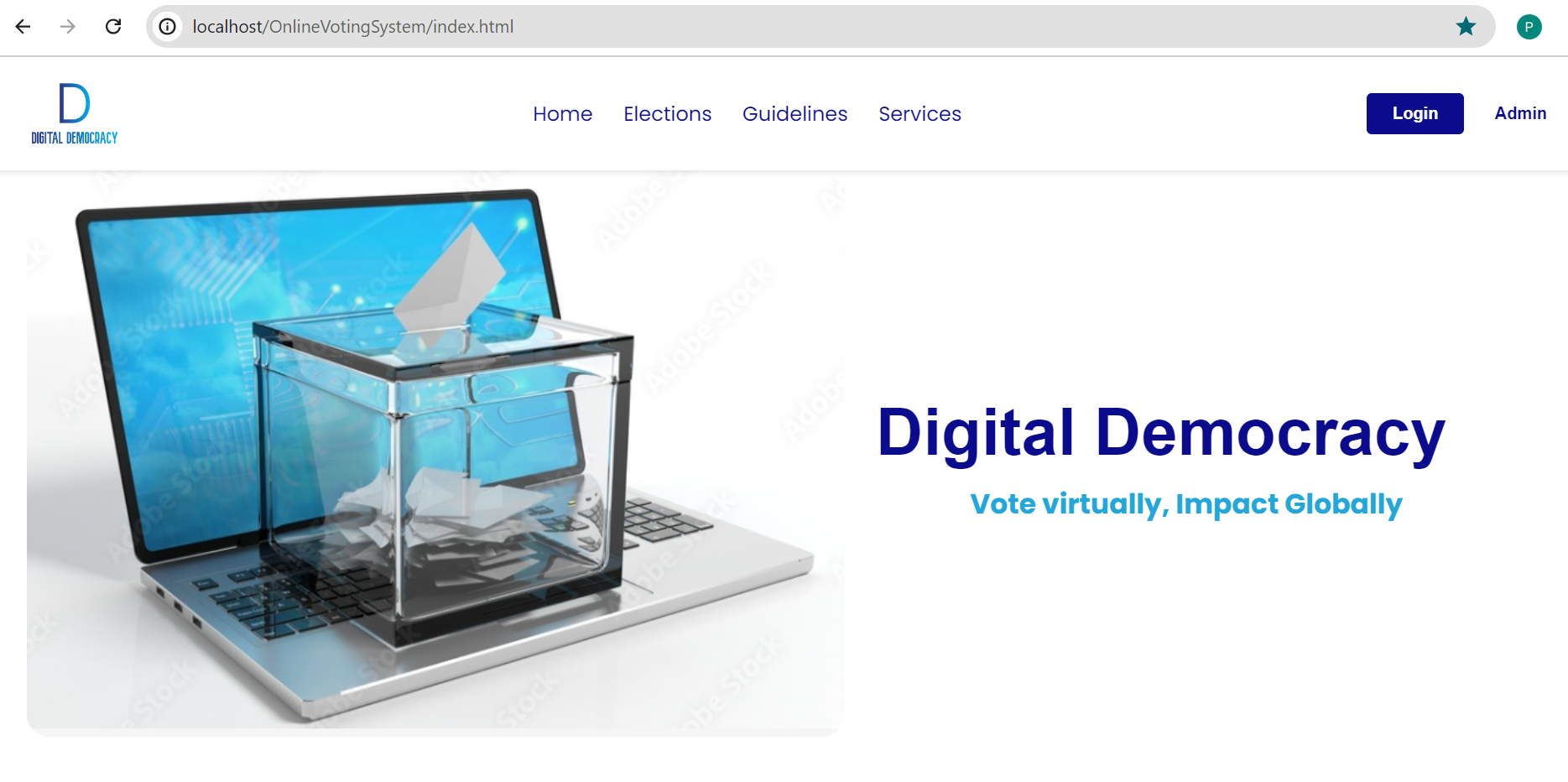


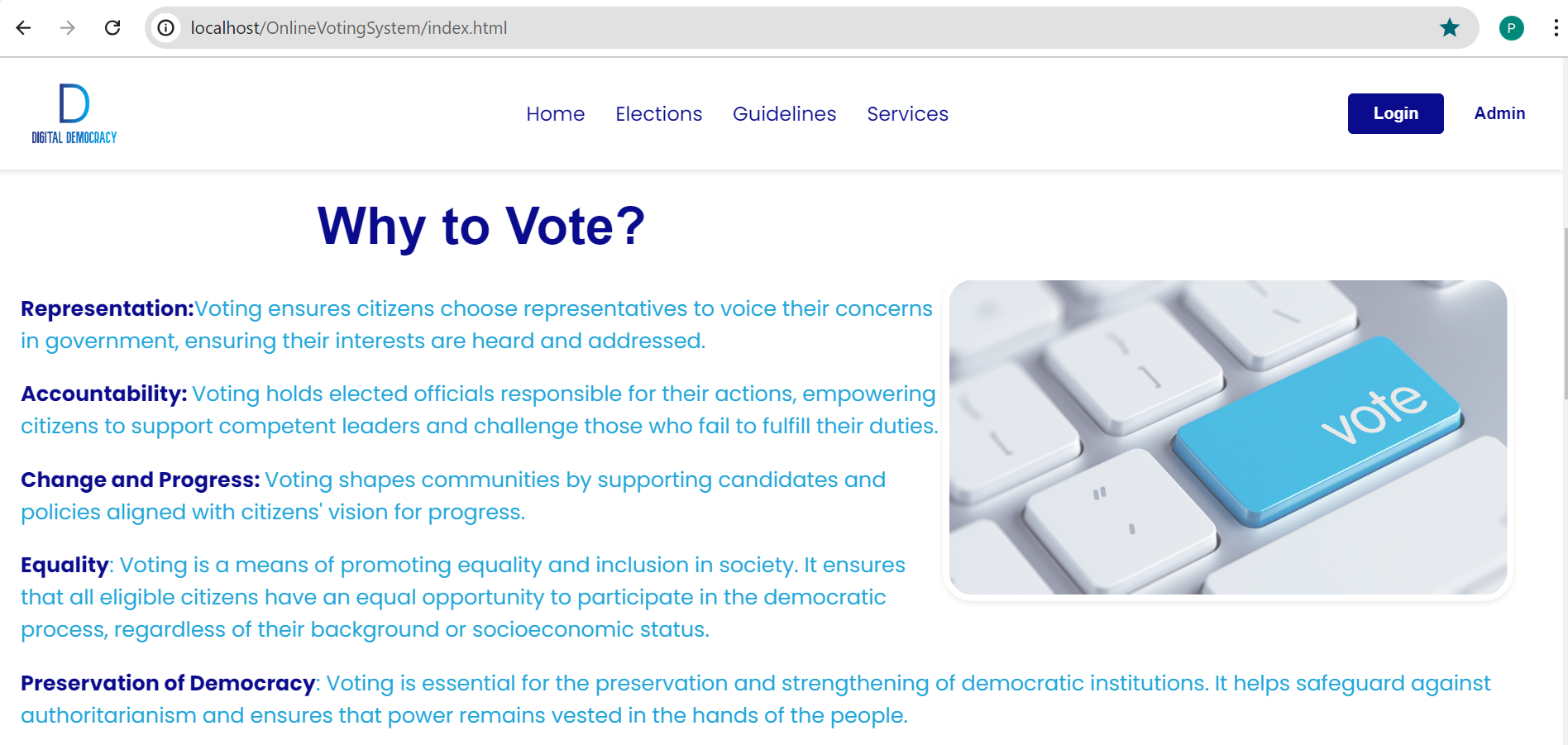
Table 5.3 Vote

# SCREENSHOTS

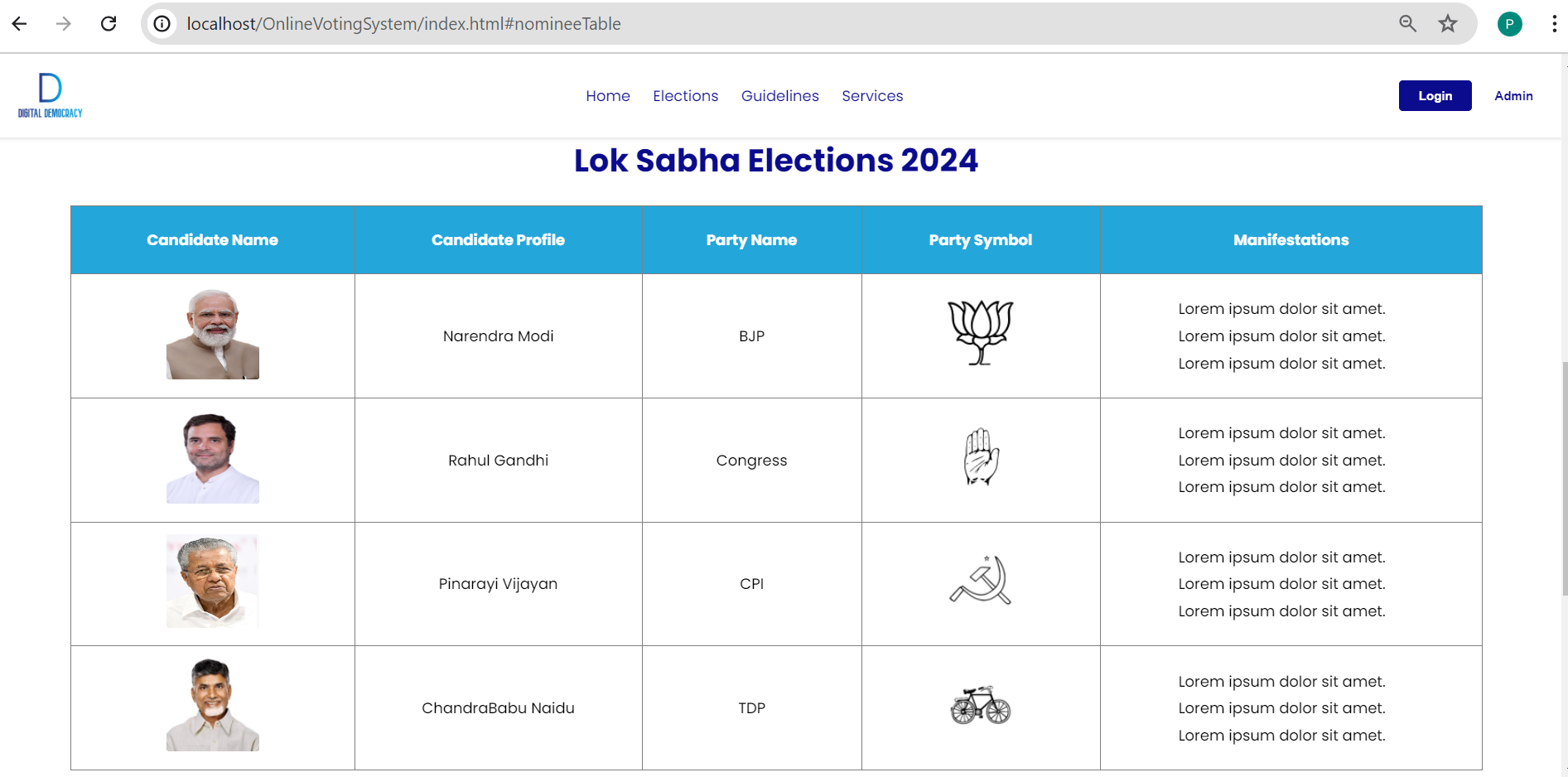
## 6.Screenshot



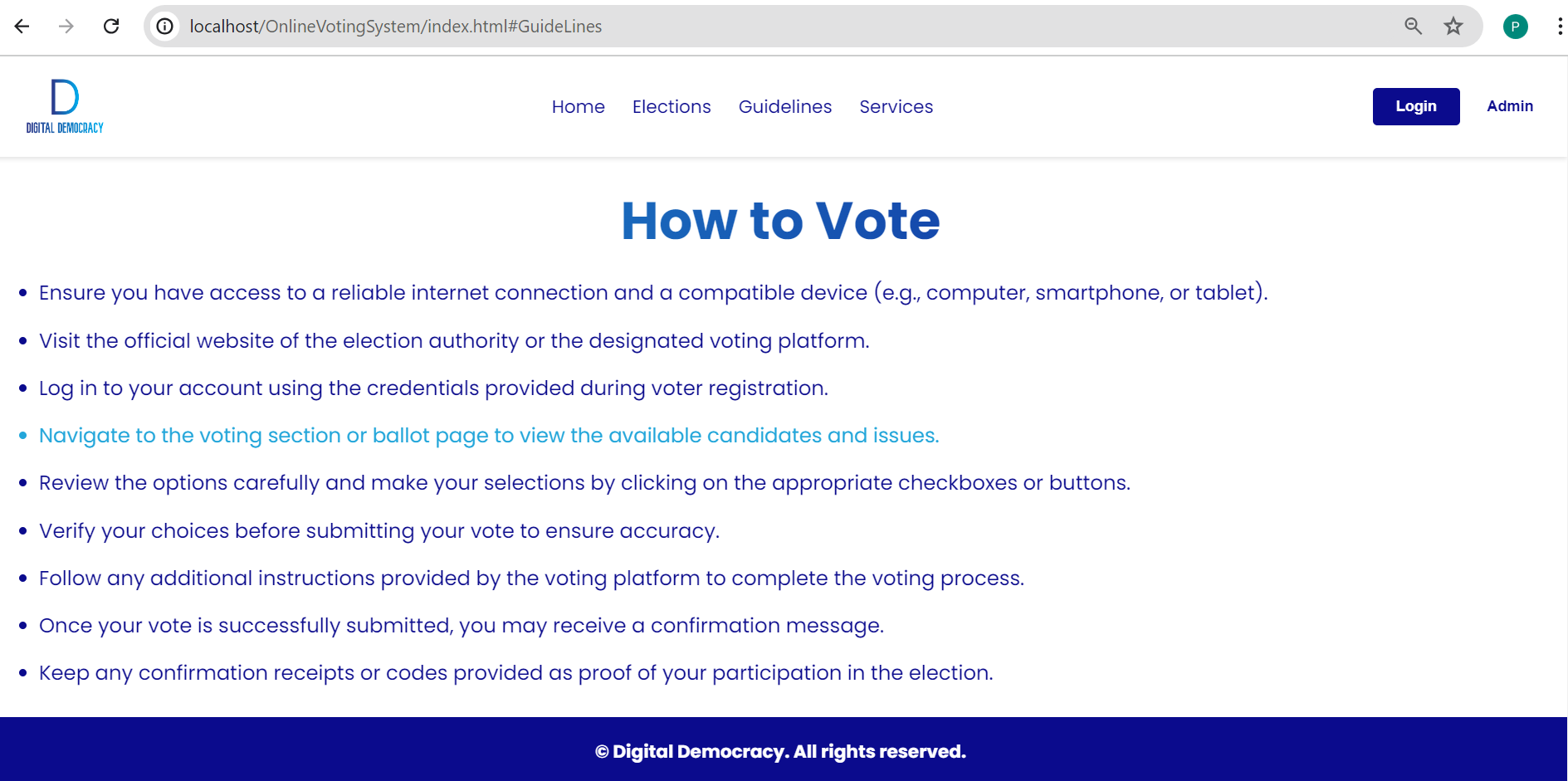
Screenshot 6.1 Home Page



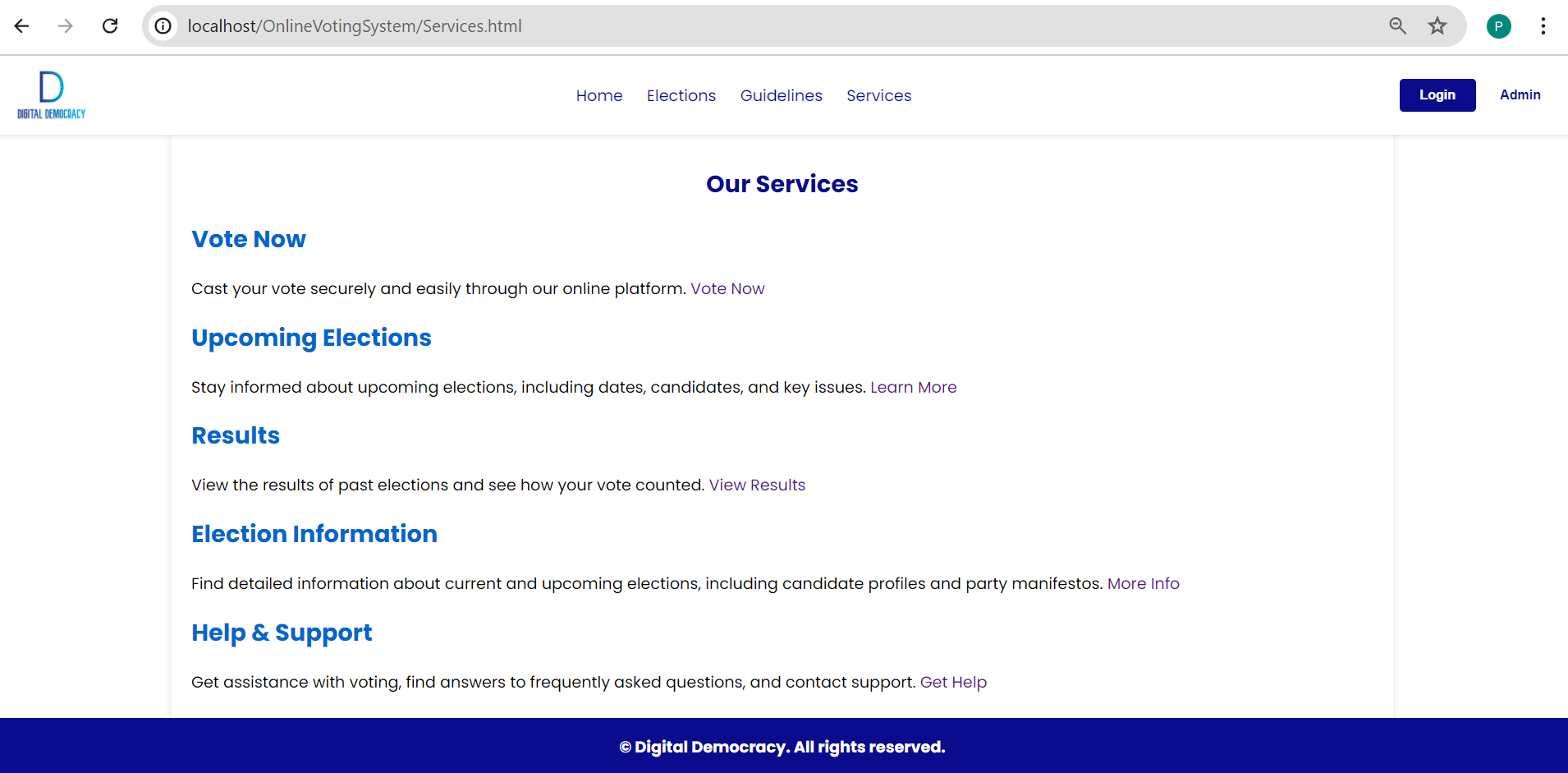
Screenshot 6.2 Home Page



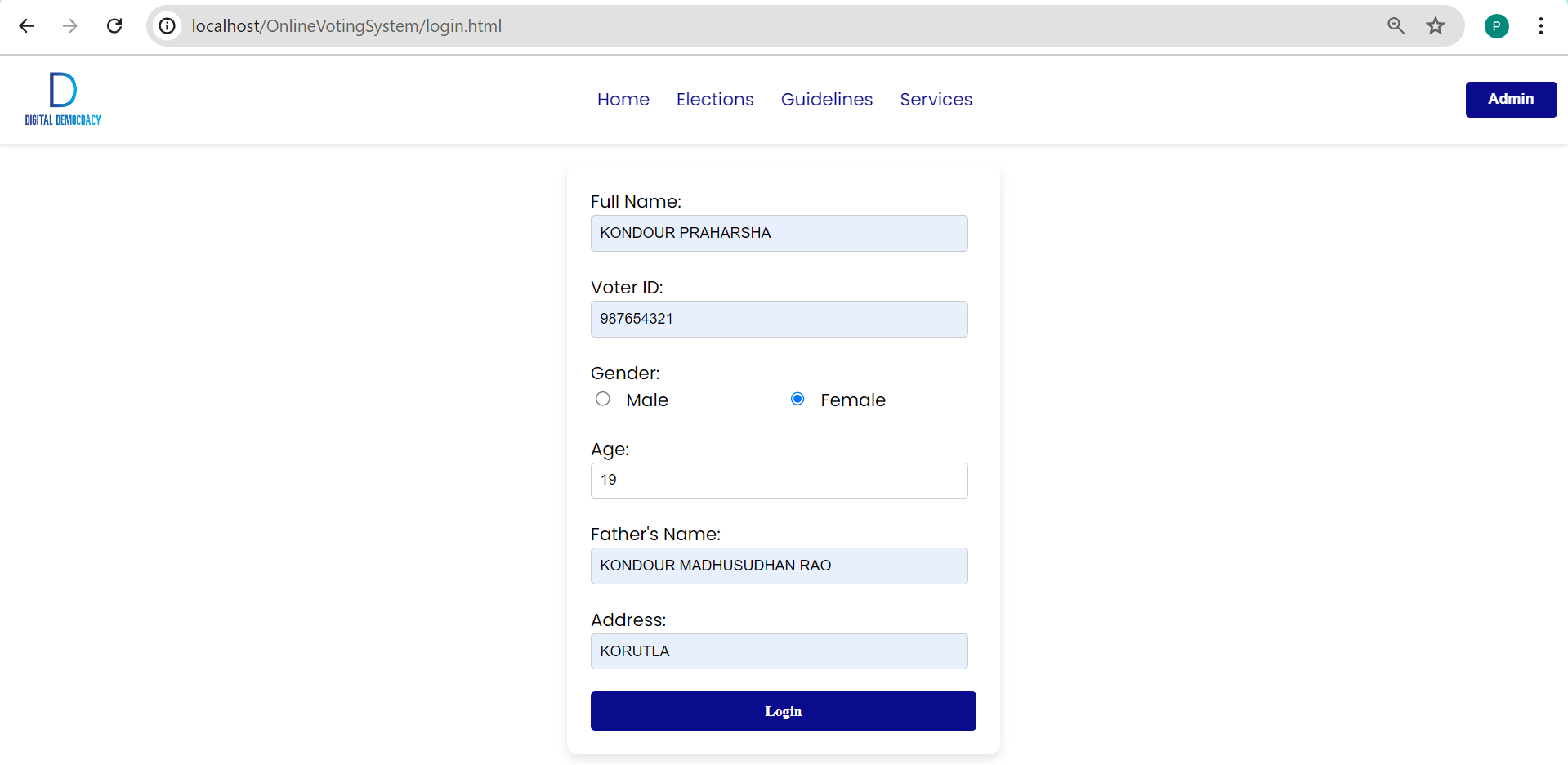
Screenshot 6.3 Election Page



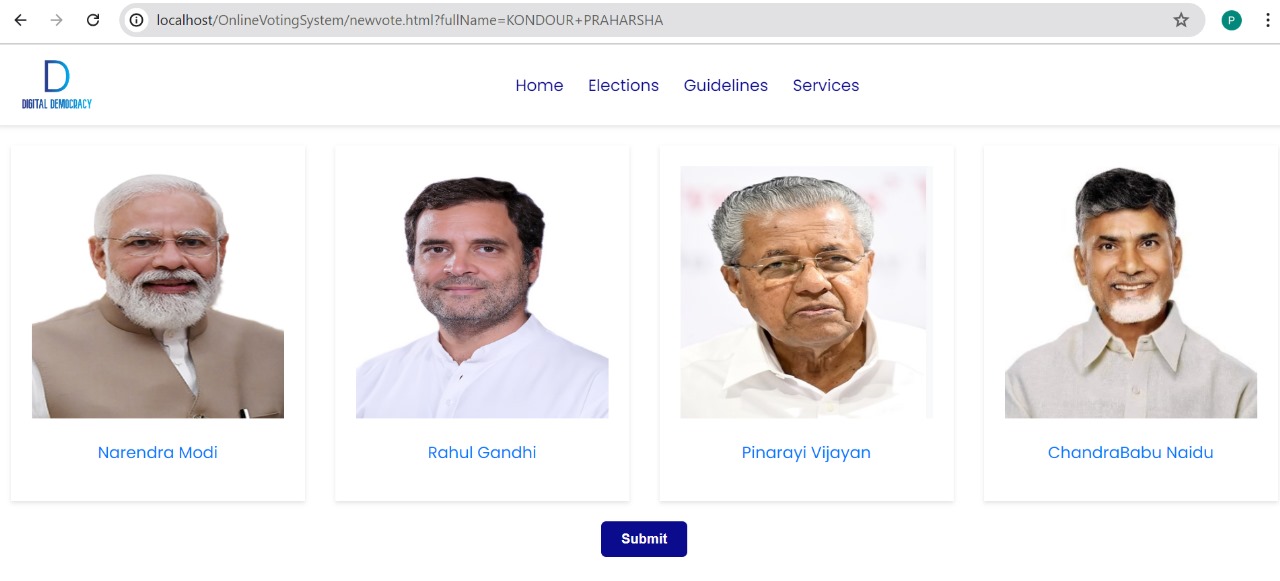
Screenshot 6.4 Guidelines Page



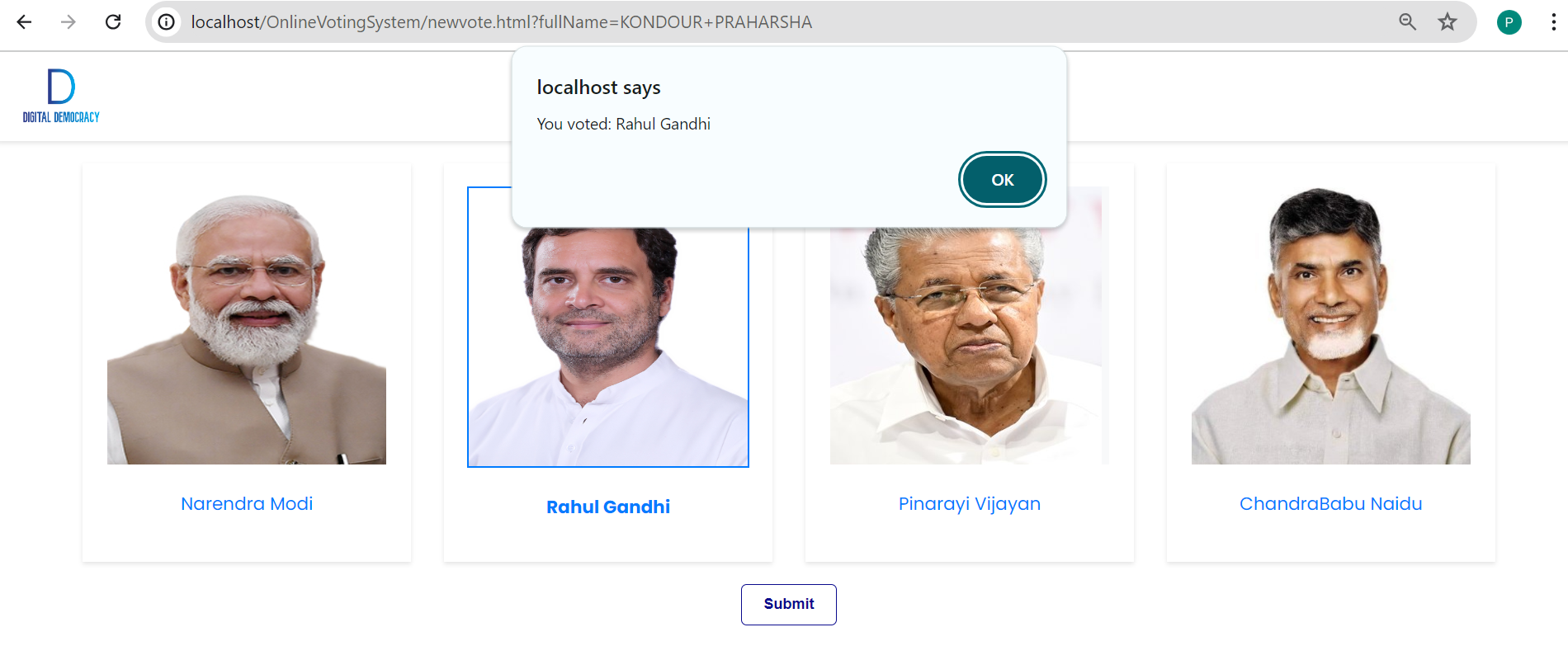
Screenshot 6.5 Services Page



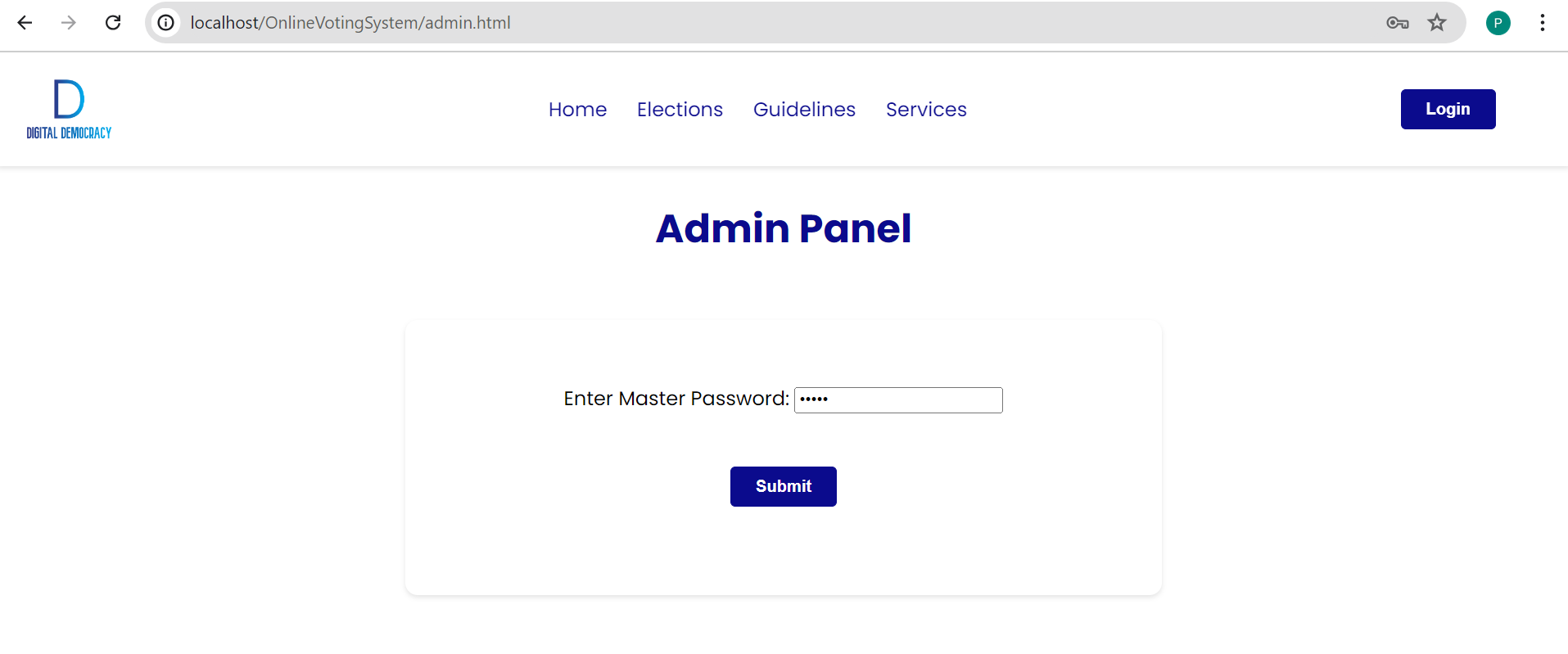
Screenshot 6.6 Login Page



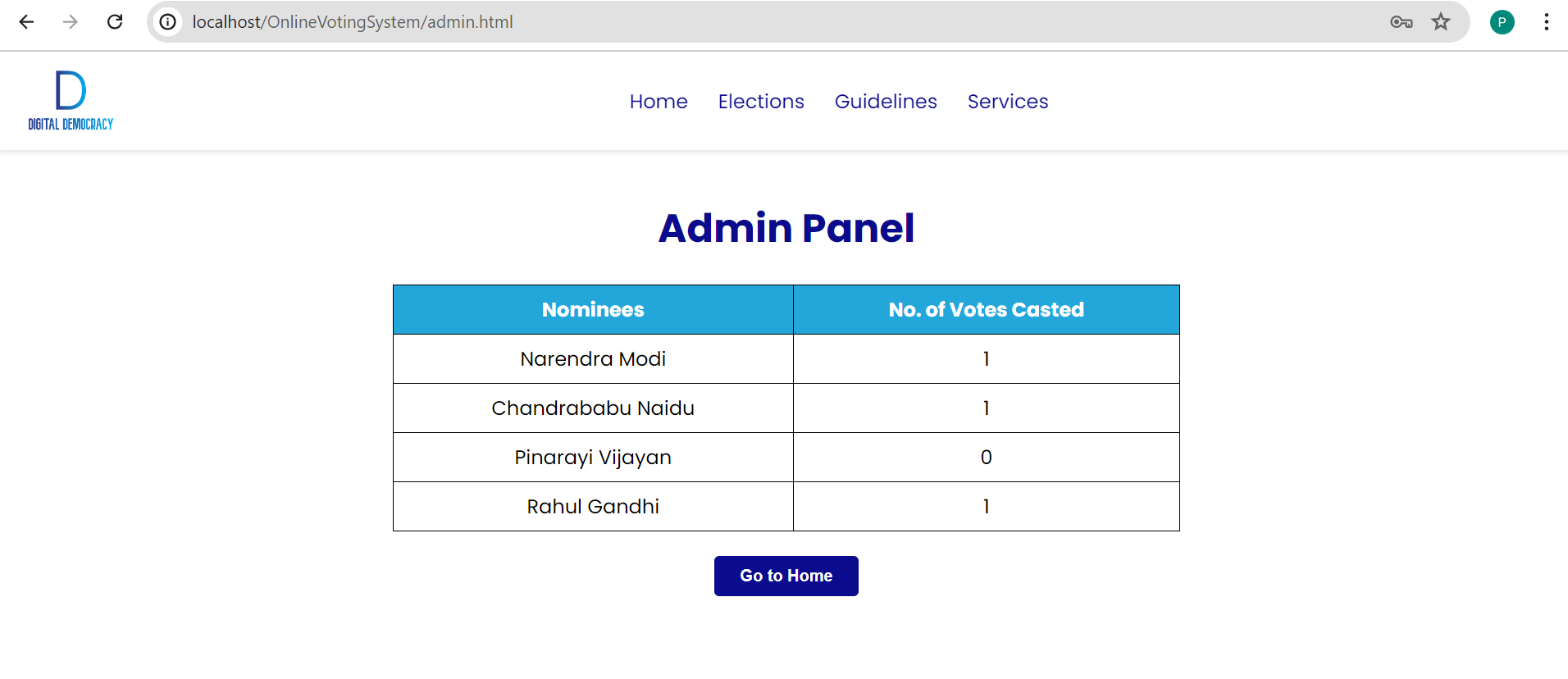
Screenshot 6.7 Vote Page



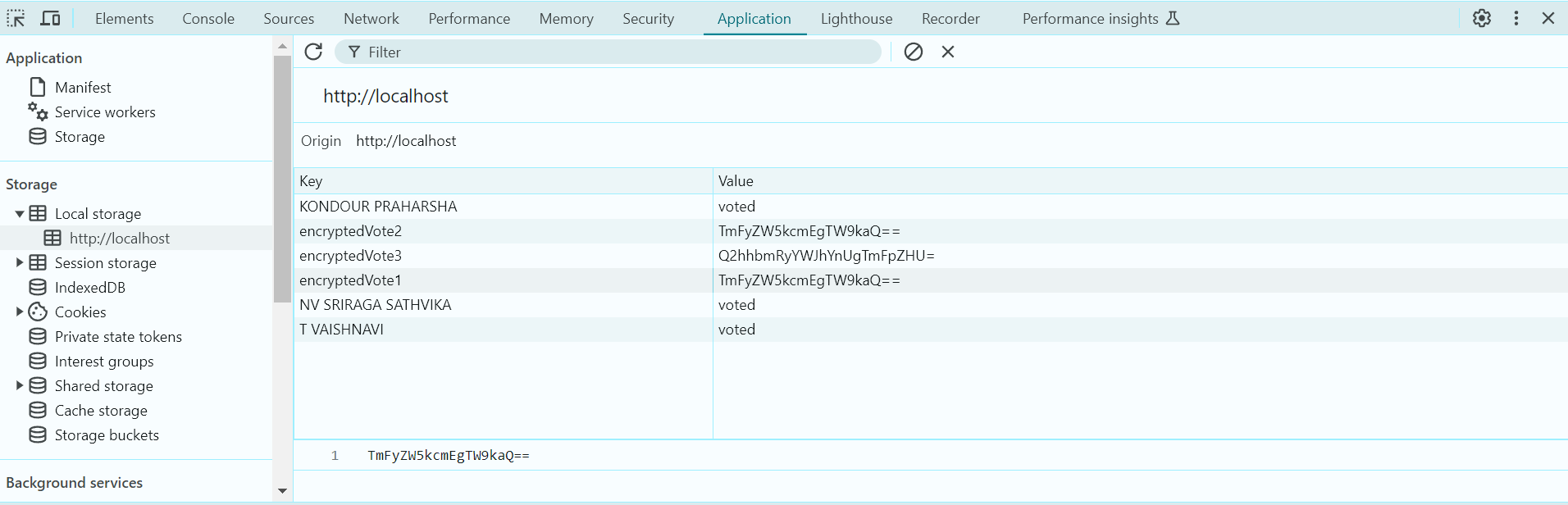
Screenshot 6.8 Vote Page



Screenshot 6.9 Admin Panel



Screenshot 6.10 Admin Panel



Screenshot 6.11 Locally storing encrypted votes

# CONCLUSION & FUTURE SCOPE

7.1 Conclusion

## 7 Conclusion & Future Scope

### The implementation of the Online Voting System represents a significant step forward in modernizing the electoral process. By ensuring secure, convenient, and efficient voting, this system enhances voter participation and trust in democratic practices. Through the use of advanced technology, it fosters greater civic engagement and upholds the integrity of elections.

### Future Scope

**Authentication**: Integration of emerging biometric authentication methods like facial recognition for enhanced security.

**Advanced Encryption Techniques:** Research and implementation of quantum-resistant encryption algorithms to withstand future computing capabilities.

# REFERENCES

### 8. References

**Websites:**

1. <https://www.eballot.com/>

2. https://innovatrics.com/trustreport/six-must-haves-for-a-trusted-election-via-smartphones/

3. <https://www.eci.gov.in/evm/>

**Books:**

1. Web Technologies, Uttam K Roy, Oxford University Press.

2. Eloquent JavaScript: A Modern Introduction to Programming.

3. Software Engineering: Sommerville, 7th edition, Pearson Education.