

# **DIGITAL VOTING**

A Dissertation submitted  
for the partial fulfillment of the degree of  
**Bachelor of Engineering in**  
**Information Technology**  
**(Session 2023 -2024 )**

**Guided By:**  
**Ms Vedpriya Dongre**

**Submitted By:**  
**Sarvada Goantiya(20I7054)**  
**Surbhi Sharma (20I7062)**  
**Viral Maru(20I7067)**

**Department of Department name**  
**Institute of Engineering & Technology**  
**Devi Ahilya Vishwavidyalaya, Indore (M.P.)**  
**([www.iet.dauniv.ac.in](http://www.iet.dauniv.ac.in))**  
**December 2023**

## **Dissertation Approval Sheet**

The dissertation entitled “**DIGITAL VOTING**” submitted by **Sarvada Goantiya, Surbhi Sharma and Viral Maru** is approved as partial fulfillment for the award of **Bachelor of Engineering in Information Technology** degree by **Devi Ahilya Vishwavidyalaya, Indore.**

**Internal Examiner**

**External Examiner**

**Director**  
**Institute of Engineering & Technology**  
**Devi Ahilya Vishwavidyalaya,**  
**Indore (M.P.)**  
[i]

## **Recommendation**

The dissertation entitled “**DIGITAL VOTING**” submitted by **Sarvada Goantiya, Surbhi Sharma and Viral Maru** is a satisfactory account of the bonafide work done under my supervision is recommended towards the partial fulfillment for the award of **Bachelor of Engineering in Information Technology** degree by **Devi Ahilya Vishwavidyalaya, Indore.**

**Date: 30-10-2023**

**Mrs Vedpriya Dongre**  
Project Guide

**Endorsed By :**

Head, Department of Information Technology

## **Candidate Declaration**

We hereby declare that the work which is being presented in this project entitled **“DIGITAL VOTING”** in partial fulfillment of degree of Bachelor of Engineering in **Information Technology** is an authentic record of our own work carried out under the supervision and guidance of **Mrs Vedpriya Dongre, Lecturer** in Department of **Information Technology**, Institute of Engineering and Technology, Devi Ahilya Vishwavidyalaya, Indore

We are fully responsible for the matter embodied in this project in case of any discrepancy found in the project and the project has not been submitted for the award of any other degree.

**Date:30-10-2023**

**Place:**

**Sarvada Goantiya**

**Surbhi Sharma**

**Viral Maru**

## **ACKNOWLEDGEMENTS**

The satisfaction and euphoria that accompany the successful completion of any task would be impossible without the mention of the people who made it possible, whose constant guidance and encouragement crowned our efforts with success.

We express our gratitude to **Mrs. Vedpriya Dongre**, our project guide, for constantly monitoring the development of the project and setting up precise deadlines. Her valuable suggestions were the motivating factors in completing the work. She has taught us the methodology to carry out the project and to present the project works as clearly as possible. It was a great privilege and honor to work and study under his guidance. We are extremely grateful for what he has offered us. We express our warm and sincere thanks for the encouragement, untiring guidance, and confidence she had shown in us.

We have great pleasure in expressing our deep sense of gratitude to **Dr. Sanjeev Tokekar, Director of the Institute of Engineering and Technology** for providing the necessary infrastructure and creating a good environment.

We would also like to thank **Dr. Vrinda Tokekar, Head of the Department of Information Technology Center**, for her constant guidance.

Finally, a note of thanks to the teaching and non-teaching staff of the Department of Information Technology, for their cooperation extended to us, who helped us directly or indirectly in the course of the project work. We would also like to thank our parents, who have always been supportive and have always believed in us.

## **ABSTRACT**

This report addresses the development of an Online Voting System, focusing on critical aspects of accessibility, security, data management, usability, and transparency in the electoral process. The primary objective was to create a platform that enables citizens to vote remotely while upholding the integrity of elections.

The project involved an in-depth analysis of existing solutions, identifying key issues and gaps in the literature. Subsequently, we developed a robust online voting system that improves accessibility by extending voting privileges to a wider population. The implementation includes stringent security measures to prevent fraud and unauthorized access.

Efficient data management capabilities were integrated to securely store and manage voter data, candidate information, and election results. A user-friendly interface ensures an intuitive voting experience for all users, promoting ease of use and engagement.

Transparency in elections is ensured through mechanisms for monitoring and auditing votes, ultimately enhancing trust in the electoral process. Despite the achievements, it's recognized that implementing two-factor authentication, such as one-time passwords, could further enhance security.

Looking ahead, future expansions may include the implementation of advanced security features like face recognition, providing an extra layer of protection. The project, overall, aims to improve online voting accessibility, security, and user experience while addressing inherent limitations.

In conclusion, the Online Voting System project has made significant strides in creating a secure and accessible platform for citizens to participate in the democratic process. It acknowledges its limitations and seeks opportunities for continued improvement, ensuring that online voting remains a reliable and convenient option for all eligible voters.

	<b>Page No</b>
<b>Dissertation Approval Sheet</b>	i
<b>Recommendation</b>	ii
<b>Candidate Declaration</b>	iii
<b>Acknowledgements</b>	iv
<b>Abstract</b>	v
<b>List of Figures</b>	vi
<b>Chapter 1 Introduction</b>	<b>01</b>
1.1 Overview and issues involved	01
1.2 Problem Definition	02
1.3 Proposed Solution	03
<b>Chapter 2 Literature Survey</b>	<b>05</b>
2.1 Methodology	05
2.2 Conceptual Requirements	05
2.3 Existing Solutions	06
<b>Chapter 3 Analysis &amp; Design</b>	<b>08</b>
3.1 Software Requirements	08
3.2 Hardware Requirements	09
3.3 Analysis Diagrams	12
3.3.1 Use case Model	12
3.3.2 Use case Description	13
3.3.2.1 Admin Login	13
3.3.2.2 Manage Voter Details	13
3.3.2.3 Manage Candidate Details	14
3.3.2.4 Generate an Election	15
3.3.2.5 Voter login	16
3.3.2.6 Cast a Vote	17

3.3.2.7 Cast a Complaint	18
3.3.2.8 Filter Election Results	20
3.4 Design Diagrams	21
3.4.1 Sequence Diagram	22
<b>Chapter 4 Implementation and Testing</b>	<b>23</b>
4.1 Database Design	23
4.2 Class diagram	27
4.3 Test Cases	30
4.4 Illustration	34
<b>Chapter 5 Conclusion</b>	<b>42</b>
<b>References</b>	<b>44</b>
<b>Appendix</b>	<b>44</b>

## LIST OF FIGURES

S No.	Figures	Content	Page No.
1	<b>Fig: 3.1</b>	Use Case Diagram	12
2	<b>Fig: 3.2</b>	Sequence Diagram	22
3	<b>Fig: 4.1</b>	ER-Diagram	23
4	<b>Fig: 4.2</b>	Database Snaps(All 18 tables)	24
5	<b>Fig: 4.3</b>	Database Snaps(Elections)	24
6	<b>Fig: 4.4</b>	Database Snaps(votes)	25
7	<b>Fig: 4.5</b>	Database Snaps(Admins)	25
8	<b>Fig: 4.6</b>	Database Snaps(voters)	26
9	<b>Fig: 4.7</b>	Database Snaps(candidates)	26
10	<b>Fig: 4.8</b>	Class Diagram	27
11	<b>Fig: 4.9</b>	Test case 1	30
12	<b>Fig: 4.10</b>	Test case 2	31
13	<b>Fig: 4.11</b>	Test case 3	31
14	<b>Fig: 4.12</b>	Admin Login	34
15	<b>Fig: 4.13</b>	Admin Dashboard	34
16	<b>Fig: 4.14</b>	Add Candidate	35
17	<b>Fig: 4.15</b>	View Voter	35
18	<b>Fig: 4.16</b>	Generate Election	36
19	<b>Fig: 4.17</b>	View Complaint	36
20	<b>Fig: 4.18</b>	Voter Login	37
21	<b>Fig: 4.19</b>	Voter Dashboard	37
22	<b>Fig: 4.20</b>	Guidelines to Vote	38
23	<b>Fig: 4.21</b>	Cast Vote	38
24	<b>Fig: 4.22</b>	Vote submitted	39

25	<b>Fig: 4.23</b>	Add EVM Votes	39
26	<b>Fig: 4.24</b>	Generate Result	40



# **Chapter-1**

## **Introduction**

### **1.1 Overview and issues involved**

The Online Voting System project emerges in response to the growing need for a modernized and secure method of conducting elections. Traditional voting systems face numerous challenges, including limited accessibility, security concerns, logistical complexities, and the need for streamlined data management. These issues are further amplified in the context of changing demographics and the rise of digital technology.

The primary goal of this project is to develop a cutting-edge online voting system that effectively addresses these challenges. The proposed solution focuses on delivering a user-friendly and accessible platform while ensuring the highest standards of security and transparency.

This project tackles key voting issues: accessibility, security, data management, usability, and transparency. It strives to make voting more accessible, particularly for individuals with mobility challenges or those distant from polling stations. Rigorous security measures will protect the integrity of the process. Efficient data management capabilities ensure the secure handling of voter and candidate data. A user-friendly interface makes voting simple for all. Lastly, the project emphasizes transparency through real-time monitoring and auditing to build trust in the electoral system. These initiatives aim to enhance the voting experience, making it more inclusive, secure, and user-friendly while promoting transparency in the democratic process.

By addressing these issues, the Online Voting System project seeks to revolutionize the way elections are conducted, setting new standards for accessibility, security, and user experience. In doing so, it aims to contribute to the advancement of democratic processes and enhance the trust and participation of citizens in the electoral system.

## 1.2 Problem definition

The online voting system project addresses several pressing challenges in traditional voting methods and seeks to introduce a more efficient and inclusive approach to the electoral process. These issues can be grouped into five key problem areas:

1. Accessibility: Traditional voting methods typically require physical presence at designated polling stations. This poses a significant barrier to various groups within the population. Individuals with mobility challenges, the elderly, those living in remote areas, and overseas citizens often face difficulties in exercising their voting rights. This lack of accessibility results in lower voter turnout and can exclude significant portions of the population from participating in the democratic process.
2. Security: The security and integrity of the voting process are paramount. Traditional methods are susceptible to various forms of fraud, including impersonation, ballot tampering, and voter coercion. Ensuring the security of votes and preventing unauthorized access to the system is of utmost importance. Without robust security measures, the credibility of the entire electoral process can be undermined.
3. Data Management: Traditional paper-based voting systems are cumbersome and error-prone in managing vast amounts of voter data, candidate information, and election results. This inefficiency can lead to data inaccuracies and delays in the announcement of results. Efficient data management is crucial for maintaining the credibility and transparency of the election process.
4. Usability: Many traditional voting methods lack a user-friendly interface. This can make the voting process confusing and intimidating for some voters. A system that is difficult to navigate can deter individuals from participating in elections, especially those who are not tech-savvy. A user-friendly and intuitive interface is essential to make the voting process accessible to all citizens.
5. Transparency: Public trust in the electoral process is crucial for a healthy democracy. Traditional methods often lack real-time monitoring and auditing

capabilities. This absence of transparency can lead to suspicions of fraud and electoral misconduct. Enhancing transparency in the electoral process is essential to build public trust and ensure that votes are counted accurately.

In summary, the project aims to address these challenges by introducing an online voting system that enhances accessibility, security, data management, usability, and transparency. By doing so, it seeks to create a more inclusive, secure, and user-friendly democratic process that fosters trust and ensures that every eligible voter can participate in the election with confidence.

### **1.3 Proposed solution**

The proposed solution is an innovative online voting system designed to revolutionize the way elections are conducted. It offers a range of functionalities that address the key issues discussed earlier while surpassing existing solutions in the market. This solution aims to enhance accessibility, security, data management, usability, and transparency in the electoral process.

The system's functionalities include remote voting, robust security measures, efficient data management, a user-friendly interface, and real-time transparency. Remote voting allows eligible voters to cast their votes from anywhere, significantly improving accessibility. Security features such as encryption, biometric authentication, and audit trails ensure the integrity of votes and protect against fraud and unauthorized access.

Efficient data management streamlines voter registration, candidate information, and vote tallying, overcoming the data management challenges of traditional paper-based systems. The user-friendly interface caters to voters of all ages and technical backgrounds, making the online voting process accessible and intuitive.

Real-time transparency is a critical aspect of the system, allowing election authorities, observers, and the public to track the voting process as it happens. This transparency enhances trust in the electoral process.

When compared to existing solutions, this online voting system offers several key advantages. It provides wider accessibility, allowing citizens with geographical or physical barriers to participate, surpassing traditional methods that require physical presence. Enhanced security features, including biometrics and encryption, make it significantly more secure than traditional paper ballots, minimizing the risks of fraud and tampering.

Efficient data management ensures timely and accurate election results, overcoming the limitations of manual vote counting and data entry. The user-centric design ensures that the online voting process is accessible and straightforward for all voters.

Real-time transparency sets it apart from traditional methods, enhancing public trust in the electoral process. In conclusion, the proposed online voting system offers a comprehensive solution to the issues faced by traditional voting methods and surpasses existing market solutions. This innovative approach aims to create a more accessible, secure, and user-centric democratic process, fostering public trust and participation in elections.

The online voting system provides a modern solution to several key issues. It enhances accessibility by enabling remote voting. Robust security measures safeguard the integrity of the voting process. Efficient data management ensures accurate results. User-friendliness enhances the voting experience for all. Real-time transparency fosters trust and accountability. Compared to traditional methods, it overcomes physical limitations, bolsters security, streamlines data handling, prioritizes usability, and elevates transparency. These advantages make it a substantial improvement over existing solutions in the market, ensuring a more inclusive, secure, and convenient electoral system.

# **Chapter-2**

## **Literature Survey**

### **2.1 Methodology**

**Requirement Analysis:** The project initiates with a thorough requirement analysis. This phase entails defining both the functional and non-functional requirements of the online voting system. These requirements encompass accessibility, security, data management, usability, and transparency. Understanding these requirements is crucial for shaping the project's direction.

**System Design:** Following the requirement analysis, the system design phase comes into play. Here, the architecture and technical components of the online voting system are meticulously outlined. This includes the development of a user-friendly interface, robust database management, and the implementation of advanced security features like two-factor authentication and face recognition.

**Implementation and Testing:** With the system design in place, the project moves on to the implementation and testing phase. This stage involves turning the design into a functional system. It includes the creation of a secure login system, the voting process, and mechanisms for result verification. Extensive testing is conducted to ensure the system's reliability and security.

**User Documentation:** User manuals and comprehensive documentation are developed to provide clear guidance to administrators and voters on how to effectively navigate and interact with the online voting system.

**Evaluation and Future Expansion:** After the system is operational, a critical evaluation is conducted to assess its success and limitations. The project's experiences and lessons learned are documented. Additionally, potential areas for future expansion or enhancement are identified, ensuring the system's continuous improvement.

### **2.2 Conceptual Requirements**

**Web Service:** The online voting system relies on web services to offer voters accessibility through a user-friendly web interface.

**Two-Factor Authentication:** Security is prioritized through the implementation of two-factor authentication, where users verify their identity using multiple means, such as an ID-card and PIN.

**Face Recognition:** To further enhance identity verification, face recognition technology is integrated into the system.

**Secure Database Management:** Given the sensitive nature of voter data and election results, robust database management capabilities are fundamental.

**User-Friendly Interface:** A user-friendly interface is a fundamental requirement to ensure that both voters and administrators can easily navigate and interact with the system.

**Transparency Mechanisms:** The system incorporates features designed to enhance transparency in the election process, allowing for effective monitoring and auditing of votes.

## **2.3 Existing Solutions**

### **Meelis Kitsing [1] - Estonia's Online Voting System**

Voters in Estonia start the online voting process by inserting their ID-card into a card reader and accessing the voting website ([www.valimised.ee](http://www.valimised.ee)). To verify their identity, voters use the personal identification number on their ID-card, which is issued along with PIN 2 and PUK codes. Both PIN codes are also used for various other online transactions requiring a digital signature. After entering the first PIN, the server checks the voter's eligibility by consulting the population register. Once eligibility is confirmed, voters are presented with a list of candidates in their electoral district. Voters select their preferred candidate, and their choice is encrypted. To confirm their decision, they must insert the digital signature in the form of the second PIN code. This submission concludes the voting process for individual voters.

**Limitation:** The system's limitation is that if someone finds another person's ID-card and PIN, they can access the voter's account and cast a vote. The system lacks two-factor authentication, such as the use of a one-time password, making it less secure.

## **Micha Germann, Flurin Conradin, Christoph Wellig, Uwe Serdül [2] - Swiss Online Voting System**

Switzerland's online voting system is designed within the framework of the country's federal structure. Voters log into their accounts to cast their votes online. To vote online, voters need specific codes, as is common with many online services. These codes are provided on the voter's polling card if they are entitled to vote online. Voters use the first code to log into the online voting system, where they can cast their vote. After voting, they can check if their vote has been correctly

recorded. The system provides various codes to allow voters to verify the accuracy of their input.

Limitation: This system lacks face recognition during the voting process, which potentially makes it less secure. Voters can log into their accounts and simply submit their votes without the added security of face recognition.

## **Joseph .D. Enoch, Nne .R. Saturday [3] - Nigeria's Online Voting System**

In Nigeria, voters log in by entering their username and password to access various parts of the system. Once logged in, voters can access voter registration data, candidate registration data, information about the electoral voting process, election party data, and general election results. To submit their votes, voters must scan their fingerprints. Votes are only submitted if the fingerprint matches. Admin users must log in by entering their username and password. After logging in, administrators can enter party candidates, create voter data, and add party information.

Limitation: This system relies solely on fingerprint authentication, lacking face recognition and two-factor authentication in the voting process.

# **Chapter-3**

## **Analysis**

### **3.1 Software Requirements**

**Web Server:** The web server for the online voting system will be Django, a high-level Python web framework. Django will provide the necessary tools for routing, handling HTTP requests, and serving web pages.

**Database Management System (DBMS):** PostgreSQL will be the chosen DBMS for storing and managing voter data, candidate information, and election results. PostgreSQL offers robust data integrity, scalability, and support for complex queries.

**Frontend Technologies:**

- **JavaScript:** The system will make extensive use of JavaScript, with a significant share of 42.8%. JavaScript will be used to create interactive and dynamic user interfaces, enhancing the user experience.
- **HTML:** HTML, constituting 37.6%, is essential for structuring and presenting web content. It will be used to design web pages and forms for voter interactions.
- **CSS:** CSS, making up 12.1%, will be used for styling web elements, ensuring a visually appealing and consistent design.
- **Python:** Python, accounting for 3.8%, will be utilized for server-side scripting within Django, ensuring the core functionality of the online voting system.
- **CoffeeScript:** Although less prominent at 3.5%, CoffeeScript may be used for enhancing JavaScript readability and maintainability.

- Less: With a minimal presence of 0.2%, Less may be employed to improve CSS development efficiency.

## Justification

- Web Server (Django): Django is a powerful, secure, and versatile web framework that simplifies web development. Its extensive libraries and features will facilitate rapid development and maintenance of the voting system.
- DBMS (PostgreSQL): PostgreSQL is a robust open-source database system known for its data integrity, scalability, and support for complex queries, making it ideal for securely storing election-related data.
- Frontend Technologies (JavaScript, HTML, CSS): These technologies are essential for creating an interactive, user-friendly, and visually appealing frontend. JavaScript enables dynamic user interactions, HTML structures web content, and CSS ensures a consistent design.
- Python: Python, used in the backend, is well-suited for handling complex server-side logic, ensuring the secure and efficient operation of the voting system.
- CoffeeScript and Less: These technologies, while having smaller shares, may be used to enhance code readability and streamline development processes, improving the maintainability of the system.

By incorporating these software requirements and technologies, the online voting system aims to deliver a robust, secure, and user-friendly experience for voters and administrators.

## 3.2 Hardware Requirements

The online voting system's hardware requirements are carefully defined to ensure performance, scalability, and data security.

- Web Servers: The voting system will be hosted on dedicated web servers with the following specifications:
- Processor: Multi-core processors with high clock speeds for efficient request handling.
- RAM: Sufficient RAM to accommodate concurrent user sessions and data processing. A minimum of 16 GB is recommended.
- Storage: Fast and redundant storage devices (SSDs) to ensure rapid data retrieval and system responsiveness.
- Network: Gigabit Ethernet connectivity for high data transfer rates.
- Redundancy: Load balancers and redundant servers for failover and high availability.

Database Server: The PostgreSQL database will require:

- Processor: A multi-core CPU with a strong emphasis on parallel processing.
- RAM: A substantial amount of RAM to optimize query performance and data caching, typically 32 GB or more.
- Storage: High-performance SSDs for efficient database access.
- Backup and Recovery: Robust backup and disaster recovery solutions for data protection.

- Client Devices: Voters and administrators will use various client devices, such as desktops, laptops, and mobile devices, with internet connectivity.

## Justification

- Web Servers: The dedicated web servers are equipped with powerful processors, ample RAM, and fast storage to handle a large number of concurrent users efficiently. Redundancy and load balancing ensure high availability and fault tolerance.
- Database Server: PostgreSQL database requires a high-performance CPU and substantial RAM to process complex queries and maintain data integrity. SSDs are crucial for data retrieval speed.
- Backup and Recovery: Data is a critical asset, and robust backup and recovery solutions are essential to safeguard against data loss or system failures.
- Client Devices: Voters and administrators may access the system from various devices. The system's web-based nature ensures compatibility with a wide range of client devices, making it accessible to a broad user base.

By adhering to these hardware requirements, the online voting system aims to deliver consistent, high-performance, and secure service for users during elections.

### 3.3 Analysis Diagrams

#### 3.3.1 Use Case Model

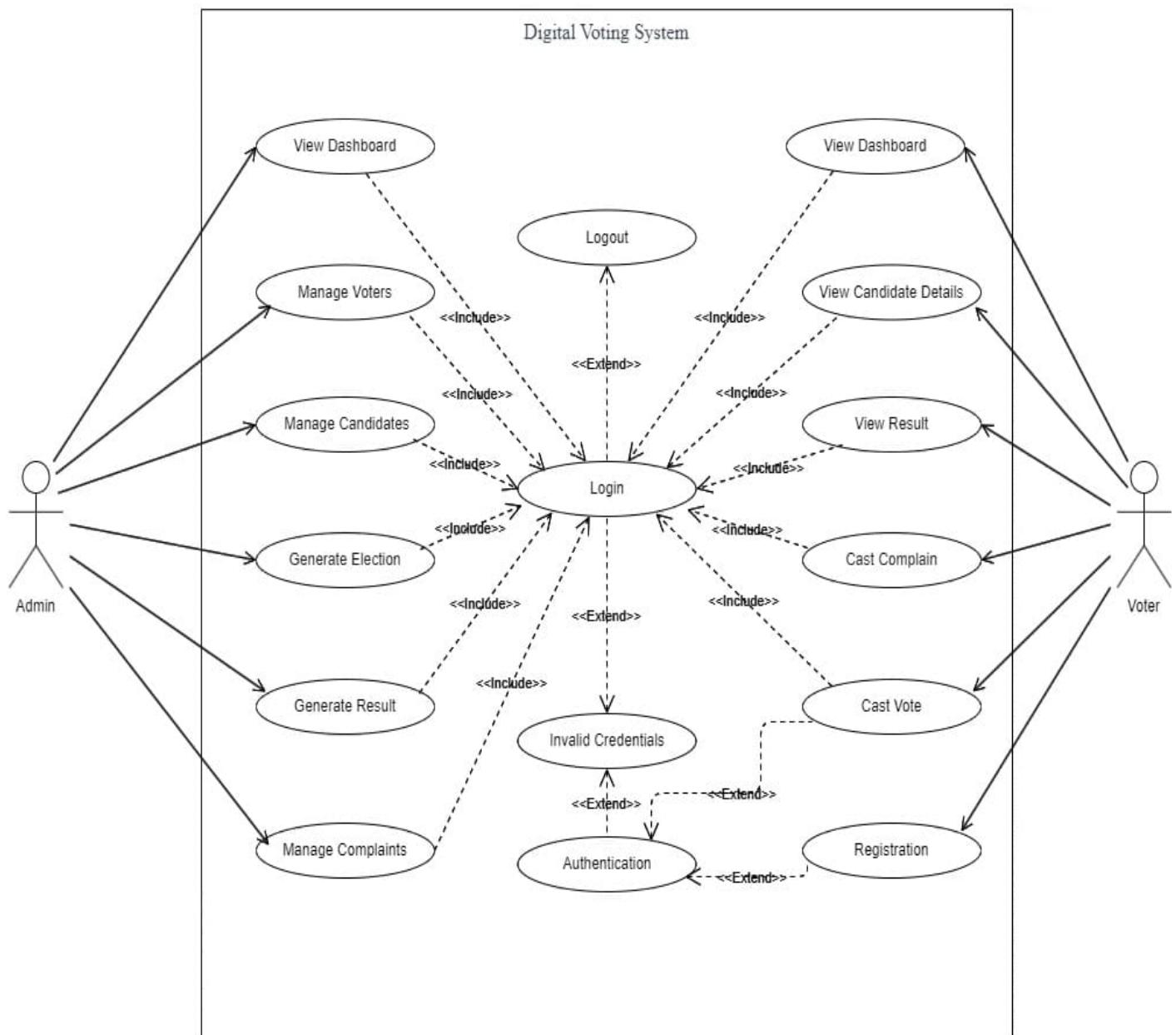


Fig: 3.1 Use Case Model

## **3.3.2 Use Case Description**

### **3.3.2.1 Admin Login**

#### **1.1 Brief Description**

This use case allows an administrator to log in to the system, gaining access to administrative features.

#### **1.2 Primary Actor**

Administrator

#### **1.3 Preconditions**

- The system is operational.
- The administrator has valid login credentials.
- The administrator is not already logged in.

#### **1.4 Postconditions**

- The administrator gains access to administrative functions.
- The system tracks the administrator's session.

#### **1.5 Basic Flow**

1. The administrator opens the system's login page.
2. The system presents the administrator with fields for username.
3. The administrator enters valid login credentials.
4. The system validates the credentials.
5. If the credentials are valid, the system logs in the administrator.
6. The system directs the administrator to the admin dashboard.

#### **1.6 Alternative Flows**

- If the administrator enters invalid credentials:
  - The system displays an error message.
  - The administrator is not logged in.

#### **1.7 Special Requirements**

None

### **3.3.2.2. Manage Voter Details**

## **2.1 Brief Description**

This use case enables the administrator to manage voter details, including registration, updates, and removal.

## **2.2 Primary Actor**

Administrator

## **2.3 Preconditions**

- The administrator is logged into the system.

## **2.4 Postconditions**

- Voter details are updated or removed as required.
- The system logs all changes made by the administrator.

## **2.5 Basic Flow**

1. The administrator navigates to the "Manage Voter Details" section.
2. The system displays a list of voter records.
3. The administrator selects a specific voter record to manage.
4. The administrator performs actions such as updating voter removing a voter from the system.
5. The system updates the voter database and logs the changes.

## **2.6 Alternative Flows**

None

## **2.7 Special Requirements**

None

### **3.3.2.3. Manage Candidate Details**

## **3.1 Brief Description**

This use case allows the administrator to manage candidate details, including registration and updates.

## **3.2 Primary Actor**

Administrator

## **3.3 Preconditions**

- The administrator is logged into the system.

### **3.4 Postconditions**

- Candidate details are updated or added as needed.
- The system logs all changes made by the administrator.

### **3.5 Basic Flow**

1. The administrator navigates to the "Manage Candidate Details" section.
2. The system displays a list of candidate records.
3. The administrator selects a specific candidate record to manage.
4. The administrator performs actions such as updating candidate information or adding new candidates.
5. The system updates the candidate database and logs the changes.

### **3.6 Alternative Flows**

None

### **3.7 Special Requirements**

None

## **3.3.2.4. Generate an Election**

### **4.1 Description**

This use case enables the administrator to generate an election event, defining candidates, voters, and voting parameters.

### **4.2 Primary Actor**

Administrator

### **4.3 Preconditions**

- The administrator is logged into the system.

### **4.4 Postconditions:**

- An election event is created with defined parameters.
- Candidates and voters are associated with the election.
- The system logs the election event's creation.

### **4.5 Basic Flow**

1. The administrator navigates to the "Generate an Election" section.
2. The system presents a form for defining election parameters.

3. The administrator fills in the election details, including candidates and eligible voters.
4. The system creates the election event, associates candidates and voters, and records the event's creation.

#### **4.6 Special Requirements**

None

##### **3.3.2.5. Voter Login**

###### **5.1 Description**

This use case enables registered voters to log in to the online voting system to access their voting privileges.

###### **5.2 Primary Actor**

Registered Voter

###### **5.3 Preconditions**

- The online voting system is operational.
- The voter is registered in the system.
- The voter is not already logged in.

###### **5.4 Postconditions**

- The voter successfully logs in and gains access to their voting options.
- The system tracks the voter's session.

###### **5.5 Basic Flow**

1. The registered voter navigates to the online voting system's login page.
2. The system presents the voter with fields to input their unique login credentials, typically a username or voter ID and a password.
3. The voter enters their valid login credentials.
4. The system validates the provided credentials.
5. If the credentials are verified as accurate, the system logs the voter in.
6. The system redirects the voter to the page displaying their voting options for the active election.

## **5.6 Alternate Flows**

- If the voter enters incorrect login credentials, the system displays an error message and does not allow login.
- If the voter is not registered in the system, they are informed that they must first register to vote.
- If the voter is already logged in, the system will not initiate another login session but will provide access to voting options.

## **5.7 Special Requirements**

- Ensure the security and confidentiality of voter login information.
- Implement measures to protect against unauthorized access or login attempts.
- Create a user-friendly and intuitive login interface.
- Maintain a secure session for each voter to safeguard data and maintain privacy

### **3.3.2.6. Cast a vote**

#### **6.1 Description**

In this use case, a registered voter successfully casts their vote through the online voting system.

#### **6.2 Primary Actor**

Registered Voter

#### **6.3 Preconditions**

- The voter is logged in to the online voting system.
- An active election is available.
- The voter has not already voted in the current election.

#### **6.4 Postconditions**

- The voter's vote is securely recorded and counted within the system.
- The voter's voting status is updated to "voted" in the system.
- The voter receives acknowledgment of their participation.

## **6.5 Basic Flow**

1. The registered voter logs in to the online voting system using their unique credentials.
2. The system displays the list of candidates or options for the active election.
3. The voter selects their preferred candidate or casts their vote as per the election requirements.
4. The voter confirms their selection.
5. The system records the vote securely and updates the voter's status to "voted."
6. The system provides an acknowledgment of the voter's participation in the election and logs the voter out.

## **6.6 Alternate Flows**

- If the voter attempts to vote without being logged in, the system directs them to the login process.
- If there is no active election available, the system notifies the voter that there are no ongoing elections.
- If the voter has already voted in the current election, the system informs them that their vote has been counted, and they cannot vote again in the same election.

## **6.7 Special Requirements**

- Implement measures to ensure that each voter can cast only one vote per election .
- Maintain data integrity to prevent double voting and uphold data accuracy.
- Create a user-friendly and secure voting interface.
- Implement robust security measures to protect the voting process and maintain voter data confidentiality.

### **3.3.2.7. Cast a Complaint**

#### **7.1 Brief Description**

This use case enables registered voters to cast complaints related to the online voting system.

## **7.2 Primary Actor**

Registered Voter

## **7.3 Preconditions**

- The online voting system is operational.
- The voter is logged in to the system.
- The voter has encountered an issue or concern they wish to report.

## **7.4 Postconditions**

- The voter's complaint is successfully submitted to the system.
- The system logs the complaint, including relevant details and timestamps.
- The system acknowledges the complaint submission to the voter.

## **7.5 Basic Flow**

1. The registered voter, while logged into the online voting system, navigates to the "Complaints" section.
2. The system presents a form for submitting complaints, including fields for describing the issue, attaching relevant documents if necessary, and selecting a category (e.g., technical issue, security concern, user experience feedback).
3. The voter completes the complaint form, providing detailed information about the issue they encountered.
4. The voter reviews the information entered and confirms their intent to submit the complaint.
5. The system securely transmits the complaint data.
6. The system records the complaint, including a timestamp and the voter's user ID.
7. The system sends an acknowledgment to the voter, confirming the successful submission of the complaint.
8. The complaint is categorized and prioritized based on its severity and nature.

## **7.6 Alternative Flows**

- If the voter cancels the complaint submission before confirmation, the system discards the entered information, and no complaint is registered

- If there is an issue with the submission process (e.g., network error), the system prompts the voter to retry the submission or contact support.

## **7.7 Special Requirements:**

- Implement secure and encrypted data transmission for complaint submissions to protect the privacy and confidentiality of the complainant
- Ensure that the complaint form is user-friendly, intuitive, and includes clear instructions for providing details.
- Implement measures to detect duplicate complaints to prevent the same issue from being reported multiple times.
- Categorize and prioritize complaints for efficient handling and resolution.

### **3.3.2.8. Filter Election Results**

#### **8.1 Brief Description**

This use case enables administrators to filter and view election results based on various criteria.

#### **8.2 Primary Actor**

Administrator

#### **8.3 Preconditions**

- The administrator is logged into the online voting system.
- An election has concluded, and results are available.

#### **8.4 Postconditions**

- The administrator successfully filters and views election results based on the selected criteria.
- The system displays the filtered results, allowing the administrator to analyze and export the data if needed.

#### **8.5 Basic Flow**

1. The administrator, after logging into the online voting system, navigates to the "Election Results" section.
1. The system presents a set of filter options to the administrator, including criteria such as election date, electoral district, candidate, and voter demographics.

3. The administrator selects the desired filter criteria (e.g., a specific election, candidate, or date range).
4. The system processes the selected criteria and retrieves the corresponding
5. The system displays the filtered results, which may include tabular data and graphical representations.
6. The administrator can further analyze the results or choose to export the data for reporting or auditing purposes.

## **8.6 Alternative Flows**

- If there are no election results available based on the selected criteria, the system informs the administrator that no data matches the chosen filters and prompts them to adjust their criteria.
- If there is an issue with the data retrieval process (e.g., database error), the system notifies the administrator and suggests reattempting the operation.

## **8.7 Special Requirements**

- Ensure that the filtering process is intuitive and user-friendly, allowing administrators to easily customize the criteria.
- Implement data visualization tools to provide graphical representations of election results for enhanced analysis.
- Enable data export functionality to allow administrators to generate reports in various formats (e.g., PDF, CSV) for documentation and auditing purposes.
- Implement role-based access control to ensure that only authorized administrators can access and filter election results.





## 3.4 Design Diagrams

### 3.4.2 Sequence diagrams.

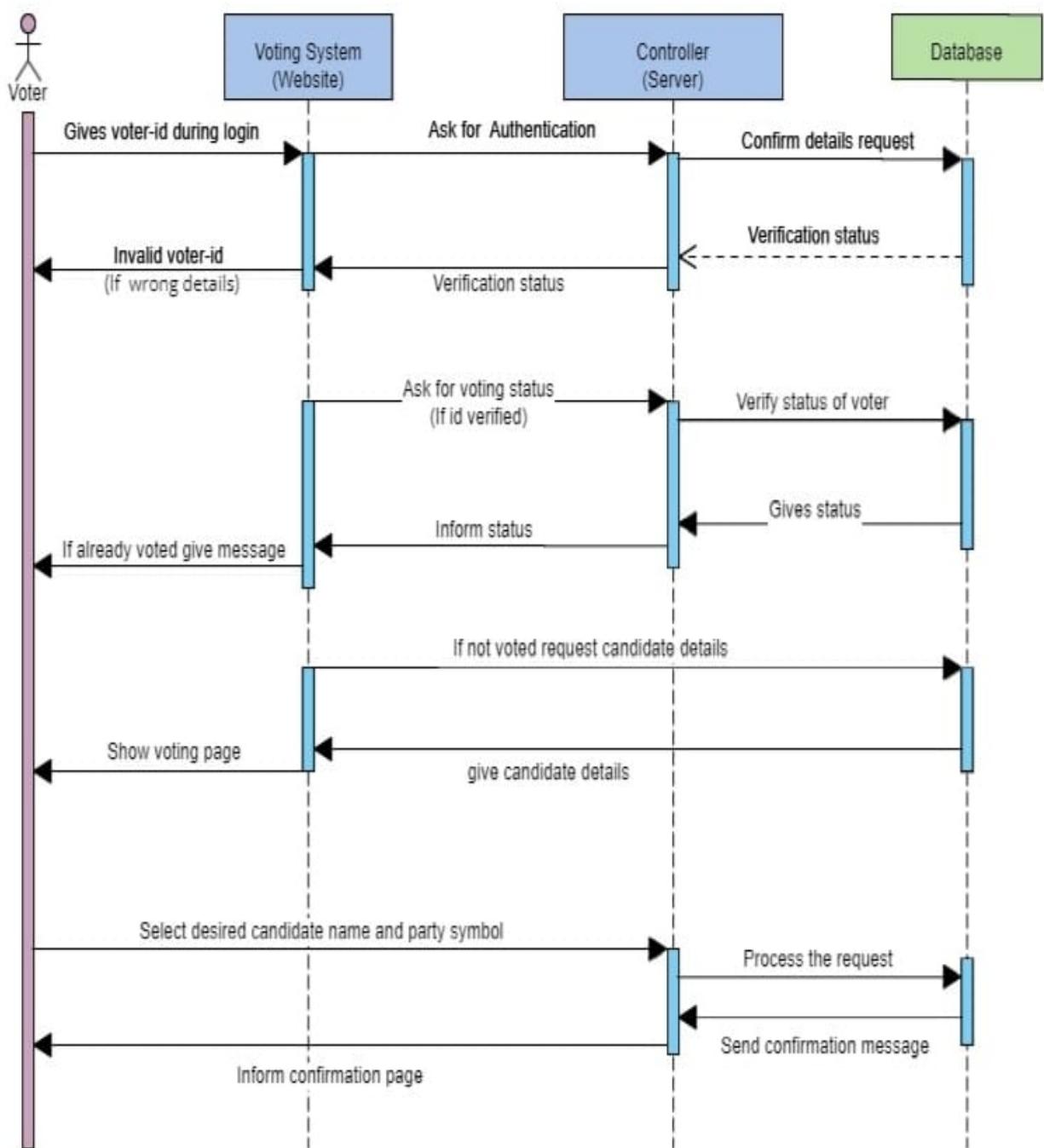


Fig: 3.2 Sequence Diagram

# Chapter – 4

## Implementation and Testing

### 4.1 Database Design

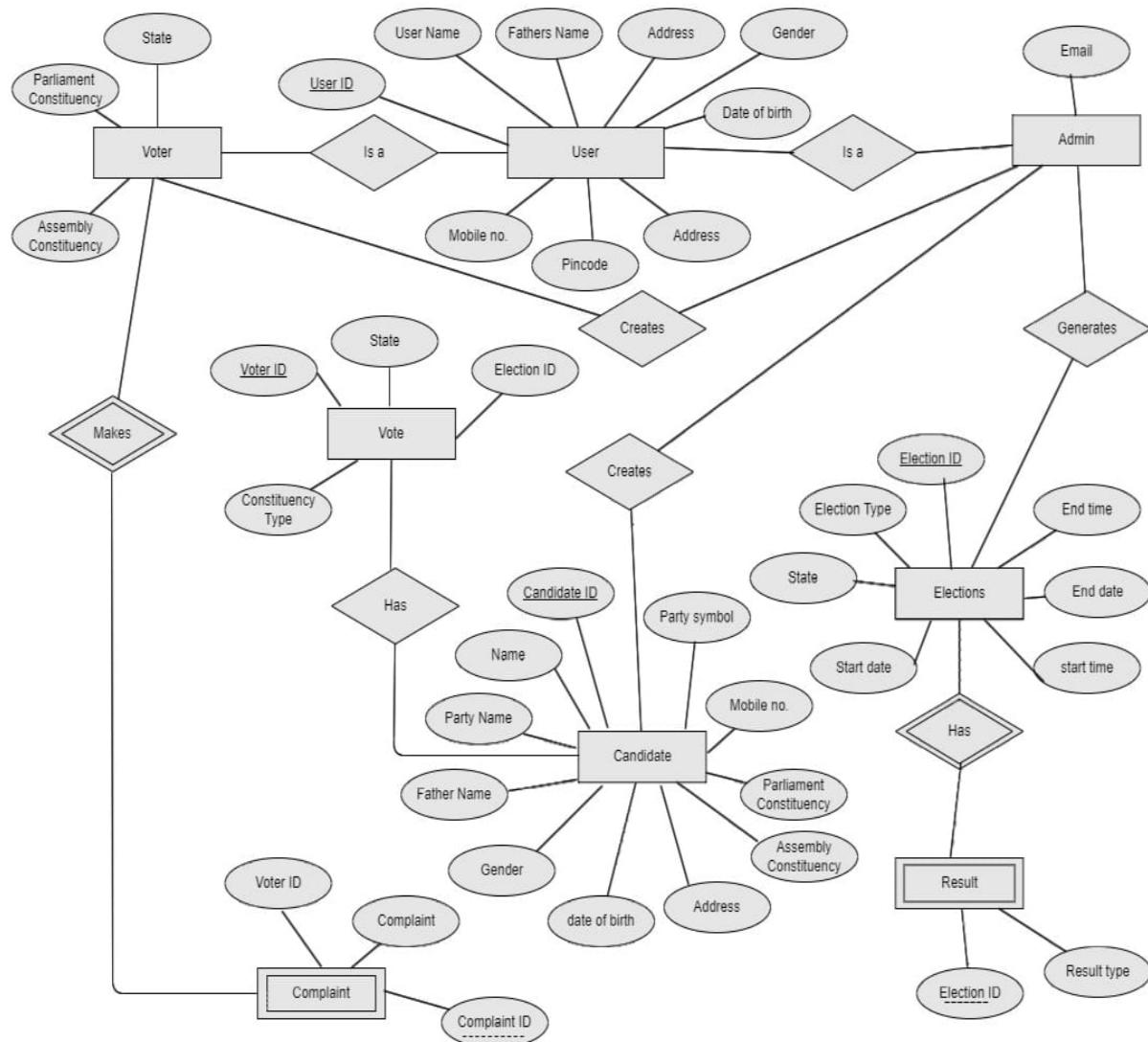


Fig: 4.1 Entity-Relationship(ER) Diagram



Fig: 4.2 Database Snaps(All 18 tables)

The screenshot shows the pgAdmin 4 interface with the title bar "pgAdmin 4". The main window has tabs for Dashboard, Properties, SQL, Statistics, Dependencies, Dependents, Processes, and a scratch pad. The SQL tab is active, displaying the following query:

```
1 SELECT * FROM public."EC_Admin_election"
2 ORDER BY id ASC
```

The results pane shows a table with the following data:

	<b>id</b> [PK] integer	<b>election_id</b> character varying (50)	<b>election_type</b> character varying (50)	<b>state</b> character varying (50)	<b>start_date</b> date	<b>start_time</b> time without time zone	<b>end_date</b> date	<b>end_time</b> time without time zone	<b>status</b> character varying (10)
1	1	111111	AC-GENERAL	GUJARAT	2023-10-27	17:45:00	2023-10-27	16:00:00	not active
2	2	222222	PC-GENERAL	GUJARAT	2023-10-27	18:19:00	2023-10-27	18:23:00	not active
3	3	333333	PC-GENERAL	GUJARAT	2023-10-27	18:40:00	2023-10-27	18:50:00	not active
4	4	444444	PC-GENERAL	GUJARAT	2023-10-27	20:05:00	2023-10-27	20:10:00	not active
5	5	555555	PC-GENERAL	GUJARAT	2023-10-27	20:15:00	2023-10-27	23:59:00	not active
6	6	666666	PC-GENERAL	GUJARAT	2023-10-28	09:24:00	2023-10-28	23:59:00	not active
7	7	777777	PC-GENERAL	GUJARAT	2023-10-29	01:26:00	2023-10-29	23:50:00	not active

The status bar at the bottom shows "Servers > PostgreSQL 16 > Databases > postgres > Schemas > public > Tables > EC\_Admin\_election", "Ln 1, Col 1", "20°C Near record", and system icons.

Fig: 4.3 Database Snaps(Elections)

pgAdmin 4

File Object Tools Help

Object Explorer

- Functions
- Materialized Views
- Operators
- Procedures
- Sequences
- Tables (18)
  - EC\_Admin\_candidates
  - EC\_Admin\_ec\_admins
  - EC\_Admin\_election
  - EC\_Admin\_reports
  - EC\_Admin\_voters
  - EC\_Admin\_votes
  - auth\_group
  - auth\_group\_permissions
  - auth\_permission
  - auth\_user
  - auth\_user\_groups
  - auth\_user\_permissions
  - djongo\_admin\_log
  - djongo\_content\_type
  - djongo\_migrations
  - djongo\_session
  - voter\_complain
  - voter\_voted
- Trigger Functions
- Types
- Views
- Subscriptions

Login/Group Roles

Tablespaces

Total rows: 14 of 14 | Query complete 00:00:00.329 | Ln 2, Col 17

Top events Event brief

ID	election_id	candidate_id	candidate_name	candidate_party	state	constituency	online_votes	evm_votes
1	222222	666666	viru	jail road mitramandal	GUJARAT	Jamnagar	0	
2	222222	888888	subbu sharma	karam karta party	GUJARAT	Jamnagar	0	
3	333333	666666	viru	jail road mitramandal	GUJARAT	Jamnagar	0	
4	333333	888888	subbu sharma	karam karta party	GUJARAT	Jamnagar	0	
5	444444	666666	viru	jail road mitramandal	GUJARAT	Jamnagar	0	
6	444444	888888	subbu sharma	karam karta party	GUJARAT	Jamnagar	0	
7	555555	666666	viru	jail road mitramandal	GUJARAT	Jamnagar	0	
8	555555	888888	subbu sharma	karam karta party	GUJARAT	Jamnagar	0	
9	666666	666666	viru	jail road mitramandal	GUJARAT	Jamnagar	0	
10	666666	888888	subbu sharma	karam karta party	GUJARAT	Jamnagar	0	
11	777777	666666	viru	jail road mitramandal	GUJARAT	Jamnagar	0	
12	777777	888888	subbu sharma	karam karta party	GUJARAT	Jamnagar	0	
13	777777	999999	gayatri sharma	Bhartiya janta party	GUJARAT	Rajkot	2	
14	777777	222222	Rajesh pathak	congress	GUJARAT	Rajkot	0	

Fig: 4.4 Database Snaps(votes)

pgAdmin 4

File Object Tools Help

Object Explorer

- Languages
- Publications
- Schemas (1)
  - public
    - Aggregates
    - Collations
    - Domains
    - FTS Configurations
    - FTS Dictionaries
    - FTS Parsers
    - FTS Templates
    - Foreign Tables
    - Functions
    - Materialized Views
    - Operators
    - Procedures
    - Sequences
- Tables (18)
  - EC\_Admin\_candidates
  - EC\_Admin\_ec\_admins
  - EC\_Admin\_election
  - EC\_Admin\_reports
  - EC\_Admin\_voters
  - EC\_Admin\_votes
  - auth\_group
  - auth\_group\_permissions
  - auth\_permission
  - auth\_user
  - auth\_user\_groups

Servers > PostgreSQL 16 > Databases > postgres > Schemas > public > Tables > EC\_Admin\_ec\_admins | Ln 1, Col 1

20°C Haze

ID	ecadmin_id	firstname	lastname	middlename	gender	dateofbirth	address	pincode
1	sharma	Surbhi	Sharma	na	female	2001-09-03	indore	452009

Fig: 4.5 Database Snaps(admin)

pgAdmin 4

File Object Tools Help

Object Explorer    Dashboard Properties SQL Statistics Dependencies Dependents Processes public.EC\_Adm... public.EC\_Adm... public.EC\_Adm... public.EC\_Adm...

Query Query History

```
1 SELECT * FROM public."EC_Admin_voters"
2 ORDER BY id ASC
```

Data Output Messages Notifications

	<b>id</b> [PK] integer	<b>voterid_no</b> character varying (10)	<b>name</b> character varying (100)	<b>father_name</b> character varying (100)	<b>gender</b> character varying (6)	<b>dateofbirth</b> date	<b>address</b> character varying (1024)	<b>mobile_no</b> bigint	<b>state</b> character varying (50)
1	1	1234567890	sabbu	govi	Female	2002-11-12	indore	7748089941	GUJARAT
2	2	3456789012	surbhi	gopi	Female	2001-09-03	gwalior	7247729784	GUJARAT
3	3	7890123456	sangeeta	govind	Female	2000-10-12	indore	7748089941	GUJARAT

Total rows: 3 of 3    Query complete 00:00:00.283    ✓ Successfully run. Total query runtime: 283 msec. 3 rows affected.    Ln 1, Col 1

Fig: 4.6 Database Snaps(voters)

pgAdmin 4

File Object Tools Help

Object Explorer    Dashboard Properties SQL Statistics Dependencies Dependents Processes public.EC\_Adm... public.EC\_Admin\_candidates/postgres@PostgreSQL 16

Query Query History

```
1 SELECT * FROM public."EC_Admin_candidates"
2 ORDER BY id ASC
```

Data Output Messages Notifications

	<b>id</b> [PK] integer	<b>candidate_id</b> character varying (10)	<b>name</b> character varying (100)	<b>father_name</b> character varying (100)	<b>gender</b> character varying (6)	<b>dateofbirth</b> date	<b>address</b> character varying (1024)	<b>mobile_no</b> bigint	<b>state</b> character varying (50)
1	1	666666	viru	goku	MALE	2002-08-31	indore	6261055689	GUJARAT
2	2	888888	subbu sharma	raghu	FEMALE	2001-09-03	indore	7247729784	GUJARAT
3	3	999999	gayatri sharma	druv sharma	FEMALE	1965-09-03	indore	7247729784	GUJARAT
4	4	222222	Rajesh pathak	mahesh pathak	MALE	1962-12-13	JADAV COLONY,	7748089941	GUJARAT

Total rows: 4 of 4    Query complete 00:00:00.247    Ln 1, Col 1

Fig: 4.7 Database Snaps(candidates)

## 4.2 Class diagram

Digital Voting System Class Diagram

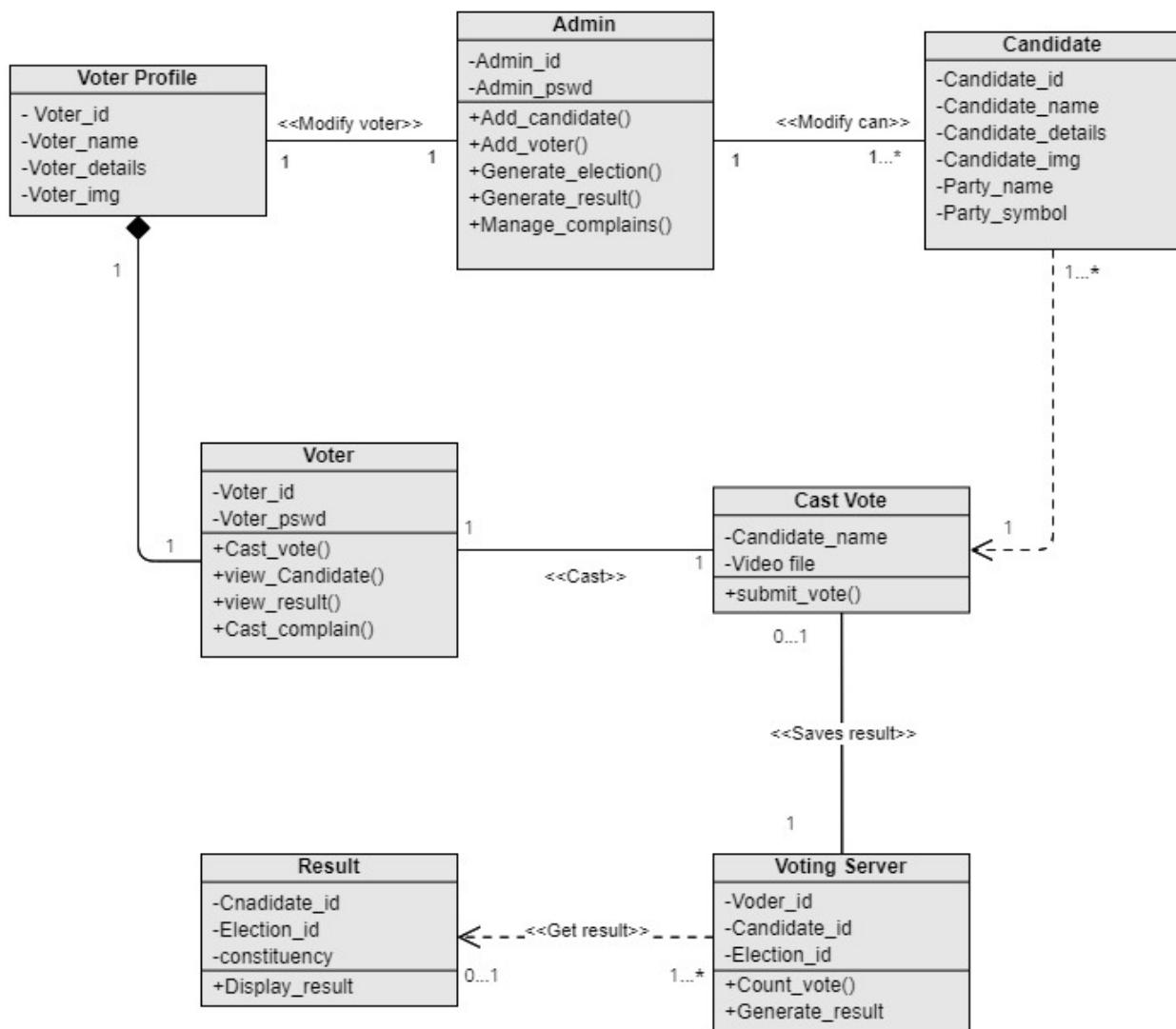


Fig: 4.8 Class Diagram

In the class diagram for the Online Voting System project, we have several essential classes representing the core elements of the system:

## **1. Voter:**

- Purpose: The Voter class represents registered voters in the system, each instance corresponding to a voter profile.
- Methods: Voter methods include voter authentication, login, and cast vote. These methods typically take parameters such as a username, password, and election ID for login and vote submission.

## **2. CastVote**

- Purpose: The CastVote class handles the voting process, tracking the voter's selections and ensuring their vote is securely recorded.
- Methods: CastVote methods manage the vote casting process, verifying the voter's eligibility, recording their choices, and updating the vote count. Parameters may include voter details, candidate IDs, and election information.

## **3. Candidate**

- Purpose: The Candidate class represents individuals running for elected positions in an election, with each instance corresponding to a candidate profile.
- Methods: Candidate methods include candidate registration and document upload. Parameters for these methods may include candidate details like name, party affiliation, and documents to be uploaded.

## **4. Admin**

- Purpose: The Admin class represents administrators who manage and oversee the online voting system, responsible for tasks like election setup and result filtering.
- Methods: Admin methods involve tasks like election creation, candidate approval, and result filtering. Parameters include election details, candidate IDs, and filtering criteria.

## **5. VotingServer**

- Purpose: The VotingServer class manages the core functionalities of the online voting system, ensuring data integrity and secure voting processes.
  - Methods: VotingServer methods handle tasks related to server setup, voter authentication, and ensuring data consistency. Parameters include server configurations and voter information.

## **6. VoterProfile**

- Purpose: The VoterProfile class stores and manages the personal information and preferences of voters.
  - Methods: VoterProfile methods enable voters to update their profile, change passwords, and upload documents. Parameters would vary based on the specific user profile changes.

## **7. Result**

- Purpose: The Result class is responsible for storing and managing election results, including vote counts and candidate standings.
  - Methods: Result methods facilitate the updating and retrieval of election results, with parameters including election details and candidate data.

The class diagram visually represents the structure of the Online Voting System, illustrating how different classes interact and encapsulate data and behavior. It provides an overview of the key elements and their relationships within the system, facilitating a deeper understanding of how data and operations are organized and executed.

## 4.3 Test Cases

Test 01: Test Case for Invalid Credentials

Login ID: sita

Password: ABD1234

System Output: Invalid Credentials



Fig: 4.9

Test Case 02: Test for Invalid Login

Login ID: sita

Password: pass123

System Output: Invalid login. Please check your credentials and try again.

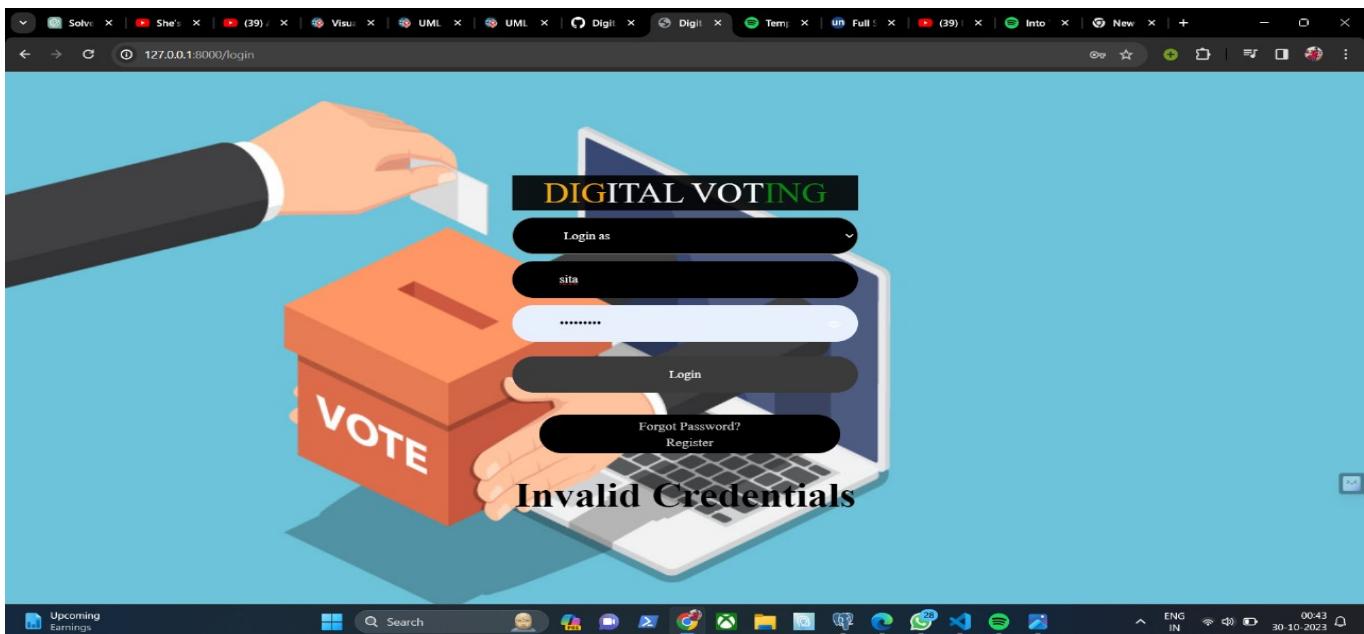


Fig: 4.10

Test Case 03: Vote for Gayatri Sharma

Selected Candidate: Candidate Name (Gayatri Sharma)

System Output: Your vote for Gayatri Sharma has been successfully recorded. Thank you for participating in the election!

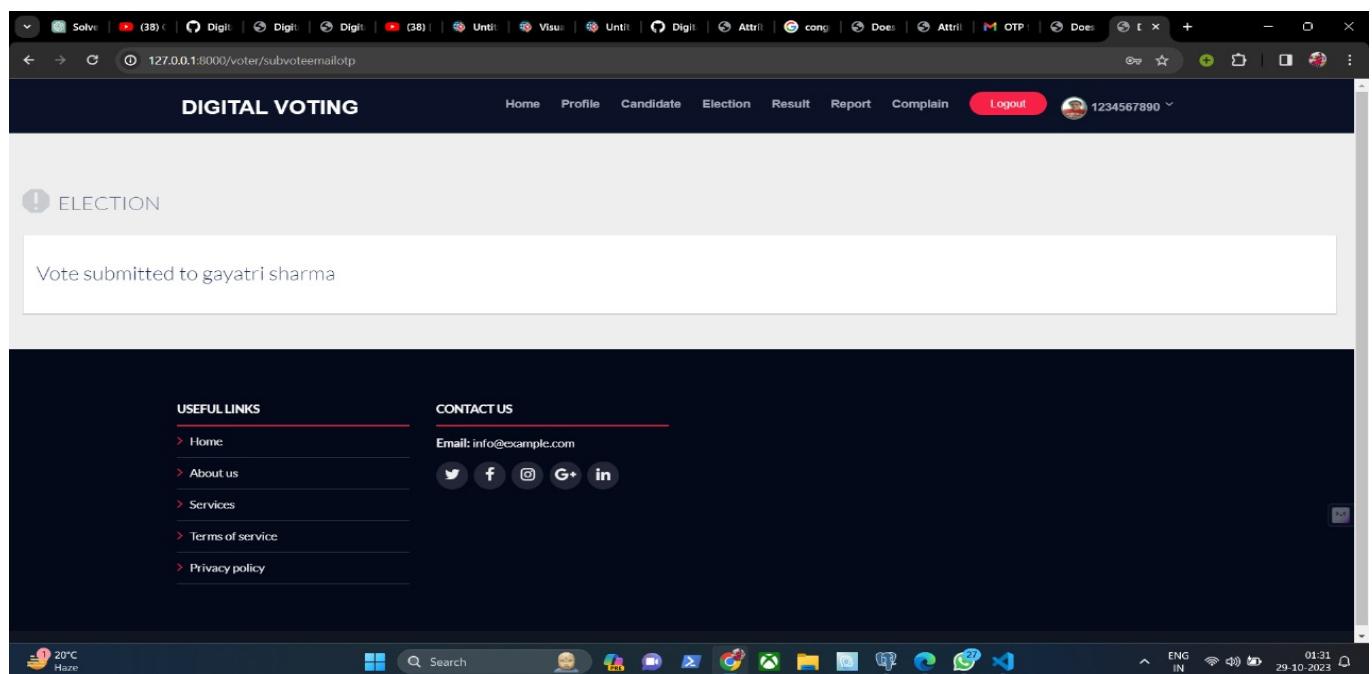


Fig: 4.11

**Black Box Testing** : Black Box Testing is a software testing method where the internal structure, logic, and code of the application are not known to the tester. Instead, the tester focuses on testing the application's functionality, behavior, and outputs by providing inputs and evaluating the corresponding outputs. This testing approach assesses the application's functionality against its specifications and requirements, ensuring that it behaves as expected from a user's perspective. It doesn't require knowledge of the underlying code and can uncover issues related to incorrect functionality, usability, performance, and security. Black Box Testing is valuable for validating that the software meets user expectations and requirements.

**User Acceptance test** : User Acceptance Testing (UAT) is the final phase of software testing in which end-users, or client representatives, evaluate the software to determine if it meets their needs and requirements. During UAT, users interact with the software to verify its functionality, usability, and overall suitability for their specific use cases.

## 4.4 Illustration



Fig:4.12 Admin Login

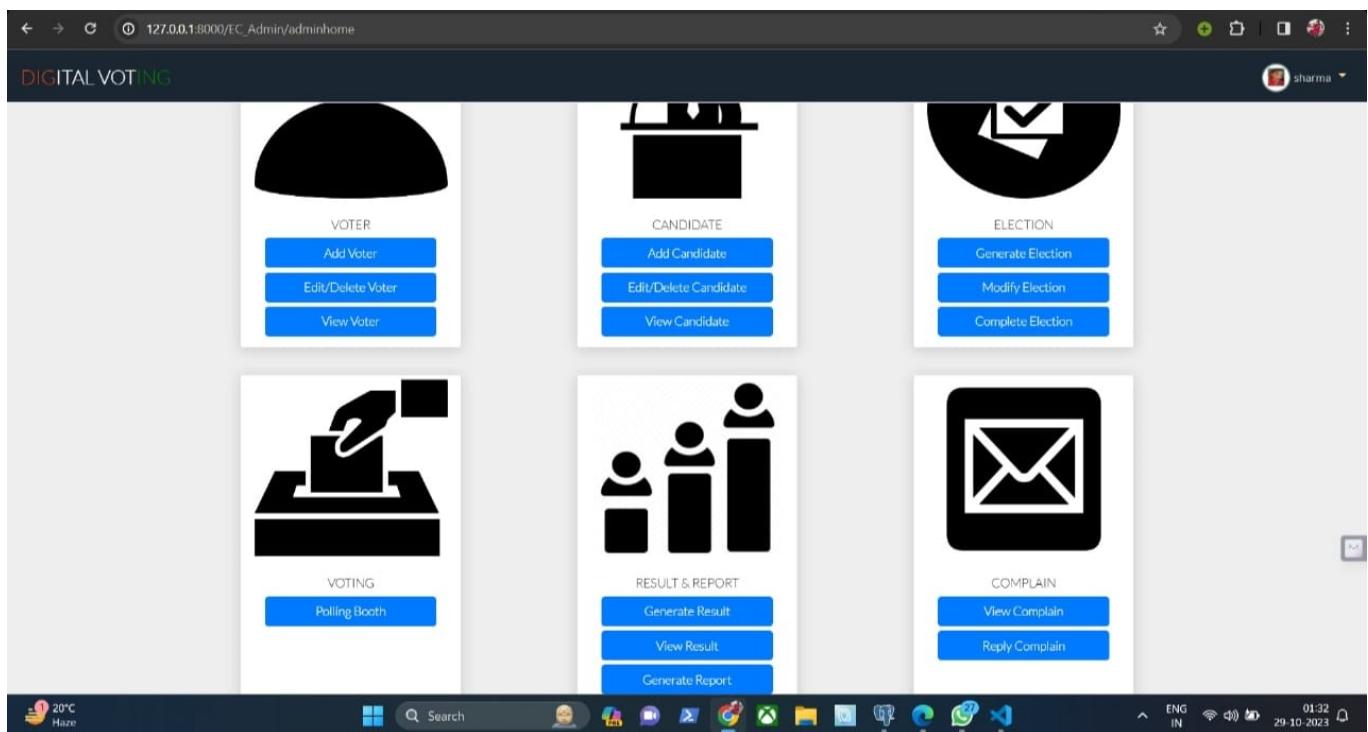


Fig: 4.13 Admin Dashboard

The screenshot shows the 'Add Candidate' page of a web application. The URL is 127.0.0.1:8000/EC\_Admin/addcandidate. The page title is 'ADD CANDIDATE'. On the left, there is a sidebar with navigation links: Home, Profile, Voter, Candidate, Election, Voting, Result & Report, Complain, and Logout. The main content area is titled 'Candidate' and contains fields for Candidate ID, Candidate Name, Father Name, Gender (with a dropdown menu showing 'GENDER'), Birth Date (with a date input field), Address, and Pincode. At the bottom of the page, there is a toolbar with various icons and system status indicators like battery level, signal strength, and date/time.

Fig: 4.14 Add Candidate(Admin Feature)

The screenshot shows the 'View Voter' page of a web application. The URL is 127.0.0.1:8000/EC\_Admin/view\_voter. The page title is 'VIEW VOTER'. The left sidebar has the same navigation links as Fig 4.14. The main content area is titled 'Voter' and includes three dropdown filters: 'State/UT', 'Constituency Type', and 'Constituency'. Below these is a blue 'Filter' button. At the bottom, there is a table header for 'List of voter' with columns: photo, voter ID, Full Name, Gender, Date of birth, State, Parliamentary, and Assembly. The table body is currently empty. The bottom of the screen features a toolbar with system icons and status information.

Fig: 4.15 View Voter

The screenshot shows a web-based digital voting application. The URL in the address bar is 127.0.0.1:8000/EC\_Admin/generateelection. The page title is "GENERATE ELECTION". On the left, there is a vertical navigation menu with options: Home, Profile, Voter, Candidate, Election, Voting, Result & Report, Complain, and Logout. The "Logout" option has a dropdown arrow. The main content area is titled "Election". It contains several input fields: "Election ID" (text input), "Election Type" (dropdown menu set to "Election T"), "State" (dropdown menu set to "Please select an State/UT"), "Start Date" (date input set to "dd-mm-yyyy"), "Start Time" (time input set to "----"), "End Date" (date input set to "dd-mm-yyyy"), "End Time" (time input set to "----"), and two buttons at the bottom: "Generate Election" (blue) and "Reset". Below the form, the Windows taskbar is visible, showing various pinned icons like File Explorer, Google Chrome, and Microsoft Office. The system tray indicates the date as 29-10-2023 and the time as 01:37.

Fig: 4.16 Generate Election

The screenshot shows a web-based digital voting application. The URL in the address bar is 127.0.0.1:8000/EC\_Admin/replycomplain. The page title is "VIEW COMPLAIN". The left navigation menu is identical to Fig 4.16. The main content area is titled "Complain". It features a table with three columns: "Complain ID", "Voter ID", and "Complain". Below the table, a message reads "No New Complaints". The Windows taskbar and system tray are visible at the bottom, showing the same date and time as Fig 4.16.

Fig: 4.17 View Complaint

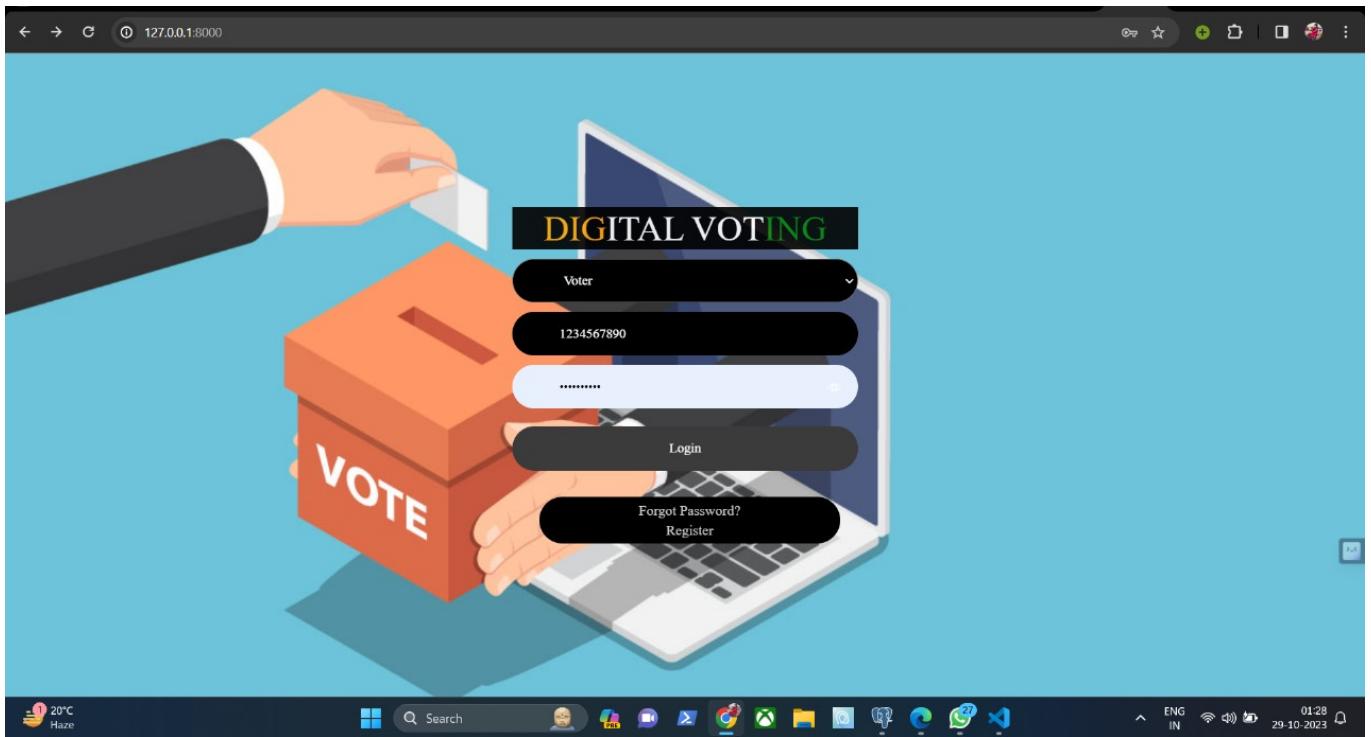


Fig: 4.18 Voter Login



Fig: 4.19 Voter Dashboard

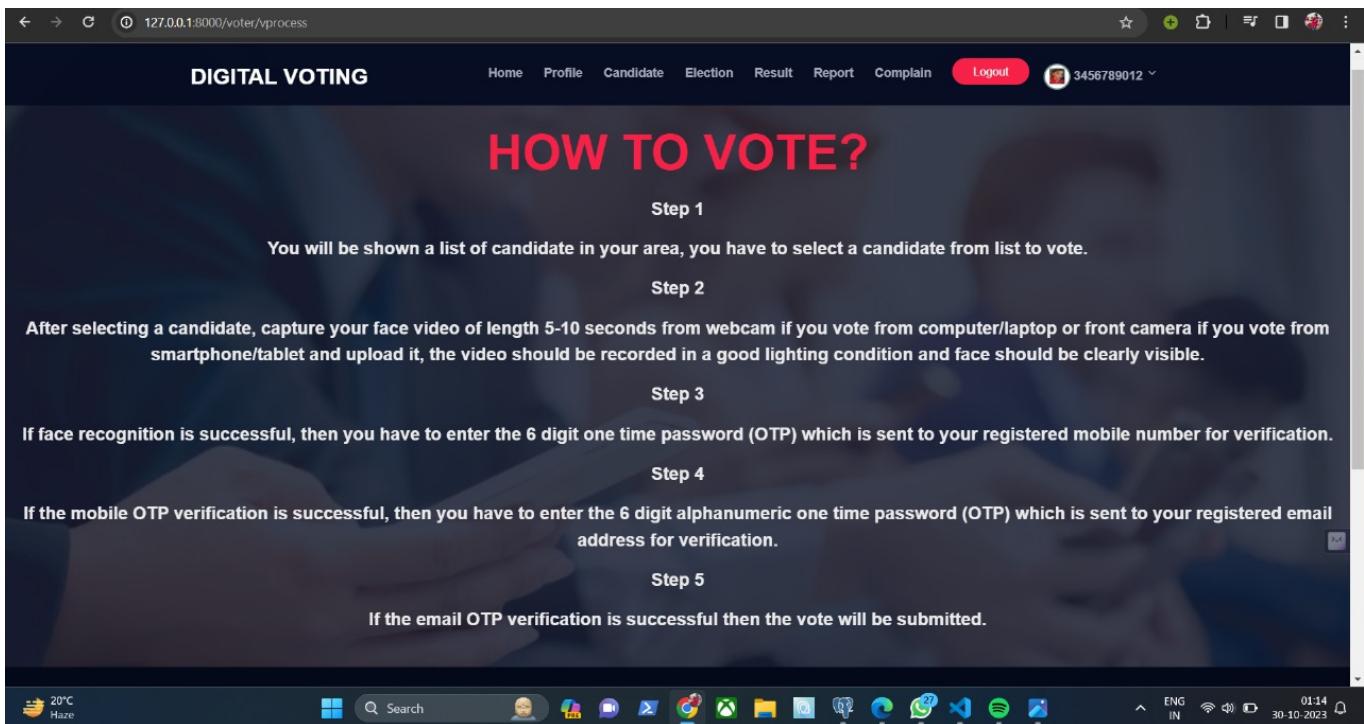


Fig: 4.20 Guidelines To Vote

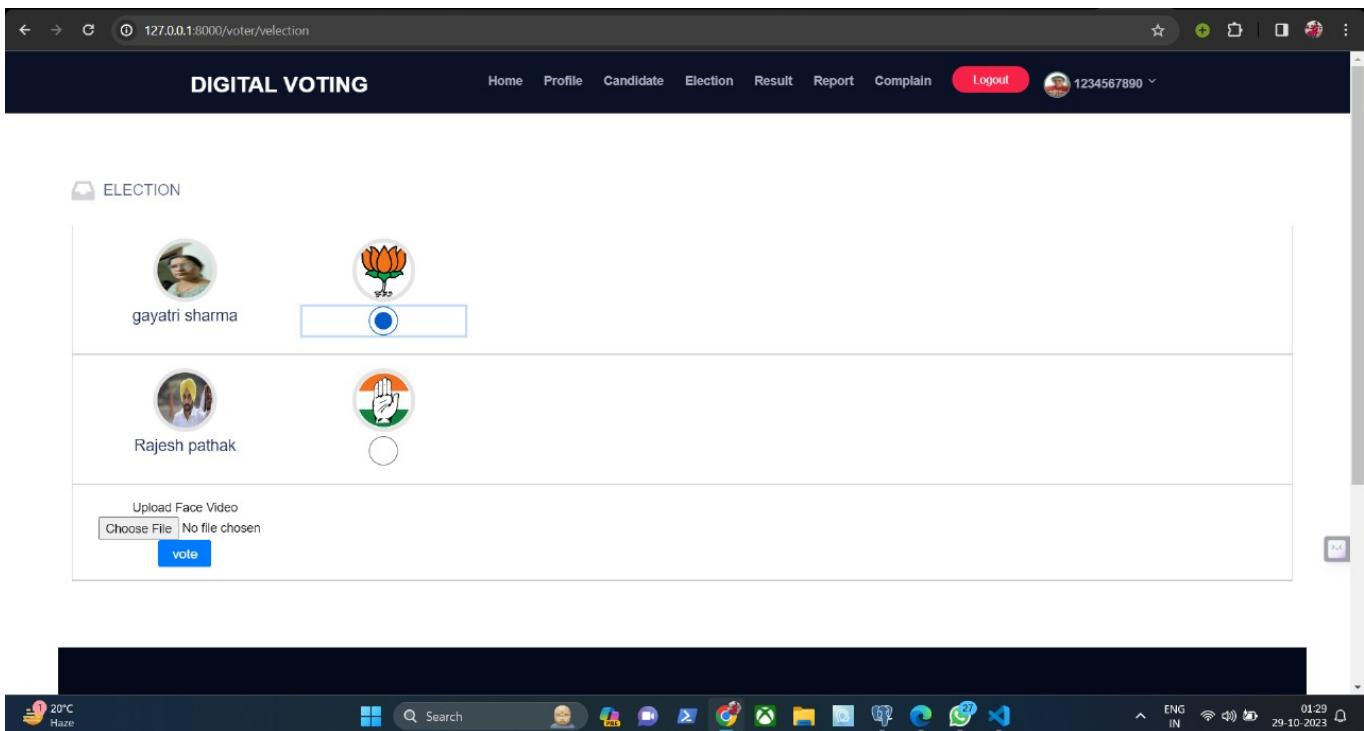


Fig: 4.21 Cast Vote

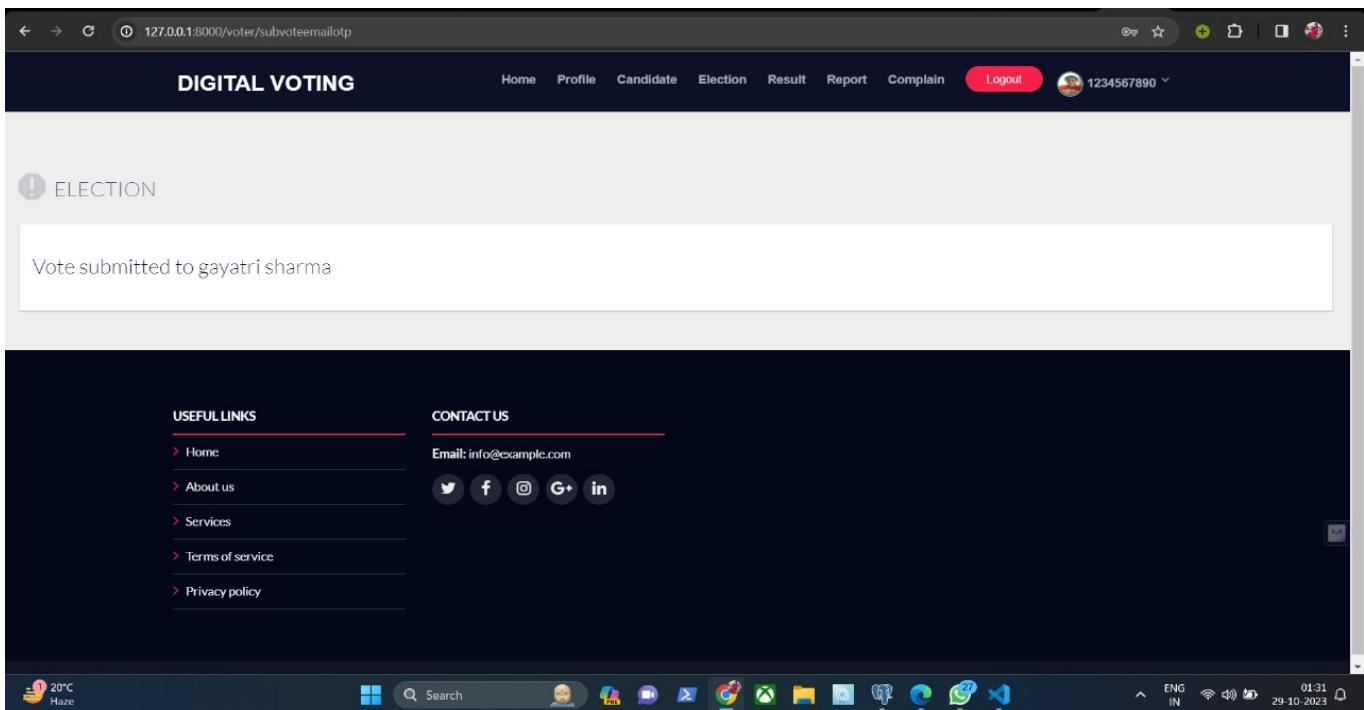


Fig: 4.22 Vote Submitted

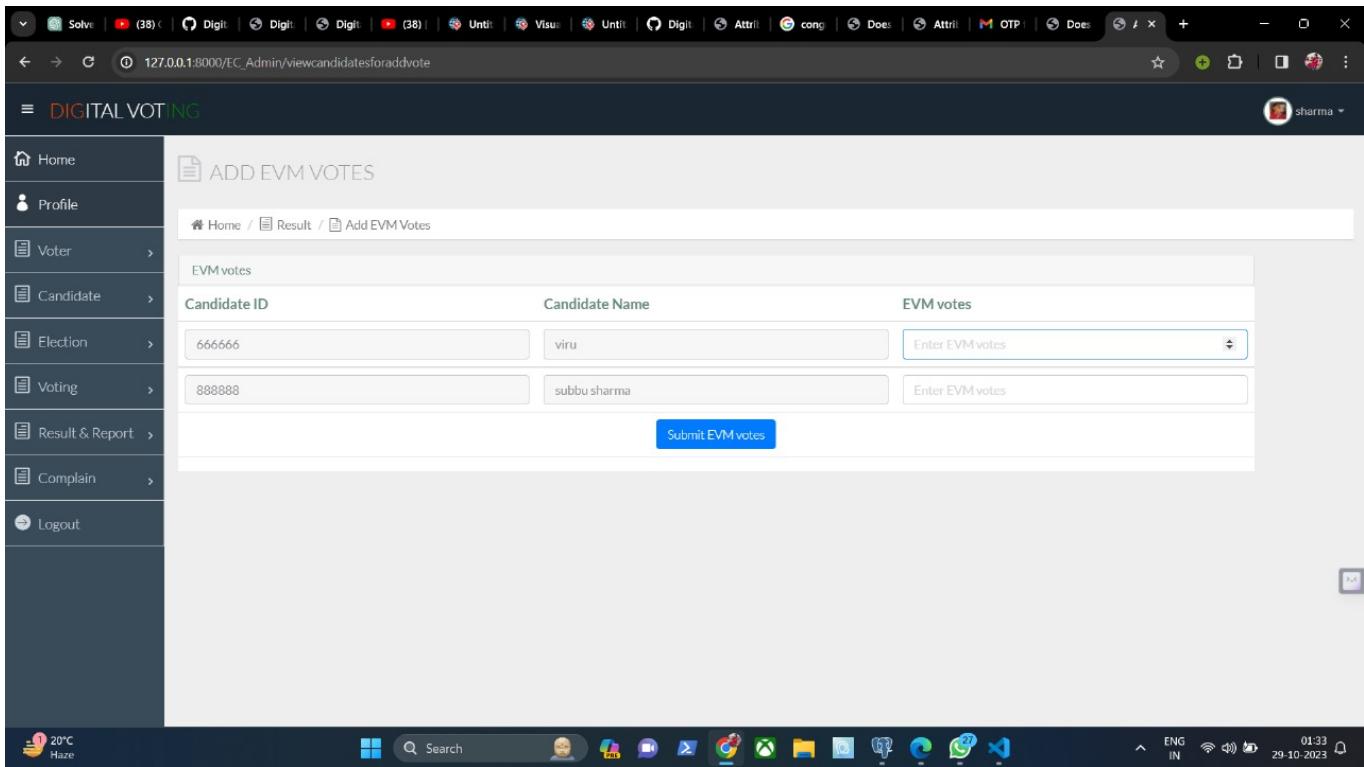


Fig: 4.23 Add EVM Votes(Admin)

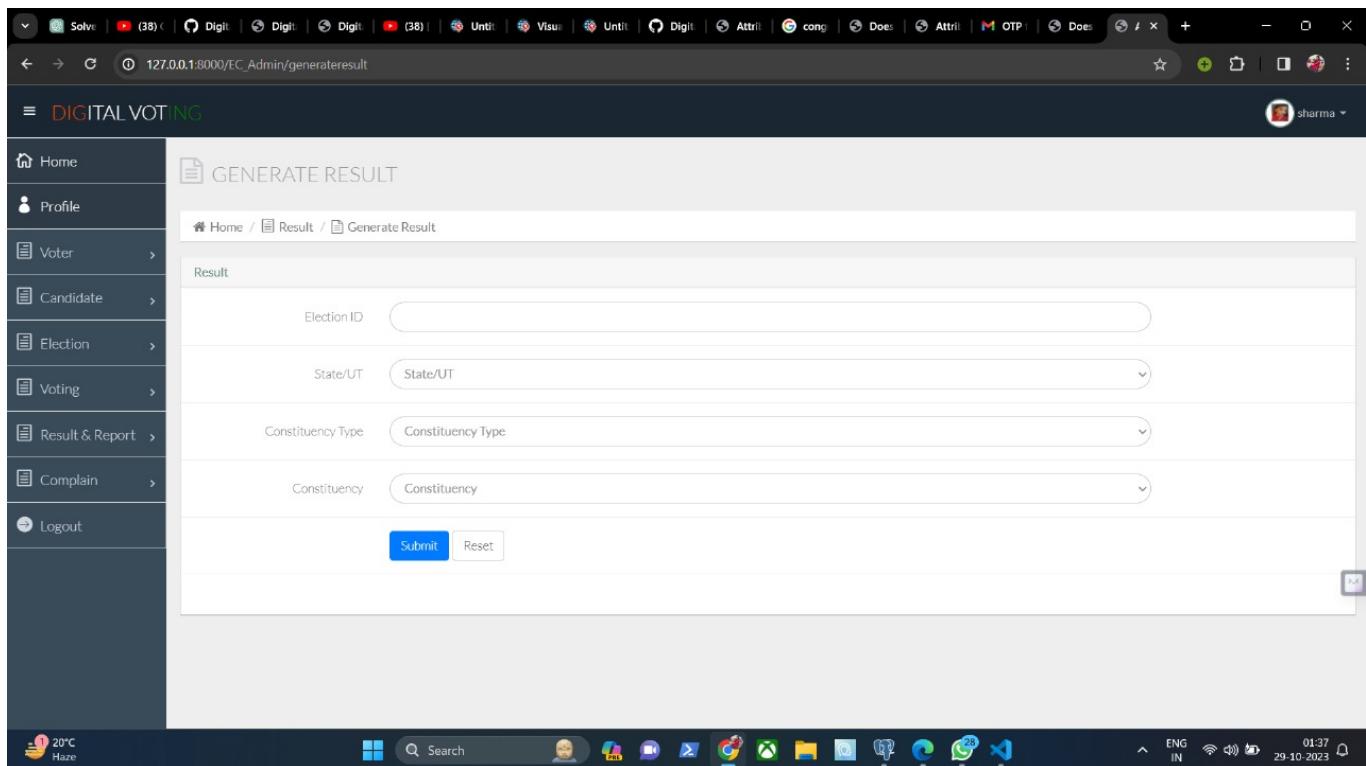


Fig: 4.24 Generate Result(Admin)

## Description

### Administrator (Admin):

The administrator, often referred to as the admin, is a key user role in the online voting system. The admin possesses privileged access to various administrative functions crucial for the smooth operation of the electoral process. Their primary responsibilities include:

- Add Voter: The admin can add new voters to the system, allowing eligible individuals to register for upcoming elections. This functionality is essential for maintaining an updated voter database.
- Add Candidate: Admins can add candidate profiles to the system, ensuring that individuals running for elected positions are correctly represented in the electoral process.
- Generate Elections: The admin has the authority to initiate new election events. This includes defining election parameters, eligible voters, and candidates participating in the election.

- View Results: After an election concludes, the admin can access and view the election results. This function provides transparency and accountability in the electoral process.

## Voter

Voters are the primary users of the online voting system, and their participation is fundamental to the democratic process. They have several key responsibilities and functions within the system:

- Register to Vote: Voters can register themselves to participate in upcoming elections. This process involves providing their essential identification and contact information to establish their voter profile.
- Cast Vote: Registered voters can cast their votes for candidates and ballot measures during active elections. This action is crucial for selecting representatives and determining the outcomes of various issues.
- View Candidate Profiles: Voters have the privilege of viewing detailed profiles of candidates running for elected positions. This allows them to make informed decisions when casting their votes.
- Add Complaint: In the event of issues or concerns related to the voting process, voters can submit complaints or grievances through the system. This feature ensures transparency and accountability.

# **Chapter – 5**

## **Conclusion and Future changes**

In conclusion, this project has been a significant learning experience, resulting in several notable achievements, recognizing its limitations, and opening the door to future expansions.

One of the most prominent achievements of this project is the substantial progress in addressing key issues related to the electoral process. The project has greatly enhanced accessibility by allowing voters to participate remotely through the online platform. This development is a significant step towards ensuring that all citizens, including those with mobility challenges and those in remote areas, can exercise their right to vote.

Security has been a top priority throughout the project. By implementing strong security measures, such as secure login methods, data encryption, and user verification, the project enhances the integrity of the voting process. This, in turn, helps prevent fraud and unauthorized access, which are critical concerns in any voting system.

Efficient data management is another commendable achievement of this project. The system's comprehensive database management capabilities ensure the secure storage and retrieval of voter data, candidate information, and election results. This efficient data handling is crucial for a smooth and reliable electoral process.

Usability, which is often overlooked in the development of voting systems, has been a significant focus. The project has designed a user-friendly interface, making the voting process intuitive and accessible to all users. This prioritization of usability enhances the overall voting experience.

Transparency, a cornerstone of fair elections, is also a fundamental aspect of this project. The implementation of mechanisms for monitoring and auditing votes contributes to building trust in the electoral process.

However, it is important to acknowledge the project's limitations. One notable limitation is the absence of a two-factor authentication method, such as one-time passwords, which could provide an additional layer of security. Addressing this limitation is essential to further enhance the system's security.

Looking to the future, there are exciting opportunities for expansion. The project can explore additional security features, including face recognition, to bolster the system's security. The adoption of more advanced authentication methods, such as two-factor authentication, can be a focus area to mitigate potential risks.

In summary, this project has successfully delivered an online voting system that extends accessibility, ensures security, facilitates efficient data management, enhances usability, and promotes transparency in the electoral process. While acknowledging its limitations, the project's achievements set the stage for further improvements and enhancements, making online voting a secure and accessible option for all citizens.

# **REFERENCES**

- [1] Meelis Kitsing. Online Participation in Estonia: Active Voting, Low Engagement, 2011.
- [2] Micha Germann, Flurin Conradin, Christoph Wellig, Uwe Serdül. Five years of internet voting for Swiss expatriates, 2014.
- [3] Joseph .D. Enoch, Nne .R. Saturday. Biometric online voting system in Nigeria, 2017
- [4] "National Institute of Standards and Technology (NIST) - Guidelines for Secure Online Voting." <https://www.nist.gov/cybersecurity> (accessed on September 28, 2022).

# **Appendix**

## **Glossary**

1. Access Control - The process of restricting or granting access to certain parts of the system based on user roles and permissions.
2. Active Election - An election that is currently in progress and available for voters to participate in.
3. Admin - Administrator: A user with privileged access to the system, responsible for managing and controlling the online voting process.
4. Administrator Privileges - The specific actions and permissions granted to administrators for managing the system.
5. Audit - The process of examining and verifying the accuracy and reliability of data.
6. Authentication - The process of verifying the identity of a user or system.
7. Candidate - An individual running for an elected position in an election.
8. EVM- Electronic Voting Machine: A device used for recording and counting votes electronically.
9. Election- The process by which registered voters choose a candidate for an elected position or vote on specific issues.

10. Error Message - A notification or prompt displayed by the system when an error or issue is encountered.
11. Face Recognition - A biometric technology that identifies or verifies individuals by analyzing and comparing their facial features.
12. High-Priority - Features or tasks that are considered critical and must be addressed with top priority.
13. Inactive Election - An election that has concluded or been temporarily suspended, and voters cannot participate in it.
14. Low-Priority - Features or tasks that are considered less critical and can be addressed after high and medium-priority items.
15. Medium-Priority - Features or tasks that are important but not as critical as high-priority items.
16. OTP - One-Time Password: A temporary password used for authentication, typically sent to the user's mobile phone or email.
17. Secure Logout - The process of logging out from an account while ensuring that the user's session and data are protected from unauthorized access.
18. SRS- Software Requirements Specification: A document outlining the detailed description of the software system's features, functionalities, and constraints.
19. User Profile - Information about a user, including their identification details, preferences, and other relevant data.
20. Voter - A registered user who is eligible to participate in an election.