ANUDIP FOUNDATION

A Project Report on

"ONLINE VOTING SYSTEM USING JDBC AND MYSQL"

Submitted By,

SONU NISAR R N	AF0460130
SUKUMAR K	AF0460124

Under the Guidance of

Ms. VANISHA NAGESH

2024-25

CHAPTER 1

INTRODUCTION

An **Online Voting System** is a digital platform designed to conduct elections efficiently and securely. Traditional voting methods involve paper ballots, manual counting, and physical polling stations, which can lead to logistical challenges, human errors, and delays in result declaration. An online voting system eliminates these issues by allowing voters to cast their votes remotely through a secure digital platform. The system ensures accuracy, transparency, and reliability while reducing election-related costs. This project is developed using **Java** for backend processing and **MySQL** for database management. Java enables secure authentication, real-time vote validation, and backend logic implementation, while MySQL efficiently stores voter details, election data, and vote records. By integrating these technologies, the system ensures seamless data processing and enhances security, preventing fraud or unauthorized access. The primary goal of this system is to provide a **tamper-proof, user-friendly, and accessible** voting platform that can be used for government elections, corporate decision-making, university elections, and other voting processes.

1.1 Existing System

The traditional voting system is largely paper-based, requiring voters to visit designated polling stations to cast their votes. This method has several limitations, including:

- **Time Consumption**: Voters must wait in long queues, and the manual counting process takes several hours or even days.
- **High Costs**: Paper ballots, polling stations, and election staff require significant financial and human resources.
- **Security Risks**: Physical ballots can be tampered with, lost, or damaged, leading to possible vote fraud.
- **Accessibility Issues**: Voters who are physically disabled, elderly, or living far from polling stations may face difficulties in participating.
- **Human Errors**: Manual vote counting is prone to mistakes, affecting election accuracy and fairness.

1.2 Proposed System

The **Online Voting System** addresses the drawbacks of the traditional voting method by providing a secure, digital platform for casting votes. The system allows voters to register, log in, and cast their votes online, ensuring a fast and seamless voting process. This platform is developed using **Java for backend logic and MySQL for database management**, ensuring efficient data processing and secure storage of election-related information.

Key features of the proposed system include:

- **User Authentication**: Secure login mechanisms (e.g., username-password, OTP, or biometric verification) ensure that only registered voters can access the system.
- **Remote Voting**: Voters can cast their votes from anywhere, eliminating the need for physical presence at polling stations.
- **Real-Time Vote Counting**: The system automatically counts votes as they are submitted, reducing errors and speeding up result declaration.
- **Data Security and Encryption**: Advanced encryption techniques ensure that votes cannot be altered or tampered with.
- **User-Friendly Interface**: A simple and intuitive design allows voters to navigate the system with ease.

1.3 Objectives

The main objectives of the Online Voting System are:

1. **To Ensure Security and Integrity**: Implement encryption and authentication mechanisms to prevent vote tampering, duplication, and unauthorized access.
2. **To Improve Accessibility**: Enable voters to cast their votes remotely, increasing participation from individuals who face difficulties in reaching polling stations.
3. **To Minimize Election Costs**: Reduce expenses associated with printing ballots, hiring election staff, and setting up physical polling booths.
4. **To Provide Real-Time Results**: Automate the vote-counting process to deliver immediate and accurate election outcomes.
5. **To Enhance Transparency and Trust**: Maintain audit logs and implement security measures to ensure election fairness and prevent fraud.
6. **To Simplify the Voting Process**: Offer a user-friendly interface that allows voters to register, log in, and vote without technical difficulties.

CHAPTER 2

SYSTEM SPECIFICATION

2.1 Hardware Requirements

- **Processor**: Intel Core i5 or higher (or equivalent AMD processor)
- **RAM**: Minimum 8 GB (16 GB recommended for better performance)
- **Storage**: At least 100 GB SSD (to store user data, vote records, and logs efficiently)
- **Internet Connection**: High-speed broadband with secure network protocols
- **Operating System**: Windows Server, Linux (Ubuntu/CentOS)

2.2 Software Requirements

Backend Development

- **Programming Language**: Java (JDK 8 or later)
- **Database Management System**: MySQL (Version 5.7 or later)
- **JDBC Connector**: MySQL Connector/J for Java-MySQL database integration

Development Tools & IDEs

- **IDE**: Eclipse, IntelliJ IDEA, or NetBeans
- **Version Control**: Git/GitHub for collaborative development and source code management
- **Database Management Tools**: MySQL Workbench or phpMyAdmin for database administration

CHAPTER 3

SYSTEM DESIGN

3.1 Entity-Relationship (ER) Diagram

The ER diagram of the Online Voting System represents the structured flow of data and relationships between different entities within the system. This system is designed to allow users to register, participate in elections, and cast their votes, while the admin manages the election process and candidates.

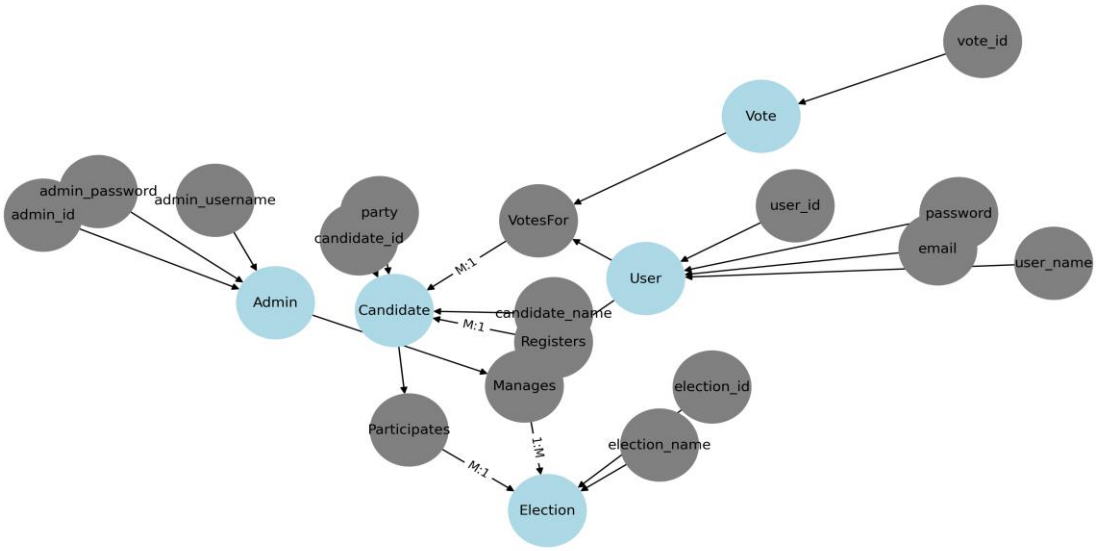

fig:3.1 Entity-Relationship (ER) Diagram

3.2 Entities and Their Roles

1. User

The **User** entity represents the voters who can register and vote in elections. Each user must have a unique identity, ensuring that only registered individuals can participate in the voting process. Users must authenticate themselves before casting a vote.

- **Attributes:**
  - user_id → A unique identifier assigned to each user.
  - user_name → The name of the voter.
  - email → The email ID used for authentication and communication.
  - password → A secured password for login credentials.

- **Role in the System:**
    - Users can **register** in the system.
    - Users can **cast their vote** for a candidate in an election.
    - Users cannot vote more than once in the same election.

2. Admin

The **Admin** entity is responsible for managing the election process, including setting up elections, adding candidates, and ensuring a fair voting process. Only an authorized admin can create and modify election details.

- **Attributes:**
    - admin_id → A unique identifier assigned to each admin.
    - admin_username → The username for logging into the system.
    - admin_password → A secured password for authentication.
- **Role in the System:**
    - Admins **create and manage elections**.
    - Admins **register candidates** for elections.
    - Admins can **view voting results**.

3. Election

The **Election** entity represents the various elections conducted in the system. Each election has a unique name and ID, allowing users to vote for different candidates in separate elections.

- **Attributes:**
    - election_id → A unique identifier for each election.
    - election_name → The name of the election (e.g., "Presidential Election 2025").
- **Role in the System:**
    - Admins can **create multiple elections**.
    - Each election can have **multiple candidates**.
    - Users can only vote in **one election at a time**.

4. Candidate

The **Candidate** entity represents individuals participating in elections. Candidates are added to elections by the admin, and users vote for them during the election process.

- **Attributes:**
    - candidate_id → A unique identifier for each candidate.
    - candidate_name → The name of the candidate.
    - party → The political party or group affiliation (if applicable).
- **Role in the System:**
    - Candidates **compete in elections**.
    - Each candidate is **linked to a specific election**.
    - Users can **vote for one candidate per election**.

5. Vote

The **Vote** entity records all the votes cast in the system. Each vote is linked to a user, an election, and a candidate.

- **Attributes:**
    - vote_id → A unique identifier for each vote.
- **Role in the System:**
    - Each user can cast **only one vote per election**.
    - The system **ensures no duplicate votes**.
    - The votes are **counted to determine the winner**.

Relationships Between Entities

1. **User Registers** → Every voter must first register and authenticate before casting a vote.
2. **Admin Manages Elections** → The admin creates and organizes elections, adding candidates for each election.
3. **Candidates Participate in Elections** → Each candidate is associated with a particular election.
4. **Users Vote for Candidates** → Each user can vote for one candidate per election, and the system stores their selection in the database.

CHAPTER 4

IMPLEMENTATION

Step 1: Start

- Display a Welcome Message to inform users that they have accessed the Online Voting System.
- Show the Main Menu with the following options:

  1. Register (For new users to create an account)
  2. Login (For registered users to log in and vote)
  3. Admin Login (For administrators to manage elections)
  4. Exit (To close the application)

Step 2: User Registration

- If the user selects Register:
  - Prompt the user to enter Name, Email, and Password.
  - Check if the email already exists in the database:
  - If the email is already registered, display "Email already exists!" and return to the main menu.
  - If the email is not found, insert the new user's details into the database.
  - If registration is successful, display "Registration Successful!".
  - If there’s an error, display "Registration Failed!" and return to the main menu.

Step 3: User Login

- If the user selects Login:
  1. Prompt the user to enter Email and Password.
  2. Validate the credentials by checking them against the database.
  3. If credentials are valid, proceed to the Voting Process.
  4. If credentials are invalid, display "Invalid Credentials!" and return to the main menu.

Step 4: Display Available Elections

- Fetch all active elections from the database and display them.
- Prompt the user to select an Election ID to participate in.

Step 5: Display Candidates & Cast Vote

- Fetch and display all candidates participating in the selected election.
- Prompt the user to enter a Candidate ID to cast their vote.
- Check if the user has already voted in the election:
    1. If yes, display "You have already voted!".
    2. If no, insert the vote into the database.
- Display "Vote Cast Successfully!" to confirm the vote.

Step 6: Admin Login

If the user selects Admin Login:

- Prompt the Admin to enter their Username and Password.
- Validate the admin credentials.
- If credentials are valid, show the Admin Dashboard with the following options:
    - Add Election
    - Add Candidate
    - View Results
    - Exit

Step 7: Admin Operations

Add Election:

- Prompt the admin to enter Election Name.
- Insert the new election into the database.
- Display "Election Added Successfully!".

Add Candidate:

- Prompt the admin to enter the Candidate's Name and Party.
- Insert the candidate's details into the database.
- Display "Candidate Added Successfully!".

View Results:

- Fetch and display the total votes per candidate for each election.

Step 8: Exit

If the user selects Exit, terminate the program and close the application.

CHAPTER 5

RESULTS AND DISCUSSION

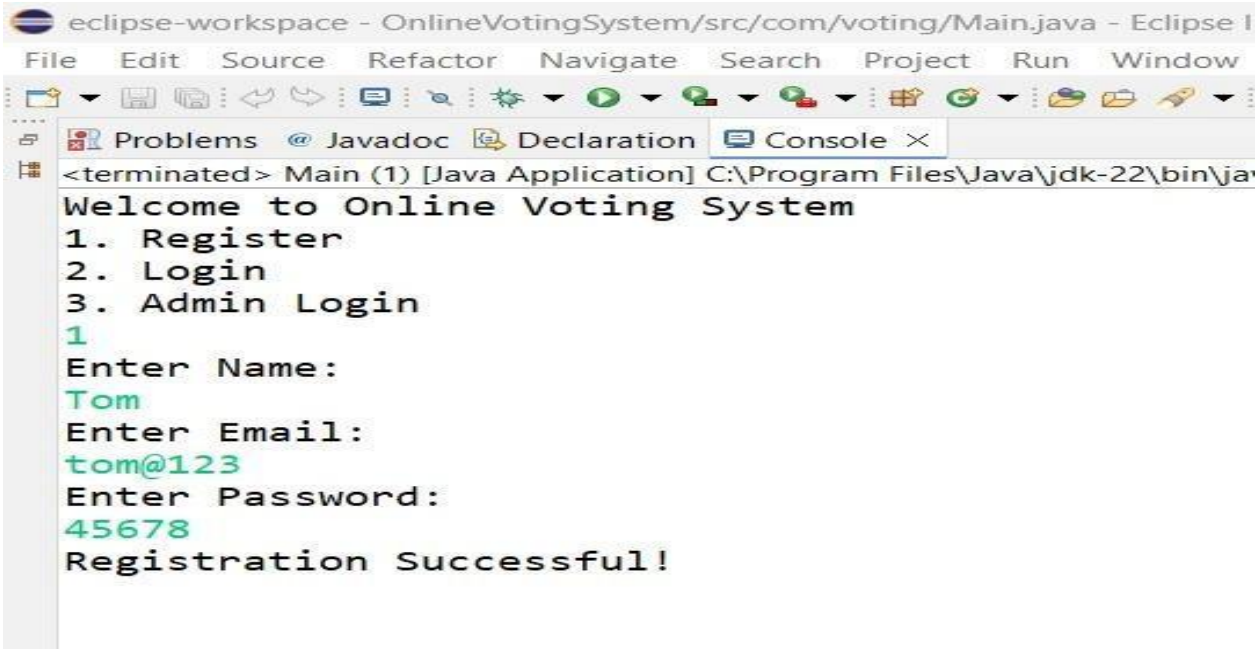

Fig 5.1: User Register.

In the figure 5.1 shows user registration, user need to enter valid name and email and as well password

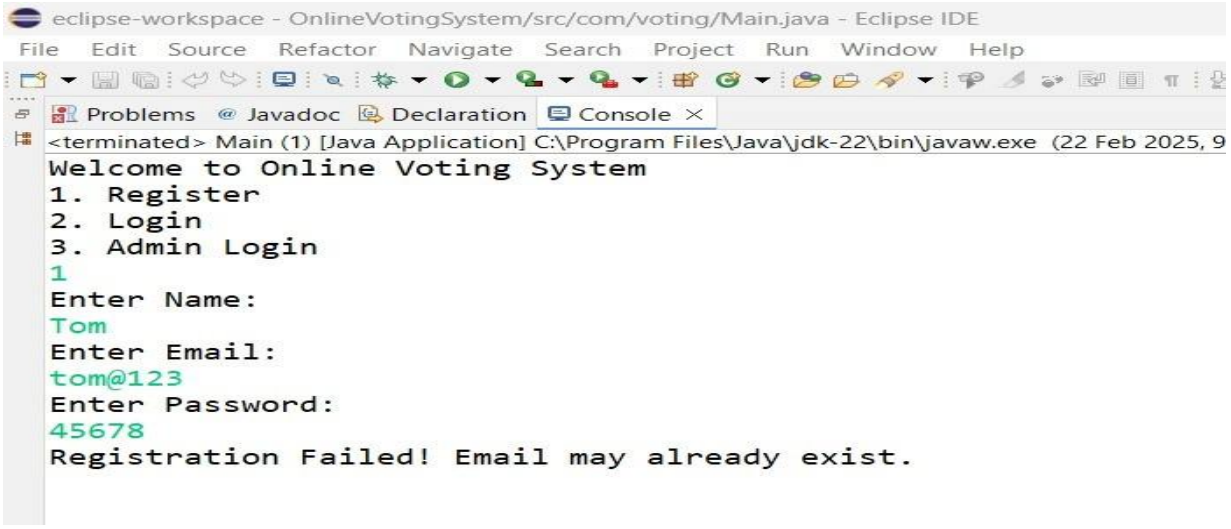

Fig 5.2: User Register Failed.

In the figure 5.2 shows user registration fail, in this project only one user can able to add a vote if user try to re-attempt to add votes it's showing Registration fail.

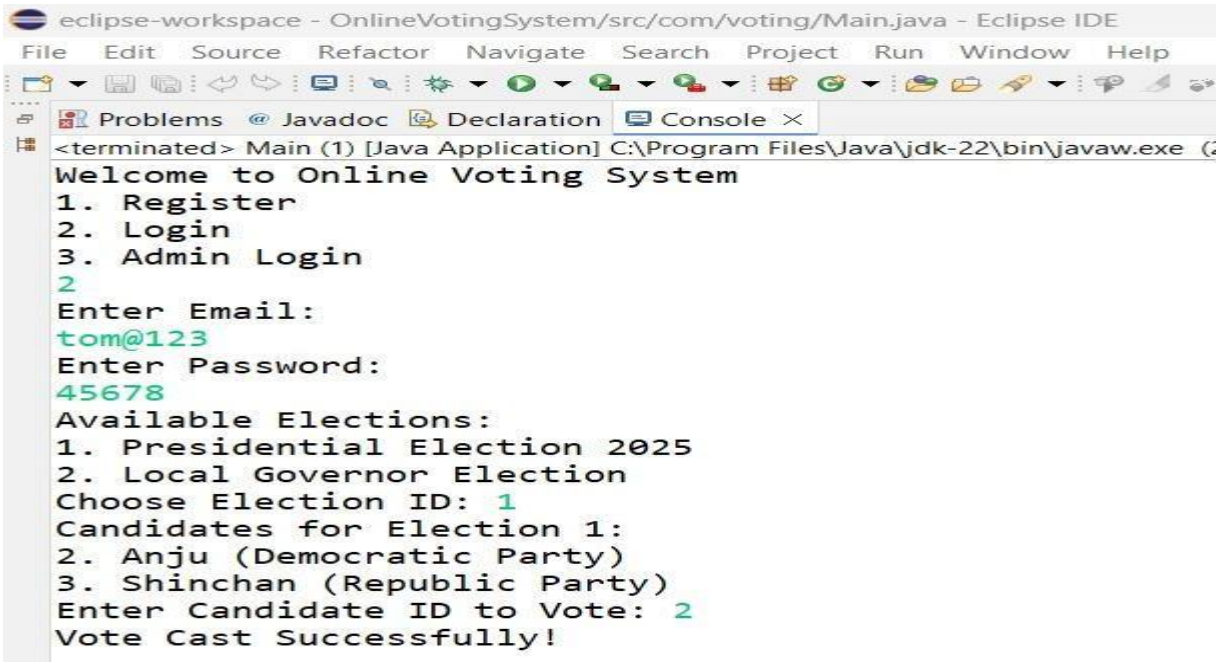

Fig 5.3: Types of Election And Voting.

In the figure 5.2 shows the types of elections, and voting here we see two different elections and candidate.

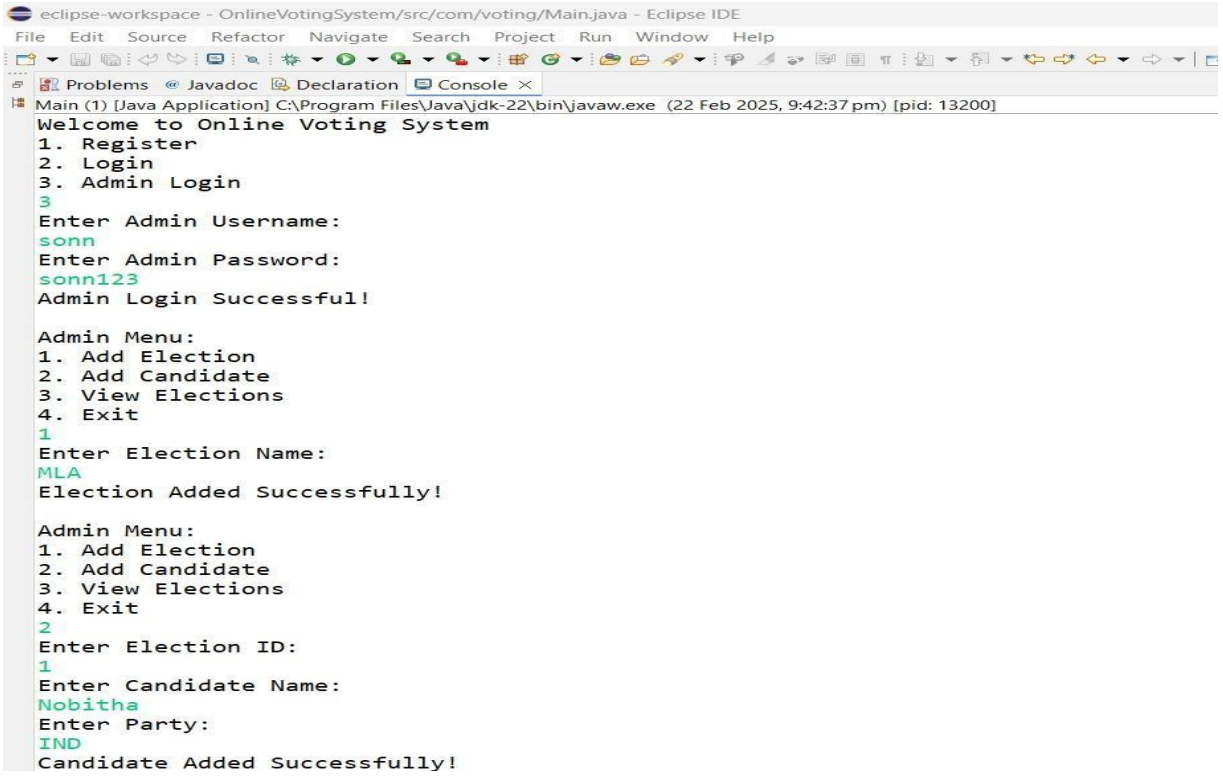

Fig 5.4: Admin Login.

```
mysql> SELECT * FROM admin;
+----+----------+----------+
| id | username | password |
+----+----------+----------+
|  1 | sonn     | sonn123  |
+----+----------+----------+
1 row in set (0.00 sec)

mysql> SELECT * FROM candidates;
+----+----------+-----------------------+-------------+
| id | name     | party                 | election_id |
+----+----------+-----------------------+-------------+
|  2 | Anju     | Democratic Party      |           1 |
|  3 | Shinchan | Republic Party        |           1 |
|  4 | Sukumar  | Liberal Party         |           2 |
|  5 | Virat    | People?s Choice Party |           2 |
|  6 | Nobitha  | IND                   |           1 |
+----+----------+-----------------------+-------------+
5 rows in set (0.00 sec)

mysql> SELECT * FROM elections;
+----+-----------------------------+---------------+----------+
| id | name                        | election_type | status   |
+----+-----------------------------+---------------+----------+
|  1 | Presidential Election 2025  | National      | Ongoing  |
|  2 | Local Governor Election     | State         | Ongoing  |
|  3 | MLA                         | NULL          | NULL     |
+----+-----------------------------+---------------+----------+
3 rows in set (0.00 sec)

mysql>
```

Fig 5.5: Voting Table

In the figure 5.5 shows voting tables, it contains admin, candidates, elections information.

```
mysql> SELECT u.name AS voter_name, c.name AS candidate_name, e.name AS election_name
    -> FROM votes v
    -> JOIN users u ON v.user_id = u.id
    -> JOIN candidates c ON v.candidate_id = c.id
    -> JOIN elections e ON v.election_id = e.id;
+------------+----------------+----------------------------+
| voter_name | candidate_name | election_name              |
+------------+----------------+----------------------------+
| sukruth    | Sukumar        | Local Governor Election    |
| Tom        | Anju           | Presidential Election 2025 |
+------------+----------------+----------------------------+
2 rows in set (0.00 sec)

mysql>
```

Fig 5.6: Lists of Voting's.

In the figure 5.6 shows the lists of voting's, and its contains voters name, candidate name and election names.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

The Online Voting System provides a secure, transparent, and efficient method for conducting elections digitally. By integrating Java and MySQL, this system ensures that user authentication, vote casting, and result computation are handled seamlessly. The use of JDBC for database connectivity allows real-time data reflection, ensuring accurate vote counts and preventing duplication or manipulation. This system significantly reduces the manual effort, paperwork, and time required in traditional elections while increasing accessibility and convenience for voters. The admin panel ensures proper election management, and the secure authentication mechanisms prevent unauthorized access. The project successfully demonstrates how technology can modernize and streamline the voting process while ensuring security and fairness.

The Online Voting System can be improved to make it safer, easier, and more efficient. A frontend interface using HTML, CSS, and JavaScript can be added so voters can access the system on any device. For better security, fingerprint or facial recognition can be used to allow only real voters to vote. Blockchain technology can help make the system secure and tamper-proof. The system can support local, state, and national elections with different levels of access. A mobile app can be developed so people can vote easily from their phones. Voters will receive email and SMS alerts about election dates, vote confirmation, and results. AI-based fraud detection can help stop fake votes. Using cloud platforms like AWS or Google Cloud will make the system faster and more reliable. Adding multiple language support will allow people to vote in their preferred language. With these improvements, the Online Voting System will become secure, easy to use, and accessible to all, making voting simple and trustworthy.

.