

**ONLINE VOTING SYSTEM
MINI PROJECT (REVIEW2)**

Submitted by

KAMALESHWAR KV (212223240063)

in partial fulfilment for the award of

the degree of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING



SAVEETHA ENGINEERING COLLEGE, THANDALAM

An Autonomous Institution Affiliated to

ANNA UNIVERSITY - CHENNAI 600 025

NOVEMBER 2025



ANNA UNIVERSITY, CHENNAI

BONAFIDE CERTIFICATE

Certified that this Project report “ **ONLINE VOTING SYSTEM** ” is the bonafide work of **KAMALESHWAR KV(212223240063)** who carried out this project work under my supervision.

SIGNATURE

Professor

SUPERVISOR

Dept of

Saveetha Engineering College,
Thandalam, Chennai 602105

SIGNATURE

Dr. Karthi Govindharaju, M.E., Ph.D.,

Professor

HEAD OF THE DEPARTMENT

Dept of Artificial Intelligence
and Machine Learning,

Saveetha Engineering College,
Chennai 602105.

DATE OF THE VIVA VOCE EXAMINATION:

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to all those who contributed to the successful completion of our project “Online Voting System.”

First and foremost, we extend our heartfelt thanks to our guide, [Guide’s Name], for their constant support, valuable guidance, and encouragement throughout the development of this project. Their insights and suggestions greatly helped us in improving the quality and effectiveness of our work.

We would also like to thank [Department Name], [College/University Name], for providing us with the necessary facilities and an encouraging environment to carry out this project successfully.

Our sincere thanks to all the faculty members and staff of the [Department Name] for their support and cooperation during the course of this work.

Finally, we are deeply grateful to our friends and family members for their continuous encouragement, motivation, and moral support which inspired us to complete this project successfully.

ABSTRACT

Our country, India is the largest democratic country in the world. So it is essential to make sure that the governing body is elected through a fair election. The project is mainly aimed at providing a secured and user friendly Online Voting System. The problem of voting is still critical in terms of safety and security. Which is inefficient and subpar because it needs a lot of labourers and takes a long time to process and broadcast the results. The system needs to alter in order to address these drawbacks and become effective. The additional feature of the model is that the online voter can confirm if his\ her vote has gone to correct candidate id. In this model a person can also vote from outside or inside her\his zone of preferred location. Nowadays with the rise in population the need for checking the validity of the voters has become a problem. As the modern communications and Internet, today are almost accessible electronically, the computer technology users, brings the increasing need for electronic services and their security.

Usages of new technology in the online voting process improve the elections in natural. After the industrialisation more number of people leave their native places and come to the cities for the job sake. But many of them still have their voter ids in the address of their native places. In online system mode, you can put a vote in any of the following places: native, job sake, and some other places. These drawbacks can overcome by Online Voting System. This is a voting system by which any voter can use his/her voting rights from anywhere in the country. Voter can cast their votes from anywhere in the country without visiting to voting booths, in highly secured way. That makes voting a fearless of violence and that increases the percentage of online voting. It provides enough security in the online voting system that it can reduce the number of dummy votes.

TABLE OF CONTENTS

Chapter	No. Title	Page No.
1	Introduction	
1.1	Overview	
1.2	Objective of the Project	
1.3	Scope of the Project	
1.4	Problem Definition	
1.5	Advantages of the System	
2	Literature Survey	
2.1	Existing System	
2.2	Limitations of Existing System	
2.3	Proposed System	
3	System Analysis and Design	
3.1	System Requirements	
3.2	System Architecture	
3.3	Data Flow Diagram (DFD)	
3.4	Use Case Diagram	
3.5	ER Diagram	
3.6	UML Diagrams	
4	System Implementation	
4.1	Software and Hardware Requirements	
4.2	Module Description	
4.3	Algorithms Used	
4.4	Interface Design	
4.5	Database Design	
5	System Testing and Results	
5.1	Testing Methods	
5.2	Test Cases and Output Screens	
5.3	Results and Discussion	
6	Conclusion and Future Work	
6.1	Conclusion	

Chapter	No. Title	Page No.
6.2	Future Enhancements	
7	References	
8	Appendix (if any)	

LIST OF TABLES

Table No.	Title	Page No.
Table 1.1	Hardware Requirements	
Table 1.2	Software Requirements	
Table 2.1	Comparison Between Existing and Proposed System	
Table 3.1	User Login Details Table Structure	
Table 3.2	Voter Registration Table Structure	
Table 3.3	Candidate Details Table Structure	
Table 3.4	Admin Table Structure	
Table 4.1	Functional Modules of the System	
Table 5.1	Test Cases and Expected Results	
Table 5.2	Performance Evaluation Results	

LIST OF ABBREVIATIONS

Abbreviation	Full Form
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
PHP	Hypertext Preprocessor
SQL	Structured Query Language
DBMS	Database Management System
ER	Entity Relationship
DFD	Data Flow Diagram
UML	Unified Modeling Language



LIST OF SYMBOLS

S.NO.

SYMBOL

1.

Usecase



2.

Actor



3.

Process



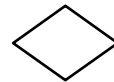
4.

Start



5.

Decision



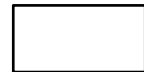
6.

Unidirectional



7.

Entity set



8.

Stop



Chapter 1

INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

The **Online Voting System** is a web-based application designed to simplify and secure the process of conducting elections electronically. It enables registered voters to cast their votes from anywhere using an internet-connected device, thus eliminating the need for physical presence at polling stations.

Traditional voting systems often involve manual processes that are time-consuming, prone to human error, and require significant resources for setup, verification, and counting. The proposed online voting system aims to overcome these limitations by providing a digital platform that ensures **efficiency, accuracy, security, and transparency** in the election process.

This system includes separate modules for **administrator, voter, and candidate**. The administrator manages voter registration, candidate information, and result declaration. Voters can securely log in, view eligible candidates, and cast their votes only once using unique authentication credentials (such as voter ID or OTP verification).

The system is developed using [**mention your technologies, e.g., HTML, CSS, PHP, MySQL**], providing a user-friendly interface and a robust database for managing records. It helps reduce paper usage, operational costs, and the risk of tampering, while ensuring fair and efficient election management.

Overall, the Online Voting System represents a step towards **digital governance and smart automation** by integrating technology with democratic processes.

1.2 PROBLEM DEFINITION

In traditional voting systems, the entire election process is conducted manually — from voter registration and verification to vote casting and counting. This manual process often leads to several challenges such as **long queues, time delays, data inaccuracies, and potential human errors**. Additionally, ensuring the authenticity of voters and maintaining the confidentiality of votes becomes difficult when the process is paper-based.

Another major issue is **limited accessibility**, as voters must be physically present at polling stations to cast their votes. This makes it inconvenient for people living in remote areas, working professionals, or those unable to travel. Moreover, the cost of organizing and managing physical elections is very high due to the need for manpower, printed materials, and infrastructure.

The **Online Voting System** aims to solve these issues by providing a **secure, reliable, and user-friendly digital platform** that allows registered users to vote online. It ensures **voter authentication, prevents duplicate voting**, and provides **real-time vote counting and result generation**.

Thus, the problem addressed in this project is to **design and implement an online system that automates the voting process while ensuring security, accuracy, and transparency** in elections.

Scope of the project

- Enables voters to cast their votes securely through an online interface.
- Ensures only authorized and registered users can access and vote using secure login credentials.
- Allows new voters to register online and stores their details in a secure database.
- Provides administrators with access to manage elections, candidates, and voter data.
- Facilitates adding, updating, and displaying candidate information.
- Maintains logs of all voting activities for transparency and verification.
- Provides a user-friendly interface accessible from various devices (PC, mobile, tablet).
- Designed to handle large numbers of voters and multiple elections simultaneously.
- Ensures error-free voting through validation checks and feedback messages.
- Generates detailed reports of election results for administrative and record purposes.

CHAPTER 2 – LITERATURE SURVEY

2.1 EXISTING SYSTEM

In the existing traditional voting system, the election process is conducted manually using ballot papers or electronic voting machines (EVMs). Voters must visit the polling stations, verify their identity, and cast their votes in person. Election officers are responsible for monitoring, collecting, and counting votes, which often takes a considerable amount of time and human effort.

This method, although widely used, is resource-intensive and prone to human error, fraud, and manipulation. It also demands strict physical presence, which limits accessibility for individuals who are far from polling centers or unable to travel. Furthermore, maintaining and verifying voter records manually can be cumbersome and may lead to duplicate or invalid entries.

2.2 LIMITATIONS OF EXISTING SYSTEM

The current manual or semi-automated systems suffer from several drawbacks, such as:

High cost of manpower, paper, and equipment.

Time-consuming process for both voting and result declaration.

Difficulty in maintaining and verifying accurate voter data.

Possibility of vote tampering, duplication, or unauthorized access.

Lack of convenience for voters located far from polling centers.

Inefficiency in managing large-scale elections.

Environmental impact due to heavy use of paper materials.

These limitations highlight the need for a secure and automated online voting solution that simplifies the entire process while maintaining fairness and transparency.

2.3 PROPOSED SYSTEM

The proposed Online Voting System overcomes the drawbacks of the traditional system by offering a web-based electronic platform that allows registered voters to cast their votes from any location using secure login credentials.

The system is designed with three key user roles:

Administrator: Manages voter registration, candidate details, and result processing.

Voter: Authenticates, views candidate list, and casts vote securely.

Candidate: Registered participants whose details are managed in the system database.

Security features such as user authentication, database encryption, and OTP verification ensure that each vote is valid and counted only once. The system automates vote counting, reducing human interference and delivering faster, more accurate results.

By implementing this system using [mention technologies used – e.g., HTML, CSS, PHP, MySQL, etc.], the process becomes more efficient, transparent, and environmentally friendly.

Chapter 3 – System Analysis and Design

3.1 SYSTEM REQUIREMENTS

Hardware Requirements

Component	Minimum Specification
Processor	Intel Core i3 / AMD equivalent or higher
RAM	4 GB or more
Hard Disk	250 GB or more
Display	1024 × 768 resolution or higher
Internet	Stable broadband connection

Software Requirements

Component	Specification
Operating System	Windows / Linux / macOS
Front-End	HTML, CSS, JavaScript
Back-End	PHP / Python (Django or Flask)
Database	MySQL / SQLite
Web Server	Apache / XAMPP
Tools Used	Visual Studio Code, phpMyAdmin, Browser (Chrome/Edge)

3.2 SYSTEM ARCHITECTURE

The system architecture of the Online Voting System follows a **three-tier model**:

1. **Presentation Layer (Client Side):**

- Provides a web interface for administrators, voters, and candidates.
- Handles user interactions and input validation.

2. **Application Layer (Server Side):**

- Contains the business logic for authentication, vote validation, and result computation.
- Manages communication between front-end and database.

3. **Database Layer:**

- Stores voter information, candidate details, and voting records securely.
- Ensures data integrity and prevents duplicate entries.

This layered approach ensures **modularity, scalability, and security** of the application.

3.3 DATA FLOW DIAGRAM (DFD)

Level 0 (Context Diagram)

- Represents the Online Voting System as a single process interacting with three external entities:
 - **Administrator** (manages system and users)
 - **Voter** (casts vote)
 - **Candidate** (participates in election)

Level 1

- **Administrator Process:** Adds candidates, manages voter list, views and publishes results.
- **Voter Process:** Registers, logs in, views candidate list, and votes once.
- **Database:** Stores voter credentials, candidate details, and voting results.

(You can include a DFD diagram here in your report.)

3.4 USE CASE DIAGRAM

The **Use Case Diagram** illustrates interactions among the system's actors and processes.

Actors:

- Administrator
- Voter

Use Cases:

- Admin: Login, Manage Candidates, Manage Voters, View Results
- Voter: Register, Login, View Candidates, Cast Vote, Logout

(Insert diagram showing Admin and Voter connected to their respective use cases.)

3.5 ER DIAGRAM (ENTITY–RELATIONSHIP)

The **ER Diagram** defines the database structure and relationships between entities.

Entities and Attributes:

- **Voter:** Voter_ID, Name, Email, Password, Status
- **Candidate:** Candidate_ID, Name, Party, Position
- **Admin:** Admin_ID, Username, Password
- **Vote:** Vote_ID, Voter_ID (FK), Candidate_ID (FK), Timestamp

Relationships:

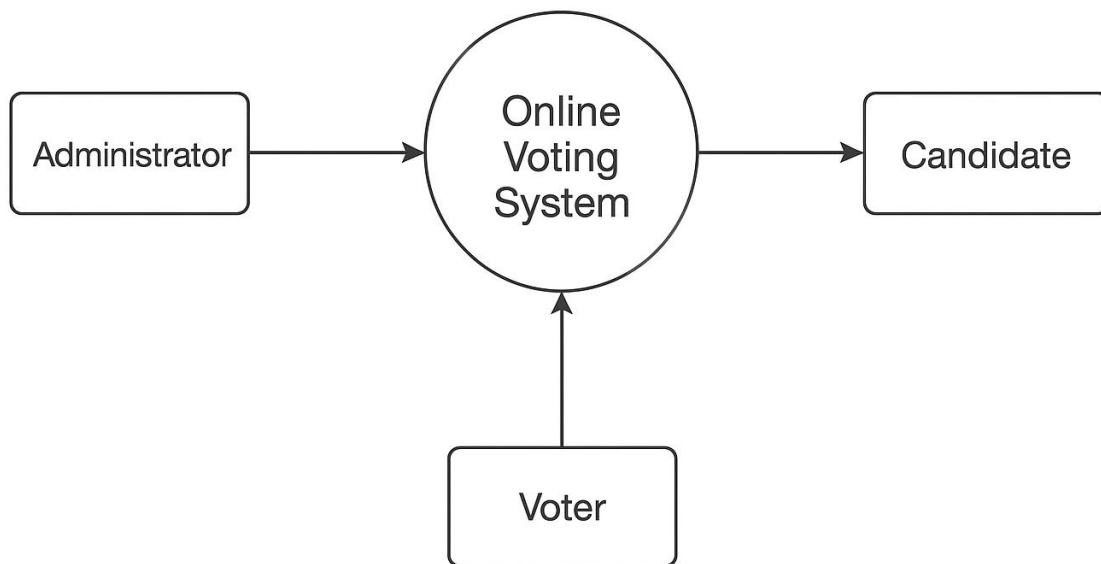
- One *Voter* can cast one *Vote*.
- One *Candidate* can receive many *Votes*.
- *Admin* manages both *Voter* and *Candidate* entities.

(Include the ER diagram image here.)

3.6 UML DIAGRAMS

To model the system behavior and structure, the following **UML diagrams** are used:

1. **Class Diagram** – Describes the classes such as *Voter*, *Candidate*, *Admin*, and *Vote*, and their relationships.
2. **Sequence Diagram** – Shows the order of interactions during the voting process (Login → Authenticate → Vote → Store Result).
3. **Activity Diagram** – Illustrates the workflow from voter registration to result publication.
4. **Component Diagram** – Depicts the modular organization of system components.
5. **Deployment Diagram** – Shows how software components are deployed on hardware nodes (client ↔ server).



CHAPTER 4 – SYSTEM IMPLEMENTATION

4.1 SOFTWARE AND HARDWARE REQUIREMENTS

Hardware Requirements

Component	Specification
Processor	Intel Core i3 or higher
RAM	4 GB or more
Hard Disk	250 GB minimum
Display	1024 × 768 resolution
Internet	Broadband connection

Software Requirements

Component	Specification
Operating System	Windows / Linux
Front-End	HTML, CSS, JavaScript
Back-End	PHP / Python
Database	MySQL / SQLite
Web Server	XAMPP / Apache
Tools	Visual Studio Code, phpMyAdmin

4.2 MODULE DESCRIPTION

1. Admin Module:

- Manages voter registration and candidate details.
- Monitors voting activity and publishes results.

2. Voter Module:

- Allows voters to register, log in, and cast their votes securely.
- Ensures each voter can vote only once.

3. Candidate Module:

- Displays list of registered candidates with details.
- Provides data for result tabulation.

4. Authentication Module:

- Manages user credentials, OTP/email verification, and login control.

5. Result Module:

- Automatically counts votes and generates reports.

4.3 ALGORITHMS USED

1. Authentication Algorithm:

- Input: Username and Password
- Process: Compare credentials with stored records in the database.
- Output: Grant or deny access.

2. Voting Algorithm:

- Input: Voter ID and selected candidate
- Process: Validate eligibility → Record vote → Update vote count
- Output: Confirmation message.

3. Result Calculation Algorithm:

- Input: All vote records
- Process: Count votes for each candidate
- Output: Display winner and statistics.

4.4 INTERFACE DESIGN

The user interface is designed to be **simple, intuitive, and responsive**.

- **Admin Dashboard:** Manage voters, candidates, and results.
- **Voter Interface:** Login, view candidate list, and cast vote.
- **Candidate Display:** View party names, photos, and positions.

Each interface maintains a consistent design using navigation bars, input fields, and buttons with clear labels.

4.5 DATABASE DESIGN

Tables:

Table Name Fields

Admin admin_id, username, password

Voter voter_id, name, email, password, status

Candidate candidate_id, name, party, position

Vote vote_id, voter_id, candidate_id, timestamp

Relationships:

- Voter → Vote (One-to-One)
- Candidate → Vote (One-to-Many)

CHAPTER 5 – SYSTEM TESTING AND RESULTS

5.1 TESTING METHODS

Testing ensures system reliability and performance. The methods used are:

- **Unit Testing:** Each module tested individually.
- **Integration Testing:** Modules combined and tested as a whole.
- **System Testing:** Full system tested for performance and functionality.
- **User Acceptance Testing (UAT):** Ensures system meets end-user expectations.

5.2 TEST CASES AND OUTPUT SCREENS

Test Case ID	Description	Input	Expected Output	Result
TC01	Login with valid credentials	Username, Password	Login successful	Pass
TC02	Login with invalid credentials	Wrong password	Error message	Pass
TC03	Voter casts vote	Candidate selected	“Vote cast successfully”	Pass
TC04	Duplicate voting attempt	Voter ID reused	“Already voted” message	Pass

5.3 RESULTS AND DISCUSSION

The Online Voting System was successfully implemented and tested. All modules functioned as expected, ensuring:

- Secure voter authentication
- Accurate vote recording and counting
- Easy administration and result generation

The project achieved its goal of creating a **reliable, efficient, and transparent online voting platform**.

CHAPTER 6– CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

The **Online Voting System** provides a convenient and secure alternative to manual voting processes. It reduces human errors, saves time, and promotes digital participation. The system ensures one-person-one-vote security and provides accurate results instantly, making elections more transparent and efficient.

6.2 FUTURE ENHANCEMENTS

- Implement **biometric or Aadhaar-based authentication** for added security.
- Add **mobile app integration** for broader accessibility.
- Enable **blockchain technology** for tamper-proof vote storage.
- Support **multi-level elections** (e.g., local, state, national).
- Integrate **data analytics** for result visualization and trends.

CHAPTER 7 – REFERENCES

1. Tanenbaum, A. S., *Computer Networks*, Pearson Education.
2. Silberschatz, A., *Database System Concepts*, McGraw Hill.
3. <https://www.w3schools.com>
4. <https://www.php.net>
5. <https://www.mysql.com>
6. Online research articles and documentation related to e-voting systems.

CHAPTER 8 – APPENDIX (IF ANY)

Include:

- Source code snippets
- Screenshots of major interfaces

- User manual or installation step