



ONLINE VOTING SYSTEM

Ms. Hole¹, Sakshi Shinde²

¹ Faculty Guide & Supervisor | ² Dattakala Group of Institutions, Faculty of Engineering, Pune, Maharashtra, India, India

Corresponding author: sakshushinde7@gmail.com

ARTICLE ID IJREK-2026-00119	ARTICLE TYPE Research Article	RESEARCH AREA Internet of Things (IoT)	PUBLICATION Volume 1, Issue Issue 1 (2026)
RECEIVED 19 October 2026	REVISED 22 October 2026	ACCEPTED 19 October 2026	PUBLISHED 22 October 2026

Abstract

The Online Voting System is a Java-based application designed to conduct elections securely and efficiently over the internet. It reduces reliance on traditional paper-based voting by allowing registered users to cast votes online. The system uses Java with MySQL as the back-end database and JDBC for database connectivity. Its core modules include administrator management, voter registration, secure login, voting, and result generation. Administrators can create elections, manage candidates, monitor votes, and view results. Registered voters authenticate before voting, and the system is designed to restrict each voter to one vote per election. The system aims to improve transparency, reduce manual errors, and speed up result processing. It may be useful for colleges, institutions, societies, and organizations conducting internal elections.

Keywords: Research; Technology

2. PROBLEM STATEMENT AND OBJECTIVES

Manual election processes require considerable coordination and can introduce delays, counting errors, duplicate voting, and difficulties in maintaining records. The problem addressed is how to organize voter registration, secure vote casting, candidate and election management, and result generation through a database-driven application. 2.1 Objectives • To provide a structured voter registration and authentication process. • To allow administrators to create elections and manage candidate information. • To restrict voting so that an eligible voter can cast only one vote per election. • To store and manage election records using a MySQL database. • To automate vote counting and result generation. • To improve the efficiency, accessibility, and transparency of institutional elections. 2.2 Scope The system is intended for colleges, institutions, societies, and organizations conducting internal elections. It includes voter and administrator roles, election and candidate management, vote recording, and result generation. Security and privacy depend on correct implementation, access control, database protection, and testing.

3. RELATED WORK

Existing electronic voting approaches explore web-based and platform-independent systems, mobile voting, cloud-based voting, and secure e-government voting. These approaches highlight the importance of authentication, convenient access, reliable vote recording, and transparent result processing. For an institutional system, a practical design must also prevent repeat voting, protect voter information, and maintain a clear separation between administrative actions and voter actions.

3.1 RESEARCH GAP

A useful implementation needs to combine a simple voting workflow with role-based access, controlled vote submission, secure record handling, and understandable result reporting. The proposed system focuses on these core functions for small-scale institutional elections.

4. PROPOSED METHODOLOGY

The proposed workflow consists of database connection, user authentication, role identification, election and candidate selection, vote submission, vote recording, and result generation. A voter signs in and the system checks whether the voter has

already voted in the selected election. If not, active elections and candidates are displayed; after candidate selection, the vote is recorded and a confirmation is shown. An administrator can manage elections, add or remove candidates, and view results.

4.1 DATA AND DATABASE DESIGN

The database may include tables for voters, candidates, votes, and elections. Relationships between these records should be designed to maintain consistency. Constraints and server-side checks should enforce the one-vote-per-election rule; sensitive data should be protected and access limited to authorized roles.

4.2 VOTING WORKFLOW

TABLE I

Step	Action	Purpose
1	Database connection and login	Connect the application and authenticate the user.
2	Role and eligibility check	Separate voter and administrator actions.
3	Candidate selection	Show active elections and available candidates.
4	Vote recording	Save one valid vote and show confirmation.
5	Result generation	Count recorded votes and display results.

5. SYSTEM ARCHITECTURE AND ACTIVITY FLOW

The system can be organized into a user interface, authentication and role-management layer, voting and election logic, and a MySQL data layer connected through JDBC. The activity flow starts with a database connection and login. Voters proceed through eligibility checks and vote submission; administrators proceed to election management and result viewing. Both roles can exit after completing the relevant actions. Activity flow User Login → Role Check → Voter: Eligibility Check → Select Candidate → Record Vote Administrator: Manage Elections / Candidates → View Results

5.1 FUNCTIONAL MODULES

TABLE II

Module	Purpose
Voter registration and login	Register users and authenticate access.
Administrator management	Create elections and manage candidates.
Voting process	Show active elections and record an eligible voter's selection.
Vote validation	Check eligibility and prevent repeat voting.
Result generation	Count recorded votes and display election results.
Database connectivity	Store and retrieve records through Java and JDBC.

6. TOOLS AND TECHNOLOGIES

- Java: application logic and user interaction.
- MySQL: storage for voter, candidate, vote, and election records.
- JDBC: connectivity between the Java application and MySQL.
- Testing: functional testing of registration, login, vote submission, repeat-vote prevention, and result generation.
- Security review: check access control, input validation, and protection of stored records.

7. ADVANTAGES AND APPLICATIONS

7.1 Advantages

- Reduces manual counting effort and can speed up result processing.
 - Centralizes election, candidate, voter, and vote records.
 - Supports controlled access for voters and administrators.
 - Can reduce duplicate voting when eligibility checks and database constraints are correctly implemented.
 - Allows remote access where the deployment and authentication arrangements support it.
- ### 7.2 Applications
- Student council and college elections.
 - Voting within societies, clubs, and institutions.
 - Internal elections in organizations and other small groups.

8. SECURITY AND LIMITATIONS

An online voting application must protect voter data, restrict administrative access, validate input, and prevent repeated or unauthorized vote submissions. Secure authentication and database permissions are essential. The reliability of results depends on implementation quality, correct database constraints, testing, and operational controls. The design described here does not by itself establish suitability for public, state, or national elections; such use would require substantially stronger independent security, privacy, accessibility, auditability, and

legal review. No measured performance results or security audit are claimed in this paper. Security considerations

- Enforce access control for voter and administrator roles.
- Validate all user input on the server side.
- Protect stored records and restrict database permissions.
- Use constraints and checks to prevent repeated or unauthorized submissions.
- Test the complete workflow and document any known weaknesses.

9. CONCLUSION AND FUTURE SCOPE

9.1 Conclusion

The Online Voting System is a Java-based application concept integrated with a MySQL database through JDBC. It brings together voter registration, candidate and election management, vote casting, and result generation in a structured workflow. By automating routine tasks, it can reduce manual effort and support more efficient institutional elections. Its practical reliability depends on sound access control, vote validation, database integrity, and thorough testing.

9.2 FUTURE SCOPE

Future development may include stronger multi-factor authentication, improved accessibility for different users, audit logs, secure result reports, and a web or mobile interface. Further work should test the implementation with documented test cases and assess privacy, security, usability, and performance before deployment.

REFERENCES

- [1] Dr. Z. A. Usmani, K. Patanwala, M. Panigrahi, and A. Nair, "Multi-Purpose Platform Independent Online Voting System," International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), 2017.
- [2] Cesar R. K. et al., "Web 2.0 E-Voting System Using Android Platform," IEEE International Conference on Progress in Informatics and Computing, pp. 1138–1142, 2010.
- [3] M. H. Sedky and E. M. R. Hamed, "A Secure e-Government's e-Voting System," Science and Information Conference, pp. 1365–1373, 2015.
- [4] R. Govindaraj, P. Kumaresan, and K. Sree Harshitha, "Online Voting System using Cloud," International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), pp. 1–4, 2020. Note: This paper describes a proposed institutional voting system. Practical reliability and security should be established through implementation, testing, and review.

Publication archival note: Official scholarly contribution published under CC BY 4.0 International License. Permanent record on <https://www.ijrek.org> | Article ID: IJREK-2026-00119