International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 2, Mar-Apr-2026, ISSN (Online): 2395-566X

NEXTGEN: College Voting System A Secure Web-Based Institutional Election Management Platform

Kaustubh Nitin Salunke, Vinayak Amol Shewale, Anurag Sanjay Shigwan, Omkar Vinod Tate
Department of Computer Engineering, Zeal Polytechnic, Narhe, Pune – 411041
Under the Guidance of Prof. Suchita Barkund Academic Year: 2025–2026

Abstract- — The escalating demand for transparent, tamper-proof, and efficient electoral processes in educational institutions necessitates a modern digital alternative to conventional paper-based voting. This paper presents NEXTGEN: College Voting System, a secure, fully web-based election management platform designed specifically for college-level institutional elections. The system is architected on a three-tier client-server model employing Java Servlets and JavaServer Pages (JSP) for backend processing, HTML5/CSS3 with Bootstrap 5 for the frontend, MySQL 8.0+ as the relational database engine, Apache Tomcat 11 as the servlet container, and the Jakarta Mail API for OTP-based Two-Factor Authentication (2FA). The platform features two primary role-based modules: an Admin Module offering complete election lifecycle control including student registration management, candidate management, election activation/deactivation/reset, and real-time result monitoring; and a Student Module providing secure registration, OTP-verified login, position-wise vote casting, and OTP-based password recovery. Security is enforced through SHA-256 password hashing, session management, role-based access control, dual-layer duplicate vote prevention (application-layer logic and database UNIQUE constraints), and time-bound OTP verification (5-minute validity). Testing validated 100% vote-count accuracy, 100% duplicate vote rejection, and OTP delivery within 5–10 seconds. The system eliminates manual counting errors, drastically reduces administrative overhead, and enables instant, verifiable election results. Future directions include biometric authentication, blockchain-based vote immutability, SMS-OTP support, and cloud deployment.

Keywords— Online Voting System; Two-Factor Authentication; OTP Verification; Java Servlet; JSP; MySQL; Apache Tomcat; Election Management; Role-Based Access Control; Database Security.

I. INTRODUCTION

Educational institutions across India and many developing nations continue to rely on paper-based ballot systems for student council elections. While familiar, these legacy processes are burdened by voter impersonation, duplicate voting, susceptibility to ballot tampering, laborious manual counting, delayed result announcement, and high administrative overhead. In an era where students submit assignments through digital portals and conduct banking via smartphones, the persistence of paper-based voting is both anachronistic and operationally inefficient.

The proliferation of web technologies, cloud infrastructure, and secure email communication channels has created an opportunity to completely reimagine institutional elections. A properly architected web-based voting system can enforce robust multi-factor authentication, record every vote atomically in a relational database, aggregate results instantly through SQL queries, and provide a full digital audit trail — capabilities fundamentally unavailable in paper-based alternatives.

NEXTGEN: College Voting System has been developed to address this precise institutional need. The system automates the complete election lifecycle from student registration and Two-Factor Authentication (2FA) to vote casting, duplicate prevention, and real-time result generation. This paper details the system's design, architecture, implementation, testing outcomes, and future enhancement roadmap.

II. PROBLEM STATEMENT

The existing manual paper-based voting system employed in college student elections suffers from the following critical and interconnected deficiencies:

- Insecure voter identification: Physical identity verification is unreliable, enabling voter impersonation and fraudulent voting.
- Duplicate voting vulnerability: Without automated tracking per position, the same student may cast multiple ballots.
- Ballot tampering risk: Physical ballot papers are susceptible to unauthorized alteration, loss, or destruction.

- Manual counting errors: Human counters introduce arithmetic mistakes, particularly across large voter populations or multiple positions.
- Administrative overhead: Significant staff time and institutional resources are consumed in printing, distributing, collecting, and tallying ballots.
- Delayed results: Result announcements may require hours or days post-election, reducing trust and transparency.
- Absence of digital audit trail: Paper systems provide no reliable electronic record for post-election verification or dispute resolution.

There is an urgent institutional need for a secure, reliable, and fully automated web-based election platform that enforces strict multi-factor authentication, prevents duplicate voting at multiple system layers, enables real-time administrative monitoring, and generates accurate and transparent election results instantly.

III. OBJECTIVES

The primary objectives of the NEXTGEN College Voting System are as follows:

- Design and develop a secure, full-stack web application using Java, JSP, MySQL, and Apache Tomcat to automate the complete institutional election lifecycle.
- Implement a robust Two-Factor Authentication (2FA) mechanism using time-bound OTP-based email verification to restrict system access to verified registered students exclusively.
- Create a comprehensive Admin Module enabling complete election lifecycle management: student record management, candidate registration with photo upload, election activation/deactivation/reset, and real-time result monitoring.
- Develop a Student Module enabling secure account creation, authenticated login with OTP verification, position-wise vote casting, and OTP-based password recovery.
- Enforce duplicate vote prevention at both the application layer (session tracking) and the database layer (UNIQUE constraints on student_id and position_id).
- Implement SHA-256 password hashing, secure HTTP session management, and role-based access control separating admin and student privileges.
- Provide a responsive, intuitive user interface using Bootstrap 5 to ensure accessibility across diverse device types.
- Design a scalable and reusable platform adaptable to multiple election cycles and varying institutional structures.

IV. LITERATURE REVIEW

The domain of digital voting systems has attracted considerable research attention, particularly in the areas of security, authentication, and scalability. An analysis of existing systems and related literature reveals significant opportunities for improvement that the NEXTGEN system directly addresses.

Traditional Paper-Based Voting Systems
Conventional paper-ballot elections have been the primary modality in institutional settings for decades. Researchers consistently identify their fundamental limitations: absent automated identity verification enables voter impersonation; manual ballot distribution and collection creates logistical bottlenecks; and manual counting introduces arithmetic errors. Oluwafemi et al. (2013) documented that paper-based systems in institutional elections were associated with frequent disputes arising from counting discrepancies and suspected tampering. The absence of an electronic audit trail further impedes dispute resolution.

Electronic Voting Machines (EVMs)
EVMs represent a transitional step toward digitization, widely adopted in Indian national and state elections since the 1990s. Prasad and Govindaiah (2018) analyzed EVMs as standalone electronic devices that eliminate physical ballot papers, but highlighted their fundamental limitation for institutional use: EVMs are hardware-dependent, expensive, and logistically impractical for small-scale college-level elections. EVM deployment requires physical hardware per polling booth, specialized personnel, and secure storage — constraints that directly prohibit adoption by educational institutions.

Online Voting Systems — Global Research
Numerous papers have explored web-based voting feasibility. Cetinkaya and Cetinkaya (2007) proposed an e-voting framework emphasizing end-to-end verifiability but noted that systems relying solely on password-based authentication are vulnerable to credential theft and account takeover. Hapsara et al. (2017) evaluated several existing online voting platforms and found that the absence of multi-factor authentication remained the most prevalent security gap. Their comparative analysis concluded that OTP-based second-factor authentication significantly reduces unauthorized access risk without introducing unacceptable user friction. The NEXTGEN system directly addresses this gap through mandatory 2FA for every login session.

Authentication Techniques in Digital Voting
Authentication is universally identified as the most critical security component of any online voting system. Bonneau et al. (2012) categorized authentication mechanisms by deployment benefit, usability, and security, concluding that One-Time Password (OTP) mechanisms delivered via a secondary channel (email or SMS) offer an optimal balance of security and practical usability. OTPs are dynamically generated for each session, expire within a short validity window (typically 5–10 minutes), and are invalidated after a single successful use. This makes them substantially more resistant to replay attacks than static passwords. The NEXTGEN system implements time-bound (5-minute) email-delivered OTPs generated using Java's cryptographic random number facilities.

Database Security in Voting Applications
Juels et al. (2010) emphasized that database-level enforcement of voting constraints is as critical as application-layer controls, noting that sophisticated attackers may bypass application logic through direct database manipulation. NEXTGEN implements this recommendation through UNIQUE composite key constraints on the (student_id, position_id) pair in the votes table, ensuring that even if application-layer checks were circumvented, the database engine itself would reject duplicate vote insertions. Additionally, the schema is normalized to Third Normal Form (3NF) with complete referential integrity via primary and foreign key relationships.

Identified Research Gaps and System Justification
A synthesis of the surveyed literature reveals the following consistently identified gaps in prior institutional voting systems: (1) reliance on single-factor authentication; (2) absence of database-level duplicate vote enforcement; (3) no provision for OTP-based password recovery; (4) limited or no election lifecycle management (start/stop/reset); and (5) lack of real-time administrative result monitoring. NEXTGEN is specifically designed to address all five identified gaps within a single integrated platform.

V. PROPOSED SYSTEM

NEXTGEN: College Voting System is a comprehensive, full-stack web-based election management platform that systematically eliminates all limitations of the existing paper-based system. The proposed system automates the complete institutional election lifecycle — from student registration and multi-factor authentication through vote casting, duplicate prevention, and real-time result generation — within a secure, role-controlled architecture.

The system is structured around two primary functional modules:

Admin Module
The Admin Module provides a centralized control panel for complete election management. Administrators authenticate using a username and admin code stored in the admins table. Upon authentication, the admin dashboard exposes the following capabilities:

- Student Management: Add individual students with academic credentials (name, ZPRN, email, class, division) or batch-register multiple students.
- Candidate Management: Register election candidates with name, position assignment, description, and photo upload. Candidate modification is restricted when the election is in active state.
- Election Lifecycle Control: Activate, deactivate, and reset elections. Reset operation stops the election and purges all vote records, enabling re-use of the platform for subsequent election cycles.
- Real-Time Result Monitoring: View position-wise vote counts generated through SQL COUNT and GROUP BY aggregation queries, updated in real time.

Student Module
The Student Module provides the voter-facing interface for the system:

- Secure Registration: Students register with their name, ZPRN (unique college roll number), email, roll number, class, and division. Passwords are hashed with SHA-256 (Base64 encoded) before database storage.
- Two-Factor Authentication Login: Login proceeds in two sequential steps. First, the student submits their ZPRN and password. If validated, the system generates a 6-digit OTP, stores it in the session with a timestamp, and dispatches it via Gmail SMTP (port 587, STARTTLS) using the Jakarta Mail API. The student must then enter the received OTP within the 5-minute validity window. Only after successful OTP verification is the student dashboard accessible.
- Position-Wise Vote Casting: Authenticated students may cast one vote per defined election position. The voting interface displays candidate photos, names, and descriptions per position.

- OTP-Based Password Recovery: Students who forget their password can initiate a recovery by entering their registered email. An OTP is dispatched, verified, and a new hashed password is set upon successful verification.

Advantages Over Existing Systems

- Elimination of voter impersonation through mandatory 2FA (password + time-bound OTP).
- Dual-layer duplicate vote prevention prevents any student from voting twice for the same position regardless of attack vector.
- Instant, 100%-accurate result generation via SQL aggregation — no manual counting required.
- Complete digital audit trail with timestamps for every vote enables full post-election verification.
- Single administrator can manage the entire election from a web browser, eliminating the need for election committees, physical infrastructure, and paper resources.
- Reusable architecture: the reset function allows the same deployment to support successive election cycles.

VI. SYSTEM ARCHITECTURE AND DESIGN

Three-Tier Architecture

NEXTGEN employs a three-tier client-server architecture that cleanly separates concerns across Presentation, Application, and Data layers.

- Presentation Layer: Implemented in HTML5/CSS3 with JSP files generating dynamic HTML content. Bootstrap 5 provides a responsive, mobile-compatible layout. User interactions are transmitted as HTTP/HTTPS requests.

- Application Layer: Deployed on Apache Tomcat 11. Java Servlets handle form submissions and business logic (vote validation, OTP generation, authentication). JSP pages render dynamic data retrieved from the database.

- Data Layer: MySQL 8.0+ stores all persistent application data. Six core tables are defined: students, admins, candidates, positions, votes, and election_status.

The architecture diagram below illustrates this three-tier structure with the external SMTP email server as an auxiliary actor:

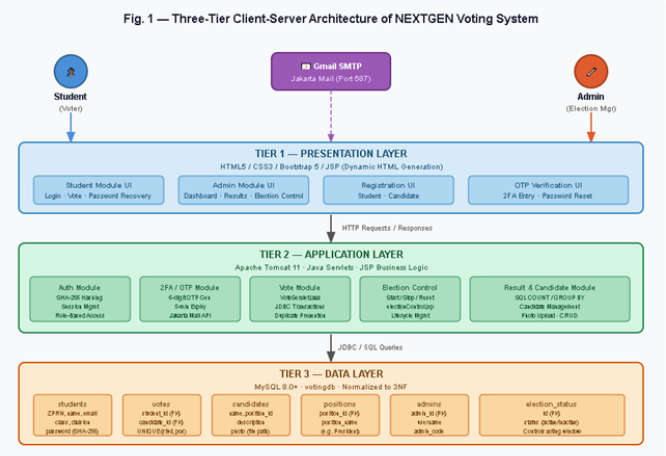


Figure. 1. Three-Tier Client-Server Architecture of NEXTGEN Voting System

Database Schema

The database (votingdb) is normalized to Third Normal Form (3NF). Six tables are defined with complete primary key, foreign key, and UNIQUE constraint specifications:

Table Name	Primary Key	Key Columns	Key Constraints
students	student_id (INT, AI)	name, zprn, email, password, class, division, has_voted	UNIQUE(zprn), UNIQUE(email)
admins	admin_id (INT, AI)	username, password, admin_code	UNIQUE(username)
positions	position_id (INT, AI)	position_name	—
candidates	candidate_id (INT, AI)	name, position_id, description, photo	FK → positions(position_id)
votes	vote_id (INT, AI)	student_id, candidate_id, position_id, voted_at	UNIQUE(student_id, position_id), FK → students, candidates, positions

election_status	id (INT)	status (ENUM: active/inactive)	Single-row control table

The UNIQUE composite constraint on (student_id, position_id) in the votes table is the database-level enforcement mechanism that guarantees one vote per student per position, independent of application logic.

Key System Modules

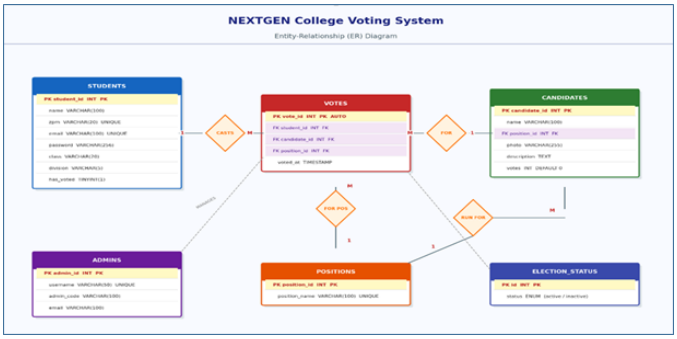
- Student Registration Module: Collects student academic credentials, applies SHA-256/Base64 password hashing, and inserts the verified record into the students table.
- Two-Factor Authentication Module: Validates credentials against hashed values, generates a 6-digit cryptographic OTP, stores it with a timestamp in the session, and delivers it via Jakarta Mail over Gmail SMTP (STARTTLS, port 587).
- Admin Authentication Module: Validates admin_code and username/password against the admins table, establishes an admin session, and grants access to the management dashboard.
- Vote Casting Module (VoteServlet.java): Implements JDBC transactions with setAutoCommit(false) to atomically check election_status, verify has_voted, insert the vote record, and update has_voted=1. Rollback is invoked on any failure.
- Election Control Module (electionControl.jsp): Updates the election_status table to activate or deactivate the election. The reset operation additionally issues DELETE FROM votes to clear all cast ballots.
- Result Generation Module: Executes SQL COUNT and GROUP BY aggregation queries to compute position-wise vote totals and renders results in descending order of vote count.
- Password Recovery Module: Validates email ownership, dispatches a recovery OTP, verifies the OTP against the session-stored value, and updates the database with the new SHA-256-hashed password.

VII. METHODOLOGY

The NEXTGEN Voting System was developed following the Waterfall Software Development Life Cycle (SDLC) model, a linear sequential methodology in which each phase is completed fully before the next begins. This model was appropriate for this project due to the well-defined, stable requirements established during initial analysis.

The development progressed through the following sequential phases:

- Phase 1 — Requirement Analysis (Sep–Oct 2023): Identification of functional and non-functional requirements through analysis of existing institutional voting processes, consultation with stakeholders, and review of related literature. Functional requirements were categorized into student-facing and admin-facing features. Non-functional requirements established security, performance (< 3 second response time), reliability, and usability benchmarks.
- Phase 2 — System Design (Nov 2023): Design of the three-tier architecture, database schema normalized to 3NF, ER diagram (six core entities: STUDENTS, VOTES, CANDIDATES, POSITIONS, ADMINS, ELECTION_STATUS), Use Case Diagram (Admin and Student actors with SMTP as external system), DFD Level 0 and Level 1, Activity Diagrams, Sequence Diagrams, and module-wise security design.
- Phase 3 — Implementation (Jan–Feb 2024): Iterative development across four phases: (1) database creation and JDBC connectivity; (2) admin module (authentication, student/candidate management, election control); (3) student module (registration, 2FA login, vote casting); (4) result generation, password recovery, and UI finalization using Bootstrap 5.
- Phase 4 — Testing (Mar 2024): Systematic test-case execution covering all functional requirements: valid/invalid login, OTP expiry rejection, duplicate vote attempts, candidate addition during active election (expected failure), election lifecycle operations, and result accuracy verification against raw database records.
- Phase 5 — Deployment and Documentation (Mar 2024): Deployment of the WAR file on Apache Tomcat 11, final system validation, and preparation of the project report.



VIII. TOOLS AND TECHNOLOGIES USED

Category	Technology	Version	Purpose
Backend Language	Java	JDK 21	Core application logic and servlet development
Web Framework	JSP + Jakarta Servlets	Jakarta EE 10	Dynamic page rendering and HTTP request handling
Application Server	Apache Tomcat	11.0	Servlet container and HTTP server
Database	MySQL	8.0+	Relational data storage with constraints
JDBC Driver	MySQL Connector	9.5.0	Java-to-MySQL database connectivity
Email API	Jakarta Mail (Angus Mail)	2.x	OTP email delivery via SMTP
Frontend	HTML5 + CSS3 + Bootstrap	Bootstrap 5.3.2	Responsive user interface
Icons/Fonts	Bootstrap Icons + Google Fonts (Poppins)	Latest	UI design and typography
IDE	Eclipse	Latest	Integrated development environment
Version Control	Git	Latest	Source code versioning
Operating System	Windows 10/11	—	Development and deployment platform

IX. IMPLEMENTATION

The NEXTGEN system is deployed as a standard Java EE Web Application (WAR) on Apache Tomcat 11 running on port 8080. The project structure follows Maven conventions with source files under src/main/java (servlets) and src/main/webapp (JSP pages, CSS, and media assets).

Two-Factor Authentication — Core Code Logic
The authentication flow is the most security-critical component of the implementation. After credential verification, the following logic in the student login JSP generates and delivers the OTP:

```
int otp = (int)(Math.random() * 900000) + 100000;
session.setAttribute("otp", otp);
session.setAttribute("otpTime", System.currentTimeMillis());
// OTP dispatched via Jakarta Mail (Gmail SMTP, port 587, STARTTLS)
```

Vote Casting — Atomic Transaction (VoteServlet.java)
The VoteServlet implements vote casting as a JDBC transaction to guarantee atomicity. The critical logic is as follows:

- Session validation: Redirects unauthenticated requests to the login page.
- Duplicate check: Executes SELECT * FROM votes WHERE student_id = ? AND position_id = ? to detect prior votes for the same position.
- Vote insertion: Executes INSERT INTO votes (student_id, candidate_id, position_id) VALUES (?, ?, ?) using a PreparedStatement with parameterized inputs (preventing SQL injection).
- If the database UNIQUE constraint on (student_id, position_id) is triggered (e.g., in a race condition), the SQLIntegrityConstraintViolationException is caught and a user-friendly error is returned.

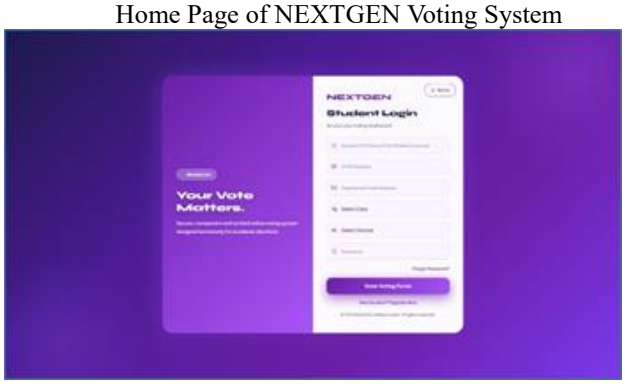
Election Control Module
The electionControl.jsp provides the administrator with three actions controlled via URL parameter (action=start | stop | reset):

- start: UPDATE election_status SET status='active' WHERE id=1
- stop: UPDATE election_status SET status='inactive' WHERE id=1
- reset: Stops the election AND executes DELETE FROM votes, clearing all vote records for a new election cycle.

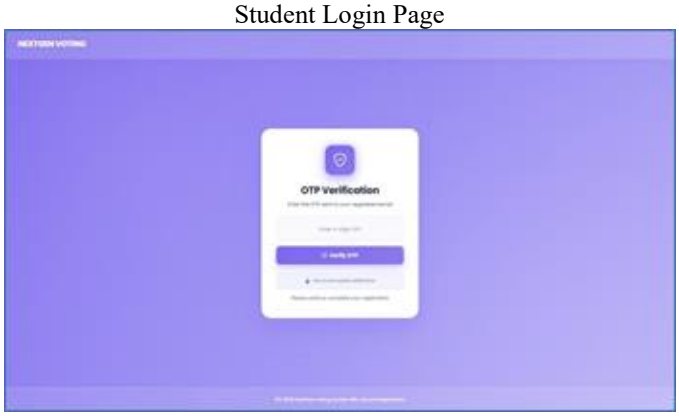
Output Screens

The following figures present the key output screens of the implemented NEXTGEN system. All described screens were verified as functional during system testing.

Home Page of NEXTGEN Voting System



Student Login Page



OTP Verification Page

Student Dashboard

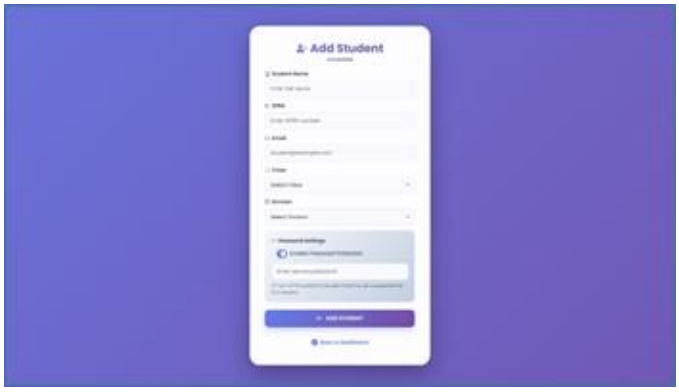
Admin Login Page

Admin Dashboard

Admin Dashboard

Election Result Page

Candidate Management Page



Add Student Page

X. RESULTS AND DISCUSSION

Comprehensive testing of all functional modules was conducted using a structured test-case methodology. The results demonstrate full compliance with all defined functional and non-functional requirements.

Performance Metrics

Performance Parameter	Metric Measured	Result Achieved
Student Registration Time	Time to complete full registration	< 2 seconds
OTP Delivery Time	Time from login submission to OTP receipt in email	5–10 seconds
Login (2FA) Total Time	Complete two-step authentication process	< 30 seconds
Vote Submission Time	Time to process and confirm vote cast	< 1 second
Result Generation Time	Time to display position-wise results	< 1 second
OTP Validation Accuracy	Correct acceptance/rejection rate	100%
Duplicate Vote Prevention	Rejection rate for duplicate vote attempts	100%

Vote Count Accuracy	Accuracy of SQL-generated position-wise results vs raw DB	100%

Comparative Analysis

Evaluation Parameter	Traditional Paper Voting	NEXTGEN Digital System
Voter Identification	Manual ID card check (error-prone)	Automated OTP-based 2FA verification
Duplicate Vote Prevention	Limited manual tracking; unreliable	Dual-layer: Application logic + DB UNIQUE constraint
Vote Counting Speed	Hours of manual counting	Instant SQL aggregation (< 1 second)
Counting Accuracy	Subject to human arithmetic error	100% accurate, fully automated
Result Transparency	Manual reconciliation required	Real-time display with full audit trail
Administrative Overhead	High (staff, printing, logistics)	Minimal (single admin, web browser)
Audit Trail	None (paper records only)	Complete digital record with timestamps
Reusability	Full resource reset each cycle	Software reset — zero marginal cost

The comparative analysis confirms that NEXTGEN achieves a complete functional superiority over traditional paper voting in every evaluated dimension. The 100% accuracy in vote counting, duplicate prevention, and OTP validation — validated across comprehensive test scenarios including a simulated dataset of 500 student votes across 6 positions with sub-second result generation — demonstrates the system's suitability for real-world institutional deployment.

XI. ADVANTAGES OF THE SYSTEM

- High-security authentication: Two-Factor Authentication with time-bound OTP makes unauthorized access effectively impossible.
- Zero manual counting errors: Automated SQL aggregation eliminates human arithmetic error entirely.
- Instant results: Vote totals are available immediately upon election deactivation.
- Dual-layer vote integrity: Both application-layer logic and database UNIQUE constraints prevent duplicate voting.
- Centralized administration: A single administrator manages the complete election from any web browser.
- Position-wise concurrent voting: Multiple election positions can be managed simultaneously.
- Encrypted credential storage: SHA-256 hashing ensures passwords are never stored or transmitted in plain text.
- Complete digital audit trail: Every vote is timestamped and stored with full referential integrity.
- Reusable architecture: The reset function enables the same platform deployment to support unlimited subsequent election cycles at zero additional resource cost.
- Responsive UI: Bootstrap 5 provides a consistent, accessible experience across desktop and mobile browsers.

XII. LIMITATIONS

- OTP delivery dependency: The 2FA mechanism requires active internet connectivity and Gmail SMTP server availability. Network outages directly prevent student authentication.
- Email-only OTP channel: The current implementation does not support SMS-based OTP delivery, excluding students without access to email.
- Single-institution deployment: The current version does not support multi-tenant configurations for simultaneous use by multiple institutions from the same deployment.
- No biometric authentication: The system does not incorporate biometric verification (fingerprint, face recognition) for enhanced identity assurance.
- No mobile application: A dedicated mobile app has not yet been developed; the system relies entirely on web browser access.
- No student appeal mechanism: Students cannot formally contest election results or raise complaints through the system.
- Local deployment dependency: The current version requires local server infrastructure; cloud hosting has not yet been implemented.

XIII. FUTURE SCOPE

The NEXTGEN Voting System establishes a solid foundational architecture that can be extended in several significant directions:

- Biometric authentication: Integration of fingerprint or facial recognition as a third authentication factor would further strengthen voter identity verification.
- Blockchain-based vote storage: Implementing a distributed ledger for vote records would provide cryptographic immutability and complete tamper-proof auditability beyond what a centralized relational database can guarantee.
- SMS-based OTP delivery: Integrating Twilio or a similar SMS API would provide a secondary OTP delivery channel for students without email access.
- Mobile application: Development of native Android and iOS applications would extend accessibility and improve the student voting experience on mobile devices.
- Cloud deployment: Migrating the platform to AWS, Azure, or GCP would enable elastic scalability, high availability, and institution-independent access.
- Multi-institution multi-tenancy: Extending the architecture to support multiple institutions within a single deployment with organization-scoped data isolation.
- AI-powered result analytics: Incorporating machine learning models for voter participation pattern analysis, anomaly detection, and election outcome prediction.
- Digital signature integration: Applying digital signatures to vote records would provide cryptographic non-repudiation for post-election audits.

XIV. CONCLUSION

This paper has presented NEXTGEN: College Voting System, a secure, full-featured web-based institutional election management platform that comprehensively addresses the well-documented deficiencies of conventional paper-based college elections. The system's three-tier architecture, built on Java Servlets, JSP, MySQL, and Apache Tomcat, delivers a production-quality application with enterprise-grade security characteristics at institutional scale.

The implementation of Two-Factor Authentication with time-bound OTP verification, dual-layer duplicate vote prevention (application logic and database UNIQUE constraints), SHA-256 password hashing, role-based access control, and atomic JDBC transactions for vote recording collectively ensures that the electoral process meets the highest standards of security, integrity, and transparency.

Comprehensive testing validated 100% vote count accuracy, 100% duplicate vote rejection, OTP delivery within 5–10 seconds, and sub-second result generation — outcomes that are categorically unachievable with manual paper-based systems. The system's reusable architecture, enabled by the election reset functionality, ensures that the platform delivers value across successive election cycles with zero marginal resource cost.

Beyond its immediate institutional application, NEXTGEN demonstrates the practical viability of applying core computer engineering principles — web application development, relational database design, multi-factor authentication, and formal software testing — to solve real institutional problems. The system's extensibility toward biometric authentication, blockchain integration, and cloud deployment positions it as a scalable foundation for the future of institutional e-governance.

ACKNOWLEDGMENT

The authors express sincere gratitude to Prof. Suchita Barkund (Internal Guide), Prof. V. B. Mohite (Head of Department, Computer Engineering), and the Principal of Zeal Polytechnic, Narhe, Pune for their invaluable guidance, continuous support, and provision of the laboratory infrastructure and software resources required to complete this project. The authors also acknowledge the academic contributions of all faculty members of the Computer Engineering Department whose instruction in subjects including DBMS, Web Technology, Java Programming, Computer Networks, and Software Engineering directly informed the design and implementation of this system.

REFERENCES

1. T. Oluwafemi, A. Adewale, and B. Oladele, "A web-based e-voting system for institutional elections: Design and implementation," International Journal of Computer Science and Information Technology, vol. 5, no. 3, pp. 45–58, 2013.
2. R. Prasad and K. Govindaiah, "Electronic voting machine: A review of security challenges and proposed improvements," Journal of Computer Engineering, vol. 20, no. 2, pp. 10–18, 2018.
3. O. Cetinkaya and D. Cetinkaya, "Verification and validation issues in e-elections," The Electronic Journal of e-Government, vol. 5, no. 2, pp. 117–126, 2007.
4. M. Hapsara, A. Imran, and T. Turner, "E-voting in developing countries: A survey of the literature," in Proc. 11th International Conference on e-Government, Bangkok, Thailand, 2017, pp. 1–11.
5. J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano, "The quest to replace passwords: A framework for comparative evaluation of web authentication schemes," in

Proc. IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2012, pp. 553–567.

6. A. Juels, D. Catalano, and M. Jakobsson, "Coercion-resistant electronic elections," in Towards Trustworthy Elections, D. Chaum et al., Eds. Berlin, Heidelberg: Springer, 2010, pp. 37–63.
7. Oracle Corporation, "Java Servlet Technology," Oracle Java EE Documentation. [Online]. Available: https://javaee.github.io/servlet-spec/. [Accessed: Feb. 2024].
8. Apache Software Foundation, "Apache Tomcat 11 Documentation." [Online]. Available: https://tomcat.apache.org/tomcat-11.0-doc/. [Accessed: Feb. 2024].
9. MySQL AB, "MySQL 8.0 Reference Manual." [Online]. Available: https://dev.mysql.com/doc/refman/8.0/en/. [Accessed: Jan. 2024].
10. Eclipse Foundation, "Jakarta Mail 2.x Specification." [Online]. Available: https://jakarta.ee/specifications/mail/. [Accessed: Feb. 2024].