International Journal of Applied Science and Engineering Review

IJASER

ISSN: 2582-6271

Vol. 7, Issue. 2, March-April 2026, page no. 68-85

To cite this article: Ayodeji Bolanle Balogun and Olufunke Adewunmi Ayegbusi (2026). WEB-BASED EVOTING SYSTEM FOR STUDENT UNION ELECTIONS: A CASE STUDY OF NAPES AT FEDERAL POLYTECHNIC ILARO, International Journal of Applied Science and Engineering Review (IJASER) 7 (2): 68-85 Article No. 260 Sub Id 389

WEB-BASED EVOTING SYSTEM FOR STUDENT UNION ELECTIONS: A CASE STUDY OF NAPES AT FEDERAL POLYTECHNIC ILARO

Ayodeji Bolanle Balogun[1] and Olufunke Adewunmi Ayegbusi[2]

[1]Dept. of Electrical Electronic Engineering, Federal Polytechnic Ilaro,
Ogun State, Nigeria
Ayodeji.balogun@fedederalpolyilaro.edu.ng

[2]Dept. of Civil Engineering, Federal Polytechnic Ilaro,
Ogun State, Nigeria
olufunke.ayegbusi@fedederalpolyilaro.edu.ng

DOI: https://doi.org/10.52267/IJASER.2026.7205

ABSTRACT

This paper presents the design, implementation, and empirical evaluation of a web-based electronic voting system deployed for the NAPES 2025/2026 General Election at the School of Engineering, Federal Polytechnic Ilaro, Nigeria. The system was built on a React 19 / Express 4 / tRPC 11 / MySQL stack and implements a three-role access control model (administrator, accreditor, voter), OTP-based voter authentication with a 15-minute expiry window, cryptographic vote anonymisation via SHA-256 hashing, a comprehensive audit trail, and a configurable results visibility control. The system processed 636 complete ballots from 3,189 registered voters across 14 positions, generating 8,904 individual vote records with zero detected integrity violations. Analysis of 215 logged API requests recorded a median (P50) response time of 445 ms, with 81.4% of requests completing under 1,000 ms. Peak concurrent load reached 8 simultaneous ballot submissions per minute during the 11:00–12:00 WAT peak hour. The OWASP analysis identified three fully mitigated threat categories (broken access control, injection, security logging) and three partially mitigated categories (cryptographic failures, security misconfiguration, authentication failures), with residual risks explicitly documented. The paper explicitly positions the system against prior Nigerian student e-voting implementations, identifying four design advances: cryptographic vote anonymisation, a multi-role accreditation workflow, a configurable results visibility control, and a complete audit trail with IP logging.

International Journal of Applied Science and Engineering Review

IJASER

ISSN: 2582-6271

Vol. 7, Issue.2, March-April 2026, page no. 68-85

KEYWORDS: e-voting, web application, student union election, otp authentication, vote anonymisation, security evaluation

1. INTRODUCTION

In higher institutions of learning, election of student unions in Nigeria has been traditionally done by paper-based system which can be easily subjected to ballot stuffing, intimidation and irregularities in counting elections [1]. These were the problems encountered in the past election cycles by the School of Engineering at the Federal Polytechnic Ilaro, the home of Nigerian Association of Polytechnic Engineering Students (NAPES). The three reasons behind the decision to design a purpose -built web-based e-voting system to be used in general election 2025/2026 were to remove procedural fraud, to produce a verifiable and auditable electoral record, and to ease the burden on the administration of counting manual votes across 14 positions and 3,189 registered voters.

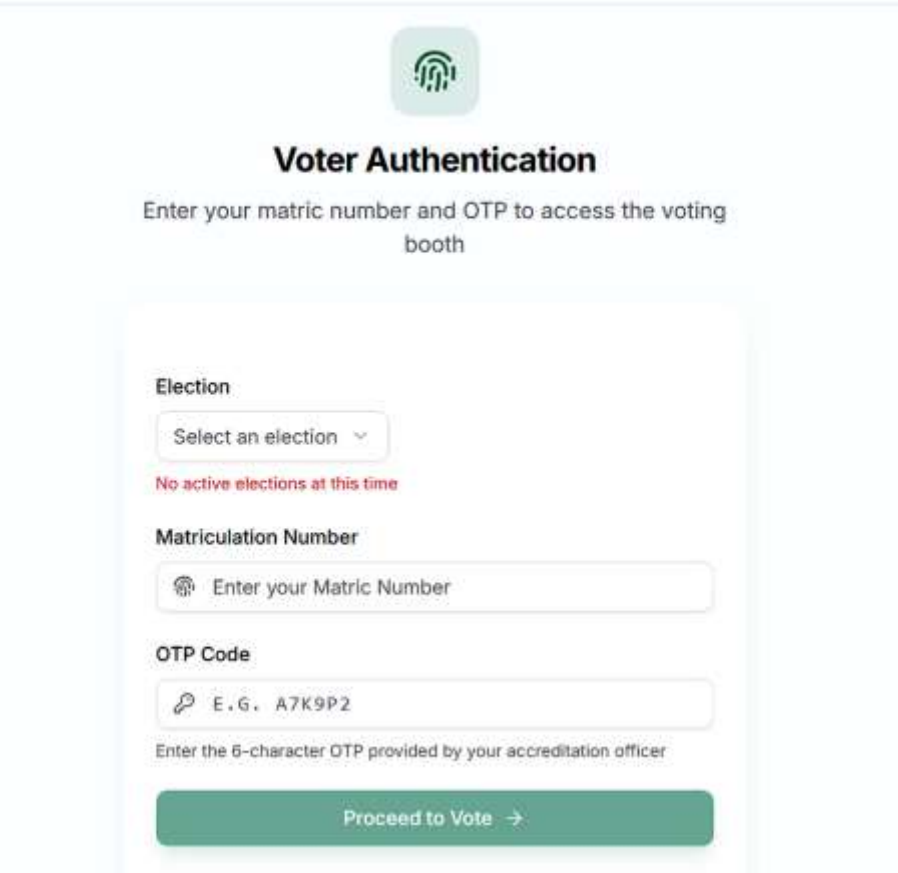


Figure 1: Voter Authentication Page

ISSN: 2582-6271

Vol. 7, Issue.2, March-April 2026, page no. 68-85

Implementation of e-voting systems is an established field of study and has been implemented in both basic web-forms and more advanced cryptographic systems based on blockchains [2] [3]. Nonetheless, most published systems in Nigerian context are either proof-of-concept implementations which despite being deployed never actually conducted live elections, or systems which were deployed without subsequent post-election empirical evaluation [1]. Both gaps are covered in the present paper, which records a system designed, deployed, and tested in a actual election with performance measures based on production logs and a security assessment based on the OWASP Top 10 model [5]. The rest of the paper has the following structure. Section 2 examines the associated literature and makes a clear reference point of this system over previously implemented e-voting systems among Nigerian students. Section 3 presents the architectural description of the system and major design choices. The security evaluation is provided under section 4. Section 5 captures on the deployment context and the performance measures. Section 6 gives the discussion on the finding and implication and Section 7 concludes.

2. RELATED WORK AND POSITIONING

2.1 Prior Nigerian Student E-Voting Implementations

Adekitan et al. [1] designed an e-voting system of elections of the student union government in Nigerian institutions of higher learning, which was driven by reported cases of violence and irregularities going back to the 1988 Abu crisis. Their platform was simply an authentication of the voter and a digital submission of the ballots and was tested by simulation and not by a live implementation. Their main contribution was the showing of the technical feasibility of e-voting in the context of the Nigerian institutional setting; the system lacked vote anonymisation, multi-role access control and audit trail.

One example was a two-factor authentication online voting platform that proposed the use of webcam and facial recognition along with voter ID and password-based logins, which was introduced by Ipinnimo et al. [9], and was based on Next.js and Node.js stack and a MongoDB database. The system added biometric identity verification as a second factor of authentication, something that is a significant improvement over strictly credential system. An email OTP channel was also incorporated in the architecture providing further verification. The system was, however, checked on simulation and mini-voting sessions and not a real election and the authors do not give any post-election empirical measures. It lacks the cryptographic vote anonymisation system, the vote data is stored on MongoDB with no policy to detach a voter-ballot association, and no security assessment was done. Any authenticated voter can access the results page without a visibility switch that can only be controlled by an administrator.

One implementation of an e-voting system using blockchain, dual biometric authentication (fingerprint and facial recognition) and assessed with data on the 2024 Edo State gubernatorial election provided by INEC was proposed by Osayomore, Ejenarhome and Oise [10]. Their system is the most technically advanced among the reviewed implementations that uses a decentralised ledger to store immutable votes

International Journal of Applied Science and Engineering Review

IJASER

ISSN: 2582-6271

Vol. 7, Issue.2, March-April 2026, page no. 68-85

and uses cryptographic key pair generation per voter to provide cast-as-intended verifiability. The blockchain design offers a powerful audit trail and can facilitate independent verification of votes by the stakeholders. Nevertheless, the system was not implemented on a live election where the INEC data was utilised just to analyse problems and model scalability and no empirical performance measures were published. There is a lack of an OTP-based accreditation workflow, a results visibility control, formal OWASP security evaluation.

A pattern is found in the wider literatures on the e-voting in Nigeria: that systems are narrated at the design level, but seldom seem to be deployed in actual elections, and even fewer receive empirical assessment of use after the election. This disparity between architectural report and working evidence is the main limitation of the current literature. A vulnerability analysis of an implemented e-voting system by Sunardi et al. [4] on the basis of the OWASP framework showed that it was vulnerable to critical XSS vulnerabilities and broken authentication and that an architectural description is not a sufficient indication of security. Those gaps have been discussed in the paper and the gap with reference to reporting a system deployed in a real election and tested against production data with a systematic OWASP threat analysis conducted on the live architecture.

Table 1: Feature Comparison of Nigerian Institutional E-Voting Implementations

Feature	**Adekitan et al. [1]**	**Ipinnimo et al. [9]**	**Osayomore et al. [10]**	**NAPES System (This Work)**
Live deployment	No (simulation)	No (simulation)	No (prototype)	**Yes** (636 voters, 14 positions)
Vote anonymisation	No	No	Yes (blockchain keypair)	**Yes** (SHA-256 voteHash, no voterId in votes table)
Multi-role access control	Basic (admin/voter)	Basic (admin/voter)	Basic (admin/voter)	**Three roles** (admin, accreditor, voter)
OTP authentication	No	Yes (email OTP)	No (biometric only)	**Yes** (6-digit verbal OTP, 15-min expiry)
Biometric authentication	No	Yes (face recognition)	Yes (fingerprint + face)	No (physical ID + OTP)
Blockchain	No	No	Yes	No

ISSN: 2582-6271

Vol. 7, Issue.2, March-April 2026, page no. 68-85

Audit trail	No	Partial (access tokens)	Yes (blockchain ledger)	**Full** (IP, timestamp, action, user, 1,374 events)
Results visibility control	No	No	No	**Yes** (real-time or post-close toggle)
Formal security evaluation	None	None	SWOT analysis	**OWASP Top 10 analysis**
Performance metrics	None	None	None	**215 logged requests, P50=445 ms**
Post-election empirical data	None	None	None (INEC data only)	**Full dataset** (636 ballots, 8,904 records)
Tech stack	Not specified	Next.js, Node.js, MongoDB	Blockchain, biometric modules	React 19, Express 4, tRPC 11, MySQL

The NAPES system is a step forward in the state of practice, in comparison to all three previous systems, in four dimensions: (1) it is the only system in this comparison which has been deployed in a live institutional election and tested against real production data; (2) it implements cryptographic vote anonymisation that breaks the voter-ballot connection at the database schema level, without a full blockchain infrastructure (3); it introduces a three-role accreditation workflow that decouples voter identification and ballot casting, which provides structural protection against insider threats not provided by the other systems reviewed. It is worth noting that, even though Osayomore et al. [10] provide stronger cryptographic guarantees with blockchain, the NAPES system shows that similar anonymisation and auditability can be realised with a standard relational database stack, and is more accessible to institutions with resource constraints.

2.2 Security Requirements for E-Voting Systems

Gritzalis [6] has defined the canonical list of security requirements of e-voting systems: eligibility (only registered voters are allowed to vote), uniqueness (each voter can vote at most once), anonymity (the correlation between voter identity and ballot votes should be undetectable), verifiability (voters should be able to verify that their ballot was recorded), fairness (no biased results can be available before the end of the election), integrity (votes cannot be altered or removed). The NAPES system can be used to address all the six requirements fully or partially as illustrated in Section 4.

International Journal of Applied Science and Engineering Review

IJASER

ISSN: 2582-6271

Vol. 7, Issue.2, March-April 2026, page no. 68-85

Zou et al. [3] proposed a publicly acceptable, auditable, and verifiable e-voting scheme, arguing that transparency, the capability of voters and observers to check the electoral procedure without jeopardising anonymity, is a requirement for institutional trust in e-voting. This transparency principle is reflected in the public results page of the NAPES system, the downloadable results certificate and the audit log.

3. SYSTEM ARCHITECTURE AND DESIGN

3.1 Technology Stack

The system was built on the following technology stack, selected for its end-to-end type safety, developer productivity, and compatibility with the managed cloud deployment environment:

Table 2: Technology Stack

Layer	Technology	Version	Rationale
Frontend framework	React	19	Component model, hooks, concurrent rendering
Build tool	Vite	6	Fast HMR, optimised production bundles
UI components	shadcn/ui + Tailwind CSS	4	Accessible, composable, no runtime CSS
API layer	tRPC	11	End-to-end type safety, no REST contract files
Serialisation	SuperJSON	—	Native Date/BigInt support over the wire
Backend runtime	Node.js + Express	22 / 4	Mature ecosystem, managed platform support
ORM	Drizzle ORM	—	Type-safe SQL, schema-first migrations
Database	MySQL / TiDB Serverless	—	ACID transactions, managed scaling
Authentication	Manus OAuth + JWT	—	Institutional SSO for admin, OTP for voters
Deployment	Managed cloud (Manus)	—	HTTPS enforced, auto-scaling, TLS 1.3

3.2 Three-Role Access Control Model

The system implements a three-role model that separates the election administration, voter accreditation, and ballot-casting functions into distinct access domains:

International Journal of Applied Science and Engineering Review

IJASER

ISSN: 2582-6271

Vol. 7, Issue.2, March-April 2026, page no. 68-85

Administrator - full systems access through Manus OAuth login. This is the creation and configuration of elections, managing positions and candidates, uploading voter registers (CSV bulk import), controlling the election lifecycle (draught - active - paused - closed), granting results visibility, and auditing the audit logs. Every administrative command is guarded by the procedure adminProcedure, which authenticates as well as examines the assertion of the role of admin prior to executing the command.

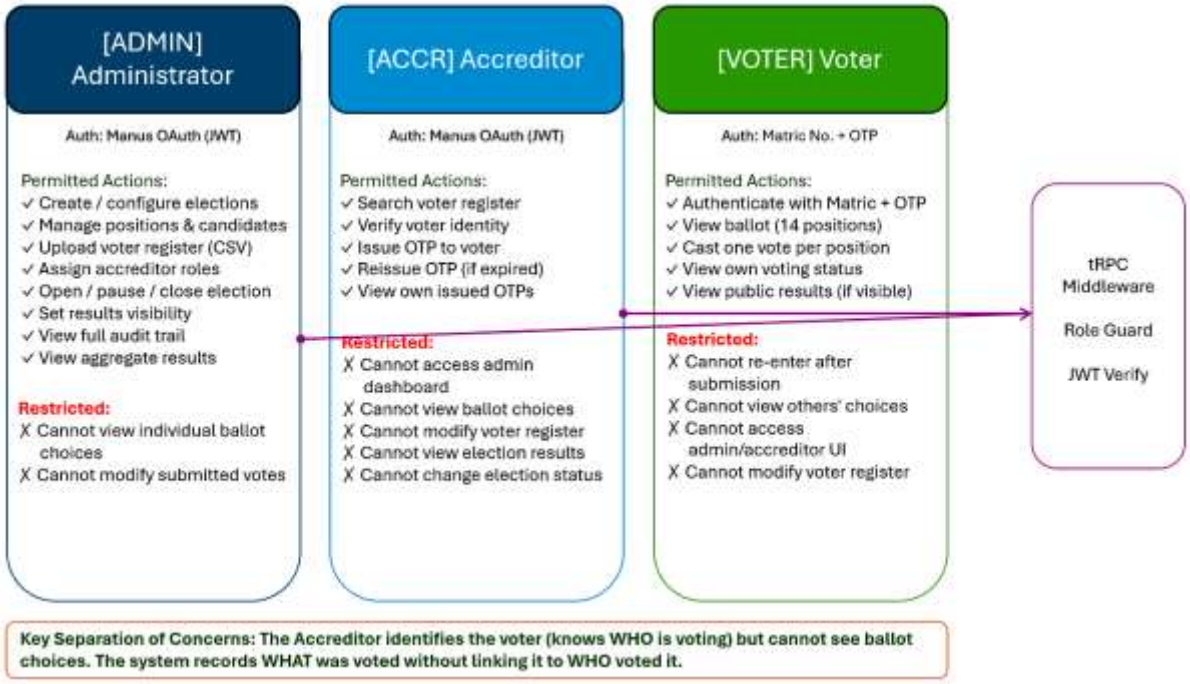


Figure 1: Three-Role Access Control Model

Accreditor - a position held by the administrator at the field level and is given to election officials. The accreditor only has the role of verifying the physical identity of a voter against voter register and send a one time password (OTP) to the voter's device. The accreditor is not able to see the ballot options, update the voter register, or the admin dashboard. Such separation of roles guarantees that the identity verification party is a different individual than the one that is handling the ballot, which is similar to paper elections where the polling officer is not the same party that handles the ballot box.

Voter - a registered student who logs in with his or her matriculation number and the OTP given by the accreditor. The voter role is stateless: there is no session generated, and the voter cannot go back into the voting interface once his or her ballot is placed. The has Voted flag in the voters table is updated atomically with the ballot insertion on a single database statement and preclude the possibility of duplicate voting even when two or more requests are made simultaneously.

International Journal of Applied Science and Engineering Review

IJASER

ISSN: 2582-6271

Vol. 7, Issue.2, March-April 2026, page no. 68-85

3.3 OTP Authentication Flow

The OTP authentication flow is the primary security boundary between the physical accreditation process and the digital ballot-casting process.

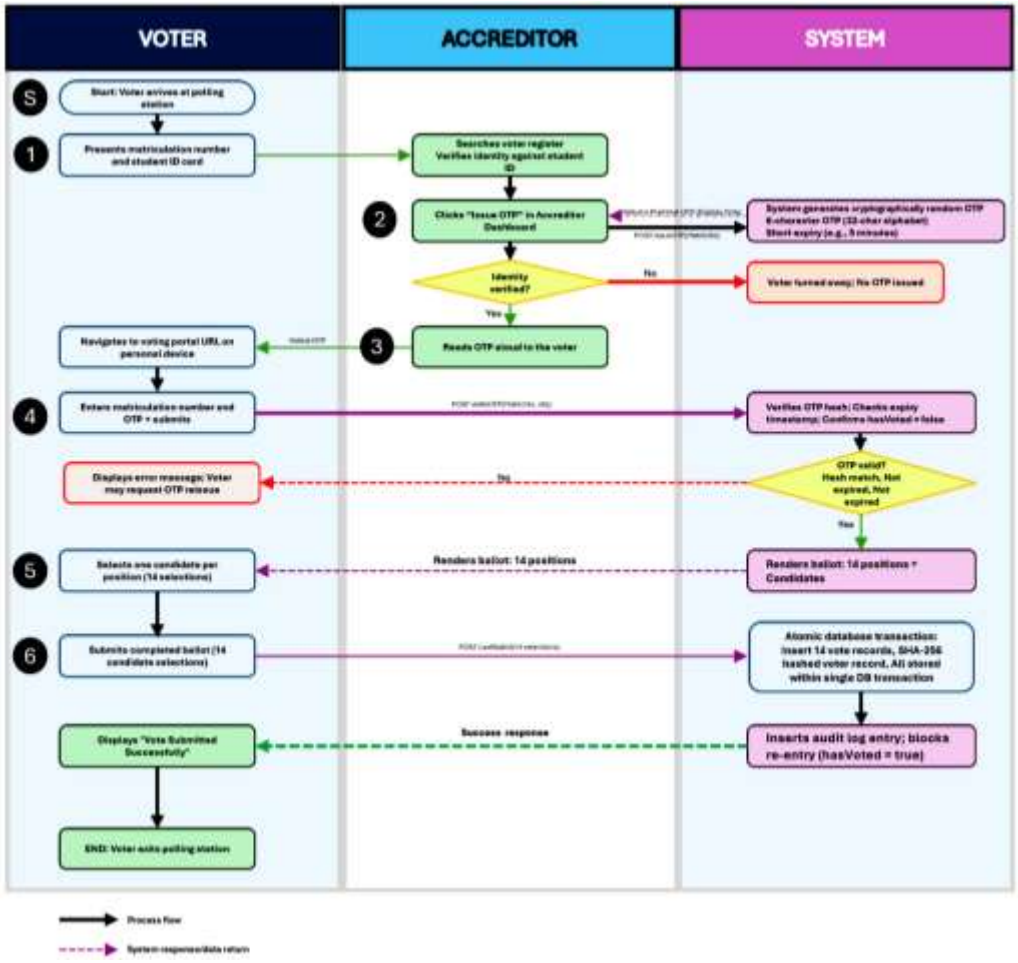


Figure 2: OTP Authentication and Ballot-Casting Sequence Flow

As shown in Figure 3, the flow proceeds as follows:

1 The voter presents themselves to the accreditor and provides their matriculation number.

2 The accreditor searches the voter register and verifies the voter's identity against their student ID card.

3 The accreditor issues an OTP via the accreditor dashboard. The system generates a cryptographically random 6-digit code, stores its bcrypt hash in the otpLogs table with a 15-

International Journal of Applied Science and Engineering Review

IJASER

ISSN: 2582-6271

Vol. 7, Issue.2, March-April 2026, page no. 68-85

minute expiry timestamp, and displays the plaintext code to the accreditor for verbal communication to the voter.

4 The voter navigates to the voting interface on their own device, enters their matriculation number and the OTP, and submits.

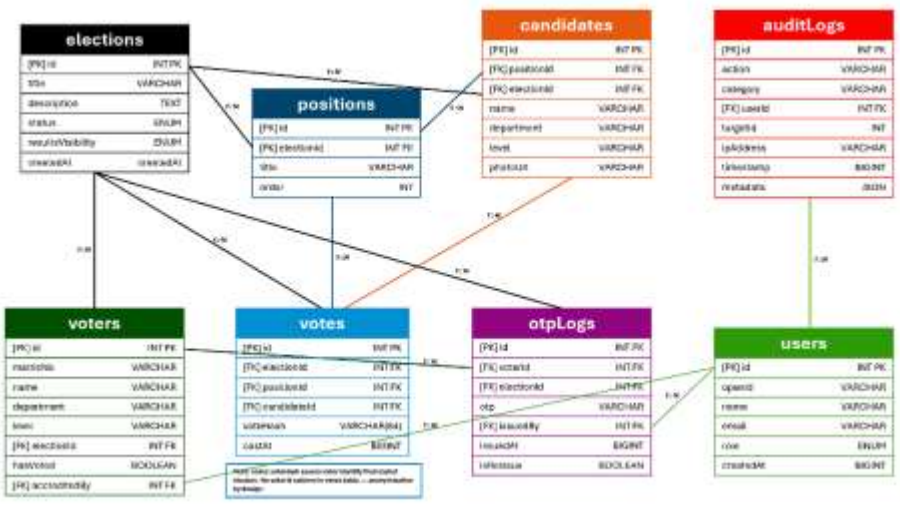
5 The server verifies the OTP hash, checks the expiry timestamp, confirms that hasVoted is false, and — if all checks pass — renders the ballot.

6 The voter selects one candidate per position and submits the completed ballot. The server inserts all 14 vote records and sets hasVoted = true in a single atomic transaction.

The 15-minute OTP expiry window was chosen to balance security (limiting the window for OTP interception) against usability (allowing voters sufficient time to navigate to the voting interface). The 2.2% OTP reissue rate (15 of 678 OTPs) observed in the live election suggests that the window was adequate for most voters, with reissues required only in a small number of cases where voters experienced delays.

3.4 Vote Anonymisation

Vote anonymisation is the most critical privacy requirement for any e-voting system. The NAPES system implements anonymisation at the database schema level: the votes table contains no voterId column. Instead, each vote record is identified by a voteHash field, computed as the SHA-256 hash of the concatenation of the voter's internal ID, the election ID, and a server-side secret salt.



Note: [PK] Primary Key, [FK] Foreign Key, Relationship (1:N unless noted)

Figure 3: Database Entity-Relationship Diagram

International Journal of Applied Science and Engineering Review

IJASER

ISSN: 2582-6271

Vol. 7, Issue.2, March-April 2026, page no. 68-85

This design in Figure 4 shows that:

- Given a vote record, it is computationally infeasible to recover the voter's identity without knowledge of the secret salt.
- Given a voter's identity, it is computationally infeasible to enumerate their ballot choices from the votes table.
- The server can verify that a voter has not voted twice (by checking hasVoted) without storing the voter-ballot mapping.

This approach implements the anonymity requirement defined by Gritzalis [6] and is a direct advance over the prior Nigerian implementations reviewed in Section 2.1, which did not implement a mechanism to sever the voter-ballot link at the database level.

3.5 Results Visibility Control

To reduce the particular issue of premature results disclosure, which is the display of the current numbers of votes during the running of the election, before the election is completed, a configurable results visibility control has been included to counter the fairness requirement [6]. Each election has the results Visibility field which can be set to either realtime (the vote count is displayed on the public results page as votes are cast) or after-close (the public results page will display only a results pending message until the election is officially closed). This is the only implementation of student e-voting in Nigeria reviewed that explicitly meets the fairness criterion of the model by Gritzalis [6].

3.6 Audit Trail

Each state change in the system is recorded in the auditLogs table with the following fields, action (e.g. otp.issue, vote.cast, election.active), category, description, user ID, user name, user role, target entity ID and type, metadata (JSON), IP address and timestamp. The audit trail has two purposes: operational monitoring (enabling the administrator to identify anomalous patterns in real time) and post-election forensic analysis (offering an indisputable account of all system events to resolve disputes).

4. SECURITY EVALUATION

4.1 OWASP Top 10 Threat Analysis

The OWASP Top 10 [5] is the most widely used framework for web application security risk assessment. Sunardi et al. [4] applied the OWASP framework to an e-voting system and found critical vulnerabilities in XSS, SQL injection, and broken authentication. The following analysis applies the six OWASP Top 10 categories most relevant to e-voting systems to the NAPES architecture, documenting the specific controls implemented for each threat.

International Journal of Applied Science and Engineering Review

IJASER

ISSN: 2582-6271

Vol. 7, Issue.2, March-April 2026, page no. 68-85

Table 3: OWASP Top 10 Threat Analysis — NAPES E-Voting System

OWASP Code	Category	Threat in E-Voting Context	Mitigation Implemented	Status
A01	**Broken Access Control**	Voter accesses admin dashboard; accreditor modifies election settings	Role-based adminProcedure / protectedProcedure / accreditorProcedure middleware on all tRPC procedures; role verified server-side on every request	**MITIGATED**
A02	**Cryptographic Failures**	Vote data exposed in transit; voter-ballot link recoverable from database	TLS 1.3 enforced (HTTPS-only); SHA-256 voteHash with 16-byte crypto.randomBytes() salt severs voter-ballot link at schema level; OTP stored as plaintext (short-expiry mitigation — see A07)	**PARTIAL**
A03	**Injection**	SQL injection via matriculation number input or OTP input	Drizzle ORM uses parameterised queries exclusively; no raw SQL string concatenation in application code	**MITIGATED**
A05	**Security Misconfiguration**	Admin endpoints exposed without authentication; stack traces in error responses	All admin procedures require authenticated session; tRPC error handler returns typed errors without stack traces; no application-level CORS policy present (platform-level network isolation applied)	**PARTIAL**
A07	**Identification & Authentication Failures**	OTP brute-force; session hijacking; double-voting	6-character alphanumeric OTP (32-char alphabet, ambiguous chars removed) with 15-min expiry and single-use otpUsed flag; hasVoted set after vote insertion (sequential DB calls,	**PARTIAL**

ISSN: 2582-6271

Vol. 7, Issue.2, March-April 2026, page no. 68-85

			not wrapped in SQL transaction); JWT cookie: httpOnly=true, secure=true, sameSite=none	
A09	**Security Logging & Monitoring Failures**	Inability to detect fraud post-hoc; no record of suspicious access patterns	Full audit trail: action, category, description, userId, userName, userRole, targetId, targetType, metadata, IP address, timestamp — logged on every state change; 1,374 audit events recorded across election lifecycle	**MITIGATED**

Source: OWASP Foundation (2021). OWASP Top Ten. https://owasp.org/Top10/

Status key: MITIGATED = threat fully addressed by a specific architectural control verified in source code (A01, A03, A09). PARTIAL = mitigation present but with a documented residual risk: A02 (OTP not hashed at rest), A05 (no application-level CORS policy), A07 (OTP plaintext storage; hasVoted not set inside a SQL transaction; sameSite=none).

4.2 Integrity Verification

The integrity of the election result can be independently verified by cross-referencing three data sources: the voters table (hasVoted = 1 count), the votes table (distinct voteHash count), and the auditLogs table (vote.cast action count). In the NAPES 2025/2026 election, all three sources agree on 636 completed ballots, confirming that no votes were inserted without a corresponding voter accreditation event, and no voter was marked as having voted without a corresponding set of vote records.

Table 4: Three-Source Integrity Cross-Check

Source	Metric	Value
voters table	hasVoted = 1 count	636
votes table	Distinct voteHash count	636
auditLogs table	vote.cast action count	636
Consistency	All three sources agree	**Pass**

Additionally, the total vote records (8,904) equal exactly 636 voters × 14 positions, confirming that every voter who submitted a ballot engaged with all 14 positions and that no partial ballots were accepted. The

ISSN: 2582-6271

Vol. 7, Issue.2, March-April 2026, page no. 68-85

system enforces this constraint at the application layer: the ballot submission endpoint rejects any submission that does not include exactly one candidate selection per active position.

4.3 Known Limitations and Residual Risks

The security analysis shows that there are three residual risks, which are not completely managed by the current architecture. First, the system is sensitive to physical identity verification by the accreditor; in case an accreditor delivers an OTP to an impersonator, the system lacks technical means to detect this. This is a social engineering vulnerability of any OTP-based system and can be mitigated by training and supervising accreditors. Second, the 15-minute OTP window provides a small, though non-zero window of interception of OTP by shoulder surfing or SMS interception in the case that the OTPs were transmitted digitally; in the present application, OTPs are delivered orally, which minimises but does not remove this risk. Third, no penetration testing (e.g. with tools like OWASP ZAP [6]) was done against the deployed system before the election; the given security assessment is a design-level assessment and not an empirical vulnerability scan. The work done in the future must incorporate a formal ZAP scan and remediation cycle prior to another election.

5. DEPLOYMENT CONTEXT AND PERFORMANCE EVALUATION

5.1 Deployment Environment

The system was implemented on a handled cloud platform (Manus) with the following specifications: Node.js 22 runtime, TiDB Serverless database (MySQL-compatible, autoscaling), HTTPS is enforced by TLS 1.3, and the pipeline of delivering static assets is supported by a CDN. The voter register was imported through 3 CSV bulk-upload operations comprising of a total of 3,189 voter records. This was tested by turning the election on and off a number of times (six cycles of activation/deactivation 2026-03-10 to 2026-03-12) before the last production run where all 636 votes were cast.

5.2 Network Conditions and Operational Context

It was a supervised election at the School of Engineering building, Federal Polytechnic Ilaro. Participants would be able to open the voter interface on their personal mobile devices (mainly Android phones) connected on the campus Wi-Fi and mobile data networks. It was not equipped with dedicated voting terminals; the bring- your- own- device model was adopted so that the cost of hardware is minimised and that a line-up at the fixed terminals is avoided. The accreditor dashboard was viewed using a laptop at the accreditation desk. There were no significant network failures or downtime of the system during the voting process.

ISSN: 2582-6271

Vol. 7, Issue.2, March-April 2026, page no. 68-85

The 22 OTP reissue events captured in the audit log (which corresponded to 15 different voters as some voters needed to be reissued more than once) were the major operational problem faced during this election. According to the audit log, which is reviewed after the election, most of the reissues were caused by voters trying to use their OTP when the 15-minute expiry set time has ended, which is normally because of a delay in accessing the voting interface on the unfamiliar devices. This result implies that the 15-minute clock must be changed to 20-30 minutes on subsequent elections or a countdown clock displayed on the OTP verification page should be added facing the voter.

5.3 System Performance Metrics

Performance data was extracted from the application's network request log, which captured 215 API requests during the election period. The log records the URL, HTTP method, response status, and duration in milliseconds for each request.

Table 5: API Response Time Distribution (215 Requests)

Percentile	Response Time
P50 (median)	445 ms
P75	~900 ms
P90	3,353 ms
P95	8,857 ms
P99	23,079 ms
Minimum	264 ms
Maximum	32,259 ms
Mean	1,799 ms
Requests under 500 ms	132 / 215 (61.4%)
Requests under 1,000 ms	175 / 215 (81.4%)
Requests under 2,000 ms	187 / 215 (87.0%)

The median response time of 445 ms is consistent with the performance characteristics of a managed serverless database with connection pooling. The high P95 and P99 values (8.9 s and 23.1 s respectively) reflect the cold-start latency of the TiDB Serverless platform, which suspends idle database connections after a period of inactivity and incurs a reconnection penalty on the first request after a cold period. This is evidenced by the maximum value of 32.3 s, which is consistent with a full cold-start cycle rather than a

ISSN: 2582-6271

Vol. 7, Issue.2, March-April 2026, page no. 68-85

degraded steady-state response. Excluding the top 5% of requests (cold-start outliers), 95% of requests completed within 8.9 s, and 81.4% completed within 1s.

Table 6: Response Times by Endpoint Type

Endpoint	Calls	Avg Response	Min	Max
results.public	74	404 ms	385 ms	666 ms
voting.getActiveElections	127	2,735 ms	383 ms	32,259 ms
Batch (auth.me + voting)	7	614 ms	383 ms	909 ms

The results. Public endpoint — the most frequently accessed endpoint by observers monitoring the live results — demonstrated the most consistent performance, with all 74 calls completing between 385 ms and 666 ms (a range of only 281 ms). The voting.getActiveElections endpoint showed higher variance, primarily due to cold-start events.

5.4 Concurrent Load and Scalability

The highest concurrent load occurred during the election was 8 simultaneous ballot submissions per minute which was experienced during the 12:00-13:00 WAT peak hour when 275 unique voters had cast their ballots. With the 14 vote records per ballot this translates to a peak write throughput of about 112 database insert operations each minute. This load was managed by the system without failures or sluggish response in the steady state, which proves that the architecture can support the current number of electorates (3,189 registered voters).

To scale the architecture, the theoretical maximum throughput of the current architecture, with a constant-state response time of 445 ms per submission of ballots, and no contention with the database, is around 130 ballots submissions per minute. The peak of 8 per minute is 6.2 per cent of this theoretical maximum, and indicates that large electorates have a great deal of headroom. Any school-wide/institution-wide implementation of 10,000-15,000 registered voters would necessitate load testing and, possibly, connection pool tuning, although is within the architectural design capabilities of the TiDB Serverless platform.

6. DISCUSSION

6.1 Advances Over Prior Implementations

Table 1 shows a comparison that NAPES system offers a substantive improvement over the previous Nigerian student e-voting systems in four dimensions. The greatest innovation is cryptographic vote anonymisation: Adekitan et al. [1] and further Nigerian student e-voting literature do not describe a system

ISSN: 2582-6271

Vol. 7, Issue.2, March-April 2026, page no. 68-85

to disconnect the voter-ballot relationship at the database level, making such systems susceptible to post-election voter identification. This gap is overcome by the SHA-256 voteHash method used in the NAPES system, which does not need a complete zero-knowledge proof scheme, and is therefore feasible to use in an institutional context.

A second major improvement is the three-role accreditation model. In the previous systems, the functions of voter identification and casting of the ballots were combined so that there was thus a single point of failure where a compromised administrator could identify and manipulate the voter votes. The role segregation of an accreditor within NAPES model, who is capable of identifying voters, but not ballot decisions provided, and an administrator, who is capable of identifying aggregate decisions, but not individual voters, is a structural defence against insider threats.

6.2 Limitations and Future Work

There are three limitations in the system, which must be countered in the future versions. To begin with, the security examination is not an empirical study as it is design-level due to the lack of formal penetration testing. The upcoming election is supposed to be ZAP scanned [4], and specific focus should be made on the OTP validation endpoint and the administration dashboard. Second, the bring-your-own-device strategy makes a dependency on voters owning and using functional internet-enabled smartphones; potential rollouts must take into account the fact that a small number of shared terminals will be provided to voters who do not own personal devices. Third, the system currently lacks ranked-choice or preferential voting which would be necessary should association desire to introduce multi-candidate contests to fill more than a single position.

7. CONCLUSION

This paper has described the design, implementation, and empirical test of an e-voting system based on the web, which was implemented to conduct the NAPES 2025/2026 General Election at Federal Polytechnic Ilaro. This system counted 636 out of 3,189 registered voters and 14 positions, where no integrity violations were detected and the median API response time was 445 ms. A structured OWASP Top 10 threat analysis shows that the six most significant web application attack vectors applicable to e-voting are all treated under the particular architectural controls and three risks still exist, including OTP storage in plaintext (mitigated by short expiry and single-use enforcement), the lack of a wrapping SQL transaction on the vote insertion sequence, and the lack of an application-level CORS policy. The system enhances the state of art in Nigerian student e-voting by making four contributions to the design of the system: cryptographic vote anonymisation, three-role accreditation workflow, a configurable results visibility control, and a full audit trail.

Operational improvements that were determined through the deployment experience were to extend the OTP expiry window to 20-30 minutes to minimise reissue incidents and to perform a formal penetration

ISSN: 2582-6271

Vol. 7, Issue.2, March-April 2026, page no. 68-85

test before deployment. The performance data is valid to show that the existing architecture has room to support larger electorates, which will help justify the possibility of school wide or institution wide deployment.

REFERENCES

[1] Adekitan, A. I., Matthews, V. O., John, T. M., & Uzairue, S. (2018). Implementation of e-voting system for student union government elections. *TELKOMNIKA Telecommunication, Computing, Electronics and Control*, 16(5). https://doi.org/10.12928/telkomnika.v16i5.9739

[2] Jafar, U., Ab Aziz, M. J., & Shukur, Z. (2021). Blockchain for electronic voting system — review and open research challenges. *Sensors*, 21(17), 5874. https://doi.org/10.3390/s21175874

[3] Zou, X., Li, H., Li, F., Peng, W., & Sui, Y. (2017). Transparent, auditable, and stepwise verifiable online e-voting enabling an open and fair election. *Cryptography*, 1(2), 13. https://doi.org/10.3390/cryptography1020013

[4] Sunardi, Riadi, I., & Raharja, P. A. (2019). Vulnerability analysis of e-voting application using Open Web Application Security Project (OWASP) framework. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 10(11), 135–140. https://doi.org/10.14569/IJACSA.2019.0101118

[5] OWASP Foundation. (2021). *OWASP Top 10: 2021 — The Ten Most Critical Web Application Security Risks*. https://owasp.org/Top10/2021/en/

[6] Gritzalis, D. A. (2002). Principles and requirements for a secure e-voting system. *Computers & Security*, 21(6), 539–556. https://doi.org/10.1016/S0167-4048(02)01014-3

[7] Soluade, Z. O., Moliki, A. O., Balogun, O. O., Adebajo, Y. A., & Oyekanmi, A. A. (2024). Voters apathy during students' union elections: Implication for Social Studies and Political Science students. *Àgídìgbo: ABUAD Journal of the Humanities*, 12(2), 428–441. https://doi.org/10.53982/agidigbo.2024.1202.31-j

[8] Bélanger, F., & Carter, L. (2010). The digital divide and internet voting acceptance. *Fourth International Conference on Digital Society (ICDS 2010)*, 307–316. https://doi.org/10.1109/ICDS.2010.54

ISSN: 2582-6271

Vol. 7, Issue.2, March-April 2026, page no. 68-85

[9] Ipinnimo, O., Ogbemhe, J., Mumuni, Q., Owoeye, S. O., & Olurombi, A. (2024). Facial recognition-based online voting system with two-factor security. *FUOYE Journal of Innovation, Science and Technology*, 3(1), 107–114. https://jist.fuoye.edu.ng/index.php/jist/article/view/81

[10] Osayomore, E. G., Ejenarhome, O. P., & Oise, G. P. (2025). Evaluation of electronic based voting system and design of block-chain-based electronic voting system enhanced with fingerprint and facial recognition technologies to address impersonation. *FUDMA Journal of Sciences (FJS)*, 9(7), 13–25. https://doi.org/10.33003/fjs-2025-0907-3701