



Republic of the Philippines  
**North Eastern Mindanao State University**

<b>PROJECT TITLE</b>	<b>:</b>	<u>NEMSU, Tagbina campus – Voting System</u>
<b>LEADER</b>	<b>:</b>	<u>Jansil T. Gasoc</u>
<b>MEMBERS</b>	<b>:</b>	<u>Ronquillo, Adrian &amp; Saligumba Niel Patrick</u>
<b>PROJECT DURATION</b>	<b>:</b>	<u>4 Months</u>

## **Abstract**

This project seeks to create and implement a safe, effective, and user-friendly University Online Voting System that simplifies the student election process without compromising transparency, integrity, and accessibility. As digital transformation in higher education institutions becomes increasingly essential, the new system provides a contemporary solution that supersedes manual and paper-voting with a complete web and mobile-based system. The system will support real-time voting, secure login and two-factor authentication, role-based access for administrators, candidates, and voters, result tracking, and monitoring in real-time. It will be able to integrate with the existing student database of the university so that only eligible students are allowed to vote. Administrative users can manage candidate registration, view the progress of the elections, and check analytics through a specific dashboard. The project is designed in five phases over a period of 16 weeks—initially planning and analysis, then configuration of the system, testing, deployment, and ongoing assessment. Major features are secure voting casting, easy-to-use interfaces on devices, extensive audit logs, and strong data protection protocols. Maintenance and optimization after deployment will keep the system efficient and responsive to future requirements. Through the adoption of this solution, the university will gain higher student participation, minimal administrative burden, enhanced election transparency, as well as an expandable platform to be used for other forms of voting apart from general elections. The system will be built keeping in view sustainability as well as long-term use to open doors towards a stable and contemporary digital election framework. The system successfully delivers on many anticipated benefits including user-friendly interfaces, automation of processes, and enhanced efficiency. However, stakeholder concerns about accuracy and security must be addressed to ensure confidence and trust in the election process.

## **Introduction**

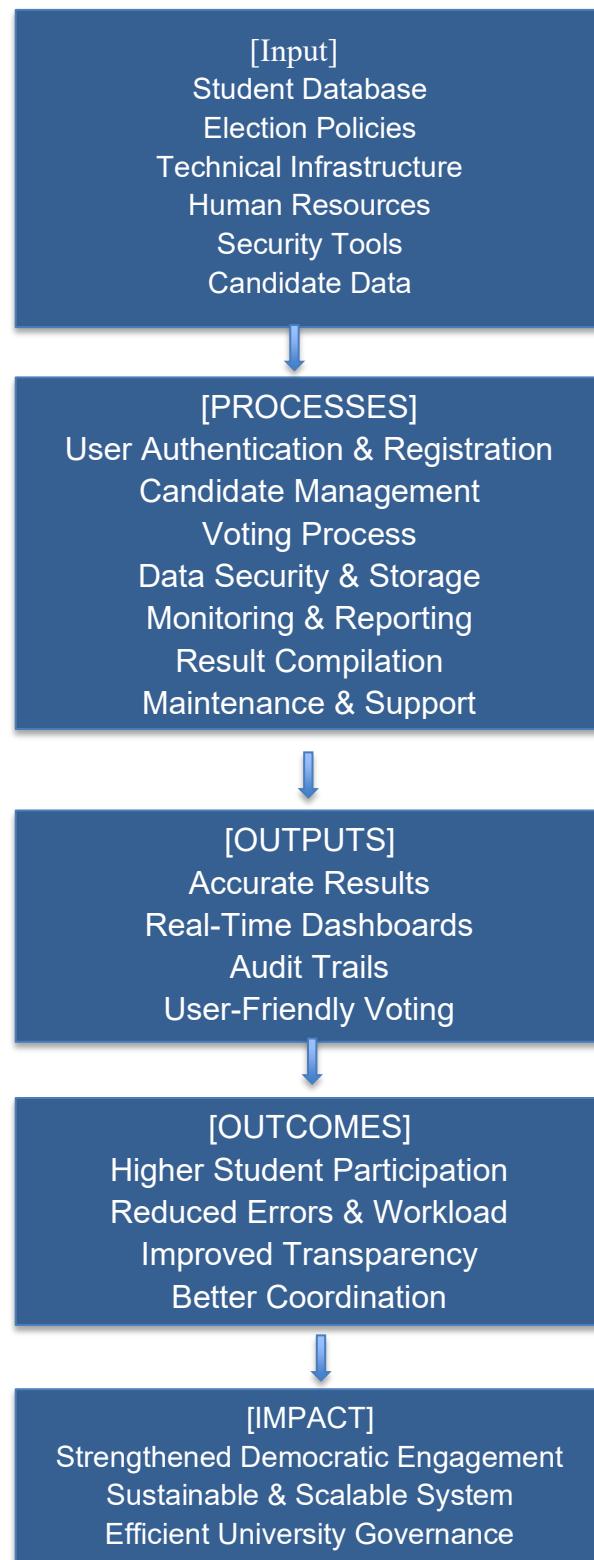
Creating and utilizing a Voting System in our university is a strategic move with multiple advantages. In the first place, it boosts efficiency by mechanizing the process of voting, eliminating long lines and manual vote counting, and providing accurate and real-time count of votes. This reduces mistakes and eliminates miscounted and lost votes. The system increases transparency and credibility, as the results can be tracked and authenticated electronically, thus maintaining integrity in the election process.

Productivity is enhanced with the digital system automating several processes like candidate handling, voter registration, and generating results, freeing admin to better organize and oversee the elections. The voter experience is greatly enhanced through an easy-to-use interface, quicker voting procedures, and immediate acknowledgement of their vote being tallied. Increased participation and a more active student body follow.

The system facilitates enforcement of university policies and election rules, while comprehensive audit trails and reports ensure transparency and accountability. Better coordination and communication among the candidates, voters, and administrators is another major benefit. The system also solves the security issue by authenticating user input, encrypting the vote data, and avoiding repeated voting. Finally, the system is flexible, so it can be a long-term solution as the university expands and election requirements change. Fundamentally, a university-wide Voting System is a holistic investment that deepens democratic engagement, improves operational effectiveness, and fosters equality and integrity in student government.

## Conceptual Frameworks

The conceptual framework illustrates how the University Online Voting System transforms inputs (resources and data) through processes into outputs, outcomes, and long-term impact. It shows the logical flow of the project



## **Significance of the Study**

The implementation of a University Online Voting System carries significant benefits for USG within the university community, including students, administrators, and the institution as a whole.

For students, the system enhances participation by providing an easy-to-use, accessible online platform that encourages greater involvement in elections, fostering a stronger sense of civic engagement within the campus. The convenience and accessibility of being able to vote from any location using web or mobile devices eliminates the need for long queues and physical presence, while transparency and trust are maintained through real-time election results and verification measures that ensure confidence in the integrity of the voting process.

University Student Government benefits from the efficiency and accuracy that comes with automating vote collection, tallying, and reporting, which significantly reduces human error and administrative workload. Real-time monitoring capabilities allow admin to track election progress, voter turnout, and anomalies immediately, enabling quick intervention if issues arise.

For candidates and student government, the system ensures fair and transparent competition through digital records and audit trails that promote trust among all participants.

## **Objectives**

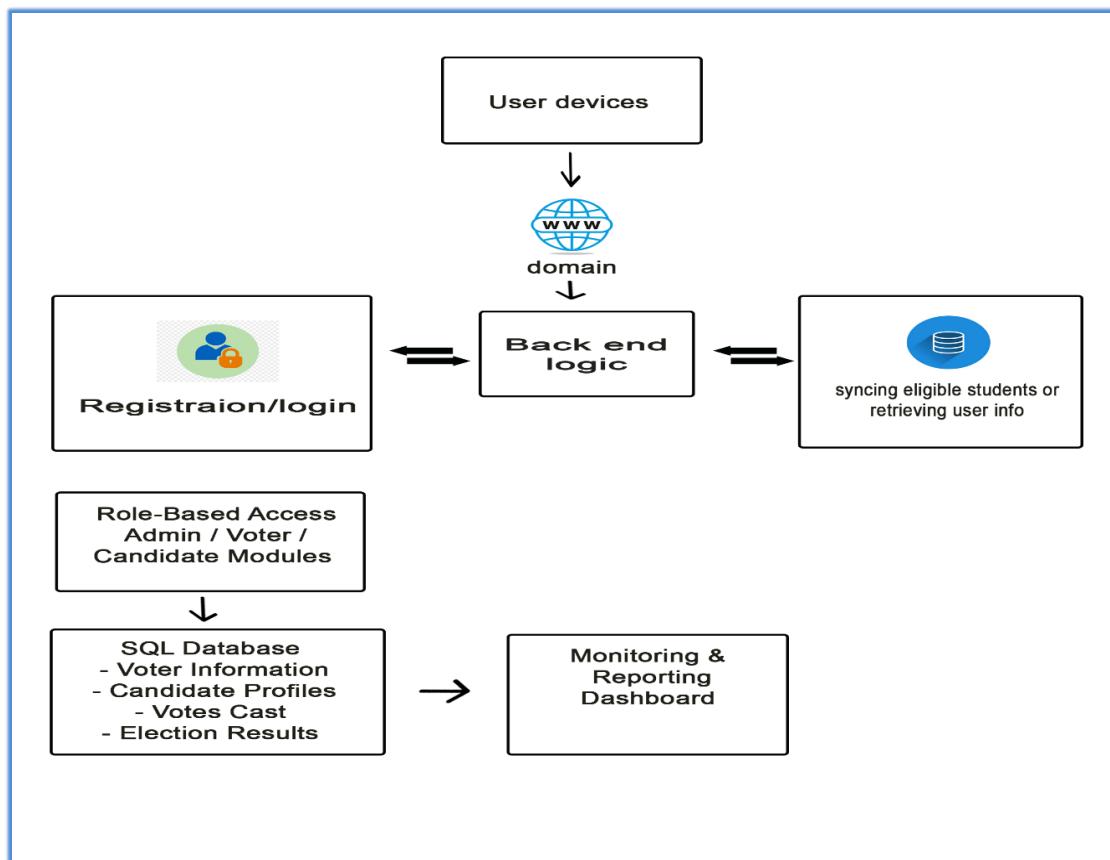
### **Primary Objectives of the University Voting System:**

- **Develop a user-friendly digital voting system** to facilitate smooth and efficient student elections.
- **Eliminate manual vote counting** and minimize human errors in the voting and tallying process.
- **Automate voting processes**, including voter registration, authentication, and real-time vote counting.
- **Integrate reporting and result** to provide transparency

## Methodology

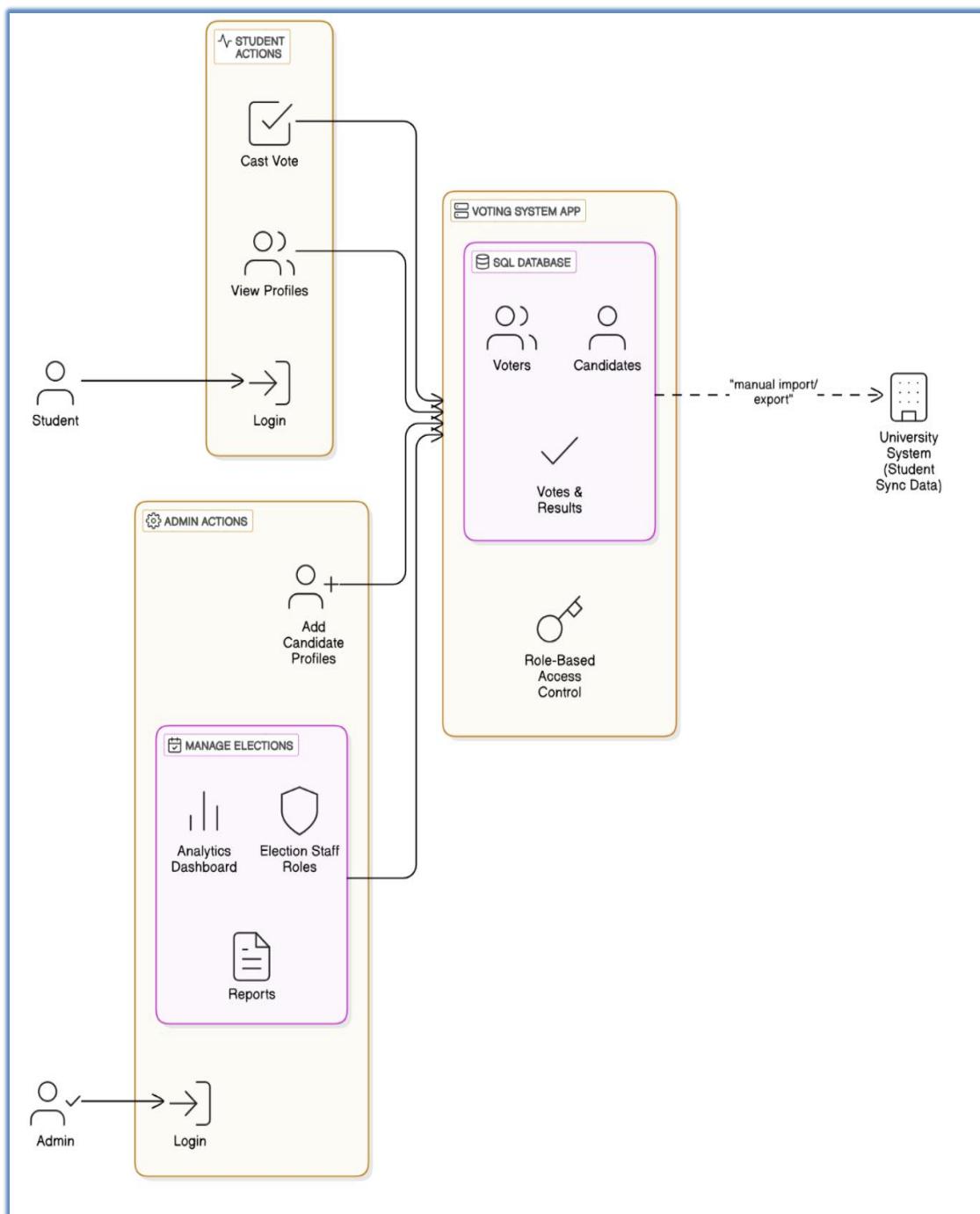
The University Online Voting System methodology describes the methodical process used to create, develop, implement, and assess a safe and effective online voting platform. In order to guarantee that the system satisfies the university's election requirements, this study uses an organized development process that incorporates both technical and administrative considerations. Two thorough testing methodologies were used to assess the system's performance and functionality: survey evaluation based on ISO quality standards and manual functional testing using test cases.

## System Architecture



This web-based student voting system uses a layered architecture to ensure secure and efficient elections. Users access the platform through any device via a web domain, connecting to a central back-end logic server that manages all operations. The system includes a registration/login module for authentication, role-based access control that separates Admin, Voter, and Candidate functions, and integration with the university's database to validate eligible students. All data—voter information, candidate profiles, votes, and results—is stored in a centralized SQL database that maintains integrity and audit trails. A monitoring dashboard provides real-time analytics and oversight by querying the database. The architecture ensures secure access through role-based permissions, seamless data synchronization with external university systems, and transparent election monitoring, creating a reliable and scalable voting solution.

## Use Case Diagram



This is a Use Case Diagram that illustrates the interactions between different actors (users) and the functionalities (use cases) of a student voting system.

## **Actors**

### 1. Student

The primary user who participates in elections by voting and viewing candidate information.

### 2. Admin

The system administrator manages elections, candidates, and students or voters and the voting process.

### 3. University System

A built-in database for students' information that validates student data with the voting application.

## **Use Cases**

### Student Use Cases

Login: Students must authenticate to access the system

Cast Vote: Submit a vote for a candidate (requires login)

View Profiles: Browse candidate information before voting (requires login)

### Admin Use Cases

Login: Admins must authenticate separately from students

Add Candidate Profiles: Create and manage candidate information in the system

Analytics Dashboard: Monitor voting statistics and participation rates

Reports: Generate official election reports and results

### System Use Cases

Role-Based Access Control: Security mechanism ensuring users can only access authorized

## **Features**

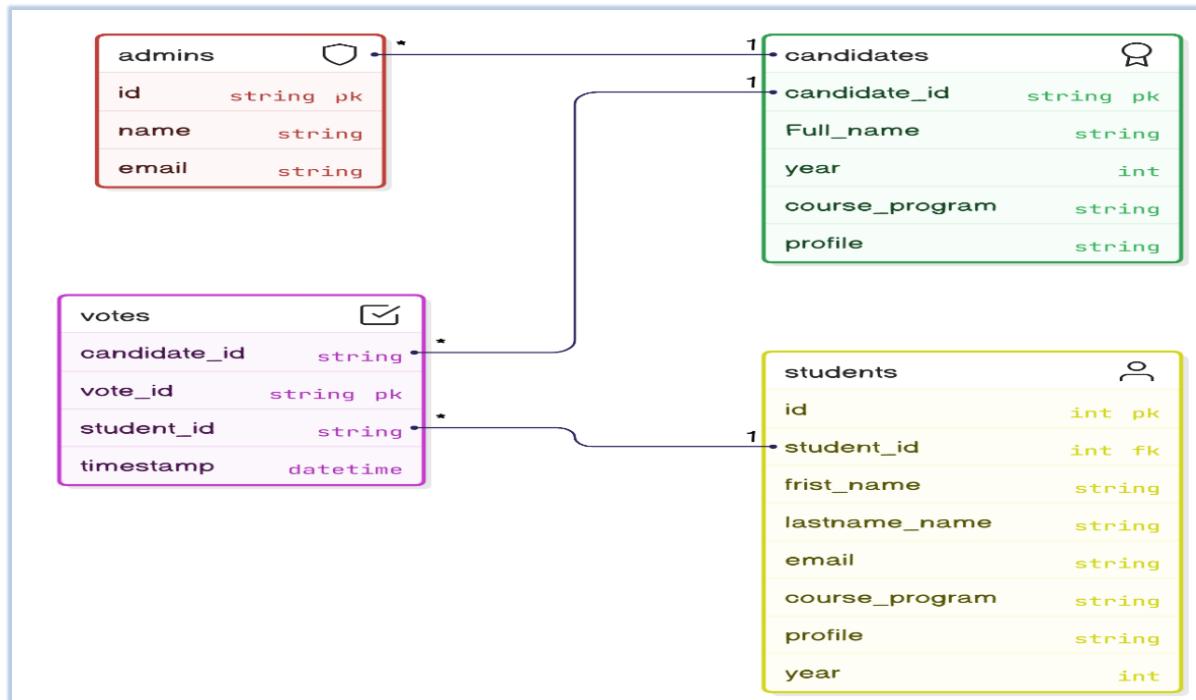
SQL Database (Purple subsection): Core data storage containing:

Voters: Registered voter information

Candidates: Candidate profiles and details

Votes & Results: Vote records and tallied results

## Entity Relationship Diagram



## Relationships

### 1. admins → candidates (1:1)

- Cardinality: One admin can manage one or more candidates
- Meaning: Admins are responsible for adding and managing candidate profiles
- The line shows that one admin record can be associated with multiple candidate records

### 2. candidates → votes (1:Many)

- Cardinality: One candidate can receive many votes
- Relationship: votes.candidate\_id references candidates.candidate\_id
- Meaning: Each vote record is linked to exactly one candidate, but a candidate can have multiple votes

### 3. students → votes (1:Many)

- Cardinality: One student can cast multiple votes (possibly in different elections or positions)
- Relationship: votes.student\_id references students.student\_id
- Meaning: Each vote is linked to the student who cast it, enabling vote tracking and preventing duplicate voting

## User Interfaces

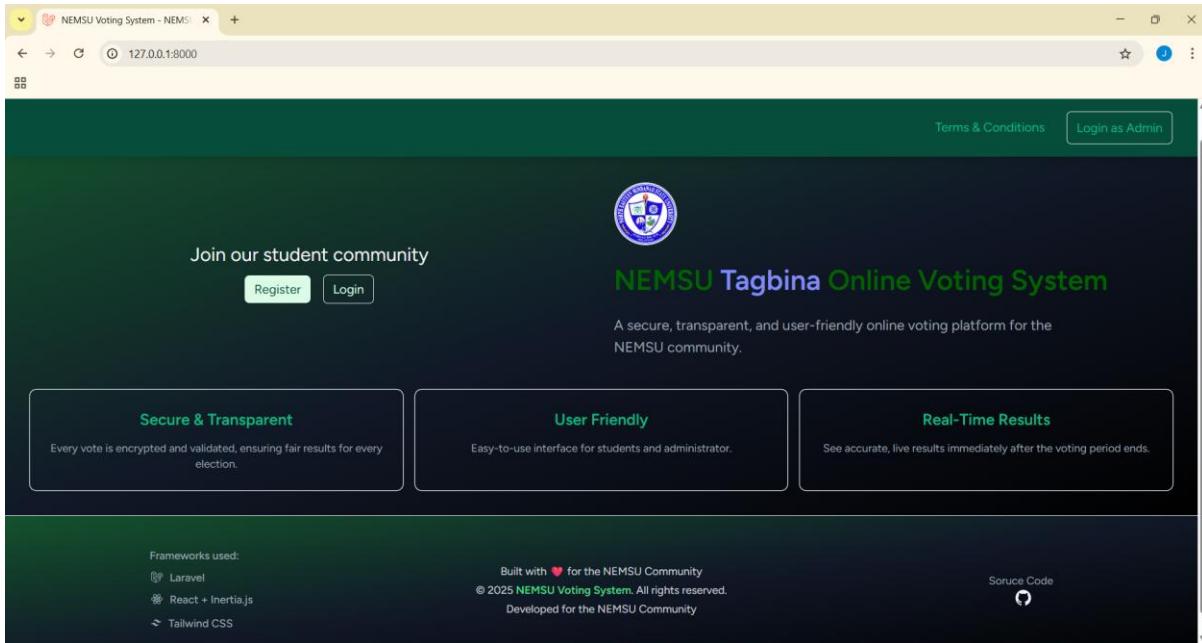


Figure 1 Landing Page

The landing page presents a modern, professional interface with a sophisticated dark color scheme featuring a gradient background

## For Admin Side

The screenshot shows the Admin Panel dashboard. The left sidebar has a dark green background with white text and icons. It includes 'Admin Panel' (with a user icon), 'Dashboard' (highlighted in green), 'Candidates', 'Validate Students', 'Verified Voters', 'Live Tally', and 'Log out'. The main area has a light beige background. It displays a 'Welcome, Test User' message and 'Admin Mode' status. There are several cards: 'Total Verified Students' (1), 'Have Voted' (0), 'Pending' (1), 'PDF Results' (button), 'Click to upload' (button), 'Import validated students' (button), 'Total Current Candidates' (5), 'Set Election' (button), and 'Election Status' (Active).

Total Verified Students	Have Voted	Pending
1	0	1

Total Current Candidates
5

Figure 2 Admin Panel dashboard

The admin can see all verified students, total candidates, download results, pending registration, and control Election management

The screenshot shows the 'Candidates Management' section of the Admin Panel. On the left is a sidebar with options: Dashboard, Candidates (which is selected and highlighted in green), Validate Students, Verified Voters, Live Tally (selected and highlighted in green), and Log out. The main area displays a table titled 'All Candidates' with the following data:

	Name	Position	Party List	Course	Votes	Image
<input checked="" type="checkbox"/>	wew	President	123	BSCS	0	
<input checked="" type="checkbox"/>	wews	External Vice President	123123	BSCS	0	
<input checked="" type="checkbox"/>	1231	Internal Vice President	12313	BSCS	0	
1231	External Vice President	1231	BSCS	0		

Figure 3 Candidate management

- Admin can see candidates, delete or remove them

The screenshot shows the 'Live Tally' view of the Admin Panel. The sidebar includes options: Dashboard, Candidates, Validate Students, Verified Voters, Live Tally (selected and highlighted in green), and Log out. The main area contains four separate bar charts representing different positions:

- President:** A bar chart with two bars: 'wew' (blue) at 0 and 'wews' (red) at 0.
- External Vice President:** A bar chart with two bars: 'wews' (red) at 0 and '1231' (blue) at 0.
- Internal Vice President:** A bar chart with two bars: '1231' (blue) at 0 and '12313' (green) at 0.
- Senator:** A bar chart with one bar: '1323' (blue) at 0.

Figure 4 Live Tally View

- This is where Live tally happens. Every student's vote to the candidates is counted automatically increases

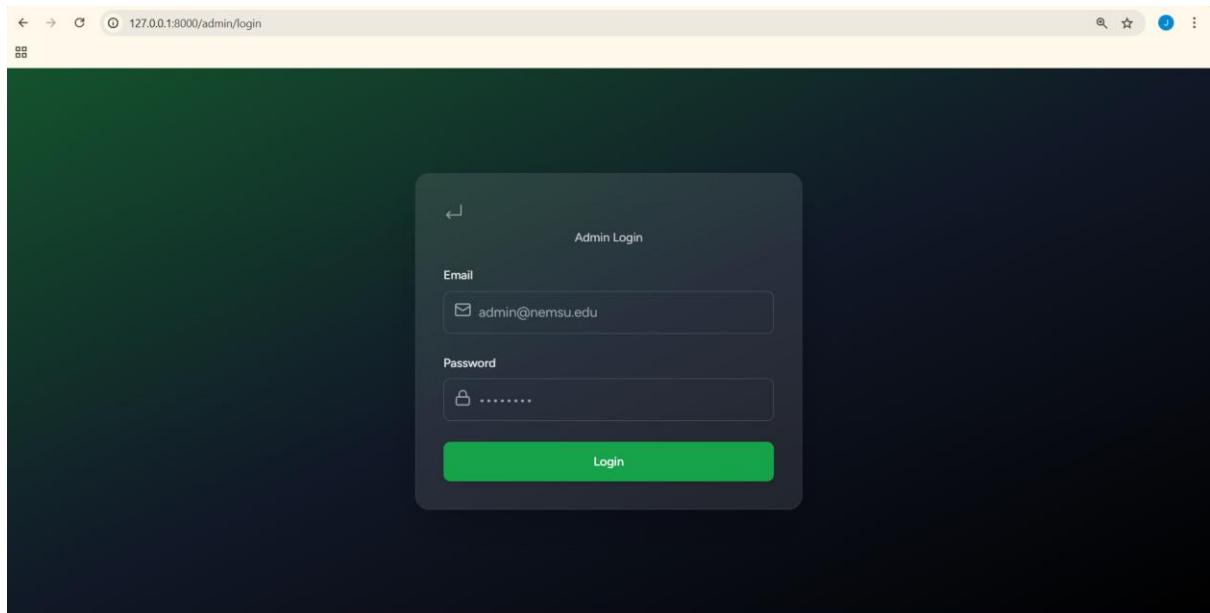


Figure 5 Admin Login Form

- Admin login form only for admin

#### For Voteres or Students Side

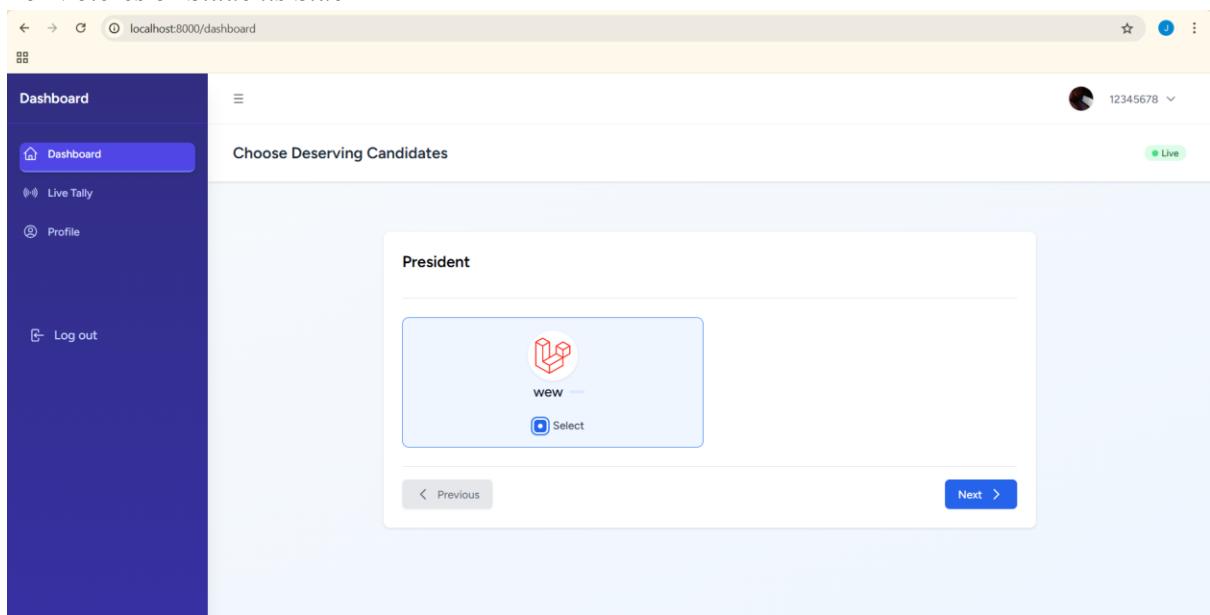


Figure 6 Casting Vote

- Students can choose the candidates they want to vote for

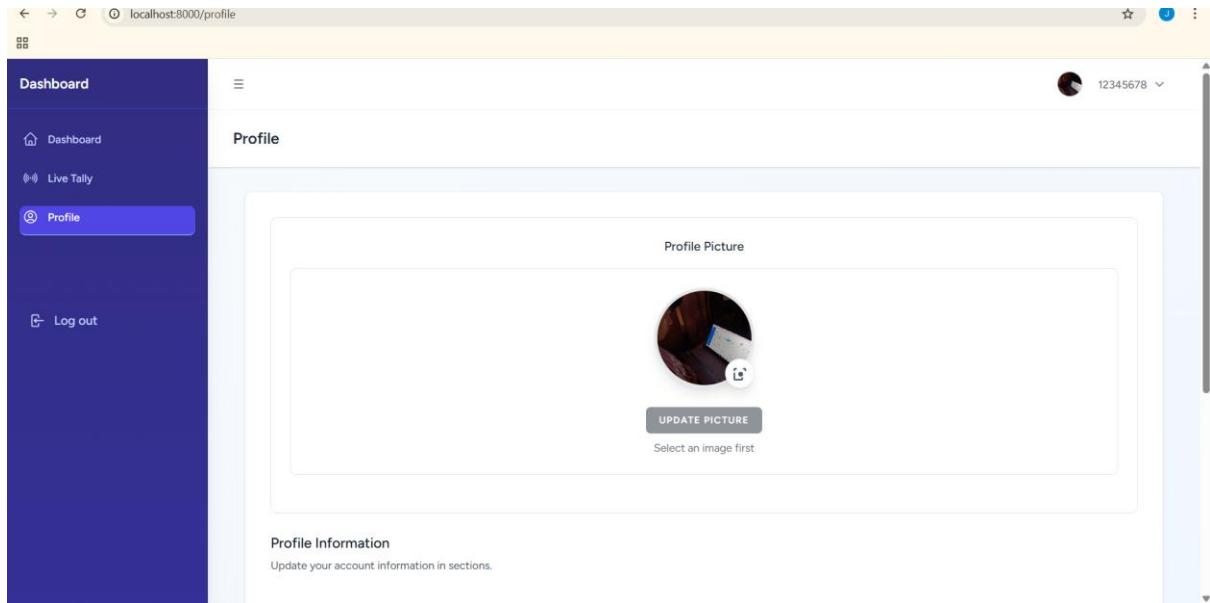


Figure 7 Profile Info View

- students can change their profile or academic information

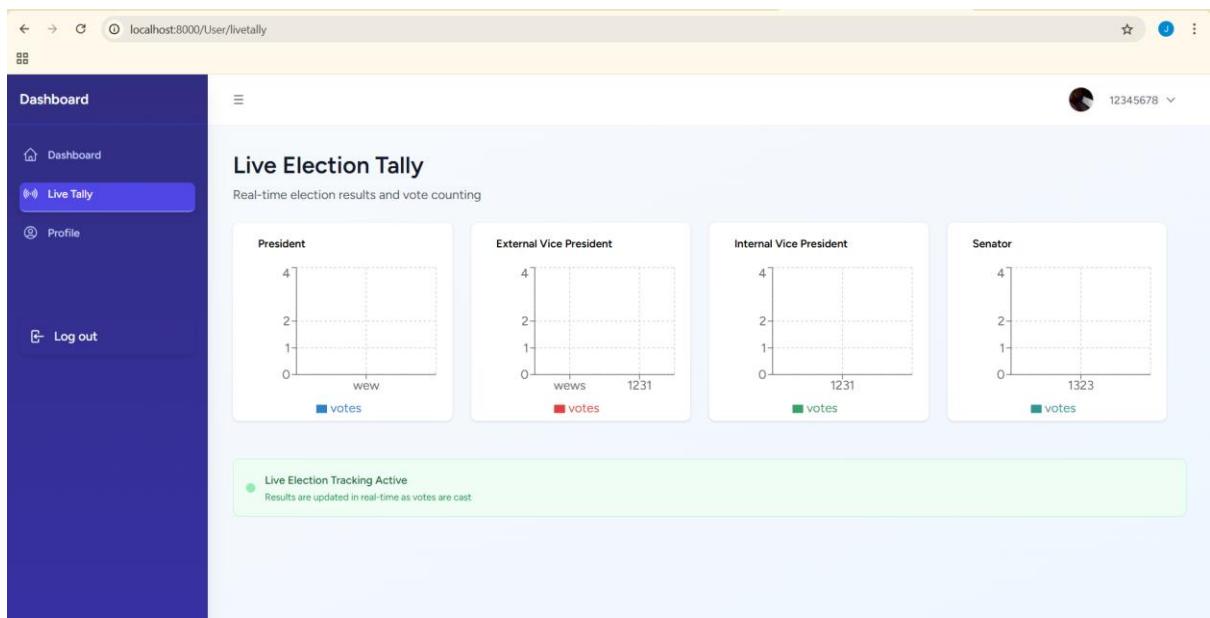


Figure 8 User Live Tally View

- This is where Live tally happens. Every student's vote to the candidates is counted automatically increases

## Testing and Evaluation Tools

### Manual Testing Using Test Cases

The researcher utilized systematic manual testing to ensure all system components functioned correctly. The system was tested using 6 key functional test cases, covering both administrative and voter-related operations. Each test case was executed five times to ensure consistency and reliability across multiple scenarios.

#### Test Case Categories

The testing framework encompassed the following core functional areas:

##### 1. Admin Functions

- **Admin login authentication:** This ensures that only authorized individuals can access the control panel. Testing this prevents unauthorized access, brute force attacks, and privilege escalation, which could lead to election tampering.
- **User privilege management:** This is critical for enforcing the "**principle of least privilege.**" It ensures that admins can correctly assign roles appropriate permissions. A failure could allow a low-level user to perform actions they shouldn't, like modifying an election.

##### 2. Election Management

- **Election setup and configuration:** This ensures that an admin can accurately create an election with all its parameters: title, candidates.
- **Election activation and deactivation:** This tests the state transitions of an election. It verifies that voters cannot vote before the election is active or after it has ended..

##### 3. Candidate Management

- **Candidate registration process:** This verifies that the system allows for candidates to be added correctly, with all necessary information (name, photo, course, position etc...). A flawed process might prevent legitimate candidates from appearing on the form.
- **Candidate removal procedures:** This ensures a candidate can be removed (e.g., if they withdraw) correctly and that this change is reflected accurately on the form *before* voting begins. It also tests that votes for a removed candidate are handled appropriately.

##### 4. Voter Management

- **Voter registration workflow:** This tests that eligible voters can register without undue difficulty and that ineligible voters are correctly rejected. It validates data collection and storage.
- **Voter authentication mechanisms:** This is arguably the most critical security test. It ensures that only registered voters can log in, typically through

- secure credentials (email/password). It must also prevent one person from voting multiple times (e.g., by detecting duplicate voter student ID)
- **Voting process execution:** This validates the entire voting UX: that the form is displayed correctly, that selections are recorded accurately, and that the "cast vote" button finalizes the vote without errors. A failure here could disenfranchise a voter or record their vote incorrectly.
- **Voter data security measures:** This ensures that personal voter data identification detail is encrypted, stored securely, and never exposed in an unauthorized way. It protects voter privacy and complies with data protection regulations.

## 5. Results Management

- **Vote tallying accuracy:** This is the most direct test. It verifies that the system counts every valid vote correctly and does not count invalid votes (e.g., tests for SQL COUNT functions, data aggregation logic). The results must be mathematically perfect.
- **Report generation functionality:** This ensures that the system can produce clear, accurate, and tamper-proof reports for auditors, officials, and the public. These reports are the official record of the election and are essential for audits and building public trust. Testing includes checking for correct data, formatting, and export functionalities

## 6. Voter Authentication and Functionality.

- System Security: Prevents unauthorized access.
- Election Integrity: Ensures "one person, one vote," which is the cornerstone of a fair election.
- Functional Correctness: Confirms that the core voting feature works as intended for a legitimate user.

## Evaluation tools

### 1.1 Data Gathering

This study employed a quantitative survey research design to evaluate student perceptions of a voting software system. The evaluation framework was based on ISO/IEC 25010:2011 software quality characteristics (formerly ISO 9126), adapted specifically for voting system applications. Five quality dimensions were assessed: Functionality, Reliability, Usability, Efficiency, and Maintainability. Portability was excluded as the system is specifically designed as a web-based application accessible through standard browsers.

### 1.2 Survey Instrument

The survey instrument consisted of 20 items distributed across five categories:

- **Functionality** (4 items): Assessed the software's ability to perform required functions accurately and securely
- **Reliability** (4 items): Evaluated system performance consistency and error management
- **Usability** (5 items): Measured user-friendliness and ease of interaction
- **Efficiency** (4 items): Examined learning curve and operational effectiveness
- **Maintainability** (3 items): Assessed system sustainability and technical requirements

Each item was evaluated using a 5-point Likert scale:

- 5 = Strongly Agree
- 4 = Agree
- 3 = Neutral
- 2 = Disagree
- 1 = Strongly Disagree

### 1.3 Data Collection

Sampling Method: The study employed convenience sampling, targeting registered students at NEMSU Tagbina who would be direct users of the voting system. Sample Size: 100 registered students from NEMSU Tagbina Inclusion Criteria: - Currently enrolled students - Registered voters in student elections - Willingness to participate in the evaluation Survey,

### 1.4 Data Analysis

#### Quantitative Analysis:

Mean scores were calculated for each survey item using the weighted average formula:

$$\text{Mean} = (n_1 \times 5 + n_2 \times 4 + n_3 \times 3 + n_4 \times 2 + n_5 \times 1) / (n_1 + n_2 + n_3 + n_4 + n_5)$$

Where:

- $n_1$  = number of "Strongly Agree" responses
- $n_2$  = number of "Agree" responses
- $n_3$  = number of "Neutral" responses
- $n_4$  = number of "Disagree" responses
- $n_5$  = number of "Strongly Disagree" responses

Category means were computed by averaging all item means within each dimension. The overall system mean was calculated by averaging all category means.

#### Interpretation Scale:

- 4.50 - 5.00: Excellent
- 4.00 - 4.49: Good
- 3.50 - 3.99: Satisfactory
- 3.00 - 3.49: Fair
- Below 3.00: Needs Improvement

**Response Distribution Analysis:** Frequency distributions were analyzed to identify patterns, consensus levels, and areas of disagreement among respondents.

### 1.5 Validity and Reliability

**Content Validity:** The survey items were derived from established ISO 9126 quality characteristics, ensuring coverage of critical software quality dimensions relevant to voting systems.

**Reliability Considerations:** The use of multiple items per category (3-5 items) enhances internal consistency and provides more robust measurement of each quality dimension.

### 1.6 Limitations

1. Sample Representation: Findings reflect perceptions of surveyed stakeholders and may not represent all potential user groups
2. Self-Reported Data: Responses based on subjective perceptions rather than objective performance metrics
3. Experience Variation: Respondents may have varying levels of exposure to and experience with the system.
4. Response Bias: Potential for acquiescence bias (tendency to agree) or social desirability bias, where respondents may provide favorable ratings rather than honest assessments.
5. Single Institution: Results based on NEMSU Tagbina students may not generalize to other institutions

## Results and Discussion

### Test Case Results

Table 1  
Test Case Results and Remarks

*Table 1 Test case results*

Unit/Module	Test Case	Step	Case	Result	Number of tests	Rmarks
1. Admin Login	1.Admin login Valid and Invalid Credentials	1. Database Seeding admin account	1. Valid email/password	pass	10	50/50
		2. Navigate to admin login from	2. Invalid email/ valid password	pass	10	
		3. Enter valid admin username	3. Valid email/ invalid apssword	pass	10	
		4. Enter Credentials	4 Invalid email/password	pass	10	
		5. Click login button	5. no existing account data	pass	10	
2. Election Management	1.Set new election, Pause, Start	1. Login as admin	1. Set Election	pass	10	50/50
		2. Navigate to "Set Election" button in Admin Dashboard	2. Deleting or end election		10	
		3. Fill in election details "Election name or title" example: election 2024-2025	3. Set to Pause		10	
		4. Pause, Delelte,	4. Set to Start	pass	10	

		Start buttons appear.				
			5. realtime start and pause	pass	10	
3. Candidate Management	1.Adding candidate to election 2.Remove candidate  3. Candidate Management	1. Login as admin	1. Adding candidtate	pass	10	50/50
		2. Navigate to candidate management	2. Deleting Candidate	pass	10	
		3. Add candidate details (name, party-list, course, position)	3. Bulking Deletion		10	
			4. add candidate without profile	pass	10	
			4. Delete selected candidates	pass	10	
4. Voter Management	1. Student Validation for voter	1.Login as admin	1. Add Student id and firstname without lastname	pass	10	50/50
		2.Navigate to voter Validation	1. Add Student id and lastname without firstname	pass	10	
			3. Add lasstname and firstname without student id	pass	10	
			1. Add Student id, with existing student id	pass	10	
	2 Remove voter account	1. Login as admin	1. Remove voter account	pass	10	
		2. Find voter in system				
5. Results Management	3. View election results	1. Login as admin and goto dashboard	1. Check vote counts		10	50/50
			2 print result direct to printer	pass	10	
	2. Download	1. Lofin as admin and go to	3. export without candidates	pass	10	

	Results or export	dashboard find the Download button				
		4. export results into pdf		pass	10	
		5. View results		pass	10	
6. Voter Authorization and FUNCTIONALITY	1. Voter Registration Valid and Invalid Credentials	1. Navigate to voter register page then input required fields	1. Invalid Student id, firstname, and lastname		10	50/50
	2. Voter Login Valid and Invalid Credentials	2. Navigate to voter login page then input required fields	2. Valid email/password	pass	10	
			3. Invalid email/ valid password		10	
	3. Voting Process	1. Go to main dashboard	4. Cast vote in active election	pass	10	
		2. Review Candidates	5. Attempt to vote twice	pass	10	

The table shows a total of six modules with multiple test cases were executed to validate the election system.

### 1. Admin Login

- Tested with valid/invalid credentials and missing data.
- All 5 cases passed successfully across 10 repetitions each.
- The system correctly handled authentication scenarios, ensuring login security.

### 2. Election Management

- Tested functionalities included setting up new elections, pausing, starting, and deleting elections.
- All 5 cases passed successfully across 10 repetitions each.
- The system consistently displayed correct election control operations (pause, delete, start, real-time status).

### 3. Candidate Management

- Test cases covered adding, deleting, bulk deletion, and handling incomplete candidate profiles.

- All 5 cases passed with full accuracy.
- The system prevented incomplete entries and handled multiple candidate management operations properly.

#### **4. Voter Management**

- Included student validation (with/without complete information) and removing voter accounts.
- All 5 cases passed across 10 tests each.
- The system reliably validated voter data and removed accounts as expected.

#### **5. Results Management**

- Test cases included viewing results, printing, exporting to PDF, and verifying data integrity.
- All 5 cases passed consistently.
- Results were accurate and export functions worked without data loss.

#### **6. Voter Authorization and Functionality**

- Tested voter registration, login (valid/invalid credentials), voting process, and prevention of duplicate voting.
- All 5 cases passed across 10 repetitions each.
- The system successfully blocked duplicate votes and handled incorrect credentials, ensuring election integrity.

The election system demonstrated high reliability and robustness, passing all tests under multiple iterations. Authentication mechanisms (admin and voter login) effectively prevented unauthorized access. Candidate and voter management modules functioned correctly, validating data and preventing incomplete or invalid entries. Election management features (pause, start, delete) worked seamlessly, while the results management module maintained accuracy and supported multiple export formats.

## Survey Results

Table 2  
Functionality, Reliability, Usability, Efficiency, and Maintainability overall Mean

*Table 2 Survey results*

Category	Mean
<b>Functionality</b>	
1. The software delivers the expected results 100% accurate vote count.	4.25
2. The software can run smoothly with other devices.	4.10
3. The software provides Secure login and registration	4.05
4. The software includes buttons and commands that are correctly labeled.	4.20
<b>Total</b>	<b>4.15</b>
<b>Reliability</b>	
1. The software implements real-time tallying for the vote	4.00
2. The software consistently demonstrates quick responsiveness.	4.10
3. The software efficiently manages errors.	4.10
4. The software provides all the necessary information for displaying candidates.	4.30
<b>Total</b>	<b>4.1</b>
<b>Usability</b>	
1. The software completes designated tasks with reliability.	4.15
2. The software features a pleasing interface.	4.05
3. The software meets user-friendly requirements for the voting system	4.10
4. The software effectively modernizes the manual voting system	4.10
5. The application operates smoothly, ensuring a seamless voting process	4.15
<b>Total</b>	<b>4.1</b>
<b>Efficiency</b>	
1. The software is designed to be user-friendly and easy to understand.	4.40
2. Users can quickly learn and navigate the software with ease.	4.40
3. Users can effectively utilize the system without the need for training.	4.05
4. The Vote Tallying feature comprehensively displays relevant candidates vote count.	4.05
<b>Total</b>	<b>4.25</b>
<b>Maintainability</b>	
1. The software operates seamlessly with minimal user intervention.	4.15
2. The software consistently responds quickly.	4.30
3. The system meets the minimum software requirements effectively.	4.20
<b>Total</b>	<b>4.2</b>

The table shows overall mean response by the respondents by each Categories from the following:

## **1. Functionality**

- Mean scores ranged from **4.05 to 4.25**, with the highest score for delivering accurate vote counts (4.25).
- Secure login/registration and properly labeled commands also received favorable ratings.
- Overall functionality mean: **4.15**.
- Interpretation: The system fulfills its intended purpose effectively and provides essential features required for voting operations.

## **2. Reliability**

- Mean scores ranged from **4.0 to 4.3**.
- The highest rating was for displaying all necessary candidate information (4.3).
- Real-time vote tallying received slightly lower but still positive feedback (4.0).
- Overall reliability mean: **4.1**.
- Interpretation: The system demonstrates stable performance and error management, with consistent responsiveness during operations.

## **3. Usability**

- Mean scores ranged from **4.05 to 4.15**.
- The application's smooth operation and reliability received the highest ratings (4.15).
- The user interface was considered pleasing and moderately easy to navigate (4.05).
- Overall usability mean: **4.1**.
- Interpretation: Users found the system user-friendly and capable of modernizing the voting process, though minor improvements in interface design could enhance the experience.

## **4. Efficiency**

- Mean scores were consistently high, ranging from **4.05 to 4.4**.
- The software's ease of use and navigability scored the highest (4.4).
- Effective utilization without training and clear vote tallying features also rated well (4.05).
- Overall efficiency mean: **4.25**.
- Interpretation: The system is highly efficient, requiring minimal training and ensuring smooth and fast operation for voters.

## **5. Maintainability**

- Mean scores ranged from **4.2 to 4.3**, the highest being quick response (4.3).
- Seamless operation with minimal user intervention scored well (4.15).
- Meeting minimum software requirements also rated positively (4.2).
- Overall maintainability mean: **4.2**.
- Interpretation: The system is maintainable, reliable under minimal supervision, and adaptable for continued use.

**Table 3**  
Performance by Category

*Table 3 Survey Category performance*

Rank	Category	Total Score	Performance	Key Strength
1	<b>Efficiency</b>	4.25	Good	Most user-friendly and intuitive
2	<b>Maintainability</b>	4.20	Good	Consistent performance
3	<b>Functionality</b>	4.15	Good	Core features are working well
3	<b>Usability</b>	4.10	Good	Technically sound
5	<b>Reliability</b>	4.10	Good	Strong user experience

The table shows the narrow 0.15-point spread between highest and lowest categories shows balanced quality across all areas. All categories scored above 4.00, indicating consistent "Good" performance across all quality dimensions with no significant weaknesses.

**Table 4**  
Profile of Respondents in terms of Program/Course

*Table 4 Respondents profile*

Program/Course	Frequency	Percentage
BSCS	2	10.00%
BSHM	5	25.00%
CTE	5	25.00%
BAT/BSA	6	30.00%
BSBA	2	10.00%
<b>Total</b>	<b>20</b>	<b>100.%</b>

Stakeholders surveyed included 20 participants from diverse academic programs across the university, representing potential system users, administrators, technical staff, and decision-makers with relevant experience or interest in the voting system. The sample included students from Computer Science (BSCS), Hospitality Management (BSHM), College of Teacher Education (CTE), BS in Agricultural Technology (BAT/BSA), and Business Administration (BSBA) programs, ensuring varied perspectives on system usability and functionality.

## **Conclusion**

The election system was thoroughly tested by functional testing as well as ISO-based questionnaire testing that ensured that the system was reliable, secure, efficient, and easy to utilize. During the execution of the test-case, all six modules (Admin Login, Election Management, Candidate Management, Voter Management, Results Management, and Voter Authorization & Functionality) passed a 100% success ratio over a total of 300 executions without any system failures or errors. This proves that the system is stable, robust, and deployable.

The ISO survey evaluation further validated the system's quality across five key attributes: Functionality (4.15), Reliability (4.1), Usability (4.1), Efficiency (4.25), and Maintainability (4.2). The results indicate that the system is not only functionally correct but also efficient and maintainable, with strong usability and reliability. The highest-rated attribute was Efficiency (4.25), confirming that users found the system intuitive and easy to operate with minimal training.

Overall, the findings conclude that the election system meets the intended objectives and quality standards for software applications. It can be considered deployment-ready for real-world use, though future improvements in real-time vote tallying speed and user interface design could further enhance its effectiveness and user satisfaction.

