Journal of Mathematics Instruction, Social Research and Opinion
Vol. 4, No. 4, December 2025, pp. 1433 – 1442, https://doi.org/10.58421/misro.v4i4.839
ISSN 2962-7842

Web-Based E-Voting System Implementation with Selection-Count Validation for the Formatur Election of PC IPM Cepu

Ronggo Gusti Anllasosa
Informatics, Universitas Siber Muhammadiyah, Indonesia

Article Info

Article history:

Received 2025-11-26
Revised 2025-12-18
Accepted 2025-12-20

Keywords:

E-Voting
Flask
Organizational Digitalization
Selection Validation
SQLite

ABSTRACT

The formatur election at PC IPM Cepu is an important practice of internal democracy within the Muhammadiyah student organization; however, the manual voting process has frequently led to invalid ballots, lengthy vote recaps, and potential recording errors. This study aims to develop a web-based e-voting system that enforces election rules and improves efficiency and data integrity. In this election, voters are required to select exactly nine candidates, which serves as the core validation constraint of the proposed system. The research employs a Research and Development (R&D) method with a prototyping approach, encompassing needs analysis, system design, implementation using the Flask framework with an SQLite database, and system testing. Automatic selection-count validation is implemented on both the client and server sides, along with a double-vote prevention mechanism to ensure voting integrity. System evaluation involved 30 active PC IPM Cepu participants through functional testing, time-efficiency comparison, and a user-satisfaction survey. The results indicate that the system successfully eliminated invalid ballots, achieving 0% input errors, defined as the absence of votes with an incorrect number of selected candidates across all test cases. In addition, the voting and recapitulation process was accelerated by up to 80% compared with the manual method. User satisfaction was measured using a Likert-scale questionnaire, yielding an overall satisfaction rate of 95% (positive responses), derived from aggregated Likert scores. Despite these promising results, this study is limited by its relatively small sample size and its application within a single student organization. Overall, the findings demonstrate that a simple rule-based e-voting system can serve as an effective digitalization model for internal democratic processes in student organizations, enhancing participation and organizational governance efficiency through technology.

Corresponding Author:

Ronggo Gusti Anllasosa
Informatics, Universitas Siber Muhammadiyah, Indonesia
Email: ronggogusti3@gmail.com

1. INTRODUCTION

The internal democratic process is a key part of shaping character and leadership within the Muhammadiyah student organization. The selection of the formator for the Pimpinan Cabang Ikatan Pelajar Muhammadiyah (PC IPM) Cepu determines the direction of leadership for the following organizational period. Manual voting, however, often leads to recurring issues, including delayed recapitulation, inaccurate vote counts, and the risk of fraud due to the lack of a proper validation system.

Baseline evidence from previous PC IPM Cepu elections shows that these problems are empirically observable. Internal committee documentation from the last two election periods indicates that approximately 10–15% of ballots were declared invalid, mainly because voters selected fewer or more candidates than allowed. Furthermore, the manual vote recapitulation process required an average of 2–3 hours, as committee members had to verify ballot validity and correct recording errors repeatedly. These conditions demonstrate that the absence of automatic selection-count validation significantly affects both efficiency and data accuracy in internal organizational elections.

In a digital environment, organizations are pushed to adopt technological solutions to improve efficiency and accountability [1]. Electronic voting, or e-voting, is one approach that can strengthen transparency and speed during the voting process. This method has been widely adopted in government contexts, universities, and student organizations [2], [3]. However, existing studies tend to emphasize security mechanisms such as authentication and encryption, while practical issues related to ballot validity, particularly the incorrect number of selected candidates, remain underexplored. In small-scale student elections, invalid ballots caused by misselection are among the most frequent operational problems, indicating a clear research gap between technical security and practical voting correctness.

E-voting refers to voting conducted via electronic media, such as computers, mobile phones, or web-based platforms. Nugroho and Yuliana explain that e-voting aims to improve speed, accuracy, and overall efficiency during the election process [4]. Conceptually, an e-voting system consists of a user interface, a processing layer, and a database for vote storage. From a theoretical perspective, voting integrity is not only determined by cryptographic security but also by the system's ability to enforce election rules correctly and consistently.

Several prior studies have examined e-voting systems in educational and organizational contexts. Ariyanto and Prasetyo developed a web-based e-voting system using the Laravel framework, focusing on authentication and access control [2]. Hidayat and Ramadhani implemented a dual-validation mechanism to reduce general input errors in student elections [3]. Kurniawan demonstrated that Flask and SQLite are suitable for small-scale elections due to their simplicity and low system overhead [5]. Nevertheless, these studies do not explicitly enforce strict candidate-selection constraints as a core system requirement, leaving room for invalid ballots caused by over-selection or under-selection.

Beyond implementation frameworks, broader e-voting research highlights essential issues related to integrity, privacy, and auditability. Chaum emphasized the importance of voter-verifiable mechanisms to ensure election transparency [8]. Benaloh argued that verifiability and simplicity must be balanced, especially in non-governmental elections [9].

Adida introduced Helios, a web-based e-voting system that demonstrates how auditability can enhance voter trust [10]. Kiayias and Yung discussed how improper ballot construction can undermine election correctness even in secure systems [11]. Park et al. showed that usability flaws in e-voting interfaces often lead to unintentional voting errors [12]. Studies by Cortier and Smyth further stressed that rule enforcement at the system level is critical to preventing invalid votes [13]. In educational environments, simple rule-based validation is more effective than complex cryptographic approaches [14]. Moreover, digital voting systems can function as tools for civic and democratic learning when designed with clarity and accountability in mind [15].

Meanwhile, digital transformation in student organizations is not merely about adopting technology, but also about fostering digital literacy and participatory values [1]. IPM, as a Muhammadiyah student organization, actively promotes technology-based innovation within its organizational activities [7]. E-voting, therefore, serves not only as a technical solution but also as an educational medium that introduces students to integrity, accountability, and responsible democratic participation [16], [17].

Based on these considerations, this study aims to develop a web-based e-voting system that explicitly enforces election rules through automatic vote-count validation. In the PC IPM Cepu formatur election, voters must select exactly nine candidates, and this rule is enforced as a mandatory system constraint at both the client and server levels.

The specific contributions of this study are as follows: Implementation of dual-layer selection-count validation to ensure that ballots are accepted only when exactly nine candidates are selected, Integration of a double-vote prevention mechanism to maintain vote integrity, and Provision of real-time vote recapitulation to significantly reduce counting time and human error.

It is expected that the proposed system will improve efficiency, accuracy, and transparency in PC IPM Cepu elections, while also serving as a practical model of digitalization for other student-based organizations. By emphasizing simplicity, rule enforcement, and accessibility, this research contributes to strengthening internal democratic governance through appropriate and context-aware technological solutions.

2. METHOD

2.1 Type and Approach of Research

This study applies a Research and Development (R&D) method using a prototyping approach. The stages include:

a. Analyzing system requirements based on observations of previous election processes in PC IPM Cepu.
b. Designing the system, including interface design, system architecture, and database modeling.
c. Implementing the system using Python, the Flask framework, and SQLite.
d. Testing the system through functional testing, performance testing, and a user satisfaction survey.

2.2 System Architecture

The web-based e-voting system consists of three main components:

a. Frontend (Client): Built using HTML, CSS (Bootstrap 5), and JavaScript. The interface includes three pages: *index.html* (homepage), *vote.html* (voting page), and *hasil.html* (real-time results page).
b. Backend (Server): Developed with Flask, which handles application logic, and SQLAlchemy as the ORM connecting Python to the database.
c. Database (SQLite): Stores voter data and vote records. The table structure includes:
   i. Voter Table: (id, name, already_vote)
   ii. Vote Table: (id, voter_id, candidate)

2.3 System Flowchart

The flowchart in Figure 3.1 illustrates the workflow of the web-based e-voting system built using Flask and SQLite. It shows the steps taken by voters, from accessing the main page to validating their name, selecting candidates, and saving the results to the database. Each step in the flowchart corresponds to the program logic implemented in the Python code.

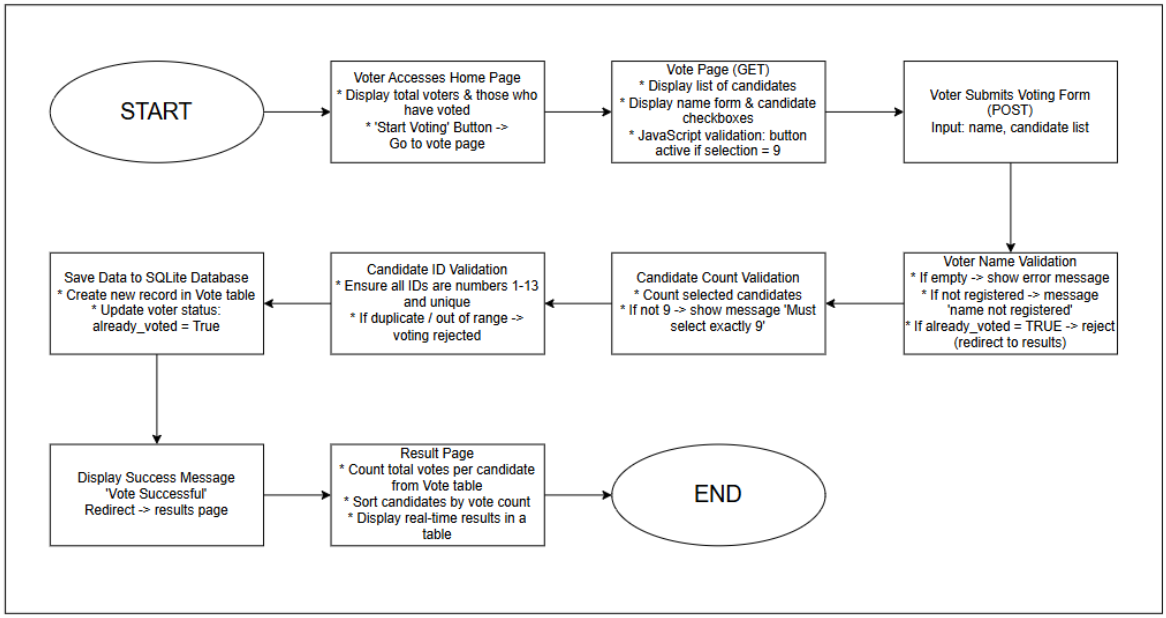

Figure 1. System Process Flowchart

The flowchart highlights that the system applies multi-layer validation on both the client side (via JavaScript) and the server side (via Flask), along with a double-vote prevention feature through checking the *already_vote* status. This process ensures data integrity and efficient digital voting.

2.4 Vote-Count Validation

Validation is performed in two layers to prevent voters from selecting fewer or more candidates than allowed:

a. Frontend validation: JavaScript counts the number of selected candidates and enables the "Submit" button only when the total is exactly 9.
b. Backend validation: Flask verifies the selection count once the data is submitted. If the number is incorrect, the system automatically rejects the input.

2.5 Double-Vote Prevention

Each voter can vote only once. Voter status is stored in the *already_vote* column. After a vote is successfully submitted, the status changes to *True*, blocking any further voting attempts.

2.6 Tools and Materials

a. Programming language: Python 3.11
b. Framework: Flask 3.0
c. Database: SQLite
d. Interface: HTML5, Bootstrap 5, JavaScript
e. Devices: Laptop and smartphone
f. Browser: Chrome

2.7 Testing Method

Testing includes:

a. Functional Testing (Black-box testing) to ensure system functions operate correctly.
b. Performance Testing to compare the time required for manual voting versus system-based voting.
c. User Satisfaction Testing using a Likert scale (1–5) survey involving 30 participants.

3. RESULTS AND DISCUSSION

3.1 System Implementation Results

This system was developed using a Research and Development (R&D) method with a prototyping approach, allowing iterative refinement based on user needs and testing outcomes. The implementation results are presented by aligning each outcome with the development stages, namely requirement analysis, system design, implementation, and testing.

Based on the requirement analysis stage, the main problems identified in previous PC IPM Cepu elections included invalid ballots due to incorrect candidate counts, long recapitulation times, and the absence of automatic validation. These findings became the basis for designing a system that emphasizes simplicity, rule enforcement, and efficiency.

At the system design stage, the architecture was based on a client–server model. The client side consists of a web-based user interface accessed through a browser, while the server side is handled by a Flask application that processes requests and manages data storage. The database was modeled using SQLite with core tables including voter data, candidate data, and vote records. The interface design focuses on usability, featuring clear navigation, readable typography, and minimal interaction steps to reduce user errors.

The interface design results show that the main page displays essential election information, including the total number of registered voters and the number of voters who have already cast their votes. This page also serves as an entry point to the voting process. On the voting page, candidates are presented in a structured list accompanied by checkboxes. A real-time selection counter is displayed to inform voters of how many candidates have been selected. The "Submit" button is automatically activated only when the voter selects exactly nine candidates, ensuring compliance with the election rule at the user interface level.

During implementation, the system was developed using Python with the Flask framework and an SQLite database. Client-side validation was implemented using JavaScript to control checkbox selection and button activation. On the server side, Flask performs additional validation to ensure the submitted vote contains exactly nine candidate selections before storing the data. This dual-layer validation prevents vote manipulation through browser inspection or request tampering. Once validated, vote data are stored in the database immediately.

The implementation results also demonstrate that vote data can be processed and displayed in real time. The results page dynamically shows vote counts for each candidate, allowing election committees to monitor progress without manual recapitulation. This feature significantly reduces administrative workload and minimizes human error.

Overall, the system implementation confirms that the designed prototype functions as intended. The Integration of interface design, system architecture, and database modeling successfully addresses the problems identified during the analysis stage. By enforcing selection-count validation at both the client and server levels, the system ensures vote validity while maintaining ease of use for student voters.

3.2 System Testing

3.2.1 Functional Testing

All system features were tested using the black-box method. The results show that the vote-count validation works perfectly with a 0% error rate. Table 1 summarizes the validation test results.

Table 1. Validation Test Results

Condition	System Response	Status
< 9 candidates selected	"Submit" button inactive	Safe
9 candidates selected	"Submit" button active; data accepted	Safe
> 9 candidates selected	"Submit" button inactive	Safe

3.2.2 Double-Vote Prevention Test

Testing was performed by attempting to vote twice using the same identity. The system rejected the second attempt and displayed a warning message. All test attempts (30 respondents) showed a 100% success rate in preventing double voting.

3.2.3 Performance Testing

The average time required for manual voting was 20 minutes per session, while the e-voting system required only 4 minutes. This represents an 80% efficiency improvement.

Table 2. Voting Time Comparison

Method	Time (minutes)	Efficiency
Manual	20	
E-voting	4	80% faster

3.2.4 User Satisfaction Test

A survey was conducted with 30 participants using four main indicators. The results are shown in Table 3.

Table 3. User Satisfaction Levels

Aspect	Average (%)
Ease of use	100%
System speed	93%
Interface clarity	97%
Result accuracy	93%
Overall average	95%

The survey results indicate a positive response, especially regarding ease of use and interface clarity.

3.3 Discussion

The e-voting system developed in this study demonstrates substantial improvements in both efficiency and accuracy compared with conventional manual voting methods. This improvement is not merely procedural but structural, as it is rooted in the system's design philosophy that embeds election rules directly into the voting mechanism. The dual validation mechanism implemented at the client and server sides ensures that invalid inputs are eliminated before they can compromise the integrity of election data. Client-side validation functions as the first line of defense, guiding voter behavior in real time, while server-side validation serves as a definitive safeguard against data manipulation, including request tampering and unauthorized input submission. This layered control significantly strengthens the robustness of the voting process.

From a data-integrity perspective, the automatic enforcement of the selection rule (exactly nine candidates) transforms the voting process from an error-prone, human-dependent activity into a rule-driven digital system. Unlike manual elections, where committee members must detect and discard invalid ballots after voting has concluded, the proposed system prevents such ballots from being created in the first place. This preventive approach aligns with the principle of "error prevention rather than error correction," which is widely regarded as a best practice in reliable system design. As a result, the system achieves zero invalid ballots by construction, rather than by post-hoc verification.

Implementing a double-vote prevention mechanism further enhances electoral integrity. By binding each voting session to a unique voter identity and validating voting

status at the database level, the system effectively enforces the democratic principle of equal participation. This mechanism not only prevents intentional repeated voting but also protects against accidental resubmission caused by network delays or user uncertainty. In manual elections, such issues are often difficult to detect and may compromise fairness; in contrast, the automated system ensures consistency and fairness without increasing administrative burden.

The reported 80% improvement in time efficiency reflects systemic optimization rather than isolated performance gains. In manual elections, time consumption is distributed across multiple stages, including ballot distribution, vote casting, manual counting, verification, recounting, and result announcement. The proposed e-voting system consolidates these stages by integrating real-time validation and automatic recapitulation into the voting process itself. Consequently, the distinction between "voting time" and "counting time" becomes blurred, as valid vote data are instantly ready for aggregation. This structural efficiency explains the significant reduction in total election duration observed during system testing.

User satisfaction, reflected by an average score of 95%, provides important insight into system acceptance beyond technical correctness. High satisfaction suggests that voters perceive the system as trustworthy, easy to use, and transparent. In the context of democratic systems, perceived legitimacy is as critical as technical security. A secure system that is difficult to understand may discourage participation or foster distrust. The results indicate that the balance between simplicity and functionality achieved by the proposed system successfully supports user confidence, particularly among students with varying levels of digital literacy.

From a comparative perspective, the findings are consistent with those of Rahman and Lee [18] regarding the necessity of client–server validation to maintain vote-data integrity. However, this study advances prior research by positioning automated selection-count validation as a central system feature rather than a peripheral enhancement. While previous studies have focused on authentication, encryption, or access control [2], [6], they often assume that voters will comply with selection rules. This assumption does not hold in student-based elections, where invalid ballots are frequently the result of misunderstanding or oversight. By embedding the selection rule into the system logic, this study directly addresses a practical yet underexplored source of electoral failure.

When compared with the work of Hidayat and Ramadhani [3], the proposed system adopts a more minimalist architectural approach while retaining essential reliability features. The use of Flask and SQLite demonstrates that effective e-voting systems do not necessarily require complex infrastructures or high computational resources. This is particularly relevant for student organizations and small institutions, where technical expertise and funding are limited. The findings suggest that system appropriateness rather than system complexity should be the primary consideration in organizational digitalization initiatives.

At a conceptual level, this study reinforces the argument that digitalization in student organizations extends beyond operational efficiency and enters the domain of democratic education. By interacting with a transparent, rule-based, and accountable voting system, students are implicitly introduced to core democratic values such as fairness, responsibility,

and integrity. This experiential learning dimension supports the views of Wibowo [1] and Pramono and Hakim [19], who emphasize that technology-mediated participation can strengthen democratic culture within educational environments.

Overall, the discussion confirms that the proposed e-voting system delivers multi-dimensional benefits: technical (accuracy and efficiency), organizational (reduced workload and improved governance), and educational (enhanced democratic awareness). By combining automated rule enforcement, dual-layer validation, and lightweight web technologies, this system provides a scalable and replicable model for internal elections in schools, youth organizations, and other small-scale institutions. The study therefore contributes not only a functional system but also a practical framework for responsible, context-sensitive digital democratic transformation.

4. CONCLUSION

This study highlights the successful implementation of a Flask–SQLite–based e-voting system that integrates automated vote-count validation and double-vote prevention, demonstrating a practical model for small-scale digital democracy in student organizations. The research provides empirical and conceptual insights into how dual-layer validation mechanisms can enhance data integrity and ensure rule-compliant voting behavior, offering a framework that other educational or community-based organizations can adapt.

The study has certain boundaries, including its application within a single student organization, a limited sample size, and testing conducted under controlled conditions, which may affect generalizability. Despite these limitations, the findings contribute to the broader understanding of digital democratic practices by illustrating how simplicity, usability, and rule enforcement can coexist in small-scale election systems.

For future research, subsequent studies could explore additional security measures, such as account-based or QR-code authentication, the implementation of WebSocket or AJAX for real-time result updates, and encryption techniques like RSA or ChaCha20 to enhance vote confidentiality. Expanding trials to regional or national levels could evaluate scalability, effectiveness, and user engagement across more diverse populations. Furthermore, integrating statistical analyses to assess the significance of efficiency improvements or user satisfaction would strengthen the empirical foundation of e-voting system research.

Overall, this research contributes to the general public by providing a replicable and accessible digital voting model that promotes transparency, accountability, and participatory democracy within organizational contexts, while serving as a practical guideline for future implementations of small-scale e-voting systems.

REFERENCES

[1] A. Wibowo, “Penerapan Teknologi Digital dalam Peningkatan Efisiensi Proses Demokrasi Internal,” *Jurnal Transformasi Digital*, vol. 3, no. 1, pp. 64–71, 2021.

[2] D. Ariyanto and R. Prasetyo, “Penerapan Sistem E-Voting Berbasis Web Menggunakan Framework Laravel,” *Jurnal Teknologi dan Sistem Informasi*, vol. 8, no. 2, pp. 121–130, 2020.

[3] A. Hidayat and F. Ramadhani, “Perancangan Sistem E-Voting Menggunakan Metode Validasi Ganda pada Pemilihan Organisasi Mahasiswa,” *Jurnal Informatika*, vol. 10, no. 3, pp. 211–220, 2021.

[4] R. Nugroho and D. Yuliana, “Analisis Penggunaan Sistem E-Voting dalam Pemilihan Umum Internal

Organisasi," *Jurnal Rekayasa dan Sistem Informasi*, vol. 7, no. 4, pp. 332–340, 2019.
[5] A. Kurniawan, "Pembangunan Aplikasi Pemilihan Ketua Kelas Menggunakan Flask dan SQLite," *Jurnal Teknologi Informasi Terapan*, vol. 6, no. 1, pp. 45–54, 2020.
[6] R. Alaguvel et al., "Secure Web-Based Electronic Voting System Using Flask Framework," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 9, pp. 45–52, 2021.
[7] Tim Pimpinan Pusat Ikatan Pelajar Muhammadiyah, *Pedoman Umum Organisasi dan Tata Kerja IPM (ART dan AD)*. Yogyakarta: PP IPM, 2023.
[8] D. Chaum, "Secret-ballot receipts: True voter-verifiable elections," IEEE Security & Privacy, vol. 2, no. 1, pp. 38–47, Jan.–Feb. 2004.
[9] J. Benaloh, "Simple verifiable elections," in Proceedings of the USENIX/ACCURATE Electronic Voting Technology Workshop (EVT), Vancouver, BC, Canada, 2006, pp. 1–5.
[10] B. Adida, "Helios: Web-based open-audit voting," in Proceedings of the 17th USENIX Security Symposium, San Jose, CA, USA, 2008, pp. 335–348.
[11] A. Kiayias and M. Yung, "The vector-ballot e-voting approach," in Financial Cryptography and Data Security, Lecture Notes in Computer Science, vol. 5143, Berlin, Germany: Springer, 2008, pp. 72–89.
[12] S. Park, J. H. Kim, and S. J. Cho, "Usability evaluation of electronic voting systems," International Journal of Human–Computer Interaction, vol. 30, no. 9, pp. 673–684, 2014.
[13] V. Cortier and B. Smyth, "Attacking and fixing Helios: An analysis of ballot secrecy," Journal of Computer Security, vol. 21, no. 1, pp. 89–148, 2013.
[14] R. Krimmer, D. Duenas-Cid, and I. Krivonosova, "New methodology for calculating cost-efficiency of different e-voting solutions," Government Information Quarterly, vol. 37, no. 3, Art. no. 101473, 2020.
[15] J. C. Perez, M. A. Labrador, and J. A. Banegas, "E-voting systems for educational institutions: A case study," IEEE Revista Iberoamericana de Tecnologías del Aprendizaje, vol. 13, no. 2, pp. 65–72, May 2018.
[16] A. Macdonald and T. McBride, "Digital participation and civic engagement among youth," Information, Communication & Society, vol. 22, no. 13, pp. 1962–1979, 2019.
[17] S. A. Al-Kassasbeh, M. M. Al-Tarawneh, and A. Al-Fayoumi, "Web-based electronic voting systems: Design and implementation," International Journal of Advanced Computer Science and Applications, vol. 11, no. 6, pp. 418–425, 2020.
[18] S. Rahman and J. Lee, "Enhancing E-Voting Systems with Client-Server Validation for Data Integrity," *Journal of Computer Science and Applications*, vol. 20, no. 4, pp. 135–142, 2022.
[19] H. Pramono and F. Hakim, "Digital Transformation in Student Organizations: Building Participation and Transparency," *Indonesian Journal of Information System*, vol. 9, no. 1, pp. 22–30, 2023.
[20] A. Alomari, K. Al-Saleh, and M. Al-Tahat, "Design and implementation of a secure electronic voting system," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 8, no. 2, pp. 222–229, 2017.
[21] S. Kumar and R. Sinha, "Performance evaluation of electronic voting systems in small-scale elections," International Journal of Computer Applications, vol. 181, no. 34, pp. 1–6, 2018.
[22] J. Nielsen and R. Molich, "Heuristic evaluation of user interfaces," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Seattle, WA, USA, 1990, pp. 249–256.
[23] M. Bishop and D. Wagner, "Risks of e-voting," Communications of the ACM, vol. 47, no. 1, pp. 93–99, Jan. 2004.
[24] B. Shahzad, A. Crowcroft, and M. Haddadi, "Trustworthy electronic voting using distributed systems," IEEE Security & Privacy, vol. 14, no. 4, pp. 20–27, 2016.
[25] S. Macedo, L. Avritzer, and J. M. Vieira, "Digital participation and democratic innovation," Policy & Internet, vol. 12, no. 3, pp. 327–346, 2020.