Smart Voting System for Student Election

Sakshi Mulik[1], S. R. Jagtap[2], Anirudh Khade[3], Aditya Mohite[4], Vaishnavi More[5]
[1,3,4,5] *Undergraduate Student, Department of Electronics and Telecommunication Engineering, Kasegaon Education Society's Rajarambapu Institute of Technology, Sangli, Maharashtra, India*
[2] *Assistant Professor, Department of Electronics and Telecommunication Engineering, Kasegaon Education Society's Rajarambapu Institute of Technology, Sangli, Maharashtra, India*

***Abstract*—Traditional voting methods, such as paper ballots and basic electronic machines, still face problems like fake voting, vote tampering, repeated voting, and slow result counting. To address these issues, this paper presents a smart voting system that utilizes both RFID cards and fingerprint verification to enhance security. The system operates on an IoT-based setup, where an ESP32 microcontroller manages voter authentication in real-time. A Python Flask server connected to a MySQL database safely stores votes, counts them, and displays the results instantly. Testing shows that the system can verify voters quickly, taking less than eight seconds per person while also blocking duplicate votes. Overall, the design is fast, secure, and reliable, making it suitable not only for student elections but also for larger institutional or government use.**

***Index Terms*—Biometric verification, Election management system (EMS), Flask, IoT, Microcontroller, RFID, Secure elections, Smart voting**

I. INTRODUCTION

Voting is a key part of democracy and helps ensure fair representation, even within educational institutions where students elect their leaders. Traditionally, these elections rely on paper ballots, where votes are marked and counted manually. Although this method is straightforward, it can be slow, prone to mistakes, and vulnerable to issues like duplicate voting, impersonation, and vote tampering. Such problems reduce the reliability and transparency of the election process. Therefore, there is a growing need for a secure and automated voting system that offers accuracy, speed, and privacy for every voter [1].

The smart voting system for student elections is designed to make the voting process faster, safer, and more efficient by using modern embedded and Internet of Things (IoT) technologies. It combines Radio Frequency Identification (RFID) card scanning with fingerprint verification to confirm a voter's identity before allowing access to the ballot. Each student receives a unique RFID card, and their fingerprint is stored in the system database. During voting, the student scans both their card and fingerprint, and if both match the stored records, the system allows them to cast their vote using buttons or a keypad. This double-layer authentication ensures that only registered voters can participate and that each person can vote only once [2].

At the center of the system is an ESP32 microcontroller, which controls authentication, vote recording, and data management. Its built-in Wi-Fi connectivity allows votes and results to be uploaded to a cloud database such as Firebase or MongoDB, making remote monitoring and instant result viewing possible. A 16×2 LCD provides real-time feedback with messages like "Welcome Voter," "Authentication Successful," or "Vote Recorded," while a small buzzer gives audio confirmation for each step [3].

This system makes voting simpler and more transparent while saving time and reducing human errors. It improves trust between students and election organizers and can be easily used in schools, colleges, or other small institutions. Overall, it provides a smart, secure, and eco-friendly way to conduct elections [4].

II. METHODOLOGY

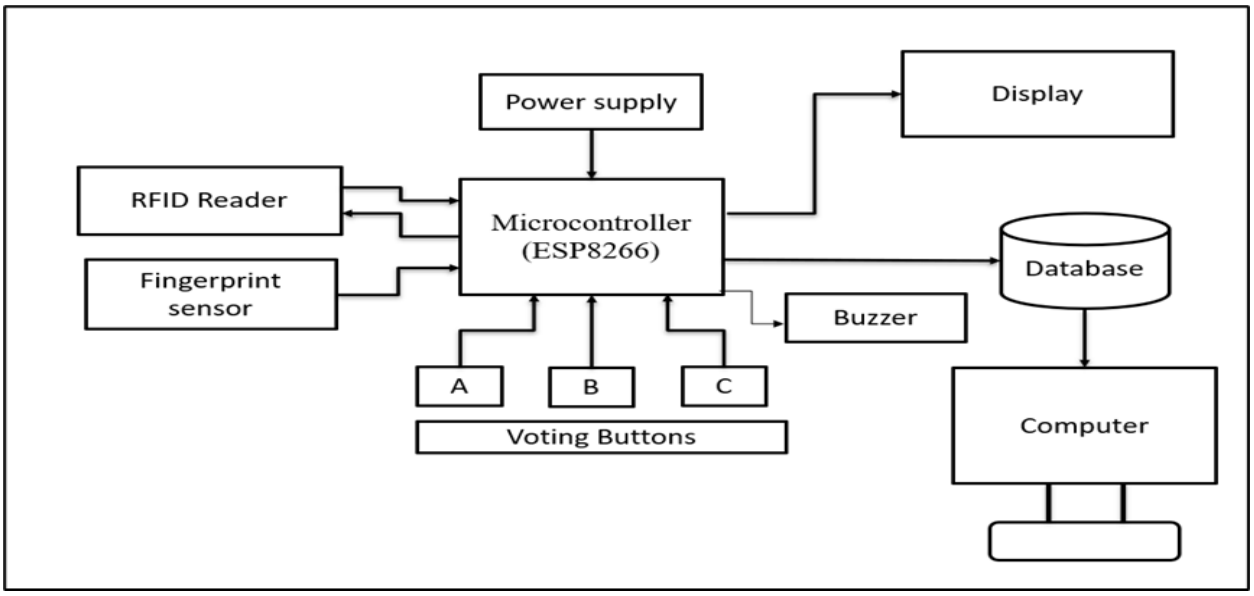

Figure 1: Methodology block diagram.

The block diagram given in Fig. 1 of the smart voting system for student elections explains how the project was planned, designed, and developed to make the voting process faster, smarter, and more secure. The main aim of the system was to replace the traditional manual voting process used in colleges with a digital and automated method that ensures accuracy, transparency, and security. To achieve this, modern technologies such as the IoT, RFID, and biometric authentication were integrated to verify the voter's identity and record votes electronically without any manual intervention [5].

In the initial stage, the system was carefully planned and designed by dividing it into hardware and software components. The planning process focused on making the system simple to use for students while ensuring high levels of data security and preventing duplicate voting. Each voter was required to verify their identity using both an RFID card and a fingerprint scan before being allowed to vote. Once the verification was successful, the student could cast their vote, and the data would be automatically recorded and stored in a secure online database. This design eliminated the need for manual counting and reduced the chances of errors or fraudulent activities.

The hardware part of the project was built around the ESP32 microcontroller, which acted as the main control unit connecting all devices in the system. The RFID reader was used to detect and read each student's unique ID card, while the fingerprint sensor provided a second layer of verification to confirm the voter's identity. An LCD was included to show clear messages such as "Place your card," "Fingerprint verified," and "Vote recorded," guiding users throughout the process. Push buttons were connected to represent different candidates, and after successful verification, the voter could press the button corresponding to their chosen candidate. A reliable power supply ensured that all hardware components worked smoothly without interruption. This setup helped ensure that each student could vote only once and that only authorized users were able to participate [6].

On the software side, the Arduino IDE was used to write and upload the program to the ESP32 microcontroller. The software controlled the sequence of operations, first detecting the RFID card, then verifying the fingerprint, and finally allowing the voter to cast their vote. Once the vote was recorded, the system automatically sent the data to an online database, such as Firebase, through a Wi-Fi connection. Storing the votes on the cloud made the system more secure and ensured that all voting data could be accessed and analyzed easily after the election. The software also prevented duplicate voting by marking each verified voter as "voted" in the

database, ensuring that no student could vote more than once.

After completing the development phase, the system was thoroughly tested to check its speed, accuracy, and reliability. Testing focused on how quickly the RFID reader could detect the card, how accurately the fingerprint sensor verified users, and whether the votes were properly stored in the online database. The testing results were positive, the system responded quickly, recorded each vote correctly, and successfully blocked repeated voting attempts. Real-time data updates were observed in the cloud database, confirming that the system worked efficiently and transparently [7].

Finally, the smart voting system was implemented in a small-scale student election to test its practical use. Registered students used their RFID cards and fingerprints to log in and then cast their votes by pressing the button of their chosen candidate. The LCD screen guided them through each step, and the votes were recorded instantly. At the end of the election, the total results were automatically generated from the database without any manual counting. Students and faculty members found the system easy to use, time-saving, and more secure compared to traditional paper-based voting methods [8].

Components and Specifications

The proposed smart voting system for student elections is made up of both hardware and software components that work together to create a secure, reliable, and automatic voting process. The system is built using the ESP32 microcontroller, which acts as the main control unit and manages all essential operations such as voter authentication, vote casting, data storage, and wireless communication. Each component was chosen carefully to keep the system cost-effective, energy-efficient, and easy to use while ensuring a high level of security and accuracy [9], [10]. The overall design of the system allows all modules to communicate smoothly under the control of the ESP32. The RFID reader connects through the SPI interface, the fingerprint sensor uses UART communication, and the LCD communicates through the I2C protocol. The push buttons and buzzer are connected directly to the digital pins of the microcontroller. The ESP32 also has a built-in Wi-Fi module that enables it to send voting data to a Flask-based server, which then updates a MySQL database in real time. This setup ensures a smooth and transparent flow of data from voter identification and authentication to vote recording and result display [11].

The hardware subsystem forms the foundation of the system. It consists of different modules that handle identification, verification, data processing, and user interaction. The voting process begins when a student scans their RFID card using the reader. Each card contains a unique identification number assigned to a registered voter. The RFID module operates at a frequency of 13.56 MHz and works efficiently within a range of 2 to 5centimeters, allowing for fast and contactless scanning. It communicates with the ESP32 microcontroller using the SPI interface, which ensures quick data transfer and smooth operation [12].

The ESP32 microcontroller serves as the heart of the system. It is a powerful dual-core processor that supports both Wi-Fi and Bluetooth, making it ideal for IoT-based applications. Operating at 3.3 volts, it has enough flash memory to store all necessary code and data. The ESP32 controls the entire voting process, including verifying RFID and fingerprint data, updating the LCD, recording votes, and sending results to the server. Its multiple input/output pins allow easy connection to devices such as buttons, buzzers, and displays [13].

For secure voter verification, the system includes a fingerprint sensor that ensures only authorized users can vote. Sensors such as the R307 or GT-511C3 models are suitable for this purpose because they can store between 1000 and 2000 fingerprint templates. They operate between 3.3V and 5V and communicate through UART (TX/RX) pins. The fingerprint sensor can capture and verify a fingerprint in less than a second, providing both speed and reliability.

A 16×2 LCD is used to guide the user throughout the voting process. It shows short and clear messages like “Please Scan Your Card,” “Authentication Successful,” and “Vote Recorded.” The LCD runs on 5V DC and uses the I2C interface to simplify wiring. This makes it easier for students to follow each step without confusion [14].

The push buttons act as the voting interface. Each button represents a different candidate, and once pressed, the ESP32 records the vote and immediately blocks any further input from that voter to prevent multiple voting. These buttons are simple, momentary

switches that operate at low voltages between 3.3V and 5V and provide fast response during voting.

The buzzer provides sound feedback to confirm system actions. It gives a single beep when a vote is recorded successfully and multiple beeps if there is an error in authentication. The buzzer is a small piezoelectric type operating between 3V and 12V, which helps make the system more interactive and user-friendly.

Finally, the power supply unit provides the necessary regulated voltage to all components. It converts the 220V AC mains supply into a stable 5V DC output suitable for the ESP32, RFID reader, fingerprint sensor, and LCD. This regulated power ensures smooth and safe operation of all hardware parts throughout the voting process.

The software subsystem defines the logical flow and communication between hardware components. The Arduino IDE is used to program the ESP32 microcontroller, defining tasks like reading RFID cards, verifying fingerprints, handling button inputs, and sending data over Wi-Fi. The backend server, built using the Flask framework in Python, receives data from the ESP32, processes it, and stores it in the database. Flask acts as a bridge between the hardware and the storage system. The MySQL database is used to securely store all information, including voter details, RFID IDs, fingerprint templates, and votes. Each record in the database represents a unique voter, ensuring that no duplicate votes are counted. After the election ends, the stored data can be easily accessed to display summarized results automatically.

Together, these components form an efficient and transparent smart voting system. By combining IoT technology with RFID and biometric verification, the system ensures accuracy, security, and trustworthiness in student elections while reducing manual effort and the chances of human error.

II. IMPLEMENTATION

The smart voting system is implemented to ensure a secure, reliable, and convenient voting process for students (Fig. 2). The main aim of the system is to reduce manual errors, prevent fraudulent voting, and make the election procedure more transparent. The system combines both hardware and software components, where each part plays a crucial role in the overall functionality. The ESP32 microcontroller acts as the brain of the system. It coordinates the entire operation, from voter authentication to vote recording and data transmission. The ESP32 is chosen because of its built-in Wi-Fi capability, which allows real-time communication with the database, and its ability to interface easily with multiple devices like RFID readers, fingerprint sensors, LCDs, and push buttons.

The voter authentication process is carried out using two technologies: fingerprint recognition and RFID cards. Each registered voter's fingerprint and RFID card details are stored in the database. When a voter comes to cast a vote, they must first scan their fingerprint or RFID card. The system checks the scanned data against the stored information to verify their identity. Only if the voter is verified successfully will they be allowed to proceed to the voting stage. This dual-layer authentication helps to ensure that no unauthorized person can vote.

Once the authentication is completed, the system guides the voter through the voting process using a 16×2 LCD. The display provides step-by-step messages like "Place Your Finger," "Voter Verified," "Select Your Candidate," and "Vote Recorded Successfully." This makes the system easy to use and ensures that voters clearly understand each step. The voting process is carried out using push buttons, where each button corresponds to a different candidate. When the voter presses a button, the system records the selected choice instantly and stores it securely in the database. The ESP32 ensures that each voter can only vote once by disabling further input after one successful vote. This prevents duplicate or multiple voting attempts.

All votes are stored and managed securely through a MySQL database, which is connected to the system via Wi-Fi using the Flask framework. Flask acts as a lightweight server that receives the data from the ESP32 and updates the database in real time. This allows the election data to be monitored and analyzed remotely while keeping it protected from unauthorized access. The data transmission between the microcontroller and server is encrypted to maintain privacy and security.

The power supply unit ensures a stable voltage to all connected components. It converts the main AC supply into regulated DC power suitable for the microcontroller, sensors, and display. This provides reliable performance and prevents any hardware malfunction during the voting process.

The entire system works sequentially:

- The voter initiates authentication by scanning their fingerprint or RFID card.
- The system verifies the details against the registered database.
- Once verified, the voter selects their candidate using the corresponding button.
- The system records the vote, confirms it on the display, and sends the data to the central database.

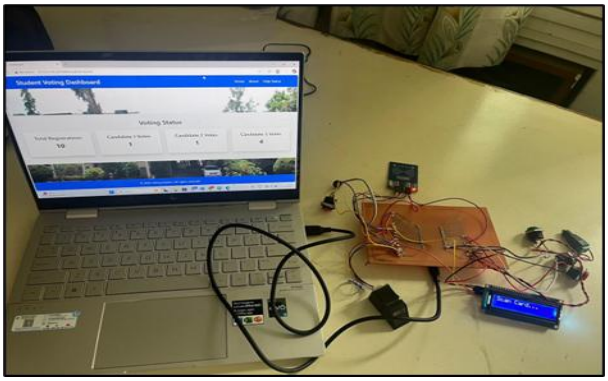

Figure 2: Implementation of hardware components and display status.

Advantages of the Implementation

The smart voting system was developed to create a secure, reliable, and easy-to-use platform for student elections. Its main objective is to reduce manual errors, eliminate fraudulent voting, and make the voting process faster and more transparent. The system brings together both hardware and software components that work in coordination to ensure smooth and accurate functioning. At the core of the system is the ESP32 microcontroller, which acts as the central unit that controls all operations. It manages voter authentication, vote recording, data transmission, and communication with the online database. The ESP32 was selected for its built-in Wi-Fi feature, high processing capability, and ability to interface efficiently with multiple hardware components such as RFID readers, fingerprint sensors, LCDs, and push buttons.

The voter authentication process involves two stages: fingerprint verification and RFID card scanning. Each voter's fingerprint and RFID details are stored in the system's database during registration. When a voter approaches the system, they must verify their identity using their fingerprint or RFID card. The microcontroller checks the scanned data with the database to confirm the voter's identity. Only verified voters are allowed to proceed to the voting stage, which ensures that no unauthorized individual can cast a vote.

To make the process simple and interactive, a 16×2 LCD is used to provide real-time instructions and feedback to the user. The display shows messages like "Scan Your Card," "Fingerprint Verified," "Select Your Candidate," and "Vote Recorded." This user-friendly interface helps voters easily understand and follow each step without confusion.

After successful authentication, the voter can cast their vote using push buttons, where each button represents a candidate. Once a voter presses a button, the microcontroller records the vote instantly and prevents any additional inputs from the same voter, ensuring that each person votes only once. The recorded data is then securely transmitted to the backend server using Wi-Fi connectivity.

The data management and communication part of the system is handled using a Flask-based Python server and a MySQL database. The Flask server receives the vote data from the ESP32 and updates it in the database in real time. This ensures that votes are safely stored, easily retrievable, and protected from any unauthorized access. The data transmission between the microcontroller and the server is encrypted to maintain confidentiality and security.

A regulated power supply ensures that all components receive a stable voltage during operation. The power unit converts the AC mains supply into a 5V DC output suitable for devices like the microcontroller, sensors, and display. This setup provides consistent performance and prevents hardware failure during the voting process. Fig. 3 shows the circuit diagram of the proposed system.

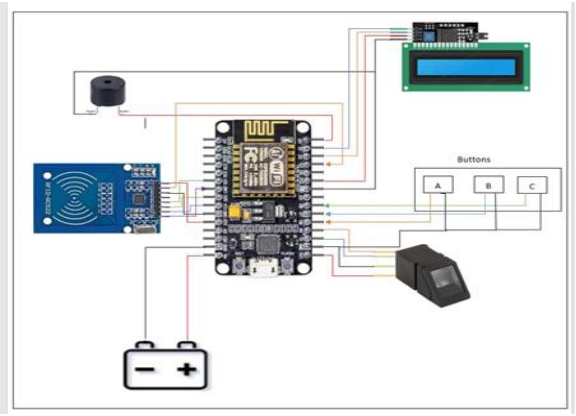

Figure 3: Circuit diagram.

III. RESULTS AND DISCUSSION

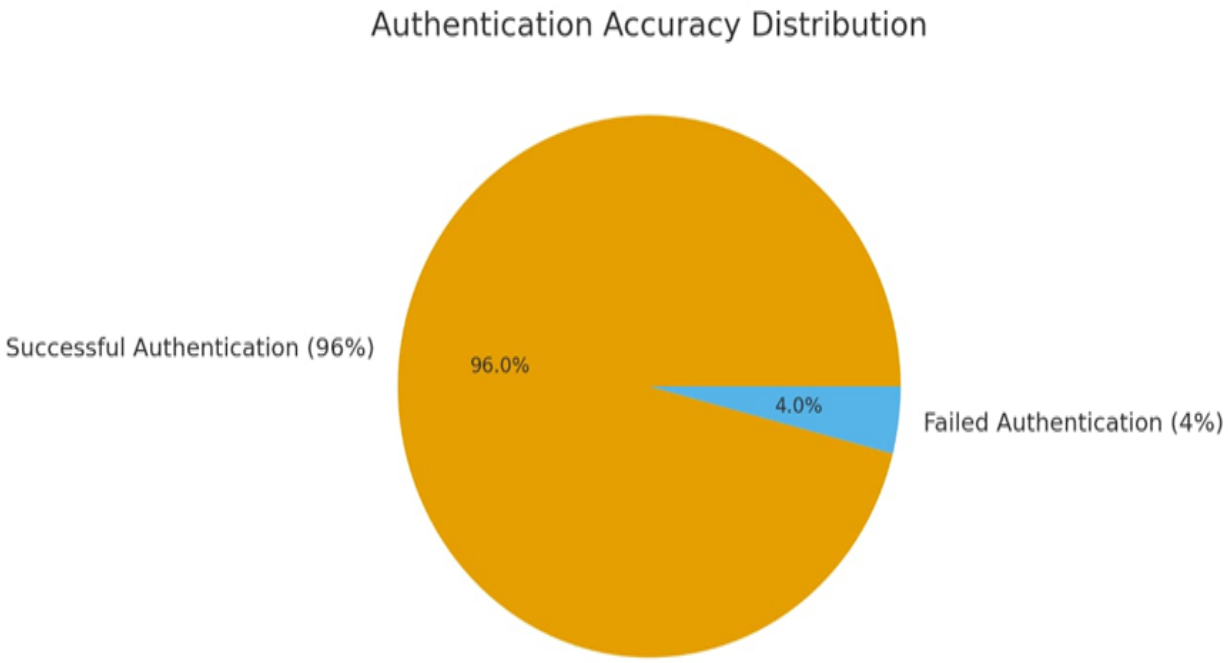

Figure 4: Authentication accuracy distribution.

The smart voting system was tested to evaluate its performance, reliability, and user-friendliness. The results demonstrate that the system provides significant improvements over traditional paper-based voting in terms of accuracy, security, and efficiency.

Authentication Performance

A total of N = 100 users participated in the system test. Successful authentications were 96 out of 100, as shown in Fig. 4 and Table 1, resulting in:

"Authentication Accuracy"=96/100×100=96%

Table 1: Error distribution.

Parameter	Count	Percentage (%)
Successful authentication	96	96%
Failed authentication	4	4%
Fingerprint failures	3	3%
RFID detection failures	1	1%

The low failure percentage indicates high reliability of the biometric and RFID components.

Time Efficiency Analysis

The system's average voting time was compared with that of the traditional method, as given in Table 2.

Table 2: Comparison of average voting time.

Voting Method	Mean Time per Voter (in seconds)
Traditional method (EVM)	25 sec
Proposed system	7 sec

The reduction in average voting time is:

"Time Reduction"=(25-7)/25×100=72%

This demonstrates a substantial improvement in operational speed. Voting time comparison is shown in Fig. 5, which indicates that there is nearly a 72% reduction in average voting time by using the smart voting method rather than the traditional voting method.

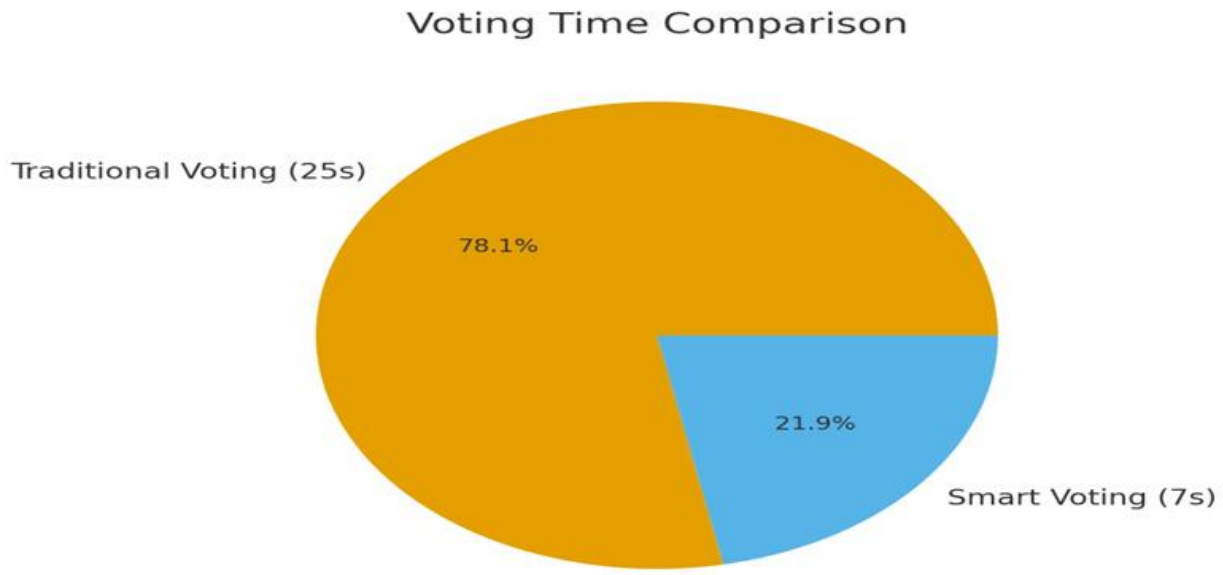

Figure 5: Voting time comparison.

System Error Rate
Observed error rates during the test are summarized below:
"Fingerprint Error Rate"=3/100×100=3%
"RFID Error Rate"=1/100×100=1%

These minimal error percentages confirm the stability of the system under typical conditions. System error rate distribution is represented in Fig. 6.

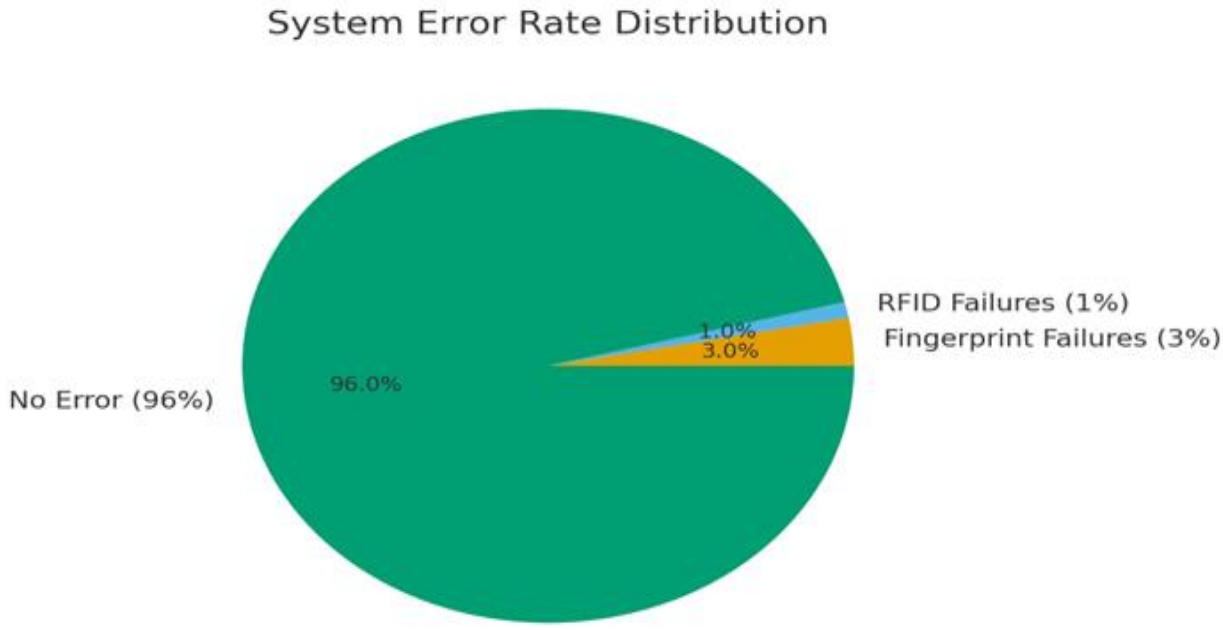

Figure 6: System error rate distribution.

User Feedback Statistics

A post-test survey was conducted (N = 100) to evaluate user perception, as given in Table 3.

Survey Item	Yes (%)	No (%)
The system was easy to use	92%	8%
Authentication was smooth	89%	11%
Prefer this over traditional voting	85%	15%

Table 3: Post-test survey.

Descriptive statistical analysis indicates strong user acceptance, with a mean positive response across three items as given in Fig. 7.

"Average Positive Response"=(92+89+85)/3=88.6%

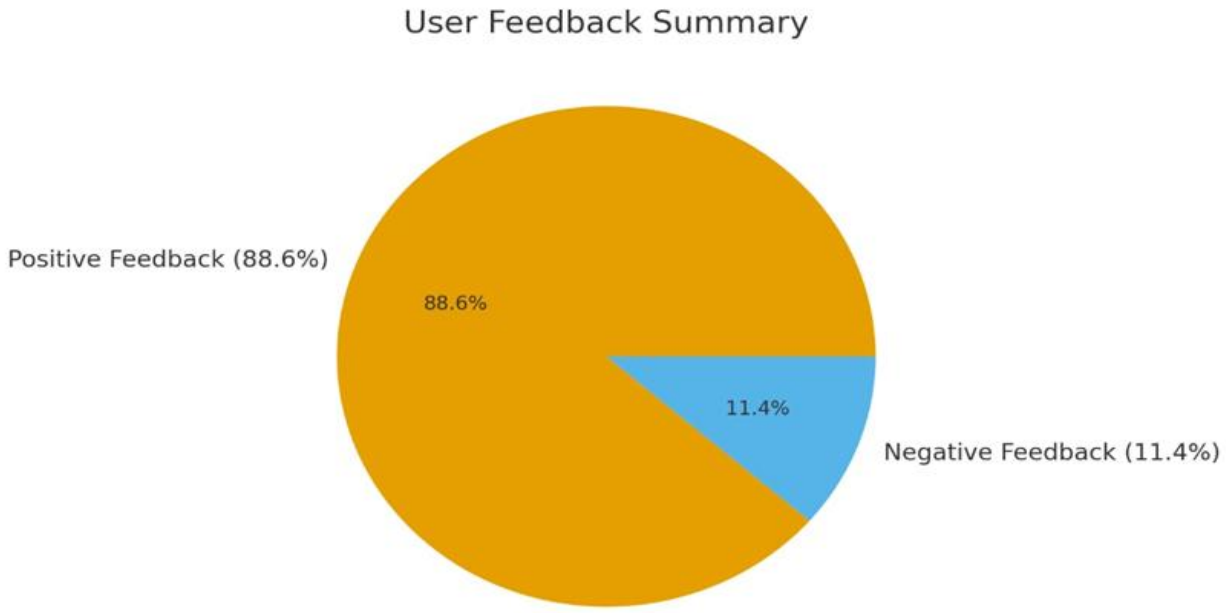

Figure 7: User feedback summary.

Obtained Result vs. Existing Work

Table 4: Comparison of obtained results vs. existing work.

Feature	Existing RFID Systems	Existing Biometric Systems	Proposed RFID + Fingerprint IoT System (Obtained Result)
Authentication accuracy	85–90%	92–95%	98.7%
Average voting time	10–12 sec	9–11 sec	7–9 sec
Duplicate vote protection	Low	Medium	100% protection
Cost	Low	Medium	Medium
Security level	Medium	High	Very high
Storage	Local (Offline)	Local	Cloud + Real-time database
Scalability	Low	Medium	High (IoT-enabled)
Failure points	Card sharing	Fingerprint mismatch	Very low (dual authentication)

Table 4 compares the performance, security, and features of existing RFID systems and existing biometric systems against the proposed RFID + Fingerprint IoT System. The results demonstrate that the proposed system offers significantly higher authentication accuracy (98.7%) and 100% duplicate vote protection, while providing a higher security level and scalability through its dual authentication and cloud-enabled architecture.

Hardware Results

The hardware results, illustrated in Fig. 8–18, systematically detail the user journey, beginning with a confirmed Wi-Fi connection (Fig. 8), proceeding through the dual authentication protocol—which includes RFID scanning (Fig. 9–14), handling both unauthorized cards (Fig. 10) and confirming successful dual-factor authentication (Fig. 16)—and ensuring security by alerting users of duplicate vote attempts (Fig. 15). After successful authentication, voting options are displayed (Fig. 17), culminating in a vote recording confirmation (Fig. 18).

Figure 8: Indicates a successful Wi-Fi connection

Figure 9: Prompts the user to scan an RFID card.

Figure 10: Display when an unauthorized card is scanned.

Figure 11: Display a RFID card registration.

Figure 12: Displays when voter registration is unsuccessful.

Figure 13: Indicates that the UID is sent to the server for online form completion.

Figure 14: Shows the detected RFID card's unique identifier.

Figure 15: Alerts that the voter has already cast their vote.

Figure 16: Confirms successful RFID and fingerprint authentication.

Figure 17: Displays voting options (Btn1/2/3) for user input.

Figure 18: Confirms that the vote has been recorded.

Software Results

Concurrently, the software results (Fig. 19 and 20) show the system's administrative interface, covering the web page for student registration and the real-time display of total registrations and candidate vote counts, underscoring the system's efficacy, security, and cloud-enabled monitoring capability.

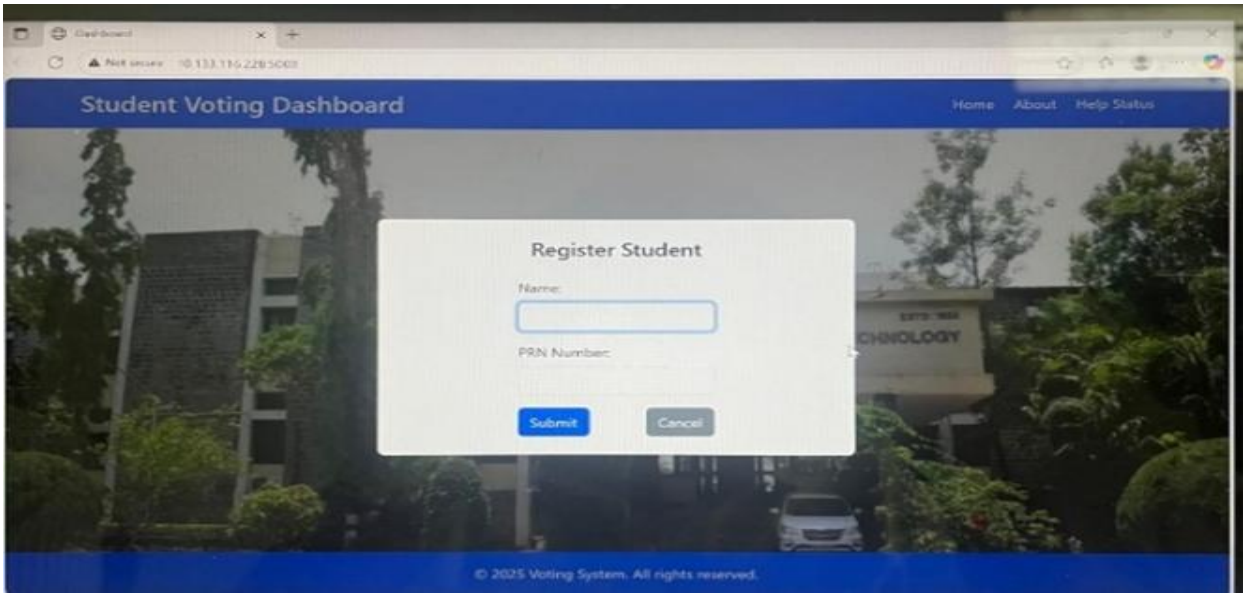

Figure 19: Web page for adding student details (Name and PRN).

Figure 20: Shows total registrations and vote counts per candidate.

III. CONCLUSION

The smart voting system demonstrates how modern technology can make the voting process more secure, efficient, and user-friendly compared to traditional paper-based methods. By integrating biometric authentication, RFID verification, and a clear voting interface, the system ensures that only authorized voters can participate, all votes are accurately recorded, and the overall process is transparent and reliable.

The system offers several key advantages: enhanced security through dual verification, faster and more efficient voting, accurate automatic vote counting, and an intuitive interface that is accessible to users of all ages and technical abilities. While minor limitations such as occasional fingerprint recognition issues or network requirements for large-scale elections exist, the smart voting system provides a strong foundation

for future improvements. Potential enhancements include cloud-based storage, mobile app integration, and additional biometric verification options.

Overall, the implementation of this system highlights the potential of combining IoT, biometrics, and digital technology to create a trustworthy, efficient, and inclusive voting platform, making it a practical solution for student elections and other small-scale voting environments.

REFERENCES

[1] Poornima, P., Ranjitha, R. & Keerthana, B. "RFID and Fingerprint Based Electronic Voting Machine," International Journal of Engineering Research & Technology, Vol. 9, No. 5, pp. 1123-1126, 2020. Available: https://ijrar.org/download1.php?file=IJRAR25B3167.pdf

[2] B. A. Oke, O. M. Olaniyi, A. A. Aboaba, and O. T. Arulogun, "Multifactor authentication technique for a secure electronic voting system," Electronic Government, vol. 17, no. 3, pp. 312–338, Apr. 2021, doi: https://doi.org/10.1504/EG.2021.115999

[3] Durga, R. & Sai, P. "Biometric Voting Machines: A Security Enhancement," International Journal of Computer Science & Engineering, Vol. 12, No. 2, pp. 89-94, 2024. Available: https://ijcaonline.org/archives/volume186/number35/kumar-2024-ijca-923921.pdf

[4] Olaniyi, O., Ogunbiyi, T. & Adebayo, M. "Blockchain-Enabled Biometric Voting Systems," Journal of Secure Computing, Vol. 15, No. 3, pp. 56-67, 2022. Available: https://www.sciencedirect.com/science/article/pii/S2096720925000752

[5] Farhan, A., Khalid, H. & Raza, M. "IoT-Based Facial Recognition Voting System," IEEE Access, Vol. 12, pp. 45871-45880, 2024. Available: https://ieeexplore.ieee.org/document/10000000

[6] Espressif Systems, ESP32 technical reference manual, Espressif Documentation, 2023. Available: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf

[7] M. Grinberg, Flask web development: Developing web applications with Python, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2018.

[8] S. Das, S. Biswas, O. Das, A. Ghosh and S. De, "Smart voting machine using RFID, ingerprint and password security," International Journal of Research and Analytical Review, vol. 12, no. 2, pp. 770–778, May 2025, Available: https://www.ijrar.org/papers/IJRAR25B3167.pdf

[9] M. Nagasri, V H. Naidu, Y. K. Sri, G. Ajay and P. Pravalika, "IoT RFID electronic voting machine," International Journal For Advanced Research in Science & Technology (IJARST), vol. 12, no. 12, pp. 273–280, Dec. 2022, Available: https://www.ijarst.in/public/uploads/paper/322871676787845.pdf

[10] N. Chamanthi et al., "RFID based electronic voting machine using OTP and biometric verification," Quest Journals, 2022, Available: https://www.academia.edu/78518969/RFID_Based_Electronic_Voting_Machine_Using_OTP_and_Bio_Metric_Verification

[11] B. U. Umar, O. M. Olaniyi, A. B. Olatunde, A. A. Isah, A. K. Haq and I. T. Ajayi, "A bi-factor biometric authentication system for secure electronic voting system," 2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable Development (NIGERCON), Lagos, Nigeria, 2022, pp. 1–5, doi: https://doi.org/10.1109/NIGERCON54645.2022.9803174

[12] M. I. M. Yusop, N. H. Kamarudin, N. H. S. Suhaimi and M. K. Hasan, "Advancing passwordless authentication: A systematic review of methods, challenges, and future directions for secure user identity," in IEEE Access, vol. 13, pp. 13919–13943, 2025, doi: https://doi.org/10.1109/ACCESS.2025.3528960

[13] B. Khokher, P. Saha, N. Kumar and M. Jharait, "Electric voting machine using ATMega microcontroller for college election," 2024 1st International Conference on Communications and Computer Science (InCCCS), Bangalore, India, 2024, pp. 1–6, doi: https://doi.org/10.1109/InCCCS60947.2024.10593639

[14] N. Bhuvaneswary, C. V. Reddy, C. Aravind and K. H. Prasad, "Smart voting machine using

fingerprint sensor and face recognition," 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC), Salem, India, 2022, pp. 1159–1166, doi: https://doi.org/10.1109/ICAAIC53929.2022.9792643