# Fingerprint-Based Access Control System for College Events

Thimmapuram Srinivas Chary
*Department of Computer Science and Engineering (Data Science)*
*Geethanjali College of Engineering and Technology*
Hyderabad, India
22r11a6739@gcet.edu.in

Dyavari Ramakrishna
*Department of Computer Science and Engineering (Data Science)*
*Geethanjali College of Engineering and Technology*
Hyderabad, India
23r15a6702@gcet.edu.in

Kokkalakonda Akhil
*Department of Computer Science and Engineering (Data Science)*
*Geethanjali College of Engineering and Technology*
Hyderabad, India
22r11a6723@gcet.edu.in

Mr. Singupuram Tirupathi Rao
*Associate Professor, Department of Computer Science and Engineering (Data Science)*
*Geethanjali College of Engineering and Technology)*
Hyderabad, India
stirupathirao.cse@gcet.edu.in

***Abstract*— The Fingerprint-Based Access Control System for College Events is a biometric-based access control system designed to enhance security and automate participant management during college events. Traditional methods such as ID cards or paper passes are often inefficient, prone to duplication, and susceptible to unauthorized access. The proposed system utilizes a fingerprint sensor integrated with a microcontroller and a web-based interface to identify and authenticate individuals in real time. When a participant places their finger on the sensor, the captured fingerprint is matched with pre-stored templates in the database. Upon successful verification, access is granted and attendance is recorded digitally. The system reduces manual intervention, eliminates human error, and maintains secure attendance logs. The prototype demonstrates the use of affordable components like the R307S optical fingerprint sensor and Arduino Uno, ensuring a cost-effective, reliable, and eco-friendly solution for institutional event management.**

***Keywords—Biometric Authentication, Fingerprint Recognition, Arduino Uno, Access Control, Event Management, Microcontroller Systems, IoT Integration***

## I. INTRODUCTION

Security and identity verification are critical components in event management within academic institutions. Manual entry systems and ID cards are prone to duplication and human error. The fingerprint-based access control system offers a secure, efficient, and automated solution for managing event participants.

This system identifies individuals based on their unique fingerprint patterns. It consists of a fingerprint sensor, Arduino Uno controller, OLED display, and a relay control mechanism. Once a valid fingerprint is detected, the system grants access and stores attendance records digitally. This reduces human involvement and enhances data integrity.

## II. SYSTEM OVERVIEW

### A. Selecting the System Components

The project is implemented using easily available and affordable components such as the Arduino Uno microcontroller, R307S fingerprint sensor, OLED display, and a buzzer with LED indicators. The Arduino acts as the central unit, managing communication between the fingerprint sensor and the display module. The OLED provides on-screen messages for user guidance, while LEDs and a buzzer give real-time status feedback.

### B. Maintaining Operational Efficiency

The system ensures quick response times, completing fingerprint verification within two seconds. It stores and matches multiple fingerprints accurately without compromising speed or reliability. A Python-based bridge (Bridge.py) synchronizes data between the microcontroller and the JSON database, ensuring that all attendance and access logs remain updated in real time.

### C. Preparing the System for Deployment

Before deployment, fingerprints of all participants are enrolled and stored securely. The modular design allows the system to be easily installed at different venues and scaled for larger applications. Its intuitive interface, portability, and automation make it highly convenient, enabling colleges to efficiently manage secure access during events.

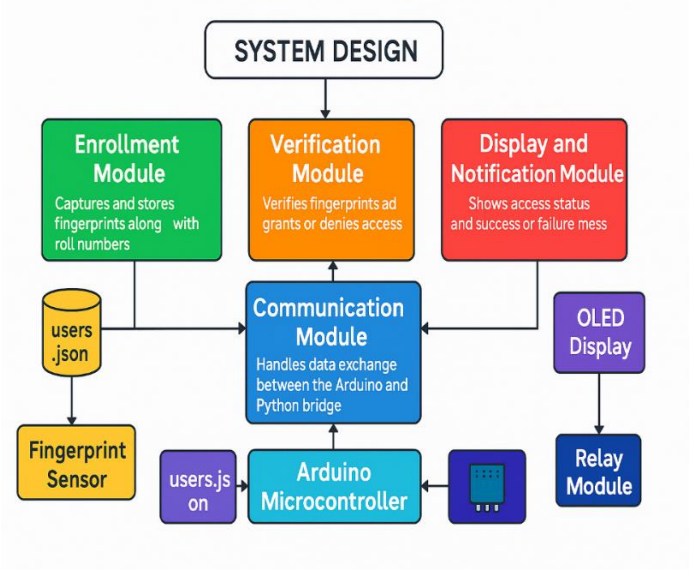



Fig 1. System Design

## III. TECHNICAL SPECIFICATIONS AND STANDARDS

### *A. Abbreviations and Acronyms*

Abbreviations and acronyms are used throughout this project to simplify technical terminology and ensure clarity in explanation. The following are the key terms frequently used in the fingerprint-based access control system:

IOT – Internet of Things, referring to the connection of devices for data exchange and automation.

OLED – Organic Light Emitting Diode, used as a compact display unit to show system messages.

JSON – JavaScript Object Notation, a lightweight data format used for storing fingerprint templates and user details.

IDE – Integrated Development Environment, specifically the Arduino IDE used for programming and uploading code.

USB – Universal Serial Bus, providing serial communication between the Arduino board and computer.

LCD – Liquid Crystal Display, an optional display component for output visualization.

API – Application Programming Interface, used for extending system functionality through software integration.

These abbreviations help maintain consistency and make the description of system components and operations easier to understand.

### *B. Units*

- The Fingerprint-Based Access Control System for College Events primarily operates using standard electrical and digital measurement units. All voltage levels are represented in volts (V), current in amperes (A), and resistance in ohms (Ω). The microcontroller and associated modules function on a 5V DC power supply, ensuring stable and safe operation of all components.
- The fingerprint sensor captures and processes digital images in pixel resolution, while time-based operations such as response or verification delays are measured in milliseconds (ms). Data storage is handled using digital units such as bytes (B) and kilobytes (KB) for lightweight database management.
- All measurements follow the International System of Units (SI), ensuring uniformity and accuracy in calculations, testing, and reporting. This standardization simplifies hardware integration and improves reliability during system deployment and analysis.

### *C. Equations*

The Project involves basic computational logic for verifying and matching fingerprint templates rather than complex mathematical formulations. The identification process is handled algorithmically using digital fingerprint templates and threshold-based matching. However, the core algorithm follows a logical comparison model that can be expressed as:

$$M = f(E, S)$$

Where:

M = Match Result

E = Enrolled Fingerprint Template

S = Scanned Fingerprint Template

A successful match occurs when the similarity score between E and S exceeds a predefined threshold T. Mathematically, this can be represented as:

$$\text{If } \text{Similarity}(E, S) \geqslant T \rightarrow \text{Access Granted}$$

$$\text{Else} \rightarrow \text{Access Denied}$$

The similarity function compares minutiae points, ridge features, and orientation data to determine fingerprint identity. This simple yet efficient logical model forms the foundation of the system's verification mechanism, ensuring real-time and accurate authentication for event participants.

Fig 2. Accessing Fingerprints

Fig 3. Analysis of Fingerprints

## IV. PROPOSED SYSTEM

The proposed Fingerprint-Based Access Control System for College Events is designed to provide a secure, automated, and efficient method for managing participant entry during college-level events. The system replaces traditional manual verification methods such as ID cards, paper passes, or attendance sheets with a biometric authentication process that ensures high accuracy and reliability. Each participant's fingerprint is unique, making it one of the most dependable identifiers for verifying individual identity and preventing unauthorized access.

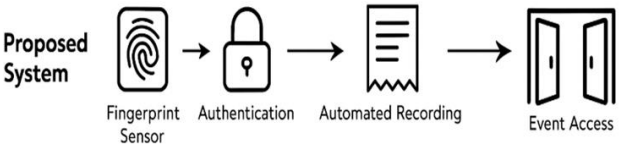



Fig 4. Proposed System

Before the event begins, all registered participants' fingerprints are enrolled and stored in the system's database along with their basic details such as name, roll number, id number and Date. During the event, when a person arrives at the entry gate, they place their finger on the fingerprint sensor. The sensor captures the fingerprint image and converts it into a digital template. This template is then compared with the pre-stored data in the microcontroller or connected database. If a match is found, the system grants access by displaying a success message and marking attendance. If the fingerprint does not match, the system denies access and alerts the event coordinator.

This automated approach eliminates the need for human intervention in verification, reduces waiting time, and prevents duplication or misuse of event passes. The system also records the time and identity of each entry, allowing organizers to monitor real-time attendance and generate accurate event reports. It can be implemented using affordable components such as an Arduino or ESP microcontroller, a fingerprint sensor module (like R307 or R305), an OLED display, and a buzzer or servo motor for gate control. Overall, the proposed system enhances security, accuracy, and efficiency in college event management. It provides a user-friendly, cost-effective, and scalable solution.

| | A | B | C | D |
|---|---|---|---|---|
| 1 | S.No | ID_No | Roll Number | Date |
| 2 | 1 | 4 | 22R11A6739 | 29-10-2025 20:11 |
| 3 | 2 | 1 | 22R11A6740 | 29-10-2025 20:15 |
| 4 | 3 | 3 | 22R11A6741 | 29-10-2025 20:17 |
| 5 | 4 | 5 | 22R11A6743 | 29-10-2025 20:19 |
| 6 | 5 | 2 | 22R11A6742 | 29-10-2025 20:18 |

Table 1. Table Showing the Unique Ids of Registered Users

## V. SYSTEM TESTING AND TROUBLESHOOTING

While developing and implementing the Fingerprint-Based Access Control System for College Events, several recurring issues were identified in both hardware and software phases. Addressing these mistakes is vital to ensure stability, accuracy, and ease of use. The most common problems observed are outlined below:

- Improper Finger Placement – Users often press the sensor too lightly, too quickly, or at an incorrect angle, causing incomplete or distorted scans. Ensuring proper placement and gentle pressure improves recognition accuracy.
- Sensor Surface Contamination – Dust, oil, or moisture on the fingerprint sensor surface can prevent accurate scanning. Regular cleaning and dry usage significantly enhance reliability.
- Duplicate Enrollment Attempts – Failing to check for existing user IDs or fingerprints during enrollment can result in database duplication. Implementing a validation process prevents redundancy.
- Power Fluctuations – Sudden voltage drops or unstable power sources can reset the microcontroller or corrupt stored data. Using a regulated 5V DC power supply and stable USB connections is recommended.
- Loose or Incorrect Wiring – Incorrect pin connections between the Arduino, fingerprint sensor, and OLED display often cause communication failures. Proper wiring and circuit testing should be conducted before deployment.
- Incorrect Serial Port Configuration – Selecting the wrong COM port or mismatched baud rate between the Arduino IDE and Python Bridge.py script leads to failed communication. Confirming settings before running the program resolves this issue.
- Data Synchronization Errors – The system may fail to update JSON data if the connection between hardware and software is interrupted. Backup mechanisms and data refresh routines should be implemented.
- Overwriting Fingerprint Templates – Enrolling new users without clearing memory may overwrite old data. Monitoring memory usage and periodic resets maintain system integrity.
- Delay in Response Time – Using inefficient loops or unoptimized code in Arduino may increase scan delays. Implementing event-driven logic helps achieve faster responses.
- Improper Grounding or Noise Interference – Electrical noise from nearby devices can disrupt serial communication. Using grounded connections and shielded cables minimizes interference.
- Lack of User Guidance – Absence of clear on-screen instructions can confuse participants. Proper display messages like "Place Finger" or "Try Again" improve user interaction.

By identifying and rectifying these issues, the fingerprint-based access system maintains high performance, data security, and operational accuracy, making it a dependable solution for college event management.

## VI. MODULE DESIGN

### *A. Enrollment Module*

The Enrollment Module is designed to register participants efficiently and securely before any event. It collects essential data such as roll number and captures the participant's fingerprint using the sensor. The fingerprint is processed, converted into a digital template, and stored in the sensor memory with a unique ID. This module includes a duplicate verification step to prevent the same fingerprint or roll number from being registered twice. Once successfully enrolled, the information is also updated in the users.json file through the

Python bridge. This ensures that every participant has a verified and unique digital identity in the system.

### *B. Verification Module*

The Verification Module handles the authentication process when a participant attempts to access the event venue. It captures the fingerprint input in real time and compares it with the previously stored templates in the sensor database. If a match is detected, the module activates the relay, allowing access, and displays "Access Granted" on the OLED screen. In case of a mismatch, it shows "Access Denied" and triggers an alert. This module is optimized for quick response, ensuring real-time verification and smooth entry management. Its primary goal is to ensure secure, accurate, and efficient access control during college events.

### *C. Data Handling Module*

The Data Handling Module is responsible for storing, updating, and managing participant information. It uses Python scripting to interact with the users.json file, which contains user IDs, roll numbers, and registration dates. During enrollment or verification, this module updates logs and maintains data consistency between the hardware and software layers. It ensures that all successful enrollments and access events are properly recorded and saved for future reference. The use of JSON allows lightweight, structured data handling and easy retrieval. This module enhances reliability by maintaining accurate, tamper-proof records of all participants and their access activities.

### *D. Display and Alert Module*

The Display and Alert Module provides real-time visual and audio feedback to participants and organizers. It uses an OLED screen to show system messages like "Place Finger," "Access Granted," or "Access Denied." Simultaneously, LEDs and a buzzer signal the system's status—green light and beep for successful verification, red light for denied access. This module ensures user clarity and system transparency, allowing users to understand the authentication outcome instantly. It operates in sync with other modules to reflect current system states accurately. The design makes the system interactive, user-friendly, and suitable for high-traffic event environments.

### *E. Communication Module*

The Communication Module is designed to establish seamless data exchange between the Arduino, Bridge.py script, and the browser interface. It ensures that every command—such as enrollment requests or verification results—is transmitted accurately through serial communication. This module converts hardware signals into readable messages and vice versa, keeping the system synchronized in real time. It is responsible for relaying responses like "ENROLL_SUCCESS" or "ACCESS_GRANTED" to the browser for display. By maintaining continuous two-way communication, it prevents data loss or delays and ensures that every participant's interaction is immediately reflected on both the hardware and software interfaces.

## VII. DATABASE DESIGN

The system uses a simple yet efficient JSON-based database to store and manage both user and event-related data. The primary file, users.json, maintains all enrolled participant records, with each entry containing attributes such as ID_No, Roll Number, Name, and Date of Registration. JSON was chosen as the storage format because of its lightweight, text-based structure, which is easy to parse, modify, and integrate with both Python and JavaScript environments. Its human-readable format also simplifies debugging and manual inspection when required.

During the enrollment phase, when a new fingerprint is captured and stored by the R307S sensor, the system automatically generates the smallest available ID number and appends a corresponding record to the database.

Additionally, the database dynamically updates through the Bridge.py script, which synchronizes communication between the Arduino microcontroller and the local JSON file in real time. This ensures that every enrollment, verification, and log entry is recorded instantly. The JSON-based structure thus provides scalability, simplicity, and reliability—ideal for managing biometric records in a lightweight, portable system.

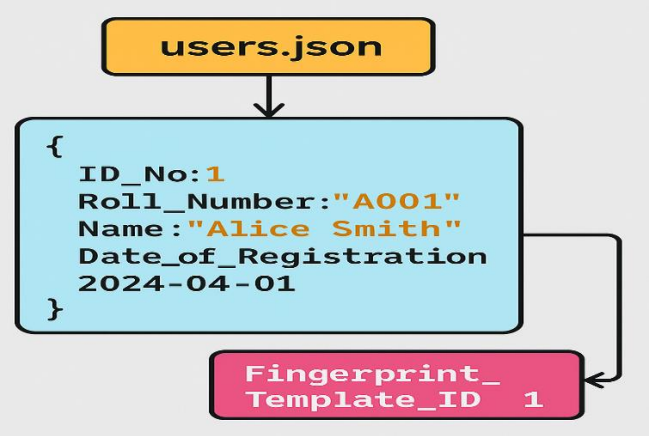



Fig 5. Data Stored

| Column | Data Type |
|---|---|
| ID_No | INTEGER |
| Roll_Number | INTEGER |
| Fingerprint_Template | BLOB |
| Date_of_Registration | TEXT |

Table 2. Database Design

The database is updated automatically through the Python Bridge when new users are added or removed, ensuring synchronization between the microcontroller and the web interface. This design allows secure, efficient, and flexible data management suitable for event-based environments where rapid registration and real-time updates are essential.

## VIII. Experimental Results

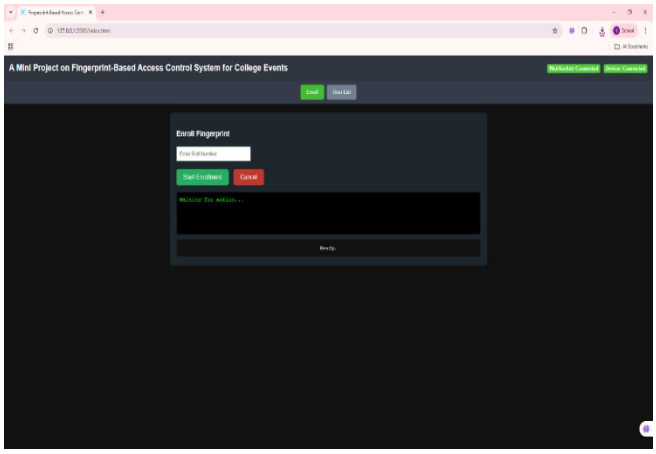
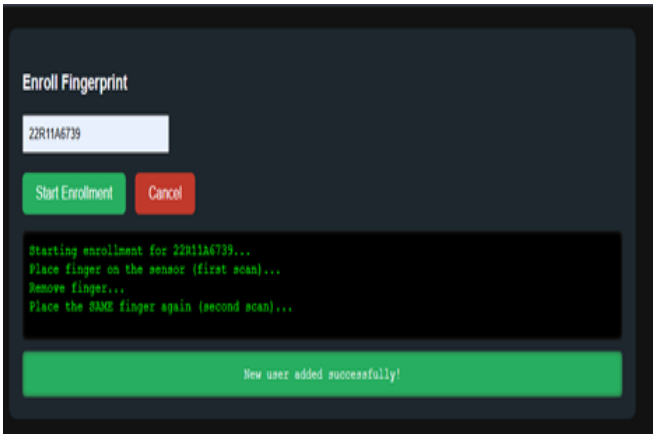


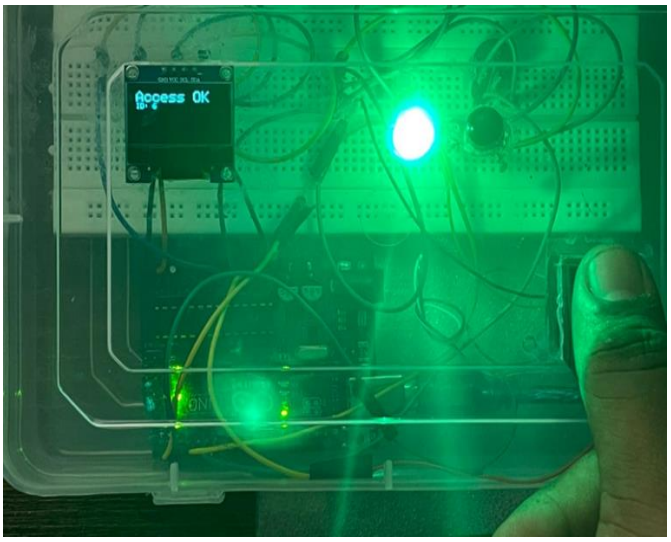

Fig 6. Home Page

(a)

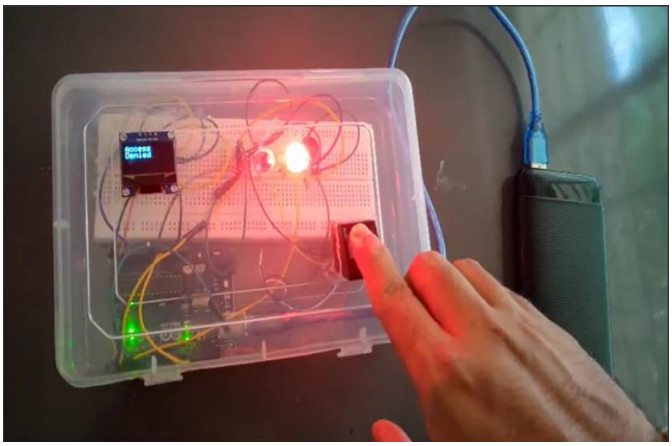



(b)

Fig 7. (a) Enrolling Fingerprint. (b) New User Added Successfully




(a)




(b)

Fig 8. (a) Access Granted. (b) Access Denied

## IX. Conclusion

The Fingerprint-Based Access Control System for College Events is a reliable and efficient solution designed to replace traditional manual attendance and access methods. It integrates both hardware and software components to ensure fast, accurate, and secure identification of participants. The system uses the Arduino Uno microcontroller as the central control unit, working in coordination with the R307S fingerprint sensor, OLED display, buzzer, and LEDs to provide real-time responses and visual feedback. Fingerprints are stored and verified using a Python-based interface that synchronizes data with a JSON database, ensuring smooth and consistent performance.

The system's modular design allows for easy installation, portability, and scalability, making it suitable for use across various college events. It enhances data security by preventing duplicate entries and unauthorized access, while its user-friendly interface ensures effortless operation even by non-technical users. The low power consumption and minimal maintenance requirements further contribute to its practicality. By automating attendance recording and access

verification, the project reduces human error, saves time, and ensures transparency in event management. Overall, the fingerprint-based access control system successfully demonstrates the application of biometric technology in educational environments, promoting digital transformation and secure, efficient event management.

## X. References

The Fingerprint-based access control system integrates both hardware and software elements to achieve secure and automated event management. Several previous studies and technical papers have contributed to the foundation of this project, focusing on biometric recognition, IoT integration, and microcontroller-based automation. These references include research on fingerprint image processing, template matching algorithms, and Arduino-based system implementation. The listed works provide insight into the design, accuracy evaluation, and performance improvement of biometric authentication systems used in educational and institutional environments.


[1] R. S. Gaonkar, Microprocessor Architecture, Programming and Applications with the 8085, 6th ed. Mumbai, India: Penram International Publishing, 2013.

[2] M. A. Mazidi, J. G. Mazidi, and R. D. McKinlay, The 8051 Microcontroller and Embedded Systems: Using Assembly and C, 2nd ed. Noida, India: Pearson Education, 2011.

[3] S. Monk, Programming Arduino: Getting Started with Sketches, 2nd ed. New York: McGraw-Hill Education, 2016.

[4] J. R. Vacca, Biometric Technologies and Verification Systems. Burlington, MA: Butterworth-Heinemann, 2007.

[5] R. Kamal, Internet of Things: Architecture and Design Principles. New Delhi, India: McGraw-Hill Education, 2021.

[6] Arduino Official Documentation, "Arduino reference," [Online]. Available: https://www.arduino.cc/reference/en/

[7] Adafruit Learning System, "Fingerprint sensor library and example projects," [Online]. Available: https://learn.adafruit.com/

[8] W3Schools, "HTML, CSS and JavaScript tutorials," [Online]. Available: https://www.w3schools.com/

[9] GeeksforGeeks, "IoT-based fingerprint attendance system using NodeMCU and R305 sensor," [Online]. Available: https://www.geeksforgeeks.org/

[10] CircuitDigest, "Fingerprint sensor interfacing with Arduino," [Online]. Available: https://circuitdigest.com/microcontroller-projects

[11] TutorialsPoint, "Web application development basics," [Online]. Available: https://www.tutorialspoint.com/

[12] A. Kumar, P. Verma, and S. Sharma, "Fingerprint-based access control system using microcontroller," Int. J. Adv. Res. Comput. Eng. Technol. (IJARCET), vol. 7, no. 4, pp. 156–161, Apr. 2018.

[13] P. Singh and R. Mehta, "IoT-based biometric authentication system for smart campus management," in Proc. IEEE Int. Conf. Comput. Intell. Commun. Netw. (CICN), 2020, pp. 1–6.

[14] M. Sharma and D. Patel, "Fingerprint recognition using minutiae and ridge feature extraction," Int. J. Comput. Appl. (IJCA), vol. 178, no. 23, pp. 1–6, Dec. 2019.

[15] S. Reddy, P. Rao, and V. Singh, "Biometric attendance system using fingerprint authentication," J. Emerg. Technol. Innov. Res. (JETIR), vol. 8, no. 5, pp. 45–50, May 2021.

[16] R. Das and K. Iyer, "Event access and security management using fingerprint biometrics," IEEE Access Conf. Proc., 2022, pp. 1–5