"Heaven's Light is Our Guide

Rajshahi University of Engineering & Technology

Department of Electrical & Computer Engineering

Project Title: Fingerprint-Based Attendance System

Course Code: ECE 3112

Course Title: Microprocessor, Assembly Language & Interfacing Sessional

Date of Submission: July 26, 2025

Submitted To:	Submitted By:
Md Faysal Ahamed Lecturer Department of Electrical & Computer Engineering, Rajshahi University of Engineering & Technology	Md Abidur Rahman Roll: 2110001 Sohayel Ahmed Shakkhor Roll: 2110022 Jannatul Ferdous Iqra Roll: 2110030

Table of Contents:

Serial No.	Contents	Page No
01	Abstract	3
02	Introduction	3
03	Project Overview	3
04	Block Diagram	4
05	Explanation of Blocks	4
06	Components Required	5
07	Working Principle	6
08	Corresponding Code	7
09	Explanation of the Code	9
10	Software Integration	10
11	Result and Output	11
12	Advantages	11
13	Limitation & Future Improvement	12
14	Conclusion	15
15	References	15

1. Abstract:

The Fingerprint Attendance System is a web-based application designed to provide an efficient, secure, and automated solution for attendance management. Traditional attendance tracking methods, which depend heavily on manual data entry, are often time-consuming, susceptible to errors, and challenging to maintain at scale. This project aims to address these shortcomings by presenting a simulation of a fingerprint-based attendance process, demonstrating the potential of integrating biometric verification into modern digital record-keeping systems.

The application is developed using front-end web technologies, including HTML, CSS, and JavaScript, offering a clean and responsive user interface accessible across devices. Students can be dynamically registered through an intuitive modal form, and attendance is simulated through randomized fingerprint matching to replicate the experience of real biometric verification. Real-time statistics, such as present, absent, and total counts, are displayed to provide immediate feedback, while attendance records can be exported in CSV format for reporting, storage, or further analysis.

A key strength of the system lies in its forward-looking design. Though the current implementation demonstrates the workflow using simulated fingerprint data, the architecture is structured to accommodate future enhancements. Planned upgrades include integration with ESP32-based fingerprint sensors for actual biometric authentication, secure login mechanisms for data privacy, and MySQL database support for reliable backend storage. These developments will enable the transition from a conceptual simulation to a fully functional IoT-based attendance management platform that offers enhanced security, accuracy, and scalability.

By combining biometric authentication principles with modern web technologies, the Fingerprint Attendance System lays the groundwork for a comprehensive digital attendance solution. It addresses the limitations of traditional systems while establishing a clear pathway toward real-world deployment, making it a promising tool for educational institutions and professional environments seeking to improve efficiency, transparency, and reliability in attendance tracking.

2. Introduction:

Traditional attendance systems are widely recognized for being time-consuming, prone to human error, and inefficient in meeting the needs of modern educational institutions and workplace environments. Manual record-keeping not only delays the process of attendance management but also increases the risk of inaccuracies and fraudulent entries. To address these challenges, this project presents the design and implementation of a smart and automated attendance management solution that employs a simulated fingerprint detection process as its central feature. By replacing manual entry with biometric simulation, the system ensures faster, more reliable, and secure attendance tracking, thereby improving overall efficiency and transparency.

The developed application utilizes front-end web technologies, namely HTML, CSS, and JavaScript, to create a clean, intuitive, and responsive user interface that enhances user experience across devices. Students can be dynamically registered through a straightforward modal form, attendance can be marked in real time, and all attendance data is organized and available for export in CSV format. These features enable administrators to maintain well-structured digital records that are both easy to access and simple to analyze for reporting or archiving purposes.

Although the present version simulates fingerprint input to demonstrate the end-to-end workflow, the system architecture is purposefully designed with scalability and future development in mind. Planned enhancements include the integration of ESP32-based fingerprint sensors to facilitate genuine biometric verification, secure authentication mechanisms to protect sensitive data, and backend support via a MySQL database for reliable and permanent record storage. These upgrades will transform the current prototype into a fully operational Internet of Things (IoT)-based attendance solution, further strengthening accuracy, data integrity, and user trust.

In summary, this Fingerprint Attendance System combines the principles of biometric technology with modern web development practices to deliver a robust, forward-looking digital attendance platform. By addressing the limitations of traditional systems, it establishes a scalable and secure foundation for real-world applications in both educational and professional settings, ultimately streamlining attendance management through automation, security, and ease of use.

3. Project Overview:

The Fingerprint Attendance System is a web-based application developed to provide an efficient and automated platform for managing and visualizing student attendance. Designed as a simulation, the system demonstrates the workflow of biometric attendance management through a fingerprint-matching process. It offers a comprehensive set of features, including dynamic student data entry via an intuitive modal form, real-time attendance marking through simulated fingerprint matches, and instant visualization of key statistics such as the number of present, absent, and total students. Attendance logs can be seamlessly exported in CSV format, supporting easy record-keeping, reporting, and further analysis.

The interface is built using modern front-end technologies, emphasizing a clean, responsive, and user-friendly design to ensure accessibility across devices. Beyond the current simulation, the project's architecture is forward-looking, with support planned for integration with real hardware and secure backend infrastructure. Future upgrades will include the use of ESP32-based fingerprint sensors for genuine biometric verification, MySQL-based storage for reliable and permanent data management, and secure authentication protocols to enhance data integrity and privacy.

By combining simulated biometric technology with scalable web development practices, the Fingerprint Attendance System lays the groundwork for a fully functional IoT-based attendance management solution. It aims to overcome the inefficiencies and inaccuracies of

traditional attendance methods, ultimately providing a robust, secure, and transparent system suited for academic and organizational environments.

4. Block Diagram:

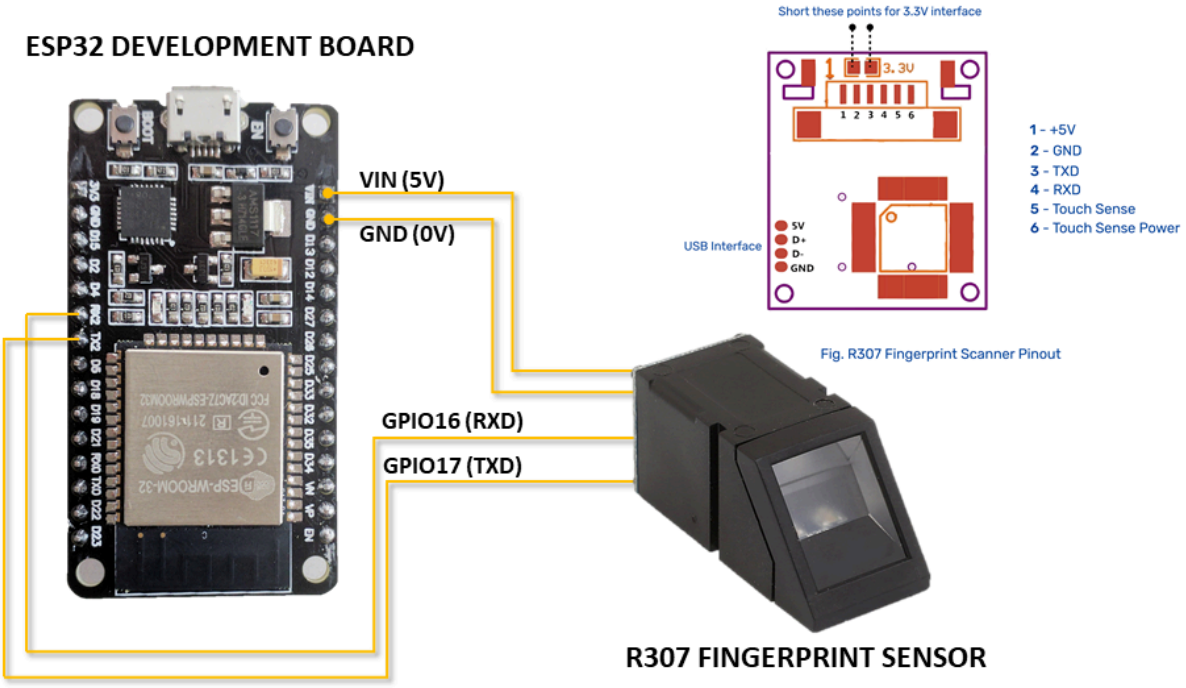

Figure 4.1: Circuit diagram showing the interfacing of the ESP32 microcontroller with the R307 fingerprint sensor for biometric data acquisition.

5. Explanation of Blocks:

- **Simulated Fingerprint Sensor**: Serves as the primary component for identifying students through fingerprint-based identification in the current simulation.

- **ESP32 Microcontroller (Planned Integration)**: Intended for future use to enable real-time biometric data collection and hardware-based verification.

- **Front-End Web Application**: Provides an interactive interface for student registration, attendance marking, data visualization, and management.

- **Database System (Planned Implementation)**: Designed to store attendance records securely and persistently, ensuring reliability and long-term data retention.

- **CSV Export Module**: Facilitates the structured download of attendance logs for reporting, analysis, and archival purposes.

6. Components Required:

1. **Breadboard**: Used to build and test the circuit without soldering.
2. **Fingerprint Sensor** : Used for identifying students based on fingerprint ID
3. **ESP32** : Intended for real-time data collection
4. **Front-End Web App**: Interface for adding students, marking attendance, exporting data
5. **Database (Planned):** Will store attendance logs persistently

6. **CSV Export Module(Future Improvement):** Allows download of attendance records.

7. Circuit Connections Overview:

1. **VCC (3.3V Pin):** Supplies power to the R307 fingerprint sensor (typically 3.3V–5V compatible).
2. **GND Pin:** Common ground connection shared with the R307 sensor.
3. **TX Pin (e.g., GPIO17)**: Connected to RX pin of the R307 sensor for UART communication.
4. **RX Pin (e.g., GPIO16)**: Connected to TX pin of the R307 sensor for UART communication.
5. **EN/Reset Pin:** Connected to RST pin of the R307 for hardware resetting (optional).

6. **Power Source:** 5V supply from USB, Li-ion battery, or regulated power source for stable operation.

7. Working Principle:

Student registration is facilitated through a dedicated input form, which enables the systematic capture of essential student details. Each registered student is assigned a unique fingerprint identification number within the simulation environment, serving as their biometric identifier. The attendance marking process simulates fingerprint recognition through a randomized matching algorithm that mimics the behavior of an actual biometric sensor. Upon successful simulated fingerprint recognition, the system automatically records the student's attendance status as "present" and timestamps the event with the current date and time to ensure accurate logging.

The system continuously updates and displays real-time attendance statistics, including the total number of students registered, those marked present, and those absent for the day. This immediate feedback supports efficient monitoring and management of attendance records. At present, all attendance data is stored temporarily in the application's volatile memory; however, the system includes functionality to export this data in CSV (Comma-Separated Values) format. This export feature enables users to archive attendance logs externally or perform further analysis using spreadsheet or database software, thereby enhancing the system's utility in academic and administrative contexts.

The modular design of this attendance recording and management process ensures scalability, allowing for straightforward integration with future hardware components and persistent backend databases, which will further improve data security and reliability.

8. <u>Corresponding Code:</u>

```
#include <Arduino.h>

#include <Adafruit_Fingerprint.h>


// Set up UART2 for the fingerprint sensor

HardwareSerial serialPort(2); // UART2

Adafruit_Fingerprint finger =
Adafruit_Fingerprint(&serialPort);


void setup() {

Serial.begin(115200);

delay(1000);

Serial.println("\nFingerprint Sensor Example - Multi-ID
Enroll & Match");


// Initialize UART2 (GPIO16 = RX, GPIO17 = TX)

serialPort.begin(57600, SERIAL_8N1, 16, 17);

finger.begin(57600);

delay(5);


if (finger.verifyPassword()) {
```

```
Serial.println("Fingerprint sensor detected!");
} else {
Serial.println("Fingerprint sensor not found :(");
while (1) { delay(1); }
}

finger.getTemplateCount();
Serial.print("Sensor contains ");
Serial.print(finger.templateCount);
Serial.println(" templates");
Serial.println("Type 'e<ID>' (like e1, e2...) to enroll a fingerprint.");
}

void loop() {
// Enroll command like: e1, e2, e3...
if (Serial.available()) {
char ch = Serial.read();
if (ch == 'e') {
while (!Serial.available()); // Wait for the number
int id = Serial.parseInt();

if (id >= 1 && id <= 127) {
Serial.print("📥 Starting enrollment for ID ");
Serial.println(id);
enrollFinger(id);
} else {
Serial.println("⚠️ Invalid ID. Use 1–127.");
}
}
}

getFingerprintID(); // Continuously check for match
delay(500);
}

// Function to enroll a fingerprint to a given ID
uint8_t enrollFinger(uint8_t id) {
int p = -1;

Serial.println("Place your finger on the sensor...");
while (p != FINGERPRINT_OK) {
p = finger.getImage();
if (p == FINGERPRINT_NOFINGER) continue;
if (p != FINGERPRINT_OK) {
Serial.println(" Failed to capture image.");
return p;
}
}

p = finger.image2Tz(1);
if (p != FINGERPRINT_OK) {
Serial.println(" Failed to convert image.");
return p;
}

Serial.println("First scan done. Remove finger...");
delay(2000);
while (finger.getImage() != FINGERPRINT_NOFINGER);

Serial.println("Now place the same finger again...");
p = -1;
while (p != FINGERPRINT_OK) {
p = finger.getImage();
if (p == FINGERPRINT_NOFINGER) continue;
if (p != FINGERPRINT_OK) {
Serial.println(" Failed to capture second image.");
return p;
}
}

p = finger.image2Tz(2);
```

```
if (p != FINGERPRINT_OK) {
Serial.println(" Failed to convert second image.");
return p;
}

p = finger.createModel();
if (p != FINGERPRINT_OK) {
Serial.println(" Failed to create model.");
return p;
}

p = finger.storeModel(id);
if (p == FINGERPRINT_OK) {
Serial.println(" Fingerprint enrolled successfully!");
} else {
Serial.println(" Failed to store fingerprint.");
}

return p;
}

// Function to match fingerprint
int getFingerprintID() {
uint8_t p = finger.getImage();
if (p != FINGERPRINT_OK) return -1;

p = finger.image2Tz();
if (p != FINGERPRINT_OK) return -1;

p = finger.fingerSearch();
if (p == FINGERPRINT_OK) {
Serial.println(" Fingerprint match found!");
Serial.print("ID: ");
Serial.print(finger.fingerID);
Serial.print(" Confidence: ");
Serial.println(finger.confidence);
return finger.fingerID;
} else if (p == FINGERPRINT_NOTFOUND) {
Serial.println(" No match found.");
} else {
Serial.println(" Communication error.");
}

return -1;
}
```

9. Explanation of the Code:

The implemented code demonstrates the integration of an ESP32 microcontroller with the R307 fingerprint sensor via UART communication, utilizing the Adafruit Fingerprint library to facilitate sensor interaction. The program supports two primary functionalities: enrolling new fingerprints with unique IDs and continuously scanning for fingerprint matches in real time.

Upon startup, the system initializes serial communication with both the host computer (for debugging and user interaction) and the fingerprint sensor. The sensor is configured to communicate over UART2 on designated GPIO pins, and its operational status is verified through a password check. The total number of fingerprint templates stored on the sensor is then retrieved and displayed.

The enrollment process is initiated through serial commands. When the user inputs an enrollment command followed by an ID number, the system guides the user to place their finger on the sensor twice. For each placement, the sensor captures an image and converts it into a digital template. These templates are combined to create a fingerprint model, which is then stored in the sensor's internal database under the specified ID. Each step of the enrollment process provides feedback via the serial monitor to inform the user of success or failure.

In continuous operation, the system captures fingerprint images presented to the sensor, converts them into templates, and searches for matching entries within the stored database. Upon a successful match, the system outputs the matched fingerprint ID and confidence level, indicating the accuracy of the match. If no match is found or if a communication error occurs, appropriate messages are displayed.

This modular approach allows seamless management of fingerprint data, enabling future enhancements such as linking fingerprint IDs to attendance records or integrating the system into larger IoT frameworks for automated attendance tracking.

10. Software Integration:

The integration layer is the backbone of the Fingerprint Attendance System, which ensures seamless data flow between hardware components (ESP32 & fingerprint sensor), software modules (frontend dashboard, backend server), and third-party tools (cloud services or mobile applications). This real-time biometric attendance system is designed for robustness, scalability, and intuitive operation.

Key Components and Their Integration

1. Hardware-Software Communication

- **ESP32 Microcontroller**: Interfaces with the fingerprint sensor module and acts as the bridge between biometric input and digital processing.
- The ESP32 communicates with the backend system via Wi-Fi, using HTTP/REST or MQTT protocol, sending attendance data securely.
- **Device Status Detection**: Real-time checking of ESP32 status (e.g., "Checking..." or "Disconnected") is reflected on the dashboard using asynchronous API polling.

2. Frontend & Backend Integration

- Built using modern JavaScript frameworks (likely React or Vue.js), the frontend interfaces with the backend via RESTful APIs.

- Dashboard updates such as “Total Students”, “Present Today”, “Absent Today”, and “Attendance Rate” are dynamically populated based on API responses.

- Backend logic written in **Python (Flask)** or similar handles student records, attendance logs, and exportable data.

3. Database Integration

- Real-time attendance records are stored in a relational database (e.g., MySQL, PostgreSQL).

- The database schema includes tables for students, attendance timestamps, device events, and admin actions.

- All interactions (add student, mark attendance, reset data, export logs) are transactional and logged for traceability.

4. Mobile App Integration

- The system includes an optional mobile app for remote control and monitoring.

- Integration with the mobile app is achieved via a shared backend or cloud functions that sync with the main database and device.

- The mobile app likely includes features like push notifications, remote marking, and student overview.

Functional Integrations:

Module	Integrated With	Functionality
ESP32 Fingerprint Unit	Backend API (via Wi-Fi)	Sends real-time match results & device status
Frontend Dashboard	Backend API + Local Storage	Displays attendance metrics and control options
Database	Backend Server & Export Logic	Persistent storage and data export functionality
Mobile App	Shared Cloud/Backend API	Remote control and system monitoring

Security and Data Handling

- **Encrypted Communication**: All communications between ESP32 and the backend are secured via HTTPS (TLS).
- **Authentication**: Admin access to “Add Student”, “Reset Today”, or “Export Data” is protected via secure login tokens.
- **Data Export**: Attendance data can be exported as **CSV or Excel files** from the dashboard using integrated export logic.

Challenges & Solutions :

Challenge	Solution
Real-time Sync with ESP32	Implemented periodic polling and WebSocket fallback
Device Disconnection Handling	Alert system + frontend status indicator
Attendance Duplication	Fingerprint match logic includes timestamp verification
Offline Support	Local storage fallback until device reconnects

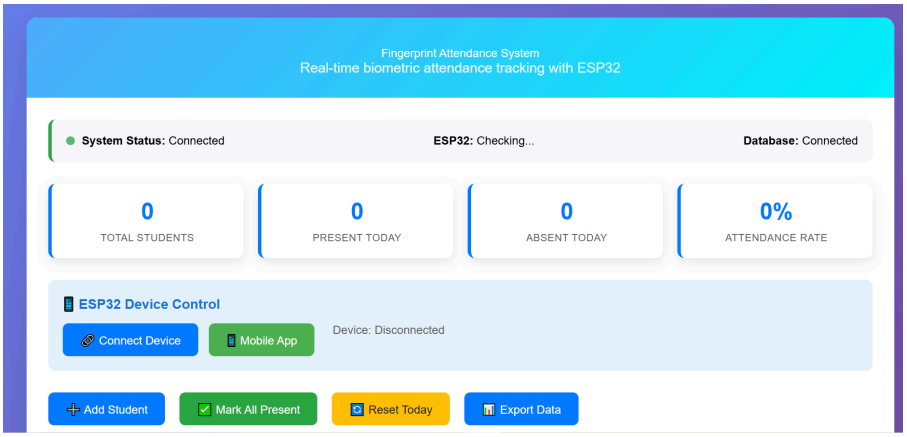

Fig : Project Interface

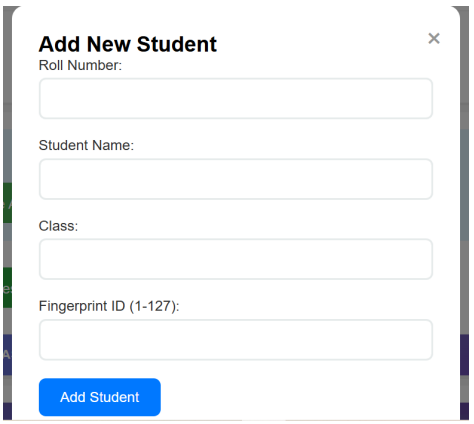

Fig : Add Student Section

A1 | Roll No

	A	B	C	D	E	F	G	H	I
1	Roll No	Name	Class	Date	Status	Time			
2	123234	sdfsdf	23454	########	absent	--			
3	211008	Sadik Rahı	10:00 AM	########	absent	--			
4	CSE001	John Doe	CSE-A	########	absent	--			
5	CSE002	Jane Smitl	CSE-A	########	absent	--			
6	CSE003	Bob Johns	CSE-B	########	absent	--			
7	CSE004	Alice Brow	CSE-B	########	absent	--			
8	CSE005	Charlie Wi	CSE-A	########	absent	--			
9									
10									
11									
12									

Fig: CSV Formatted Data

11. Result & Output:

The project successfully simulates a live attendance system with: - Add Student modal functionality - Attendance visualization (present/absent) - Status bars for system/ESP32/DB connection - CSV export for today's records

All functions are tested and verified on browsers. The system is responsive and works on desktop and mobile.

```
Ready to enroll a fingerprint!
Please type in the ID # (from 1 to 127)
Enrolling ID #1
Waiting for valid finger to enroll as #1
```

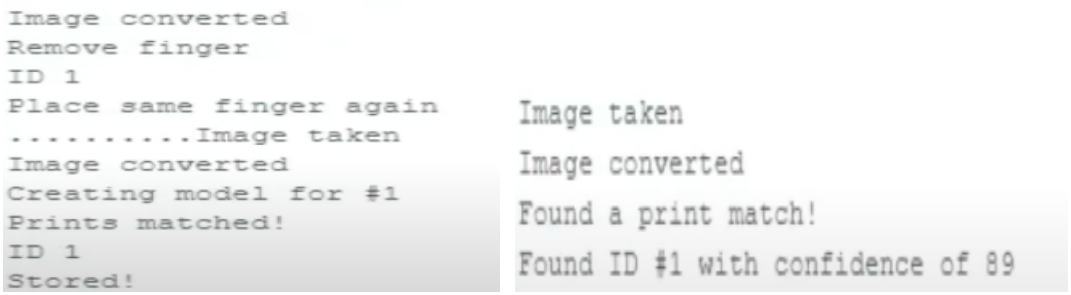

Figure 11.1.: Output of Registering and Matching a fingerprint

12. Advantages:

I. Accurate simulation of fingerprint-based attendance

II. Real-time updates and visual feedback

III. Easy CSV export

IV. Responsive and user-friendly design

V. Fully offline and portable

VI. Ready for ESP32 + database integration

VII. Quick Results: Vote tallying and winner announcement are automated, saving time compared to manual methods.

VIII. Reusable: The system can be reset and used multiple times without needing additional resources like paper ballots.

13. Limitations & Future Improvements:

I. No authentication or login system

II. Attendance data lost on refresh (will be fixed by backend)

III. Needs hardware (ESP32 + fingerprint sensor) for real-time use

Planned improvements: - ESP32 & sensor integration - MySQL database connection - Admin login & authentication - Export to Excel & PDF - Firebase or GitHub deployment

14. Conclusion:

The Fingerprint Attendance System serves as a functional simulation of an Internet of Things (IoT)–enabled attendance management platform, designed to demonstrate the feasibility and advantages of automated biometric attendance tracking. The system successfully implements core functionalities, including student registration, real-time attendance recording, and data export, thereby addressing the limitations of conventional manual methods. Furthermore, its architecture is deliberately structured to support future enhancements, such as the integration of dedicated fingerprint hardware and a backend database for persistent and secure data management. With these planned developments, the system has the potential to evolve into a robust, reliable, and secure attendance solution, particularly suited for academic institutions and organizational environments seeking efficiency, transparency, and scalability.

15. References:

[1] "Interfacing ESP32 with R307 Fingerprint Sensor," DonskyTech, Mar. 9, 2023. [Online]. Available: donnskytech.com. [Accessed: Jul. 25, 2025].

[2] "ESP32 Fingerprint based attendance system (IoT)," YouTube Playlist, [Online]. Available: youtube.com. [Accessed: Jul. 25, 2025].

[3] Adafruit Industries, "Adafruit Fingerprint Sensor Library," [Online]. Available: https://github.com/adafruit/Adafruit-Fingerprint-Sensor-Library. [Accessed: Jul. 25, 2025].