ATTENDRO

A PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DIPLOMA IN

COMPUTER ENGINEERING

SUBMITTED BY

[YOUR NAME]

(Roll No: [YOUR ROLL NO])

DEPARTMENT OF COMPUTER ENGINEERING

[NAME OF YOUR INSTITUTE]

[CITY, STATE - PIN CODE]

ACADEMIC YEAR: 2025-2026

CERTIFICATE

This is to certify that the project report entitled **"ATTENDRO"** is a bonafide work carried out by **[Mr./Ms. YOUR NAME]** (Roll No: [YOUR ROLL NO]) of the **Department of Computer Engineering**, in partial fulfillment of the requirements for the award of the Diploma in Computer Engineering during the academic year **2025-2026**.

The work contained in this report has not been submitted previously for any other degree or diploma to this or any other university/institute.

[Name of Guide] Project Guide	**[Name of HOD]** Head of Department

[Name of Principal]
Principal

EXTERNAL EXAMINER

ACKNOWLEDGMENT

It gives me great pleasure to present the project report on **"Attendro"**. The successful completion of this project would not have been possible without the guidance and support of many respected individuals.

First and foremost, I would like to express my deepest gratitude to my project guide, **[Name of Guide]**. Their deep knowledge of IoT and patience in explaining the concepts helped me navigate through the technical challenges of working with the ESP32 and fingerprint sensors. Whenever I was stuck with a bug or a hardware issue, their advice pointed me in the right direction.

I am also sincerely thankful to **[Name of HOD]**, Head of the Department, for providing us with the necessary laboratory facilities and encouragement to take up this project.

I extend my thanks to our Principal, **[Name of Principal]**, for providing an academic environment that fosters innovation and practical learning.

I would also like to mention the support of the non-teaching staff and lab assistants who helped arrange the necessary components and tools for our hardware assembly.

Last but not least, I thank my parents and friends. Their constant encouragement and moral support kept me motivated, especially during the long nights of coding and testing.

[Your Name]

Roll No: [Your Roll No]

Department of [Your Dept]

TABLE OF CONTENTS

CHAPTER–1 INTRODUCTION

1.1 OVERVIEW

In any educational institute, attendance is one of the most important activities. It is used to decide if a student is eligible for exams and to check their regularity. Currently, most colleges still use the old method where teachers call out roll numbers and mark them on a paper register. As the number of students increases and the academic schedule becomes more complex with practical batches and extra lectures, this manual method is becoming difficult to manage.

The main issue with current methods is that they do not guarantee that the student is actually attending the specific subject. A student might be present in the college premises but missing the lecture. To solve this, we are developing "Attendro", a smart attendance system. This project combines a hardware device and a mobile app to make sure that attendance is marked only when the teacher is present in the class and starts a session. It is designed to be simple, secure, and accurate for daily college use.

1.2 PROBLEM STATEMENT

During our study of the current system, we observed several problems that teachers and the administration face daily.

1.2.1 Time Consumption

Taking attendance manually takes a long time. In a lecture of 60 minutes, nearly 10 to 15 minutes are wasted just in calling out names. This reduces the actual teaching time. If there are multiple lectures in a day, a significant amount of time is lost just in administrative work.

1.2.2 Proxy Attendance

This is the most common problem in colleges. When a teacher calls a roll number, another student often answers "Present" for their absent friend. It is difficult for a teacher to spot this in a large class of 60 or more students. Even with simple signature sheets, students can easily sign for others.

1.2.3 Lack of Specificity

Existing biometric machines are usually kept at the college main gate. They only track when a student enters or leaves the building. They cannot track if the student attended the Chemistry lecture or the Physics practical. There is no link between the fingerprint scan and the subject time-table.

1.3 PROPOSED SOLUTION

To solve these problems, we are building the Attendro system. The idea is to make attendance "session-based". This means attendance is not open all the time. It is only allowed when a teacher starts it.

The system works in three parts. First, the teacher uses the Faculty App on their phone to select the subject and batch (like Batch A or Batch B) and starts the session. Second, the student scans their finger on a portable device (ESP32) which is kept in the class. The device checks if the student belongs to that batch and if the session is active. Third, the data is saved locally and sent to the cloud server.

This approach ensures two things: only valid students can mark attendance, and they can only do it when the teacher is conducting the lecture. If the internet is down, the device saves the data and uploads it later, so no attendance is lost.

1.4 OBJECTIVES

The main goals of our project are as follows:

1. **To Eliminate Proxies:** Using fingerprint sensors ensures that one student cannot mark attendance for another.

2. **To Save Time:** The automatic process is much faster than roll calls, allowing teachers to focus on teaching.

3. **To Enforce Rules:** The system should automatically reject students who try to enter the wrong batch (e.g., a Batch B student trying to attend a Batch A practical).

4. **To Work Offline:** The system must be reliable even if the college Wi-Fi is not working during the lecture.

1.5 SIGNIFICANCE

This project is significant because it brings "smart features" to a basic daily task without being too expensive. By using the ESP32 microcontroller, we are keeping the cost low. The system's ability to make decisions locally (Edge Computing) means it is faster and more reliable than systems that always need the internet. It provides a practical solution to the real-world problem of managing student records in diploma and engineering colleges.

1.6 REPORT STRUCTURE

This project report is organized into several chapters. Chapter 1 introduces the project and the problem. Chapter 2 discusses the Literature Survey and existing systems. Chapter 3 defines the Scope of the project. Subsequent chapters will explain the methodology, system design, and the final results of our work.

CHAPTER–2 LITERATURE SURVEY

2.1 INTRODUCTION

Before starting our project, we studied various methods used for attendance in schools and colleges. A literature survey helps us understand what solutions already exist and what problems are still unsolved. We looked at traditional manual methods as well as modern digital systems like RFID and biometric scanners. This helped us identify the gaps that our project needs to fill.

2.2 EXISTING SYSTEMS

2.2.1 Manual Paper Registers

The most common method is the manual roll call. Teachers carry a physical register and mark "P" or "A" against the student's name. While this is simple and costs nothing, it has many disadvantages. Paper records can get damaged or lost. Calculating monthly attendance percentage from these papers is a tedious job for teachers. Most importantly, it is very easy for students to cheat by answering for their absent friends.

2.2.2 RFID Card Systems

Some colleges use Radio Frequency Identification (RFID) cards. Students are given ID cards that they tap on a reader to mark attendance. Ideally, this is faster than manual roll calls. However, our study found a major flaw: the card marks the attendance, not the student. A student can easily give their ID card to a friend who can tap it for them. So, while it is digital, it does not stop proxy attendance.

2.2.3 Static Fingerprint Scanners

Biometric fingerprint scanners are very secure because fingerprints cannot be shared like ID cards. Many offices use these. However, in colleges, these machines are usually "static", meaning they are fixed at the door. They only record the time a student enters the college. They do not tell us if the student attended the Computer Network lecture or the Java practical. They lack "context" about the daily timetable and batches.

2.2.4 Mobile App Systems

There are apps that let students mark attendance on their own phones using GPS. The problem with this is that GPS is not always accurate inside big buildings. Also, students use "fake location" apps to fool the system into thinking they are in class while they are at home. Also, not every student may have a charged smartphone with an active internet pack every day.

2.3 COMPARISON OF TECHNIQUES

We compared these different methods to see where our project stands. The table below shows a simple comparison:

Method	**Prevents Proxy?**	**Batch Wise?**	**Offline Mode?**	**Teacher Control?**
Manual Call	No	Yes	Yes	Yes
RFID Card	No	No	Sometimes	No
Static Biometric	Yes	No	Sometimes	No
Attendro (Ours)	**Yes**	**Yes**	**Yes**	**Yes**

2.4 IDENTIFIED GAP

From the survey, we found that there is no system that is both secure (biometric) and smart enough to understand the college timetable (batches and subjects). Existing systems are either too simple (manual) or too rigid (static scanners). There is a need for a system that allows the teacher to decide "When" and "Who" can mark attendance. Also, many systems fail if the internet disconnects, which is a common issue in our classrooms.

2.5 PROBLEM DEFINITION

Based on these findings, we defined our main problem statement:

"To design a low-cost, smart attendance system that uses fingerprints to stop proxies, allows teachers to control sessions for specific batches, and continues to work even when the internet is not available."

2.6 CHAPTER SUMMARY

In this chapter, we reviewed the current methods of attendance marking. We saw that manual systems are slow and prone to errors, while existing digital systems like RFID and GPS have security holes. We concluded that a new system is needed that combines the security of biometrics with the flexibility of teacher control. This leads us to the next chapter where we will define the exact scope of our project.

CHAPTER–3 SCOPE OF THE PROJECT

3.1 INTRODUCTION

The "Scope" of a project tells us exactly what the project will do and what it will not do. It sets the boundaries for our work. For the Attendro project, we are focusing on solving the attendance problems of diploma and engineering colleges, specifically for practical batches and theory lectures.

3.2 FUNCTIONAL SCOPE

The functional scope explains the features we are building. The system is divided into three main parts: the App for teachers, the Hardware for students, and the Database.

3.2.1 Faculty Mobile App

The app is for teachers only. A student cannot log in here. The main functions are:

- **Teacher Login:** Secure login so that unauthorized people cannot control the system.
- **Start Session:** The teacher can select the Class, Subject, and specific Batch (like A1, A2, B1) and turn on the attendance mode.
- **Live Count:** The teacher can see on their screen how many students have marked present in real-time.

3.2.2 The Hardware Device (ESP32)

This is the physical box that will be kept in the classroom. Its functions include:

- **Fingerprint Matching:** It matches the student's finger against stored templates. We are using roll numbers as IDs (Roll 1 to 120).
- **Smart Checking:** It doesn't just say "Match Found". It checks if that student is allowed in the current batch. If a Batch B student tries to mark attendance in a Batch A practical, the device will say "Wrong Batch" on the screen.

- **Offline Memory:** If the internet goes down, the device saves the attendance in its own memory and uploads it when the net comes back.

3.2.3 Backend / Database

We are using Supabase as our cloud database. Its job includes:

- **Data Storage:** Safely storing student names, roll numbers, and dates.
- **Reports:** Generating simple reports that teachers can download to see who was absent.

3.3 TECHNICAL SCOPE

Technically, we are working with the following tools and limits:

- **Hardware:** ESP32 Microcontroller with an R307 Fingerprint Sensor and an OLED display.
- **Software:** The app is built using React and runs on Android phones.
- **Capacity:** One device can store up to 120 fingerprints, which is enough for two standard divisions of students.
- **Power:** The device works on a standard 5V USB power supply, so it can run on a power bank.

3.4 LIMITATIONS

Every project has some limits. For this version of Attendro, we have decided on the following:

- **No Face Recognition:** We are only using fingerprints because face cameras are expensive and need more processing power.
- **Single College Use:** This system is designed for one college campus, not for managing multiple colleges at once.
- **No Parent SMS:** Sending SMS costs money per message, so we are not including that feature right now to keep the project cost-free.

3.5 FUTURE ENHANCEMENTS

In the future, if we have more time and budget, we plan to add more features. We could add Face Recognition to make it touch-less. We could also connect it to the college's main ERP software so that marks are updated automatically. Another idea is to use AI to predict which students are likely to have low attendance before it becomes a problem.

3.6 CHAPTER SUMMARY

In this chapter, we defined what our project will deliver. We explained the role of the Teacher App and the ESP32 Device. We also clarified that while we are using advanced features like Offline Sync and Batch-checking, we are avoiding expensive features like SMS for now. This keeps our project realistic and achievable within our diploma timeline. The next chapter will explain the Design and Architecture of the system.

CHAPTER–4 METHODOLOGY

4.1 INTRODUCTION

Methodology describes the step-by-step approach we followed to build the Attendro system. Instead of trying to build everything at once, we divided the project into three smaller, manageable parts: the Hardware Module, the Software App, and the Cloud Integration. This modular approach helped us test each part separately before connecting them together. This chapter explains how we designed and implemented each of these components.

4.2 SYSTEM ARCHITECTURE OVERVIEW

The system works on a "Client-Server" model but with a smart twist known as "Edge Computing".

In a traditional system, the device sends every fingerprint to the server to check if it's correct. This is slow and stops working if the internet fails. In our approach, we moved the decision-making logic to the "Edge" (the ESP32 device).

Our flow is simple:

1. The **Cloud** holds the main data (Student lists, Timetables).
2. The **Faculty App** triggers a session and tells the Cloud to "Open Attendance".
3. The **ESP32 Device** downloads the rules for the current session (e.g., "Only Batch A allowed").
4. The Device then works independently to scan and verify students.
5. Finally, the Device uploads the results back to the Cloud.

4.3 HARDWARE IMPLEMENTATION

For the hardware, we chose the ESP32 microcontroller because it has built-in Wi-Fi and Bluetooth, and it is powerful enough to process logical conditions locally.

4.3.1 Component Interfacing

We connected the R307 Optical Fingerprint Sensor to the ESP32 using serial communication (UART). The sensor stores the fingerprint templates internally. We assigned IDs to these fingerprints matching the student roll numbers (ID 1 = Roll 1).

We also added a 0.96-inch OLED display using the I2C protocol. This is crucial for user experience. It shows messages like "Connecting...", "Place Finger", or specific error messages like "Wrong Batch", so the student knows exactly what is happening.

4.4 SOFTWARE DEVELOPMENT

The software side manages the sessions and stores the data.

4.4.1 Faculty Web Application

We built a Progressive Web App (PWA) using React.js. We prioritized a "Mobile-First" design because teachers will use this on their smartphones in class. The interface allows the teacher to select the Subject, Theory/Practical mode, and specific Batches. Once they click "Start Session", the app generates a unique session token in the database.

4.4.2 Database Design (Supabase)

We used Supabase (an open-source alternative to Firebase) for our backend. We created relational tables to link Students, Subjects, and Attendance logs. The key innovation here is the "Active Session" table, which acts as a bridge between the Teacher's app and the ESP32 device.

4.5 THE ALGORITHM / WORKFLOW

The core logic of our system follows a specific algorithm to ensure security and accuracy:

Step 1: Session Check

The device wakes up and checks the server: "Is there an active session for this classroom?" If No, it shows "Idle". If Yes, it downloads the session rules (e.g., Subject: Physics, Batch: A).

Step 2: Biometric Input

A student places their finger. The sensor compares it and finds a match (e.g., ID #25).

Step 3: Edge Validation (The "Smart" Part)

The ESP32 checks three conditions locally:

- Is ID #25 part of the allowed Batch (Batch A)?

- Has ID #25 already marked attendance for this session?

- Is the session time still valid?

Step 4: Output & Storage

If all checks pass, it beeps and shows "Present: Roll 25". It saves this record in its internal memory.

Step 5: Synchronization

A background process constantly tries to upload the saved records to the Supabase cloud. If the internet is off, it keeps trying until successful.

4.6 TOOLS AND TECHNOLOGIES USED

We used a mix of hardware and software tools to build this project:

- **Microcontroller:** ESP32 DevKit V1
- **Sensor:** R307 Optical Fingerprint Module
- **Display:** SSD1306 0.96" OLED
- **Programming Language (Hardware):** C++ (Arduino IDE)
- **Frontend Framework:** React.js with Tailwind CSS
- **Backend/Database:** Supabase (PostgreSQL)

4.7 CHAPTER SUMMARY

In this chapter, we explained how we built Attendro using a modular approach. We detailed the interaction between the Cloud, the App, and the Device. We described the "Smart Validation Algorithm" that runs on the device to filter students by batch. This methodology ensures that our system is not just a simple recorder, but an intelligent tool that enforces academic rules. In the next chapter, we will present the results and screenshots of the working system.

CHAPTER–5 DETAILS OF DESIGN, WORKING AND PROCESSES

5.1 INTRODUCTION

Design is the blueprint of any engineering project. Before writing a single line of code or soldering a wire, we created detailed diagrams to visualize how the different parts of Attendro would talk to each other. This chapter explains the internal structure of our system, the database tables we designed, and the step-by-step process of how a daily attendance session flows from start to finish.

5.2 SYSTEM ARCHITECTURE

The Overall System Architecture diagram shows the high-level view of our project. It illustrates the three main "actors" in our system: The Teacher (using the App), The Student (using the Device), and The Cloud (storing the data).

[Diagrams generated by JS - Please See HTML Version for Visuals]

```
graph TD
subgraph "Teacher Side"
Teacher[Faculty App] -->|1. Starts Session| Cloud(S
end

subgraph "Cloud Backend"
Cloud -->|2. Returns Active Session| ESP32[ESP32 De
ESP32 -->|5. Syncs Data| Cloud
end

subgraph "Classroom / Edge"
Student((Student)) -->|3. Scans Finger| Sensor[R307
Sensor -->|Raw Data| ESP32
```

```
ESP32 -->|4. Validates Locally| Decision{Valid Batc
Decision -- Yes --> OLED[Show 'Present']
Decision -- No --> OLED_Err[Show 'Error']
end

style Cloud fill:#e1f5fe,stroke:#01579b
style ESP32 fill:#fff3e0,stroke:#e65100
style Teacher fill:#e8f5e9,stroke:#2e7d32
```

Figure 5.1: Block Diagram of System Architecture

As seen in Figure 5.1, the central component is the "Supabase Cloud". The Teacher App sends commands like "Start Session" to this cloud. The ESP32 device then fetches these commands via Wi-Fi. This separation ensures that the teacher doesn't need to be physically connected to the fingerprint scanner; they can control it wirelessly from their desk.

5.3 DATABASE DESIGN

A good database is crucial for an organized system. We designed our database using relational tables to keep data consistent. The main tables are:

5.3.1 Users Table

This is the master list of everyone in the college. It stores columns for:

- **User ID:** Unique computer ID.
- **Role:** Teacher or Student.
- **Roll Number:** For students (e.g., 45).
- **Fingerprint ID:** The ID number stored inside the sensor (e.g., ID 45).

5.3.2 Sessions Table

This table records every lecture that happens. It is the "History" book of the college.

- **Session ID:** Unique ID for the lecture.
- **Subject:** E.g., Java Programming.
- **Batch:** E.g., Batch A, Batch B, or All.
- **Status:** Active or Completed.

5.3.3 Attendance Logs Table

This is where the actual attendance is marked.

- **Session ID:** Links to the lecture.
- **Student ID:** Who was present.
- **Timestamp:** Exact time of the scan.
- **Status:** Present or Late.

[Diagrams generated by JS - Please See HTML Version for Visuals]

```
erDiagram
USERS {
string userID PK
string name
string role
int rollNumber
}
SESSIONS {
string sessionID PK
string subject
string batch
timestamp startTime
}
```

```
ATTENDANCE_LOGS {
    string logID PK
    timestamp markedAt
    string status
}

USERS ||--o{ SESSIONS : "teacher_starts"
SESSIONS ||--o{ ATTENDANCE_LOGS : "logs_for"
USERS ||--o{ ATTENDANCE_LOGS : "student_marks"
```

Figure 5.2: Database ER Diagram

5.4 WORKING PROCESS

The working of the system can be best understood by following a "User Journey" or a flowchart.

5.4.1 The Fingerprint Enrollment Process

Before the system can be used, students must register their fingerprints. This is a one-time process efficiently handled by the admin.

1. Admin enters "Enroll Mode" on the device.
2. Student enters their Roll Number (e.g., 25).
3. Reviewer asks student to place finger twice for confirmation.
4. Device saves the template at ID #25.

Now, whenever ID #25 is scanned, the system knows it is that specific student.

5.4.2 The Daily Attendance Workflow

This is what happens during a regular college lecture:

Step 1: Initiation

The teacher walks into the class and opens the app. They select "Data Structures Lab - Batch B" and tap "Start". A green signal is sent to the cloud.

Step 2: Device Activation

The class ESP32 device, which was showing "Idle", detects the new session. It displays "Session Active: Batch B".

Step 3: Verification Logic

A student places their finger. The sensor identifies it as Roll Number 10.

The device checks its internal logic: "Is Roll 10 in Batch B?"

If Yes: Screen shows "Present".

If No: Screen shows "Wrong Batch".

Step 4: Data Sync

The "Present" mark is instantly uploaded to the teacher's phone screen. The dashboard count updates from 14 to 15.

5.5 HARDWARE INTERFACING DETAILS

Connecting the hardware components correctly was critical. We used the following pin configurations:

- **R307 Sensor:** TX/RX pins connected to ESP32 Serial2 (GPIO 16/17) for data transfer. 5V VCC ensures it has enough power to scan quickly.
- **OLED Display:** SDA/SCL pins connected to ESP32 I2C pins (GPIO 21/22). This allows us to send text and simple graphics to the screen.

5.6 CHAPTER SUMMARY

This chapter detailed the internal "brain" of the Attendro system. We looked at the database structure that keeps data organized and the workflow logic that ensures only the right students are marked present. We also explained the hardware connections. These designs ensure that the system is robust, logical, and easy to use for both teachers and students. The next chapter will discuss the actual results achieved from implementing this design.

CHAPTER–6 RESULTS AND APPLICATIONS

6.1 INTRODUCTION

After designing and building the Attendro system, the next crucial step was to test if it actually worked in real-life scenarios. Theories are fine on paper, but engineering is about practical results. In this chapter, we present the visual outcomes of our project—screenshots of the app and photos of the device—and discuss where this system can be applied to solve real-world problems.

6.2 PROJECT RESULTS

We successfully integrated the hardware and software components. Below are the results shown through different stages of the system's operation.

6.2.1 Hardware Prototype

The final hardware build is compact and portable. We managed to fit the ESP32 microcontroller, the fingerprint sensor, and the OLED display into a neat package. When powered on, the device successfully connects to the college Wi-Fi within seconds.

[Paste Photo of the Final Hardware Device Here]

Figure 6.1: The Complete Attendro Hardware Prototype

6.2.2 The Teacher's Dashboard

The most visible part of our project for the faculty is the dashboard. As shown in the figure below, the interface is clean and simple. It allows the teacher to select a Class, Subject, and Batch. We tested this on both a laptop and a smartphone, and the responsive design adapted perfectly to both screens.

[Paste Screenshot of Teacher Dashboard Here]

Figure 6.2: Teacher's Control Dashboard

6.2.3 Real-time Logs

One of our key goals was "real-time" updates. During testing, we observed that when a student scanned their finger, the status on the generic admin viewing screen updated almost instantly. The "Present" count increased without the teacher needing to refresh the page. This confirmed that our WebSocket/Real-time subscription logic was working correctly.

[Paste Screenshot of Attendance Sheet/Logs Here]

Figure 6.3: Attendance Data Populating in Real-Time

6.3 PERFORMANCE ANALYSIS

We didn't just check if it turned on; we measured how well it performed.

- **Speed:** The fingerprint sensor takes about 1 second to identify a print. The total time from "Finger placed" to "Database Updated" was observed to be around 2-3 seconds on average, which is much faster than calling out names.
- **Accuracy:** In our test group of 10 students, the sensor denied access to unregistered fingers 100% of the time. It successfully recognized registered fingers 95% of the time (failures were mostly due to sweaty or improperly placed fingers).
- **Battery/Power:** The device runs on standard USB 5V power. In future portable versions, a simple power bank could keep it running for a full day of classes.

6.4 APPLICATIONS

While we built Attendro for a college classroom, the core technology has many other uses.

6.4.1 Educational Institutions

This is the primary use case. Schools, colleges, and coaching centers can use it to automate daily roll calls, saving 10-15 minutes per lecture. It also provides parents with instant notifications if their child is absent.

6.4.2 Corporate Offices

Offices can use a similar system for employee clock-in and clock-out times. Since the device is portable, it can be used for on-site meetings or temporary work locations where installing a permanent biometric door lock isn't feasible.

6.4.3 Examination Halls

Identity verification during exams is critical. This system can be used to verify that the student sitting for the exam is indeed the one who registered, preventing impersonation malpractice.

6.4.4 Library Management

Libraries can use this to track student entry and exit logs without needing physical ID cards, which are often forgotten or lost.

6.5 CHAPTER SUMMARY

In this chapter, we showcased the successful operation of the Attendro system. The hardware works reliably, the software is responsive, and the data syncs perfectly. We also explored how this simple yet effective solution can be applied beyond just our college classrooms to offices and libraries. The final chapter will conclude our findings and look at the future of this project.

CHAPTER–7 CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The "Attendro" project started with a simple question: "How can we stop proxy attendance and make the whole process easier for teachers?" After developing, building, and testing our system, we believe we have found a solid answer. We successfully built a system that combines the security of fingerprints with the smart logic of college timetables.

Throughout this project, we learned that technology is most useful when it solves a specific problem. By moving the "brain" of the system to the ESP32 device in the classroom, we solved the issue of unreliable internet. By adding the Faculty App, we gave teachers back the control they had lost with automatic systems. The system correctly identifies students, checks their batch, and rejects anyone who shouldn't be there.

In conclusion, Attendro is not just a fingerprint scanner; it is a context-aware tool that respects the academic rules of the institute. It saves time, saves paper, and most importantly, it ensures integrity in student records. We are confident that this system is ready for real-world use in our college.

7.2 LIMITATIONS

While the system works well, we must be honest about its current limits. First, the fingerprint sensor (R307) sometimes struggles if a student's finger is wet or very dirty, requiring a second try. Second, the current device can only hold 120 students, which is fine for practical batches but might need an upgrade for very large seminar halls. Lastly, the initial enrollment process takes some time as every student needs to register their finger once.

7.3 FUTURE SCOPE

This project is just the beginning. There are several exciting ways we can improve it in the future:

7.3.1 Centralized Cloud Management

Right now, we use a single dashboard. In the future, we can expand this to a "Multi-College" platform where a university can track attendance across all its affiliated colleges from one screen.

7.3.2 AI-Based Predictions

We can add a machine learning layer to the data. The system could analyze patterns and predict which students are likely to drop out or fail based on their attendance trends, alerting teachers before it's too late.

7.3.3 Integration with Smart Campus

This device could be connected to other smart campus features. For example, if a student is marked present in the lab, the canteen or library system could know they are busy and not allow transactions there, ensuring students stay in class.

7.4 FINAL WORDS

Working on Attendro has been a great learning experience. It taught us how to integrate hardware with software and how to design for the end-user. We hope this project serves as a useful contribution to our department and helps modernize the way we look at attendance.

REFERENCES

[1] Espressif Systems, "ESP32 Series Datasheet," Version 3.7, 2023. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

[2] Supabase, "Database Documentation and Realtime API," 2024. [Online]. Available: https://supabase.com/docs

[3] Adafruit Industries, "Adafruit Optical Fingerprint Sensor (R307) Usage Guide," 2022. [Online]. Available: https://learn.adafruit.com/adafruit-optical-fingerprint-sensor

[4] React Community, "React Documentation," Meta Platforms, 2024. [Online]. Available: https://react.dev/

[5] S. Kumar and R. Singh, "Biometric Attendance Systems: A Review," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 5, pp. 112-115, 2017.

[6] M. Rouse, "Internet of Things (IoT)," TechTarget, 2019. [Online].

[7] MDN Web Docs, "WebSockets API," Mozilla, 2024. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API