



National Textile University
Department of Computer Science

Project:

Fingerprint Attendance System

Submitted to:

Sir Nasir Mehmood

Submitted by:

Arooj Komal (23-NTU-CS-1140)

Ayesha Khalid (23-NTU-CS-1248)

Faseeha Ahmad (23-NTU-CS-1251)

Subject:

Embedded IoT Systems

Section:

BSAI

Semester:

5th

Contents

IoT Based Fingerprint Attendance System	3
Using ESP32, AS608 Fingerprint Sensor, Blynk, Node-RED and Google Sheets	3
1. Executive Summary	3
2. Introduction	3
3. Project Objectives	3
4. System Overview.....	3
4.1 System Architecture.....	3
4.2 Hardware Components	4
4.3 Software Components	4
4.4 Blynk Widget Configuration	4
4.5 Working Principle.....	4
4.6 Data Flow	5
4.7 Hardware Pin Connections	5
4.8 Circuit diagram.....	6
4.8 Advantages.....	6
4.9 Applications.....	6
5. Phase 1: Implementation Using Blynk	7
Libraries used:	7
Features Implemented:	7
Limitations of Blynk:	7
6. Phase 2: Implementation Using Node-RED	7
Advantages of Node-RED:.....	7
Node-RED Functions:	7
7. System Working Principle	8
8. Google Sheets Data Structure.....	8
9.Flowchart	9
10.Screenshots.....	10
11.Results and Observations	12
12. Testing and Validation	13
13. Conclusion.....	13
14. Future Enhancements	13

IoT Based Fingerprint Attendance System

Using ESP32, AS608 Fingerprint Sensor, Blynk, Node-RED and Google Sheets

1. Executive Summary

This project presents an **IoT-based Fingerprint Attendance System** that was implemented in **two phases**. In the first phase, the system was developed using the **Blynk IoT platform** for dashboard control and monitoring. In the second phase, the system was enhanced and migrated to **Node-RED**, providing greater flexibility, control, and customization.

The system uses an **ESP32 microcontroller**, **AS608 fingerprint sensor**, and **OLED display** to authenticate users biometrically and automatically record attendance. Attendance data is stored in **Google Sheets** with real-time date and time fetched via the internet (NTP). The final solution is secure, automated, and suitable for educational institutions and organizations.

2. Introduction

Traditional attendance systems are manual, time-consuming, and prone to errors such as proxy attendance. Biometric authentication provides a reliable solution by uniquely identifying each individual.

This project aims to design and implement an **IoT-enabled fingerprint attendance system** that records attendance automatically and stores it in the cloud. The project was first implemented using **Blynk** and later improved using **Node-RED** to overcome limitations and add advanced control.

3. Project Objectives

- To design a fingerprint-based attendance system
- To implement automatic enrollment of new users
- To authenticate users using biometric fingerprints
- To provide real-time feedback using OLED display
- To store attendance records automatically in Google Sheets
- To develop IoT dashboards using Blynk and Node-RED
- To prevent duplicate attendance entries on the same day

4. System Overview

4.1 System Architecture

The system consists of four main layers:

1. **Hardware Layer** – ESP32, AS608, OLED
2. **Communication Layer** – WiFi
3. **Cloud Layer** – Blynk & Google Sheets
4. **User Interface Layer** – Blynk Dashboard

4.2 Hardware Components

ESP32

Controls the system, manages WiFi, processes fingerprints, and sends data to the cloud.

AS608 Fingerprint Sensor

Used for fingerprint enrollment and authentication. Each fingerprint is stored with a unique ID.

OLED Display

Shows system status such as booting, WiFi connection, enrollment instructions, and attendance confirmation.

4.3 Software Components

- **Blynk Platform:** Dashboard control, status display, and user input
- **Google Sheets:** Cloud database for attendance records
- **Arduino Framework:** Used to program ESP32

4.4 Blynk Widget Configuration

Widget	Virtual Pin	Purpose
Enroll Button	V0	Start enrollment
Attendance Button	V1	Mark attendance
Status Label	V2	Show system status
Name Label	V3	Display user name
Name Input	V4	Enter name for enrollment

4.5 Working Principle

1. System powers ON
2. ESP32 connects to WiFi
3. User enters name in Blynk
4. User presses Enroll
5. Fingerprint is scanned and stored
6. User presses Attendance
7. Fingerprint is verified
8. Attendance is uploaded to Google Sheets

9. OLED shows confirmation

4.6 Data Flow

Fingerprint → ESP32 → WiFi → Blynk Cloud → Google Sheets → Dashboard

4.7 Hardware Pin Connections

AS608 Fingerprint Sensor → ESP32

AS608 Pin	ESP32 Pin
VCC	5V
GND	GND
TX	GPIO 16
RX	GPIO 17

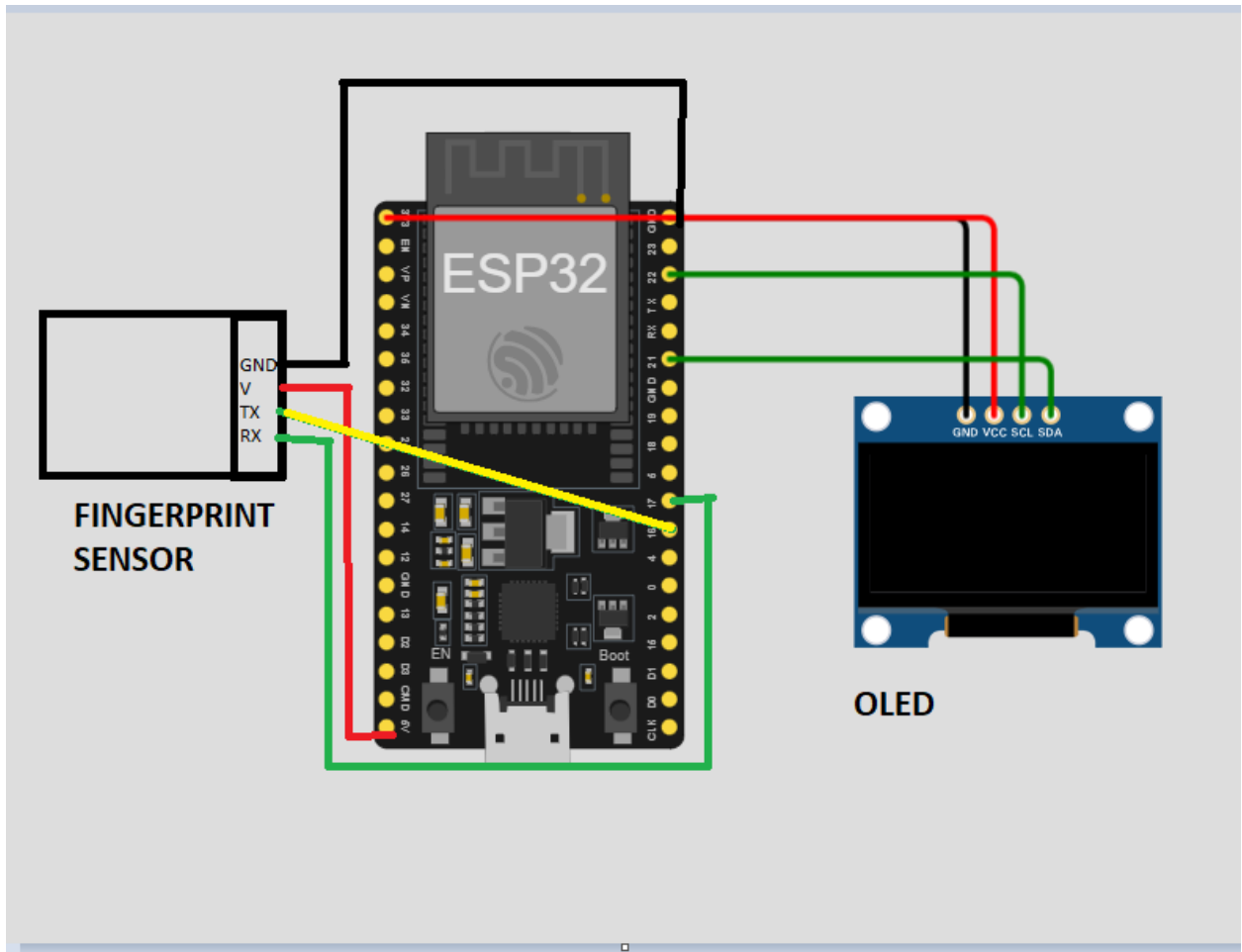
OLED Display (I2C) → ESP32

OLED Pin	ESP32 Pin
VCC	3.3V
GND	GND
SDA	GPIO 21
SCL	GPIO 22

Power Summary

- ESP32: USB 5V
- AS608: 5V
- OLED: 3.3V

4.8 Circuit diagram



4.8 Advantages

- Automatic attendance
- Real-time monitoring
- Cloud storage
- Easy to use dashboard
- Reduced manual errors

4.9 Applications

- Schools
- Colleges
- Offices
- Training centers

5. Phase 1: Implementation Using Blynk

In the first phase, the project was implemented using the **Blynk IoT platform**.

Libraries used:

- adafruit/Adafruit Fingerprint Sensor Library
- adafruit/Adafruit SSD1306
- adafruit/Adafruit GFX Library
- blynkkk/Blynk

Features Implemented:

- Enroll button to register new users
- Text input widget for user name
- Attendance button to mark attendance
- OLED feedback messages
- Google Sheets integration

Limitations of Blynk:

- Limited widget customization
- Dependency on mobile application
- Less flexibility in logic handling
- Difficulty in handling complex workflows

Due to these limitations, the project was further enhanced using Node-RED.

6. Phase 2: Implementation Using Node-RED

In the second phase, **Node-RED** was used to replace Blynk as the dashboard and control interface.

Advantages of Node-RED:

- Fully customizable web dashboard
- Better control over logic and data flow
- Easier integration with Google Sheets
- Real-time visualization and debugging

Node-RED Functions:

- Trigger enrollment and attendance
- Display system status
- Prevent duplicate attendance
- Send data to Google Sheets

7. System Working Principle

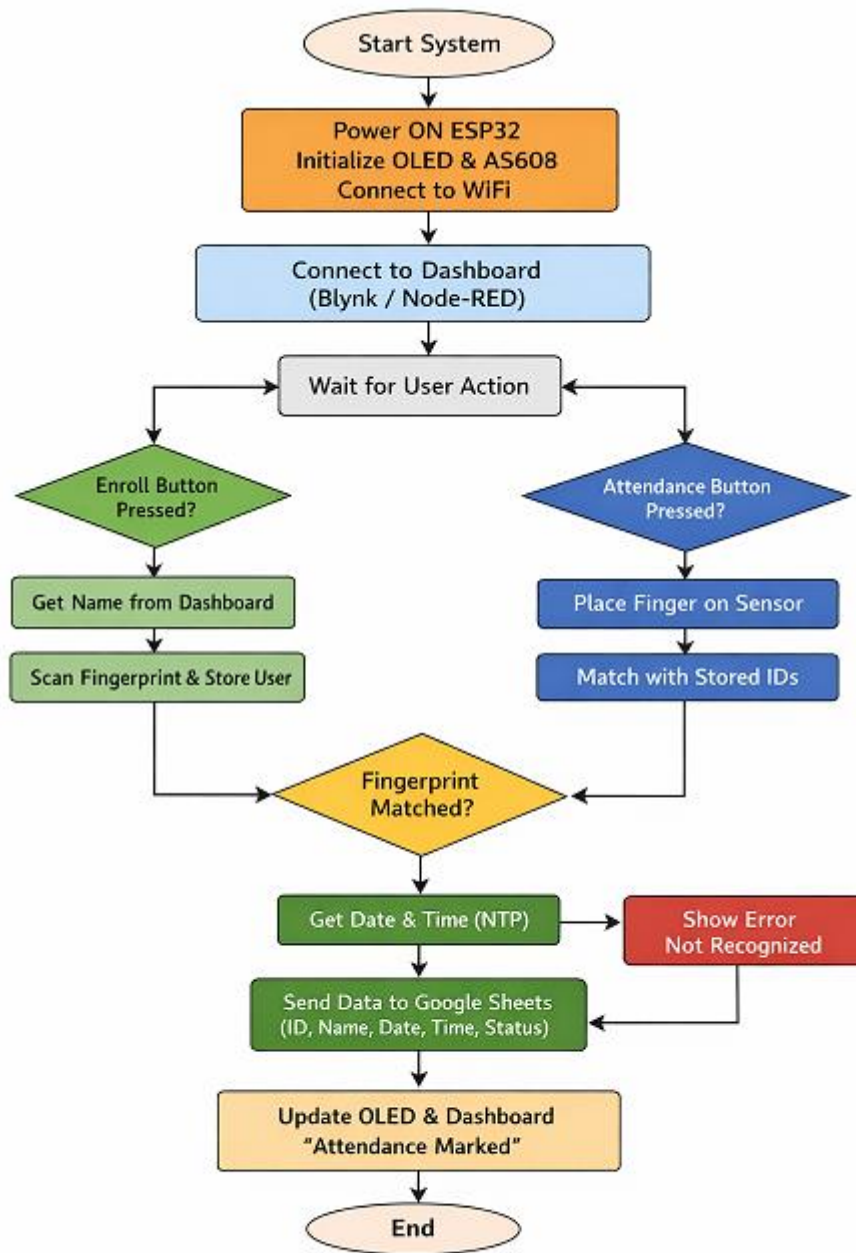
1. ESP32 connects to Wi-Fi
2. Fingerprint sensor and OLED are initialized
3. User enrolls fingerprint (two scans)
4. Fingerprint ID is mapped to user name
5. User scans finger for attendance
6. ESP32 verifies fingerprint
7. Date and time fetched using NTP
8. Attendance sent to Google Sheets
9. Duplicate attendance is prevented

8. Google Sheets Data Structure

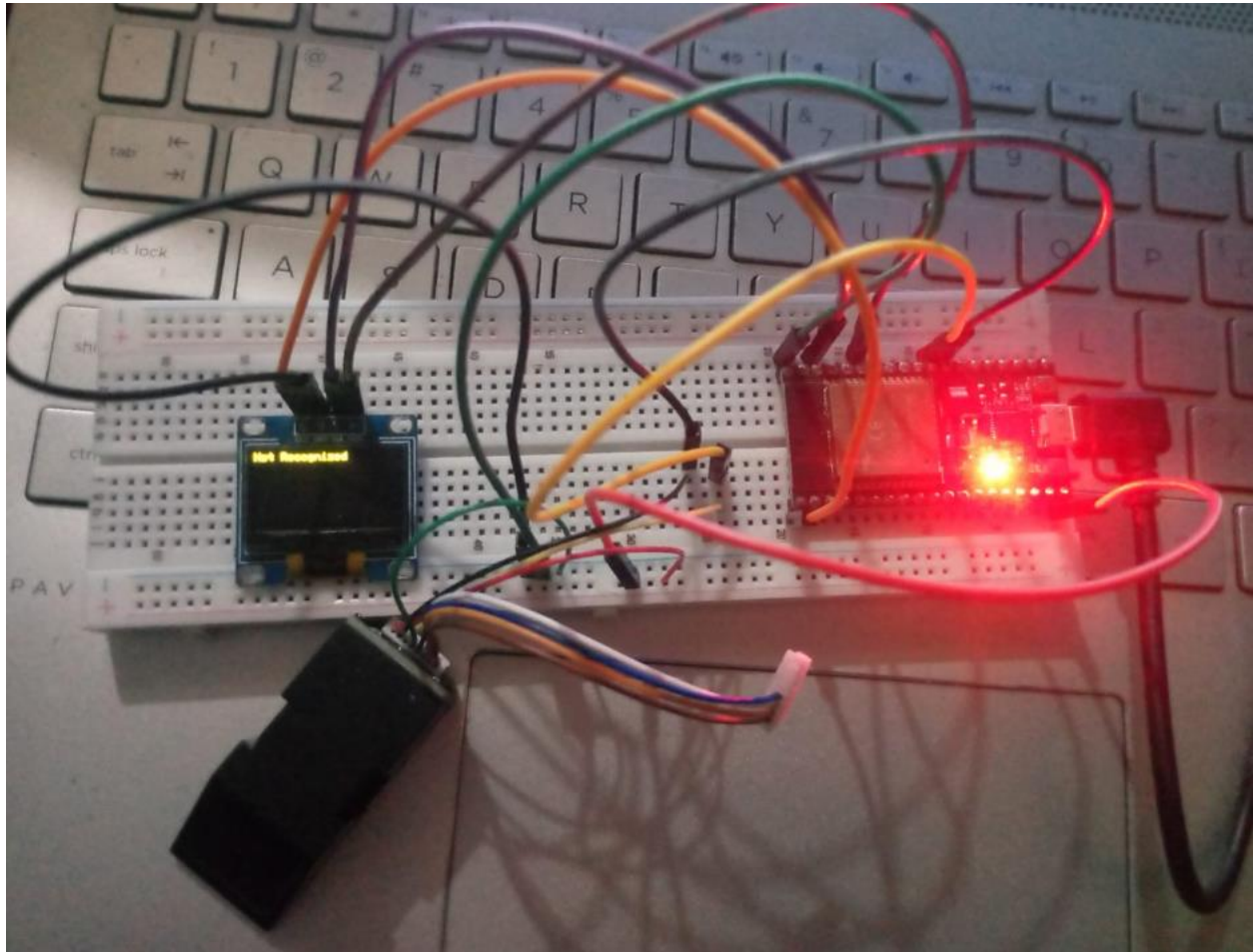
Finger_ID	Name	Date	Time	Status
1	Ali	07-01-2026	09:15	Present

Google Apps Script updates existing records if attendance is already marked for the day.

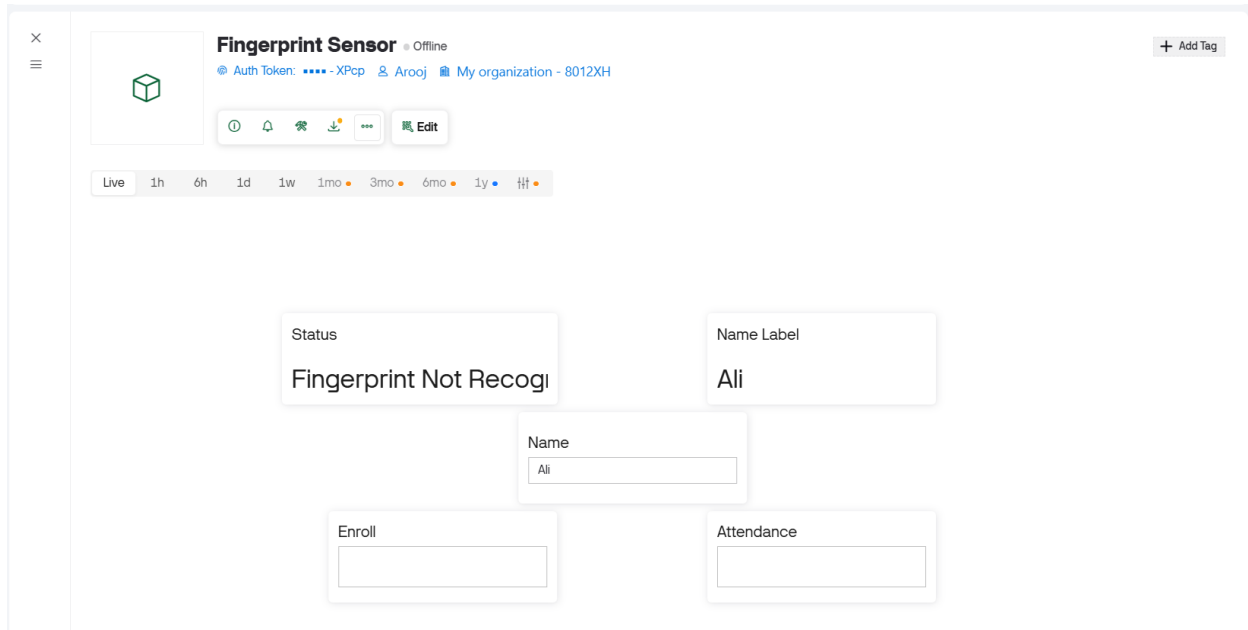
9.Flowchart



10.Screenshots







Fingerprint_Attendance ☆ 📁 ☁

File Edit View Insert Format Data Tools Extensions Help

100% | £ % .0 .00 123 | Defaul... | - 10 + | B I ⚡ A | 🎨 📐 📏 📏

	A	B	C	D	E	F	G	H	I
1	Finger_ID	Name	Date	Time	Status				
2	1	Rubeet	07-01-2026	01:42	Present				
3	2	Arooj	07-01-2026	01:43	Present				
4	3	Fareeha	07-01-2026	01:45	Present				
5	4	Rohan	07-01-2026	01:47	Present				
6	2	Ayesha	07-01-2026	13:34	Present				
7	5	Faseeha	07-01-2026	13:38	Present				
8	6	Alishba	07-01-2026	13:41	Present				
9	7	Ali	09-01-2026	04:15	Present				
10									
11									
12									

11.Results and Observations

- Accurate fingerprint authentication
- Real-time attendance logging
- Correct date and time from NTP server
- No duplicate attendance entries
- Smooth dashboard operation in Node-RED
- OLED provides clear user feedback

12. Testing and Validation

Test Case	Result
Valid fingerprint	Attendance marked
Same user twice	Duplicate prevented
Invalid fingerprint	Access denied
Wi-Fi disconnection	Auto reconnection
Power reset	System resumes

13. Conclusion

The IoT-based Fingerprint Attendance System successfully automates attendance management using biometric authentication and cloud storage. Implementing the project in two phases—first with Blynk and then with Node-RED—helped overcome platform limitations and improved system reliability and flexibility.

14. Future Enhancements

- Face recognition integration
- Offline attendance storage
- SMS / Email notifications
- Admin web portal
- Mobile application