

Biometric Fingerprint Attendance System

by

ANDREW JOHN	18BEC1278
N.T. SRIHARI	18BEC1180
S. ABINESH VEL	18BEC1340
VIGHNESH M	18BEC1223

A project report submitted to

Dr Prakash V

SCHOOL OF ELECTRONICS ENGINEERING

in partial fulfilment of the requirements for the course of

ECE4003 – EMBEDDED SYSTEM DESIGN

in

B. Tech. ELECTRONICS AND COMMUNICATION ENGINEERING



VIT[®]

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Vandalur – Kelambakkam Road

Chennai – 600127

JUNE 2020

BONAFIDE CERTIFICATE

Certified that this project report entitled “**BIOMETRIC FINGERPRINT ATTENDANCE SYSTEM**” is a bonafide work of **ANDREW JOHN - 18BEC1278, N.T. SRIHARI - 18BEC1180, S. ABINESH VEL - 18BEC1340, and VIGHNESH M - 18BEC1223** who carried out the Project work under my supervision and guidance for **ECE4003 - Embedded System Design**.

Dr. Prakash V

Associate Professor

School of Electronics Engineering (SENSE),

VIT University, Chennai

Chennai – 600 127.

ABSTRACT

The main objective of the project is to create a mobile Biometric Attendance system that will help reduce the hassle of marking Attendance and save precious time. Moreover, this system will ensure that the “ proxy “ culture will reduce significantly because each individual has a unique fingerprint and it is difficult to forge it. Drawing inspiration from this concept we have developed a prototype of a Biometric Attendance System that registers the students with their respective fingerprints and uses the same to mark the attendance for them and records the in-time and out-time and every other necessary information required.

This will help in acknowledging the students present inside the campus. Therefore eliminating the risk of losing track of a particular student/person in case of an emergency, this would prove to be very useful in Elementary and Primary schools where a lot of young children are present. So we have tried to incorporate all these into our prototype and create an easy to use mobile Attendance System.

ACKNOWLEDGEMENT

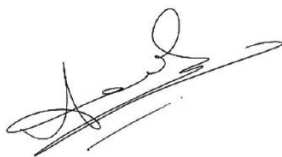
We wish to express our sincere thanks and deep sense of gratitude to our project guide, **Dr. Prakash V**, Associate Professor, School of Electronics Engineering, for her consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We are extremely grateful to **Dr. Sivasubramanian. A**, Dean of School of Electronics Engineering, VIT Chennai, for extending the facilities of the School towards our project and for his unstinting support.

We express our thanks to our Head of the Department **Dr. Vetrivelan. P** for his support throughout the course of this project.

We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the course.

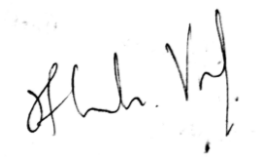
We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.



ANDREW JOHN



N T SRIHARI



ABINESH VEL



VIGHNESH M

TABLE OF CONTENTS

S. No			TITLE	P No.
			ABSTRACT	3
			ACKNOWLEDGMENT	4
1			INTRODUCTION	7
	<i>1.1</i>		<i>OBJECTIVES AND GOALS</i>	7
	<i>1.2</i>		<i>APPLICATIONS</i>	7
	<i>1.3</i>		<i>FEATURES</i>	8
2			DESIGN & IMPLEMENTATION	9
	<i>2.1</i>		<i>COMPONENT & TECHNICAL SPECIFICATION</i>	9
	<i>2.2</i>		<i>TOOLS REQUIRED</i>	10
	<i>2.2</i>		<i>OVERALL ARCHITECTURE</i>	15
	<i>2.3</i>		<i>HARDWARE ANALYSIS</i>	15
		<i>2.3.1</i>	BLOCK DIAGRAM	15
		<i>2.3.2</i>	CIRCUIT DIAGRAM OF HARDWARE	16
		<i>2.3.3</i>	INTEGRATION OF HARDWARE	17
	<i>2.4</i>		<i>SOFTWARE ANALYSIS</i>	18
		<i>2.4.1</i>	BLOCK DIAGRAM	18
		<i>2.4.2</i>	FLOWCHART OF SOFTWARE	19
		<i>2.4.3</i>	INTEGRATION OF SOFTWARE	20
3			CODING & IMPLEMENTATION	21
	<i>3.1</i>		<i>FUNCTIONS & LIBRARIES USED</i>	21
	<i>3.1</i>		<i>ARDUINO CODE</i>	22
	<i>3.2</i>		<i>HTML & PHP CODE</i>	40
4			RESULT & ANALYSIS	54

	4.1		<i>SCHEMATIC & PCB OF BFAS</i>	54 - 55
	4.2		<i>REAL-TIME IMPLEMENTATION OF BFAS</i>	56
	4.3		<i>SNAPSHOTS OF DESIGNED WEBSITE</i>	58
5			CONCLUSION	60
	5.1		CONCLUSION	60
	5.2		FUTURE SCOPE	60
	5.3		COST ANALYSIS	61
6			REFERENCES	62
7			BIODATA	63

**BFAS - Biometric Fingerprint Attendance System*

CHAPTER I

INTRODUCTION

1.1 OBJECTIVES AND GOALS

In this project, we will be building a biometric attendance system using fingerprint sensors, NodeMCU and also will be creating a website that'll host the database containing all the records. We will get inputs from the user which will be then passed to the NodeMCU which will check for the fingerprint from the database and make a move accordingly. Through this we will be able to store all the information with just a single fingerprint entry.

The main objective is to create a safe and secure way to keep a record that cannot be forged or misused by using fingerprints and storing all these in a secure database which can later be accessed for any validation purposes. With this system, we are able to record the name, in-time, out-time and date. Moreover, another important aspect is making sure that this system is mobile, akin which can be moved around without any hassle, therefore we can use it around a campus at various times.

1.2 APPLICATIONS

There are numerous applications for the use of Biometric Technology, but the most common ones are given below:

- **Logical Access Control:** This market application refers to gaining access to a computer network either at the place of the business or corporation or via a secured remote connection from a distant location.
- **Physical Access Control:** Physical Access Entry refers to giving an employee of a business or a corporation access to a secure building, or even a secure office from within it.
- **Time and Attendance:** The use of Biometric Technology can play an integral role in Time and Attendance based applications, by combatting the weaknesses such as one employee fraudulently reports the time

worked for another employee when they did not show up for their required work shift, and he or she still gets paid for it.

- **Law Enforcement:** Law enforcement agencies across all levels of the Federal Government are also starting to use Biometric Technology to confirm the identity of any suspects or wanted felons. It has been traditionally Fingerprint Recognition which is the most widely used modality.
- **Surveillance:** Surveillance is simply keeping tabs of a large group of people, and from there, determining any abnormal behavior from an established baseline. In this instance, it is Facial Recognition which is used the most than Biometric Fingerprint.

1.3 FEATURES

- Integrated image collecting and algorithm chip together, ALL-in-One, in a compact setup.
- The fingerprint reader can conduct secondary development, can be embedded into a variety of end products and here we have connected it to the NodeMCU to make it work as per our requirement.
- Low power consumption, low cost, small size, excellent performance.
- Professional optical technology, precise module manufacturing techniques.
- Good image processing capabilities can successfully capture images up to resolution 500 dpi, therefore making sure that the fingerprint captured is proper and doesn't not contain any irregularities.
- In the website we have added real time data integration, that is, once the fingerprint has been captured, it'll automatically update the database and the results will be displayed on the website within seconds, thereby making sure that all the systems are in sync with the same latency.

CHAPTER II

DESIGN & IMPLEMENTATION

2.1 COMPONENT & TECHNICAL SPECIFICATION

Fingerprint Sensor is used to integrate with the arduino and pass on the data to the server through SQL. SQL database is used for storing the data to the website

The Arduino(Node-MCU) is programmed in a way that it checks with the browser for existing data and then proceeds to make a new entry to the website. For incorporating the database into the website , we are using the XAMPP framework along with PHP language.

An OLED Display is used to show whether that fingerprint has been registered (already existing) or not registered. A Li-ION battery is used making it portable (18V, 2500mAh), so that it can move around as per convenience. A Battery Charger (Module - TP4056) - for charging the battery so that it can be reused accordingly , thus reducing the total cost. MT3608 Boost Converter - It is used to step up the input voltage to a higher value as per the load requirement.

A Switch is used to trigger the ON - OFF operations after being verified by the system. XAMPP is also used to make sure that all the interactions are user friendly.

Table - 1: Technical Specifications of R305.

Power	DC 4.2V-6V	Interface	UART(TTL logical level)/ USB 2.0
Working current	Typical: 50mA	Matching Mode	1:1 and 1:N
Baud rate	(9600*N)bps, N=1 ~ 12 (default N=6)	Character file size	256 bytes
Image acquiring time	<0.5s	Template size	512 bytes
Storage capacity	1000	Security level	5 (1, 2, 3, 4, 5(highest))
FAR	<0.001%	FRR	<0.1%
Average searching time	< 1s (1:1000)	Window dimension	19mm*21mm
Working environment	Temp: -10°C- +40°C	Storage environment	Temp: -40°C- +85°C
	RH: 20%-85%		RH: <85%
Outline Dimention	Split type	Module: 44.1*20*23.5 mm	

2.2 TOOLS REQUIRED

Hardware Required:

- NodeMCU
- R305 Fingerprint Sensor
- 0.96" OLED Display
- Li-Ion Battery (18650)
- TP4056 Li-Ion Battery Charger
- MT3608 Boost Converter
- On/Off switch
- Breadboard
- Connecting Wires.

Software Required:

- XAMPP
- PHP
- HTML
- Arduino IDE

a. NodeMCU ESP8266

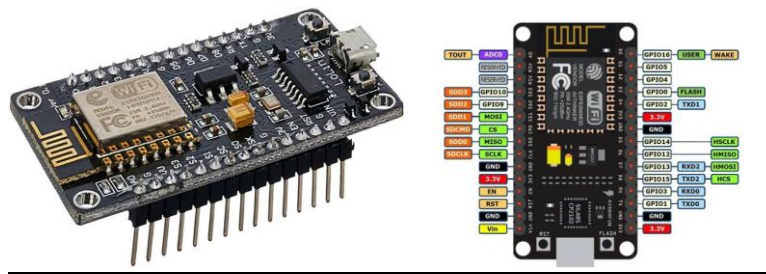


Fig - 1: NodeMCU and it's pinout

NodeMCU is an open-source Lua based firmware and development board specially targeted for IoT based Applications. It includes firmware that runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module.

NodeMCU ESP8266 Specifications & Features

- Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
- Operating Voltage: 3.3V
- Input Voltage: 7-12V
- Digital I/O Pins (DIO): 16
- Analog Input Pins (ADC): 1
- UARTs: 1
- SPIs: 1
- I2Cs: 1

- Flash Memory: 4 MB
- SRAM: 64 KB
- Clock Speed: 80 MHz
- USB-TTL based on CP2102 is included onboard, Enabling Plug n Play
- PCB Antenna

b. R305 Biometric Fingerprint Sensor

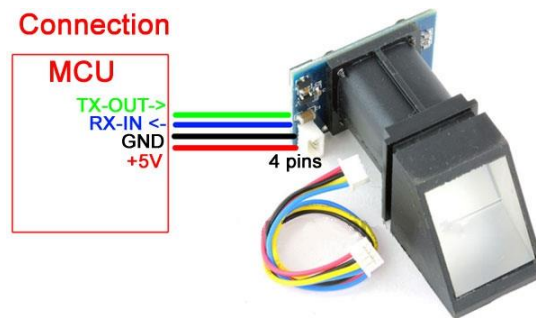


Fig - 2: R305 Biometric Fingerprint Sensor it's pinout

Fingerprint Sensor (R305) -TTL UART is a fingerprint sensor module with TTL UART interface. The user can store the fingerprint data in the module and can configure it in 1:1 or 1: N mode for identifying the person. The fingerprint module can directly interface with a 3v3 or 5v Microcontroller. A level converter (like MAX232) is required for interfacing with the PC.

R305 Biometric Fingerprint Sensor Specifications & Features

- Power DC : 3.6V-6.0V
- Interface : UART (TTL logic level)/ USB 1.1
- Working current : 100mA
- Peak Current : 150mA
- Matching Mode: 1:1 and 1:N
- Baud rate (9600*N)bps, N=1-12 (default N=6 57600bps)
- Character file size: 256 bytes
- Image acquiring time : <0.5s
- Template size : 512 bytes
- Storage capacity: 256
- Security level : 5 (1, 2, 3, 4, 5(highest))
- FAR : <0.001%
- FRR: <0.1%
- Average searching time: < 0.8s (1:880)
- Window dimension : 18mm*22mm

c. 0.96" OLED Display

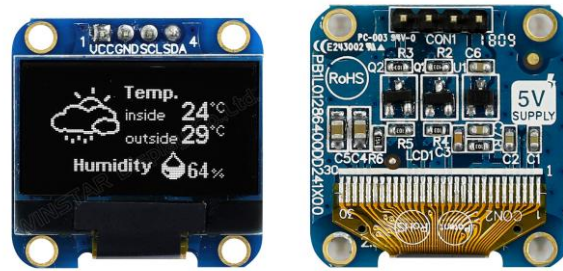


Fig - 3: 0.96" OLED Display

WEA012864D-03 version 4 pin OLED display module is a diagonal size 0.96 inch COG OLED display with PCB board on it. This WEA012864D-03 version OLED display is made of 128x64 pixels. WEA012864D-03 module is built in with SSD1306BZ IC, it communicates via I2C interface only, VCC 3V /5V, I/O level 5V to 3V, with conversion circuit, 1/64 duty cycle. The WEA012864D-03 model has a smaller PCB than the WEA012864D-01 version of which outline size is 27.3x27.3x2.37mm with mounting holes on board and 4 metal pins on module.

0.96" OLED Display Specifications & Features

- OLED Driver IC: SSD1306
- Resolution: 128 x 64
- Visual Angle: >160°
- Display Color: Area Color (White)
- Input Voltage: 3.3V ~ 6V
- Compatible I/O Level: 3.3V, 5V
- Only Need 2 I/O Port to Control
- Full Compatible with Arduino
- Working temperature: -30°C ~ 70°C
- Interface: I2C
- Support a wide range of voltage input

d. Li-Ion Battery (18650)

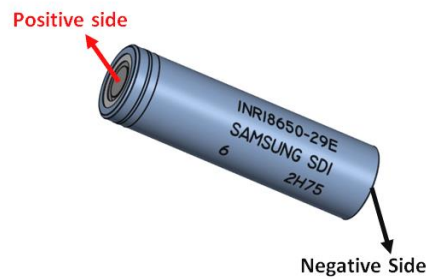


Fig - 4: Li-Ion Battery (18650)

Li-Ion Battery (18650) Specifications & Features

- Nominal Voltage: 3.6V
- Nominal Capacity: 2,850 mAh
- Minimum Discharge Voltage: 3V
- Maximum Discharge current: 1C
- Charging Voltage: 4.2V (maximum)
- Charging current: 0.5C
- Charging Time: 3 hours (approx)
- Charging Method: CC and CV
- Cell Weight: 48g (approx)
- Cell Dimension: 18.4mm (dia) and 65mm (height)

e. TP4056 Li-Ion Battery Charger

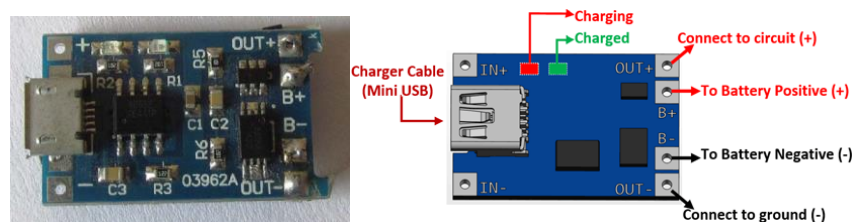


Fig - 5: TP4056 Li-Ion Battery Charger it's pinout

TP4056 Li-Ion Battery Charger Specifications & Features

- This module can charge and discharge Lithium batteries safely
- Suitable for 18650 cells and other 3.7V batteries
- Charging current – 1A (adjustable)
- Input Voltage: 4.5V to 5.5V
- Full charge voltage 4.2V

- Protects battery from overcharging and over discharging
- No reverse polarity protection.

f. MT3608 Boost Converter

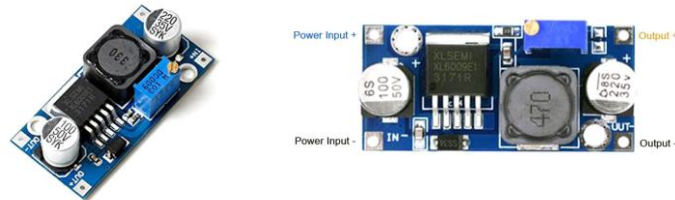


Fig - 6: MT3608 Boost Converter and it's pinout

This DC-DC switching boost converter is capable of driving a 4A load with excellent line and load regulation. The main switching component XL6009 IC is available in fixed output voltages of 3.3 V, 5V, 12V, and an adjustable output version. It is an efficient switching regulator and the output efficiency is significantly higher in comparison with the popular boost regulators. At higher input voltages, the regulator operates at a switching frequency of 400kHz thus allowing the overall board size to be smaller and space-saving.

MT3608 Boost Converter Specifications & Features

- Input voltage: 3 - 32V
- Output voltage: 5 - 35V (adjustable)
- Output current: Maximum output current 4A
- Note: The higher the voltage, the load current increases.
- Efficiency of this regulator upto <94%
- Load regulation: 0.5%
- Voltage regulation: 0.5%
- Adjustable potentiometer onboard for output voltage adjustment.
- Non - isolated constant voltage module.
- Non - synchronous rectification.
- Short circuit proportion: current limiting since the recovery.
- Dimension: 45*20*14 mm (L*W*H).

2.3 OVERALL ARCHITECTURE

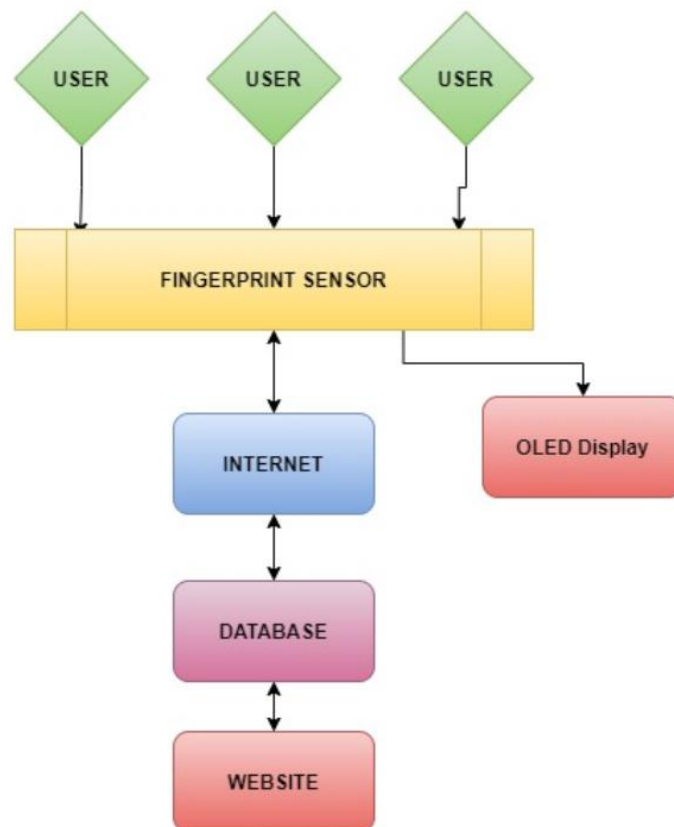


Fig - 7: Overall Architecture of Biometric Fingerprint Attendance System

2.4 HARDWARE ANALYSIS

2.4.1 BLOCK DIAGRAM

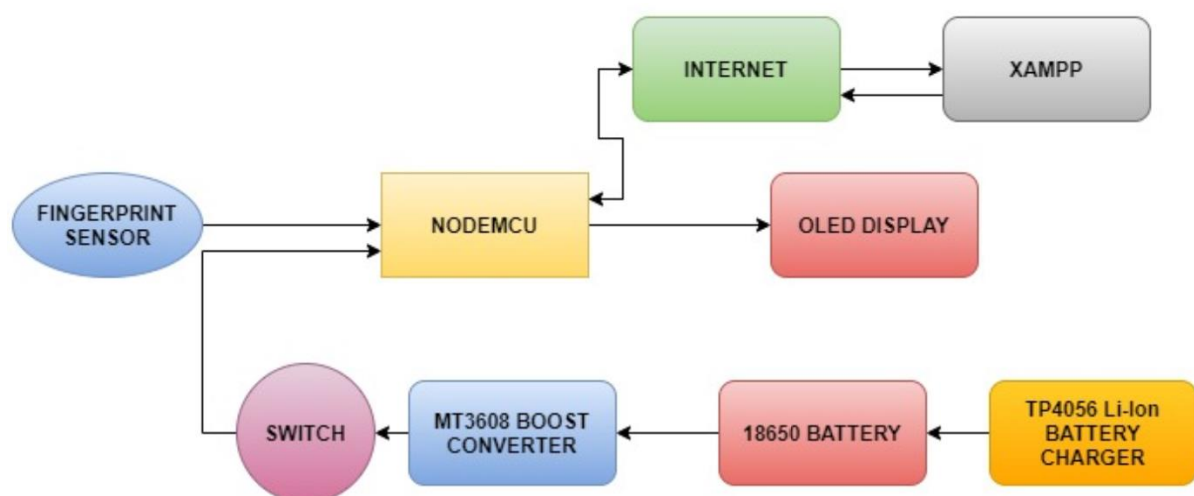


Fig - 8: Hardware Design of Biometric Fingerprint Attendance System

The user presses their fingerprint against the fingerprint sensor. The fingerprint is sent to the NodeMCU.

First, it will check for the fingerprint whether it is correctly pressed or not. If it's correctly pressed, it proceeds to the next step. If not, it displays an error and it asks the user to input the fingerprint again. Once the fingerprint is entered properly, NodeMCU connects to the XAMPP database and the fingerprint entered will be compared with the already existing fingerprint templates for verification and once the fingerprint is verified, it will be sent to the backend for further processes.

The NodeMCU is also connected to an OLED display, which displays a welcome message before entering the fingerprint and also it displays the verification of the fingerprint once it is verified.

Concerning the Power Module, the NodeMCU is powered by a 18650 battery, which is connected to a TP4056 Li-Ion battery charger. The power from the battery is rised using a MT3608 Boost converter. The power module is connected to the system using a switch in between them.

2.4.2 CIRCUIT DIAGRAM OF HARDWARE

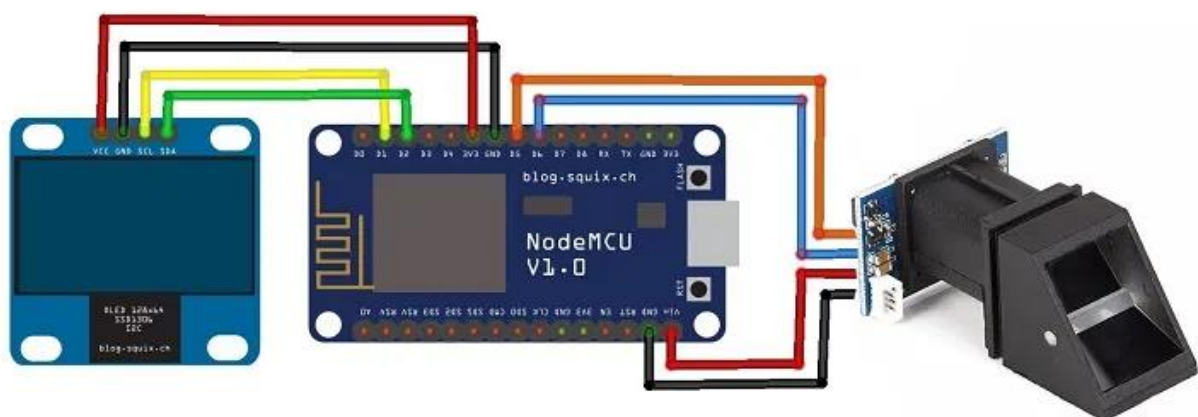


Fig - 9: Prototype Implementation of the Biometric Fingerprint Attendance System

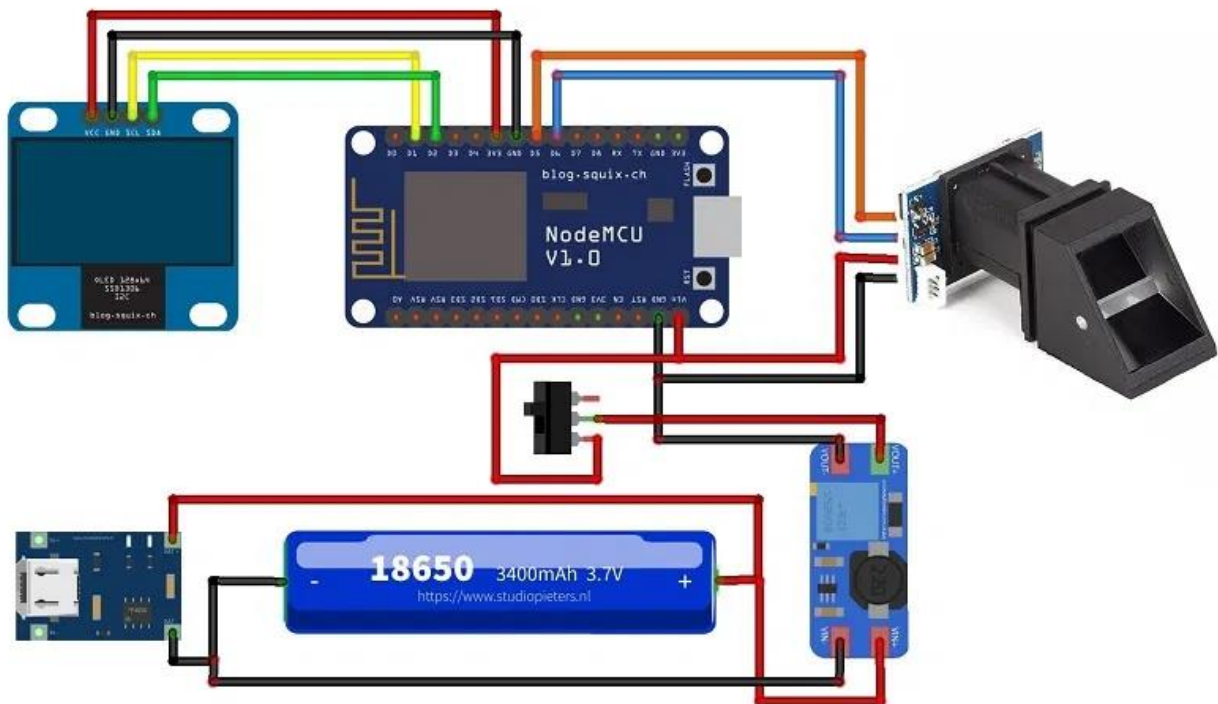


Fig - 10: Real Time Implementation of the Biometric Fingerprint Attendance System

2.4.3 INTEGRATION OF HARDWARE (SYSTEM INTEGRATION)

- From the OLED Display, Connect the Vcc, GND, SCL, SDA pins to NodeMCU 's 3V3, GND, D1, D2 digital pins, respectively.
- For the Fingerprint Sensor, Connect GND, 5V Vin, Tx, Rx to NodeMCU 's GND, Vin(also in vin switch), D5, D6, respectively
- Concerning the Switch, Connect the Vin of the Switch to NodeMCU 's Vin and Connect the Acc of Switch to MT3608 's Vout+, respectively.
- For MT3608 Boost Converter, Connect the Vout- to GND in NodeMCU, Vin- to Battery's to -Ve Terminal of Battery & in BAT- of TP4065, Vin+ to +Ve Terminal of Battery & BAT+ of TP4065, respectively.

2.5 SOFTWARE ANALYSIS

2.5.1 BLOCK DIAGRAM

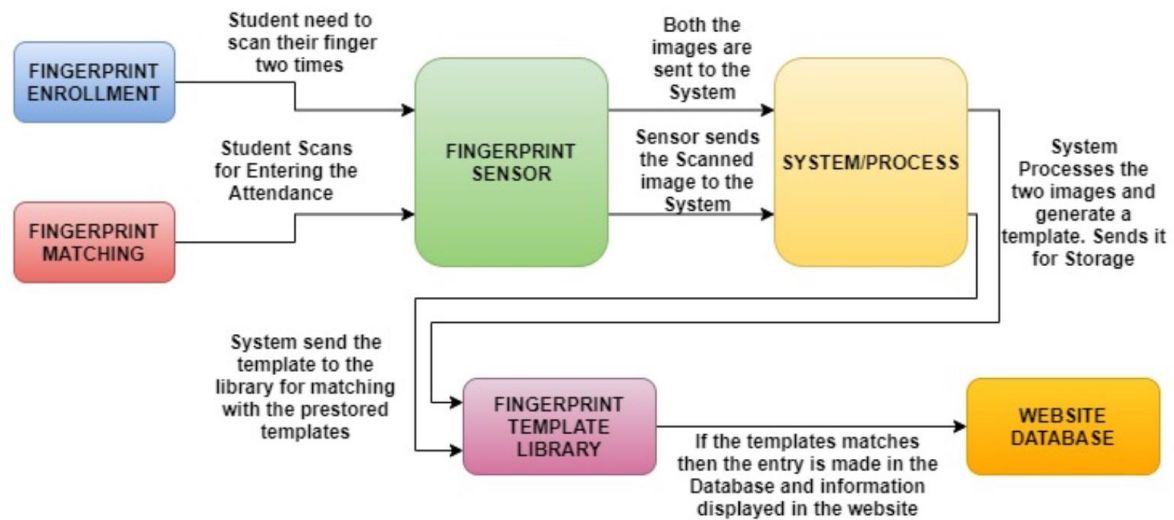


Fig - 11: Software Design of Biometric Fingerprint Attendance System

At first, the connection is made using the dbconnection module in the backend which connects the database to the website which later divides into various processes that we can use directly on the website as well. Then once the connection is made, the fingerprint received at the backend through the NodeMCU is then compared with the already existing information as to whether the particular fingerprint is already present or not in the database

If present, it moves onto the next process of storing the information such as in-time, out-time into the database. If not present, it then creates a new entry for the user and adds them to the user log database before continuing to the next process of storing the information about in-time and stuff as such.

Once all the verification is done, the getdata node is called and it collects and transfers all the information from the NodeMCU has received through the fingerprint and stores it under the respective user with the help of the manage_users module. In the website, we can directly add a new user using the manage_users module and look for all existing users present in the database in the users page. In the website we have access to a variety of tasks such as checking for a particular record by entering the date that we require etc

Moreover, all the data collected through the NODEMCU is passed onto the database and is stored in the form of an excel sheet. Therefore if necessary we can extract it in that format of any particular given date.

2.5.2 FLOWCHART OF SOFTWARE

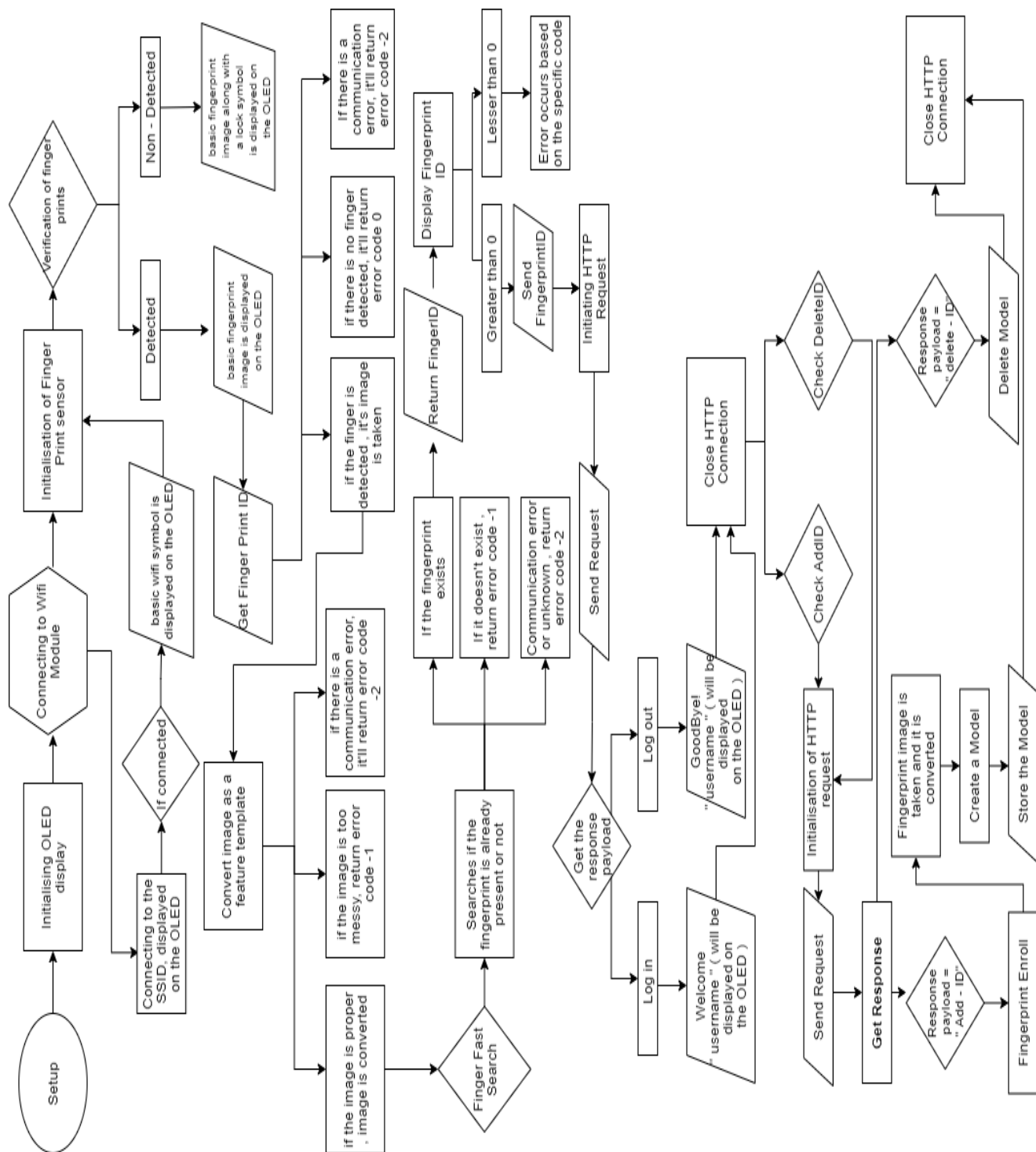


Fig - 12: Flow chart of Biometric Fingerprint Attendance System

2.5.3 INTEGRATION OF SOFTWARE (SYSTEM INTEGRATION)

- First the website will ask the user whether he/she is a new user or whether we need to manage users based on the need of the technical specialist.
- Next If it's a new user, then with the help of Fingerprint Scanner, we will scan the fingerprint (need to give fingerprint input twice) and input the New User's Credentials.
- For Recording, if the fingerprint is given once, then the website takes it as an Entry and records the in-time and if the same fingerprint is given once again later that day, the website takes it as an Exit and records the out-time.

CHAPTER III

CODING & IMPLEMENTATION

3.1 LIBRARIES AND FUNCTION USED

FUNCTIONS:

1. *DisplayFingerprintID()* - Displays the successful input of fingerprint of the user on the OLED Display.
2. *SendFingerprintID()* - Sends fingerprint from NodeMCU to the PHP where it is connected to the backend database created by MySQL
3. *getFingerprintID()* - Gets Fingerprint from the user on the R305 Fingerprint Sensor.
4. *ChecktoDeleteID()* - Searches for the fingerprint on the website for deleting the ID Template of any user.
5. *deleteFingerprint()* - Deletes the fingerprint of the person after searching it on the website.
6. *ChecktoAddID()* - Searches for the fingerprint on the website for adding the ID Template of any user.
7. *getFingerprintEnroll()* - Enrols the fingerprint of the person after getting it from the NodeMCU on the website.
8. *confirmAdding()* - Confirmation asking the admin for adding the new user and assigning the credentials to that particular user.
9. *connectToWiFi()* - Connects the NodeMCU to the WiFi

LIBRARIES:

1. *<SPI.h>* - Serial Peripheral Interface (SPI) is a synchronous serial data protocol used by microcontrollers for communicating with one or more peripheral devices quickly over short distances. It can also be used for communication between two microcontrollers. With an SPI connection there is always one master device (usually a microcontroller) which controls the peripheral devices.
2. *<Wire.h>* - Wire library allows you to communicate with I2C / TWI devices. On the Arduino boards with the R3 layout (1.0 pinout), the SDA (data line) and SCL (clock line) are on the pin headers close to the AREF pin.
3. *<ESP8266WiFi.h>* - Connects ESP8266 module to a Wi-Fi network to start sending and receiving data.

4. <SoftwareSerial.h> - The Arduino hardware has built-in support for serial communication on pins 0 and 1 (which also goes to the computer via the USB connection). The native serial support happens via a piece of hardware (built into the chip) called a UART. This hardware allows the Atmega chip to receive serial communication even while working on other tasks, as long as there room in the 64 byte serial buffer.
5. <ESP8266WebServer.h> - Web-server module. Supports only one simultaneous client, knows how to handle GET and POST.
6. <ESP8266HTTPClient.h> - Library which provides the methods to send HTTP requests.
7. <Adafruit_GFX.h> - The Adafruit_GFX library for Arduino provides a common syntax and set of graphics functions for all of our LCD and OLED displays. This allows Arduino sketches to easily be adapted between display types with minimal efforts.
8. <Adafruit_SSD1306.h> - This is a library for our Monochrome OLEDs based on SSD1306 drivers.
9. <Adafruit_Fingerprint.h> - Arduino library for interfacing to the fingerprint sensor in the Adafruit shop.

3.1 ARDUINO CODE

```

/*****libraries*****/
#include <SPI.h>
#include <Wire.h>
#include <ESP8266WiFi.h>
#include <SoftwareSerial.h>
#include <ESP8266WebServer.h>
#include <ESP8266HTTPClient.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <Adafruit_Fingerprint.h>
/*****
//Fingerprint scanner Pins
#define Finger_Rx 3 //D5
#define Finger_Tx 1 //D6
// Declaration for SSD1306 display connected using software I2C
#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels
#define OLED_RESET 0 // Reset pin # (or -1 if sharing Arduino reset pin)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
/*****
SoftwareSerial mySerial(Finger_Rx, Finger_Tx);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
/*****
/* Set these to your desired credentials. */
const char *ssid = "Andrew"; //ENTER YOUR WIFI SETTINGS
const char *password = "*****";
/*****

```

```

String postData ; // post array that will be send to the website
String link = "http://192.168.1.5/esd4003/getdata.php"; //computer IP or the server domain
int FingerID = 0; // The Fingerprint ID from the scanner
uint8_t id;
//*****Biometric Icons*****
#define Wifi_start_width 54
#define Wifi_start_height 49
const uint8_t PROGMEM Wifi_start_bits[] = {
0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x1f,0xf0,0x00,0x00,0x00,
0x00,0x03,0xff,0xff,0x80,0x00,0x00,
0x00,0x1f,0xf0,0x1f,0xf0,0x00,0x00,
0x00,0x7e,0x00,0x00,0xfc,0x00,0x00,
0x01,0xf0,0x00,0x00,0x1f,0x00,0x00,
0x03,0xc0,0x00,0x00,0x07,0xc0,0x00,
0x0f,0x00,0x00,0x00,0x01,0xe0,0x00,
0x1c,0x00,0x00,0x00,0x00,0x70,0x00,
0x38,0x00,0x07,0xc0,0x00,0x38,0x00,
0x70,0x00,0xff,0xfe,0x00,0x1e,0x00,
0xe0,0x03,0xfc,0x7f,0xc0,0x0e,0x00,
0x00,0x1f,0x80,0x03,0xf0,0x00,0x00,
0x00,0x3c,0x00,0x00,0x78,0x00,0x00,
0x00,0xf0,0x00,0x00,0x1c,0x00,0x00,
0x01,0xe0,0x00,0x00,0x0c,0x00,0x00,
0x03,0x80,0x00,0x00,0x00,0x00,0x00,
0x03,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x3f,0xf8,0x07,0x1e,0x00,
0x00,0x00,0xff,0xfe,0x1f,0xbf,0x80,
0x00,0x03,0xe0,0x04,0x7f,0xff,0xc0,
0x00,0x07,0x80,0x00,0xff,0xff,0xe0,
0x00,0x0e,0x00,0x00,0xff,0xff,0xe0,
0x00,0x0c,0x00,0x00,0x7f,0xff,0xc0,
0x00,0x00,0x00,0x00,0xfe,0x07,0xe0,
0x00,0x00,0x00,0x03,0xf8,0x03,0xf8,
0x00,0x00,0x07,0xe7,0xf9,0xf1,0xfc,
0x00,0x00,0x1f,0xe7,0xf1,0xf9,0xfc,
0x00,0x00,0x3f,0xe7,0xf3,0xf9,0xfc,
0x00,0x00,0x3f,0xe7,0xf1,0xf1,0xfc,
0x00,0x00,0x3f,0xe3,0xf8,0xe3,0xfc,
0x00,0x00,0x3f,0xf3,0xfc,0x07,0xf8,
0x00,0x00,0x1f,0xf0,0x7f,0x0f,0xc0,
0x00,0x00,0x0f,0xe0,0x7f,0xff,0xe0,
0x00,0x00,0x07,0xc0,0xff,0xff,0xe0,
0x00,0x00,0x00,0x00,0x7f,0xff,0xe0,
0x00,0x00,0x00,0x00,0x3f,0xff,0x80,
0x00,0x00,0x00,0x00,0x1f,0xbf,0x00,
0x00,0x00,0x00,0x00,0x03,0x18,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,
};
#define Wifi_connected_width 63
#define Wifi_connected_height 49

```

```

const uint8_t PROGMEM Wifi_connected_bits[] = {
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x03,0xff,0xff,0x80,0x00,0x00
,0x00,0x00,0x3f,0xff,0xff,0xf8,0x00,0x00
,0x00,0x01,0xff,0xff,0xff,0xff,0x00,0x00
,0x00,0x0f,0xff,0xff,0xff,0xff,0xe0,0x00
,0x00,0x3f,0xff,0xc0,0x07,0xff,0xf8,0x00
,0x00,0xff,0xf8,0x00,0x00,0x3f,0xfe,0x00
,0x03,0xff,0x80,0x00,0x00,0x03,0xff,0x80
,0x07,0xfe,0x00,0x00,0x00,0x00,0xff,0xc0
,0x1f,0xf8,0x00,0x00,0x00,0x00,0x3f,0xf0
,0x3f,0xe0,0x01,0xff,0xff,0x00,0x0f,0xf8
,0x7f,0x80,0x0f,0xff,0xff,0xe0,0x03,0xfc
,0xff,0x00,0x7f,0xff,0xff,0xfc,0x01,0xfe
,0xfc,0x01,0xff,0xff,0xff,0xff,0x00,0x7e
,0x78,0x07,0xff,0xc0,0x07,0xff,0xc0,0x3c
,0x00,0x0f,0xfc,0x00,0x00,0x7f,0xe0,0x00
,0x00,0x1f,0xf0,0x00,0x00,0x1f,0xf0,0x00
,0x00,0x3f,0xc0,0x00,0x00,0x07,0xf8,0x00
,0x00,0x7f,0x00,0x01,0x00,0x01,0xfc,0x00
,0x00,0x7e,0x00,0x7f,0xfc,0x00,0xfc,0x00
,0x00,0x3c,0x03,0xff,0xff,0x80,0x78,0x00
,0x00,0x00,0x07,0xff,0xff,0xc0,0x00,0x00
,0x00,0x00,0x1f,0xff,0xff,0xf0,0x00,0x00
,0x00,0x00,0x3f,0xf0,0x1f,0xf8,0x00,0x00
,0x00,0x00,0x3f,0x80,0x03,0xf8,0x00,0x00
,0x00,0x00,0x3f,0x00,0x01,0xf8,0x00,0x00
,0x00,0x00,0x1c,0x00,0x00,0x70,0x00,0x00
,0x00,0x00,0x00,0x01,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x0f,0xe0,0x00,0x00,0x00
,0x00,0x00,0x00,0x1f,0xf0,0x00,0x00,0x00
,0x00,0x00,0x00,0x3f,0xf8,0x00,0x00,0x00
,0x00,0x00,0x00,0x3f,0xf8,0x00,0x00,0x00
,0x00,0x00,0x00,0x3f,0xf8,0x00,0x00,0x00
,0x00,0x00,0x00,0x1f,0xf0,0x00,0x00,0x00
,0x00,0x00,0x00,0x0f,0xe0,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
};
#define FinPr_start_width 64
#define FinPr_start_height 64
const uint8_t PROGMEM FinPr_start_bits[] = {
0x00,0x00,0x00,0x1f,0xe0,0x00,0x00,0x00
,0x00,0x00,0x01,0xff,0xfe,0x00,0x00,0x00
,0x00,0x00,0x03,0xff,0xff,0x80,0x00,0x00
,0x00,0x00,0x0f,0xc0,0x0f,0xe0,0x00,0x00
,0x00,0x00,0x1f,0x00,0x01,0xf8,0x00,0x00
,0x00,0x00,0x3c,0x00,0x00,0x7c,0x00,0x00

```



```

,0x00,0x00,0x78,0x00,0x00,0x3e,0x00,0x00
,0x00,0x00,0xf0,0x3f,0xf8,0x0f,0x00,0x00
,0x00,0x01,0xe0,0xff,0xfe,0x07,0x80,0x00
,0x00,0x03,0xc3,0xff,0xff,0x03,0x80,0x00
,0x00,0x03,0x87,0xc0,0x07,0xc3,0xc0,0x00
,0x00,0x07,0x0f,0x00,0x03,0xe1,0xc0,0x00
,0x00,0x0f,0x0e,0x00,0x00,0xe0,0xe0,0x00
,0x00,0x0e,0x1c,0x00,0x00,0xf0,0xe0,0x00
,0x00,0x0c,0x3c,0x1f,0xe0,0x70,0xe0,0x00
,0x00,0x00,0x38,0x3f,0xf0,0x38,0x70,0x00
,0x00,0x00,0x78,0x78,0xf8,0x38,0x70,0x00
,0x00,0x00,0x70,0x70,0x3c,0x18,0x70,0x00
,0x00,0x00,0xe0,0xe0,0x1e,0x1c,0x70,0x00
,0x00,0x03,0xe1,0xe0,0x0e,0x1c,0x70,0x00
,0x00,0x0f,0xc1,0xc3,0x0e,0x1c,0x70,0x00
,0x00,0x3f,0x03,0xc3,0x8e,0x1c,0x70,0x00
,0x00,0x3e,0x03,0x87,0x0e,0x1c,0x70,0x00
,0x00,0x30,0x07,0x07,0x0e,0x18,0xe0,0x00
,0x00,0x00,0x0e,0x0e,0x0e,0x38,0xe0,0x00
,0x00,0x00,0x3e,0x1e,0x1e,0x38,0xe0,0x00
,0x00,0x00,0xf8,0x1c,0x1c,0x38,0xe0,0x00
,0x00,0x03,0xf0,0x38,0x3c,0x38,0xe0,0x00
,0x00,0x3f,0xc0,0xf8,0x78,0x38,0xe0,0x00
,0x00,0x7f,0x01,0xf0,0x70,0x38,0xf0,0x00
,0x00,0x78,0x03,0xe0,0xe0,0x38,0x70,0x00
,0x00,0x00,0x0f,0x81,0xe0,0x38,0x7c,0x00
,0x00,0x00,0x3f,0x03,0xc0,0x38,0x3e,0x00
,0x00,0x00,0xfc,0x0f,0x80,0x38,0x1e,0x00
,0x00,0x07,0xf0,0x1f,0x1c,0x1c,0x04,0x00
,0x00,0x3f,0xc0,0x3e,0x3f,0x1e,0x00,0x00
,0x00,0x7f,0x00,0xf8,0x7f,0x0f,0x00,0x00
,0x00,0x38,0x01,0xf0,0xf7,0x07,0xc0,0x00
,0x00,0x00,0x07,0xe1,0xe3,0x83,0xf8,0x00
,0x00,0x00,0x3f,0x87,0xc3,0xc0,0xfc,0x00
,0x00,0x01,0xfe,0x0f,0x81,0xe0,0x3c,0x00
,0x00,0x0f,0xf8,0x1f,0x00,0xf0,0x00,0x00
,0x00,0x1f,0xc0,0x7c,0x00,0x7c,0x00,0x00
,0x00,0x1e,0x01,0xf8,0x00,0x3f,0x00,0x00
,0x00,0x00,0x07,0xe0,0x78,0x0f,0xc0,0x00
,0x00,0x00,0x3f,0x81,0xfe,0x07,0xf0,0x00
,0x00,0x01,0xfe,0x07,0xff,0x01,0xf0,0x00
,0x00,0x07,0xf8,0x0f,0x87,0x80,0x30,0x00
,0x00,0x07,0xc0,0x3f,0x03,0xe0,0x00,0x00
,0x00,0x06,0x00,0xfc,0x01,0xf8,0x00,0x00
,0x00,0x00,0x03,0xf0,0x00,0x7e,0x00,0x00
,0x00,0x00,0x0f,0xc0,0x00,0x3f,0x80,0x00
,0x00,0x00,0x7f,0x00,0xf8,0x0f,0x80,0x00
,0x00,0x00,0xfc,0x03,0xfe,0x01,0x80,0x00
,0x00,0x00,0xf0,0x1f,0xff,0x80,0x00,0x00
,0x00,0x00,0x00,0x7f,0x07,0xe0,0x00,0x00
,0x00,0x00,0x00,0xfc,0x03,0xf8,0x00,0x00
,0x00,0x00,0x03,0xf0,0x00,0x78,0x00,0x00
,0x00,0x00,0x0f,0xc0,0x00,0x18,0x00,0x00
,0x00,0x00,0x0f,0x01,0xf8,0x00,0x00,0x00
,0x00,0x00,0x00,0x07,0xfe,0x00,0x00,0x00
,0x00,0x00,0x00,0x1f,0xfe,0x00,0x00,0x00
,0x00,0x00,0x00,0x1e,0x0e,0x00,0x00,0x00
,0x00,0x00,0x00,0x18,0x00,0x00,0x00,0x00
};
//-----

```

```

#define FinPr_valid_width 64
#define FinPr_valid_height 64
const uint8_t PROGMEM FinPr_valid_bits[] = {
0x00,0x00,0x03,0xfe,0x00,0x00,0x00,0x00
,0x00,0x00,0x1f,0xff,0xe0,0x00,0x00,0x00
,0x00,0x00,0x7f,0xff,0xf8,0x00,0x00,0x00
,0x00,0x00,0xfc,0x00,0xfe,0x00,0x00,0x00
,0x00,0x03,0xe0,0x00,0x1f,0x00,0x00,0x00
,0x00,0x07,0xc0,0x00,0x07,0x80,0x00,0x00
,0x00,0x0f,0x80,0x00,0x03,0xe0,0x00,0x00
,0x00,0x0e,0x03,0xff,0x01,0xe0,0x00,0x00
,0x00,0x1c,0x1f,0xff,0xe0,0xf0,0x00,0x00
,0x00,0x3c,0x3f,0xff,0xf0,0x78,0x00,0x00
,0x00,0x78,0x7c,0x00,0xf8,0x3c,0x00,0x00
,0x00,0x70,0xf0,0x00,0x3c,0x1c,0x00,0x00
,0x00,0xe1,0xe0,0x00,0x1e,0x1c,0x00,0x00
,0x00,0xe1,0xc0,0x00,0x0f,0x0e,0x00,0x00
,0x00,0xc3,0x81,0xfc,0x07,0x0e,0x00,0x00
,0x00,0x03,0x83,0xff,0x07,0x8e,0x00,0x00
,0x00,0x07,0x07,0x8f,0x83,0x87,0x00,0x00
,0x00,0x0f,0x0f,0x03,0xc3,0x87,0x00,0x00
,0x00,0x1e,0x0e,0x01,0xc3,0x87,0x00,0x00
,0x00,0x3c,0x1c,0x00,0xe1,0x87,0x00,0x00
,0x00,0xf8,0x1c,0x30,0xe1,0x87,0x00,0x00
,0x07,0xf0,0x38,0x70,0xe1,0x86,0x00,0x00
,0x07,0xc0,0x78,0x70,0xe3,0x8e,0x00,0x00
,0x02,0x00,0xf0,0xf0,0xe3,0x8e,0x00,0x00
,0x00,0x01,0xe0,0xe0,0xe3,0x8e,0x00,0x00
,0x00,0x03,0xc1,0xe1,0xc3,0x8e,0x00,0x00
,0x00,0x0f,0x83,0xc3,0xc3,0x8e,0x00,0x00
,0x00,0x7f,0x07,0x83,0x83,0x0e,0x00,0x00
,0x07,0xfc,0x0f,0x07,0x83,0x0e,0x00,0x00
,0x07,0xf0,0x1e,0x0f,0x03,0x0e,0x00,0x00
,0x07,0x80,0x7c,0x1e,0x03,0x07,0x00,0x00
,0x00,0x00,0xf8,0x3c,0x03,0x87,0x80,0x00
,0x00,0x03,0xf0,0x78,0x03,0x83,0xc0,0x00
,0x00,0x1f,0xc0,0xf0,0x02,0x00,0x00,0x00
,0x00,0xff,0x01,0xe1,0xc0,0x0c,0x00,0x00
,0x07,0xfc,0x03,0xc3,0xe1,0xff,0xc0,0x00
,0x07,0xe0,0x0f,0x87,0xc7,0xff,0xf0,0x00
,0x07,0x00,0x3f,0x0f,0x0f,0xff,0xfc,0x00
,0x00,0x00,0x7c,0x3e,0x3f,0xff,0xfe,0x00
,0x00,0x03,0xf8,0x7c,0x3f,0xff,0xff,0x00
,0x00,0x1f,0xe0,0xf0,0x7f,0xff,0xff,0x80
,0x00,0xff,0x83,0xe0,0xff,0xff,0xff,0x80
,0x01,0xfc,0x07,0xc1,0xff,0xff,0xe3,0xc0
,0x01,0xe0,0x1f,0x01,0xff,0xff,0xc3,0xc0
,0x00,0x00,0xfe,0x01,0xff,0xff,0x87,0xe0
,0x00,0x03,0xf8,0x13,0xff,0xff,0x0f,0xe0
,0x00,0x1f,0xe0,0x73,0xff,0xfe,0x1f,0xe0
,0x00,0x7f,0x81,0xf3,0xff,0xfc,0x1f,0xe0
,0x00,0xfc,0x03,0xe3,0xef,0xf8,0x3f,0xe0
,0x00,0x60,0x0f,0xc3,0xc7,0xf0,0x7f,0xe0
,0x00,0x00,0x3f,0x03,0xc3,0xe0,0xff,0xe0
,0x00,0x00,0xfc,0x03,0xc1,0xc1,0xff,0xe0
,0x00,0x07,0xf0,0x13,0xe0,0x83,0xff,0xe0
,0x00,0x0f,0xc0,0x7b,0xf8,0x07,0xff,0xe0
,0x00,0x0f,0x01,0xf9,0xfc,0x0f,0xff,0xc0
,0x00,0x00,0x07,0xf1,0xfe,0x1f,0xff,0xc0
,0x00,0x00,0x1f,0xc0,0xff,0x3f,0xff,0x80

```

```

,0x00,0x00,0x7e,0x00,0xff,0xff,0xff,0x80
,0x00,0x00,0xfc,0x00,0x7f,0xff,0xff,0x00
,0x00,0x00,0xf0,0x1f,0x3f,0xff,0xfe,0x00
,0x00,0x00,0x00,0x7f,0x1f,0xff,0xfc,0x00
,0x00,0x00,0x01,0xff,0x8f,0xff,0xf8,0x00
,0x00,0x00,0x03,0xe0,0xe3,0xff,0xe0,0x00
,0x00,0x00,0x01,0x80,0x00,0x7f,0x00,0x00
};
//-----
#define FinPr_invalid_width 64
#define FinPr_invalid_height 64
const uint8_t PROGMEM FinPr_invalid_bits[] = {
0x00,0x00,0x03,0xfe,0x00,0x00,0x00,0x00
,0x00,0x00,0x1f,0xff,0xe0,0x00,0x00,0x00
,0x00,0x00,0x7f,0xff,0xf8,0x00,0x00,0x00
,0x00,0x00,0xfc,0x00,0xfe,0x00,0x00,0x00
,0x00,0x03,0xe0,0x00,0x1f,0x00,0x00,0x00
,0x00,0x07,0xc0,0x00,0x07,0x80,0x00,0x00
,0x00,0x0f,0x80,0x00,0x03,0xe0,0x00,0x00
,0x00,0x0e,0x03,0xff,0x01,0xe0,0x00,0x00
,0x00,0x1c,0x1f,0xff,0xe0,0xf0,0x00,0x00
,0x00,0x3c,0x3f,0xff,0xf0,0x78,0x00,0x00
,0x00,0x78,0x7c,0x00,0xf8,0x3c,0x00,0x00
,0x00,0x70,0xf0,0x00,0x3c,0x1c,0x00,0x00
,0x00,0xe1,0xe0,0x00,0x1e,0x1c,0x00,0x00
,0x00,0xe1,0xc0,0x00,0x0f,0x0e,0x00,0x00
,0x00,0xc3,0x81,0xfc,0x07,0x0e,0x00,0x00
,0x00,0x03,0x83,0xff,0x07,0x8e,0x00,0x00
,0x00,0x07,0x07,0x8f,0x83,0x87,0x00,0x00
,0x00,0x0f,0x0f,0x03,0xc3,0x87,0x00,0x00
,0x00,0x1e,0x0e,0x01,0xc3,0x87,0x00,0x00
,0x00,0x3c,0x1c,0x00,0xe1,0x87,0x00,0x00
,0x00,0xf8,0x1c,0x30,0xe1,0x87,0x00,0x00
,0x07,0xf0,0x38,0x70,0xe1,0x86,0x00,0x00
,0x07,0xc0,0x78,0x70,0xe3,0x8e,0x00,0x00
,0x02,0x00,0xf0,0xf0,0xe3,0x8e,0x00,0x00
,0x00,0x01,0xe0,0xe0,0xe3,0x8e,0x00,0x00
,0x00,0x03,0xc1,0xe1,0xc3,0x8e,0x00,0x00
,0x00,0x0f,0x83,0xc3,0xc3,0x8e,0x00,0x00
,0x00,0x7f,0x07,0x83,0x83,0x0e,0x00,0x00
,0x07,0xfc,0x0f,0x07,0x83,0x0e,0x00,0x00
,0x07,0xf0,0x1e,0x0f,0x03,0x0e,0x00,0x00
,0x07,0x80,0x7c,0x1e,0x03,0x07,0x00,0x00
,0x00,0x00,0xf8,0x3c,0x03,0x87,0x80,0x00
,0x00,0x03,0xf0,0x78,0x03,0x83,0xc0,0x00
,0x00,0x1f,0xc0,0xf0,0x02,0x00,0x00,0x00
,0x00,0xff,0x01,0xe1,0xc0,0x00,0x00,0x00
,0x07,0xfc,0x03,0xc3,0xe1,0xff,0xc0,0x00
,0x07,0xe0,0x0f,0x87,0xc7,0xff,0xf0,0x00
,0x07,0x00,0x3f,0x0f,0x0f,0xff,0xf8,0x00
,0x00,0x00,0x7c,0x3e,0x1f,0xff,0xfe,0x00
,0x00,0x03,0xf8,0x7c,0x3f,0xff,0xff,0x00
,0x00,0x1f,0xe0,0xf0,0x7f,0xff,0xff,0x00
,0x00,0xff,0x83,0xe0,0xfe,0xff,0xbf,0x80
,0x01,0xfc,0x07,0xc0,0xfc,0x7f,0x1f,0xc0
,0x01,0xe0,0x1f,0x01,0xf8,0x3e,0x0f,0xc0
,0x00,0x00,0xfe,0x01,0xf8,0x1c,0x07,0xe0
,0x00,0x03,0xf8,0x13,0xf8,0x00,0x0f,0xe0
,0x00,0x1f,0xe0,0x73,0xfc,0x00,0x1f,0xe0
,0x00,0x7f,0x81,0xf3,0xfe,0x00,0x3f,0xe0

```

```
,0x00,0xfc,0x03,0xe3,0xff,0x00,0x7f,0xe0
,0x00,0x60,0x0f,0xc3,0xff,0x80,0xff,0xe0
,0x00,0x00,0x3f,0x03,0xff,0x00,0x7f,0xe0
,0x00,0x00,0xfc,0x03,0xfe,0x00,0x3f,0xe0
,0x00,0x07,0xf0,0x13,0xfc,0x00,0x1f,0xe0
,0x00,0x0f,0xc0,0x79,0xf8,0x08,0x0f,0xe0
,0x00,0x0f,0x01,0xf9,0xf8,0x1c,0x0f,0xc0
,0x00,0x00,0x07,0xf1,0xfc,0x3e,0x1f,0xc0
,0x00,0x00,0x1f,0xc0,0xfe,0x7f,0x3f,0x80
,0x00,0x00,0x7e,0x00,0xff,0xff,0xff,0x80
,0x00,0x00,0xfc,0x00,0x7f,0xff,0xff,0x00
,0x00,0x00,0xf0,0x1f,0x3f,0xff,0xfe,0x00
,0x00,0x00,0x00,0x7f,0x1f,0xff,0xfc,0x00
,0x00,0x00,0x01,0xff,0x8f,0xff,0xf8,0x00
,0x00,0x00,0x03,0xe0,0xe3,0xff,0xe0,0x00
,0x00,0x00,0x01,0x80,0x00,0x7f,0x00,0x00
};
```

```
//-----
```

```
#define FinPr_failed_width 64
```

```
#define FinPr_failed_height 64
```

```
const uint8_t PROGMEM FinPr_failed_bits[] = {
0x00,0x00,0x3f,0xe0,0x00,0x00,0x00,0x00
,0x00,0x01,0xff,0xfe,0x00,0x00,0x00,0x00
,0x00,0x0f,0xc0,0x1f,0x80,0x00,0x00,0x00
,0x00,0x1e,0x00,0x03,0xc0,0x00,0x00,0x00
,0x00,0x78,0x00,0x00,0xf0,0x00,0x00,0x00
,0x00,0xe0,0x00,0x00,0x38,0x00,0x00,0x00
,0x01,0xc0,0x00,0x00,0x1c,0x00,0x00,0x00
,0x03,0x80,0x00,0x00,0x0e,0x00,0x00,0x00
,0x07,0x00,0x7f,0xe0,0x07,0x00,0x00,0x00
,0x06,0x01,0xff,0xf8,0x03,0x00,0x00,0x00
,0x0c,0x03,0xc0,0x3c,0x03,0x80,0x00,0x00
,0x1c,0x0f,0x00,0x0e,0x01,0x80,0x00,0x00
,0x18,0x0c,0x00,0x03,0x00,0xc0,0x00,0x00
,0x18,0x18,0x00,0x01,0x80,0xc0,0x00,0x00
,0x30,0x38,0x00,0x01,0xc0,0xe0,0x00,0x00
,0x30,0x30,0x0f,0x00,0xc0,0x60,0x00,0x00
,0x30,0x30,0x3f,0xc0,0xe0,0x60,0x00,0x00
,0x70,0x60,0x78,0xe0,0x60,0x60,0x00,0x00
,0x60,0x60,0x60,0x60,0x60,0x70,0x00,0x00
,0x60,0x60,0x60,0x60,0x60,0x30,0x00,0x00
,0x60,0x60,0x60,0x60,0x30,0x30,0x00,0x00
,0x60,0x60,0x60,0x30,0x30,0x20,0x00,0x00
,0x60,0x60,0x60,0x30,0x30,0x01,0xe0,0x00
,0x60,0x60,0x60,0x30,0x30,0x0f,0xfc,0x00
,0x60,0x60,0x60,0x30,0x30,0x3f,0xff,0x00
,0x60,0x60,0x60,0x30,0x18,0x78,0x03,0x80
,0x60,0x60,0x60,0x30,0x1c,0x60,0x01,0x80
,0x60,0x60,0x30,0x38,0x0c,0xc0,0x00,0xc0
,0x00,0x60,0x30,0x18,0x00,0xc0,0x00,0xc0
,0x00,0x60,0x30,0x18,0x00,0xc0,0x00,0xc0
,0x00,0xe0,0x30,0x0c,0x01,0xc0,0x00,0xe0
,0x00,0xc0,0x18,0x0e,0x01,0xc0,0x00,0xe0
,0x60,0xc0,0x18,0x07,0x01,0xc0,0x00,0xe0
,0x01,0xc0,0x1c,0x03,0x81,0xc0,0x00,0xe0
,0x01,0x80,0x0c,0x01,0xc1,0xc0,0x00,0xe0
,0x03,0x80,0x0e,0x00,0xf1,0xc0,0x00,0xe0
,0x0f,0x00,0x06,0x00,0x01,0xc0,0x00,0xe0
,0x3e,0x01,0x03,0x00,0x01,0xc0,0x00,0xe0
,0x30,0x03,0x83,0x80,0x1f,0xff,0xfe
```

```

,0x00,0x03,0x81,0xc0,0x3f,0xff,0xff,0xff
,0x00,0x07,0xc0,0xe0,0x30,0x00,0x00,0x03
,0x00,0x0e,0xc0,0x78,0x30,0x00,0x00,0x03
,0x00,0x3c,0x60,0x1e,0x30,0x00,0x00,0x03
,0x00,0x78,0x70,0x0f,0x30,0x00,0x00,0x03
,0x03,0xe0,0x38,0x03,0x30,0x00,0x00,0x03
,0x07,0x80,0x1c,0x00,0x30,0x00,0x00,0x03
,0xc0,0x00,0x0f,0x00,0x30,0x00,0x00,0x03
,0xc0,0x00,0x03,0x80,0x30,0x01,0xe0,0x03
,0x00,0x18,0x01,0xe0,0x30,0x03,0xf0,0x03
,0x00,0x18,0x00,0x7c,0x30,0x07,0x38,0x03
,0x00,0x0c,0x00,0x1f,0x30,0x06,0x18,0x03
,0x18,0x0e,0x00,0x07,0x30,0x06,0x18,0x03
,0x0c,0x07,0x80,0x00,0x30,0x07,0x38,0x03
,0x0e,0x03,0xc0,0x00,0x30,0x03,0x30,0x03
,0x07,0x00,0xf0,0x00,0x30,0x03,0x30,0x03
,0x03,0x00,0x7e,0x00,0x30,0x03,0x30,0x03
,0x01,0x80,0x1f,0xc0,0x30,0x03,0x30,0x03
,0x01,0xc0,0x03,0xe1,0x30,0x07,0xf8,0x03
,0x00,0xf0,0x00,0x01,0x30,0x03,0xf0,0x03
,0x00,0x38,0x00,0x00,0x30,0x00,0x00,0x03
,0x00,0x1e,0x00,0x00,0x30,0x00,0x00,0x03
,0x00,0x07,0xc0,0x00,0x30,0x00,0x00,0x03
,0x00,0x01,0xff,0x80,0x3f,0xff,0xff,0xff
,0x00,0x00,0x3f,0x80,0x1f,0xff,0xff,0xfe
};
//-----
#define FinPr_scan_width 64
#define FinPr_scan_height 64
const uint8_t PROGMEM FinPr_scan_bits[] = {
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x1f,0xf8,0x00,0x00,0x00
,0x00,0x00,0x00,0xf7,0xff,0x00,0x00,0x00
,0x00,0x00,0x01,0xfc,0x7f,0xc0,0x00,0x00
,0x00,0x00,0x03,0xc0,0x03,0xe0,0x00,0x00
,0x00,0x00,0x07,0x80,0x00,0xf0,0x00,0x00
,0x00,0x00,0x0e,0x00,0x00,0x3c,0x00,0x00
,0x00,0x00,0x1c,0x1f,0xfc,0x1c,0x00,0x00
,0x00,0x00,0x38,0x7f,0xfe,0x0e,0x00,0x00
,0x00,0x00,0x78,0xf8,0x0f,0x87,0x00,0x00
,0x00,0x00,0x71,0xe0,0x03,0xc7,0x00,0x00
,0x00,0x00,0xe3,0x80,0x01,0xc3,0x80,0x00
,0x00,0x00,0xc3,0x83,0xc0,0xe3,0x80,0x00
,0x00,0x00,0xc7,0x0f,0xf0,0x71,0x80,0x00
,0x00,0x00,0x06,0x1f,0xf8,0x71,0xc0,0x00
,0x00,0x00,0x0e,0x1c,0x3c,0x31,0xc0,0x00
,0x00,0x00,0x1c,0x38,0x1c,0x31,0xc0,0x00
,0x00,0x00,0x38,0x70,0x0e,0x39,0xc0,0x00
,0x00,0x01,0xf0,0x71,0x8e,0x39,0xc0,0x00
,0x00,0x03,0xe0,0xe1,0x86,0x31,0xc0,0x00
,0x00,0x03,0x81,0xe3,0x8e,0x31,0x80,0x00
,0x00,0x00,0x03,0xc3,0x8e,0x33,0x80,0x00
,0x00,0x00,0x07,0x87,0x0c,0x73,0x80,0x00
,0x00,0x00,0x1f,0x0e,0x1c,0x73,0x80,0x00
,0x7f,0xff,0xff,0xff,0xff,0xff,0xff,0xfe
,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff

```

```
,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff
,0x7f,0xff,0xff,0xff,0xff,0xff,0xff,0xfe
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x03,0xf0,0x1e,0x3e,0x1c,0x00,0x00
,0x00,0x03,0x80,0x7c,0x77,0x0f,0x00,0x00
,0x00,0x00,0x01,0xf0,0xe3,0x07,0xc0,0x00
,0x00,0x00,0x07,0xe3,0xc3,0x81,0xf0,0x00
,0x00,0x00,0x3f,0x87,0x81,0xc0,0x60,0x00
,0x00,0x01,0xfc,0x1f,0x00,0xf0,0x00,0x00
,0x00,0x01,0xe0,0x3c,0x00,0x7c,0x00,0x00
,0x00,0x00,0x00,0xf8,0x78,0x1f,0x00,0x00
,0x00,0x00,0x07,0xe0,0xfc,0x0f,0xc0,0x00
,0x00,0x00,0x3f,0x83,0xef,0x03,0xc0,0x00
,0x00,0x00,0xfc,0x0f,0x87,0x80,0x00,0x00
,0x00,0x00,0x70,0x1f,0x03,0xe0,0x00,0x00
,0x00,0x00,0x00,0x7c,0x00,0xf8,0x00,0x00
,0x00,0x00,0x01,0xf0,0x00,0x3e,0x00,0x00
,0x00,0x00,0x0f,0xc0,0xf8,0x0f,0x00,0x00
,0x00,0x00,0x1f,0x03,0xfe,0x02,0x00,0x00
,0x00,0x00,0x0c,0x0f,0x8f,0x80,0x00,0x00
,0x00,0x00,0x00,0x3f,0x03,0xe0,0x00,0x00
,0x00,0x00,0x00,0xf8,0x00,0xf0,0x00,0x00
,0x00,0x00,0x01,0xe0,0x00,0x30,0x00,0x00
,0x00,0x00,0x01,0xc0,0xf8,0x00,0x00,0x00
,0x00,0x00,0x00,0x07,0xfe,0x00,0x00,0x00
,0x00,0x00,0x00,0x0f,0x8e,0x00,0x00,0x00
,0x00,0x00,0x00,0x06,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
};
//*****
```

```
void setup() {
```

```
Serial.begin(115200);
```

```
//-----initiate OLED display-----
```

```
// SSD1306_SWITCHCAPVCC = generate display voltage from 3.3V internally
if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address 0x3D for 128x64
Serial.println(F("SSD1306 allocation failed"));
for(;;); // Don't proceed, loop forever
}
```

```
// Show initial display buffer contents on the screen --
// the library initializes this with an Adafruit splash screen.
// you can delete these three lines if you don't want to get the Adfruit logo appear
display.display();
delay(2000); // Pause for 2 seconds
display.clearDisplay();
```

```
//-----
```

```
connectToWiFi();
```

```
//-----
```

```

// set the data rate for the sensor serial port
finger.begin(57600);
Serial.println("\n\nAdafruit finger detect test");

if (finger.verifyPassword()) {
  Serial.println("Found fingerprint sensor!");
  display.clearDisplay();
  display.drawBitmap( 34, 0, FinPr_valid_bits, FinPr_valid_width, FinPr_valid_height, WHITE);
  display.display();
} else {
  Serial.println("Did not find fingerprint sensor :(");
  display.clearDisplay();
  display.drawBitmap( 32, 0, FinPr_failed_bits, FinPr_failed_width, FinPr_failed_height, WHITE);
  display.display();
  while (1) { delay(1); }
}
//-----

finger.getTemplateCount();
Serial.print("Sensor contains "); Serial.print(finger.templateCount); Serial.println(" templates");
Serial.println("Waiting for valid finger...");

//-----*test the connection*-----

//SendFingerprintID( FingerID );

}
//*****
void loop() {

  //check if there's a connection to WiFi or not
  if(WiFi.status() != WL_CONNECTED){
    connectToWiFi();
  }
  //-----
  //If there no fingerprint has been scanned return -1 or -2 if there an error or 0 if there nothing, The ID start
  //form 1 to 127
  FingerID = getFingerprintID(); // Get the Fingerprint ID from the Scanner
  delay(50); //don't need to run this at full speed.

  //-----

  DisplayFingerprintID();

  //-----

  ChecktoAddID();

  //-----

  ChecktoDeleteID();

  //-----
}
//*****Display the fingerprint ID state on the OLED*****
void DisplayFingerprintID(){
  //Fingerprint has been detected
  if (FingerID > 0){
    display.clearDisplay();

```

```

display.drawBitmap( 34, 0, FinPr_valid_bits, FinPr_valid_width, FinPr_valid_height, WHITE);
display.display();

SendFingerprintID( FingerID ); // Send the Fingerprint ID to the website.

}
//-----
//No finger detected
else if (FingerID == 0){
display.clearDisplay();
display.drawBitmap( 32, 0, FinPr_start_bits, FinPr_start_width, FinPr_start_height, WHITE);
display.display();
}
//-----
//Didn't find a match
else if (FingerID == -1){
display.clearDisplay();
display.drawBitmap( 34, 0, FinPr_invalid_bits, FinPr_invalid_width, FinPr_invalid_height, WHITE);
display.display();
}
//-----
//Didn't find the scanner or there an error
else if (FingerID == -2){
display.clearDisplay();
display.drawBitmap( 32, 0, FinPr_failed_bits, FinPr_failed_width, FinPr_failed_height, WHITE);
display.display();
}
}
//*****send the fingerprint ID to the website*****
void SendFingerprintID( int finger ){

  HTTPClient http; //Declare object of class HTTPClient
  //Post Data
  postData = "FingerID=" + String(finger); // Add the Fingerprint ID to the Post array in order to send it
  // Post methode

  http.begin(link); //initiate HTTP request, put your Website URL or Your Computer IP
  http.addHeader("Content-Type", "application/x-www-form-urlencoded"); //Specify content-type header

  int httpCode = http.POST(postData); //Send the request
  String payload = http.getString(); //Get the response payload

  Serial.println(httpCode); //Print HTTP return code
  Serial.println(payload); //Print request response payload
  Serial.println(postData); //Post Data
  Serial.println(finger); //Print fingerprint ID

  if (payload.substring(0, 5) == "login") {
    String user_name = payload.substring(5);
    // Serial.println(user_name);

    display.clearDisplay();
    display.setTextSize(2); // Normal 2:2 pixel scale
    display.setTextColor(WHITE); // Draw white text
    display.setCursor(15,0); // Start at top-left corner
    display.print(F("Welcome"));
    display.setCursor(0,20);
    display.print(user_name);
    display.display();
  }
}

```



```

else if (payload.substring(0, 6) == "logout") {
String user_name = payload.substring(6);
// Serial.println(user_name);

display.clearDisplay();
display.setTextSize(2); // Normal 2:2 pixel scale
display.setTextColor(WHITE); // Draw white text
display.setCursor(10,0); // Start at top-left corner
display.print(F("Good Bye"));
display.setCursor(0,20);
display.print(user_name);
display.display();
}
delay(1000);

postData = "";
http.end(); //Close connection
}
//*****Get the Fingerprint ID*****
int getFingerprintID() {
uint8_t p = finger.getImage();
switch (p) {
case FINGERPRINT_OK:
//Serial.println("Image taken");
break;
case FINGERPRINT_NOFINGER:
//Serial.println("No finger detected");
return 0;
case FINGERPRINT_PACKETRECEIVEERR:
//Serial.println("Communication error");
return -2;
case FINGERPRINT_IMAGEFAIL:
//Serial.println("Imaging error");
return -2;
default:
//Serial.println("Unknown error");
return -2;
}
// OK success!
p = finger.image2Tz();
switch (p) {
case FINGERPRINT_OK:
//Serial.println("Image converted");
break;
case FINGERPRINT_IMAGEMESS:
//Serial.println("Image too messy");
return -1;
case FINGERPRINT_PACKETRECEIVEERR:
//Serial.println("Communication error");
return -2;
case FINGERPRINT_FEATUREFAIL:
//Serial.println("Could not find fingerprint features");
return -2;
case FINGERPRINT_INVALIDIMAGE:
//Serial.println("Could not find fingerprint features");
return -2;
default:
//Serial.println("Unknown error");
return -2;
}
}

```

```

// OK converted!
p = finger.fingerFastSearch();
if (p == FINGERPRINT_OK) {
//Serial.println("Found a print match!");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
//Serial.println("Communication error");
return -2;
} else if (p == FINGERPRINT_NOTFOUND) {
//Serial.println("Did not find a match");
return -1;
} else {
//Serial.println("Unknown error");
return -2;
}
// found a match!
//Serial.print("Found ID #"); Serial.print(finger.fingerID);
//Serial.print(" with confidence of "); Serial.println(finger.confidence);

return finger.fingerID;
}
//*****Check if there a Fingerprint ID to delete*****
void CheckToDeleteID(){

HTTPClient http; //Declare object of class HTTPClient
//Post Data
postData = "DeleteID=check"; // Add the Fingerprint ID to the Post array in order to send it
// Post methode

http.begin(link); //initiate HTTP request, put your Website URL or Your Computer IP
http.addHeader("Content-Type", "application/x-www-form-urlencoded"); //Specify content-type header

int httpCode = http.POST(postData); //Send the request
String payload = http.getString(); //Get the response payload

if (payload.substring(0, 6) == "del-id") {
String del_id = payload.substring(6);
Serial.println(del_id);
deleteFingerprint( del_id.toInt() );
}

http.end(); //Close connection
}
//*****Delete Fingerprint ID*****
uint8_t deleteFingerprint( int id) {
uint8_t p = -1;

p = finger.deleteModel(id);

if (p == FINGERPRINT_OK) {
//Serial.println("Deleted!");
display.clearDisplay();
display.setTextSize(2); // Normal 2:2 pixel scale
display.setTextColor(WHITE); // Draw white text
display.setCursor(0,0); // Start at top-left corner
display.print(F("Deleted!\n"));
display.display();
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
//Serial.println("Communication error");
display.clearDisplay();
display.setTextSize(1); // Normal 1:1 pixel scale

```

```

display.setTextColor(WHITE); // Draw white text
display.setCursor(0,0); // Start at top-left corner
display.print(F("Communication error!\n"));
display.display();
return p;
} else if (p == FINGERPRINT_BADLOCATION) {
//Serial.println("Could not delete in that location");
display.clearDisplay();
display.setTextSize(1); // Normal 1:1 pixel scale
display.setTextColor(WHITE); // Draw white text
display.setCursor(0,0); // Start at top-left corner
display.print(F("Could not delete in that location!\n"));
display.display();
return p;
} else if (p == FINGERPRINT_FLASHERR) {
//Serial.println("Error writing to flash");
display.clearDisplay();
display.setTextSize(1); // Normal 1:1 pixel scale
display.setTextColor(WHITE); // Draw white text
display.setCursor(0,0); // Start at top-left corner
display.print(F("Error writing to flash!\n"));
display.display();
return p;
} else {
//Serial.print("Unknown error: 0x"); Serial.println(p, HEX);
display.clearDisplay();
display.setTextSize(2); // Normal 2:2 pixel scale
display.setTextColor(WHITE); // Draw white text
display.setCursor(0,0); // Start at top-left corner
display.print(F("Unknown error:\n"));
display.display();
return p;
}
}
//*****Check if there a Fingerprint ID to add*****
void ChecktoAddID(){

HTTPClient http; //Declare object of class HTTPClient
//Post Data
postData = "Get_Fingerid=get_id"; // Add the Fingerprint ID to the Post array in order to send it
// Post methode

http.begin(link); //initiate HTTP request, put your Website URL or Your Computer IP
http.addHeader("Content-Type", "application/x-www-form-urlencoded"); //Specify content-type header

int httpCode = http.POST(postData); //Send the request
String payload = http.getString(); //Get the response payload

if (payload.substring(0, 6) == "add-id") {
String add_id = payload.substring(6);
Serial.println(add_id);
id = add_id.toInt();
getFingerprintEnroll();
}
http.end(); //Close connection
}
//*****Enroll a Fingerprint ID*****
uint8_t getFingerprintEnroll() {

int p = -1;

```

```

display.clearDisplay();
display.drawBitmap( 34, 0, FinPr_scan_bits, FinPr_scan_width, FinPr_scan_height, WHITE);
display.display();
while (p != FINGERPRINT_OK) {
p = finger.getImage();
switch (p) {
case FINGERPRINT_OK:
//Serial.println("Image taken");
display.clearDisplay();
display.drawBitmap( 34, 0, FinPr_valid_bits, FinPr_valid_width, FinPr_valid_height, WHITE);
display.display();
break;
case FINGERPRINT_NOFINGER:
//Serial.println(".");
display.setTextSize(1); // Normal 2:2 pixel scale
display.setTextColor(WHITE); // Draw white text
display.setCursor(0,0); // Start at top-left corner
display.print(F("scanning"));
display.display();
break;
case FINGERPRINT_PACKETRECEIVEERR:
display.clearDisplay();
display.drawBitmap( 34, 0, FinPr_invalid_bits, FinPr_invalid_width, FinPr_invalid_height, WHITE);
display.display();
break;
case FINGERPRINT_IMAGEFAIL:
Serial.println("Imaging error");
break;
default:
Serial.println("Unknown error");
break;
}
}

// OK success!

p = finger.image2Tz(1);
switch (p) {
case FINGERPRINT_OK:
display.clearDisplay();
display.drawBitmap( 34, 0, FinPr_valid_bits, FinPr_valid_width, FinPr_valid_height, WHITE);
display.display();
break;
case FINGERPRINT_IMAGEMESS:
display.clearDisplay();
display.drawBitmap( 34, 0, FinPr_invalid_bits, FinPr_invalid_width, FinPr_invalid_height, WHITE);
display.display();
return p;
case FINGERPRINT_PACKETRECEIVEERR:
Serial.println("Communication error");
return p;
case FINGERPRINT_FEATUREFAIL:
Serial.println("Could not find fingerprint features");
return p;
case FINGERPRINT_INVALIDIMAGE:
Serial.println("Could not find fingerprint features");
return p;
default:
Serial.println("Unknown error");
return p;
}

```

```

}
display.clearDisplay();
display.setTextSize(2); // Normal 2:2 pixel scale
display.setTextColor(WHITE); // Draw white text
display.setCursor(0,0); // Start at top-left corner
display.print(F("Remove"));
display.setCursor(0,20);
display.print(F("finger"));
display.display();
//Serial.println("Remove finger");
delay(2000);
p = 0;
while (p != FINGERPRINT_NOFINGER) {
p = finger.getImage();
}
Serial.print("ID "); Serial.println(id);
p = -1;
display.clearDisplay();
display.drawBitmap( 34, 0, FinPr_scan_bits, FinPr_scan_width, FinPr_scan_height, WHITE);
display.display();
while (p != FINGERPRINT_OK) {
p = finger.getImage();
switch (p) {
case FINGERPRINT_OK:
//Serial.println("Image taken");
display.clearDisplay();
display.drawBitmap( 34, 0, FinPr_valid_bits, FinPr_valid_width, FinPr_valid_height, WHITE);
display.display();
break;
case FINGERPRINT_NOFINGER:
//Serial.println(". ");
display.setTextSize(1); // Normal 2:2 pixel scale
display.setTextColor(WHITE); // Draw white text
display.setCursor(0,0); // Start at top-left corner
display.print(F("scanning"));
display.display();
break;
case FINGERPRINT_PACKETRECEIVEERR:
Serial.println("Communication error");
break;
case FINGERPRINT_IMAGEFAIL:
Serial.println("Imaging error");
break;
default:
Serial.println("Unknown error");
break;
}
}

// OK success!

p = finger.image2Tz(2);
switch (p) {
case FINGERPRINT_OK:
//Serial.println("Image converted");
display.clearDisplay();
display.drawBitmap( 34, 0, FinPr_valid_bits, FinPr_valid_width, FinPr_valid_height, WHITE);
display.display();
break;
case FINGERPRINT_IMAGEMESS:

```

```

Serial.println("Image too messy");
return p;
case FINGERPRINT_PACKETRECEIVEERR:
Serial.println("Communication error");
return p;
case FINGERPRINT_FEATUREFAIL:
Serial.println("Could not find fingerprint features");
return p;
case FINGERPRINT_INVALIDIMAGE:
Serial.println("Could not find fingerprint features");
return p;
default:
Serial.println("Unknown error");
return p;
}

// OK converted!
Serial.print("Creating model for #"); Serial.println(id);

p = finger.createModel();
if (p == FINGERPRINT_OK) {
//Serial.println("Prints matched!");
display.clearDisplay();
display.drawBitmap( 34, 0, FinPr_valid_bits, FinPr_valid_width, FinPr_valid_height, WHITE);
display.display();
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
Serial.println("Communication error");
return p;
} else if (p == FINGERPRINT_ENROLLMISMATCH) {
Serial.println("Fingerprints did not match");
return p;
} else {
Serial.println("Unknown error");
return p;
}

Serial.print("ID "); Serial.println(id);
p = finger.storeModel(id);
if (p == FINGERPRINT_OK) {
//Serial.println("Stored!");
display.clearDisplay();
display.drawBitmap( 34, 0, FinPr_valid_bits, FinPr_valid_width, FinPr_valid_height, WHITE);
display.display();
confirmAdding();
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
Serial.println("Communication error");
return p;
} else if (p == FINGERPRINT_BADLOCATION) {
Serial.println("Could not store in that location");
return p;
} else if (p == FINGERPRINT_FLASHERR) {
Serial.println("Error writing to flash");
return p;
} else {
Serial.println("Unknown error");
return p;
}
}

//*****Check if there a Fingerprint ID to add*****
void confirmAdding(){

```

```

HTTPClient http; //Declare object of class HTTPClient
//Post Data
postData = "confirm_id=" + String(id); // Add the Fingerprint ID to the Post array in order to send it
// Post methode

http.begin(link); //initiate HTTP request, put your Website URL or Your Computer IP
http.addHeader("Content-Type", "application/x-www-form-urlencoded"); //Specify content-type header

int httpCode = http.POST(postData); //Send the request
String payload = http.getString(); //Get the response payload

display.clearDisplay();
display.setTextSize(1.5); // Normal 1:1 pixel scale
display.setTextColor(WHITE); // Draw white text
display.setCursor(0,0); // Start at top-left corner
display.print(payload);
display.display();
delay(1000);
Serial.println(payload);

http.end(); //Close connection
}
//*****connect to the WiFi*****
void connectToWiFi(){
  WiFi.mode(WIFI_OFF); //Prevents reconnection issue (taking too long to connect)
  delay(1000);
  WiFi.mode(WIFI_STA);
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);

  display.clearDisplay();
  display.setTextSize(1); // Normal 1:1 pixel scale
  display.setTextColor(WHITE); // Draw white text
  display.setCursor(0, 0); // Start at top-left corner
  display.print(F("Connecting to \n"));
  display.setCursor(0, 50);
  display.setTextSize(2);
  display.print(ssid);
  display.drawBitmap( 73, 10, Wifi_start_bits, Wifi_start_width, Wifi_start_height, WHITE);
  display.display();
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("Connected");
  display.clearDisplay();
  display.setTextSize(2); // Normal 1:1 pixel scale
  display.setTextColor(WHITE); // Draw white text
  display.setCursor(8, 0); // Start at top-left corner
  display.print(F("Connected \n"));
  display.drawBitmap( 33, 15, Wifi_connected_bits, Wifi_connected_width, Wifi_connected_height, WHITE);
  display.display();
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP()); //IP address assigned to your ESP

}
//=====

```

3.2 PHP & HTML CODE

a. index.php - Homepage for Biometric Fingerprint Attendance System.

```

<!DOCTYPE html>
<html>
<head>
  <title>Users</title>
  <link rel="stylesheet" type="text/css" href="Users.css">
  <script>
    $(window).on("load resize ", function() {
      var scrollWidth = $('tbl-content').width() - $('tbl-content table').width();
      $('tbl-header').css({'padding-right':scrollWidth});
    }).resize();
  </script>
</head>
<body>
  <?php include'header.php'; ?>
  <main>
    <section>
      <!--User table-->
      <h1 class="slideDown animated">Users Information</h1>
      <div class="tbl-header slideInRight animated">
        <table cellpadding="0" cellspacing="0" border="0">
          <thead>
            <tr>
              <th>ID | Name</th>
              <th>Serial Number</th>
              <th>Gender</th>
              <th>Finger ID</th>
              <th>Date</th>
              <th>Time In</th>
            </tr>
          </thead>
        </table>
      </div>
      <div class="tbl-content slideInRight animated">
        <table cellpadding="0" cellspacing="0" border="0">
          <tbody>
            <?php
              //Connect to database
              require'connectDB.php';

              $sql = "SELECT * FROM users WHERE NOT username="" ORDER BY id DESC";
              $result = mysqli_stmt_init($conn);
              if (!mysqli_stmt_prepare($result, $sql)) {
                echo '<p class="error">SQL Error</p>';
              }
              else{
                mysqli_stmt_execute($result);
                $result1 = mysqli_stmt_get_result($result);
                if (mysqli_num_rows($result1) > 0){
                  while ($row = mysqli_fetch_assoc($result1)){
                    ?>
                      <TR>
                        <TD><?php echo $row['id']; echo" | "; echo $row['username'];?></TD>
                        <TD><?php echo $row['serialnumber'];?></TD>
                        <TD><?php echo $row['gender'];?></TD>
                        <TD><?php echo $row['fingerprint_id'];?></TD>

```



```

        <TD><?php echo $row['user_date'];?></TD>
        <TD><?php echo $row['time_in'];?></TD>
    </TR>
<?php
    }
}
}
?>
</tbody>
</table>
</div>
</section>
</main>
</body>
</html>

```

b. **header.php** - PHP consists of the “USERS”, “USERS LOG” & “MANAGE USERS” Buttons.

```

<?php
session_start();
?>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
<link rel="stylesheet" type="text/css" href="header.css">
</head>
<header>
<div class="header">
    <div class="logo">
        <a href="index.php">Biometric Attendance</a>
    </div>
</div>

<div class="topnav" id="myTopnav">
    <a href="index.php">Users</a>
    <a href="UsersLog.php">Users Log</a>
    <a href="ManageUsers.php">Manage Users</a>
    <a href="javascript:void(0);" class="icon" onclick="navFunction()">
        <i class="fa fa-bars"></i></a>
</div>
</header>
<script>
function navFunction() {
    var x = document.getElementById("myTopnav");
    if (x.className === "topnav") {
        x.className += " responsive";
    } else {
        x.className = "topnav";
    }
}
</script>

```

c. manage_users_up.php - PHP that analyzes the given input of the user.

```

<table cellpadding="0" cellspacing="0" border="0">
<tbody>
<?php
//Connect to database
require'connectDB.php';

$sql = "SELECT * FROM users WHERE del_fingerid=0 ORDER BY id DESC";
$result = mysqli_stmt_init($conn);
if(!mysqli_stmt_prepare($result, $sql)) {
    echo '<p class="error">SQL Error</p>';
}else{
    mysqli_stmt_execute($result);
    $resultl = mysqli_stmt_get_result($result);
    if (mysqli_num_rows($resultl) > 0){
        while ($row = mysqli_fetch_assoc($resultl)){
            ?>
                <TR>
                <TD><?php
                    if ($row['fingerprint_select'] == 1) {
                        echo "<img src='icons/ok_check.png' title='The selected UID'>";
                    }
                    $fingerid = $row['fingerprint_id'];
                ?>
                <form>
                    <button type="button" class="select_btn" id="<?php echo $fingerid;?>"
title="select this UID"><?php echo $fingerid;?></button>
                </form>
                </TD>
                <TD><?php echo $row['username'];?></TD>
                <TD><?php echo $row['gender'];?></TD>
                <TD><?php echo $row['serialnumber'];?></TD>
                <TD><?php echo $row['user_date'];?></TD>
                <TD><?php echo $row['time_in'];?></TD>
                </TR>
            <?php
        }
    }?></tbody></table>

```

d. ManageUsers.php - PHP that gets users from the Fingerprint Sensor.

```

<!DOCTYPE html>
<html>
<head>
    <title>Manage Users</title>
    <link rel="stylesheet" type="text/css" href="manageusers.css">
    <script>
        $(window).on("load resize ", function() {
            var scrollWidth = $('tbl-content').width() - $('tbl-content table').width();
            $('tbl-header').css({'padding-right':scrollWidth});
        }).resize();
    </script>
    <script src="https://code.jquery.com/jquery-3.3.1.js"
        integrity="sha1256-2Kok7MbOyxpgUVVvAk/HJ2jigOSYS2auK4Pfzbm7uH60="
        crossorigin="anonymous">
    </script>

```

```

<script src="js/jquery-2.2.3.min.js"></script>
<script src="js/manage_users.js"></script>
<script>
    $(document).ready(function(){
        $.ajax({
            url: "manage_users_up.php"
        }).done(function(data) {
            $('#manage_users').html(data);
        });
        setInterval(function(){
            $.ajax({
                url: "manage_users_up.php"
            }).done(function(data) {
                $('#manage_users').html(data);
            });
        },5000);
    });
</script>
</head>
<body>
<?php include'header.php';?>
<main>
    <h1 class="slideDown animated">Edit User's Information or Add New User</h1>
    <div class="form-style-5 slideInDown animated">
        <div class="alert">
            <label id="alert"></label>
        </div>
        <form>
            <fieldset>
                <legend><span class="number">1</span> User Fingerprint ID:</legend>
                <label>Enter Fingerprint ID between 1 & 127:</label>
                <input type="number" name="fingerprintid" id="fingerprintid" placeholder="User
Fingerprint ID">
                <button type="button" name="fingerprintid_add" class="fingerprintid_add">Add
Fingerprint ID</button>
            </fieldset>
            <fieldset>
                <legend><span class="number">2</span> User Info</legend>
                <input type="text" name="name" id="name" placeholder="User Name">
                <input type="text" name="number" id="number" placeholder="Serial
Number">
                <input type="email" name="email" id="email" placeholder="User
Email">
            </fieldset>
            <fieldset>
                <legend><span class="number">3</span> Additional Info</legend>
                <label>
                    Time In:
                    <input type="time" name="timein" id="timein">
                    <input type="radio" name="gender" class="gender"
value="Female">Female
                    <input type="radio" name="gender" class="gender" value="Male"
checked="checked">Male
                </label>
            </fieldset>
            <button type="button" name="user_add" class="user_add">Add User</button>
            <button type="button" name="user_upd" class="user_upd">Update
User</button>
            <button type="button" name="user_rmo" class="user_rmo">Remove
User</button>

```

```

        </form>
    </div>

    <div class="section">
    <!--User table-->
        <div class="tbl-header slideInRight animated">
            <table cellpadding="0" cellspacing="0" border="0">
                <thead>
                    <tr>
                        <th>Finger.ID</th>
                        <th>Name</th>
                        <th>Gender</th>
                        <th>S.No</th>
                        <th>Date</th>
                        <th>Time in</th>
                    </tr>
                </thead>
            </table>
        </div>
        <div class="tbl-content slideInRight animated">
            <table cellpadding="0" cellspacing="0" border="0">
                <div id="manage_users"></div>
            </table>
        </div>
    </div>

</main>
</body>
</html>

```

e. **users_log_up.php** - PHP that gets in-time, out-time and date of the user, records, and keeps track of the users with a log.

```

<table cellpadding="0" cellspacing="0" border="0">
<tbody>
<?php

    session_start();
    //Connect to database
    require'connectDB.php';

    if (isset($_POST['log_date'])) {
        if ($_POST['date_sel'] != 0) {
            $_SESSION['seldate'] = $_POST['date_sel'];
        }
        else{
            $_SESSION['seldate'] = date("Y-m-d");
        }
    }

    if ($_POST['select_date'] == 1) {
        $_SESSION['seldate'] = date("Y-m-d");
    }
    else if ($_POST['select_date'] == 0) {
        $seldate = $_SESSION['seldate'];
    }

    $sql = "SELECT * FROM users_logs WHERE checkindate='$seldate' ORDER BY id DESC";
    $result = mysqli_stmt_init($conn);

```

```

if (!mysqli_stmt_prepare($result, $sql)) {
    echo '<p class="error">SQL Error</p>';
}
else{
    mysqli_stmt_execute($result);
    $resultl = mysqli_stmt_get_result($result);
    if (mysqli_num_rows($resultl) > 0){
        while ($row = mysqli_fetch_assoc($resultl)){
            ?>
                <TR>
                <TD><?php echo $row['id'];?></TD>
                <TD><?php echo $row['username'];?></TD>
                <TD><?php echo $row['serialnumber'];?></TD>
                <TD><?php echo $row['fingerprint_id'];?></TD>
                <TD><?php echo $row['checkindate'];?></TD>
                <TD><?php echo $row['timein'];?></TD>
                <TD><?php echo $row['timeout'];?></TD>
                </TR>
            <?php
            }
        }
    }
    ?>
</tbody>
</table>

```

f. **UsersLog.php** - PHP that has a list of all the users and history of all the users.

```

<!DOCTYPE html>
<html>
<head>
    <title>Users Logs</title>
    <link rel="stylesheet" type="text/css" href="userslog.css">
    <script>
        $(window).on("load resize ", function() {
            var scrollWidth = $('tbl-content').width() - $('tbl-content table').width();
            $('tbl-header').css({'padding-right':scrollWidth});
        }).resize();
    </script>
    <script src="https://code.jquery.com/jquery-3.3.1.js"
        integrity="sha1256-2Kok7MbOyxpgUVvAk/HJ2jigOSYS2auK4Pfzbm7uH60="
        crossorigin="anonymous">
    </script>
    <script src="js/jquery-2.2.3.min.js"></script>
    <script src="js/user_log.js"></script>
    <script>
        $(document).ready(function(){
            $.ajax({
                url: "user_log_up.php",
                type: 'POST',
                data: {
                    'select_date': 1,
                }
            });
            setInterval(function(){
                $.ajax({
                    url: "user_log_up.php",

```

```

    type: 'POST',
    data: {
        'select_date': 0,
    }
}).done(function(data) {
    $('#userslog').html(data);
});
},5000);
});
</script>
</head>
<body>
<?php include'header.php'; ?>
<main>
<section>
<!--User table-->
<h1 class="slideDown animated">User's Daily Log</h1>
    <div class="form-style-5 slideInDown animated">
        <form method="POST" action="Export_Excel.php">
            <input type="date" name="date_sel" id="date_sel">
            <button type="button" name="user_log" id="user_log">Select Date</button>
            <input type="submit" name="To_Excel" value="Export to Excel">
        </form>
    </div>
<div class="tbl-header slideInRight animated">
    <table cellpadding="0" cellspacing="0" border="">
        <thead>
            <tr>
                <th>ID</th>
                <th>Name</th>
                <th>Serial Number</th>
                <th>Fingerprint ID</th>
                <th>Date</th>
                <th>Time In</th>
            </tr>
        </thead>
    </table>
</div>
<div class="tbl-content slideInRight animated">
<div class="tbl-content slideInRight animated">
    <table>
        <tbody>
            <?php
                //Connect to database
                require'connectDB.php';

                $sql = "SELECT * FROM users WHERE NOT username="" ORDER BY id DESC";
                $result = mysqli_stmt_init($conn);
                if (!mysqli_stmt_prepare($result, $sql)) {
                    echo '<p class="error">SQL Error</p>';
                }
                else{
                    mysqli_stmt_execute($result);
                    $resultl = mysqli_stmt_get_result($result);
                    if (mysqli_num_rows($resultl) > 0){
                        while ($row = mysqli_fetch_assoc($resultl)){
                            ?>
                                <TR>
                                <TD><?php echo $row['id'];?></TD>
                                <TD><?php echo $row['username'];?></TD>

```

```

        <TD><?php echo $row['serialnumber'];?></TD>
        <TD><?php echo $row['fingerprint_id'];?></TD>
        <TD><?php echo $row['user_date'];?></TD>
        <TD><?php echo $row['time_in'];?></TD>
    </TR>
<?php
    }
}
}
?>
</tbody>
</table>
</div>
<div id="userslog"></div>
</div>

</section>
</main>
</body>
</html>

```

g. **connectDB** - Connects the PHP to MySQL and interfaces the whole website with NodeMCU

```

<?php
/* Database connection settings */
    $servername = "localhost";
    $username = "root";           //put your phpmyadmin username.(default is "root")
    $password = "";              //if your phpmyadmin has a password put it here.(default is "root")
    $dbname = "biometricattendace";

    $conn = mysqli_connect($servername, $username, $password, $dbname);
    if ($conn->connect_error) {
        die("Database Connection failed: " . $conn->connect_error);
    }
?>

```

h. **Export Excel.php** - PHP that exports an excel sheet of all the users.

```

<?php
//Connect to database
require'connectDB.php';

$output = "";

if(isset($_POST["To_Excel"])){

    if ( empty($_POST['date_sel'])) {

        $Log_date = date("Y-m-d");
    }
    else if ( !empty($_POST['date_sel'])) {

        $Log_date = $_POST['date_sel'];
    }
}

```

```

}
$sql = "SELECT * FROM users_logs WHERE checkindate='$Log_date' ORDER BY id DESC";
$result = mysqli_query($conn, $sql);
if($result->num_rows > 0){
    $output .= '
        <table class="table" bordered="1">
        <TR>
        <TH>ID</TH>
        <TH>Name</TH>
        <TH>Serial Number</TH>
        <TH>Fingerprint ID</TH>
        <TH>Date log</TH>
        <TH>Time In</TH>
        <TH>Time Out</TH>
        </TR>;
        while($row=$result->fetch_assoc()) {
            $output .= '
                <TR>
                <TD> '.$row['id'].'</TD>
                <TD> '.$row['username'].'</TD>
                <TD> '.$row['serialnumber'].'</TD>
                <TD> '.$row['fingerprint_id'].'</TD>
                <TD> '.$row['checkindate'].'</TD>
                <TD> '.$row['timein'].'</TD>
                <TD> '.$row['timeout'].'</TD>
                </TR>;
            }
            $output .= '</table>';
            header('Content-Type: application/xls');
            header('Content-Disposition: attachment; filename=User_Log'.$Log_date.'.xls');

            echo $output;
            exit();
        }
        else{
            header( "location: UsersLog.php" );
            exit();
        }
    }
    ?>

```

i. **getdata.php** - Gets data from the NodeMCU and sends it to MySQL

```

<?php
//Connect to database
require 'connectDB.php';

if (isset($_POST['FingerID'])) {

    $fingerID = $_POST['FingerID'];

    $sql = "SELECT * FROM users WHERE fingerprint_id=?";
    $result = mysqli_stmt_init($conn);
    if (!mysqli_stmt_prepare($result, $sql)) {
        echo "SQL_Error_Select_card";
        exit();
    }
    else{

```



```

        mysqli_stmt_bind_param($result, "s", $fingerID);
        mysqli_stmt_execute($result);
        $resultl = mysqli_stmt_get_result($result);
        if ($row = mysqli_fetch_assoc($resultl)){
            //*****
            //An existed fingerprint has been detected for Login or Logout
            if (!empty($row['username'])){
                $Uname = $row['username'];
                $Number = $row['serialnumber'];
                $sql = "SELECT * FROM users_logs WHERE fingerprint_id=? AND checkindate=CURDATE() AND
timeout="";
                $result = mysqli_stmt_init($conn);
                if (!mysqli_stmt_prepare($result, $sql)) {
                    echo "SQL_Error_Select_logs";
                    exit();
                }
            }
            else{
                mysqli_stmt_bind_param($result, "i", $fingerID);
                mysqli_stmt_execute($result);
                $resultl = mysqli_stmt_get_result($result);
                //*****
                //Login
                if (!$row = mysqli_fetch_assoc($resultl)){

                    $sql = "INSERT INTO users_logs (username, serialnumber, fingerprint_id, checkindate,
timein, timeout) VALUES (?, ?, ?, CURDATE(), CURTIME(), ?)";
                    $result = mysqli_stmt_init($conn);
                    if (!mysqli_stmt_prepare($result, $sql)) {
                        echo "SQL_Error_Select_login1";
                        exit();
                    }
                }
                else{
                    $timeout = "";
                    mysqli_stmt_bind_param($result, "sdis", $Uname, $Number, $fingerID, $timeout);
                    mysqli_stmt_execute($result);

                    echo "login".$Uname;
                    exit();
                }
            }
            //*****
            //Logout
            else{
                $sql="UPDATE users_logs SET timeout=CURTIME() WHERE fingerprint_id=? AND
checkindate=CURDATE()";
                $result = mysqli_stmt_init($conn);
                if (!mysqli_stmt_prepare($result, $sql)) {
                    echo "SQL_Error_insert_logout1";
                    exit();
                }
            }
            else{
                mysqli_stmt_bind_param($result, "i", $fingerID);
                mysqli_stmt_execute($result);

                echo "logout".$Uname;
                exit();
            }
        }
    }
}
}

```

```

//*****
//An available Fingerprint has been detected
else{
    $sql = "SELECT fingerprint_select FROM users WHERE fingerprint_select=1";
    $result = mysqli_stmt_init($conn);
    if (!mysqli_stmt_prepare($result, $sql)) {
        echo "SQL_Error_Select";
        exit();
    }
    else{
        mysqli_stmt_execute($result);
        $resultl = mysqli_stmt_get_result($result);

        if ($row = mysqli_fetch_assoc($resultl)) {
            $sql="UPDATE users SET fingerprint_select=0";
            $result = mysqli_stmt_init($conn);
            if (!mysqli_stmt_prepare($result, $sql)) {
                echo "SQL_Error_insert";
                exit();
            }
            else{
                mysqli_stmt_execute($result);

                $sql="UPDATE users SET fingerprint_select=1 WHERE fingerprint_id=?";
                $result = mysqli_stmt_init($conn);
                if (!mysqli_stmt_prepare($result, $sql)) {
                    echo "SQL_Error_insert_An_available_card";
                    exit();
                }
                else{
                    mysqli_stmt_bind_param($result, "i", $fingerID);
                    mysqli_stmt_execute($result);

                    echo "available";
                    exit();
                }
            }
        }
    }
    else{
        $sql="UPDATE users SET fingerprint_select=1 WHERE fingerprint_id=?";
        $result = mysqli_stmt_init($conn);
        if (!mysqli_stmt_prepare($result, $sql)) {
            echo "SQL_Error_insert_An_available_card";
            exit();
        }
        else{
            mysqli_stmt_bind_param($result, "i", $finger_sel, $fingerID);
            mysqli_stmt_execute($result);

            echo "available";
            exit();
        }
    }
}
}
}
}
//*****
//New Fingerprint has been added
else{
    $Uname = "";

```

```

$Number = "";
$gender = "";

$sql = "SELECT fingerprint_select FROM users WHERE fingerprint_select=1";
$result = mysqli_stmt_init($conn);
if (!mysqli_stmt_prepare($result, $sql)) {
    echo "SQL_Error_Select";
    exit();
}
else{
    mysqli_stmt_execute($result);
    $resultl = mysqli_stmt_get_result($result);
    if ($row = mysqli_fetch_assoc($resultl)) {
        $sql = "UPDATE users SET fingerprint_select = 0";
        $result = mysqli_stmt_init($conn);
        if (!mysqli_stmt_prepare($result, $sql)) {
            echo "SQL_Error_insert";
            exit();
        }
        else{
            mysqli_stmt_execute($result);

            $sql = "INSERT INTO users (username , serialnumber, gender, fingerprint_id,
fingerprint_select) VALUES (?, ?, ?, ?, 1)";
            $result = mysqli_stmt_init($conn);
            if (!mysqli_stmt_prepare($result, $sql)) {
                echo "SQL_Error_Select_add";
                exit();
            }
            else{
                mysqli_stmt_bind_param($result, "sdsi", $Uname, $Number, $gender, $fingerID);
                mysqli_stmt_execute($result);

                echo "succesfull";
                exit();
            }
        }
    }
}
else{
    $sql = "INSERT INTO users (username , serialnumber, gender, fingerprint_id,
fingerprint_select) VALUES (?, ?, ?, ?, 1)";
    $result = mysqli_stmt_init($conn);
    if (!mysqli_stmt_prepare($result, $sql)) {
        echo "SQL_Error_Select_add";
        exit();
    }
    else{
        mysqli_stmt_bind_param($result, "sdsi", $Uname, $Number, $gender, $fingerID);
        mysqli_stmt_execute($result);

        echo "succesful2";
        exit();
    }
}
}
}
}
}
}
if (isset($_POST['Get_Fingerid'])) {

```

```

if ($_POST['Get_Fingerid'] == "get_id") {
    $sql= "SELECT fingerprint_id FROM users WHERE add_fingerid=1 AND username="";
    $result = mysqli_stmt_init($conn);
    if (!mysqli_stmt_prepare($result, $sql)) {
        echo "SQL_Error_Select";
        exit();
    }
    else{
        mysqli_stmt_execute($result);
        $resultl = mysqli_stmt_get_result($result);
        if ($row = mysqli_fetch_assoc($resultl)) {
            echo "add-id".$row['fingerprint_id'];
            exit();
        }
        else{
            echo "Nothing";
            exit();
        }
    }
}
else{
    exit();
}
}
if (!empty($_POST['confirm_id'])) {

    $fingerid = $_POST['confirm_id'];

    $sql="UPDATE users SET fingerprint_select=0 WHERE fingerprint_select=1";
    $result = mysqli_stmt_init($conn);
    if (!mysqli_stmt_prepare($result, $sql)) {
        echo "SQL_Error_Select";
        exit();
    }
    else{
        mysqli_stmt_execute($result);

        $sql="UPDATE users SET add_fingerid=0, fingerprint_select=1 WHERE fingerprint_id=?";
        $result = mysqli_stmt_init($conn);
        if (!mysqli_stmt_prepare($result, $sql)) {
            echo "SQL_Error_Select";
            exit();
        }
        else{
            mysqli_stmt_bind_param($result, "s", $fingerid);
            mysqli_stmt_execute($result);
            echo "Fingerprint has been added!";
            exit();
        }
    }
}
if (isset($_POST['DeleteID'])) {

    if ($_POST['DeleteID'] == "check") {
        $sql = "SELECT fingerprint_id FROM users WHERE del_fingerid=1";
        $result = mysqli_stmt_init($conn);
        if (!mysqli_stmt_prepare($result, $sql)) {
            echo "SQL_Error_Select";
            exit();
        }
    }
}

```

```
else{
    mysqli_stmt_execute($result);
    $resultl = mysqli_stmt_get_result($result);
    if ($row = mysqli_fetch_assoc($resultl)) {

        echo "del-id".$row['fingerprint_id'];

        $sql = "DELETE FROM users WHERE del_fingerid=1";
        $result = mysqli_stmt_init($conn);
        if (!mysqli_stmt_prepare($result, $sql)) {
            echo "SQL_Error_delete";
            exit();
        }
        else{
            mysqli_stmt_execute($result);
            exit();
        }
    }
    else{
        echo "nothing";
        exit();
    }
}
}
else{
    exit();
}
}
?>
```

CHAPTER IV

RESULT & ANALYSIS

4.1 SCHEMATIC & PCB OF BIOMETRIC FINGERPRINT ATTENDANCE SYSTEM

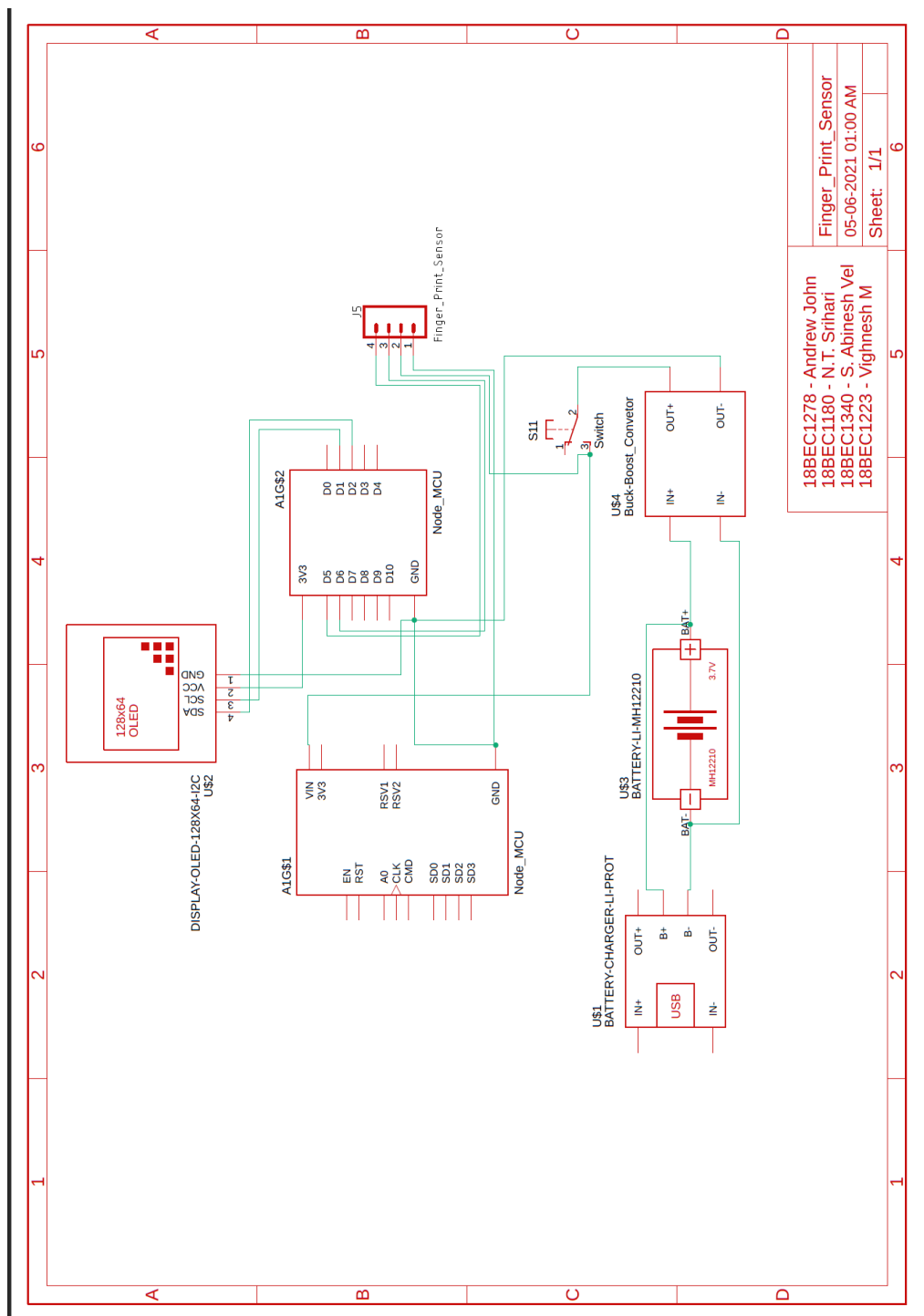


Fig - 13: Schematic Design of Biometric Fingerprint Attendance System

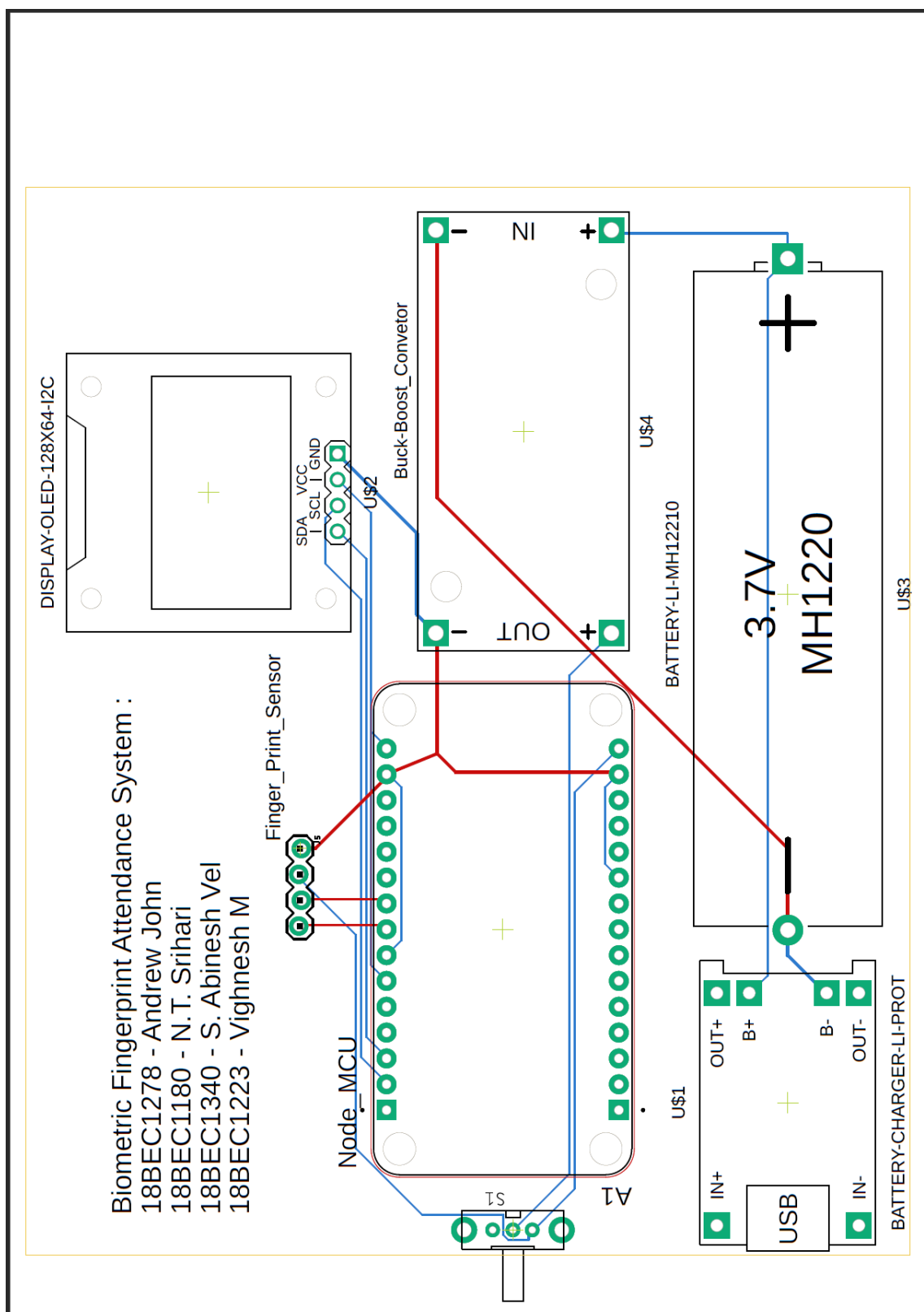


Fig - 14: PCB Design of Biometric Fingerprint Attendance System

4.2 REAL TIME IMPLEMENTATION OF BIOMETRIC FINGERPRINT ATTENDANCE SYSTEM

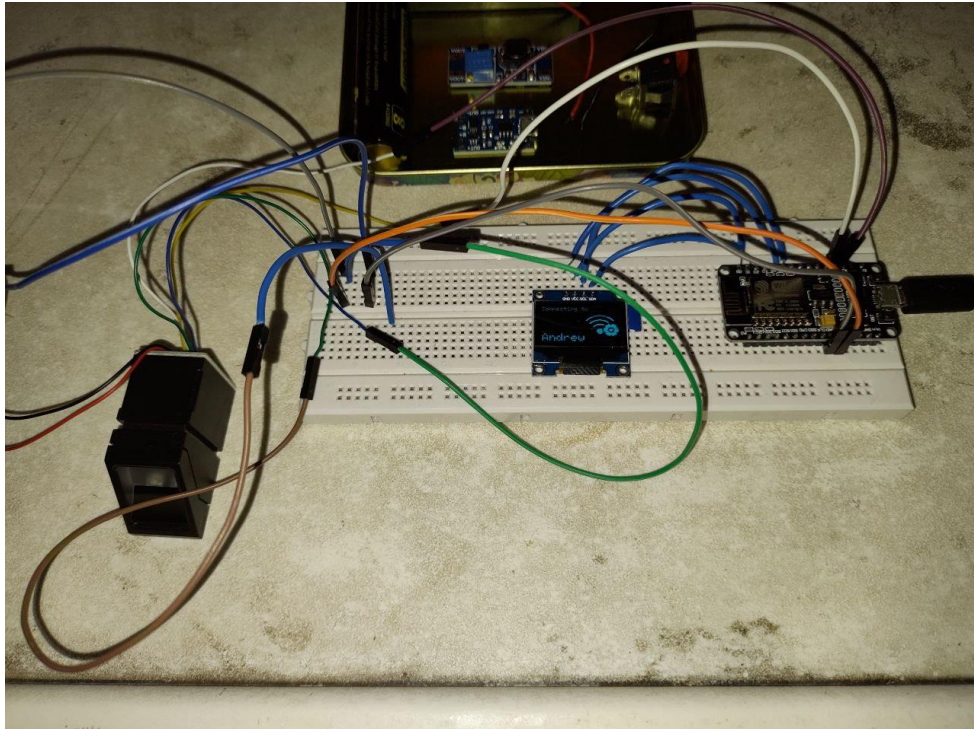


Fig - 15: Scanning WiFi for Connection

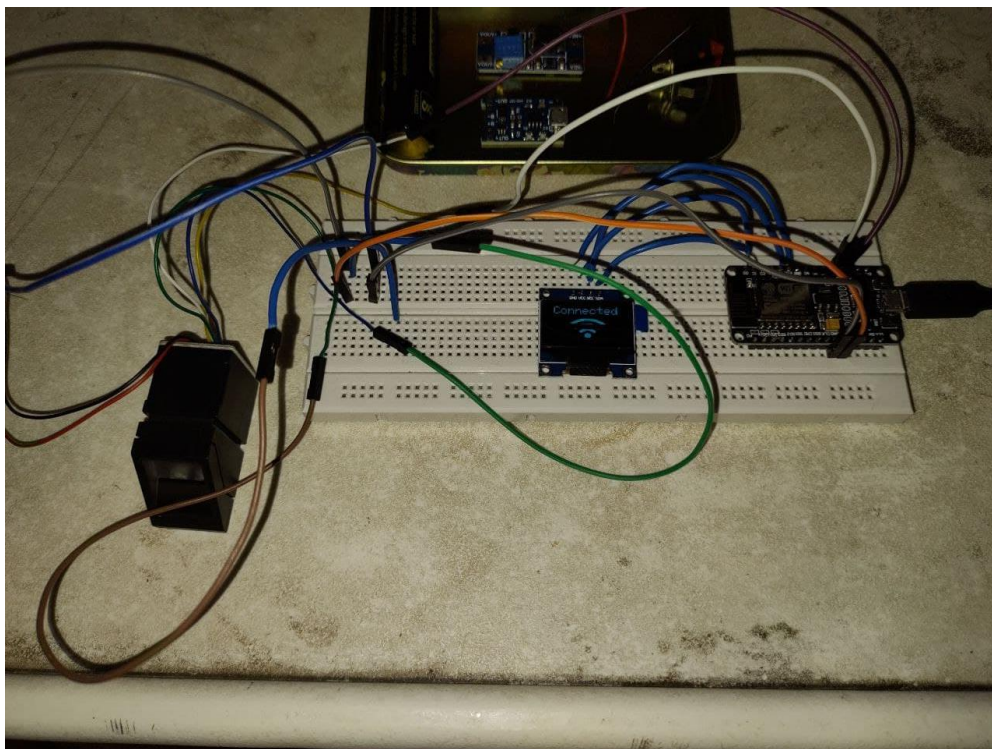


Fig - 16: WiFi Connected Successfully

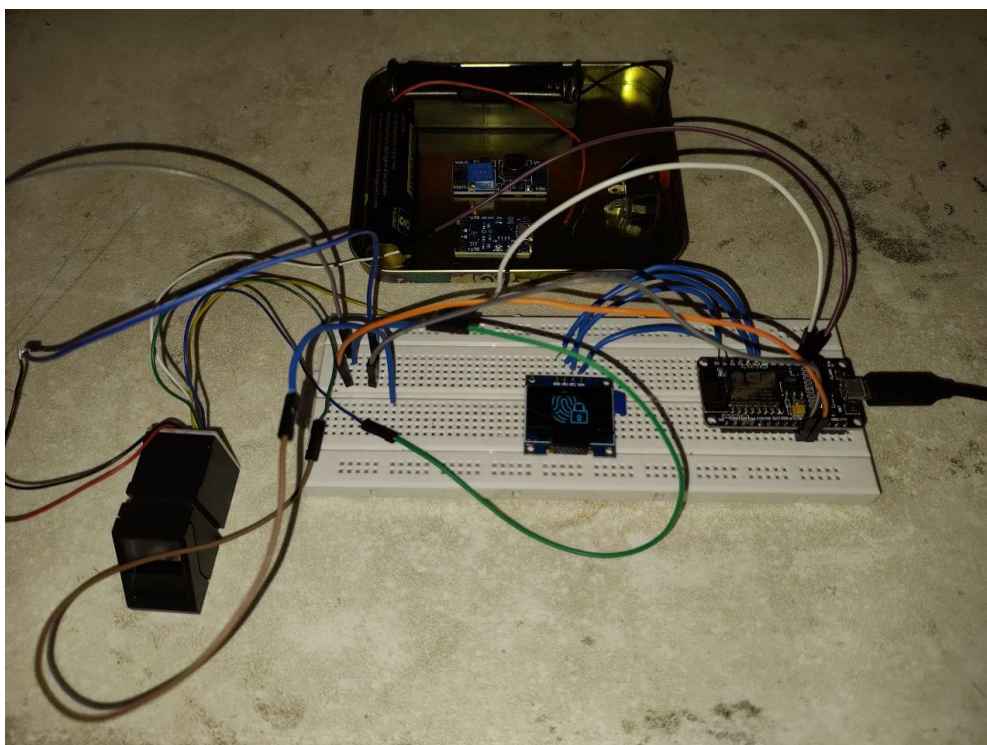


Fig - 17: Fingerprint Sensor is Locked for Authentication

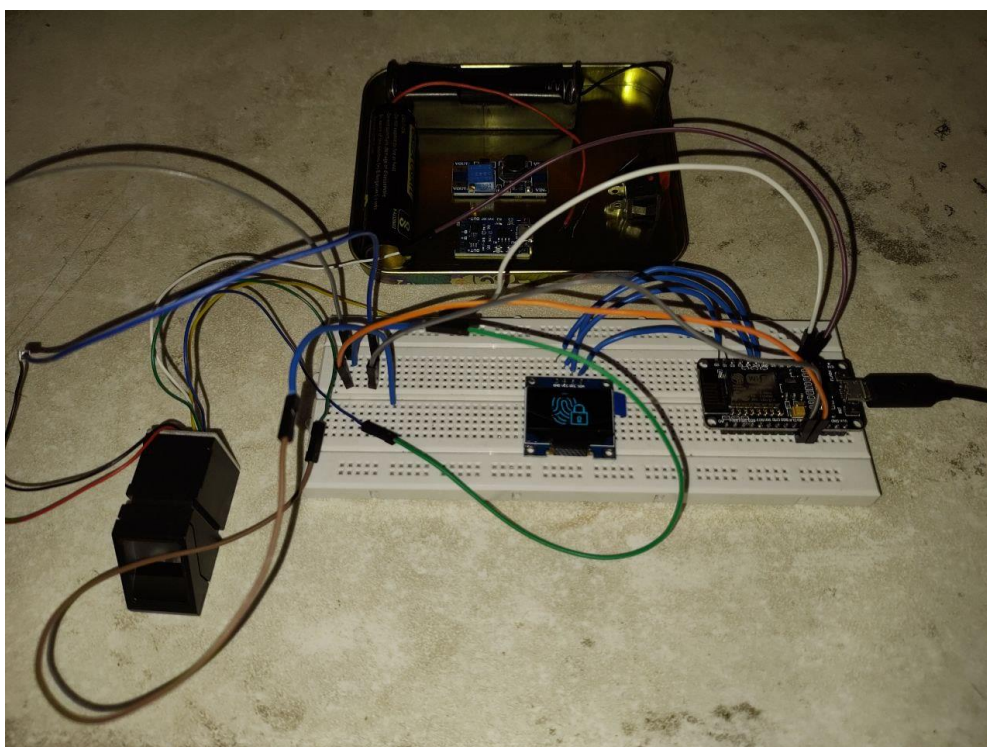


Fig - 18: Real Time Hardware Implementation with MT3608 Boost Converter and TP4056 Li-ion Battery Charger

4.3 SNAPSHOTS OF THE DESIGNED WEBSITE

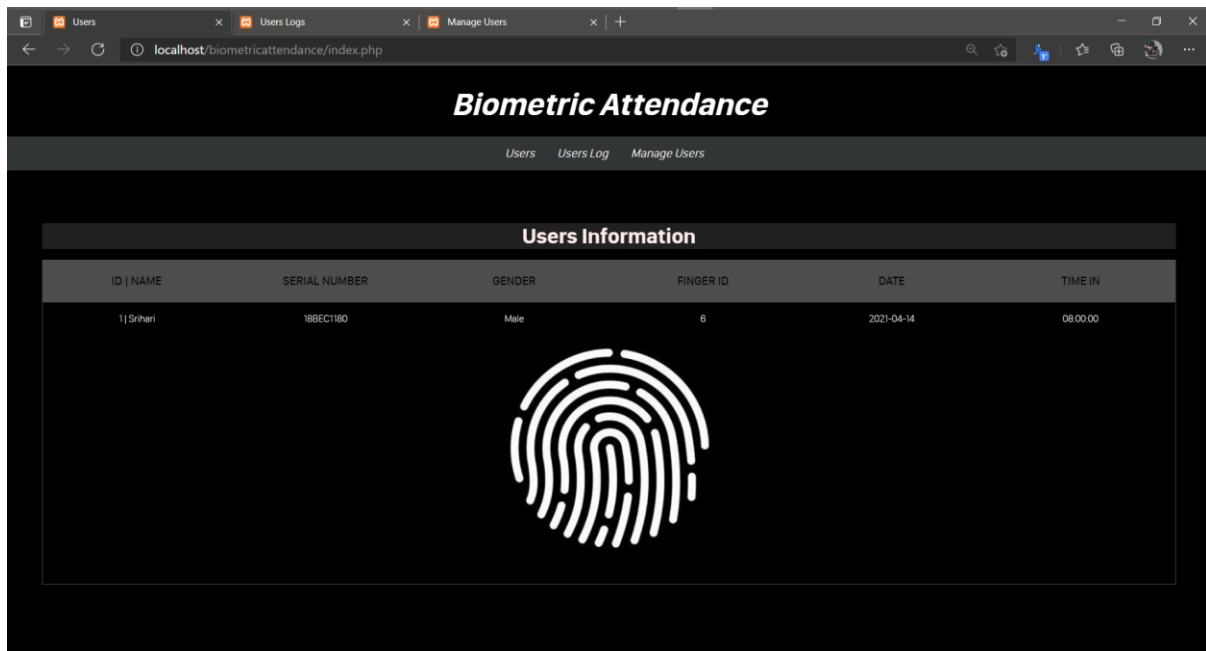


Fig - 19: Users Information

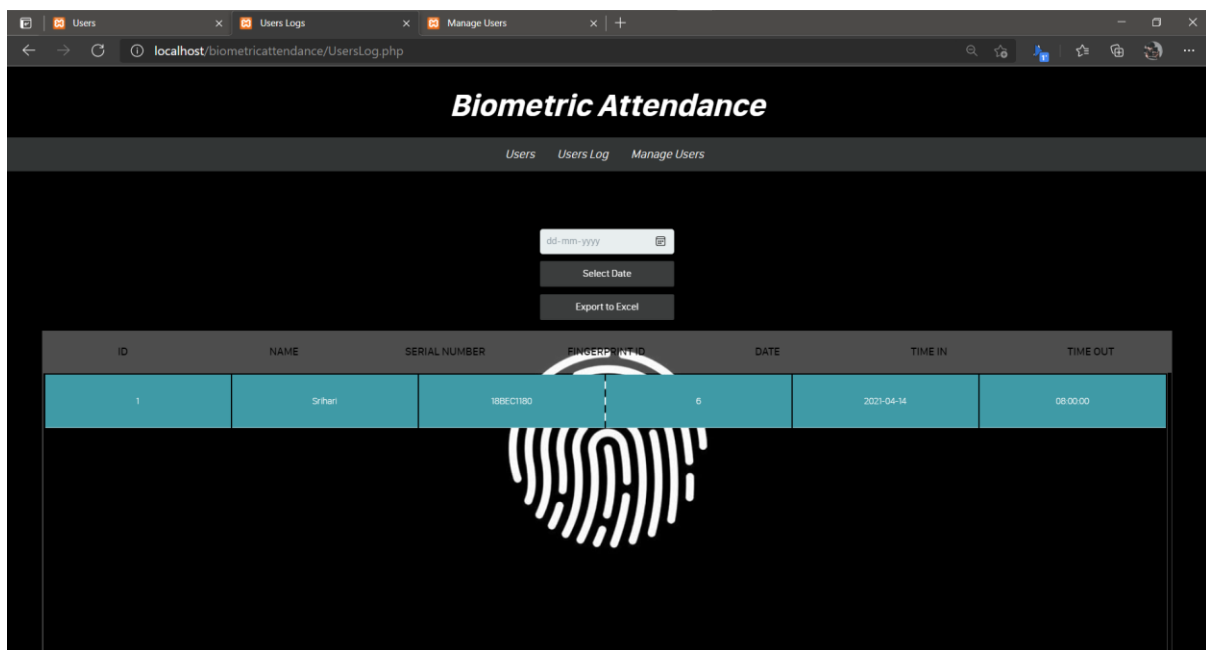


Fig - 20: Users Log

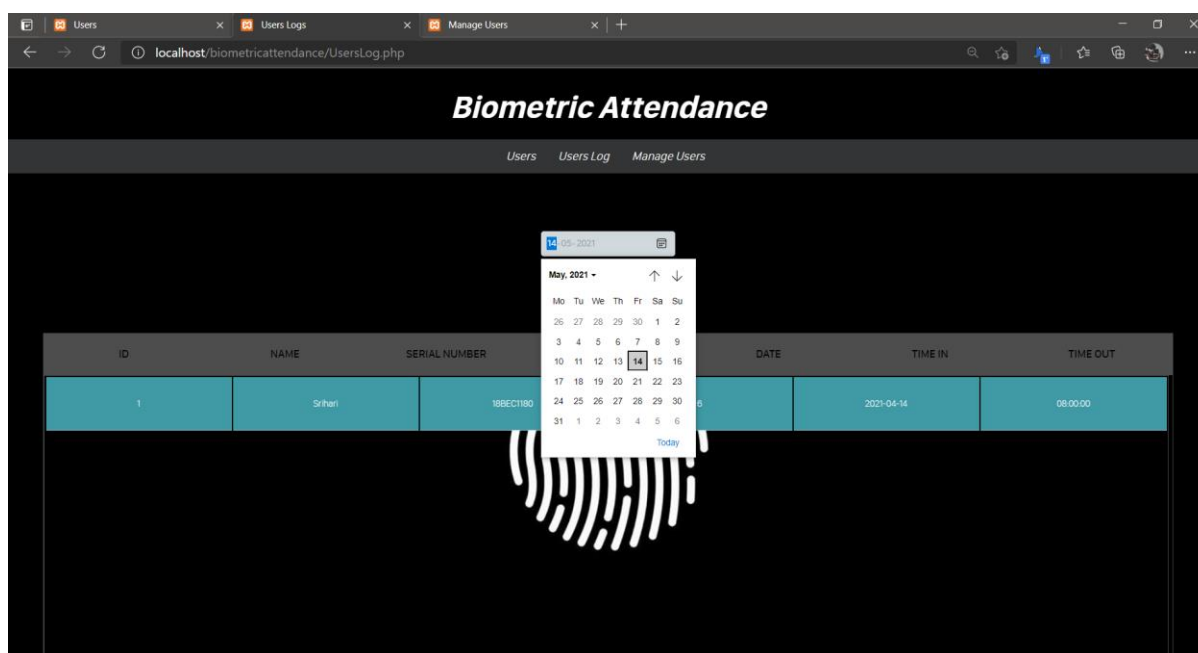


Fig - 21: Users Date Search

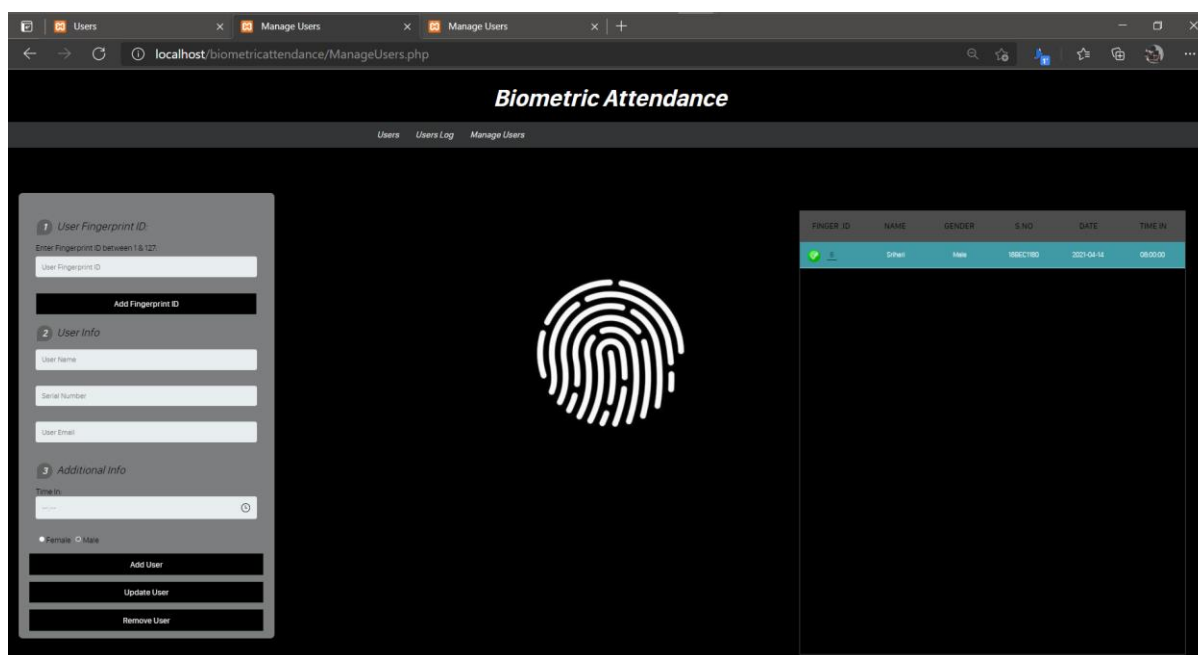


Fig - 22 Manage Users

CHAPTER V

CONCLUSION

5.1 CONCLUSION

The Biometric Attendance System was designed to function as a portable system which can record the attendance on any given location on any given day inside the campus by recording the necessary information and also storing it in a safe and secure database in a website which can later be accessed for clarifications.

It is designed to be highly easy to operate for the users and fully in sync with the company/organization ideas and records by adhering to the safety standards necessary for withholding that the information provided by the system is true and cannot be forged whatsoever. So with this, the organization can monitor the attendance systems properly and have no probability of being cheated by the employees whatsoever. So we finally conclude that the proposed system is therefore highly efficient and highly secure with ease of use.

5.2 FUTURE SCOPE

Attendance systems have always been a difficult task to work around with, especially with the creative ways people find to forge the entries as well, this could prove costly, especially to a newly starting organisation as it could be simply paying the employees without even noticing if they are inside the workplace. To eliminate this, the Biometric system is best suited, as it can never be forged or misused and is unique to every single human being. Moreover, Biometric systems usually get complex with the involvement of various processes, so we have made a system that will eliminate the complexity and make it easier to use for both the users and also the stakeholders.

And usually these systems are just static and have no option to be moved around, however the project we have come up with has the ability to be moved around, therefore in case of an emergency such as another Attendance system failing, one of these systems could replace that and still function the same. And this is possible because all the information is safely transferred to the database which is in control of the organization thereby maintaining uniformity.

With this we also aim to be eco-friendly as paper consumption by a particular organization, be it school or an office could come down by a large amount as in

small scale schools, Attendance is usually marked in notebooks made of paper, with this system introduced, it would reduce the usage of paper since all of the records is digitalized, thereby incorporating the idea of digital India into it as well.

5.3 COST ANALYSIS

NodeMCU -	₹ 350
R307 Fingerprint sensor -	₹ 1500
0.96" OLED Display -	₹ 250
Li-Ion Battery (18650) -	₹ 150
TP4056 Li-Ion Battery Charger -	₹ 75
MT3608 Boost Converter -	₹ 75
On/Off switch -	₹ 15
Breadboard -	₹ 100
Connecting Wires -	₹ 150
Labour -	₹500
TOTAL =	₹3165 + <u>GST%</u>

COMPLETE DEMO AND PRESENTATION OF THE PROJECT :

Biometric Fingerprint Attendance System | ECE 4003 - Embedded System Design | FINAL PROJECT REVIEW : <https://youtu.be/x5v1AYyIYI0>

REFERENCES

<https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet>

https://www.rhydolabz.com/miscellaneous-other-widgets-c-205_124/fingerprint-sensor-r305-ttl-uart-p-1085.html

<https://www.electroniccomp.com/0.96-inch-i2c-iic-128x64-oled-display-module-4-pin-white-color>

<https://www.winstar.com.tw/products/oled-module/graphic-oled-display/4-pin-oled.html>

<https://components101.com/batteries/18650-lithium-cell>

<https://components101.com/modules/tp4056a-li-ion-battery-chargingdischarging-module>

<https://components101.com/modules/xl6009-dc-dc-step-down-voltage-regulator-module>

<https://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/readme.html>

<https://techtutorialsx.com/2016/07/17/esp8266-http-get-requests/>

<https://learn.adafruit.com/adafruit-gfx-graphics-library>

<https://www.arduino.cc/reference/en/libraries/adafruit-fingerprint-sensor-library/>

<https://www.arduino.cc/>

<https://www.w3schools.com/>

<https://www.google.co.in/>

BIODATA



Name : Andrew John
 Mobile Number : 90037 35112
 E-mail : andrewjohn.j2018@vitstudent.ac.in
 Permanent Address : E405, Radiance Mandarin 200 Feet Thoraipakkam -
 Pallavaram Radial Road Thoraipakkam,
 Chennai - 600097



Name : N. T. Srihari
 Mobile Number : 98405 22487
 E-mail : nt.srihari2018@vitstudent.ac.in
 Permanent Address : F115, Unity Enclave, Medavakkam Road,
 Mambakkam, Chennai - 600127



Name : S. Abinesh Vel
 Mobile Number : 75501 82575
 E-mail : sabinesh.vel@vitstudent.ac.in
 Permanent Address : No 1775, 2nd main road, Poompuhar Nagar,
 Kolathur, Chennai - 600099



Name : Vighnesh M
 Mobile Number : 81481 66514
 E-mail : vighnesh.m2018@vitstudent.ac.in
 Permanent Address : 4A/25, Andavar Street, Choolaimedu, Chennai-94

THIS PAGE IS LEFT BLANK INTENTIONALLY.