Biometric Attendance System using Fingerprints and Featuring IoT

A project report submitted in partial fulfillment of
the requirements for the degree of

Bachelor of Engineering

by

Aditya Kelkar (Roll No. 7579)
Sebastian Chennattu (Roll No. 7564)
Aaron Anthony (Roll No. 7554)

Under the guidance of
Prof. Sushma Nagdeote

DEPARTMENT OF ELECTRONICS ENGINEERING
Fr. Conceicao Rodrigues College of Engineering, Bandra (W), Mumbai - 400050
University of Mumbai
April 16, 2019

We dedicate this work to our family and friends.

We are grateful for having their constant motivation and support.

Internal Approval Sheet

CERTIFICATE

This is to certify that the project entitled **"Biometric Attendance System using Fingerprints and Featuring IoT"** is a bonafide work of **Aditya Kelkar(7579), Sebastian Chennattu(7564), Aaron Anthony(7554)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **Bachelor Of Engineering** in **Electronics.**

Sushma Nagdeote
Supervisor/Guide

Dr. Deepak V. Bhoir
Head of Department

Dr.Srija Unnikrishnan
Principal

Approval Sheet

Project Report Approval

This project report entitled by **Biometric Attendance System using Fingerprints and Featuring IoT** by **Aditya Kelkar, Sebastian Chennattu, Aaron Anthony** is approved for the degree of Bachelor of Engineering

Examiners

1.———————————

2.———————————

Date:

Place:

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Aditya Kelkar (Roll No. 7579) **(sign)**_____

Sebastian Chennattu (Roll No. 7564) **(sign)**_____

Aaron Anthony (Roll No. 7554) **(sign)**_____

Date: April 16, 2019

Abstract

The project aims at building a biometrics based attendance registry system for educational institutions such as colleges. The project brings together concepts of hardware and software to give an end consumer product which would be an entire system targeted at replacing the current manner of attendance marking. The project also utilizes the concepts of Internet of Things (IoT), primarily for data transfer. The system has been broadly divided into the hardware and software aspects. The hardware section has two subsections which are individual devices, namely the pod and the dock. The pod is a portable device which is to be circulated in the classroom among students. It houses a fingerprint sensor which scans the fingerprint kept on it while also identifying the student on the basis of the scanned fingerprint. The identified student's ID is then stored on onboard memory and is ready for transfer. The dock acts as a charging station. The charging process also sends a trigger to the microcontroller in the dock which initiates data transfer via the ESP8266 Wifi module. The data is sent to the database on a web server where the Database Management System (DBMS) updates the attendance tables accordingly. The database can be downloaded by the respective faculty in an excel supported format. Besides, the class teacher, the concerned HOD and the principal can also access the database. There is also a provision planned to allow the class teachers and HOD to alter the attendance entries if required.

Furthermore, an alerting system is also planned to be incorporated which would send a notification (as an e-mail or text message) to the class teacher, mentor, students and parents/guardians in case of attendance default.

Acknowledgments

We take great pleasure in presenting this report on **Biometric Attendance System using fingerprints featuring IoT**. We take this opportunity to express our sincerest thanks towards our project guide Prof. Sushma Nagdeote, Fr. C.R.C.E, Bandra (W), Mumbai, for providing the technical guidelines, and the suggestions pertaining to this project and also for always having our backs. We enjoyed the the entire process and discussing the work progress with her during our interactions.

We thank Dr.D.V.Bhoir, Head of Electronics Dept., Dr.S.Unnikrishnan, the Principal and the management of C.R.C.E., Mumbai for their encouragement and providing necessary infrastructure for pursuing the project.

We also thank all non-teaching staff for their valuable support in completing our project.

Aditya Kelkar(Roll No. 7579)

Sebastian Chennattu(Roll No. 7564)

Aaron Anthony(Roll No. 7554)

Date: April 16, 2019

Contents

List of Figures

Chapter 1

Introduction

Marking of attendance is one of the regular activities that happen in almost every institute, educational or non-educational, every day. The current system of checking attendance in majority of the colleges involves the respective professor taking a roll call or passing an attendance sheet. The professor then proceeds to upload the data onto an excel file or an online database by using information from the sheet used before. While this method is being used for a long time now, it still faces a few basic problems which could be easily eliminated with the help of technology. Primarily, the issue is of proxy attendances being marked. Secondly, the human involvement in the form of professors opens up an avenue for human error. Thirdly, it takes up a considerable time of the professors which could be used towards other productive tasks.

We therefore propose a system wherein the entire attendance marking and maintenance procedure is done by a smart device instead of relying on the age old pen and paper technique. The system makes efficient use of hardware and software principles and also features connectivity in the form of IoT. The end result is a system which identifies the students, maintains an online attendance register in the form of a database and provides the professors with a detailed attendance record of any pupil, on demand.

1.1 Background

The inspiration for this project stems from our daily experience in college regarding the attendance marking process. The traditional pen and paper methods which involve circulation of an attendance sheet, or roll call by teachers, suffer from a few fundamental drawbacks. The former infamously suffers from proxy entries while the later wastes time that educators rarely have considering the already high load of curriculum completion.

Both methods further require the teacher to manually enter the attendance into either an online or offline register. This again wastes time which could rather be used for other tasks. Observations of these problems before us led us towards the idea of this project.

1.2 Origin

The idea takes inspiration from the biometric attendance systems already in place and improves on them. Fingerprint sensors are primarily used in these systems which are used in offices, colleges, etc. Just like our system, they also rely on optical fingerprint scanners to take an image of the finger placed on the sensor and then compare them with a database of fingerprints. However, these are bulky, static devices which lack ease of use. We also ride on the IoT wave gripping the world and present a new way of entering the attendance entries onto a centralised server. Although there have been attempts at making projects similar to ours, they largely focused on uploading of the attendance data onto a cloud based server. None of them focused on providing mobility which directly relates to ease-of-use.

1.3 Need

- It has become very inconvenient to manually collate student attendance for each lecture and check if their attendance is till the required mark.
- Due to the academic rule that a student must attend at least 75% lectures any kind of impersonation must be completely eliminated in the lecture, this can be done by using a biometric system for attendance logging.
- Teachers are already entrusted with multiple academic and co-curricular responsibilities. Attendance monitoring, a monotonous and mechanical task would be eliminated, thus freeing up time for the teacher to concentrate on other more important tasks.

- Reduce paper consumption thereby improving on the environmental impact of the institute.

Chapter 2

Literature Review

The referred papers and our observations have been compiled in fig 2.1 and 2.2:

Reference number:	Title of the Paper:	Outcomes discussed:	Result:	Research Gap
1) ISSN(Online): 2347-2820	Attendance Monitoring System Automation Using Fingerprint Module[1]	i)This paper discusses the usage of an arm7 based processor which is coupled along with an LCD screen and a finger print sensor which is interfaced along with a PC using a JAVA API. ii)The author concludes that the system can completely eliminate all problems associated with modern day attendance.	i)The authors have created a system that collects fingerprint images stores them into a database and matches the attendance according to the received fingerprint. ii) They have an LCD screen which displays some response of the system.	i) This project uses the serial communication ports of the arm7 processor to communicate with the JAVA api that is present on the computer. ii)This approach may be good for representing data but it makes the entire system bulky. iii)Fingerprint data stored on a database is a security hazard. According to current standards fingerprint data is stored locally on the fingerprint sensor.
2) ISSN: 2321-9939	Attendance System Using Fingerprint Identification with GUI[2]	i) Main aim of this method is to develop transparent attendance system and keep real time data and display online data for parents and other academic use. ii)Attendance System Using Fingerprint Identification with GUI is reliable and easy to implement which gives us accurate results.	i)The authors have successfully created a system using an 8051 microcontroller, a fingerprint scanner and a graphic user interface that stores user finger print data into a database and compares it to received fingerprints during the time of attendance. ii)The stored attendance and other details like student present ,absent can be accessed on a web page	i)This project similar to the previous project possesses a great security hazard by storing the fingerprint data in a different location. ii)This system also requires a direct connection to a computer for transferring data of fingerprint to the database. iii)Connection to the PC is also required during operation.

Figure 2.1: Literature Review

Reference number:	Title of the Paper:	Outcomes discussed:	Result:	Research Gap
3)NIGERCON.2013. 6715633.	Biometric-based attendance system with remote real-time monitoring for tertiary institutions in developing countries.[3]	i) The proposed system is made up of the data capture system (enrolment system), the check-in/check-out system (Identification and Verification system) and the web-based reporting and monitoring system. ii) The system will also generate a report concerning the percentage attendance of students for the courses at the end of the semester.	i)The authors have created a biometric attendance system using three fundamental technologies namely Fingerprint sensor , web- apps and databases. ii) The authors have designed a web portal for accessing the student and teacher attendance.	i)The entire attendance system needs to be attached to a computer for storing data and collecting attendance.
4) ISSN (Online): 2278 – 8875	Fingerprint Based Attendance System Using Microcontroller and LabView[4]	i) This project is implemented using an 8051-microcontroller interfaced with a fingerprint sensor along with databases on LABVIEW. ii)LABVIEW stores database of students with name, date, in-time and out time .	i)The authors were successfully able to implement and store fingerprint data into a database created using LABVIEW. ii)The authors were also able to calculate the time saved by using their system per student approximately 56 seconds.	i)Although the system provides an efficient method to store and display attendance records, it has to always be attached to the pc to store the data.

Figure 2.2: Literature Review

Chapter 3

Problem Statement and Solution

3.1 Problem statement

"To create an attendance registry system which would use biometric parameters to take attendance, then upload the results onto a central database, and provide the data in a suitable format to the end user."

The problem statement for our project can be broken down into three parts which form our objectives. These have been discussed in the following chapter.

3.2 Proposed solution to the problem

The current project focuses on an economical, power efficient, compact and ergonomic prototype which would simulate the entire attendance monitoring procedure that we have envisioned. We aim to project the advantages of this system and encourage a possible shift towards it.

Chapter 4

Project Description

4.1 Block Diagram

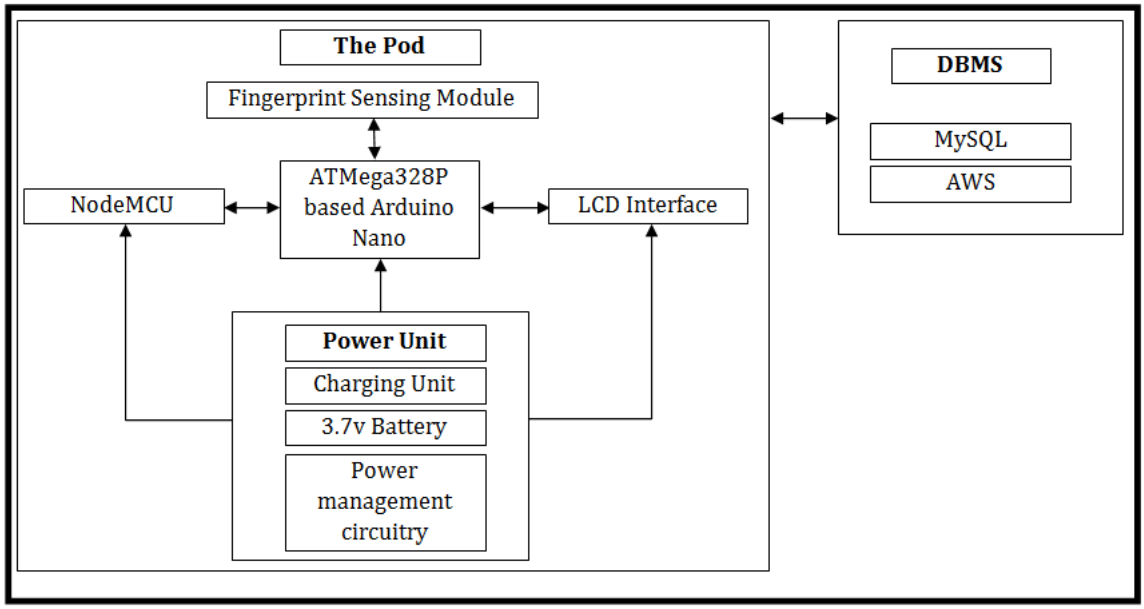

Figure 4.1: Block Diagram

4.2 Objective

The objectives of the project are as follows:

- To successfully identify the students by means of using their fingerprints
- To successfully update the database with the attendance entries using IoT
- To successfully provide a user friendly environment to access the data

4.3 Components

4.3.1 Arduino Nano

Arduino Nano[5] is a development board based on the AtMega328p microcontroller which is mounted as a MLF package IC. It features 32KB flash memory, 2KB SRAM and 1KB EEPROM.It has been developed on the AVR architecture with a clock frequency of 16MHz. It allows for an input voltage range of 7-12V DC and operates on 5V. It has 22 digital i/o pins of which 6 can be used for PWM. It supports Serial, I2C and SPI communication protocols, and can be coded using the Arduino IDE software. It has a small dimension (18x45 mm), has a current consumption of 19mA and weighs a modest 7g.

Figure 4.2: Arduino Nano

4.3.2 Fingerprint sensor GT-511C3

The GT-511C3[6] is an optical fingerprint recognition module which relies on image comparison for identification of fingerprints. It has high speed and accuracy and supports 1:1 and 1:N verification. It supports simple UART and USB communication protocols. It features an ARM Cortex M3 core as the CPU and can store upto 200 fingerprints. It operates on 3.3-6V DC and has an operating current of less than 130mA. It works on simple serial commands from the master device which involves transfer of hexadecimal codes denoting the actions and responses. It has an identification period of less than a second and can enroll a new entry in less than 3 seconds.

Figure 4.3: GT-511C3 fingerprint recognition module

4.3.3 NodeMCU

The ESP8266[7] is a low cost WiFi microchip with full TCP/IP stack and microcontroller capability manufactured by Espressif Systems[7].It features a 32-bit RISC microprocessor core running at 80MHz. It has 16 GPIO pins and supports I2C, SPI, UART interfaces. It also supports IEEE 802.11 b/g/n WiFi and allows WEP or WPA/WPA2 security authentication or open networks. It is capable of handling analog inputs using an on-board 10bit successive approximation ADC. There are many open source SDKs available such as NodeMCU, Mongoose OS, etc. Of these NodeMCU is widely used. NodeMCU is an open source IoT platform.

Figure 4.4: NodeMCU

4.3.4 I2C compatible LCD screen- 16x2

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications.LCD's are economical, easily programmable, have no limitation of displaying special and even custom characters (unlike in seven segments), animations and so on. The model used in this project also supports I2C which eliminates the need of having multiple connections between the screen and the microcontroller.

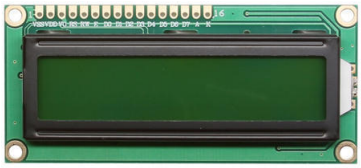

Figure 4.5: LCD-16x2

4.3.5 3.7V LiPo Cell

A 3.7v 1100mAh LiPo cell is the power unit for the device. The selection of the Lipo battery was done on the basis of the average current drawn by the individual devices and to ensure that the device lasted throughout the entire day. Some of the calculations for average current drawn have been shown in figure 4.7.

This will give an average of about 6 hours of usage with a 1100mAh battery, since we have tried to reduce power consumption by turning off or putting microcontrollers that are not always in use into sleep mode.

DEVICES	AVERAGE CURRENT DRAWN
Arduino Nano	33.4 mA
Nodemcu	80mA
Fingerprint Sensor (GT511C3)	100mA
L.C.D display	30mA
Total:	234.4mA

Figure 4.6: Approximate power consumption of all the components

Figure 4.7: Approximate power consumption of all the components

4.3.6 Power Management IC (TP-4056)

The TP4056[8] is a complete constant-current/constant-voltage linear charger for single cell lithium-ion batteries and lithium-polymer batteries. Its SOP package and low external component count make the TP4056 suited for portable applications. Thermal feedback regulates the charge current to limit the die temperature during high power operation or high ambient temperature this feature has not been implement in this project as it is a relatively low power operation. The charge voltage is fixed at 4.2V, and the charge current can be programmed externally with a single resistor. Other features include current monitor, under voltage lockout, automatic recharge and two status pins to indicate charge termination and the presence of an input voltage.

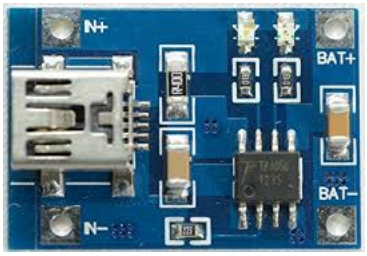

Figure 4.8: IC TP4056

4.3.7 Protection IC (FS321F)

The FS312F-G[9] battery protection IC is designed to protect lithium-polymer battery from damage or degradation from overcharge, over discharge and overcurrent. It has an extremely accurate overcharging detection voltage that ensures safe and full utilization charging. It also offers very low standby current drains from the cell while in storage that is it has an Ultra-Low Quiescent Current at 3μA. It also offers an Ultra-Low Power-Down Current at 0.1A.

4.3.8 Boost Converter (B6287M)

The B6287T[10] is a fixed frequency, current mode boost converter with an operating frequency of up to 1.2MHz allowing the external inductors and capacitors to be selected in smaller sizes.The built-in soft-start feature reduces the start-up surge current and automatically switches to PFM mode at light load. Includes input undervoltage lockout, current limit, and over-temperature protection. The small size of the package saves more space on the PCB It has an input voltage range from 2V to 24V and can provide up to 28V with 97% efficiency. The boost converter is used to supply current to the NodeMCU, Finger Print Sensor unit and the LCD screen.

Figure 4.9: B6287M Boost Converter

Chapter 5

Features

5.1 Choice of Fingerprint Sensor

Fingerprint sensors are sensors that sample the pattern of a person's fingerprint and then compare it with a pre-defined template to find a match. Depending on how the sensor takes the sample, the sensors can be broadly classified into 4 types-

- Optical Sensors
- Capacitive Sensors
- Ultrasonic Sensors
- Thermal Line Sensors

Optical Fingerprint Sensors: Optical fingerprint sensors take a digital image of the fingerprint as sample template. The image is then converted to a code which is a sequence specifying the pixel values. This code is then compare with the database of codes from pre-defined fingerprints to find a match.[11]

Capacitive Fingerprint Sensors: Capacitive or CMOS fingerprint sensors generate an electric image of the fingerprint. Capacitive scanners are hard to forge because they cannot be fooled with fingerprint images. They are more expensive than optical sensors.[11]

Ultrasonic Fingerprint Sensors: Ultrasonic scanners use very high frequency sound wave to read pattern of fingerprints. Ultrasonic sound waves reflected form the fingertip surface are measured by the sensor and fingerprint pattern image is produced. Performance of ultrasonic sensors stay unaffected by dirtiness of finger surface as it doesn't

capture image like optical sensors.[11]

Thermal Line Sensors: These sensors read a fingerprint pattern by measuring temperature variation in fingertip ridges and valleys. It requires finger to be moved over a linearly arranged narrow array of thermal sensors. They are small in size and require finger movement to measure fingerprint patterns.[11]

We use an optical sensor as it is widely available, and cheaper to use.

5.2 The Database Management System and Related Services

5.2.1 Choice of DBMS

Although database management systems all perform the same basic task, which is to enable users to create, edit and access information in databases, how they accomplish this can vary. Additionally, the features, functionality, and support associated with each management system can differ significantly. When comparing different popular databases, one should consider how user-friendly and scalable each DBMS is as well as how well it will integrate with other products one is using. Additionally, we had to take into account the cost of the management system and the support available for it. There are a number of popular databases systems available - both paid and free. We have used a free database for our project.
The databases we took into consideration were :

- MySQL
- MongoDB
- MariaDB
- PostgreSQL

We have decided to use MySQL for our project for the following reasons:

It's a freeware, but it is frequently updated with features and security improvements. There are also a variety of paid editions designed for commercial use. With the freeware version, there is a greater focus on speed and reliability instead of including a vast array of features, which can be good or bad depending on what you are attempting to do. This database engine allows you to select from a variety of storage engines that enable you to change the functionality of the tool and handle data from different table types. It also has an easy to use interface, and batch commands let you process enormous amounts of data. The system is also incredibly reliable and doesn't tend to block computing resources.

5.2.2 Services for the DBMS

Initially we were using PhpMyadmin where we created our database and used 000webhost.com to host our php pages. phpMyAdmin is a free software tool written in PHP, intended to handle the administration of MySQL over the Web. phpMyAdmin supports a wide range of operations on MySQL and MariaDB. Frequently used operations (managing

databases, tables, columns, relations, indexes, users, permissions, etc) can be performed via the user interface, while you still have the ability to directly execute any SQL statement.

000webhost is a unique web hosting service where you have the chance to host our webpages for free of cost with adequate security features and smooth connectivity with phpMyAdmin.

The major problem with phpMyAdmin was the lack of scalability and the fact that it didn't provide all the services that we required. We have therefore replaced it with Amazon Web Service (AWS).

Amazon Web Services (AWS) is a subsidiary of Amazon that provides on-demand cloud computing platforms to individuals, companies and governments, on a metered pay-as-you-go basis. In aggregate, these cloud computing web services provide a set of primitive, abstract technical infrastructure and distributed computing building blocks and tools. One of these services is Amazon Relational Database Service (or Amazon RDS) is a distributed relational database service by Amazon Web Services (AWS). It is a web service running "in the cloud" designed to simplify the setup, operation, and scaling of a relational database for use in applications. Administration processes like patching the database software, backing up databases and enabling point-in-time recovery are managed automatically. Scaling storage and compute resources can be performed by a single API call as AWS does not offer an ssh connection to RDS instances.

Chapter 6

Software

6.1 Software For Arduino Nano

The nano acts as the main controller for the system which has been coded using the Arduino IDE. The code follows a sequential pattern. The first part is checking if an authorized fingerprint is present on the scanner. Only if that condition is satisfied, further options will be made available to the user. These options include-

- Enrollment of a new student*
- Enrollment of a new teacher**
- Attendance*
- Deletion of a student entry*
- Master Reset**

* = Selections requiring access level "teacher/staff"
** = Selections requiring access level "administrator"

The options are selected through a push button interface and by observing the selections on the 16x2 LCD screen. The enrollment options call the respective functions which enroll teachers and students according to the selections. The IDs from 5-14 are reserved for access level "teacher/staff". The IDs from 0-4 are reserved for access level "administrator". The remaining IDs are kept for students.

The Attendance option asks for the teacher's ID again as a second verification and then checks the students' fingerprints. On finding a match, the IDs are added to an array. At the end of the class, the teacher would place their finger again on the scanner denoting the end of the lecture and a stop to the attendance marking process. This is followed by a data transfer from the Arduino Nano to the NodeMCU using a serial link at 9600 baud.

The Deletion option deletes the fingerprint placed on the scanner from its database while the Master Reset deletes all the fingerprints except those in the administrator access level.

Snippets of the code can be found in Appendix A section 1.

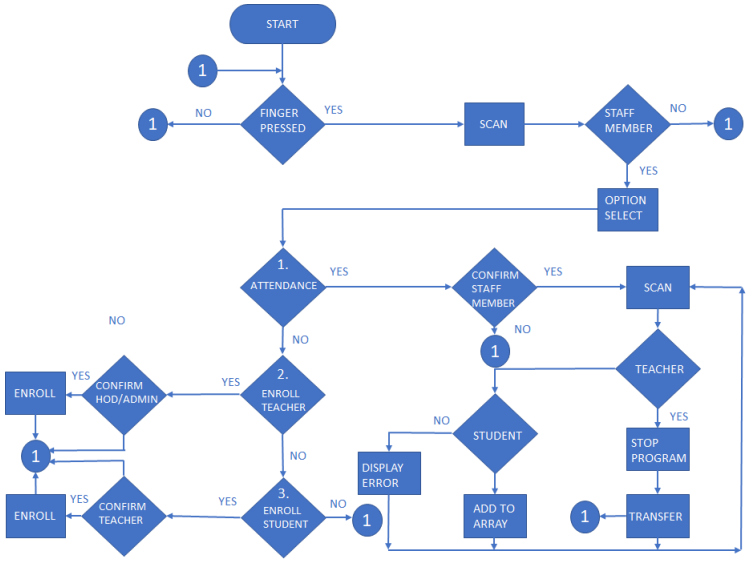

Figure 6.1: Flowchart for the Pod

6.2 Software for NodeMCU

The Nodemcu connects to the Arduino through a serial interface at 9600 baud using software serial ports. The following events occur when data is received:

- When the data is being received from the Arduino the Nodemcu stores it sequentially into its file system called as the SPIFFS.
- The file created in the spiffs is "tid_id.txt" this text file contains a list of the students that are present in the lecture for which the attendance was taken.
- This continues for all other lectures, after all the lectures have been conducted and when the attendance pod is connected to the mains power supply for charging. The

spiffs file system will read out the files present in the Nodemcu's file system and send them over to an online database through a url after connecting to a wiFi network.

- After the completion of sending file the spiffs file system is cleared.

As mentioned in the explanation for data transfer Serial Peripheral Interface Flash File System, or SPIFFS for short is a light-weight file system for microcontrollers with an SPI flash chip. The on-board flash chip of the ESP8266 has plenty of space for other files since we have a 4MB version. This SPIFFS lets us access the flash memory as if it was a normal file system and lets us read and write files, create folders on the system. The advantage of using the spiffs is that even if there is an unforeseen shutdown the data is stored on the flash memory and will be deleted by the system only after all the data has been sent.

The code snippets can be found in Appendix A section 2.

6.3 Transfer to DataBase

The esp8266 connects to the local area network. All the data is currently stored in an array on the flash of the esp8266. We initialise the host to the URL of our webpage (000.webhostapp.com) where the php codes are hosted to push the data into the database. When the Arduino code runs following events occur:

- If there is any data present in the array, the esp8266 tries to connect with the webpage host on port 80.
- If port 80 is free and the host is active, the connection is secured and data ready to be sent.
- The php file Is called from the esp8266 along with the values to be inserted into the database.
- If the page responds back, the 'GET' function is used to send the data to the php page.
- Once the data (SID, TID) has been sent to the php page, connection from esp8266 is terminated and now the php page handles the remaining part.

When the data is in the php page the following events occur:

- The php page first invoked is insert.php, which in turn first calls the db_connect.php page which configures the database connections and verifies the credentials, which are stored in db_config.php, and connects the page to the database and makes it ready for data transfer.

- Then insert.php initialises the query to be executed, the query is the insert statement into the table elex_attendance.

- If the insertion was successful then the pages echo's "successful". Otherwise its echo's "unsuccessful".

- In case if any parameters are missing it will echo "missing parameters".

Thus, in this way data is sent from the esp8266 to the database via php pages. The data is dynamically updated simultaneously in the amazon database and excel sheet. The data is also retrieved by the JSP pages when called by the user. Code snippets can be found in Appendix A section 3.

Chapter 7

Hardware

7.1 Power Management System

Most biometric sensors used in the market today are statically located and require a person to approach the device to mark their attendance. These systems have their own advantages and drawbacks. One of the biggest disadvantages is that the system is bulk and due to its static location most users have to stand around it in a queue which is a waste of time.

Our system combats these disadvantages by having a small size factor and being portable. However, considering that the device is portable, it is necessary to have a battery and a charging dock. Since the battery is being charged and discharged during the usage of the device, it would be wise to make a protection circuit which is capable of protection against over-voltage, over-current, and discharge. Some of the important points while selecting a power unit are as follows-

- The battery should provide constant voltage and current and should deliver 5v/1A
- Safe operation and charging of the battery have to be ensured
- The battery should be able to support a larger number of charging cycles
- The respective power requirements of the components and the on-time.

As stated in the earlier sections, we have come up with a system using the following components-

- Power management IC- TP4056
- Protection IC- FS321F

- Boost converter IC- B6287M
- Battery- LiPo 3.7v 1100mAh

With the TP4056, we used the typical application circuit from the datasheet to develop the final circuit. The TP4056 circuit is shown in figure 7.1.

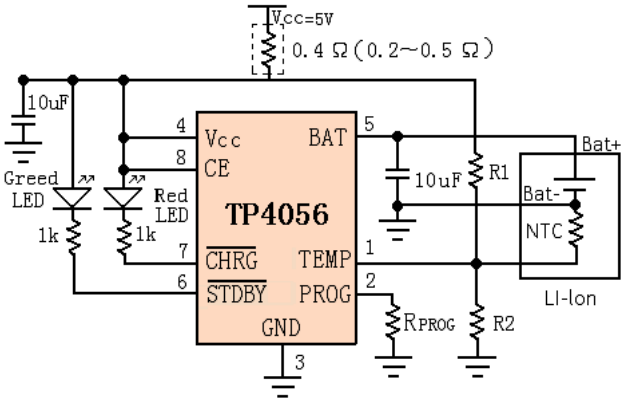

Figure 7.1: TP4056 Typical Application

The programmable resistance Rprog as shown above was set to 1.2kâĎę to get a charge current up to 1000mA.The illustration in figure 7.2 indicates the charge cycle of a 1000mAh battery. As seen the TP-4056 offers a soft-start that limits the inrush current.

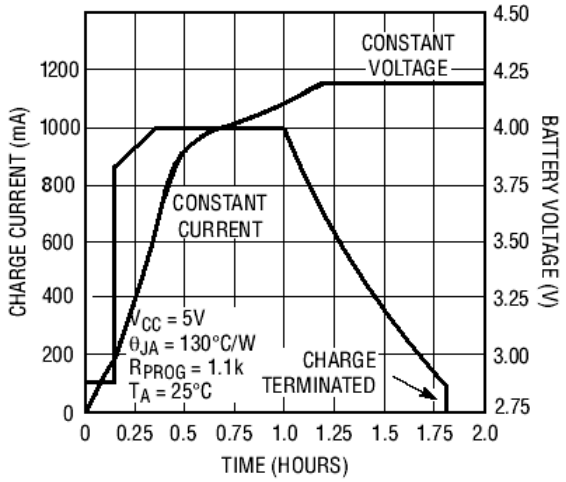

Figure 7.2: Charge cycle of a LiPo battery

For the IC FS312F, we again referred the typical application from the datasheet. This is shown in figure 7.3.

By using the above components and the reference circuits we have designed the following protection circuit for the Portable Biometric Attendance System:

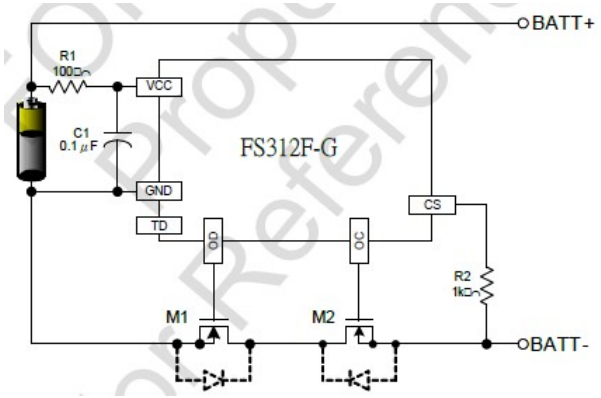

Figure 7.3: FS312F Typical Application

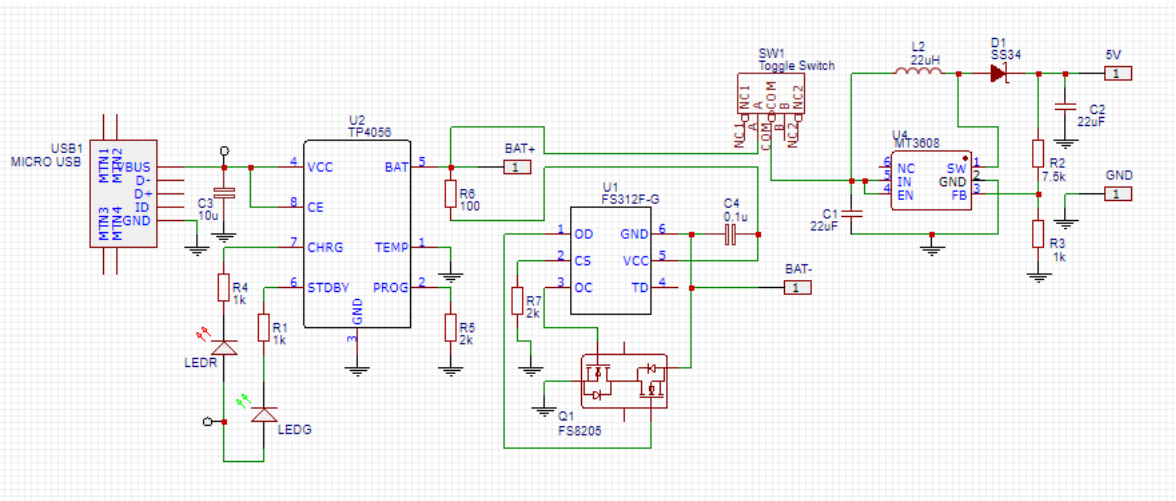

Figure 7.4: Circuit for Battery Charging and Protection

Chapter 8

Product Design

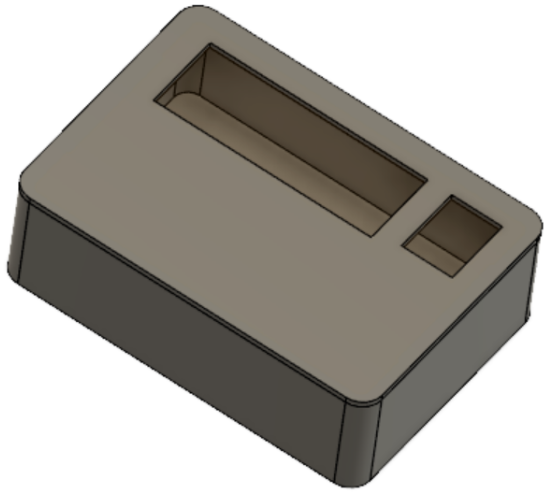

Figure 8.1: Isometric View

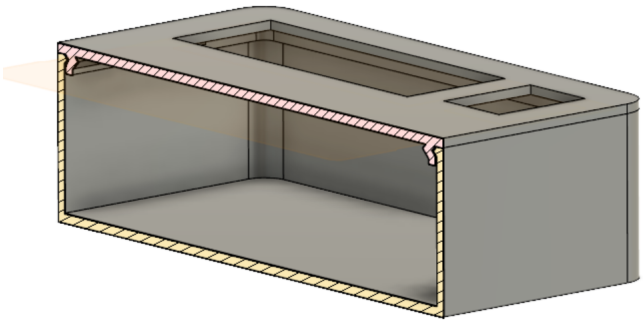

Figure 8.2: Sectional View showing Nubs

The box for the pod has been designed using Autodesk Fusion360 software. It was designed as a combination of 2 individual members- the bottom case and the top cover.

Figure 8.3: Top View

They were designed to snap fit using nubs on the case and indentations on the cover.

Chapter 9

Fabrication

As stated in the previous chapter, the pod is a combination of two separate parts- the case and cover. The cover houses the fingerprint scanner, the LCD screen and the buttons, while the case houses all the remaining components namely, the battery and its circuit, the Arduino Nano and the NodeMCU.

The pod was developed through 3D printing using the MeshMixer software for generating the .stl files. ABS and PLA are the commonly used options for 3D printing. ABS stands for Acrylonitrile butadiene styrene, and is a comminly used thermoplastic polymer. It is highly impact resistant, tough and resistant to heat. It is mildly toxic compound and cannot be recycled easily.

PLA or PolyLactic Acid is a biodegradable and bioactive thermoplastic aliphatic polyester derived from renewable biomass, typically from fermented plant starch such as from corn, cassava, sugarcane or sugar beet pulp. It has comparatively higher tensile strength than ABS and also has a lower density which reduces the overall weight of the product[5]. It is also easier to use, and doesn't release any unplesant or toxic fumes during the moulding process. We decided to proceed with PLA keeping these points in mind.

The final outer dimensions of the pod came out to 120mmx85mmx35mm.

Chapter 10

Implementation Details

10.1 Methodology

As we discuss the methodology, we would like to re-iterate the problem statement: "to create an attendance registry system which would use biometric parameters to take attendance and then upload the results onto a central database." The first step was to analyse and see the difficulties faced by teachers when it comes to attendance in its current form and the following drawbacks were observed:

1. Students sign on attendance sheets which are then updated manually by the teachers.
2. Students mark a lot of false entries and hence accurate attendance is not possible.
3. It's a very tedious task to maintain and update these attendance sheets.
4. There is no user-friendly environment to display the attendance.

These provided us woth out primary objectives which were:

- To successfully identify the students by using their fingerprints
- To successfully update the database using IOT
- To successfully provide a user-friendly environment to access the data

Second step we took was to choose the appropriate devices to satisfy our objectives. In order to successfully identify students using their fingerprints we needed to select a sensor which will be able to:

- Withstand rough use.
- Detect the fingerprints correctly despite being used by over 50 students continuously.

- Low power consumption.
- Having a high storage capacity (At least 100 fingerprint ID).
- Low rejection rate and a verification rate of less than 2 seconds.

The sensor which satisfied all these criteria was GT-511C3 and was therefore, chosen.

In order to successfully update database using IOT, we needed a device to connect the system to the internet and hence we choose the ESP8266 WIFI module. It was a cheap, effective and secure standalone device with a lot of functionality. Now to send data from the ESP8266 to the database we were looking at a few options:

- MQTT
- HTML
- PHP

Upon further research we realized that MQTT was not a secure protocol and that messages being sent from the device to the internet could indeed be hacked. Hence, we discarded that. HTML on the other hand was a simpler and easier option to use to transfer data but upon research we found out that HTML has data failure issues and that data being sent from one place to another some parts often go missing. Hence, we finally decided to use PHP because of its easy, very fast and very secure protocols. Hence, we choose to script PHP pages to send data to our database.

We used Amazon Web Services as it provided us with quality platform at zero cost which was reasonable considering the nascent stage of the pproduct development. Using its free tier services, we choose Amazon Relation Database because of its simplicity in setting up, modifying and controlling. Thus, this way we successfully update our database using all these tools.

10.1.1 Working

1. Capturing of fingerprints into the Nano
    - To get started, we register each fingerprint that we want to store by sending the corresponding command and pressing your finger against the reader three times.
    - This way we register all the teachers first. Once the teacher IDs are stored, we start storing the student IDs in an order as per the student data present on the database.

- The fingerprint sensor initialises a unique ID to every user registered.
- Later, Verification of every user is done by the fingerprint scanner itself. This ID upon verification is stored in the Nano to be sent further to the database.
- Nano has a Flash memory of 32KB which is used for temporarily storing the fingerprint data in its memory before transferring it to the next unit, Esp8266.

2. Transfer to NodeMCU and beyond

   - The data of all the students present in the Nano is sent to this module and is stored in its flash memory(1MB) until the transfer function is called.
   - Once transfer is initiated, the data is sent to a php page via the internet from where we insert the data into the database.
   - We use "GET" function to obtain the php page in the Arduino code and "INSERT" function to insert the data from the php page into the database.

3. At the DataBase

   - We are using Amazon Relational Database Services, provided by amazon web services. it is easy to set up, operate, and scale a relational database in the cloud.
   - It has various database instances; we are using MySQL. The database instance created can be controlled, modified by using MySQL workbench.
   - We have five tables within this database, namely, student (for storing student data), teacher (for storing teacher data), elex_attendance (for storing student attendance), status_attendance (for storing teacher lecture count) and Admin (for storing the credentials of teachers for portal).
   - The student table has a unique ID for every student along with their name and roll no. The student ID is same as the Fingerprint sensor ID e.g.: Fingerprint ID 1 will correspond to student ID 1 on the database.
   - Meanwhile Teacher table has the unique IDs of all the teachers along with their name and subjects. elex_attendance Is where the Student IDs along with the Teacher IDs (teacher taking the attendance) is logged into.
   - status_attendance is where the no. of lectures happening per teacher and total lectures taken is stored.

- The data from the database is then connected to a webpage where the attendance status is displayed for all the students. The webpage is built using jsp and is hosted using flask.api.

Chapter 11

Summary And Future Enhancements

11.1 Summary

This system ensures a quick, hassle free and paperless system suitable to classrooms of the 21st century. The manual updation of attendance and it's calculation has now been replaced with a better system that features a fully automated process of attendance which provides:

1. Reduction, if not complete removal, of false attendance entries.
2. Reduced load on teachers to maintain attendance registries.
3. Easier analysis of attendance.
4. Extremely secure database and no issue of tampering can occur.

On its deployment, the pod can be activated using the biometric credentials of teachers. The pod recognises the teacher ID and associates it with their respective subjects. The students then mark in their attendance by verifying their biometric credentials. Finally the pod is closed by the teacher and the data is sent to the database via the esp8266 through PHP pages, hosted on webhostapp. We obtain the attendance of all students on a webpage which is a portal created for teachers only. It can also be obtained on Excel sheets as well. In the portal we can obtain the attendance for students present for a particular lecture, number of students present for a particular lecture, attendance of a particular student in one subject etc. Thus making this system more efficient and robust as compared to the regular attendance system which requires us to query out the data needed. Here, it's all available on one platform.

11.2 Future Enhancements

- The size can be brought down to allow easier usage.
- A mobile phone application for the students could be developed which would allow them to track their attendance easily.
- The system could possibly be made more secure by using other forms of biometric data such as retinal scans, facial or voice recognition, etc.
- Informing the involved parties if a student's attendance falls below the threshold.
- Use of a touchscreen instead of LCD and buttons to reduce space.
- Inclusion of a wireless charging mechanism
- Infographics to observe trends in students' attendance patterns as a form of feedback.

Appendix A

Code Snippets

A.1 Code for Attendance Monitoring

```
void Attendance() {
 int activeButton = 0;
    int mast=1;
    while(mast){
    yield();
    fps.SetLED(true);
    if (fps.IsPressFinger()){
      yield();
    fps.CaptureFinger(false);
    int t = fps.Identify1_N();
    yield();
    if (t<=14){
      lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Teacher ID");
    lcd.setCursor(3,1);
    lcd.print(t);
    //att[ap]=t;
    //ap++;
    delay(2000);
    int flag=0;
    while(flag==0){
```

```
        yield();
        lcd.clear();
        if (fps.IsPressFinger())
        {
          fps.CaptureFinger(false);
          int id = fps.Identify1_N();
          if (id <200)
          {

            idref=id;
            if(id==t)//id<11
            {
              flag++;
              att[ap]=id;
              ap++;
              lcd.setCursor(0,0);
              lcd.print("Att Over");
              delay(1000);
              lcd.clear();
              lcd.setCursor(0,0);
              lcd.print("Transferring: ");
              int i;
              Serial.write(ap);
              for(i=0;i<ap;i++){

              Serial.write(att[i]);
              lcd.setCursor(0,1);
              lcd.print(i+" of "+ap);
              delay(10);
up:           while(Serial.available()==0);
              char n=Serial.read();
              if(n=='n')
                continue;
```

```
            else
              goto up;
            }
            if(i==ap) Serial.write(235);
            ap=0;
            return;
          }
       else
          {
          lcd.setCursor(0,0);
          lcd.print("Verified ID:");
          lcd.setCursor(3,1);
          lcd.print(id);
          delay(1000);
          }
      }
      else
      {//if unable to recognize
      lcd.setCursor(0,0);
      lcd.print(" Error: Finger");
      lcd.setCursor(0,1);
      lcd.print(" Not Found");
      }
      }
    if(idref!=201){
    att[ap]=idref;
    idref=201;
    ap++;
    }
    }
}
}
}
```

```
  yield();
  while (activeButton == 0) {
    int button;
    readKey = analogRead(0);
    if (readKey < 790) {
      delay(100);
      readKey = analogRead(0);
    }
    button = evaluateButton(readKey);
    switch (button) {
      case 4:  // This case will execute if the "back" button is pressed
        button = 0;
        activeButton = 1;
        break;
    }
  }
}
```

A.2 Code for transferring data to NodeMCU SPIFFS

```
int* arrays_to_store[] = {&array_to_store[0], &array_to_store2[0]};
void setup() {
  delay(1000);
  Serial.begin(115200);
  Serial.println();
  //Initialize File System
  if (SPIFFS.begin())
    Serial.println("SPIFFS Initialize....ok");
  else
    Serial.println("SPIFFS Initialization...failed");
  //Format File System
  if (SPIFFS.format())
    Serial.println("File System Formated");
  else
```

```
    Serial.println("File System Formatting Error");

  write_to_spiff(array_to_store);
}

void loop() {

  //Read File data


    //char* temp_filename = (char*)filename;
    //temp_filename += array_to_store[0];
    Serial.println("Reading from file : ");
    Serial.print(filename);

    File f = SPIFFS.open(filename+(char)2, "r");

    if (!f) {
      Serial.println("file open failed");
    }
    else
    {
      Serial.println("Reading Data from File:");
      //Data from file
      for (int j = 0; j < f.size(); j++) //Read upto complete file size
      {
        Serial.println((char)f.read());
      }
      f.close();  //Close file
      Serial.println("File Closed");
    }
    delay(5000);
```

```
}

void write_to_spiff(int array_to_store[]) {
  //Create New File And Write Data to It
  //w=Write Open file for writing

    int size_of_array=*(&array_to_store+1)-array_to_store;
    char* temp_filename = (char*)filename;

    temp_filename += char(array_to_store[0]);
    Serial.println(temp_filename);
    File f = SPIFFS.open(temp_filename, "w");
    if (!f)
      Serial.println("file open failed");
    else
    {
      //Write data to file
      Serial.println("Writing Data to File");
      for (int i = 0; i < size_of_array; i++)
      {
        f.print(array_to_store[i]);
      }
      f.close();  //Close file
    }

    Serial.println("Data is written into spiff into file: " + (String)tem
    delay(1000);
}
```

A.3 Code for sending data to the database

```
std::vector<int> my_attendance;
static int ptr=0;
```

```
int at=1;
bool send_data= true;


void setup() {
  Serial.begin(115200);
  NodeMCU.begin(9600);
  delay(100);
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  Serial.print("Netmask: ");
  Serial.println(WiFi.subnetMask());
  Serial.print("Gateway: ");
  Serial.println(WiFi.gatewayIP());

}
void loop(){

sending_data_to_database(my_attendance);

}
```

```
//////////////////SEND ARRAY TO DATABASE///////////////////
void sending_data_to_database(vector <int> myattendance ){

  if (send_data){
  Serial.print("connecting to ");
  Serial.println(host);

  int i=2;
  while (send_data){
  WiFiClient client;
  const int httpPort = 80;
  if (!client.connect(host, httpPort)) {
    Serial.println("connection failed");
    return;
  }
  Serial.print("po");
  String url = "/api/insert.php?tid=" + String(myattendance[1]) + "&sid="
  Serial.print("Requesting URL: ");
  Serial.println(url);
  client.print(String("GET ") + url + " HTTP/1.1\r\n" +
               "Host: " + host + "\r\n" +
               "Connection: close\r\n\r\n");
 while (client.connected() || client.available())
{
  if (client.available())
  {
    String line = client.readStringUntil('\n');
    Serial.println(line);
  }
}
  Serial.println();
  Serial.println("closing connection");
```

```
  i=i+1;
  Serial.println(i);
  if(i==myattendance[0]-1){
     send_data=false;
     break;
    }
  }
 }

}
```

A.4 Code For Database

A.4.1 Configuration

```
define('DB_USER', "root");      // Your database user name
define('DB_PASSWORD', "**********");                    // Your database
define('DB_DATABASE', "attendance"); // Your database name
define('DB_SERVER', "**********.c6yo0zghxvzo.us-east-2.rds.amazonaws.com
```

A.4.2 Connection

```
function connect() {

        //importing dbconfig.php file which contains database credentials
        $filepath = realpath (dirname(__FILE__));

        require_once($filepath."/dbconfig.php");

                // Connecting to mysql (phpmyadmin) database
        $con = mysql_connect(DB_SERVER, DB_USER, DB_PASSWORD) or die(mys

        // Selecing database
        $db = mysql_select_db(DB_DATABASE) or die(mysql_error()) or die(m

        // returing connection cursor
```

```
        return $con;
```

A.4.3 Insertion

```
if (isset($_GET['tid']) && isset($_GET['sid'])) {

    $TID = $_GET['tid'];
    $SID = $_GET['sid'];
     // Include data base connect class
    $filepath = realpath (dirname(__FILE__));
        require_once($filepath."/db_connect.php");

     // Connecting to database
    $db = new DB_CONNECT();

    // Fire SQL query to insert data in weather
    $result = mysql_query("INSERT INTO elex_attendance(SID,TID) VALUES('$

    // Check for successful execution of query
    if ($result) {
        // successfully inserted
        $response["success"] = 1;
        $response["message"] = "inserted into elex_attendance.";

        // Show JSON response
        echo json_encode($response);
    } else {
        // Failed to insert data in database
        $response["success"] = 0;
        $response["message"] = "Something has gone wrong";
```

References

[1] Attendance Monitoring System Automation Using Fingerprint Module (By Dept. of Electronics and Telecommunication, MESCOE, Pune, Maharashtra, India, ISSN (Online): 2347-2820)

[2] Attendance System Using Fingerprint Identification with GUI (Department of Electronics & Telecommunication Engineering, Shivaji University Kolhapur, Maharashtra. INDIA ISSN: 2321-9939)

[3] K. A, LazUzoechi, Opara F.K.- Biometric-based Attendance System with Remote Real-time Monitoring system. Department Of Electrical/Electronic Engineering. November 2013, DOI: 10.1109/NIGERCON.2013.6715633 2013 IEEE International Conference on Emerging & Sustainable Technologies for Power & ICT in a Developing Society (NIGERCON)

[4] Fingerprint Based Attendance System Using Microcontroller and LabView Devendra Kumar Yadav1 , Sumit Singh2 , Prof. Shashank Pujari3 , Pragyan Mishra4, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering Vol. 4, Issue 6, June 2015. ISSN (Print) : 2320 - 3765 ISSN (Online): 2278- 8875

[5] https://store.arduino.cc/usa/arduino-nano

[6] ADH Technologies,"Optical Fingerprint Recognition Embedded Module", GT511C3 datasheet, April 2016, v2.1

[7] https://www.espressif.com/en/products/hardware/esp8266ex/overview

[8] NanJing Top Power ASIC Corp,"Standalone Linear Li-lon Battery Charger with Thermal Regulation in SOP-8", TP4056 datasheet.

[9] Fortune Semiconductor Corporation, "One Cell Li-ion/Polymer Battry Protection IC", FS321F datasheet, May 2014, rev 1.2.

[10] Sodergard, Anders; Mikael Stolt (2010). 3. Industrial Production of High Molecular Weight Poly(Lactic Acid). In Rafael Auras; Loong-Tak Lim; Susan E. M. Selke; Hideto Tsuji. Poly(Lactic Acid): Synthesis, Structures, Properties, Processing, and Applications. pp. 27 to 41. doi:10.1002/9780470649848.ch3. ISBN 9780470649848.

[11] https://www.bayometric.com/biometric-devices-cost/