UNIVERSITI TEKNIKAL MALAYSIA MELAKA

DESIGN AND IMPLEMENTATION OF SMART BIOMETRIC CHAIR FOR CLASSROOM ATTENDANCE SYSTEM

This report is submitted in accordance with the requirement of the Universiti Teknikal Malaysia Melaka (UTeM) for the Bachelor of Electronics Engineering Technology (Telecommunications) with Honours.

by

RAJA AHMAD FAWZAN SHAHAB BIN RAJA AHMAD FARIDZ

FACULTY OF ELECTRICAL AND ELECTRONIC ENGINEERING

TECHNOLOGY

2021

DECLARATION

I hereby, declared this report entitled DESIGN AND IMPLEMENTATION OF SMART BIOMETRIC CHAIR FOR CLASSROOM ATTENDANCE SYSTEM is the results of my own research except as cited in references.

Signature: ……………………………………

Author : RAJA AHMAD FAWZAN SHAHAB BIN RAJA AHMAD FARIDZ

Date: 16/02/2021

APPROVAL

This report is submitted to the Faculty of Electrical and Electronic Engineering Technology of Universiti Teknikal Malaysia Melaka (UTeM) as a partial fulfilment of the requirements for the degree of Bachelor of Electronics Engineering Technology (Telecommunications) with Honours. The member of the supervisory is as follow:

Zhossain

Signature: ……………………………………………….

Supervisor : DR. A.K.M. ZAKIR HOSSAIN

Signature: ……………………………………………….

Co-supervisor: NURULHALIM BIN HASSIM

ABSTRAK

Dalam arus globalisasi ini, teknologi kini merupakan salah satu aspek yang semakin mengembang maju, termasuk teknologi berkaitan biometrik. Sehubungan itu, tujuan dokumen ini dibuat adalah untuk mengkaji penghasilan projek prototaip bertajuk “Design and Implementation of Smart Biometric Chair for Classroom Attendance System”. Prototaip ini merupakan sebuah projek inovasi berkaitan sistem kedatangan yang berasaskan cap jari. Peranti ini berkebolehan mengimbas dan mengesan cap jari pelajar bagi tujuan pengambilan kedatangan. Namun, sehingga kini kaedah pengambilan kedatangan masih lagi diambil atau direkod secara manual dengan menggunakan kaedah pengambilan kedatangan berasaskan kertas, dimana pelajar dikehendaki menandatangani kedatangan mereka sebelum kelas bermula. Kaedah ini semakin lapuk dan tidak efisyen kerana terdapat kebarangkalian bahawa pelajar boleh memalsukan tandatangan mereka. Oleh itu, peranti ini boleh membantu dalam meningkatkan tahap kecekapan pengambilan kedatangan pelajar kerana mereka perlu mengimbas dan mengesan cap jari mereka terlebih dahulu sebelum kelas bermula. Setelah proses mengimbas dan mengesan, maklumat mereka disimpan dalam pangkalan data dan maklumat tersebut boleh dilihat melalui laman web. Dengan itu, peranti ini boleh menyumbang kepada institusi pendidikan dari segi pengambilan kedatangan yang lebih efisien justeru menjadikan kaedah ini lebih dipercayai berbanding kaedah pengambilan kedatangan berasaskan kertas.

ABSTRACT

Nowadays, technologies are rapidly advancing in the technological world, including biometric technologies. Therefore, the purpose of this paper is to study the development of “Design and Implementation of Smart Biometric Chair for Classroom Attendance System”, an innovative fingerprint-based attendance device. The proposed device is able to scan and detect students’ fingerprint for their attendances to be recorded. However, ‘till this day, attendances are still being recorded manually using a sheet-based attendance method. Furthermore, students are required to sign their attendance before a class starts. This method is very inefficient as there are probabilities that the students can fabricate their signatures. Therefore, this device is capable to make the attendance recording method more efficient as the students have to scan their fingerprints once they are seated. After the scan has been made, their data is kept in a database and their data can be seen through a website. Hence, this device is enable to contribute to the educational institutions in terms of attendance recording in a more efficient way thus make it more reliable compared to the sheet-based attendance method.

DEDICATION

To my beloved parents, I dedicate all my thanks and gratitude to both of them for constantly providing me love and care with motivational encouragement and also constantly supporting me both morally and financially.

ACKNOWLEDGEMENTS

Thanks to Allah, The Lord of The World. With His grace of blessings, I have completed the Bachelor's Degree Project II report entitled "Design and Implementation of Smart Biometric Chair for Classroom Attendance System" in a timely manner.

In preparation of this paper, I would like to offer my utmost gratitude to Dr. A.K.M. Zakir Hossain as my supervisor in constantly teaching me with his valuable and informative knowledge, guiding me with patience throughout the semester in preparation for this paper. Also, not to forget my co-supervisor, Mr. Nurulhalim bin Hassim for providing me with his insights on the project.

I would also like to thank those who have helped me in every way, so that the paper can be completed. May Allah reward all for their kindness. Ameen.

I realized that this paper is far from perfect, therefore criticisms and suggestions are welcome from all entity to construct and shape me to be successful in the near future.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background

Biometrics includes a vast number of technologies in which different human characteristics are used to identify authenticity. This includes a person's face, hand, voice, iris print, fingerprint, or even signature, which can be used to verify the identity of people who wants to gain access to computers, planes, databases or other areas that are limited. Based on (Sumit Singh, 2015), the skin on human fingers and also palms displays a swirl motion-like patterns of ridges and valleys. The author further stated that these patterns are unique for every person and it is scientifically proven (Sumit Singh, 2015). This proves that biometric system is the most effective and flexible method in verifying information whether in accessing high-security areas or recording attendance.

Therefore, a biometric system is introduced to apply to an existing attendance system. Biometric will provide a solution to an attendance system (Kaium Khan *et al.*, 2019).

In this section, research studies are carried out based on the previous works that are related to this project. It outlines the commonalities and differences point of views between researchers and their methods in applying a biometric system, specifically a fingerprint system to an attendance system.

1.2 Problem Statement

According to (Krishnamurthi *et al.*, 2015), nowadays, attendances are still being recorded manually, for example a teacher calling out students names one by one. The author further stated that this method is time-wasting and troublesome (Krishnamurthi *et al.*, 2015). Moreover, to support the statement, (Kaium Khan *et al.*, 2019) mentioned that a signature is likely to be fabricated if the attendance recording is a sheet-based attendance system. Hence, a biometric-based attendance system is introduced to replace the outdated paper-based method. According to the authors in (Kaium Khan *et al.*, 2019), it is stated that a biometric system is utilized to scan biological traits on humans.

Based on (Sumit Singh, 2015), the skin on human fingers and also palms displays a swirl motion-like patterns of ridges and valleys. The author further stated that these patterns are unique for every person and it is scientifically proven (Sumit Singh, 2015). Furthermore, (Arunkumar and Arun Raja, 2015) expressed that accuracy and reliability are the two most critical criteria for biometric applications with advanced embedded computers too. Hence, why biometric system is popular and universally used for identification purpose (Kaium Khan *et al.*, 2019).

Based on (Krishnamurthi *et al.*, 2015), a fingerprint-based system automatically recorded the attendance. However, (Kamelia *et al.*, 2018) suggested that biometric interface requires the user to place a finger on a disc scanner that will be read. According to the authors in (Krishnamurthi *et al.*, 2015), they stated that the process of attendance management in integrating fingerprint authentication, it consists of two processes which are enrolment and authentication. The paper further stated that during the registration process, the unique biometric traits of the user is recorded and saved in a memory (Krishnamurthi *et al.*, 2015). Meanwhile, (Krishnamurthi *et al.*, 2015) stated that during the authentication process, the recorded biometric traits are compared to all the existing data in a flash memory to matchmaking.

1.3 Objective

The objectives of this research are as follows:

I. To develop a biometric fingerprint-based classroom attendance system.

II. To construct a database that keeps the information of the class for non-concerning personnel.

III. To execute benchmarking to an existing attendance systems.

1.4 Scope of Research

The work scopes that will cover in this project comprise the following areas:

The fingerprint-based biometric system will verify and record attendance via a fingerprint reader and the recorded information can be monitored through computer devices such as laptops or personal computers in real-time using Wi-Fi. The target groups for this project are students and teachers. Students will place their fingerprint onto the fingerprint reader for their attendance to be verified and recorded, meanwhile teachers will monitor the attendance through their laptops or personal computers via a website database. The microcontroller that will be used in this project is NodeMCU ESP8266. It has an additional feature which is the Wi-Fi module that creates a connection that allows teachers to monitor the attendance. Furthermore, as for the database to keep the data, a

PHP-coded website will be used to store attendances. Lastly, a prototype of the fingerprint reader module will be attached to a chair to allow the attendance process being made.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this part chapter, a thorough literature research has been conducted based on the previous works to better understand the research problem being studied. Detailed discussion has been done and analysed on the relationships of each works obtained from various authors that are relevant to this project.

2.2 An Introduction to Biometrics System

A biometric technology or system is basically a design authentication system that identifies a person by determining the originality or authenticity of a particular human body structure or behavioural trait owned by a person. A crucial aspect in creating and developing a functional biometric system is to assess how a person is identified. Depending on the application factors, a biometric system can be called an identification system or a verification system.

A verification system recognizes a person's body traits by comparing the recorded biometric characteristics with the person’s own recorded biometric template system. It conducts a one-to-one comparison to determine the identity of the claim individual truth. A verification system accepts or rejects a submission request for identification.

An identification system recognizes a person by searching the entire template database for a match. It conducts one-to-many comparisons to establish the identity of the individual. In an identification system, the system installs a user’s identity without the user having to approve the identity.

The term authentication is also frequently used in the biometric field, sometimes synonymously with verification. Factually, in the language of information and communications technology (ICT), verifying users means letting the system identify the identity of the user, verification or identification..

Biometrics capability of automated methods for identifying a person based on certain biological or behavioural characteristics of the person is an essential requirement for personal device use in the public domain. Similar to facial recognition, fingerprint is also one of the most popular biometrics because it is a highly accurate method of identifying a person’s identity. (Kamelia *et al.*, 2018) stated that the fingerprint sensor has an average response time of 1.39 seconds, thus the answer to why it is widely used. Furthermore, biological facial or fingerprinting geometric properties and behavioural characteristics such as expressional variables are unique for every person. Biometrics is about identity-confirmation and security techniques that rely on measurable, unique individual biological properties. For example, facial shapes or patterns can be used to access a computer, a room or a security vault.

Biometrics refers to the identification of a person according to the person’s physiological or behavioural characteristics. Nowadays, there are many biometric devices based on characteristics that are unique to everybody. There are several biological properties are now used in biometrics system, for example, the iris, voice, face and fingerprints (Zainal *et al.*, 2014). These properties can be used to positively identify a person. Many biometric devices are based on recording and matching of biometric characteristics to produce a positive verification.

Moreover, the interface of each biometric devices are different from one another but the process are relatively similar. As for example in fingerprinting, according to (Monday *et al.*, 2019), the system has two procedures: registration and identification. During registration, a person's fingerprint is captured and its unique features are extracted and stored in the database with the user's data as the subject template. During the identification, the person's fingerprint is retrieved and extracted and the feature is compared to the template in the database (Monday *et al.*, 2019).

The comparison of various biometrics system could be utilized to identify any human physiological or behavioural characteristics as long as they meet these requirements:

- Universality, which outlines that each individual must have their own biological traits or biometric.
- Distinctiveness, which depicts that every person should be different, in terms of their biological traits.
- Permanence, which means that the biological traits should be constant and fixed over a period of time.

However, in the application of biometric systems, there are multiple limitations that should be addressed and considered, including:

- Performance, it implies the accomplishable authentication speed, precision, and sturdy resource requirements, as well as, to attain the desired authentication speed and precision such as functional or environmental factors that affect recognition properties.
- Acceptability, it refers to the extent to which people are willing to accept a certain amount biometric identifier in their daily lives.

- Circumvention, this reflects how easy it is to trick the system with fraudulent methods. A decent biometric system must have acceptable recognition speed and accuracy with practical resource requirements, harmless to users, accepted by the users and fraudulent-resistance methods.

Statistically, (S. Barra, 2016) impliedly stated that in terms of market share based on the Biometric Market Report, fingerprint-based biometric technology continue to be the most dominant biometric technology, accounting for more than 60% of non-AFIS biometric revenues (Table 1). Note that AFIS is used for the applications of forensic.

Table 1

Comparison of biometric identifiers.

Source: (S. Barra, 2016)

Biometric identifier	**Universality**	**Distinctiveness**	**Permanence**	**Collectability**	**Performance**	**Acceptability**	**Circumvention**
DNA	H	H	H	L	H	L	L
Ear	M	M	H	M	M	H	M
Face	H	L	M	H	L	H	H
Facial thermo gram	H	H	L	H	M	H	L
Fingerprint	**M**	**H**	**H**	**M**	**H**	**H**	**M**
Gait	M	L	L	H	L	H	M
Hand Geometry	M	M	M	H	M	M	M
Hand Vein	M	M	M	M	M	M	L
Iris	H	H	H	M	H	L	L
Keystroke	L	L	L	M	L	M	M
Odour	H	H	H	L	L	M	L
Retina	H	H	M	L	H	L	L
Signature	L	L	L	H	L	H	H
Voice	M	L	L	M	L	H	H

Note: H – High, M – Medium, L – Low

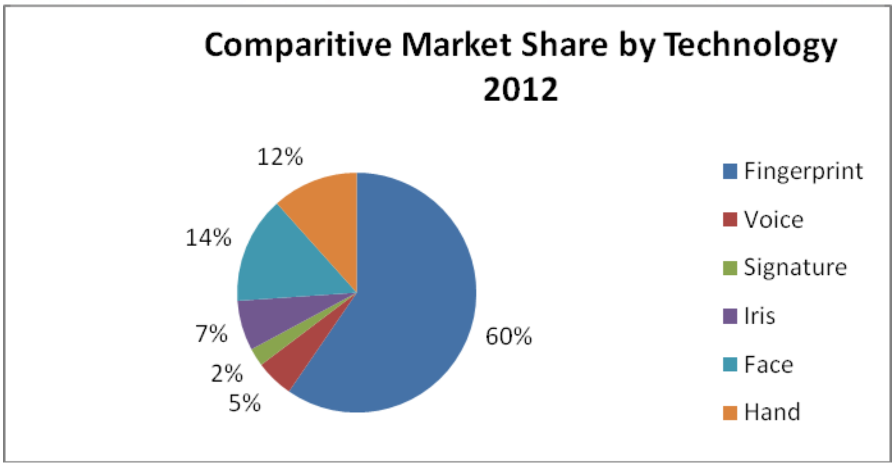


Figure 1: Approximation of the revenue of multiple biometrics in the year 2012 based on Biometric Market Report (International Biometric Group©)

2.3 An Introduction to Fingerprint

As stated in Chapter 1, the patterns of ridges and valleys are unique for every person and it is scientifically proven (Sumit Singh, 2015). For decades, law enforcement worldwide has been categorizing and determining identities by matching the key points of ridge endings and branches. Fingerprints are also unique for each individuals, including twins and this is also scientifically proven. Nowadays, biometric technologies such as fingerprint or face recognition devices for desktop or laptop are commercially available worldwide at a relatively affordable cost. In addition, with the use of these devices, typed-in passwords are no longer needed, instead only a touch or a scan can provide instant access.

Moreover, the fingerprint identification method currently is widely used all over the world in various occasions. As for example, law enforcement agencies use fingerprint identification to identify criminals, or a driving institute uses fingerprint identification to register applicants to lectures or computerized test. On top of that, fingerprints also are unique because of its minutiae properties (the small, precise details of the fingerprint).

2.4 Evolution of Fingerprint Recognition

The modern era or history of fingerprint recognition started in the late 19th century with the development of identification bureaus that were in-charged of keeping records of a person’s traits or according to their biometrics. The studying of the fingerprint identification back then had become a requirement or a necessity to law enforcements. This was due to the increasing number of crime rates happened all over the world and with the aid of the fingerprint identification technology, a criminal could be identified just by through their fingerprints.

Initially, in the history of fingerprinting, Dr. Henry Faulds was the first person to publish the study of a potentiality of fingerprinting use in forensics to the world entitled "On the Skin-furrows of the Hand" in 1880 (B. Sawe, 2019). After that, according to (Jiang, 2015), the first meticulous study on the classification of fingerprint was done by Sir Francis Galton in the late 1880s. Classification was presented as a means of indexing fingerprint to expedite the search in the database (Jiang, 2015). Ten years later, Edward Richard Henry continued Galton's work by refining the subject and produced a concept of fingerprint "delta" and "core" in fingerprint classification (Jiang, 2015).

Furthermore, it was Henry's classification technique that composes the basis for the majority of modern classification schemes. Nowadays, most law enforcement agencies worldwide currently apply some of the Galton-Henry classification technique to narrow down the search but the technique is still not as consistent and reliable as the automatic fingerprint identification system (AFIS) in reducing the search space.

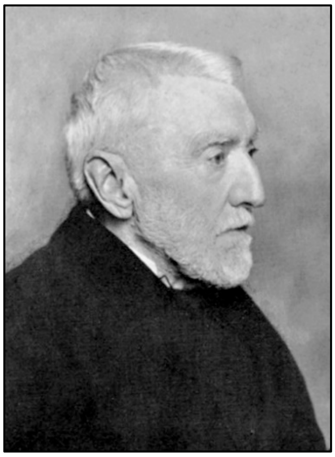


Figure 2(a): Dr. Henry Faulds (Vasilieva, 2014)

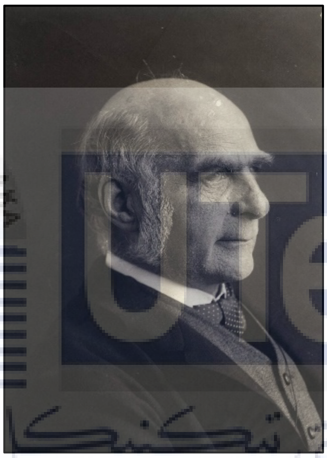


Figure 2(b): Sir Francis Galton (Sir Francis Galton - National Portrait Gallery, 2020)

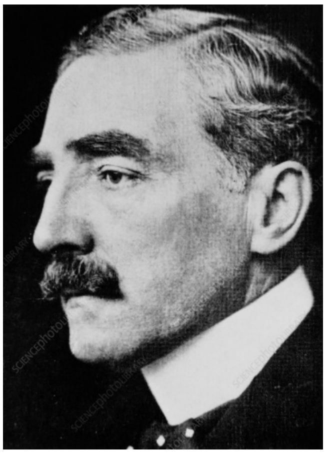


Figure 2(c): Edward Richard Henry (LIBRARY, 2020)

2.5 A Review of Literatures

2.5.1 Biometrics Recognition with Raspberry Pi

In (Arunkumar and Arun Raja, 2015), the authors proposed in designing a biometric authentication system using the raspberry pi. The aim of this paper was to introduce an exclusive application of biometric authentication throughout the world. This is due to the fact that the use of biometric authentication has been one of the most popular authentication system in the world of technology mainly due to its accuracy. Currently, the application of the biometric authentication is widely used in numerous aspects whether to ensure a safe privacy, maintaining security and individual identification (Arunkumar and Arun Raja, 2015). Furthermore, in the technical part, this paper suggested the use of fingerprint reader as collecting users' identity, raspberry pi as its hardware device in controlling the process of authentication, php as its software database in displaying users' information that has been collected and postgresql which acts as its database manager.

2.5.1.1 U.are.U 4500 Fingerprint reader

The fingerprint module used was U.are.U 4500 Fingerprint reader. This module was used to make an identification and verification of users. This reader supports both operating system (OS) whether it is Windows or Linux. This module needed users to place their fingerprint onto the glowing optical window. This module also supports any kinds of OS therefore making it reliable. Besides that, this module also allows data transfer through USB port, and also provide encryption of data making it more secure.

Figure 3(a): U.are.U 4500 Fingerprint reader (Arunkumar and Arun Raja, 2015)

2.5.1.2 Raspberry Pi

Raspberry Pi is a physical hardware device that can interface GSM and GPS. Its properties are the use of 8GB SD to install the Raspbian OS and also for storage. The USB port is used to connect to other devices such as laptops and the power is supplied through the data cable.

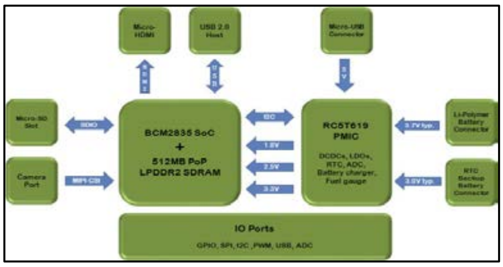

Micro-HDMI
USB 2.0 Host
Micro-USB Connector
HDMI
USB
5V
Micro-SD Slot
SDIO
Camera Port
MIPI CSI
BCM2835 SoC + 512MB PoP LPDDR2 SDRAM
I2C
1.8V
2.5V
3.3V
RC5T619 PMIC
DCDCs, LDOs, RTC, ADC, Battery charger, Fuel gauge
3.7V typ.
3.0V typ.
Li-Polymer Battery Connector
RTC Backup Battery Connector
IO Ports
GPIO, SPI, I2C ,PWM, USB, ADC

Figure 3(b): Raspberry Pi Hardware Components (Arunkumar and Arun Raja, 2015)

2.5.1.3 PHP

PHP is a server-side scripting language designed development of web, however it can be used also as common purpose of programming language which is used in this project. PHP is used to display users' information.

2.5.1.4 PostgreSQL

PostgreSQL, usually merely "Postgres", is a product-related direction system that focuses on resilience and standards-compliance. As a server of a database, firmly storing information within the Raspberry Pi, supporting best practices and permitting it to be retrieved at the request of different computer code applications like PHP. It will handle workloads starting from small machine applications to multiple web-facing applications, even through the terminal on the Raspberry Pi server. The most recent versions conjointly give a replica of the database that's accessible and ascendible in both Windows and Linux.

In conclusion to this paper, the author stated that fingerprint recognition is one of the most widely used biometric technologies mainly because of its accuracy (Arunkumar and Arun Raja, 2015). Nowadays, it has been utilized in several real time applications. However, recognizing fingerprints by installing in Linux primarily based on embedded computers such as raspberry pi is still a major complexity. This work is entirely done on the Linux primarily based embedded computer known as raspberry pi, which postgresql is used as a database management, the creation of webpage using PHP language, and also fingerprint reader access, authentication and recognition were entirely done on raspberry pi.

2.5.2 Biometrics Attendance System

The authors in (Sumit Singh, 2015) suggested designing an attendance system using a biometric identifier which is a fingerprint recognition also with the aid of LabView as its database management to store information of the users. The objective of this paper was to introduce a modified attendance system from the paper-based method that could benefitted the institution system in recording students' attendance. The benefits of the proposed system were that it could save time in recording attendance compared to the paper-based methods. The fingerprint-based attendance system used in this paper was mainly due to the uniqueness of the human fingerprint and that every human being in the world has different fingerprint patterns. According to (Sumit Singh, 2015), the patterns are unique for each person and it is scientifically proven. Furthermore, in the technical part, this paper proposed the use of fingerprint module in collecting users' identity, the use of SFG demo software for interfacing, LabView as the database management and Proteus software for simulation purposes. However, in this paper, it only stated the use of two microcontrollers; Microcontroller 1 and Microcontroller 2, but it does not stated specifically on what kind of microcontrollers were they using.

2.5.2.1 R305 Fingerprint module

R305 fingerprint module was used in this project for recognition purposes. It is a relatively much cheaper compared to U.are.U 4500 Fingerprint reader in the previous paper. It identify users' fingerprints when placed on the glass.

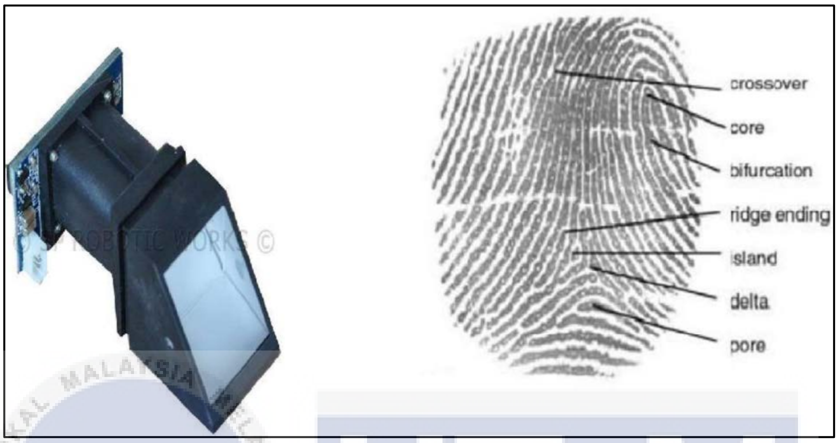


Figure 4(a): R305 Fingerprint module (Sumit Singh, 2015)

2.5.2.2 Proteus software

Proteus software was used for simulation purposes. It portrayed the construction and built of software and hardware before the exact implementation of the system. Proteus also equips accurate virtual apparatus like logic analyzer, oscilloscope, in monitoring the wave signals of the system. Step-by-step switch interfacing develop the process of constructing the microcontroller system. Besides that, Proteus also allows assembly language coding to be downloaded and assembled using the virtual 8051 microcontroller. For coding of C language, Keil compiler was used. Other than that, to configure and run the system, the generated hex code was used.

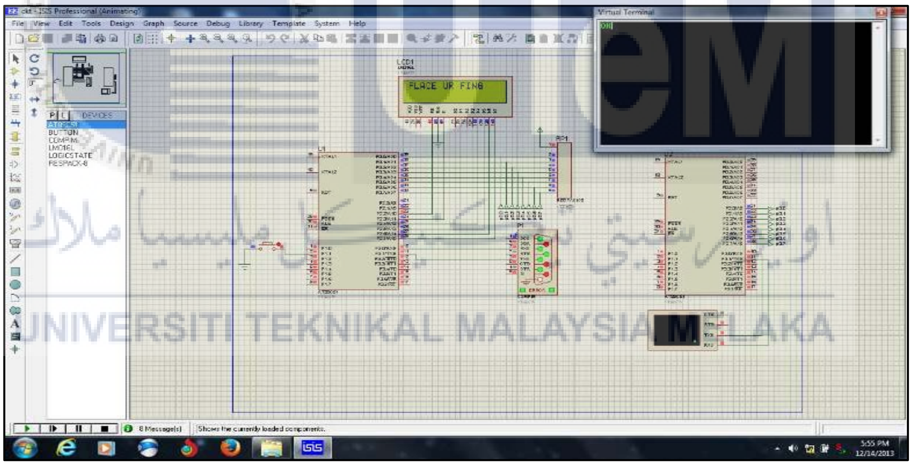

ckt - ISIS Professional (Animating)
File View Edit Tools Design Graph Source Debug Library Template System Help
DEVICES
BUTTON
COMPIM
LM016L
LOGICSTATE
RESPACK-8
LCD1
PLACE UR FING
Virtual Terminal
8 Message(s)
Shows the currently loaded components.
5:55 PM
12/14/2013

Figure 4(b): Proteus software (Sumit Singh, 2015)

2.5.2.3 LabVIEW software

As for database management, LabVIEW software was used in this project to store and keep students' records of attendance.

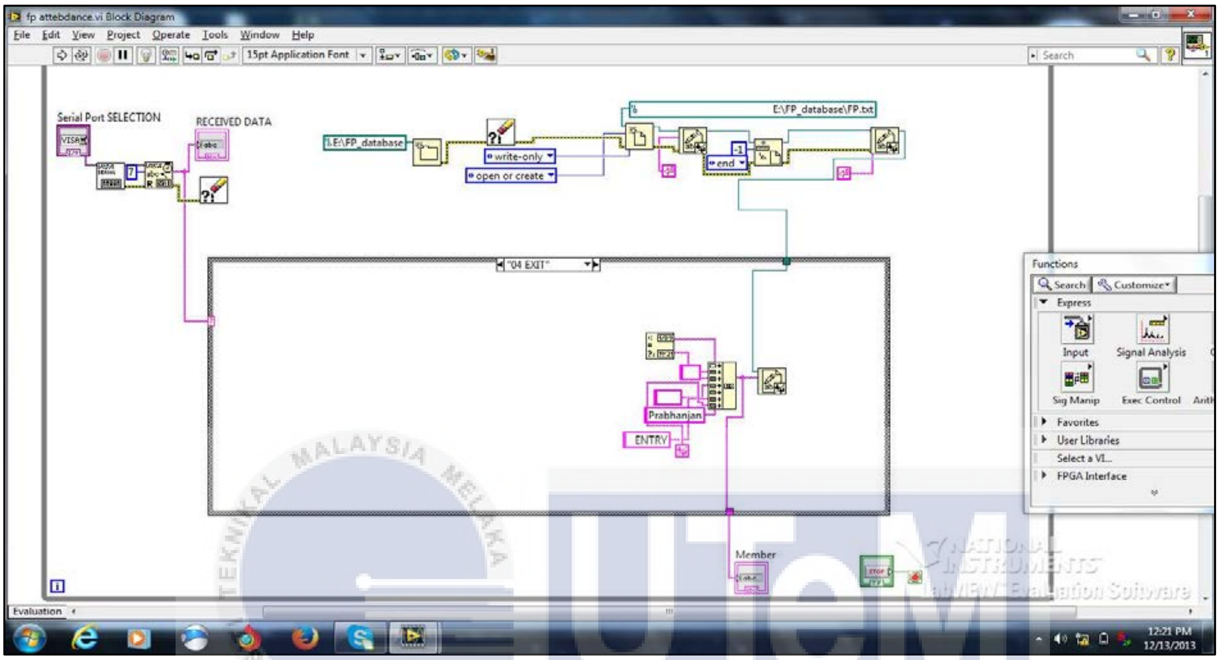


Figure 4(c): LabVIEW database back panel (Sumit Singh, 2015)

In conclusion, this paper portrayed the design development of portable attendance system that relies on fingerprint identification. The system helped to cut back several problems such as denying the probabilities of fabricating recording attendance. This authentication system adds additional security therefore there would be no anonymous fingerprint that is in a position to tamper with the recorded information, and also the portability saves time in taking attending rather than queuing in line. Future works are creating this technique wireless and make use of the IOT (internet of things) concept.

2.5.3 Fingerprint Attendance System with Arduino

In (Krishnamurthi *et al.*, 2015), the authors introduced in designing an attendance system using a biometric identifier which is a fingerprint recognition. As to store students ID, the flash memory from Arduino hardware was used, also in a way functions as a database management. Similarly to the previous works that have been discussed, the objective of this paper was to introduce an attendance system that also could benefit the universities in recording students' attendance. The advantages of using a fingerprint-based attendance system were that it saves a lot of time in recording attendance compared to the traditional paper-based methods. According to (Krishnamurthi *et al.*, 2015), passing the attendance sheet around the class to collect students' signatures was a major downside, and this method allows for more signature fabrication happened.

2.5.3.1 Optical fingerprint reader

The R305 fingerprint reader (Figure 7) consists of TTL UART features to establish a direct connection to the microcontroller to a laptop or computer using a data cable.

2.5.3.2 Arduino

Arduino consists of hardware and software. The software is an open-source platform associated with ASCII text file constituent and software company, project and user community that designs and manufactures kits for building digital devices and interactive objects that may sense and management the physical world. Arduino boards can also be purchased and self-assembled, or as homemade kits.

2.5.3.3 Light-Emitting Diode (LED)

A light-emitting diode (LED) is a source of semiconductor light. It is a PN-junction diode that produce and emits light once activated. This impact of light from LED is named electro-luminescence, and therefore the colour of the light is controlled by the energy resistance.

2.5.3.4 Jumper wires

Jumper wires were used to connect the Arduino hardware to components on breadboard, and with the connection being established will then the system is powered.

2.5.3.5 Breadboard

Breadboard is a board that allows users to execute tests on the board using the required components to establish a connection. With the aid of breadboard, users are also able to calibrate or troubleshoot the circuit if the circuit does not work. The use of breadboard is essential in determining a good construction of a circuit.

2.5.3.6 Flash Memory

Flash memory in the Arduino acted as a database management in storing and keeping students’ attendance. During the authentication process, the identified fingerprint will be compared to all of the recorded fingerprint and if it is a match, the attendance will marked as present.

In conclusion, according to the author, biometric system is an effective method in identity verification and protecting from any fabrication method. Based on analysis made by the author, the biometric can be set, thus the identity can be confirmed. The author also suggested that the use of biometric for authentication reasons should be widely used whether among students and teachers or employees in a company. Furthermore, in terms of performance and efficiency, the biometric method is more reliable and user-friendly compared to the paper-based method.

2.5.4 Wireless Attendance System with Zigbee

The authors in (Kamaraju and Kumar, 2015) proposed in creating an attendance management system wirelessly using Zigbee connection with SDRAM as its database management (Kamaraju and Kumar, 2015). The author stated that a transmitter module comprises of a Tooan OP-100N optical fingerprint sensor was used to capture attendances, a digital signal processing method was used for element processing and also Zigbee was used to establish connection of the system. As for its database management, SDRAM was used to store users' information.

2.5.4.1 Optical fingerprint sensor

The optical fingerprint sensor was used to implement the transmitter module and one of the main factor the author chose this sensor was because of its high quality resolution.

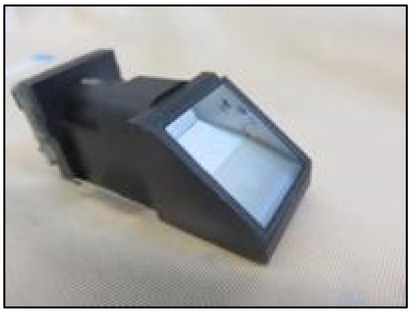


Figure 5(a): Tooan OP-100N optical fingerprint sensor (Kamaraju and Kumar, 2015)

2.5.4.2 Zigbee

Zigbee acted as a transmitter module to transfer the data to the receiver section. It is also able to establish a wireless communication for the digital signal processing to further process the data.

2.5.4.3 SDRAM database

SDRAM acted as a database attendance management system to store the users' information.

2.5.4.4 LCD

LCD displays the information on whether the fingerprint has been registered or not.

Figure 5(b): LCD (Kamaraju and Kumar, 2015)

2.5.4.5 MATLAB

MATLAB was used in this project to demonstrate various mathematical functions and multiple processing methods for image processing of fingerprints.

In conclusion, during the identification phase, the system collected real-time fingerprint profile signals, processed the fingerprint image, extracted fingerprint features, formed the template according to the features, and then match the template with the stored templates profile in database to verify or identify the person. Lastly, the LCD presents the results of recognition process. A program was coded in 'C' language to implement the algorithms for enhancement, matching processing and minutiae extraction. Apart from that, MATLAB was used to demonstrate the various functions and processing methods used in image processing of the fingerprint. The outputs for all the trials were recorded. Finally, the final output of the designed system gave satisfactory results.

2.5.5 Fingerprint Attendance System with NodeMCU ESP8266

In (Devikar, Piyush, Krishnamoorthy Ajit, Bhanage, Aditya, 2016), the authors suggested designing an attendance system using an Internet of Things (IoT) method by using a fingerprint-based biometric scanner. The microcontroller that was used in this project was the NodeMCU ESP8266 integrated with a biometric scanner which is the fingerprint sensor (Devikar, Piyush, Krishnamoorthy Ajit, Bhanage, Aditya, 2016). As for its database, Google Spreadsheet was used to store and keep the users' information. The author did not stated the type of fingerprint scanner used in this paper, but according to the figures attached, it was either R305 or R307 fingerprint module.

2.5.5.1 NodeMCU ESP8266

NodeMCU ESP8266 is an open-source development. Features like GPIO, PWM, I2C and ADC are all integrated in a one board. The GPIO feature has 10 pins and every pins can be PWM. The hardware is somehow like Arduino IO, which is effective in reducing the piled up work for manipulating and configuring hardware. Apart from that, due to its relatively low price, solid design and Wi-Fi features, it defines the IoT product efficiently. Other than that, Lua script is used to program the NodeMCU. Since, the board has Arduino like hardware IO; it can be directly programmed using the Arduino IDE by erasing the NodeMCU firmware. Arduino IDE is additional acquainted and has large documentation and support community.

2.5.5.2 Google Spreadsheet

Google Sheets is a web-based application that sets up for updating, modifying and creating spreadsheets and live online data can be shared. What makes Google Sheet dynamic is because of Ajax. It allows storing and organising multiple types of data like Microsoft Excel. Google Sheets features are easy to create and made modifications on spreadsheets from simple all the way to a more complex. Google Sheet also has a compatibility with CSV (Comma-Separated Values) and Microsoft Excel and can also be saved as HTML.

2.5.5.3 Fingerprint scanner

The author did not state the type of fingerprint scanner used in this paper, but based on Figure 12 below, the fingerprint scanner is either R305 or R307 fingerprint module.

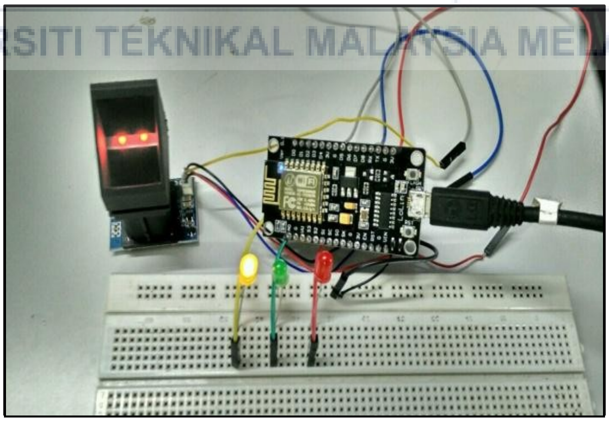


Figure 6(a): Circuit construction (Devikar, Piyush, Krishnamoorthy Ajit, Bhanage, Aditya, 2016)

2.5.5.4 Block Diagram

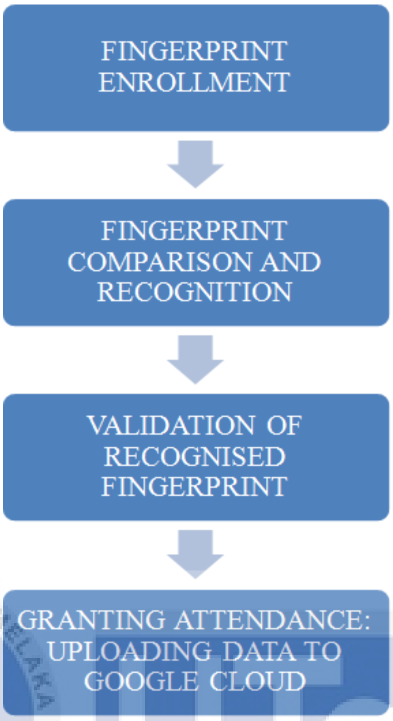
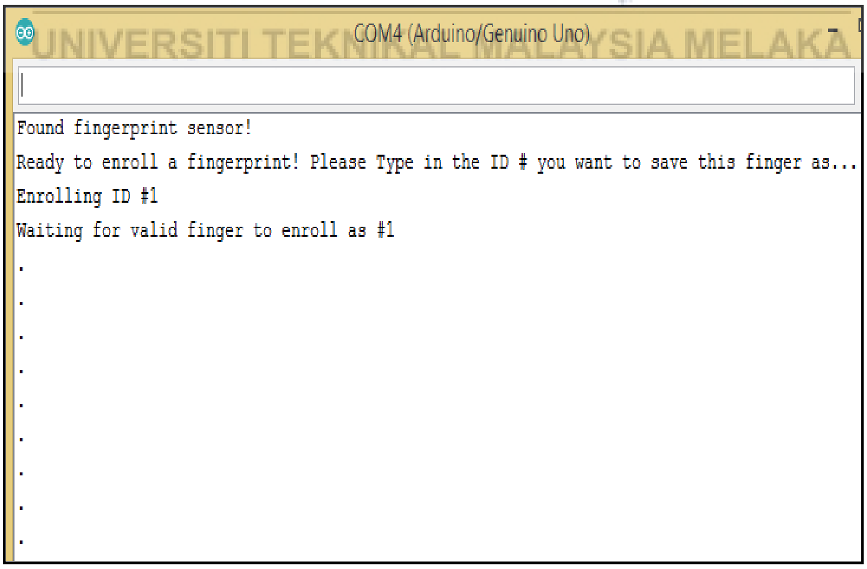

FINGERPRINT ENROLLMENT
FINGERPRINT COMPARISON AND RECOGNITION
VALIDATION OF RECOGNISED FINGERPRINT
GRANTING ATTENDANCE: UPLOADING DATA TO GOOGLE CLOUD

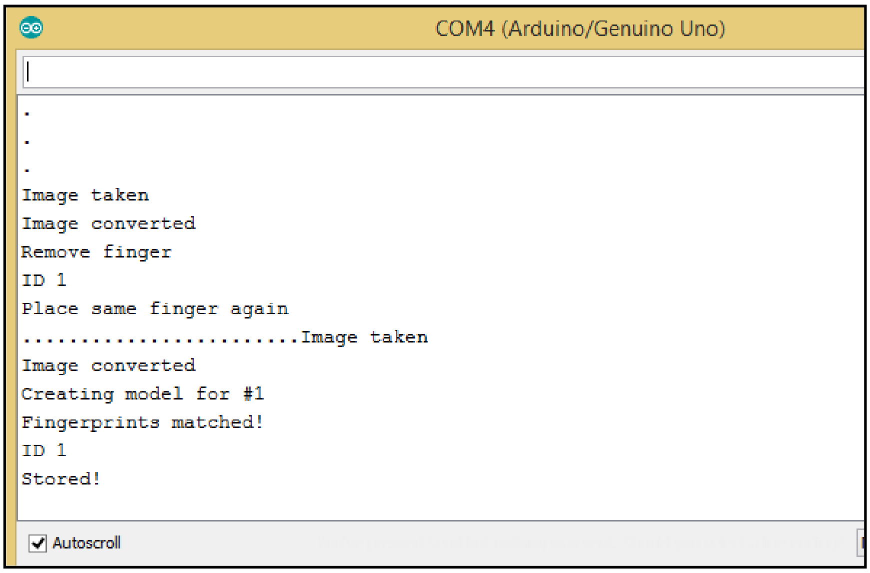
Figure 6(b): Block diagram of recording attendance (Devikar, Piyush, Krishnamoorthy Ajit, Bhanage, Aditya, 2016)

2.5.5.5 Enrolment Process



```
Found fingerprint sensor!
Ready to enroll a fingerprint! Please Type in the ID # you want to save this finger as...
Enrolling ID #1
Waiting for valid finger to enroll as #1
.
.
.
.
.
.
.
.
.
```

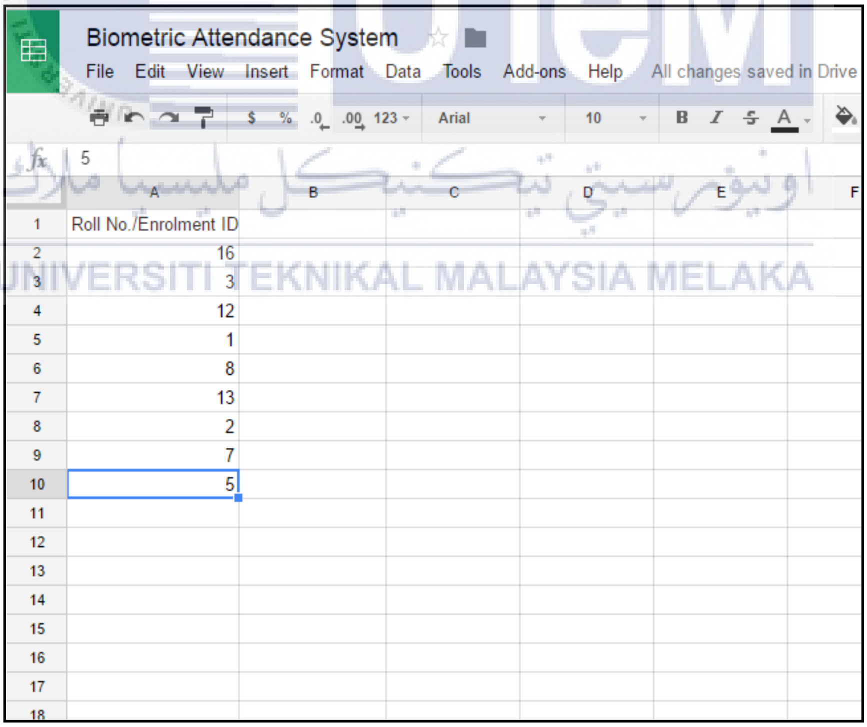
Figure 6(c): Enrolment process (Devikar, Piyush, Krishnamoorthy Ajit, Bhanage, Aditya, 2016)



```
.
.
.
Image taken
Image converted
Remove finger
ID 1
Place same finger again
........................Image taken
Image converted
Creating model for #1
Fingerprints matched!
ID 1
Stored!
```

Figure 6(d): Enrolment process (Devikar, Piyush, Krishnamoorthy Ajit, Bhanage, Aditya, 2016)

2.5.5.6 Attendance Data



	A
1	Roll No./Enrolment ID
2	16
3	3
4	12
5	1
6	8
7	13
8	2
9	7
10	5
11	
12	
13	
14	
15	
16	
17	
18	

Figure 6(e): Enrolment process (Devikar, Piyush, Krishnamoorthy Ajit, Bhanage, Aditya, 2016)

In conclusion to this paper, (Devikar, Piyush, Krishnamoorthy Ajit, Bhanage, Aditya, 2016) said that the old method of manually recording attendance of students was highly inefficient and time-wasting. An IoT-based portable biometric attendance system can be proven to be of benefit to universities as it proves to be highly accurate and secure. (Devikar, Piyush, Krishnamoorthy Ajit, Bhanage, Aditya, 2016) further stated that the cost involved in making this system was quite cheap when compared to the conventional biometric attendance system. Other than that, cloud computing that was used to store the attendance data and records made the system easier to manage and retrieve. Lastly, the use of fingerprint scanner guaranteed the reliability of the attendance system and also user-friendly.

Table 2

Journal comparison of related and relevant previous papers.

No.	Author	Title	Platform	Method
1	(Arunkumar and Arun Raja, 2015)	Biometrics Authentication using Raspberry Pi	Raspberry Pi	Fingerprint-based
2	(Sumit Singh, 2015)	Fingerprint Based Attendance System Using Microcontroller and LabView	8051 Microcontroller	Fingerprint-based
3	(Krishnamurthi et al., 2015)	Fingerprint Based Attendance System	Arduino	Fingerprint-based
4	(Kamaraju and Kumar, 2015)	Wireless Fingerprint Attendance Management System	Zigbee	Fingerprint-based
5	(Devikar, Piyush, Krishnamoorthy Ajit, Bhanage, Aditya, 2016)	IoT Based Biometric Attendance System	NodeMCU ESP8266	Fingerprint-based

Collectively, after a thorough dissection of these five journals, it can be concluded that the fingerprint-based method is the most efficient and reliable method or system and is still widely used till this day compared to other biometric identifiers such as facial recognition or retinal scan. This is due to the flexibility of the system itself. It is cost-effective, high accuracy verification process and has a quick response rate, which makes it relevant and useful.

CHAPTER 3

METHODOLOGY

3.1 Introduction

In this chapter, detailed methods and procedures are discussed to support the objectives after reviewing previous related literatures on how the project could be made. It outlines the crucial aspects of the project as this chapter consists of a flowchart that determines the flow of the project, a block diagram that depicts the graphical display on the related connection of the project, components that will be used to build the prototype and also procedures that are needed to ensure that the methodology implementation are in sequential order.

3.2 Project Flowchart

The flowchart below depicts the overall implementation flow of the project. It consists from conducting a thorough research on the project, writing a literature review, hardware and software implementation and all the way to the result outcome.

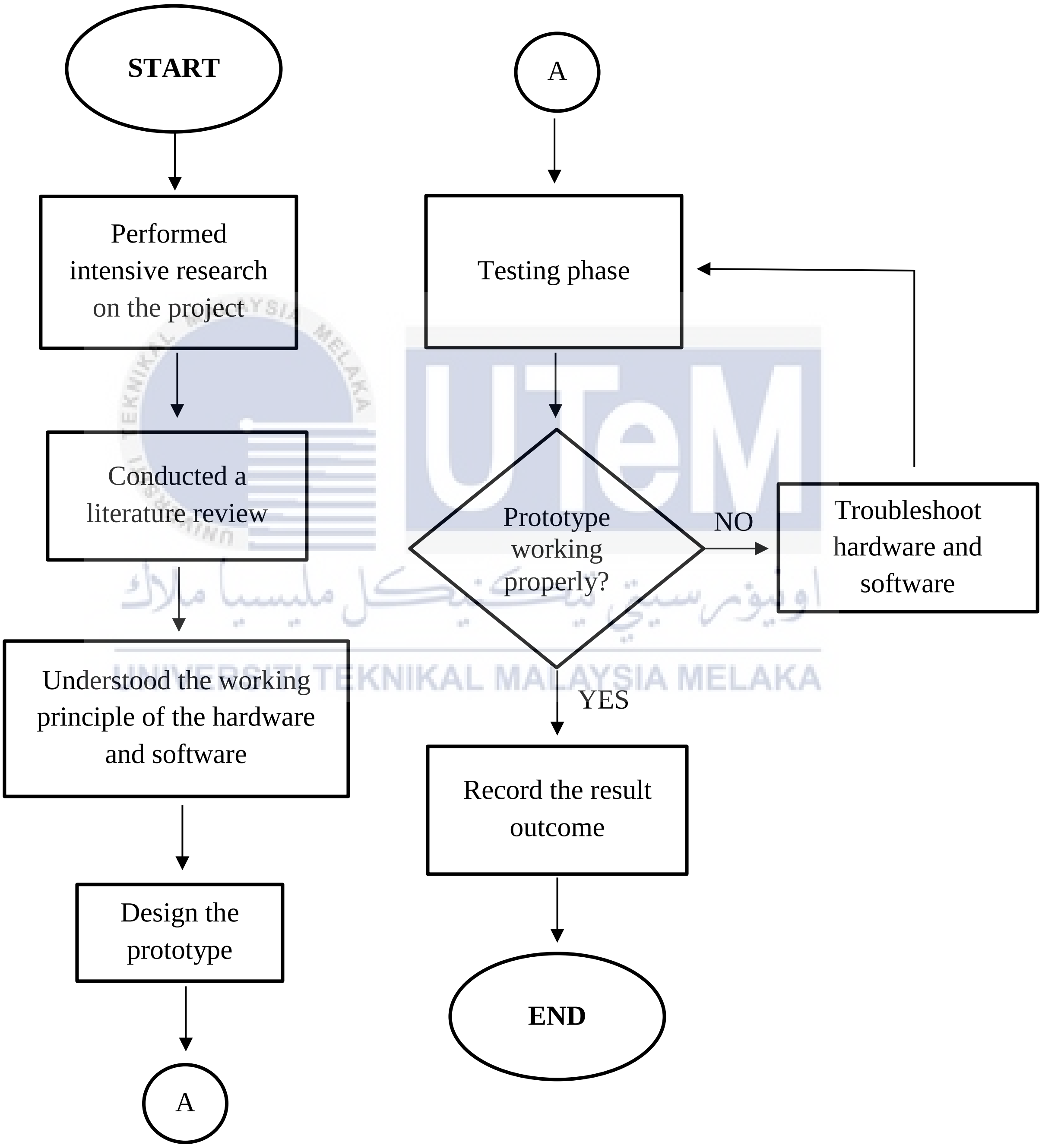

START
Performed intensive research on the project
Conducted a literature review
Understood the working principle of the hardware and software
Design the prototype
A
A
Testing phase
Prototype working properly?
NO
Troubleshoot hardware and software
YES
Record the result outcome
END

Figure 7: Project Flowchart

3.3 Conceptual Design of the Project

Basically, the “Design and Implementation of Smart Biometric Chair for Classroom Attendance System” is a fingerprint-based biometric prototype device that is attached to a chair armrest. Initially, prior recording the attendances, registrations are made first with users giving out their unique data traits by the device scanning their fingerprints. Then, the data is stored in a database to keep the users’ log information. After the registration process is completed, then only the proper classroom attendances can be recorded.

Objectively, for the attendance recording process to happen, the device scans users’ fingerprints and thus records their attendances in the database. Once the attendances have been recorded, the users’ attendances logs information are displayed in an open-source website database that displays the name of students, students’ fingerprint ID, time-in and time-out timeline just to name a few. The logs are also can be exported to an excel sheet and downloaded.

3.4 Bill of Materials

The table below shows types of components required to assemble the device.

Table 3

The required components.

No.	Item	Description	Quantity
1	Breadboard	-	1
2	Jumper Wires	Female-to-Female	4
3	Fingerprint Sensor	R305 fingerprint module	1
4	NodeMCU	ESP8266	1
5	OLED Display	0.96" OLED Display	1

3.5 Hardware Utilization

In this project, various types of hardware components will be used for the system, mainly the R305 or R307 fingerprint sensor, the NodeMCU ESP8266 and the OLED display. The fingerprint sensor will be used for scanning purposes for recording students' attendances. Meanwhile, the NodeMCU will be used as a microcontroller to provide power supply to the components. Apart from that, it has Wi-Fi feature that allows the components to be connected to the internet. Furthermore, an OLED display will display the notification when the attendance is recorded.

3.5.1 R305/R307 Fingerprint Module

This fingerprint module is equipped with TTL UART interface for a straight up direct connections to a microcontroller or to PC or through a USB adapter. This module can be interfaced directly with any microcontroller whether in Arduino, Raspberry Pi or others.

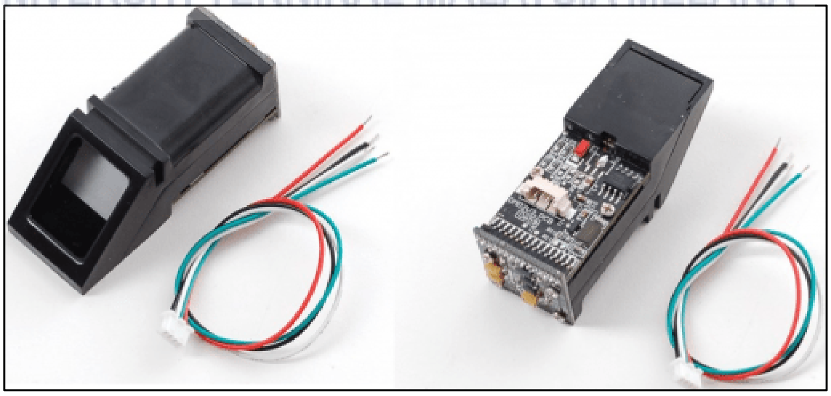


Figure 8: Front and Back Display of R307 Fingerprint Module

3.5.2 NodeMCU ESP8266 Wi-Fi Module

This module equipped with LUA script can be directly interfaced and is compatible with various programming languages, for example in this project would be using the Arduino IDE software to allow configuration of the coding process. In addition to that, this module is also equipped with Wi-Fi feature that enables stable long range connections.

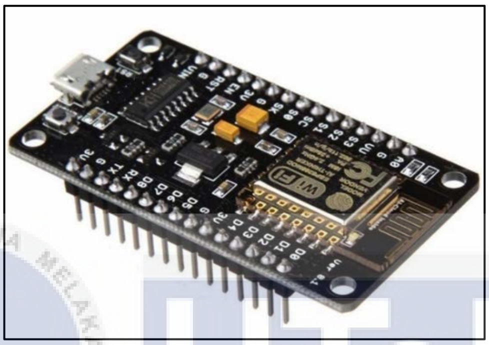


Figure 9: NodeMCU ESP8266

3.5.3 OLED Display

This module has a 0.96 inch blue OLED display. Similarly with the fingerprint module, this module which consisted of SPI/IIC protocols can be simply interfaced with any other microcontroller. It has characteristics of a display panel and 4 sources of pin.

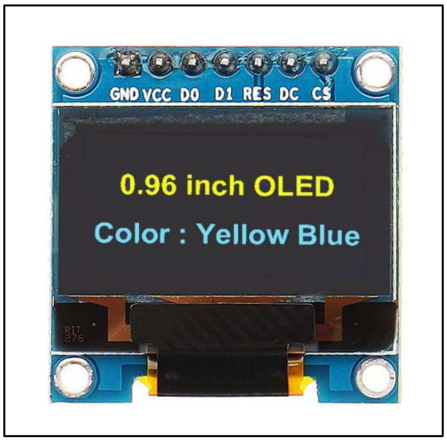


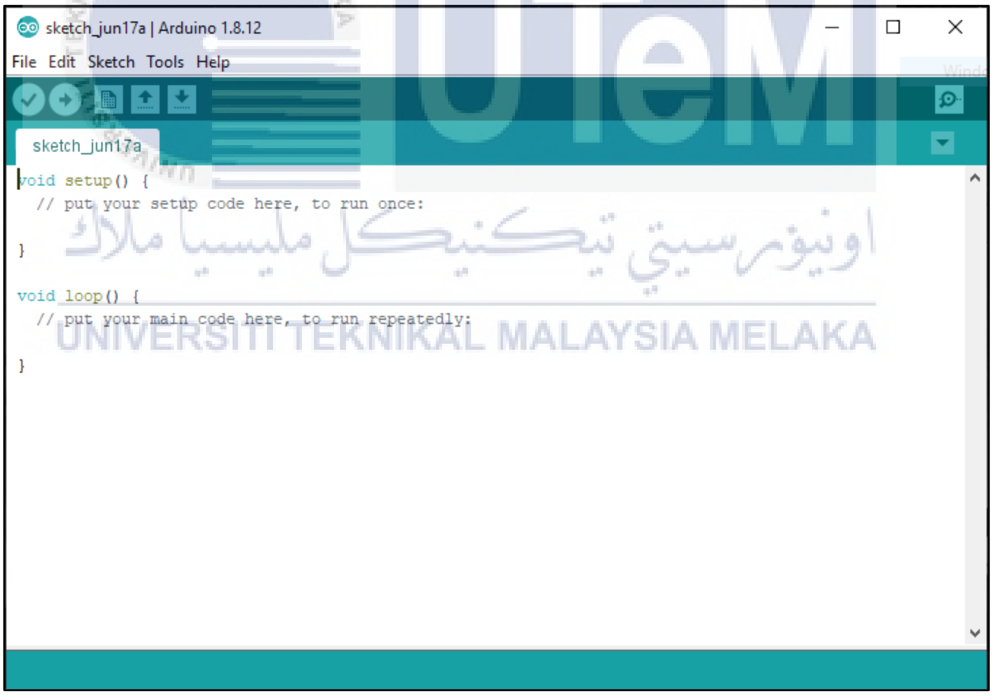
Figure 10: OLED Display

3.6 Software Utilization

In this project, two software will be used, mainly the Arduino IDE software and the XAMPP software. The Arduino IDE software is used to program the coding onto the NodeMCU interfacing with other components. Relatively, as for XAMPP software, it will be used as a database to store and keep the attendees' record.

3.6.1 Arduino IDE

The Arduino IDE functions as a programming software to program the coding onto the NodeMCU to be interfaced to the fingerprint sensor and the OLED display.



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Figure 11: Arduino IDE interface

3.6.2 XAMPP software

XAMPP software will be used as a database to store and keep the records of attendances. It provides the building of an offline website, using a local web server on the computer. The set-up of the website is locally done by using the computer IP address as a server to load and store data in a local host

3.7 Block Diagram of the Project

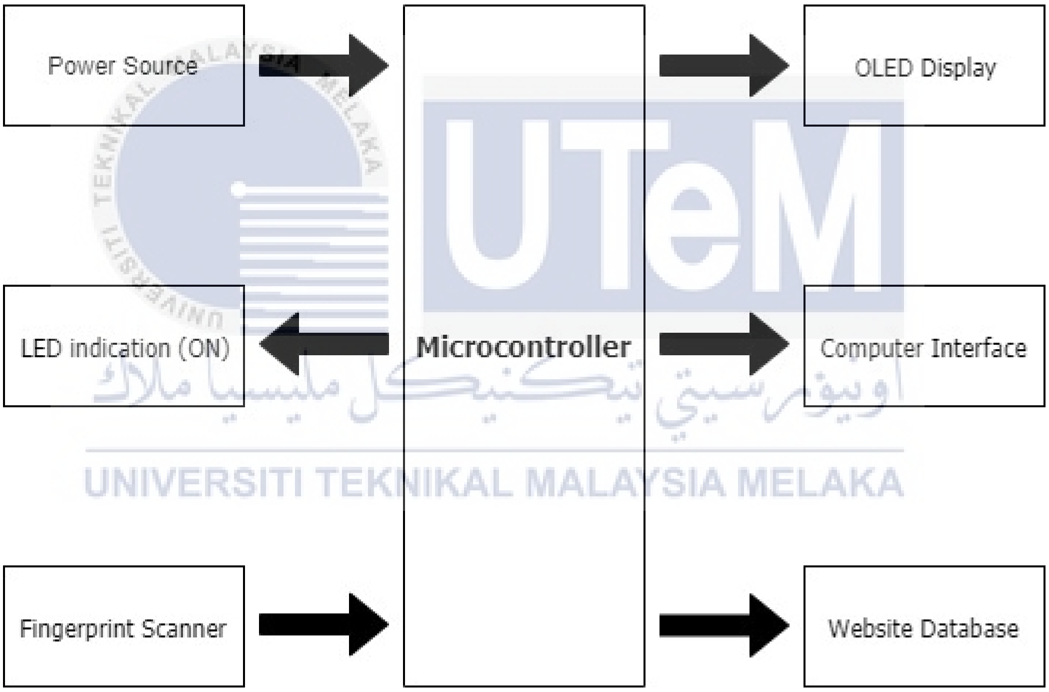


Figure 12: Block Diagram of Smart Biometric Chair for Classroom Attendance System

Based on the figure above, the block diagram represents the basic working concept of Smart Biometric Chair for Classroom Attendance System. The power source of the microcontroller of NodeMCU ESP8266 is obtained from the USB adapter connected to the PC. Once the connection has been made, an LED light from the microcontroller blinks a few times to indicate that the power is up and there is a connection. Then, for the attendance recording, users place their fingerprints onto the scanner for the first time for registration purposes. The registration process allows the data being collected and stored in a database for later use.

After that, for the proper attendance recording, users place their fingerprints onto the scanner again to clock-in for a particular class, and place the fingerprints again to clock-out. In addition to that, the OLED display depicts the attendance status on the OLED panel in writing. Apart from that, as for database, a free and an open-source website database is used to store the users' attendance. The database also allows exportation to Excel sheet of the attendance for document purposes.

3.8 Schematic Circuit Connection of the Project

The below figure shows the schematic circuit connection on how an OLED Display & fingerprint module is interfaced to NodeMCU ESP8266. The D1 and D2 pins of the NodeMCU is connected to the I2C pins of OLED Display, SCL and SDA pins respectively. Similarly, as for the fingerprint sensor, it is connected to D5 and D6 pins of UART. The colour of Tx and Rx wires of the fingerprint sensor may vary accordingly. However, according to figure below, the colour of Tx wire is yellow and the colour of Rx wire is orange. The connection must be accurate for the fingerprint module to be detected by the NodeMCU. A voltage of 5V is supplied through the Vin pins of the NodeMCU to the R305 fingerprint sensor. Similarly, the OLED Vcc pin is connected to 3.3V pin of the NodeMCU. Furthermore, a breadboard circuit diagram concept is added below for illustrative purposes (Figure 20).

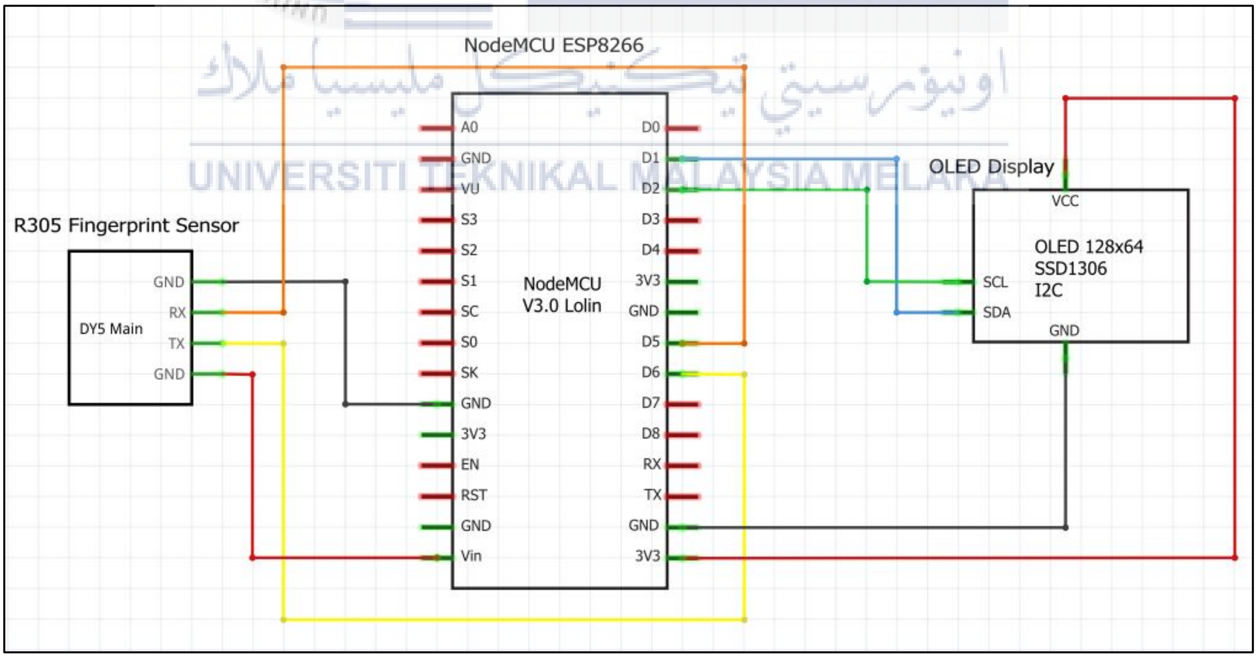


Figure 13: Schematic Circuit

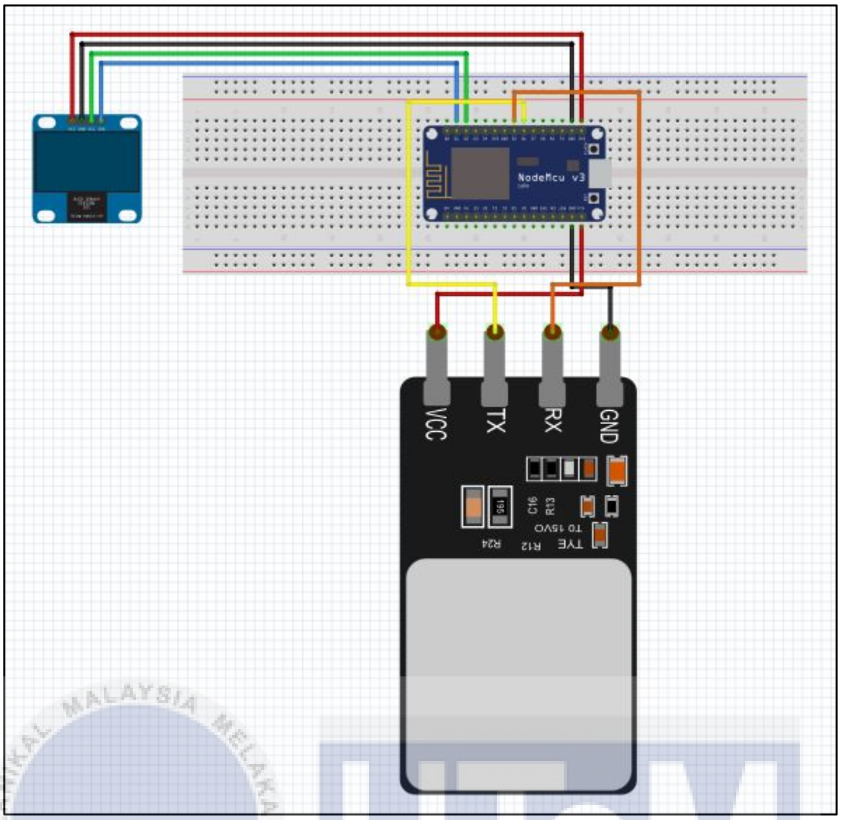


Figure 14: Breadboard Circuit Diagram Concept

As a conclusion, after outlining the design concept and the possible outcome of the project, it is certainly achievable in implementing the prototype with the aid of the fundamental working principle based on the flowchart and block diagram. However, it is undeniable that the outcome of the project may vary from the original concept. This could be due to various factors such as an instrumental error, programming error, simulation error and others just to name a few. Therefore, it is crucial to be meticulous in analysing, constructing, designing, implementing and also in testing phase to ensure that the project is faultless and accurate.

CHAPTER 4

RESULT AND DISCUSSION

4.1 Introduction

In this chapter, comprehensive methods and procedures of the actual result of the project are discussed to support the objectives after designing the expected outcome of the hardware, developing the programming and implementing the programming to the product. It outlines the technical process of the project as this chapter consists of figures of the actual result that depicts the working process of the project chronologically. The discussion includes the registration of users' fingerprints and the recording of attendance in the database. Other than that, this chapter also discussed on the benchmarking of this product with another existing product to support one of the objectives.

4.2 Hardware Design and Coding Development of the Device

This section are dissected into three parts of the working process. First, the hardware design which consists of the circuit construction on a breadboard, the circuit connection with the components' wires and jumper wires, and the expected hardware casing design for the device to be placed on an armrest of a chair. Next, the second part is the development of the coding using the Arduino software. Meanwhile, as for the third part is the construction of the database using the XAMPP software. The development of the Arduino coding and the construction of the database using the XAMPP software are in correlation between one another basically to activate the connection and the working function of the project with the respective components essentially the NodeMCU ESP8266, the R305 fingerprint sensor and the OLED display.

4.2.1 Hardware Design

Initially, one of the components, which is the fingerprint sensor would be used for this project was suggested with two versions as discussed in Chapter 3 whether it would be the R305 fingerprint module or the R307 fingerprint module. After testing both of the modules, it has been decided that the R305 fingerprint sensor would be the more suitable choice for this project. The reason behind this was prior to the decision, initially the R307 was the first to be tested with the available connection. During the testing phase, it was discovered that the R307 fingerprint module could not be connected. This was probably due to the device's faulty condition. Thus, the R305 fingerprint module is used for this project. Below is the figure of the circuit construction of the project.

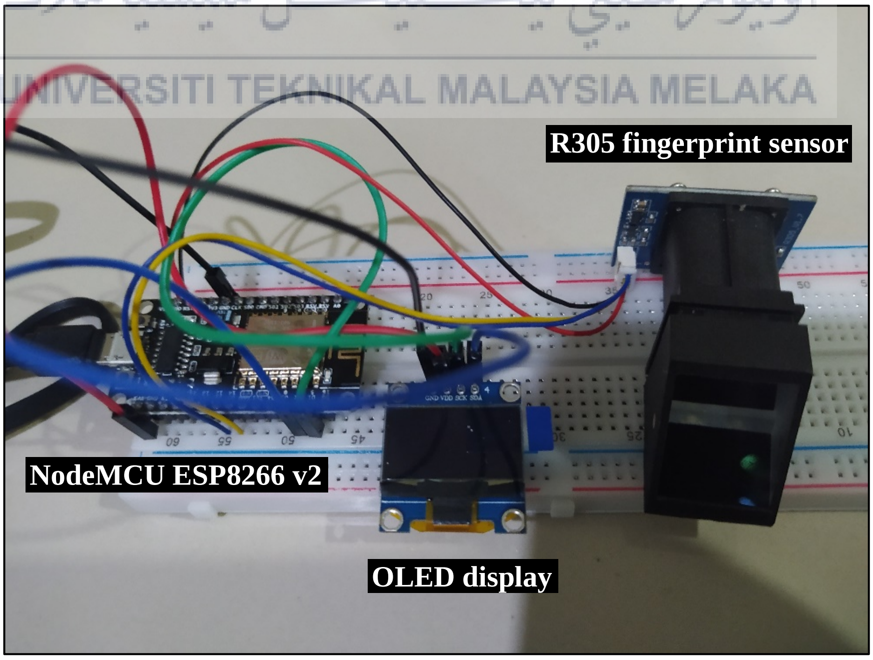


Figure 15: Circuit Construction

After the wiring connection is constructed, then proceeded to the configuration of the connection to witness if the connection was correct by supplying power into the NodeMCU board. Therefore, based on the figure below, the circuit was powered up and tested. The OLED display was turned on and displayed outputs as the micro USB cable was plugged into the NodeMCU board. This showed that the connection is correct.

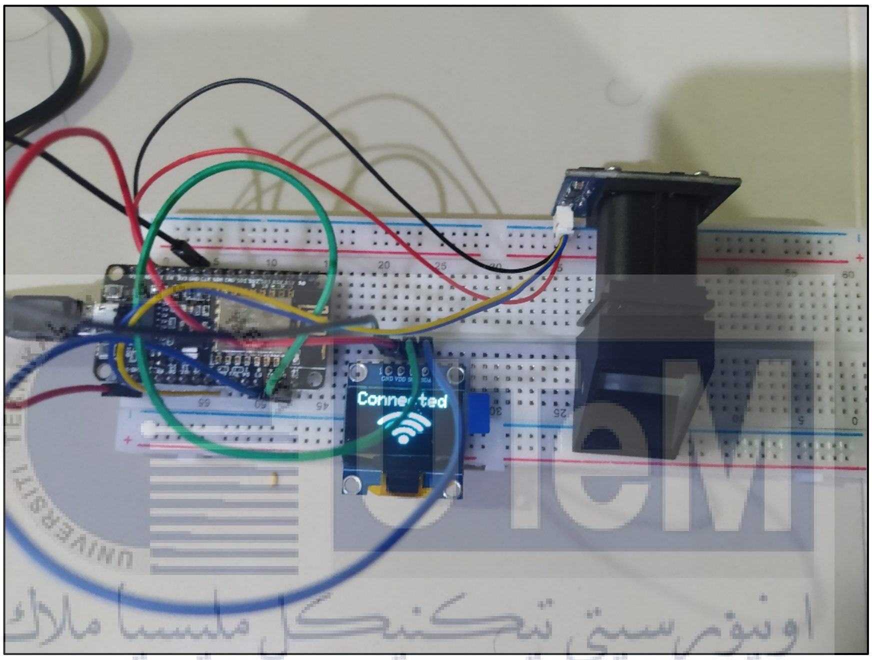


Figure 16: Circuit Connection Configuration

After the circuit construction and the wiring connection on the breadboard were successful, then proceeded to the connection of the circuit by using only jumper wires to be interconnected between the components. The four wires of the R305 fingerprint module were soldered into four clipped female-to-female jumper wires. Furthermore, the OLED display pins were also soldered. The reason behind this was to simplify the instalment and setup of the circuit to be able to fit the circuit into the hardware casing later on. Below is the circuit connection that has been soldered.

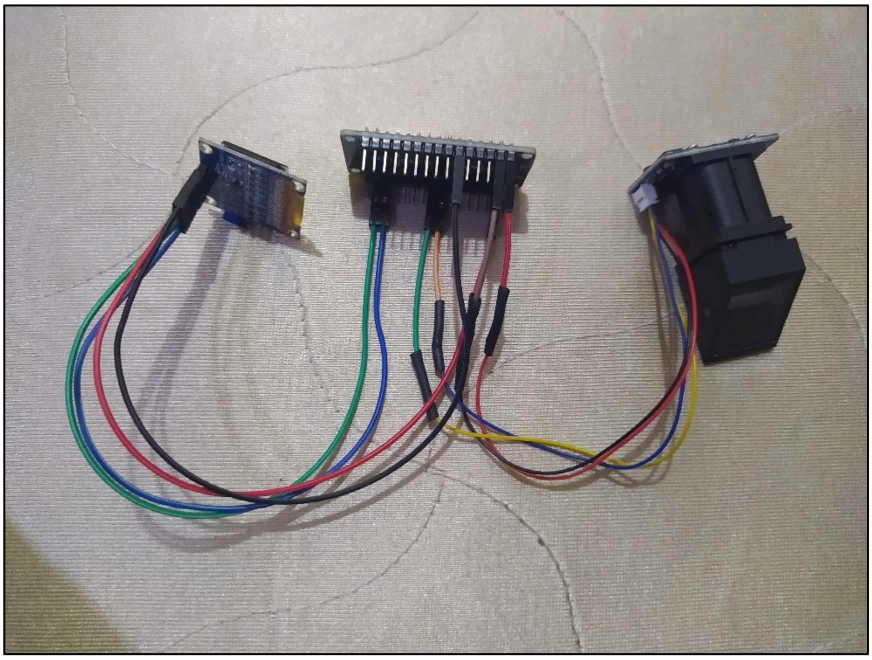


Figure 17: Circuit Connection (Soldered)

After the circuit connection has been soldered, the process then proceeded to the hardware casing design. A standard rectangular-shaped box was used as a reference in determining the hardware case for the project. The box for the hardware case was shown as below.

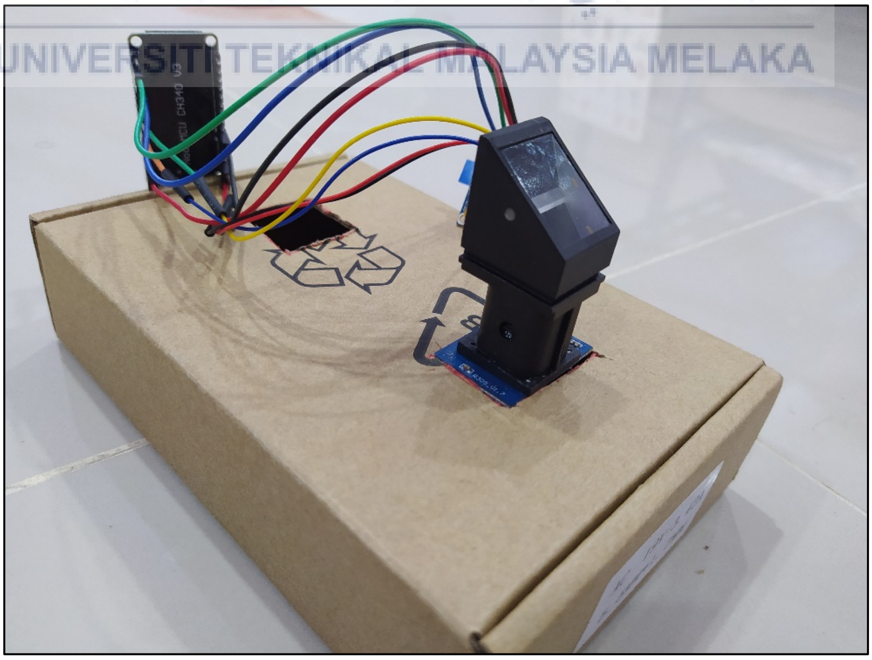


Figure 18: Hardware Casing Design Concept

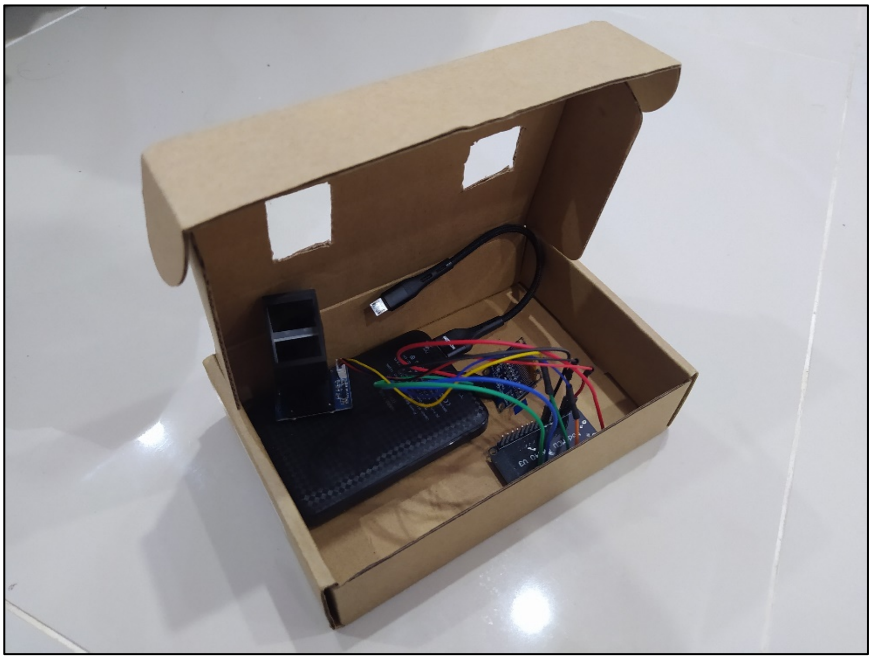


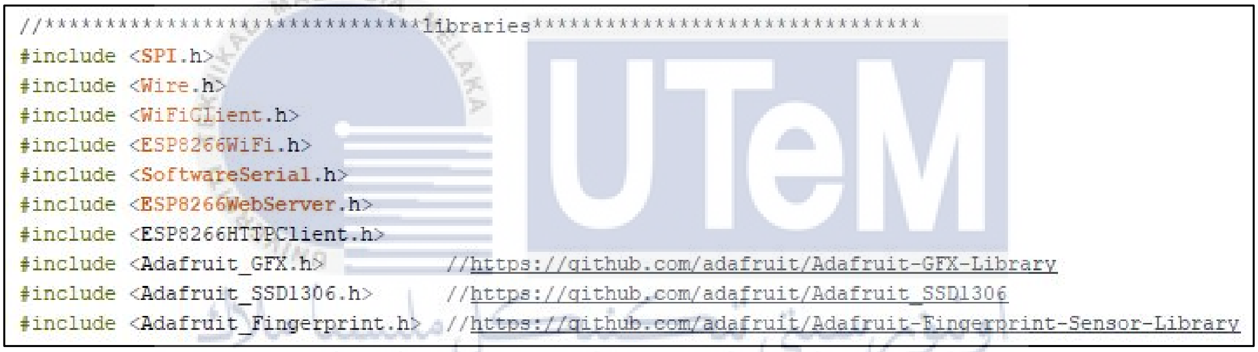
Figure 19: Inside Look of the Hardware Casing Design Concept

4.2.2 Development of the Arduino Coding

The development process of the coding for this project is basically acts as a catalyst for this project. The constructed programming codes for the Arduino software have to be accurate and error-free in order to allow the connectivity activation on the NodeMCU ESP8266 to control the overall system.

4.2.2.1 Arduino Coding for R305 Fingerprint Module and OLED display

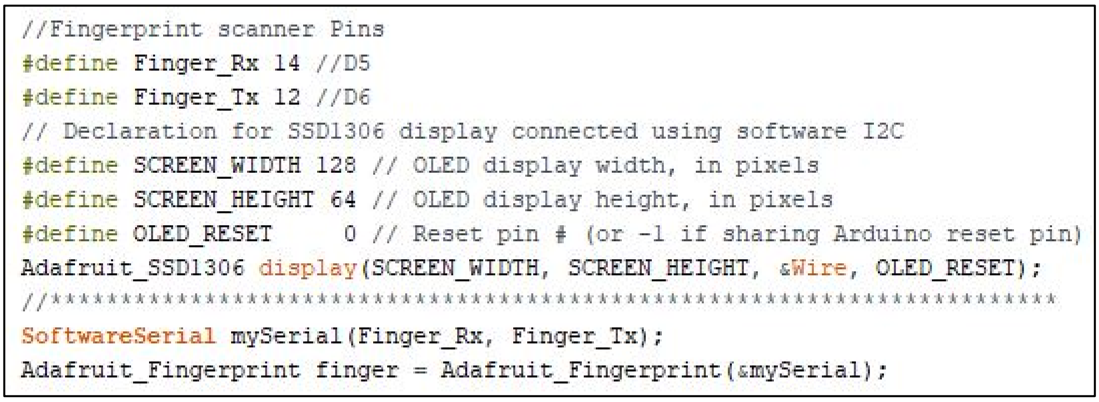
Programming codes were developed for both components, the R305 fingerprint module and the OLED display. Firstly, for the Arduino software to be able to read the inserted programming codes, a few related libraries were installed. The Adafruit libraries were installed independently through GitHub website. The Adafruit_GFX is the graphics central core of all the display outputs, providing series set of graphics interface such as lines, points, just to name a few. As for the Adafruit_SSD1306, this parameter is for the output of the OLED display. Apart from that, the parameter Adafruit_Fingerprint is to activate the fingerprint sensor features of fingerprint detection and verification.



```
//*********************************libraries*********************************
#include <SPI.h>
#include <Wire.h>
#include <WiFiClient.h>
#include <ESP8266WiFi.h>
#include <SoftwareSerial.h>
#include <ESP8266WebServer.h>
#include <ESP8266HTTPClient.h>
#include <Adafruit_GFX.h>          //https://github.com/adafruit/Adafruit-GFX-Library
#include <Adafruit_SSD1306.h>      //https://github.com/adafruit/Adafruit_SSD1306
#include <Adafruit_Fingerprint.h>  //https://github.com/adafruit/Adafruit-Fingerprint-Sensor-Library
```

Figure 20: Related libraries for Arduino Coding

After the libraries have been loaded up in the Arduino software, then the coding process started by first declaring the parameters for both of the components. Below are the coding in declaring the R305 fingerprint sensor and the OLED display.

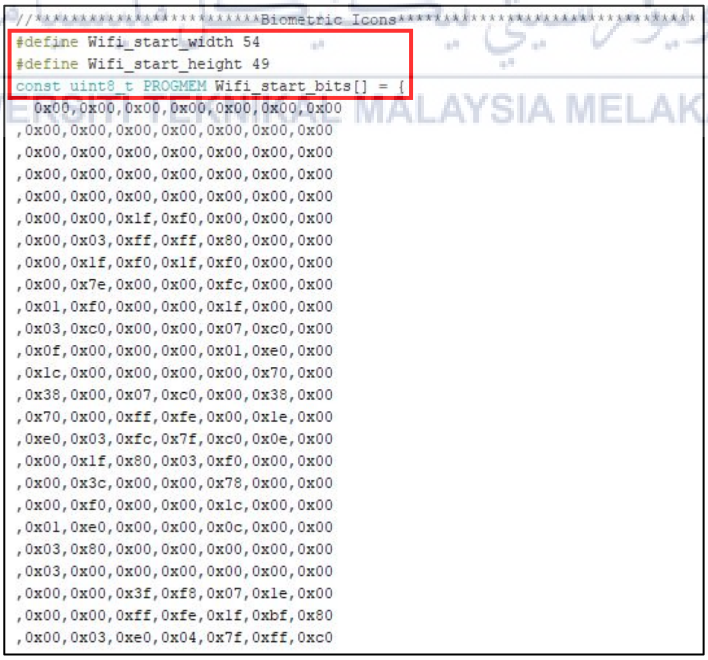


```
//Fingerprint scanner Pins
#define Finger_Rx 14 //D5
#define Finger_Tx 12 //D6
// Declaration for SSD1306 display connected using software I2C
#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels
#define OLED_RESET     0 // Reset pin # (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);
```

Figure 21: Declaring R305 fingerprint sensor and OLED display

4.2.2.2 Coding for OLED Display Icons

The function of this coding was that it showed the appeared icons on the OLED display screen. Below is the coding for the OLED display icons.

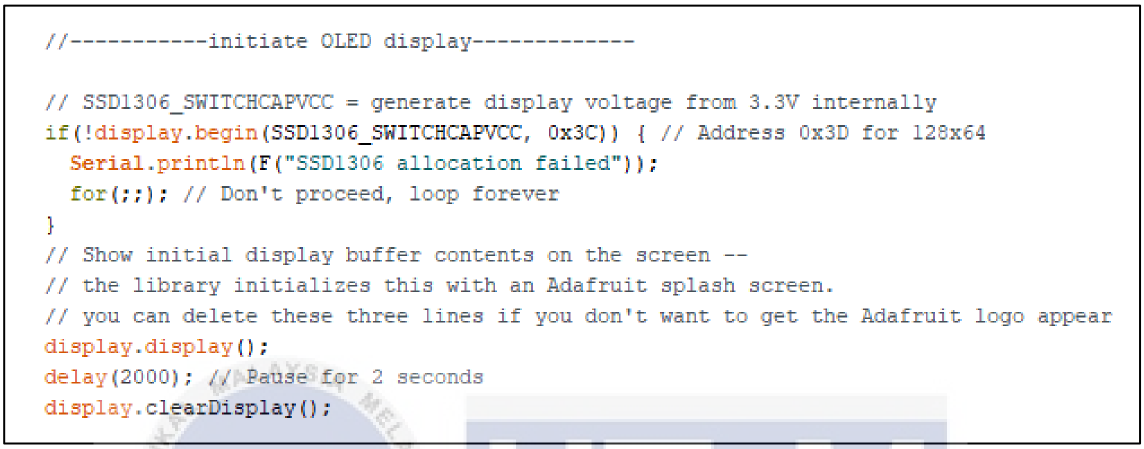


```
//*************************Biometric Icons*******************************
#define Wifi_start_width 54
#define Wifi_start_height 49
const uint8_t PROGMEM Wifi_start_bits[] = {
  0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,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
```

Figure 22: Coding for OLED display icons

4.2.2.3 Coding for Initiating OLED Display

Previously, the coding shown was to setup the OLED display icons for the icons or logos to be appeared on the screen. Meanwhile, in this part, the coding was for the initiation of the OLED display, basically to generate the display.

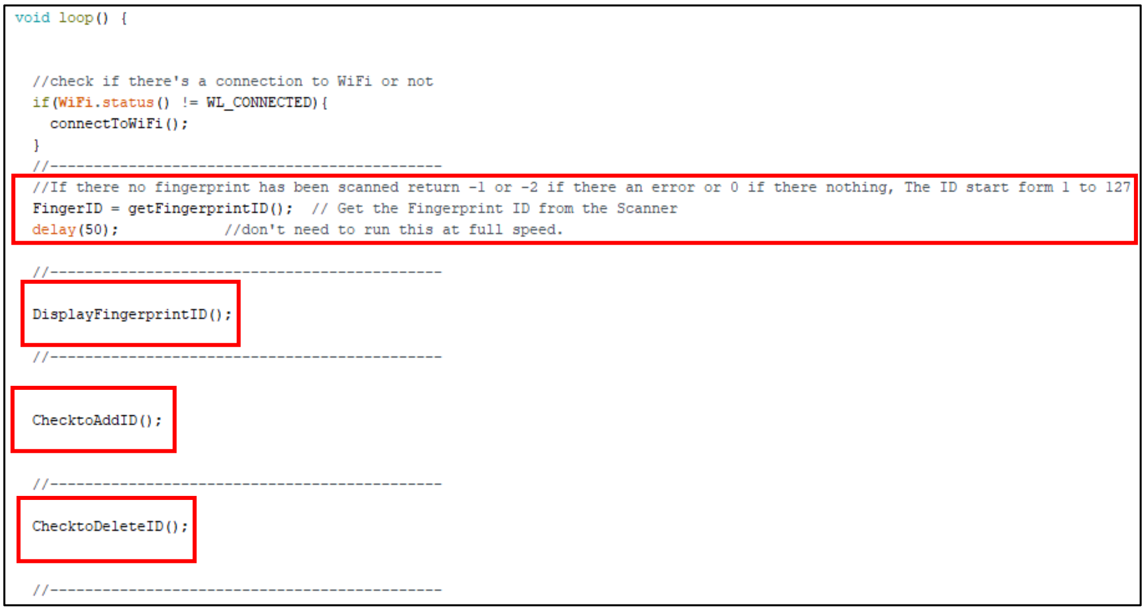


```
//-----------initiate OLED display-------------

// 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 Adafruit logo appear
display.display();
delay(2000); // Pause for 2 seconds
display.clearDisplay();
```

Figure 23: Coding for OLED display initiation

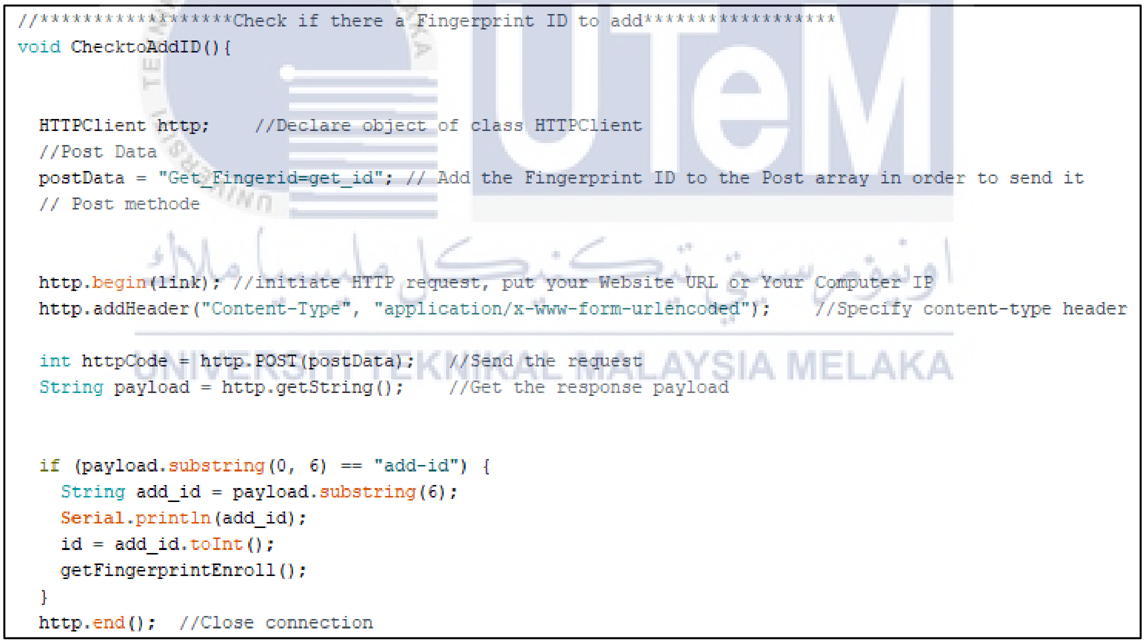
4.2.2.4 Coding for R305 Fingerprint Sensor

The main loop for the fingerprint sensor had only three functions. Figure 31 showed the coding to send the fingerprint ID data scanning phase when a user registers by scanning their fingerprint. Figure 32 showed the coding to check if there is any user ID to add into the scanner. Meanwhile, as for Figure 33, it showed the coding to check if there is any user ID to remove from the scanner.



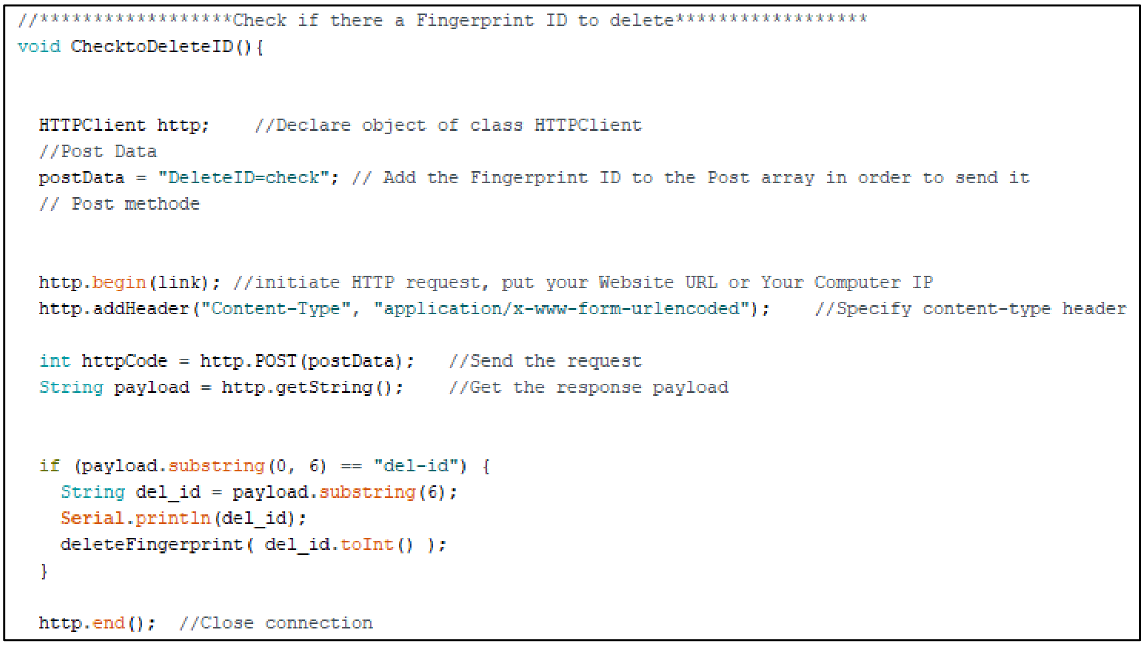
```
void loop() {


  //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();

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

Figure 24: Coding to receive Fingerprint ID



```
//******************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
```

Figure 25: Coding to observe adding a Fingerprint ID



```
//******************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
```

Figure 26: Coding to observe deleting a Fingerprint ID

4.2.3 Construction of the Database using the XAMPP Software

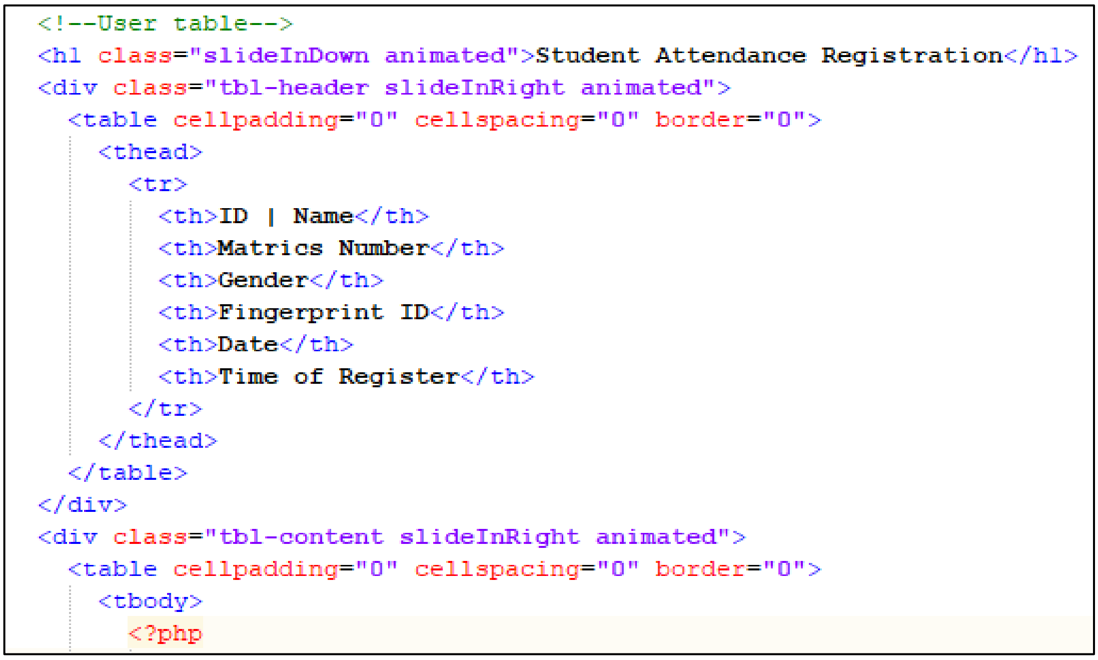
Similarly to development of the Arduino coding, the construction of the database using the XAMPP software was also vital for the project as the database displays the output interface of registered users. The constructed php scripts for the database also have to be accurate and error-free in order to activate the connectivity on the R305 fingerprint scanner to allow the scanning process.

4.2.3.1 PHP Scripts

PHP is a scripting language that deploys a process called server-side. Basically it is an operation that are carried out in a client-server interconnection in a computer network, performed by the server. In a simple term, a client is a computing device such as laptops or personal computers, meanwhile a server is a software that supplies services that executes requests from the client. The function of a server-side generally it used to create websites.

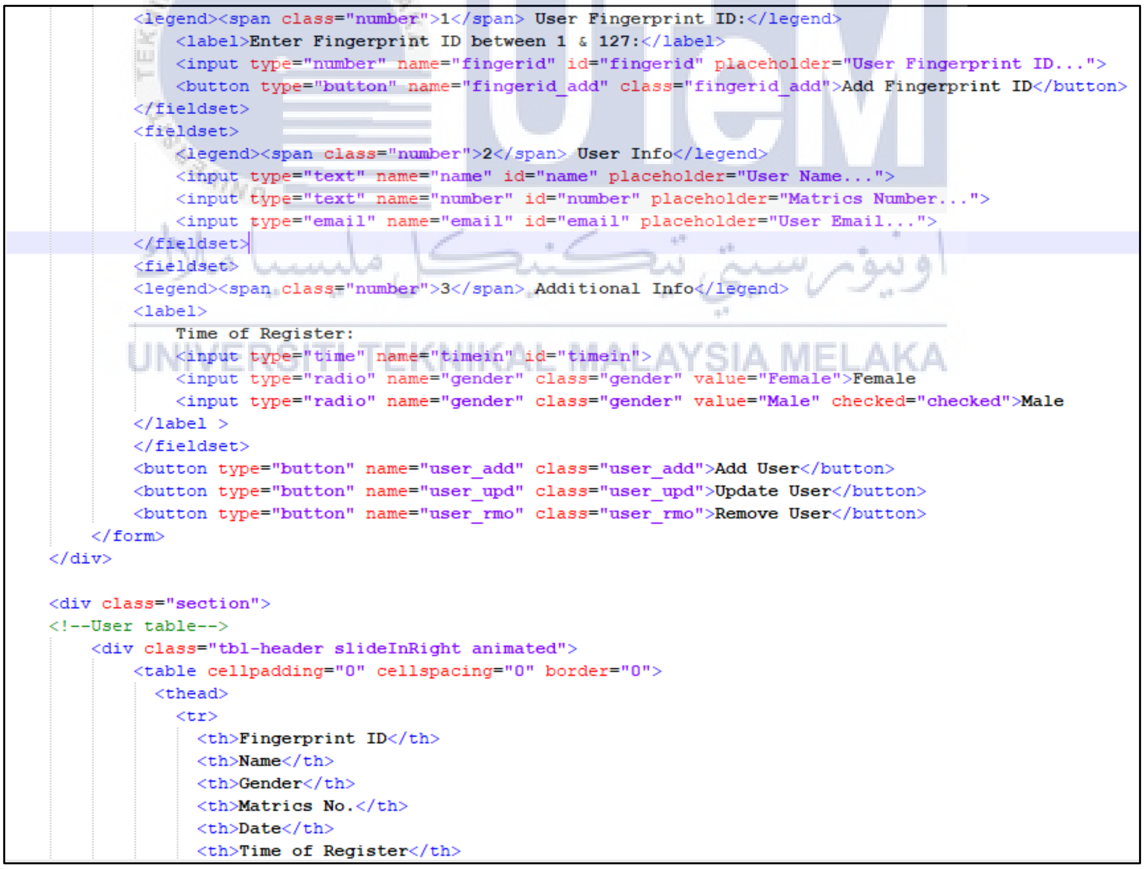
PHP Scripts Parameters of the Database

These figures determine the parameters that were setup for the wording on the database layout.



```
<!--User table-->
<h1 class="slideInDown animated">Student Attendance Registration</h1>
<div class="tbl-header slideInRight animated">
  <table cellpadding="0" cellspacing="0" border="0">
    <thead>
      <tr>
        <th>ID | Name</th>
        <th>Matrics Number</th>
        <th>Gender</th>
        <th>Fingerprint ID</th>
        <th>Date</th>
        <th>Time of Register</th>
      </tr>
    </thead>
  </table>
</div>
<div class="tbl-content slideInRight animated">
  <table cellpadding="0" cellspacing="0" border="0">
    <tbody>
      <?php
```

Figure 27: Parameters for Student Attendance Registration Layout



```
    <legend><span class="number">1</span> User Fingerprint ID:</legend>
      <label>Enter Fingerprint ID between 1 & 127:</label>
      <input type="number" name="fingerid" id="fingerid" placeholder="User Fingerprint ID...">
      <button type="button" name="fingerid_add" class="fingerid_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="Matrics Number...">
      <input type="email" name="email" id="email" placeholder="User Email...">
    </fieldset>
    <fieldset>
    <legend><span class="number">3</span> Additional Info</legend>
    <label>
      Time of Register:
      <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>Fingerprint ID</th>
       <th>Name</th>
       <th>Gender</th>
       <th>Matrics No.</th>
       <th>Date</th>
       <th>Time of Register</th>
```

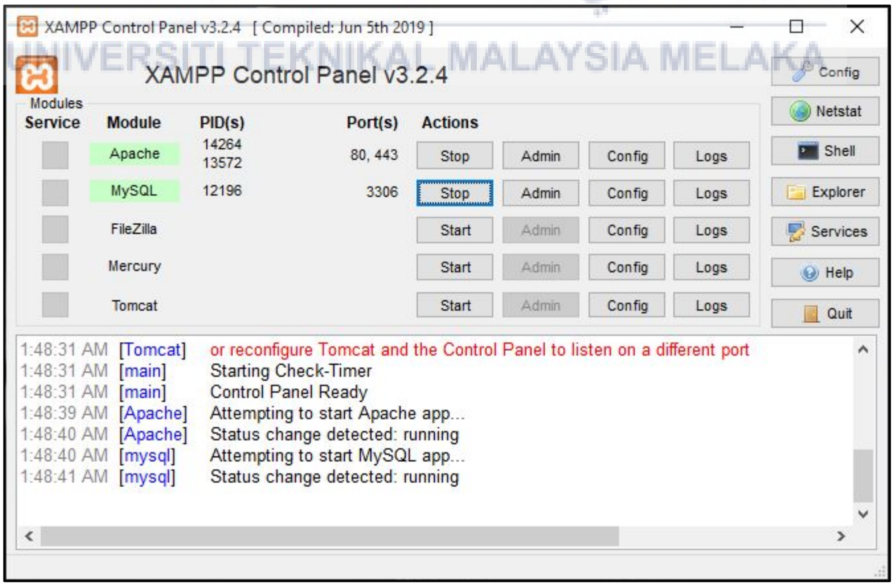
Figure 28: Parameters for Manage Users Layout

4.3 Project Analysis (Step-By-Step)

In this section, the device was experimented and tested to observe and analyze the actual working process of the project as a whole. The process outlined with the activation of the database system, the creation of the database onto a web browser, the registration processes which involved adding users' fingerprint ID, registering users' fingerprint ID with two methods which were authentication and verification methods, enrolment process and removing users' fingerprint ID. As stated in Chapter 3, the setup of the website was locally done by using the IP address of the laptop as a computer program called server to pack and store data in a localhost.

4.3.1 Activation of the Database System using the XAMPP Software

Firstly, the process started with the activation of the database system using the XAMPP software. From the XAMPP software, the Apache and MySQL services were started to activate the database system.



XAMPP Control Panel v3.2.4 [Compiled: Jun 5th 2019]

XAMPP Control Panel v3.2.4

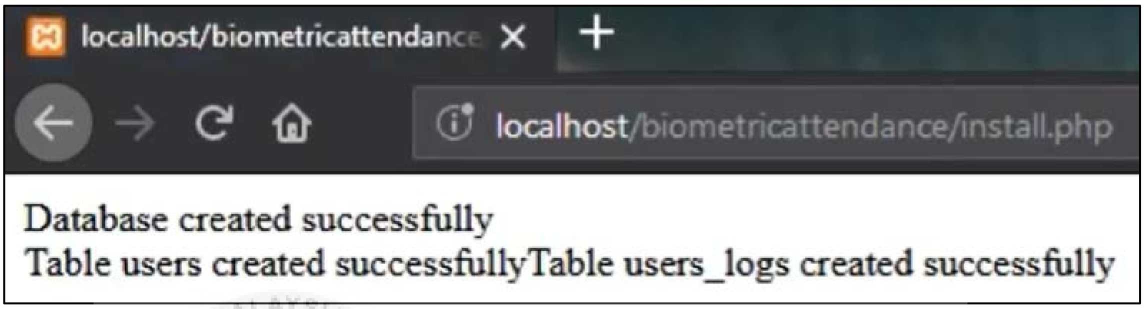
Service	Module	PID(s)	Port(s)	Actions			
	Apache	14264 13572	80, 443	Stop	Admin	Config	Logs
	MySQL	12196	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
1:48:31 AM  [Tomcat]   or reconfigure Tomcat and the Control Panel to listen on a different port
1:48:31 AM  [main]     Starting Check-Timer
1:48:31 AM  [main]     Control Panel Ready
1:48:39 AM  [Apache]   Attempting to start Apache app...
1:48:40 AM  [Apache]   Status change detected: running
1:48:40 AM  [mysql]    Attempting to start MySQL app...
1:48:41 AM  [mysql]    Status change detected: running
```

Figure 29: XAMPP software

4.3.2 Database Creation onto a Web Browser

After that, a new tab was opened using a web browser, Google Chrome to be exact, and a URL address *localhost/biometricattendance/install.php* was inserted. The URL address basically was to create the database to be installed locally onto the web browser using the computer IP address as a server.



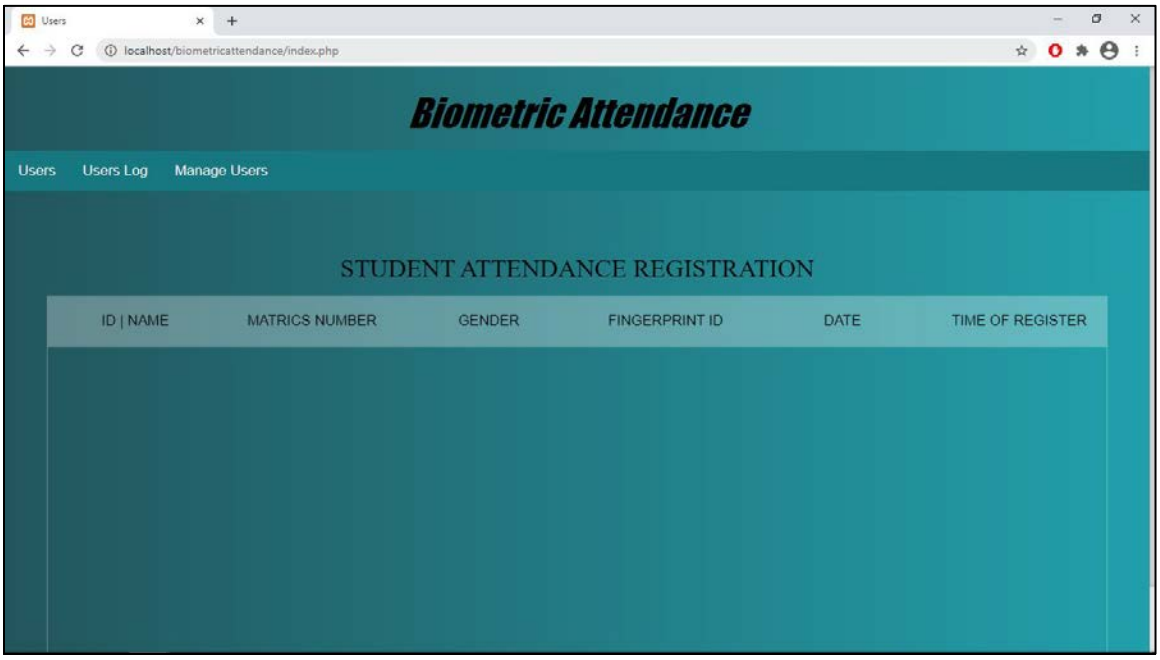
localhost/biometricattendance
localhost/biometricattendance/install.php

Database created successfully
Table users created successfullyTable users_logs created successfully

Figure 30: Database created onto the web browser

4.3.3 Database Layout

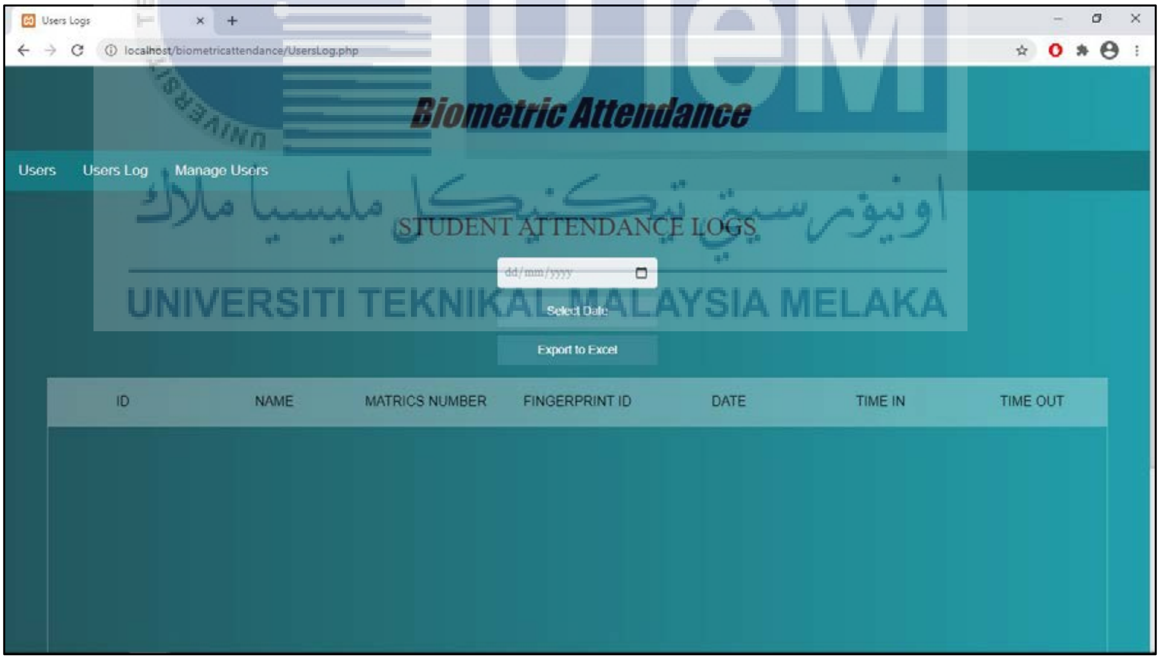
After the creation of the database, another URL address *localhost/biometricattendance/index.php* was inserted and it displayed the database layout. All of users information can be observe from this layout. There were three sections: Users; Users Log; Manage Users. The Users section displayed ‘Student Attendance Registration’ interface. All of the registered users can be viewed in this section. As for the Users Log, it showed ‘Student Attendance Logs’. All of the data of incomings and outgoings of students in a class session can be observe here. Furthermore, for the Manage Users, it showed ‘Student Attendance Registration Management’. Basically, all of the data management of students regarding adding, updating or removing students’ data can be observed and updated here.



STUDENT ATTENDANCE REGISTRATION

ID | NAME	MATRICS NUMBER	GENDER	FINGERPRINT ID	DATE	TIME OF REGISTER

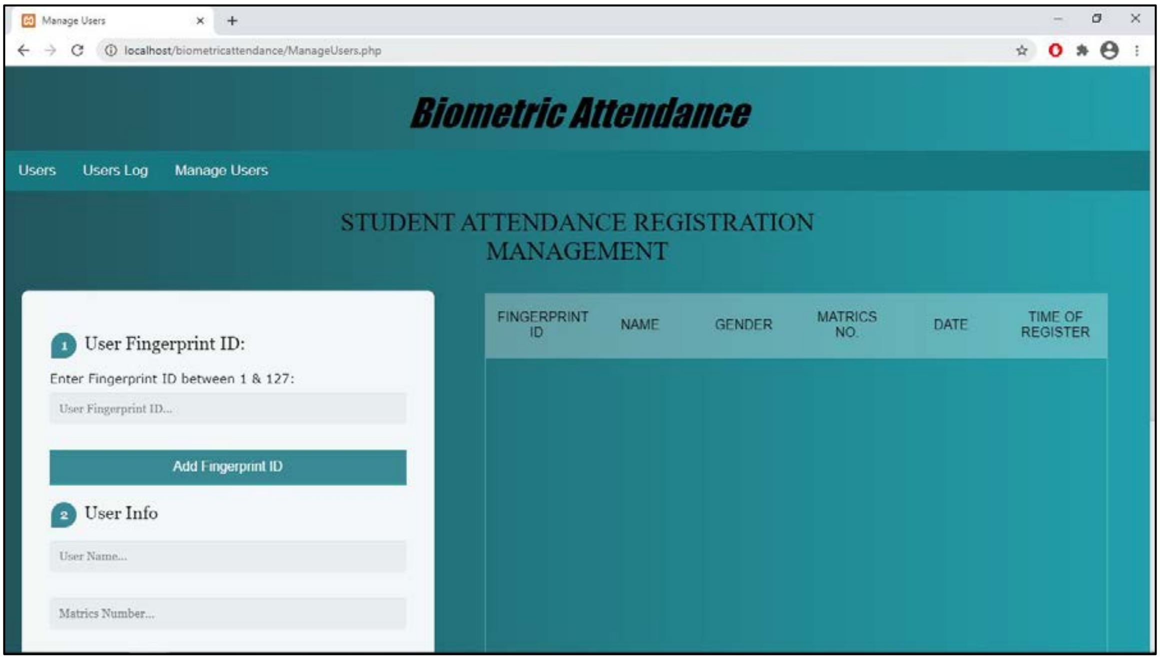
Figure 31: Student Attendance Registration (Users)



STUDENT ATTENDANCE LOGS

ID	NAME	MATRICS NUMBER	FINGERPRINT ID	DATE	TIME IN	TIME OUT

Figure 32: Student Attendance Logs (Users Log)



Biometric Attendance

STUDENT ATTENDANCE REGISTRATION MANAGEMENT

1 User Fingerprint ID:

Enter Fingerprint ID between 1 & 127:

User Fingerprint ID...

2 User Info

User Name...

Matrics Number...

FINGERPRINT ID	NAME	GENDER	MATRICS NO.	DATE	TIME OF REGISTER

Figure 33: Student Attendance Registration Management (Manage Users)

4.3.4 Registration Process of a User’s Fingerprint ID

In this part, the components were powered up by a supplied voltage using a power bank to the NodeMCU ESP8266 board. The OLED display showed the Wifi connecting message ‘Connecting to alshahab’ where ‘alshahab’ is the Wifi username. After the Wifi is connected, the OLED display showed a ‘Connected’ message and a fingerprint icon appeared.

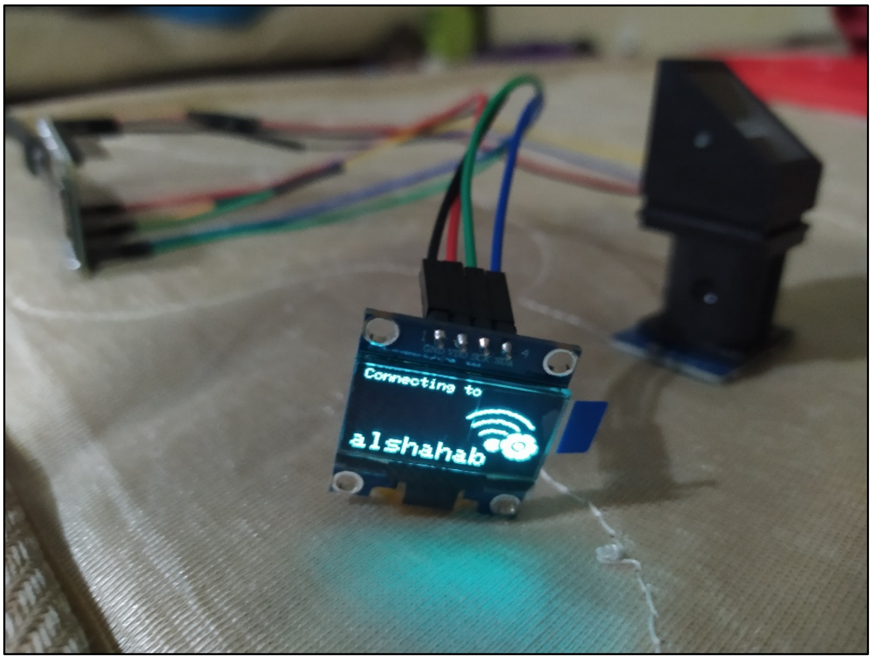


Figure 34: Connecting to Wifi

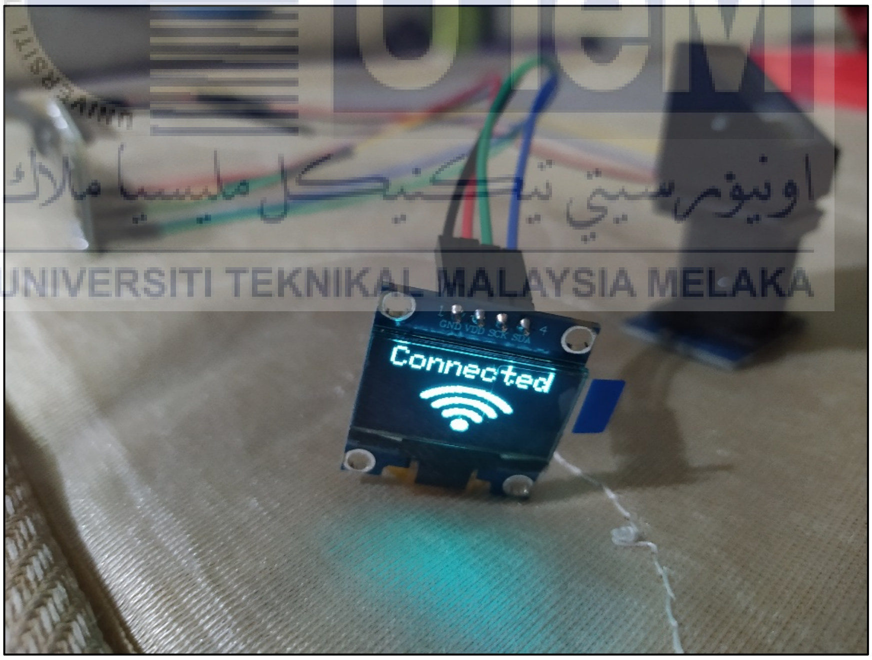


Figure 35: Wifi Connected

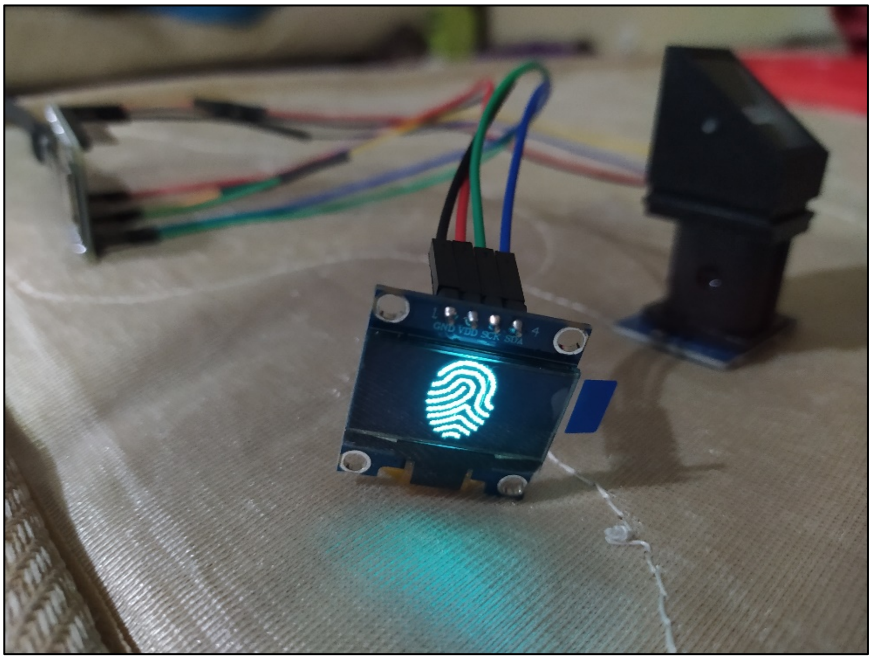
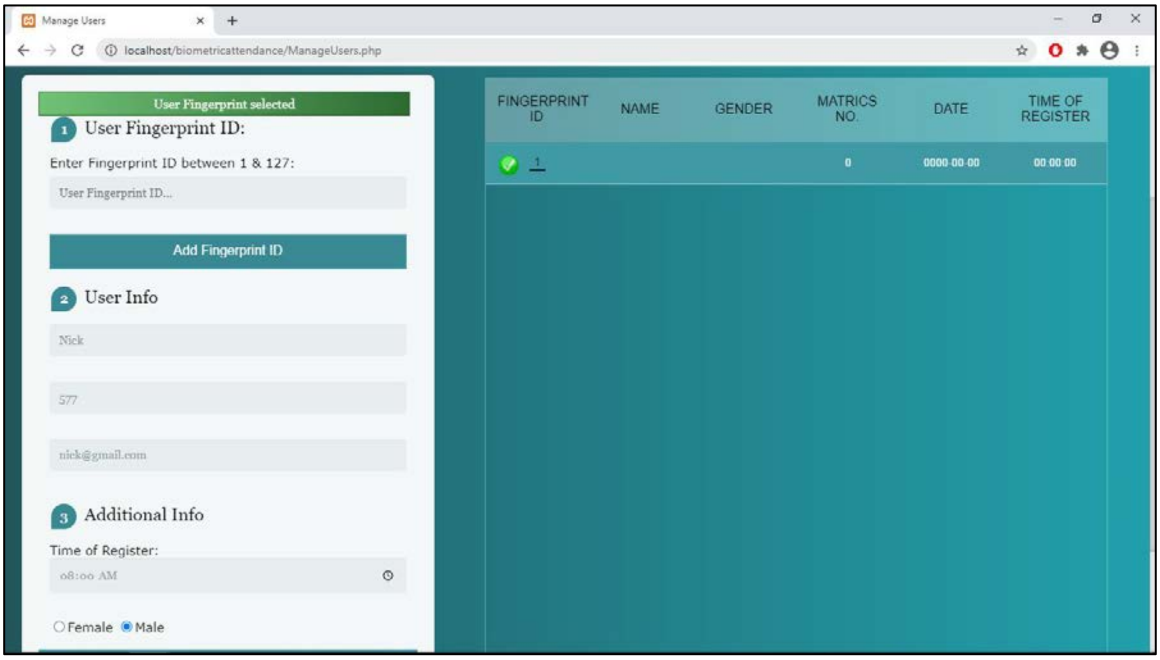


Figure 36: Fingerprint Icon

Adding and Registering a User’s Fingerprint ID

At the web browser database, Fingerprint ID number ‘1’ was inserted and added. Once it has been added, the fingerprint ID number 1 was selected, thus the displaying message of ‘User Fingerprint Selected’.



User Fingerprint selected

1 User Fingerprint ID:

Enter Fingerprint ID between 1 & 127:

User Fingerprint ID...

Add Fingerprint ID

2 User Info

Nick

577

nick@gmail.com

3 Additional Info

Time of Register:

08:00 AM

○ Female ● Male

FINGERPRINT ID	NAME	GENDER	MATRICS NO.	DATE	TIME OF REGISTER
✓ 1			0	0000-00-00	00:00:00

Figure 37: User's Fingerprint ID has been added

Once the fingerprint ID has been added, automatically the fingerprint scanner was ready to scan and receive a user's fingerprint traits. A 'scanning' message also was displayed on the OLED display.

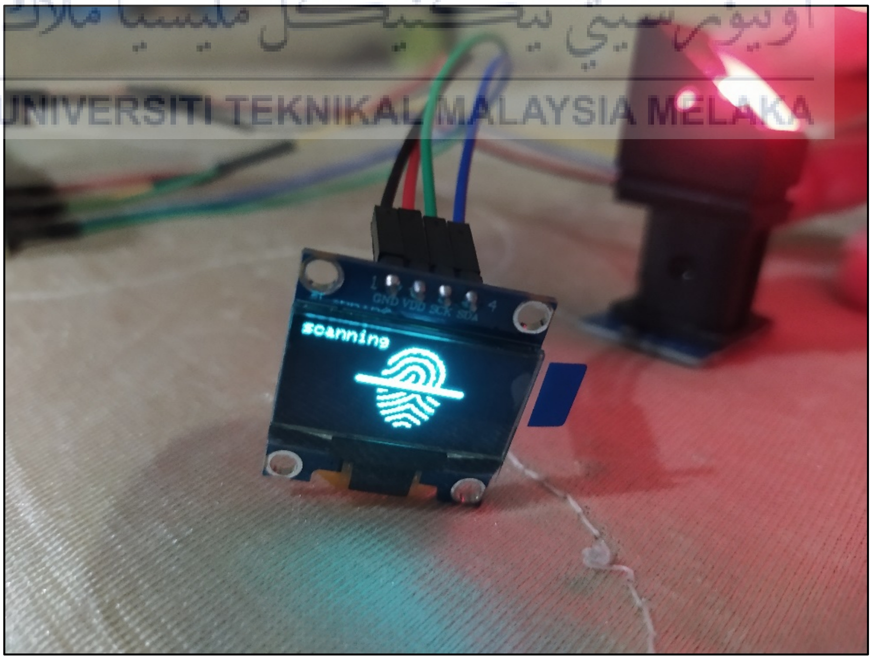


Figure 38: Fingerprint Scanner ready to scan

The user placed his finger onto the scanner. During this process, as stated earlier, there were two methods that are needed to complete the registration process: authentication and verification. The scanner had to scan his finger two times to complete the registration. First, the user placed his finger to scan for the first time for authentication. A ticked fingerprint icon is displayed indicating the authentication process was successful. Then, a message ‘Remove finger’ was displayed to notify the user to remove his finger. Next, the ‘scanning’ message displays again signalling the user to scan his thumb for second time for verification and the ticked fingerprint icon displayed again to notify user that the verification process was successful. Thus, the registration is completed.

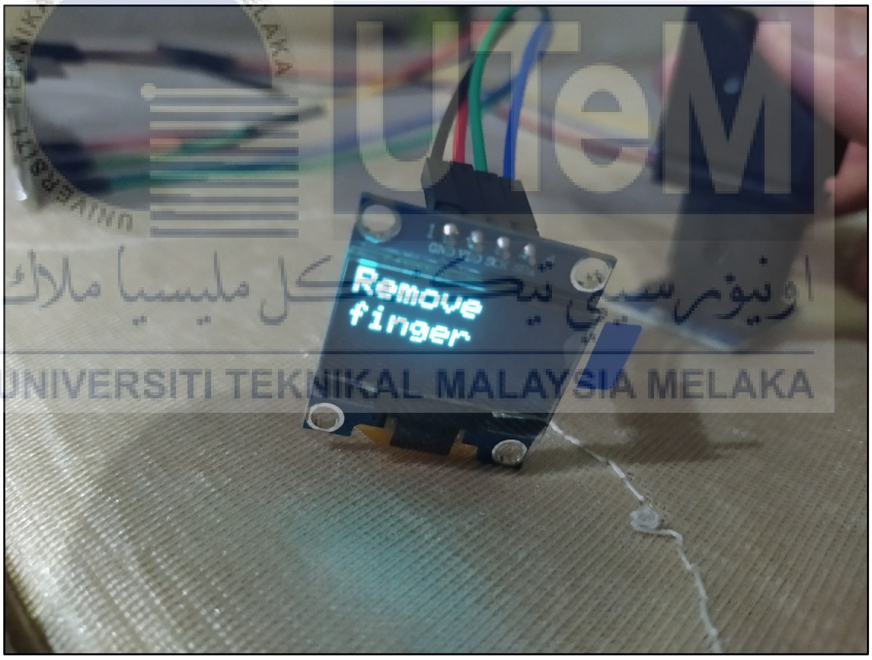


Figure 39: First scan

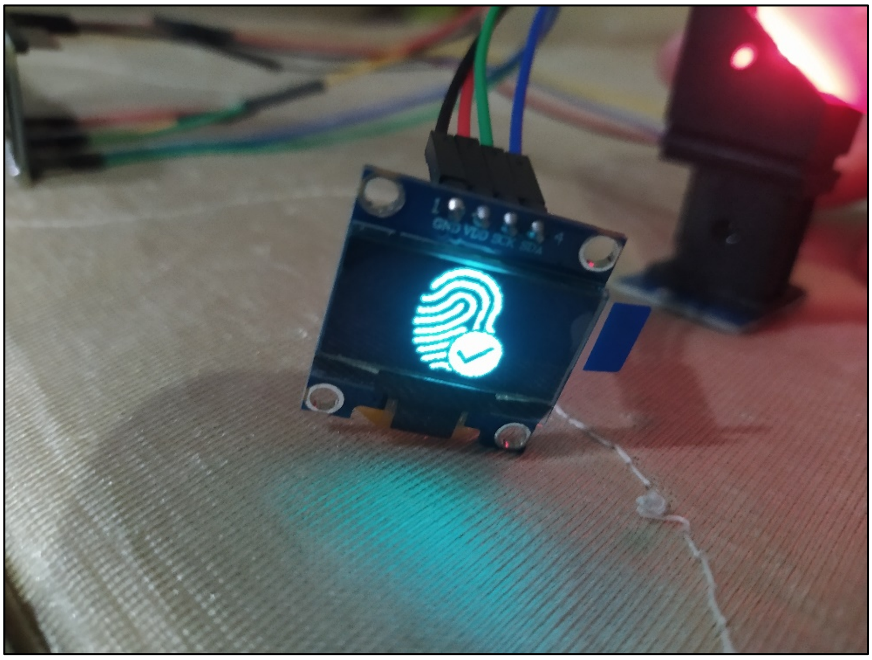


Figure 40: Authentication successful

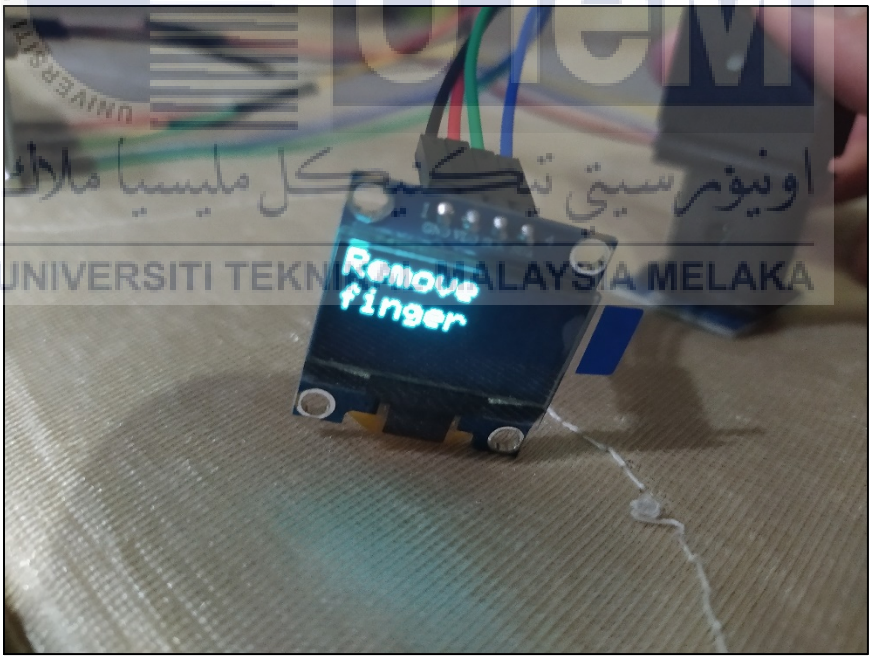


Figure 41: Second scan

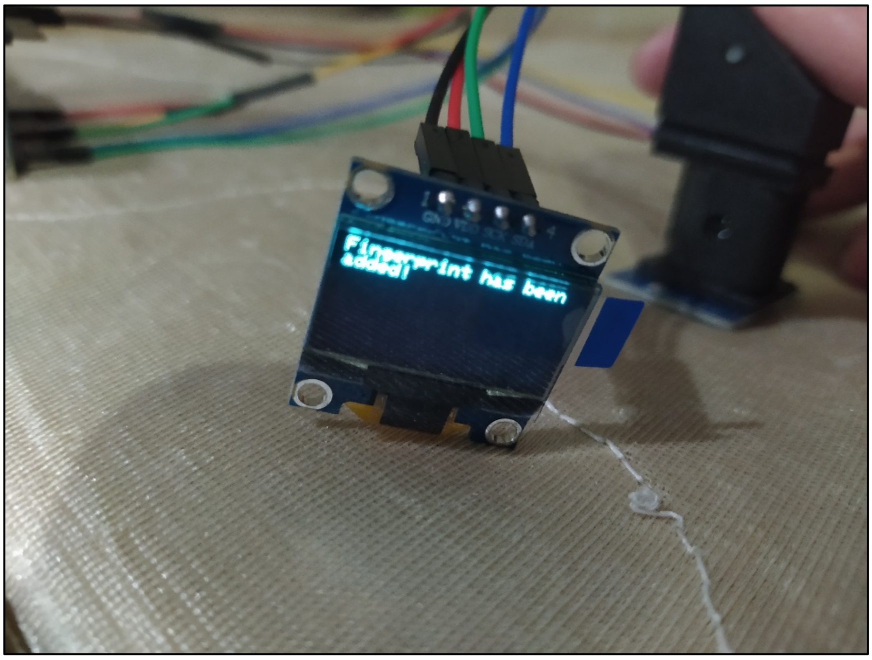
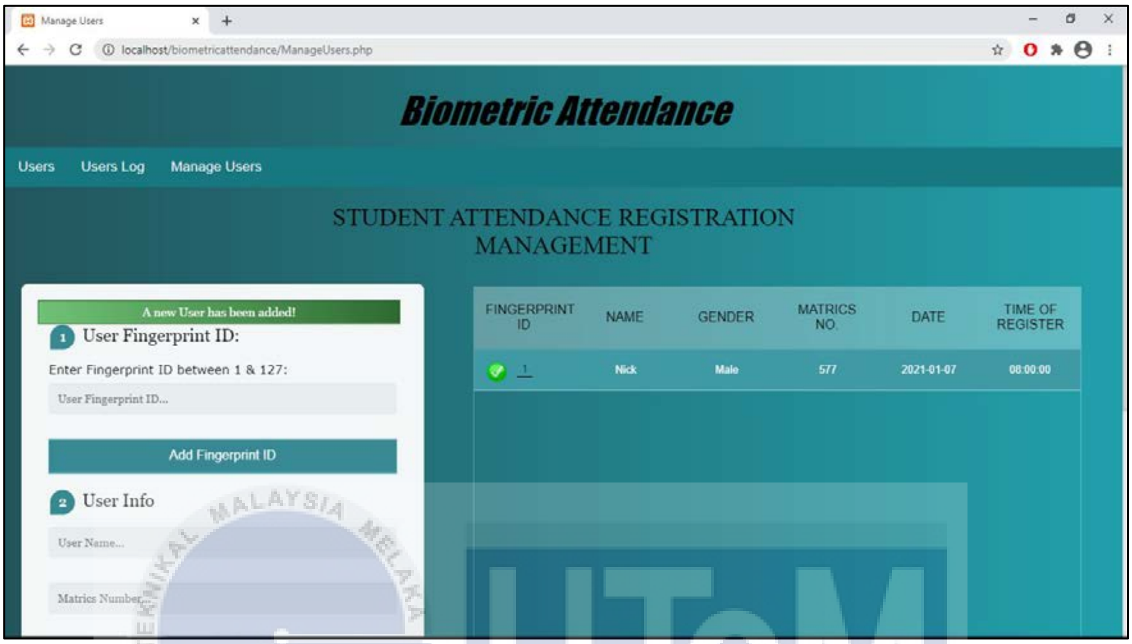


Figure 42: ‘Fingerprint has been added’ message displayed

Once the fingerprint registration is completed, the user’s information was updated manually by inserting the name, matrics number, gender, email and time of register.



Biometric Attendance

STUDENT ATTENDANCE REGISTRATION MANAGEMENT

A new User has been added!

1 User Fingerprint ID:

Enter Fingerprint ID between 1 & 127:

User Fingerprint ID...

2 User Info

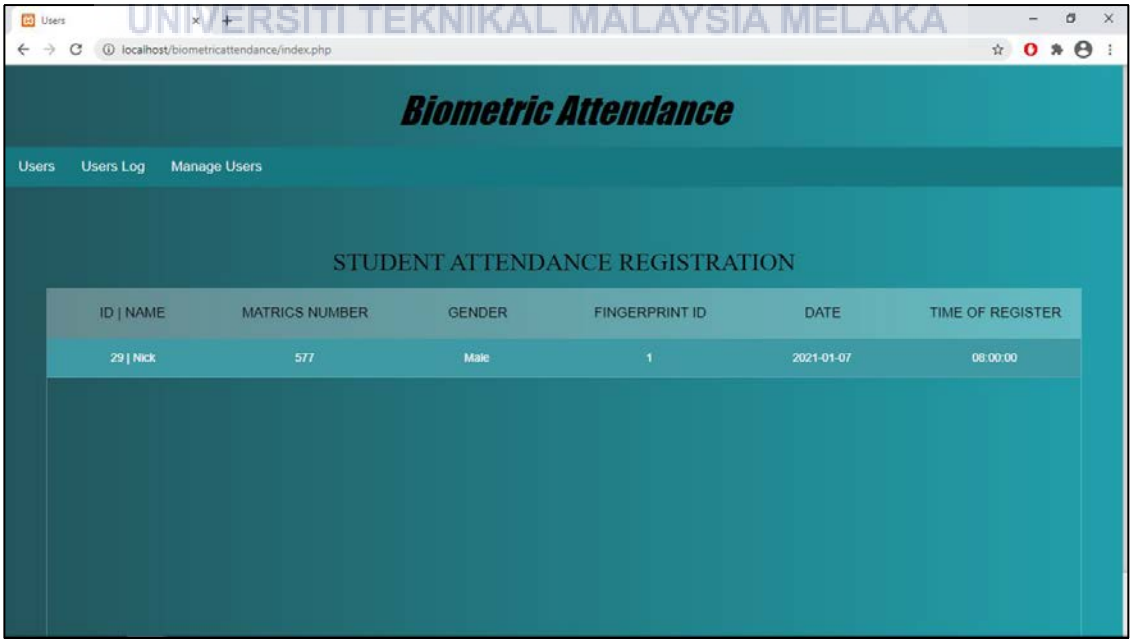
User Name...

Matrics Number...

FINGERPRINT ID	NAME	GENDER	MATRICS NO.	DATE	TIME OF REGISTER
✓ 1	Nick	Male	577	2021-01-07	08:00:00

Figure 43: Manually updated user’s information

After that, the registration is completed with the user’s fingerprint ID and information.



Biometric Attendance

STUDENT ATTENDANCE REGISTRATION

ID | NAME	MATRICS NUMBER	GENDER	FINGERPRINT ID	DATE	TIME OF REGISTER
29 | Nick	577	Male	1	2021-01-07	08:00:00

Figure 44: Registration completed

4.3.5 Enrolment of Class Session

Once the registration has been fully completed, then only the enrolment of a class session could take place. The enrolment process started with the user scan his finger for the first time to clock-in to class. The attendance status can be observed at the 'Student Attendance Logs' section. After that, when the user finishes the class, the user then scan his finger for the second time to clock-out from the class. Therefore, the attendance for the day was recorded.

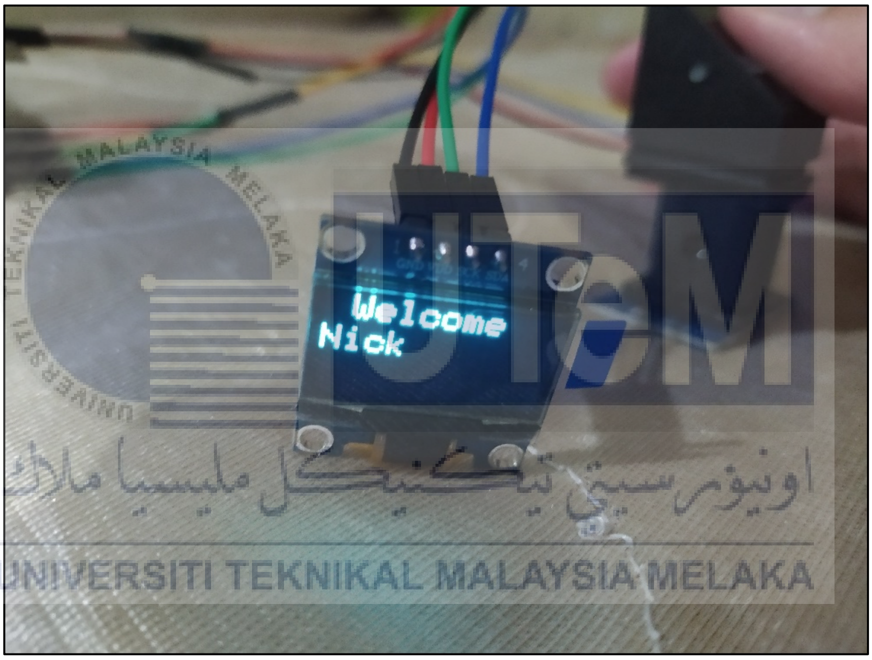


Figure 45: Check-In to Class

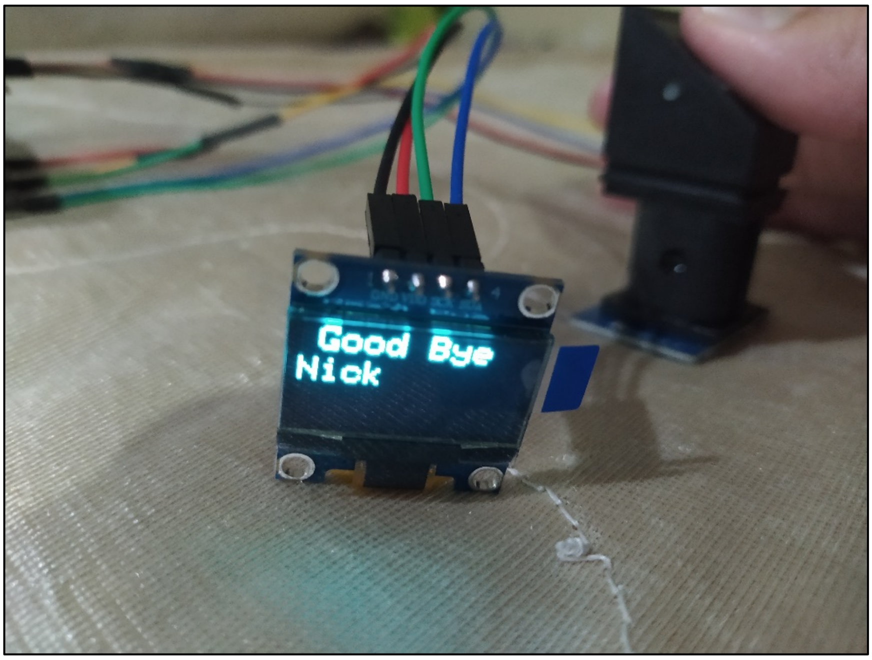
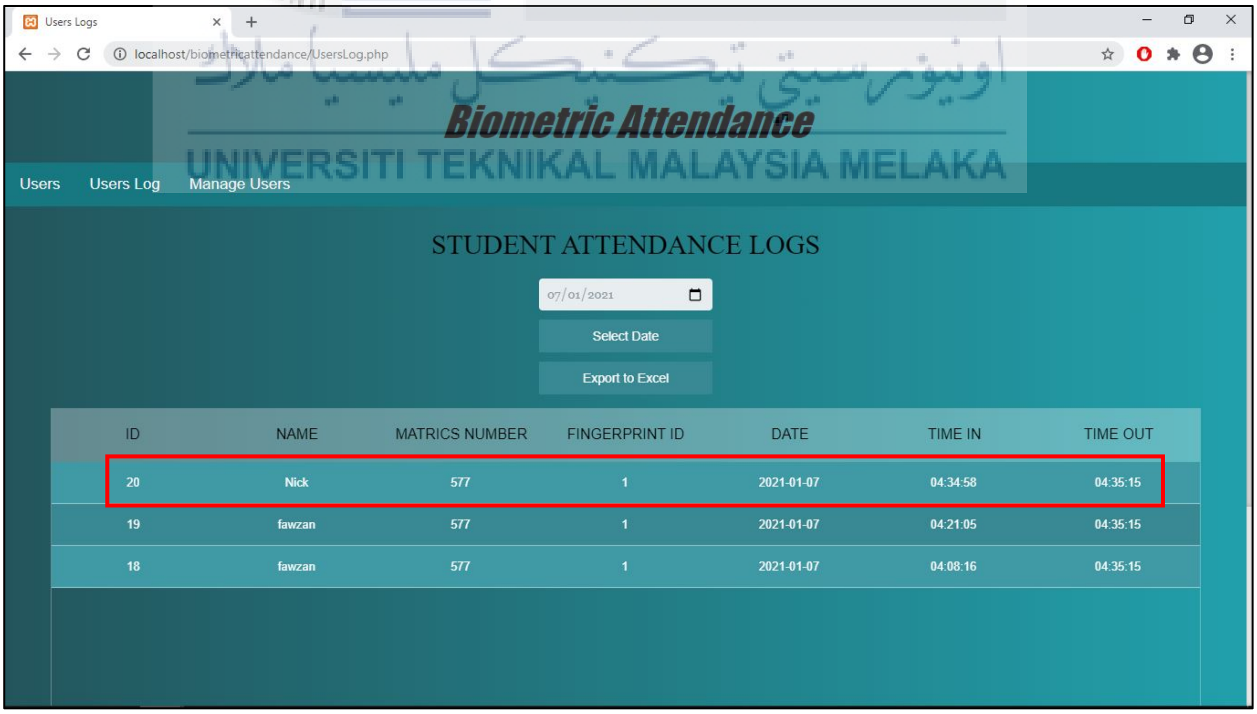


Figure 46: Check-Out from Class

Below is the recorded attendance in a particular class for the day.



Biometric Attendance

Users Users Log Manage Users

STUDENT ATTENDANCE LOGS

07/01/2021

Select Date

Export to Excel

ID	NAME	MATRICS NUMBER	FINGERPRINT ID	DATE	TIME IN	TIME OUT
20	Nick	577	1	2021-01-07	04:34:58	04:35:15
19	fawzan	577	1	2021-01-07	04:21:05	04:35:15
18	fawzan	577	1	2021-01-07	04:08:16	04:35:15

Figure 47: Recorded Attendance

4.4 Final Product of the Project

After ensuring all of the technical parts, which are designing a hardware casing concept and testing the actual working process of the project, then the final product designing process is carried out to complete the overall process of the project.

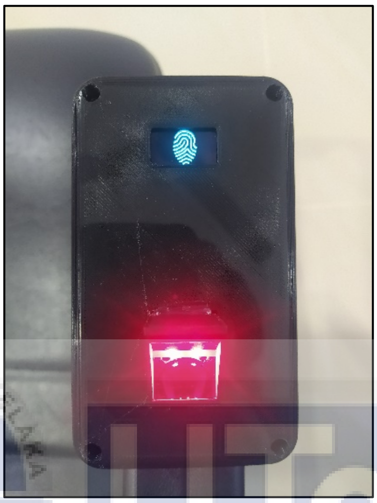


Figure 48: Finished Product of the Device

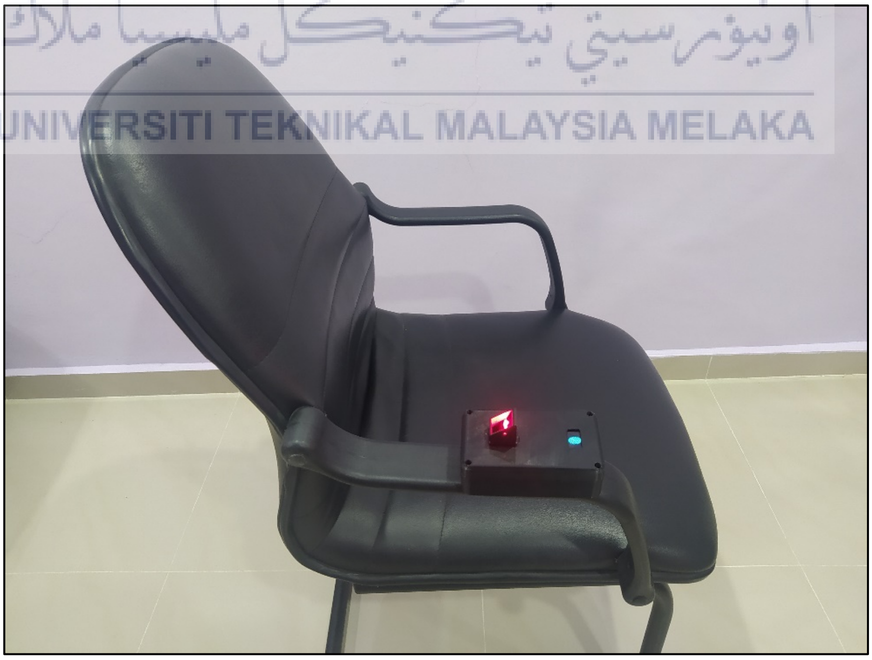


Figure 49: Final Look of the Product

4.5 Benchmarking the Smart Biometric Chair for Classroom Attendance System with an Existing Biometric Attendance System

In this context, benchmarking is a process of analyzing the performance of the newly designed and developed Smart Biometric Chair for Classroom Attendance System with an existing biometric attendance system. For experimental purposes, the biometric product used for this analysis is the OEM's Fingerprint Attendance Device for Office.

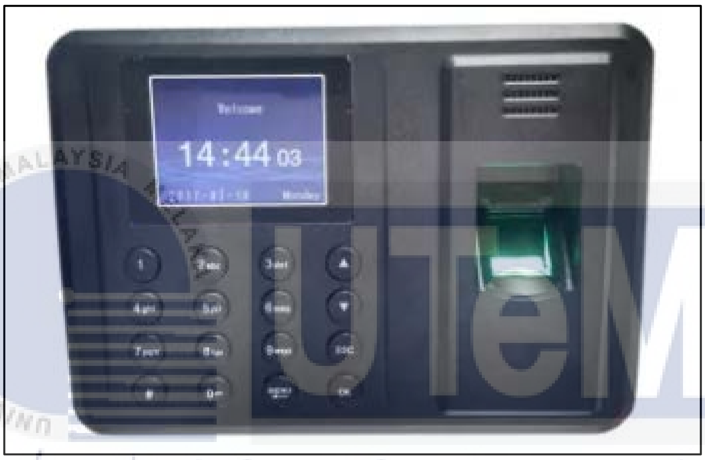


Figure 50: OEM's Fingerprint Attendance Device for Office

Detailed Explanation on the Comparison

Based on Table 4, it showed the comparison between two products: the Smart Biometric Chair for Classroom Attendance System and the OEM's Fingerprint Attendance Device for Office. Basically, the table depicts the specifications comparison between both of the products to observe which one is better.

Firstly, both of their features are fingerprint-based attendance system. Next, the fingerprint capacity for the OEM brand has 1024 users, while the Smart Biometric Chair has only 127 users. This is a huge gap in terms of user capacity between these two products. Moving on to the power supply, the Smart Biometric Chair uses a 5000mAh portable power bank, while the OEM brand uses a wired charging adapter. As for the display, the Smart Biometric Chair uses an OLED display, while the OEM brand uses an LCD display. Furthermore, for both fingerprint accuracy rate (FAR) and fingerprint response rate (FRR), the Smart Biometric Chair has an accuracy rate less than 0.001% and less than 0.1% response rate. Meanwhile for the OEM brand, the status of both of the specifications are not available. This clearly indicates that the Smart Biometric Chair is more reliable in terms of FAR and FRR. Lastly, in terms of the pricing range, the price for the Smart Biometric Chair is RM 147.45, meanwhile for the OEM brand is RM 119.00.

Finally, to summarize things up, it can be concluded that the performance in terms of their effectiveness and reliability, really depends on the needs and demands of the clients. If an organization requires large amount of users, then they can opt for the OEM brand. Meanwhile, if an organization is looking for a fast and effective attendance system, then the Smart Biometric Chair is suitable for them.

Table 4

Comparison between two products.

Device / Specifications	Smart Biometric Chair for Classroom Attendance System	OEM's Fingerprint Attendance Device for Office
Features	Fingerprint-based	Fingerprint-based
Fingerprint Capacity	127 users	1024 users
Power Supply	5000mAh Portable Power Bank	Wired Charging Adapter
Display	OLED Display	LCD Display
Fingerprint Accuracy Rate (FAR)	<0.001%	N/A
Fingerprint Response Rate (FRR)	<0.1%	N/A
Price (RM)	RM 147.45	RM 119.00

CHAPTER 5

CONCLUSION AND FUTURE RECOMMENDATIONS

5.1 Introduction

In this part chapter, it discussed the conclusion of the overall design and development of the project dated back to innovation of ideas to design this project until its completion. It covered the accomplished objectives, comparison of literature reviews, its methodology, and the highlights of its working process as well as the benchmarking with an existing product. On top of that, this part chapter also outlined the future recommendations for this project as the project has potentials and much room for improvements.

5.2 Conclusion

In conclusion, this project entitled Design and Implementation of Smart Biometric Chair for Classroom Attendance System has fulfilled and met all of the objectives.

The process started from construction of ideas, brainstorming in searching for a suitable innovative ideas that basically could contributes to the society in improving their quality of life. After the title has been finalized, then intensive researches from multiple sources were conducted. Multiple peer-reviewed credible literature reviews were studied to obtain the full concept of biometric projects in making for this project. Next, proceeding to the methodology of the project. The expected hardware components and programming software were considered in the making of this project. After all of the hardware and software were taken into consideration on what components and programming software that will be used for the project, then the processes of construction of the circuit and the development of the codings started. After circuit construction and troubleshooting, then it proceeded into the hardware design and its testing phase. After the testing phase is completed, then an analysis of the result was conducted. After completing the result analysis, then the project can be concluded and also has a discussion on its future recommendations in improving this project to make it better in the later future.

The main key highlights of this project was that it is a chair-bounded device that supports fingerprint-based attendance system. The chair-bounded method is more ergonomically efficient than the standard paper-based attendance system method.

5.3 Future Recommendations

There are a few recommended innovations for this project that could make it more efficient than it already is. The first recommendation would be the type of the fingerprint scanner that is able to store more users. Currently, this product could only store up to only 127 users. Therefore, if an innovation involves using another huge storage capacity fingerprint scanner is a plus.

The second recommendation is the construction of a full-fledged database that is able to share URL address to a certain people. It would be more convenient if the attendance can be shared to a certain people.

The third recommendation would be an embedded fingerprint scanner into a chair armrest. The chair would be more ergonomic as the scanner does not interfere with a user’s arm or hand.

These recommendations would certainly be more beneficial to the life quality of society as it contributes in making the environment more conducive and overall improving the fluidity of society’s working life.

GANTT CHART PSM I

PROJECT TASK	PLANNING OF THE PROJECT (WEEK)														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Chapter 1 Materials Research	🟩	🟩	🟩	🟩				🟥							
Chapter 1: Introduction	🟩	🟩	🟩	🟩				🟥							
i) Background & Objective		🟩	🟩	🟩				🟥							
ii) Problem Statement			🟩	🟩				🟥							
iii) Scope of Research			🟩	🟩	🟩			🟥							
Chapter 2 Materials Research					🟩	🟩	🟩	🟥	🟩	🟩	🟩	🟩	🟩		
Chapter 2: Literature Review					🟩	🟩	🟩	🟥	🟩	🟩	🟩	🟩	🟩		
i) Journal Comparison								🟥		🟩	🟩	🟩			
Chapter 3: Methodology								🟥		🟩	🟩	🟩	🟩		
i) Expected result								🟥		🟩	🟩	🟩	🟩		
Preparing for PSM I Presentation								🟥						🟩	🟩

REFERENCES

[1] Arunkumar, L. and Arun Raja, A. (2015) ‘Biometrics Authentication Using Raspberry Pi’, *International Journal for Trends in Engineering & Technology*.

[2] Devikar, Piyush, Krishnamoorthy Ajit, Bhanage, Aditya, M. S. C. (2016) ‘IoT Based Biometric Attendance System’, *International Journal of Advanced Research in Computer and Communication Engineering*. doi: 10.17148/IJARCCE.

[3] Jiang, X. (2015) ‘Fingerprint Classification’, in *Encyclopedia of Biometrics*. doi: 10.1007/978-1-4899-7488-4_56.

[4] Kaium Khan, M. A. *et al.* (2019) ‘A Portable and Less Time Consuming Wireless Biometric Attendance System for Academic Purpose Using NodeMCU Microcontroller’, in *2018 21st International Conference of Computer and Information Technology, ICCIT 2018*. doi: 10.1109/ICCITECHN.2018.8631954.

[5] Kamaraju, M. and Kumar, P. A. (2015) ‘Wireless fingerprint attendance management system’, in *Proceedings of 2015 IEEE International Conference on Electrical, Computer and Communication Technologies, ICECCT 2015*. doi: 10.1109/ICECCT.2015.7226163.

[5] Kamelia, L. *et al.* (2018) ‘Real-Time Online Attendance System Based on

Fingerprint and GPS in the Smartphone', in *Proceeding of 2018 4th International Conference on Wireless and Telematics, ICWT 2018*. doi: 10.1109/ICWT.2018.8527837.

[6] Krishnamurthi, K. *et al.* (2015) 'Fingerprint Based Attendance System', *IJARCCE*. doi: 10.17148/ijarcce.2015.43149.

[7] Monday, H. N. *et al.* (2019) 'Enhanced attendance Management System: A Biometrics System of Identification Based on Fingerprint', in *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference, IEMCON 2018*. doi: 10.1109/IEMCON.2018.8614776.

[8] Sumit Singh, D. K. Y. (2015) 'Fingerprint Based Attendance System Using Microcontroller and LabView', *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*. doi: 10.15662/ijareeie.2015.0406029.

[9] Zainal, N. I. *et al.* (2014) 'Design and development of portable classroom attendance system based on Arduino and fingerprint biometric', in *2014 the 5th International Conference on Information and Communication Technology for the Muslim World, ICT4M 2014*. doi: 10.1109/ICT4M.2014.7020601.

[10] Vasilieva, N., 2014. *History At Your Fingertips*. [online] CABLOOK online magazine: read, watch, listen. Available at:

<http://www.cablook.com/mixlook/istoriya-na-konchikah-paltsev/> [Accessed 12 June 2020].

[11] Npg.org.uk. 2020. *Sir Francis Galton - National Portrait Gallery*. [online] Available at: <https://www.npg.org.uk/collections/search/portrait/mw127193> [Accessed 12 June 2020].

[12] LIBRARY, S., 2020. *Edward Henry, British Fingerprint Pioneer - Stock Image - H408/0228*. [online] Science Photo Library. Available at: <https://www.sciencephoto.com/media/225773/view/edward-henry-british-fingerprint-pioneer> [Accessed 12 June 2020].

[13] Sawe, B., 2020. *Henry Faulds – Father Of Fingerprints*. [online] WorldAtlas. Available at: <https://www.worldatlas.com/articles/henry-faulds-important-figures-throughout-history.html> [Accessed 12 June 2020].

[14] Barra, Silvio., (2016) 'Design of a Multi-Biometric Platform, Based on Physical Traits and Physiological Measures: Face, Iris, Ear, ECG and EEG' 7 Mar. 2016.

APPENDIX

```
#include <SPI.h>

#include <Wire.h>

#include <WiFiClient.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>



#define Finger_Rx 14

#define Finger_Tx 12


#define SCREEN_WIDTH 128

#define SCREEN_HEIGHT 64

#define OLED_RESET     0

Adafruit_SSD1306     display(SCREEN_WIDTH,     SCREEN_HEIGHT,     &Wire,
          OLED_RESET);



SoftwareSerial mySerial(Finger_Rx, Finger_Tx);
```

```
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);


const char *ssid = "alshahab";

const char *password = "0192797905";


String postData ;

String link = "http://192.168.0.168/biometricattendance/getdata.php";

int FingerID = 0;

uint8_t id;


#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
,0x00,0x00,0x1f,0xe7,0xf3,0xf9,0xfc
,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
,0x00,0x00,0x00,0x3f,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,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
,0x00,0x00,0x00,0x7f,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
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
};

void setup() {

  Serial.begin(115200);



  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("SSD1306 allocation failed"));
    for(;;);
  }

  display.display();
  delay(2000);
  display.clearDisplay();
```

```
connectToWiFi();



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...");

}

void loop() {

 if(WiFi.status() != WL_CONNECTED){
  connectToWiFi();
 }

 FingerID = getFingerprintID();
 delay(50);


 DisplayFingerprintID();



 ChecktoAddID();



 ChecktoDeleteID();
```

```
}
void DisplayFingerprintID(){
 if (FingerID > 0){
  display.clearDisplay();
  display.drawBitmap( 34, 0, FinPr_valid_bits, FinPr_valid_width, FinPr_valid_height,
      WHITE);
  display.display();

  SendFingerprintID( FingerID );

 }

 else if (FingerID == 0){
  display.clearDisplay();
  display.drawBitmap( 32, 0, FinPr_start_bits, FinPr_start_width, FinPr_start_height,
      WHITE);
  display.display();
 }

  else if (FingerID == -1){
  display.clearDisplay();
  display.drawBitmap( 34, 0, FinPr_invalid_bits, FinPr_invalid_width,
      FinPr_invalid_height, WHITE);
  display.display();
  }
   else if (FingerID == -2){
   display.clearDisplay();
   display.drawBitmap( 32, 0, FinPr_failed_bits, FinPr_failed_width,
       FinPr_failed_height, WHITE);
   display.display();
  }
}

void SendFingerprintID( int finger ){

  HTTPClient http;

  postData = "FingerID=" + String(finger);


  http.begin(link);
  http.addHeader("Content-Type", "application/x-www-form-urlencoded");

  int httpCode = http.POST(postData);
  String payload = http.getString();

  Serial.println(httpCode);
  Serial.println(payload);
  Serial.println(postData);
  Serial.println(finger);



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

   display.clearDisplay();
   display.setTextSize(2);
   display.setTextColor(WHITE);
   display.setCursor(15,0);
   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);

   display.clearDisplay();
   display.setTextSize(2);
   display.setTextColor(WHITE);
   display.setCursor(10,0);
   display.print(F("Good Bye"));
   display.setCursor(0,20);
   display.print(user_name);
```

```
    display.display();
  }
  delay(1000);

  postData = "";
  http.end();
}

int getFingerprintID() {
  uint8_t p = finger.getImage();
  switch (p) {
    case FINGERPRINT_OK:

      break;
    case FINGERPRINT_NOFINGER:
      return 0;
    case FINGERPRINT_PACKETRECIEVEERR:
      return -2;
    case FINGERPRINT_IMAGEFAIL:
      return -2;
    default:
      return -2;
  }
  p = finger.image2Tz();
  switch (p) {
```

```
  case FINGERPRINT_OK:
   break;
  case FINGERPRINT_IMAGEMESS:
    return -1;
  case FINGERPRINT_PACKETRECIEVEERR:
    return -2;
  case FINGERPRINT_FEATUREFAIL:
    return -2;
  case FINGERPRINT_INVALIDIMAGE:
    return -2;
  default:
    return -2;
}

p = finger.fingerFastSearch();
if (p == FINGERPRINT_OK) {
} else if (p == FINGERPRINT_PACKETRECIEVEERR) {
  return -2;
} else if (p == FINGERPRINT_NOTFOUND) {
  return -1;
} else {
  return -2;
}


return finger.fingerID;
```

```
}

void ChecktoDeleteID(){

  HTTPClient http;

  postData = "DeleteID=check";

  http.begin(link);
  http.addHeader("Content-Type", "application/x-www-form-urlencoded");

  int httpCode = http.POST(postData);
  String payload = http.getString();

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

  http.end();
}

uint8_t deleteFingerprint( int id) {
```

```
uint8_t p = -1;

p = finger.deleteModel(id);



if (p == FINGERPRINT_OK) {

  display.clearDisplay();
  display.setTextSize(2);
  display.setTextColor(WHITE);
  display.setCursor(0,0);
  display.print(F("Deleted!\n"));
  display.display();
} else if (p == FINGERPRINT_PACKETRECIEVEERR) {

  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(0,0);
  display.print(F("Communication error!\n"));
  display.display();
  return p;
} else if (p == FINGERPRINT_BADLOCATION) {

  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(0,0);
  display.print(F("Could not delete in that location!\n"));
  display.display();
  return p;
} else if (p == FINGERPRINT_FLASHERR) {

  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(0,0);
  display.print(F("Error writing to flash!\n"));
  display.display();
  return p;
} else {

  display.clearDisplay();
  display.setTextSize(2);
  display.setTextColor(WHITE);
  display.setCursor(0,0);
  display.print(F("Unknown error:\n"));
  display.display();
  return p;
}
```

```
}

void ChecktoAddID(){

  HTTPClient http;

  postData = "Get_Fingerid=get_id";

  http.begin(link);
  http.addHeader("Content-Type", "application/x-www-form-urlencoded");

  int httpCode = http.POST(postData);
  String payload = http.getString();

  if (payload.substring(0, 6) == "add-id") {
    String add_id = payload.substring(6);
    Serial.println(add_id);
    id = add_id.toInt();
    getFingerprintEnroll();
  }
  http.end();
}
```

```
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:

   display.clearDisplay();
   display.drawBitmap(    34,    0,    FinPr_valid_bits,    FinPr_valid_width,
      FinPr_valid_height, WHITE);
   display.display();
   break;
  case FINGERPRINT_NOFINGER:

   display.setTextSize(1);
   display.setTextColor(WHITE);
   display.setCursor(0,0);
   display.print(F("scanning"));
   display.display();
```

```
    break;
  case FINGERPRINT_PACKETRECEIVER:
    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;
  }
}


  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_PACKETRECEIVER:
  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);
display.setTextColor(WHITE);
display.setCursor(0,0);
display.print(F("Remove"));
display.setCursor(0,20);
display.print(F("finger"));
display.display();

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:

    display.clearDisplay();
    display.drawBitmap( 34, 0, FinPr_valid_bits, FinPr_valid_width, FinPr_valid_height, WHITE);
    display.display();
    break;
  case FINGERPRINT_NOFINGER:
    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.setCursor(0,0);
    display.print(F("scanning"));
    display.display();
    break;
  case FINGERPRINT_PACKETRECEIVER:
    Serial.println("Communication error");
    break;
  case FINGERPRINT_IMAGEFAIL:
    Serial.println("Imaging error");
    break;
  default:
    Serial.println("Unknown error");
    break;
  }
}

  p = finger.image2Tz(2);
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:
    Serial.println("Image too messy");
    return p;
  case FINGERPRINT_PACKETRECEIVER:
    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;
}

  Serial.print("Creating model for #");  Serial.println(id);

p = finger.createModel();
if (p == FINGERPRINT_OK) {
    display.clearDisplay();
    display.drawBitmap( 34, 0, FinPr_valid_bits, FinPr_valid_width, FinPr_valid_height,
          WHITE);
    display.display();
} else if (p == FINGERPRINT_PACKETRECEIVER) {
    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) {
  display.clearDisplay();
  display.drawBitmap( 34, 0, FinPr_valid_bits, FinPr_valid_width, FinPr_valid_height,
        WHITE);
  display.display();
  confirmAdding();
} else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    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;
  }
}
void confirmAdding(){

  HTTPClient http;

  postData = "confirm_id=" + String(id);

  http.begin(link);
  http.addHeader("Content-Type", "application/x-www-form-urlencoded");

  int httpCode = http.POST(postData);
  String payload = http.getString();
  display.clearDisplay();
  display.setTextSize(1.5);
  display.setTextColor(WHITE);
  display.setCursor(0,0);
  display.print(payload);
  display.display();
  delay(1000);
  Serial.println(payload);

  http.end();  //Close connection
}

void connectToWiFi(){
    WiFi.mode(WIFI_OFF);
    delay(1000);
    WiFi.mode(WIFI_STA);
    Serial.print("Connecting to ");
    Serial.println(ssid);
    WiFi.begin(ssid, password);



    display.clearDisplay();
    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.setCursor(0, 0);
    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);
    display.setTextColor(WHITE);
    display.setCursor(8, 0);
    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());


}
```