MINOR PROJECT

REPORT ON

FINGERPRINT BIOMETRIC

ATTENDANCE SYSTEM USING ARDUINO

UNDER THE SUPERVISION OF:

DR. VANDANA NATH

ASSISTANT PROFESSOR

UNIVERSITY SCHOOL OF INFORMATION

COMMUNICATION AND TECHNOLOGY

GURU GOVIND SINGH INDRAPRASTHA UNIVERSITY

SUBMITTED BY:

SALONI GUPTA

B.TECH-ECE

ENROLLMENT NO: 70216412816

CERTIFICATE

This is to certify that the project report entitled “Fingerprint Biometric Scanner using Arduino” done by Miss Saloni Gupta, enrollment no. 70216412816 is an authentic and original work carried out by her University School of Information and Communication Technology under my guidance. The matter embodied in this project has not been submitted earlier for the award of any degree or diploma to the best of my knowledge and belief.

Signature of the guide
Dr. Vandana Nath
(Assistant Professor)

ACKNOWLEDGMENT

I would like to express my sincere gratitude towards **Prof. Arvinder Kaur**, Dean, University School of Information & Technology, GGSIPU and **Dr. Ravindra Kumar Purwar**, coordinator, University School of Information & Technology, GGSIPU who gave me the opportunity to work in this project.
I feel to acknowledge my indebtedness and a deep sense of gratitude to my guide **Dr. Vandana Nath (Assistant Professor)** whose valuable guidance and kind supervision given to me throughout the course which shaped the present work as its show. I would like to thank him for his assistance and insightful comments and for willingly sharing his expertise with me.

I would like to thank the Department of Computer Science -USICT, GGSIPU who has played an important role in strengthening my career by providing full cooperation and encouragement throughout the tenure of the project.

Also, I am grateful to all my family members and their constant support.

TABLE OF CONTENTS

1. Introduction

    1.1 Problem statement 6
    1.2 Introduction of Arduino 7
    1.3 Advantage of Arduino 8
    1.4 Disadvantage of Arduino 9
    1.5 Software and Hardware in Arduino 10

2. Implementation using ThingsBoard 11

3. Working of Fingerprint Biometric Scanner attendance system 14

4. System Requirement 15
    5.1 Hardware Requirement 15
    5.2 Software requirement 15

5. Simulation done 16

7. Result Obtained 18

8. Conclusion and Future Work 21

9. Bibliography 22

ABSTRACT

A fingerprint reader is one of the most widely accepted biometric modalities worldwide. This technology can be deployed in many areas, including national projects, banking, educational institutions, and corporate organizations, just to name a few. Biometric fingerprint readers can be used in any of the major deployment projects, banking, educational institutes, and corporate organizations.

Biometric fingerprint readers can also be used in any PC in addition to large deployment projects. Most people use this software to protect their personal devices because it is a simple, easy and inexpensive program. It eliminates the need to memorize special character alphanumeric passwords and provides instant access to your device by simply touching it.

PROBLEM STATEMENT

In this project our aim is to leverage this IoT into the boring attendance system to make it smart and more effective. Most conventional attendance systems available today store the information over a micro SD card and have to be connected to software via a computer to access the information. Here, we will build a biometric attendance system using Arduino that scans for finger print and on successful identification of the person it will log the information to a cloud platform like Things Board by using the ESP8266 Wi-Fi module. This information can then be displayed in the dashboard of Things Board making it available for the required authorities to view and analysis information over the internet without having any direct physical access to the hardware.

INTRODUCTION OF ARDUINO

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide

LCD DISPLAY

BIOMETRIC FINGER

ARUDINO UNO BOARD

Components used in project

ADVANTAGES OF ARDUINO

Easy to use: Arduino's greatest advantage is its easy-to-use architecture. As Arduino comes in a complete kit shape including a 5V regulator, a burner, an oscillator, a microcontroller, serial communication port, LED and link headers. You don't have to think about programmer or any other interface connections. Just connect it to your USB port.

Examples of codes:Arduino's library of examples inside Arduino's code is another major advantage. Use an example of voltage measurement to explain this advantage. For example, if you want to measure voltage using the ATmega8 microcontroller and you want to display the output on the computer screen, you need to go through the entire process. The process will begin with learning the microcontent of the ADCs Micro-controller for measurement, learned serial display communication and ended with USB – Serial converters.

Effortless functions: You will notice some functions during Arduino's coding that make life so easy. Arduino's automatic unit conversion capability is another advantage. You can say that you don't have to worry about the conversions of units during debugging. Simply use all your strength on the main parts of your projects. There's no need to worry about side issues.

Large community: On the internet, there are many forums where people talk about the Arduino. Through Arduino, engineers, hobbyists and professionals do their projects. About everything, you can easily find help. In addition, the Arduino website itself describes Arduino's existence and every feature.

DISADVANTAGE OF ARDUINO

Structure*:*Yes, the structure of Arduino is its disadvantage as well. During building a project you have to make its size as small as possible. But with the big structures ofArduino we have to stick with big sized PCB's. If you are working on a small microcontroller like ATmega8 you can easily make your PCB as small as possible.
Cost*:*The most important factor which you cannot deny is cost. This is the problem which every hobbyist, Engineer or Professional has to face. Now, we must consider that the Arduino is cost effective or not.

Easy to use*:*In my opinion, if you started your journey of microcontrollers with Arduino then it will be very difficult for you to make the complex intelligent circuitries in future. The easy to use hardware/software of Arduino unable a person to learn the basics of many things likes Serial communication, ADC, I2C etc.

SOFTWARE REQUIRED

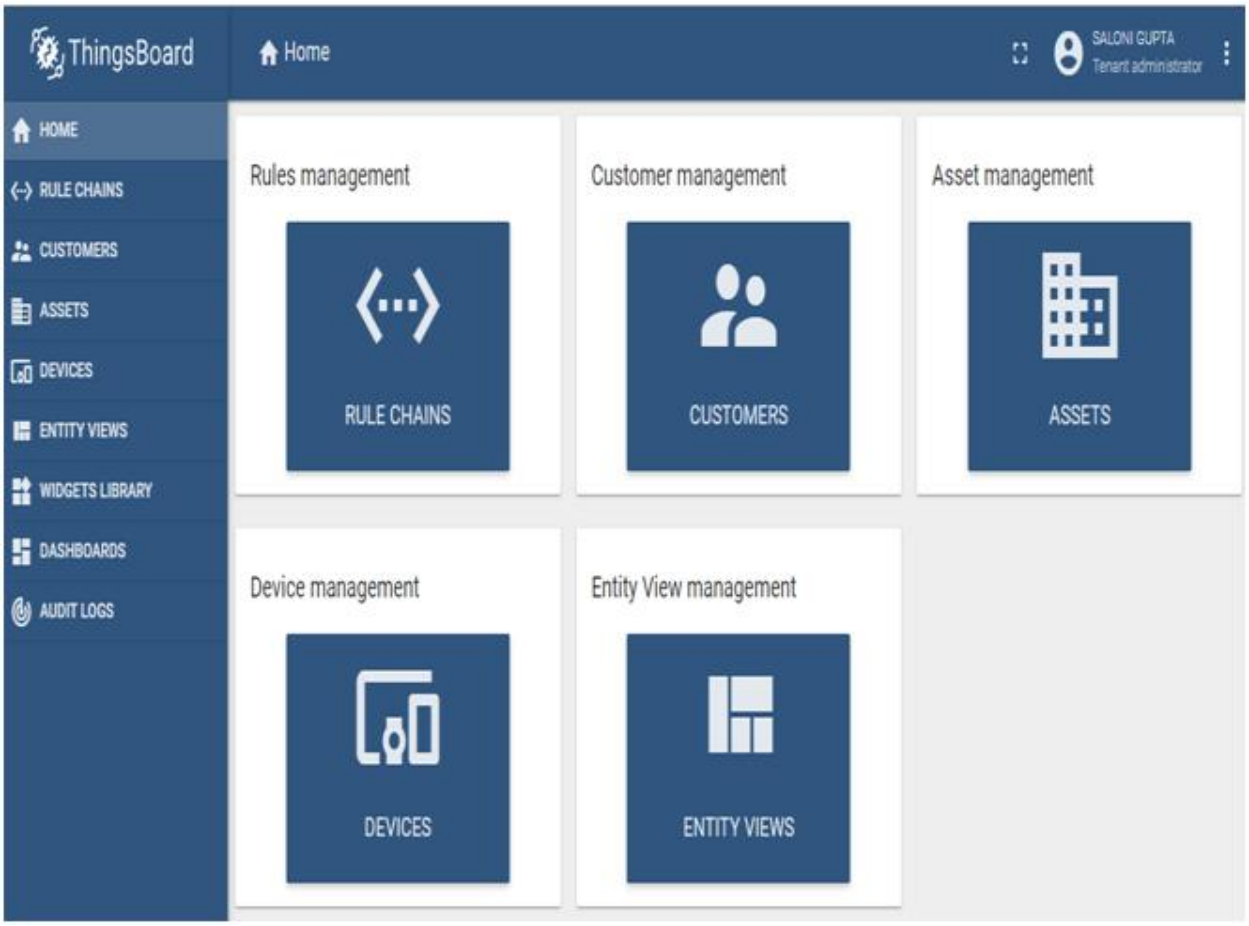
The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It originated from the IDE for the language Processing and Wiring. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus.

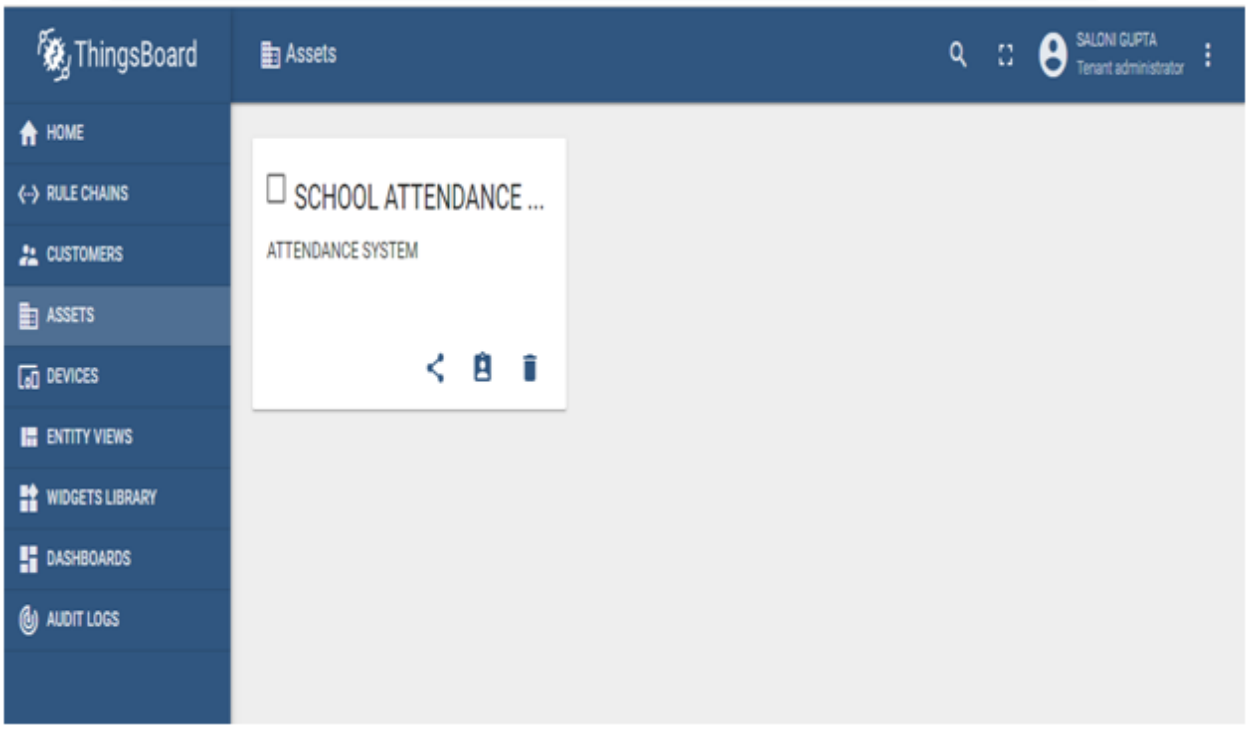
HARDWARE REQUIRED

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It originated from the IDE for the language Processing and Wiring. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one-click mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus.

IMPLEMENTATION USING THINGSBOARD

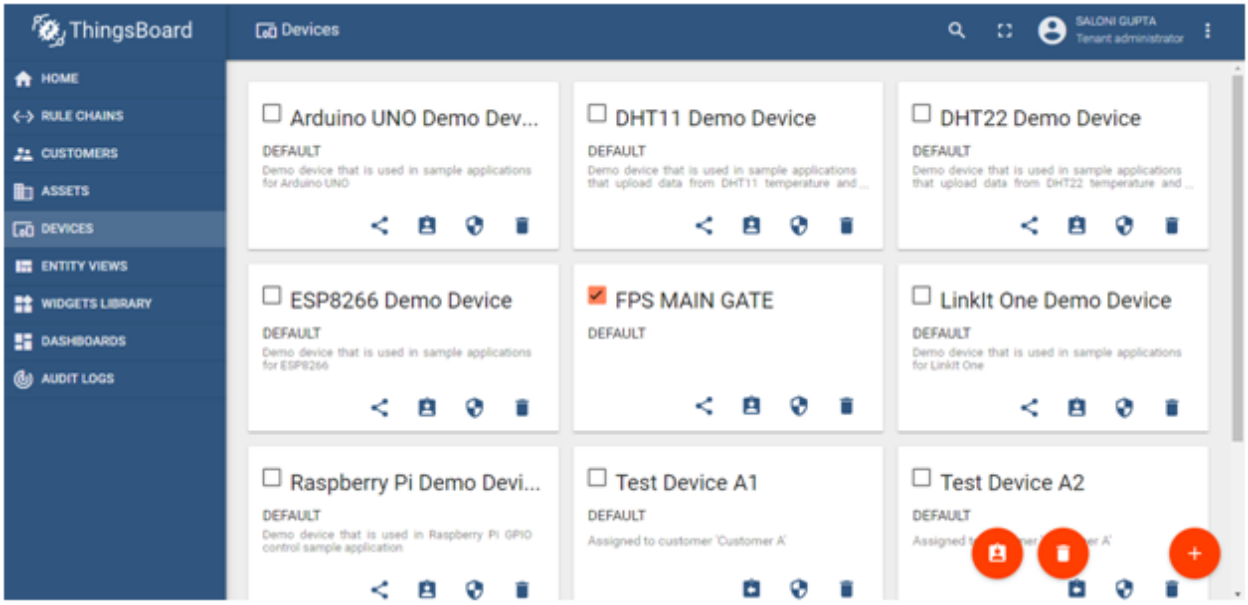
On ThingsBoard we have two important terms, Assets and Devices. You can think of assets as buildings, warehouses, industry, farmland etc and devices as the sensor or devices present in that particular asset. So every asset will have one or more devices in it based on the project, here we will have one asset and one device in our asset.

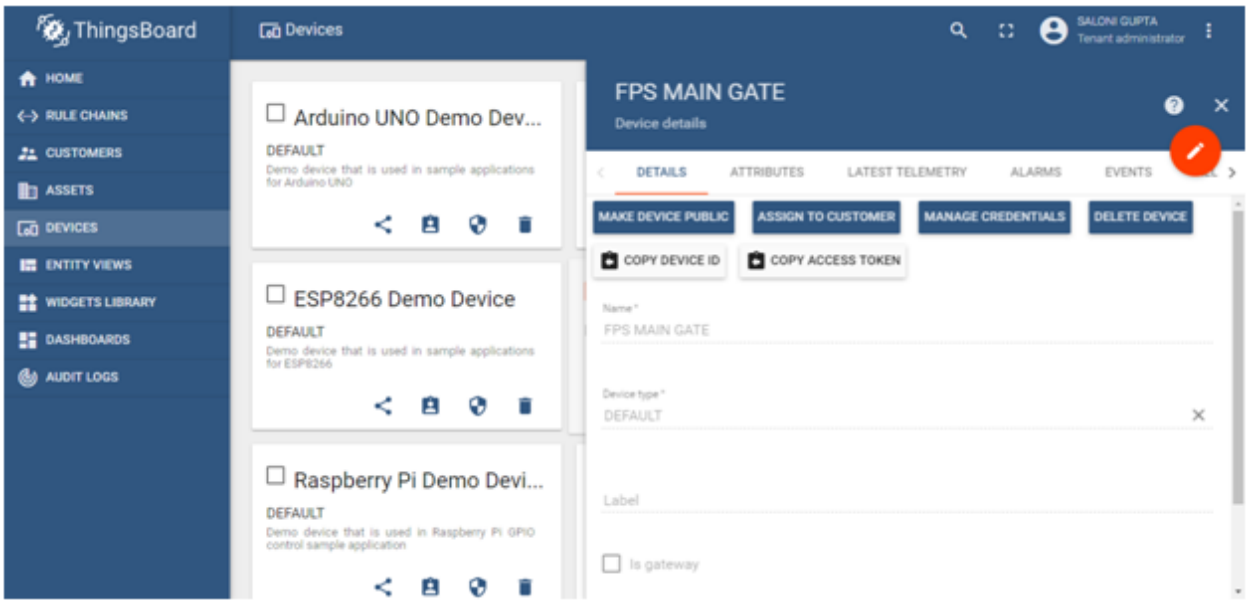




Adding a Device to the Asset

Now that we have created an asset we should add a device to it. To do so click on the device tab on the left panel and then click on the add icon on the bottom right corner of the screen. You will get a similar pop up in which you to name the device I have name mine as FPS main gate and device type as default




ThingsBoard
Devices
SALONI GUPTA
Tenant administrator
HOME
RULE CHAINS
CUSTOMERS
ASSETS
DEVICES
ENTITY VIEWS
WIDGETS LIBRARY
DASHBOARDS
AUDIT LOGS
Arduino UNO Demo Dev...
DEFAULT
Demo device that is used in sample applications for Arduino UNO
ESP8266 Demo Device
DEFAULT
Demo device that is used in sample applications for ESP8266
Raspberry Pi Demo Devi...
DEFAULT
Demo device that is used in Raspberry Pi GPIO control sample application
FPS MAIN GATE
Device details
DETAILS
ATTRIBUTES
LATEST TELEMETRY
ALARMS
EVENTS
MAKE DEVICE PUBLIC
ASSIGN TO CUSTOMER
MANAGE CREDENTIALS
DELETE DEVICE
COPY DEVICE ID
COPY ACCESS TOKEN
Name *
FPS MAIN GATE
Device type *
DEFAULT
Label
Is gateway

Working of Fingerprint Biometric Scanner attendance system

We will be writing two Arduino scripts for this program. One for the ESP8266-01 Module and the other is for Arduino UNO. The reason is that Arduino was not able to handle both GT511C3 sensor and ESP8266 through AT commands via software serial. Hence we will write two codes, one for Arduino in which it will communicate with the FPS and send the obtained values via software serial to ESP8266. The other code will be written for ESP8266 which will enable the module to be connected to the Thingsboard server and then will receive the values from Arduino through serial communication to update them on Thingsboard Dashboard.

First of all, the user needs to enroll fingerprints of the user with the help of push buttons. To do this, user need to press ENROLL key and then LCD asks for entering ID for the fingerprint to save it in memory by ID name. So now the user needs to enter ID by using UP/DOWN keys. After selecting ID, user needs to press the OK key (DEL key). Now LCD will ask to place a finger over the fingerprint module. Now the user needs to place his finger over finger print module and then the module takes finger image. Now the LCD will say to remove finger from fingerprint module, and again ask to place finger again. Now user needs to put his finger again and module takes an image and convert it into templates and stores it by selected ID into the fingerprint module's memory. Now the user will be registered and he/she can feed attendance by putting their finger over fingerprint module.By the same method, all the users will be registered into the system.

The Program on Arduino is pretty simple. When a fingerprint is enrolled a template is created and an ID is allocated to the person. So all we have to do it look for the ID and relate it to a name and send this name to ESP826 through serial communication. This name will then be sent to the ThingsBoard device.

SYSTEM REQUIREMENT

Hardware Required

1.Arduino Uno Board
2.16.2 LCD DISPLAY
3.Arduino Wifi Shield
4.ESP8266-01

5.GT511C3 Fingerprint Sensor

Software Required

1.Arduino ide

2.MC Programming Language: C

SIMULATION DONE

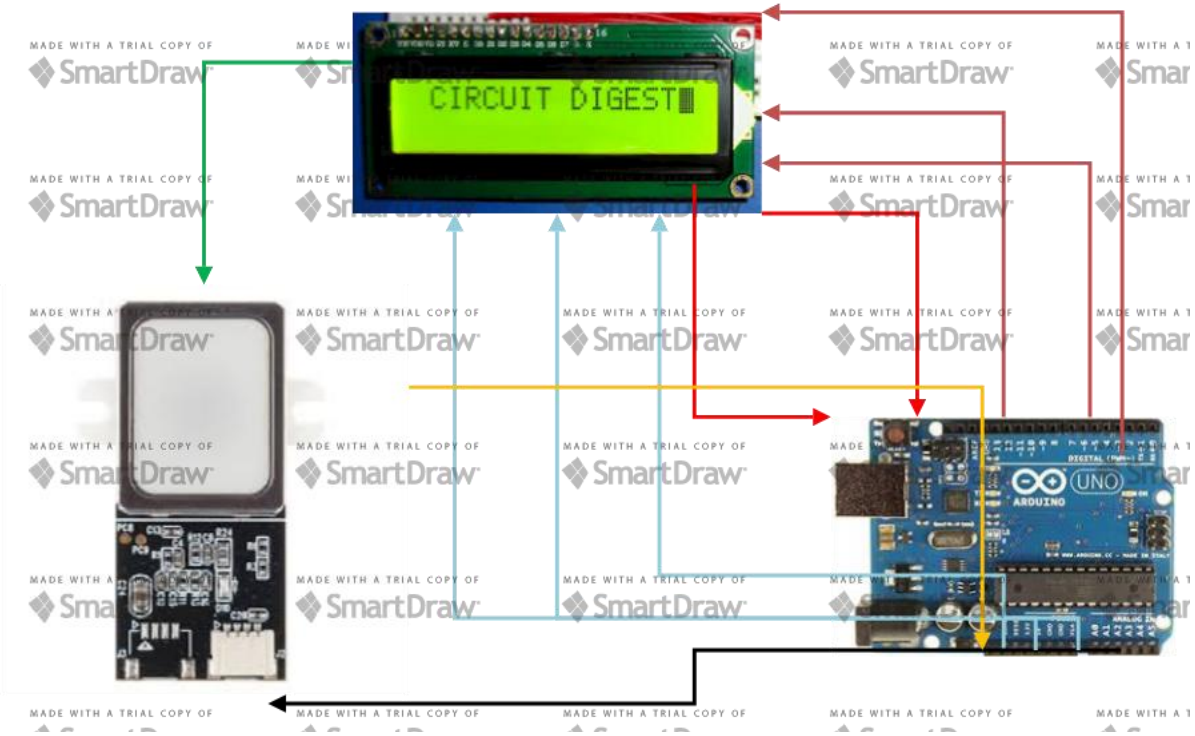
```
/*
 * Arduino with GT511C2 FingerPrint Sensor (FPS)

 */

char *Name_List[]= {"ADMIN", "Sujata", "Sanghavi", "Samikshya", "SALONI", "SIMRAN", "Aayush", "Guest 1", "Guest 2", "Guest 3", "Guest 4"};

#include "FPS_GT511C3.h"
#include "SoftwareSerial.h" //Software serial library
#include <LiquidCrystal.h> //Library for LCD
#include "OnePinCapSense.h" //Library to sensor capacitive touch
SoftwareSerial ESP(12, 13); // RX, TX

FPS_GT511C3 fps(9, 8); //FPS connected to D9 and D8

const int rs = 7, en = 6, d4 = 5, d5 = 4, d6 = 3, d7 = 2; //Mention the pin number for LCD connection
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);//Initialize LCD method

int capSensePin10  = 10; //Pin to which casing of sensor is connected
int id = 200;
OnePinCapSense opcs = OnePinCapSense();

void setup()
{
  Serial.begin(9600);
  ESP.begin (9600);

  lcd.begin(16, 2); //Initialise 16*2 LCD
  lcd.print("GT511C3 FPS"); //Intro Message line 1
  lcd.setCursor(0, 1);
  lcd.print("with Arduino"); //Intro Message line 2
  delay(2000);
  lcd.clear();

  fps.Open();         //send serial command to initialize fps
  fps.SetLED(false);   //turn on LED so fps can see fingerprint
}

void loop()
{
```

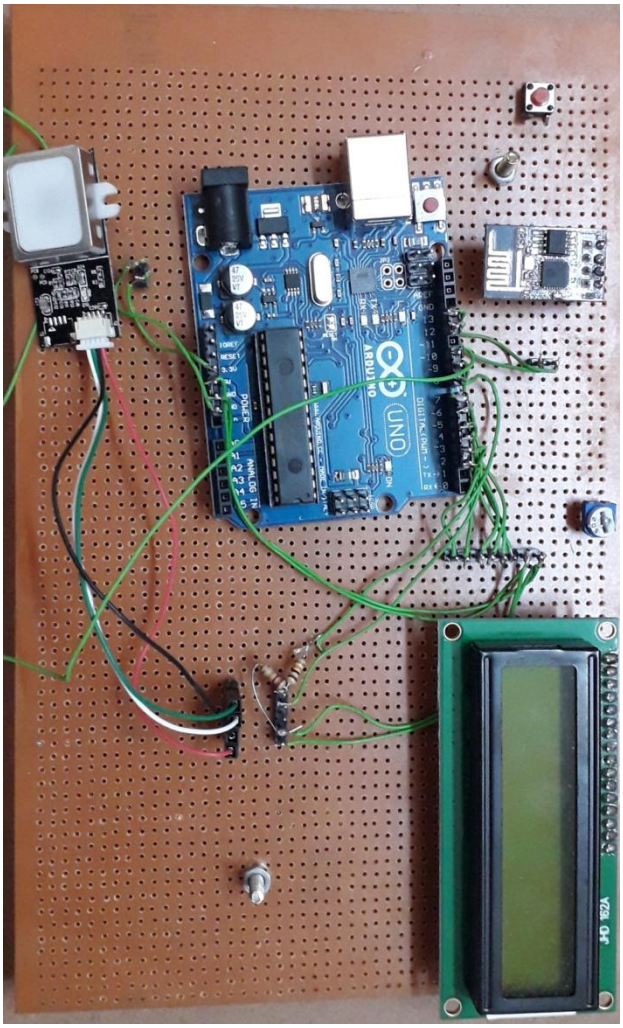
```
 int capSense10 = opcs.readCapacitivePin(capSensePin10) ;
 if( capSense10 < 100)
 { fps.SetLED(true); delay(500);}
 Serial.println(capSense10);

// Identify fingerprint test
  if (fps.IsPressFinger())
  {
    fps.CaptureFinger(false);
    id = fps.Identify1_N();
      lcd.clear();
      if (id==200)
      {
      lcd.print("Unkown"); lcd.setCursor(0,1); lcd.print("Try Again!!");//If not recognised
      }
      else if (id==0)
      {
       lcd.print("Welcome"); lcd.setCursor(0,1); lcd.print("***ADMIN***");//If not recognised
       delay(2000);
       Enroll();
      }
      else
      {
        lcd.print("Thank You"); lcd.setCursor(0,1); lcd.print(Name_List[id]);
ESP.println(Name_List[id]);
        Serial.println("Sent Value to ESP");
      }

      delay(1000);
      fps.SetLED(false);
  }
  else
  {
    fps.SetLED(false);
    lcd.clear();
    lcd.print("Engineer Student"); //Display intro text
    lcd.setCursor(0,1);
    lcd.print("Press Finger");
  }
}

void Enroll() //Enrol function from library exmaple program
{
  int enrollid = 0;
  bool usedid = true;
  while (usedid == true)
  {
    usedid = fps.CheckEnrolled(enrollid);
    if (usedid==true) enrollid++;
  }
  fps.EnrollStart(enrollid);
```

```
  // enroll
  lcd.clear();
  lcd.print("Enroll #");
  lcd.print(enrollid);
  while(fps.IsPressFinger() == false) delay(100);
  bool bret = fps.CaptureFinger(true);
  int iret = 0;
  if (bret != false)
  {
    lcd.clear();
    lcd.print("Remove finger");
    fps.Enroll1();
    while(fps.IsPressFinger() == true) delay(100);
    lcd.clear(); lcd.print("Press again");
    while(fps.IsPressFinger() == false) delay(100);
    bret = fps.CaptureFinger(true);
    if (bret != false)
    {
      lcd.clear(); lcd.print("Remove finger");
      fps.Enroll2();
      while(fps.IsPressFinger() == true) delay(100);
      lcd.clear(); lcd.print("Press yet again");
      while(fps.IsPressFinger() == false) delay(100);
      bret = fps.CaptureFinger(true);
      if (bret != false)
      {
        lcd.clear(); lcd.print("Remove finger");
        iret = fps.Enroll3();
        if (iret == 0)
        {
          lcd.clear(); lcd.print("Enrolling Success");
        }
        else
        {
          lcd.clear();
          lcd.print("Enroll Failed:");
          lcd.print(iret);
        }
      }
      else lcd.print("Failed 1");
    }
    else lcd.print("Failed 2");
  }
  else lcd.print("Failed 3");
}
```

CIRCUIT DIAGRAM OF BIOMETRIC ATTENDANCE SYSTEM

The circuit of this fingerprint based attendance system project, as shown in the above diagram is quite simple. It has Arduino for controlling all the process of the project, push button for enrolling, deleting, selecting IDs and for attendance, LEDs for indication and LCD to instruct user and showing the resultant messages.





Screenshot of setup

CONCLUSION AND FUTURE WORK

You can also display the data in real time and the employee's name will appear on this window when you press the finger on the FPS. That is, we created a smart IoT-based attendance system that updates all of our dashboard data that can be viewed, analyzed and reported online without the need for physical hardware contact.
The entire project can be powered directly through the Arduino UNO's Vin pin by a 12V adapter. The 16x2 LCD is attached to display the name of the designated person.

Another important thing to note is that the FPS ' metal casing is attached to Arduino's pin 10. If not in use, this is to shut off the FPS. As we know, the FPS needs a blue LED to turn on To take and recognize fingerprint optical images. But if this light is kept on the sensor at all times, it tends to get hot, which could reduce its life. Fortunately found a method mentioned in its datasheet to overcome the problem of using the metal casing of the sensor as a touch sensor.

BIBLIOGRAPHY

1.https://circuitdigest.com/microcontroller-projects/

2.fingerprint-attendance-system-using-arduino-uno

3.www.tutorialspoint.com