K. J. Somaiya College of Engineering, Vidyavihar, Mumbai-77
(Autonomous College Affiliated to University of Mumbai)

BIOMETRIC ATTENDANCE SYSTEM

by

Hersh Vitekar - 1913063 (IoT-2)

Dhanshree Chavan - 1913072 (IoT-2)

Rahul Doshi - 1913078 (IoT-3)

Khushbu Parmar - 1913100 (IoT-3)

Internal Assessment-II

Departmental Elective- Internet of Things (2UTE616)

TY B.Tech (Semester-VI)

January to May 2022 Term

K.J. Somaiya College of Engineering, Vidyavihar, Mumbai-77.

(Autonomous College affiliated to University of Mumbai)

Certificate

This is to certify that the report on "BIOMETRIC ATTENDANCE SYSTEM"is bona fide record of the work done by

1. Hersh Vitekar - 1913063

2. Dhanshree Chavan - 1913072

3. Rahul Doshi - 1913078

4. Khushbu Parmar - 1913100

in the academic year **2021-22**, **Department of Electronics and Telecommunication Engineering** for the course "**Internet of Things (2UTE616)"**

_____ Signature of Faculty

Date:

Place: Mumbai-77

Title of Application: BIOMETRIC ATTENDANCE SYSTEM

Brief Description of the application:

Biometrics is the most suitable means of identifying and authenticating individuals in a reliable and fast way through unique biological characteristics.

A Biometric is a device that captures employees' daily attendance via fingerprints. The biometric device helps the organization track the attendance of its employees systematically. It makes use of biometrics of each employee to keep a record of their in and out time during working hours.

Biometric factors are defined by seven characteristics: universality, uniqueness, permanence, collectability, performance, acceptability, and circumvention. Biometrics allows a person to be identified and authenticated based on recognizable and verifiable data, unique and specific. Biometrics can be defined as the most practical means of identifying and authenticating individuals in a reliable and fast way through unique biological characteristics.

● Biometric authentication: Biometric authentication compares data for the person's characteristics to that person's biometric "template" to determine resemblance. ● The reference model is first stored.

● The data stored is then compared to the person's biometric data to be authenticated.

In this mode, the question is: "Are you indeed, Mr or Mrs. X?"

● Biometric identification: Biometric identification consists of determining the identity of a person.

● The aim is to capture an item of biometric data from this person. It can be a photo of their face, a record of their voice, or an image of their fingerprint. ● This data is then compared to the biometric data of several other persons kept in a database.

In this mode, the question is simple: "Who are you?" What is Biometric attendance?

Biometrics attendance measures and verifies the biological characteristics of a human. It helps to keep track of the schedule of workers, volunteers, or students. It can accurately record the entry and exit times of the individual. So the employer/authority will have clarity and there will be no dispute between who is in and out for the duty.

There are mainly 2 types of biometric techniques:

1. Behavioral biometrics - It works on the verification method. It applies to offices, business centers, and various educational institutions.
2. Physical biometrics - It works for both identification and verification methods. The

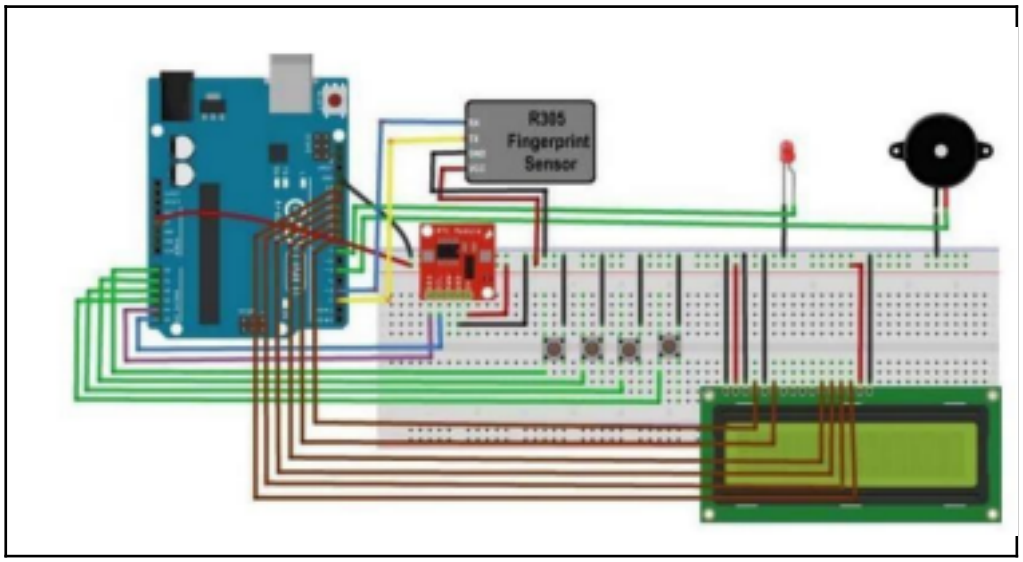
banks and other investigative departments use it for secure transactions and criminal investigations.

Benefits of Biometric Attendance System

- The biometric Attendance system increases the accuracy of payroll during the pay period.
- Biometrics provides accurate time and attendance records to the payroll system as it saves employees working time and decreases overhead.

- The biometric attendance management system will alert employees to excessive overtime situations which helps to balance the workload.
- The biometric system improves accountability and responsibility for employees because it can accurately identify late attendance, and frequent, extended, or unscheduled breaks.

Circuit diagram/block diagram along with detailed:



Working of the circuit:

Working on this fingerprint attendance system project is fairly simple. First of all, the user needs to enroll fingerprints of the user with the help of pushbuttons. To do this, the user needs to press the ENROLL key and then LCD asks for entering the ID for the fingerprint to save it in memory by ID name. So now the user needs to enter an ID by using the UP/DOWN keys. After selecting the ID, the 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 the fingerprint module and then the module takes the finger image. Now the LCD will say to remove the finger from the fingerprint module and ask to place the finger again. Now the user needs to put his finger again and the module takes an

image and converts it into templates and stores it by selecting the ID into the fingerprint module's memory. Now the user will be registered and he/she can feed attendance by putting their finger over the fingerprint module. By the same method, all the users will be registered into the system. Now if the user wants to remove or delete any of the stored ID or fingerprint, then he/she needs to press the DEL key. Once the delete key is pressed LCD will ask to select the ID that needs to be deleted. Now the user needs to selectID and press the OK key (same DEL key). Now the LCD will let you know that the fingerprint has been deleted successfully.

Details of Components used:

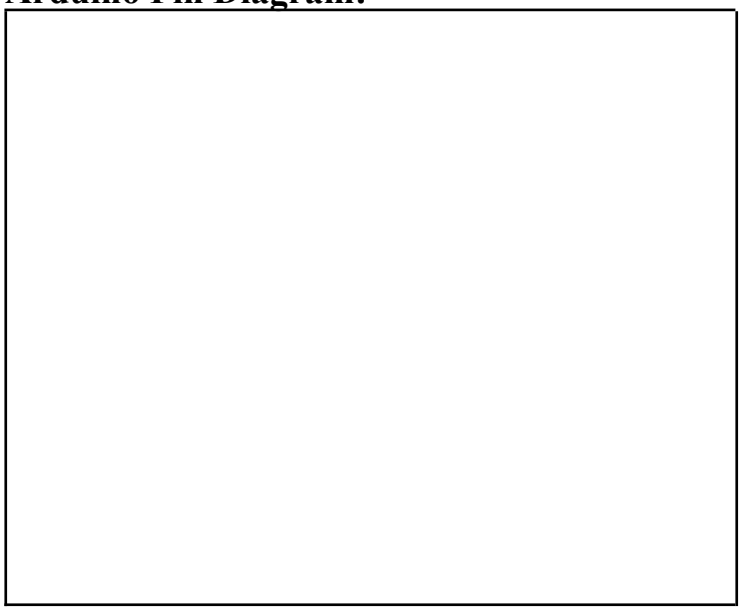
Serial No.	List of Components	Quantity
1	Arduino Uno	1
2	Fingerprint Module	1
3	RTC Module	1
4	16x2 LCD with 12C Module	1
5	Pushbuttons	4
6	Buzzer	1
7	LEDs	1
8	Breadboard and Jumper Wires	1

1. ARDUINO UNO:

The main purpose of the microcontroller is to enroll and search the fingerprint. In enrolling, this controller reads the template from the fingerprint sensor and enrolls the ID number.

This displays the ID number on the serial monitor. And then, this controller checks the fingerprint with the stored template in the searching process. If the fingerprint is correct, the display values are shown in excel. Otherwise, the controller doesn't give any output.

Arduino Pin Diagram:



2. R305 SCANNER:

In this Fingerprint Sensor Based Biometric Attendance System using Arduino, we used a Fingerprint Sensor module to authenticate a true person or employee by taking their finger input in the system. This sensor reads the fingerprint pattern. The scan image is converted as a template and saved in memory. This is an interface which can be directly connected to the Arduino UART. TheR305/R307 fingerprint scanner has a TTL UART.

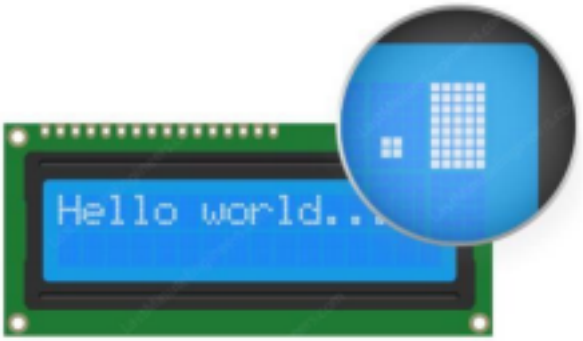
R305/R307	Arduino Mega	Arduino Uno
GND	GND	GND
Vcc	5V	5V
Rx	18	3
Tx	19	2

3. DS3231 RTC MODULE

RTC Module DS3231 is used for registering scanning/entering/existing time of the user. RTC modules are simply TIME and DATE remembering systems which have a battery setup which in the absence of external power keeps the module running. This keeps the TIME and DATE up to date. So we can have accurate TIME and DATE from the RTC module whenever we want.

4. LCD WITH 12C MODULE:

As the name implies, the LCD module communicates with Arduino through I2C communication. A typical I2C LCD display consists of a HD44780 based character LCD display and an I2C LCD adapter. True to its name, these LCDs are ideal for displaying text/characters only. A 16×2 character LCD, for example, has an LED backlight and can display 32 ASCII characters in two rows with 16 characters on each row. The LCD displays the time record and every function happening via the push button. If you look closely, you can actually see the little rectangles for each character on the display and the pixels that make up a character. Each of these rectangles is a grid of 5×8 pixels.



5. PUSHBUTTONS

The functions of each button are: 1. Register/Back Button – Used for enrolling new fingerprints as well as reversing the back process or going back 2. Delete/OK Button – This Button is used for deleting the earlier stored fingerprint system as well as granting access as an OK selection. 3. Forward Button – Used for moving forward while

selecting the memory location for storing or
deleting fingerprints. 4. Reverse Button – Used for
moving backward while selecting memory location for storing or deleting fingerprints.

6. LED

The LED is used for power indication.

Specifications:

1. Arduino Uno:

- Clock Speed:16MHz
- Operating Voltage:5V
- Maximum supply Voltage (not recommended):20V
- Supply Voltage (recommended):7-12V
- Analog Input Pins:6
- Digital Input/Output Pins:14
- DC Current per Input/Output Pin:40mA
- DC Current in 3.3V Pin:50mA
- SRAM:2KB

- EEPROM:1 KB
- Flash Memory:32KB of which 0.5KB used by boot loader

2. LCD 12C:

- Screen type: Dual Colour LCD
- Screen Resolution: 128*64 pixels
- Screen Active Area (L*W): 47.1*26.5mm
- Individual Pixel Size: 0.33*0.33 mm
- Communication Mode: I2C(100Kbit/s and 400Kbit/s)
- Controller: STM8S005KBT6
- Operating Frequency:16 MHz
- Weight: 20g

3. Fingerprint Sensor (R305) -TTL UART:

- Power DC : 3.6V-6.0V
- Interface : UART (TTL logical level)/ USB 1.1
- Working current : 100mA
- Peak Current : 150mA
- Matching Mode: 1:1 and 1:N
- Baud rate (9600*N)bps, N=1-12 (default N=6 57600bps)
- Character file size: 256 bytes
- Image acquiring time : <0.5s
- Template size : 512 bytes

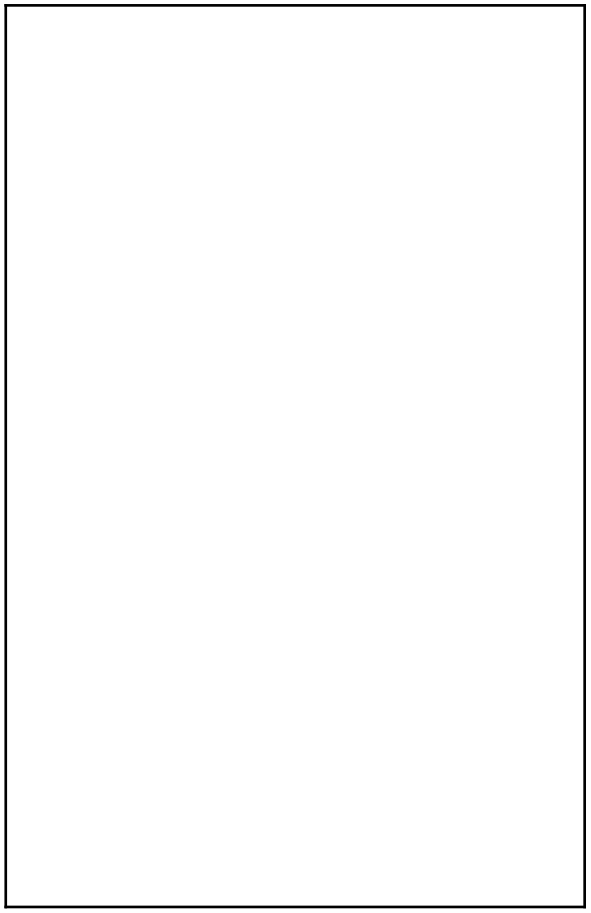
• Storage capacity: 256
• Security level : 5 (1, 2, 3, 4, 5(highest))
• FAR : <0.001%
• FRR: <0.1%
• Average searching time: < 0.8s (1:880)
• Window dimension : 18mm*22mm

4. DS3231 RTC MODULE

• Operating voltage of DS3231 MODULE: 2.3V – 5.5V
• Can operate on LOW voltages
• Consumes 500nA on battery backup
• Maximum voltage at SDA , SCL : VCC + 0.3V
• Operating temperature: -45ºC to +80ºC

Software Requirements: All of the fingerprint sensor's functions are controlled by the Adafruit Fingerprint sensor library. The library contains the methods and functions to drive the fingerprint scanner. Also, the DS3231 Library is used for the RTC Module

Flowchart:



Code:

```
#include "Adafruit_Fingerprint.h" //fingerprint library header file
#include<EEPROM.h> //command for storing data
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,4);
#include <SoftwareSerial.h>
SoftwareSerial fingerPrint(2, 3); //for tx/rx communication between arduino & r305
fingerprint sensor #include <Wire.h>
#include "RTClib.h" //library file for DS3231 RTC Module
RTC_DS3231 rtc;
uint8_t id;
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&fingerPrint);
#define register_back 14
#define delete_ok 15
#define forward 16
#define reverse 17
#define match 5
#define indFinger 7
#define buzzer 5
#define records 10 // 10 for 10 user
```

```
int user1,user2,user3,user4,user5,user6,user7,user8,user9,user10;
DateTime now; void
 setup()
 {
 delay(1000);
Serial.begin(9600);
lcd.begin(16,2); lcd.init();
 lcd.backlight();
pinMode(register_back, INPUT_PULLUP);
pinMode(forward, INPUT_PULLUP);
pinMode(reverse, INPUT_PULLUP);
pinMode(delete_ok, INPUT_PULLUP);
pinMode(match, INPUT_PULLUP);
pinMode(buzzer, OUTPUT);
pinMode(indFinger, OUTPUT);
digitalWrite(buzzer, LOW);
if(digitalRead(register_back) == 0)
 {
digitalWrite(buzzer, HIGH);
delay(500);
digitalWrite(buzzer, LOW);
lcd.clear();
lcd.print("Please wait !");
lcd.setCursor(0,1); lcd.print("Downloading
Data"); Serial.println("Please wait");
Serial.println("Downloading Data..");
Serial.println();
Serial.print("S.No. "); for(int
i=0;i<records;i++)
{ digitalWrite(buzzer,
HIGH); delay(500);
digitalWrite(buzzer, LOW);
Serial.print(" User ID");
Serial.print(i+1);
Serial.print(" ");
}
Serial.println(); int
eepIndex=0; for(int
i=0;i<30;i++) {
if(i+1<10)
Serial.print('0');
Serial.print(i+1); Serial.print("
"); eepIndex=(i*7);
download(eepIndex);
eepIndex=(i*7)+210;
```

```
download(eepIndex);
```



```
eepIndex=(i*7)+420;
download(eepIndex);
eepIndex=(i*7)+630;
download(eepIndex);
eepIndex=(i*7)+840;
download(eepIndex);
eepIndex=(i*7)+1050;
download(eepIndex);
eepIndex=(i*7)+1260;
download(eepIndex);
eepIndex=(i*7)+1470;
download(eepIndex);
eepIndex=(i*7)+1680;
download(eepIndex);
Serial.println();
}
}
if(digitalRead(delete_ok) == 0)
{ lcd.clear();
lcd.print("Please Wait");
lcd.setCursor(0,1);
lcd.print("Reseting.....");
for(int i=1000;i<1005;i++)
EEPROM.write(i,0);
for(int i=0;i<841;i++)
EEPROM.write(i, 0xff);
lcd.clear();
lcd.print("System Reset");
delay(1000);
} lcd.clear(); lcd.print("
Fingerprint "); lcd.setCursor(0,1);
lcd.print("Attendance System");
delay(2000); lcd.clear();
digitalWrite(buzzer, HIGH);
delay(500); digitalWrite(buzzer,
LOW); for(int
i=1000;i<1000+records;i++)
{
if(EEPROM.read(i) == 0xff)
EEPROM.write(i,0);
} finger.begin(57600);
Serial.begin(9600);
lcd.clear(); lcd.print("Finding
```

```
Module..");
lcd.setCursor(0,1);
```



```
delay(2000); if
(finger.verifyPassword())
{
Serial.println("Found fingerprint sensor!");
lcd.clear(); lcd.print("
Module Found");
delay(2000); } else
{
Serial.println("Did not find fingerprint sensor :(");
lcd.clear();
lcd.print("Module Not Found");
lcd.setCursor(0,1); lcd.print("Check
Connections");
while (1); }
if (! rtc.begin())
Serial.println("Couldn't find RTC");
 // rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
if (rtc.lostPower())
{
Serial.println("RTC is NOT running!");
// following line sets the RTC to the date & time this sketch was compiled
rtc.adjust(DateTime(2022, 4, 18, 11, 0, 0));
// This line sets the RTC with an explicit date & time, for example to set
// June 7, 2018 at 11am you would call:
// rtc.adjust(DateTime(2018, 6, 7, 11, 0, 0));
}
lcd.setCursor(0,0); lcd.print("
Press Match to ");
lcd.setCursor(0,1); lcd.print("
Start System"); delay(3000);
user1=EEPROM.read(1000);
user2=EEPROM.read(1001);
user3=EEPROM.read(1002);
user4=EEPROM.read(1003);
user5=EEPROM.read(1004);
lcd.clear();
digitalWrite(indFinger, HIGH);
} void loop() { now =
rtc.now();
lcd.setCursor(0,0);
lcd.print("Time: ");
```

```
lcd.print(now.hour(),
DEC);
lcd.print(':');
lcd.print(now.minute(), DEC);
lcd.print(':');
```



```
lcd.print(now.second(), DEC);
lcd.print(" "); lcd.setCursor(0,1);
lcd.print("Date: ");
lcd.print(now.day(), DEC);
lcd.print('/');
lcd.print(now.month(), DEC);
lcd.print('/');
lcd.print(now.year(), DEC);
lcd.print(" "); delay(500); int
result=getFingerprintIDez();
if(result>0) {
digitalWrite(indFinger, LOW);
digitalWrite(buzzer, HIGH);
delay(100);
digitalWrite(buzzer,  LOW);
lcd.clear();   lcd.print("ID:");
lcd.print(result);
lcd.setCursor(0,1);
lcd.print("Please Wait....");
delay(1000);
attendance(result); lcd.clear();
lcd.print("Attendance ");
lcd.setCursor(0,1);
lcd.print("Registered");
delay(1000);
digitalWrite(indFinger, HIGH);
return; } checkKeys();
delay(300);
}
// dmy hms - 7 bytes
void attendance(int id)
{ int
user=0,zipLoc=0;
if(id == 1) {
zipLoc=0;
user=user1++; } else
if(id == 2)
{ zipLoc=210;
user=user2++;
} else if(id ==
```

```
3) {
zipLoc=420;
user=user3++;
} else if(id ==
4)
```



```
{ zipLoc=630;
user=user4++;
} else if(id ==
5) { zipLoc=0;
user=user5++;
} else if(id ==
6)
{ zipLoc=840;
user=user5++;
} else if(id ==
7)
{
zipLoc=1050;
user=user7++;
} else if(id ==
8)
{
zipLoc=1260;
user=user8++;
} else if(id ==
9)
{ zipLoc=1470;
user=user
k9++; } else
if(id == 10)
{ zipLoc=1680;
user=user8++;
}
/*else if(id == 5) // fifth user
{
zipLoc=840;
user=user5++
; }*/ else
return;
int eepIndex=(user*7)+zipLoc; EEPROM.write(eepIndex++,
now.hour());
EEPROM.write(eepIndex++, now.minute());
EEPROM.write(eepIndex++, now.second());
EEPROM.write(eepIndex++, now.day());
```

```
EEPROM.write(eepIndex++, now.month());
EEPROM.write(eepIndex++, now.year()>>8 );
EEPROM.write(eepIndex++, now.year());
EEPROM.write(1000,user1);
EEPROM.write(1001,user2);
EEPROM.write(1002,user3);
EEPROM.write(1003,user4);
```



```
// EEPROM.write(4,user5); // fifth user
}
void checkKeys() {
if(digitalRead(register_back) == 0)
{ lcd.clear();
lcd.print("Please
Wait"); delay(1000);
while(digitalRead(register_back) == 0);
Enroll(); }
else if(digitalRead(delete_ok) == 0)
{ lcd.clear();
lcd.print("Please Wait");
delay(1000); delet(); } }
void Enroll() { int count=1;
lcd.clear(); lcd.print("Enter
Finger ID:"); while(1) {
lcd.setCursor(0,1);
lcd.print(count);
if(digitalRead(forward) == 0)
{ count++;
if(count>r
ecords)
count=1;
delay(500)
; }
 else if(digitalRead(reverse) == 0)
{ count--;
if(count<1)
count=records;
delay(500); }
else if(digitalRead(delete_ok) == 0)
{ id=count;
getFingerprintEnroll();
for(int i=0;i<records;i++)
{
if(EEPROM.read(i) != 0xff)
{
```

```
EEPROM.write(i, id);
break; } } return; }
else if(digitalRead(register_back) == 0)
{ return;
}
} } void delete() { int
count=1; lcd.clear();
lcd.print("Enter Finger ID");
while(1) {
```



```
lcd.setCursor(0,1);
lcd.print(count);
if(digitalRead(forward) == 0)
{ count++;
if(count>records)
count=1;
delay(500);
}
else if(digitalRead(reverse) == 0)
{ count--;
if(count<1)
count=records;
delay(500); }
else if(digitalRead(delete_ok) == 0)
{ id=count;
deleteFingerprint(id);
for(int i=0;i<records;i++)
{ if(EEPROM.read(i) ==
id)
{
EEPROM.write(i, 0xff);
break; } } return; }
else if(digitalRead(register_back) == 0)
{ return;
}
} }
uint8_t getFingerprintEnroll()
{ int p = -1; lcd.clear();
lcd.print("finger ID:");
lcd.print(id);
lcd.setCursor(0,1);
lcd.print("Place
Finger"); delay(2000);
while (p != FINGERPRINT_OK)
{ p =
finger.getImage();
```

```
switch (p)
{
case FINGERPRINT_OK:
Serial.println("Image taken"); lcd.clear(); lcd.print("Image taken"); break; case
FINGERPRINT_NOFINGER:
Serial.println("No Finger"); lcd.clear();
lcd.print("No Finger Found"); break; case
FINGERPRINT_PACKETRECIEVEERR:
Serial.println("Communication error");
```



```
lcd.clear(); lcd.print("Comm
Error");
break; case
FINGERPRINT_IMAGEFAIL:
Serial.println("Imaging error");
lcd.clear(); lcd.print("Imaging
Error"); break; default:
Serial.println("Unknown error");
lcd.clear(); lcd.print("Unknown
Error"); break; }
} // OK success! p =
finger.image2Tz(1);
switch (p) { case
FINGERPRINT_OK:
Serial.println("Image converted");
lcd.clear(); lcd.print("Image
converted"); break; case
FINGERPRINT_IMAGEMESS:
Serial.println("Image too messy");
lcd.clear(); lcd.print("Image too
messy");
return p; case
FINGERPRINT_PACKETRECIEVEERR:
Serial.println("Communication error");
lcd.clear(); lcd.print("Comm
Error");
return p; case
FINGERPRINT_FEATUREFAIL:
Serial.println("Could not find fingerprint features"); lcd.clear(); lcd.print("Feature Not
Found");
return p; case
FINGERPRINT_INVALIDIMAGE:
Serial.println("Could not find fingerprint features");
lcd.clear(); lcd.print("Feature
Not Found");
```

```
return p; default:
Serial.println("Unknown error");
lcd.clear(); lcd.print("Unknown
Error");
return p;
}
Serial.println("Remove finger");
lcd.clear(); lcd.print("Remove
Finger"); delay(2000); p = 0;
while (p != FINGERPRINT_NOFINGER) {
p = finger.getImage();
```



```
}
Serial.print("ID "); Serial.println(id); p
= -1;
Serial.println("Place same finger again");
lcd.clear(); lcd.print("Place
Finger");
lcd.setCursor(0,1);
lcd.print(" Again");
while (p != FINGERPRINT_OK) {
p = finger.getImage(); switch (p) {
case FINGERPRINT_OK:
Serial.println("Image taken");
break; case
FINGERPRINT_NOFINGER:
Serial.print(".");
break; case
FINGERPRINT_PACKETRECIEVEERR:
Serial.println("Communication error");
break; case
FINGERPRINT_IMAGEFAIL:
Serial.println("Imaging error");
break; default:
Serial.println("Unknown error");
return; }
}
// OK success! p =
finger.image2Tz(2);
switch (p) { case
FINGERPRINT_OK:
Serial.println("Image converted");
break; case
FINGERPRINT_IMAGEMESS:
Serial.println("Image too messy");
return p; case
```

```
FINGERPRINT_PACKETRECIEVEERR:
Serial.println("Communication error");
return p; case
FINGERPRINT_FEATUREFAIL:
Serial.println("Could not find fingerprint features");
return p; case FINGERPRINT_INVALIDIMAGE:
Serial.println("Could not find fingerprint features");
return p; default:
Serial.println("Unknown error");
return p;
}
// OK converted!
```



```
Serial.print("Creating model for #"); Serial.println(id);
p = finger.createModel(); if (p ==
FINGERPRINT_OK) {
Serial.println("Prints matched!");
} else if (p == FINGERPRINT_PACKETRECIEVEERR) {
Serial.println("Communication error");
return p;
} else if (p == FINGERPRINT_ENROLLMISMATCH) {
Serial.println("Fingerprints did not match"); return  p;
} else {
Serial.println("Unknown error");
return p;
}
Serial.print("ID "); Serial.println(id);
p = finger.storeModel(id); if (p ==
FINGERPRINT_OK) {
Serial.println("Stored!");
lcd.clear(); lcd.print("
Finger Stored!");
delay(2000);
} 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;
} }
```

```
int getFingerprintIDez()
{ uint8_t p =
finger.getImage();
if (p != FINGERPRINT_OK)
return -1;
p = finger.image2Tz();
if (p != FINGERPRINT_OK)
return -1; p =
finger.fingerFastSearch(); if
(p != FINGERPRINT_OK)
{ lcd.clear(); lcd.print("Finger
Not Found");
lcd.setCursor(0,1);
```



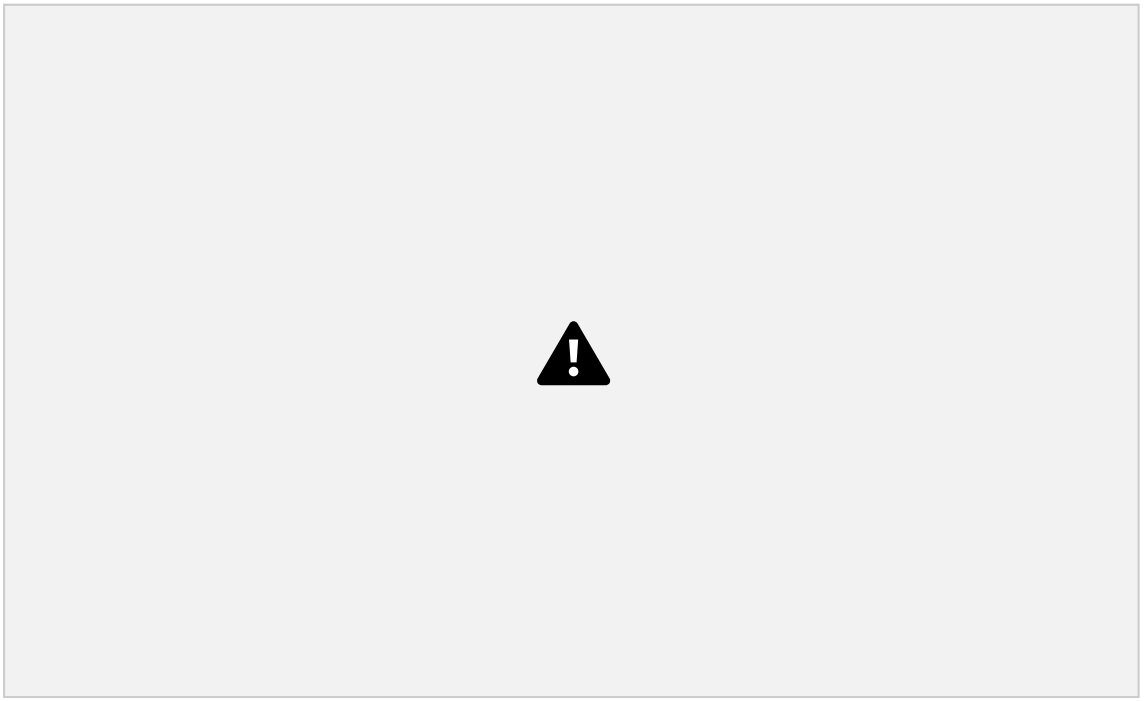
```
lcd.print("Try Later");
delay(2000); return -1;
}
// found a match!
Serial.print("Found ID #");
Serial.print(finger.fingerID);
return finger.fingerID;
}
uint8_t deleteFingerprint(uint8_t id)
{ uint8_t p = -
1; lcd.clear();
lcd.print("Ple
ase wait"); p
=
finger.delete
Model(id);
if (p == FINGERPRINT_OK)
{
Serial.println("Deleted!");
lcd.clear();
lcd.print("Finger
Deleted");
lcd.setCursor(0,1);
lcd.print("Successfully");
delay(1000); } else
{
Serial.print("Something  Wrong");
lcd.clear();   lcd.print("Something
Wrong");          lcd.setCursor(0,1);
lcd.print("Try    Again    Later");
delay(2000); return p; } }
void download(int eepIndex)
```

```
    {
    if(EEPROM.read(eepIndex) != 0xff)
    {
    Serial.print("T->");
    if(EEPROM.read(eepIndex)<10)
    Serial.print('0');
    Serial.print(EEPROM.read(eepIndex++));
    Serial.print(':');
    if(EEPROM.read(eepIndex)<10)
    Serial.print('0');
    Serial.print(EEPROM.read(eepIndex++));
    Serial.print(':');
    if(EEPROM.read(eepIndex)<10)
    Serial.print('0');
```



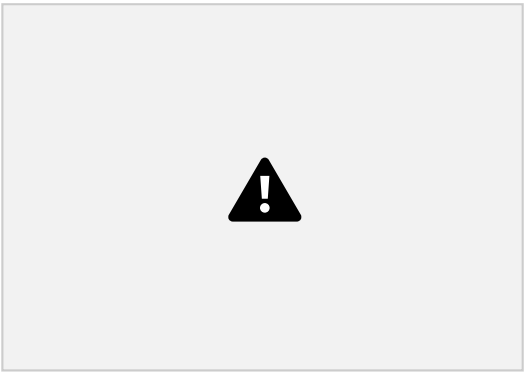
```
Serial.print(EEPROM.read(eepIndex++));
Serial.print(" D->"); if(EEPROM.read(eepIndex)<10)
Serial.print('0');
Serial.print(EEPROM.read(eepIndex++));
Serial.print('/');
if(EEPROM.read(eepIndex)<10)
Serial.print('0'); Serial.print(EEPROM.read(eepIndex++));
Serial.print('/');
Serial.print(EEPROM.read(eepIndex++)<<8 | EEPROM.read(eepIndex++));  } else
{
Serial.print("----------------------------");
}
Serial.print(" ");
}
```

Project Setup



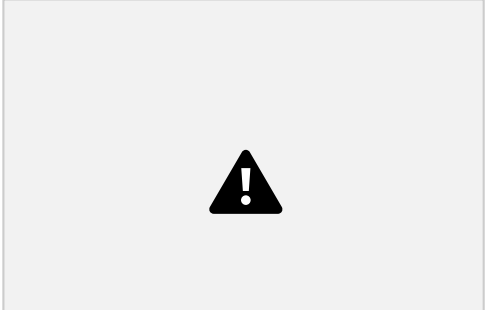
Application:

1. Airport Security



Making the journey through airport terminals more seamless for passengers is a goal shared by
airports around the world. Biometric technology to verify passenger identities has been used in several large international airports for a number of years and the technology is quickly spreading to other locations across the globe. In many airports, the top biometric modality choice for immigration control is iris recognition. In order to use iris recognition, travelers are first enrolled by having a photo of their iris and face captured by a camera. Then, their unique details are stored in an international database for fast, accurate identification at ports of entry and exit that use iris recognition for traveler identity verification. When travelling, instead of waiting in long queues to be processed, passengers simply walk into a booth and look into an iris camera. The camera then photographs the iris and a software program then matches the details with the information stored on the database.

2. Time and Attendance



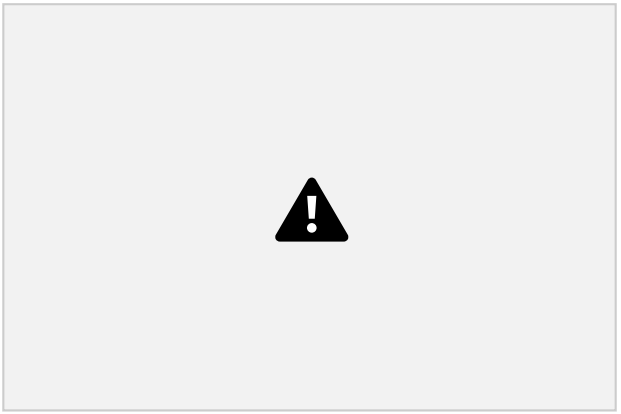
Workforce management is another field where the use of biometrics is on the rise. Fraudulent employee time and attendance activities are a common phenomenon in organizations throughout the world. According to

an American Payroll Association study, the average employee reportedly steals approximately 4 and a half hours per week, which is equivalent to 6 weeks' vacation if extrapolated over a year. To solve this issue, companies are implementing biometric time clocks on their work sites. A biometric time and attendance system is the automated method of recognizing an employee based on a physiological or behavioral characteristic. The most common biometric features used for employee identification are faces, fingerprints, finger veins, palm veins, irises, and voice patterns. When an employee attempts identification by their biological traits, a biometric hardware device compares the new scan to all available templates in order to find an exact match. Even government organizations now rely on biometrics for ensuring timely attendance of staff and accurate payroll calculations.

3. Law Enforcement

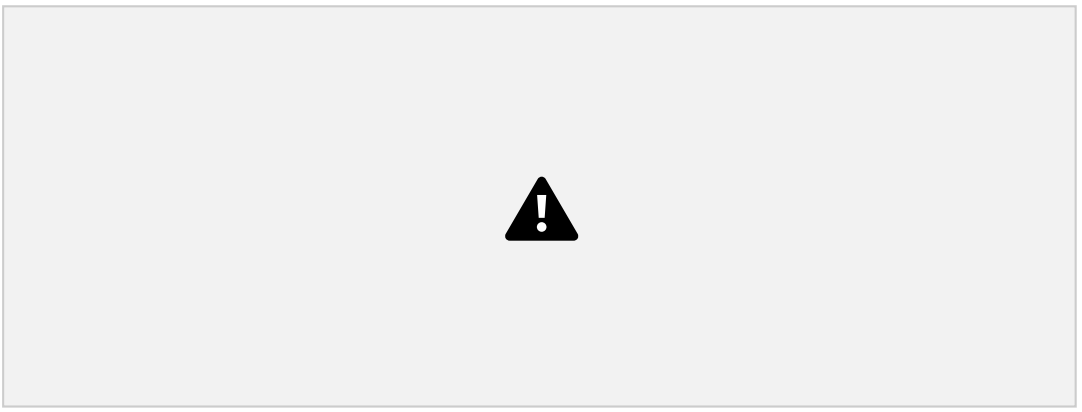
Organizations like the Federal Bureau of Investigations (FBI) and Interpol have been using biometrics in criminal investigations for years. Today, biometrics is widely used by law enforcement agencies across the world for the identification of criminals. In

2008, the Chinese Police adopted an ABIS solution to allow forensic fingerprint examiners the ability to cross check inmate identities for possible matches within the database Biometrics is also widely used for jail and prison management. Biometrics provides a modern solution by which the Jail Authority, Public Safety Departments, and Governments can safely and securely manage prisoner identities.



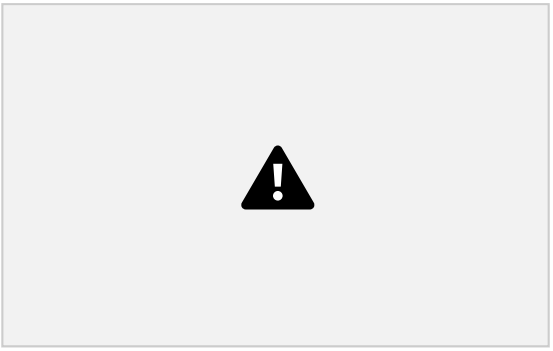
4. Access Control & Single Sign On (SSO)

The primary reason behind more and more organizations and personnel across the globe adopting biometric technology for access control and Single Sign On (SSO) is because traditional authentication tactics like passwords are insufficient for personal identification. Passwords only provide evidence or proof of knowledge whereas biometrics provides unique advantages because it relies on identifying someone by "who they are" compared to "what you know "or "what you have." Today, biometrics is widely used around the world for home access control, mobile phone access, vehicle access authentication and Single Sign On (SSO).



5. Banking – Transaction Authentication

Biometrics in banking has increased a great deal in the last few years and is being implemented by banks throughout the world. As global financial entities become more digitally-based, banks are implementing biometric technology to improve customer and employee identity management in an effort to combat fraud, increase transaction security, and enhance customer convenience. Customers are also fedup with identity theft and the inconveniences associated with constantly having to prove their identities. As a result, more and more customers are looking for banks that have biometric authentication in place prompting banks to more closely research the technology for implementation.



6. Surveillance

Surveillance is simply keeping tabs of a large group of people, and from there, determining any abnormal behavior from an established baseline. In this instance, it is Facial Recognition which is used the most, and in fact, is the most feared amongst the American public. The primary reason for this is that this modality can be secretly deployed into CCTV cameras, in order to positively identify any known criminals or suspects.

Modification/Improvement/Method:

A method for the biometric attendance system of a user is given, which includes a computer device and a biometric matching service. The approach entails acquiring information from a user's biometric sample in order to find data sources that are relevant to the user. The technique also includes obtaining a plurality of biometric samples from the user by utilising the user's relevant data sources. To evaluate whether the captured biometric sample represents the user, the technique includes comparing the captured biometric sample to a plurality of possible captured biometric samples.

Biometrics suffers from the fact that the matching algorithms cannot be compared to the hashes of passwords, as we said.

This means that two biometric measures cannot be compared with each other without them, at some point, being "in plaintext" in the memory of the device doing the matching. Therefore, biometric checks must be carried out on a trusted secure device, which means the alternatives are to have a centralized and supervised server, a trusted biometric device, or a personal security component.

Smart ID cards This security need is why tokens and smart cards (I.D.s or banking cards now) are the ideal companions for a biometric system. Numerous national identity cards (Portugal, Ecuador, South Africa, Mongolia, Algeria, etc.) now incorporate digital security features based on the "Match-on-Card" fingerprint matching algorithm. Unlike conventional biometric processes, the "Match-on-Card" algorithm allows fingerprints to be matched locally with a reference frame thanks to a microprocessor built into the biometric I.D. card without having to connect to a central biometric database (1:1 matching).

Integrating a fingerprint scanner into smart cards is another form of delivering a safe and convenient way to authenticate people. These biometric sensor cards open up a new dimension in identification with an easy-to-use, portable, and secure device.They were launched in 2018 for the first time by the Bank of Cyprus and Thales for EMV cards (contactless and contact payment). They use fingerprint recognition instead of a PIN code to authenticate the cardholder. There's more. The cards support access, physical or online identity verification services. As the user's biometric data is stored on the card, not on a central database, customer details are highly protected if the bank suffers a cyber-attack. Likewise, if the card was to become lost or stolen, the holder's fingerprint could not be replicated. Put it in another way: the biometric identifiers are checked locally and protected, as they are stored solely on the card. They never leave the card.

Conclusion:

Biometric Attendance systems offer higher security, convenience, accountability, and accurate audit trails – all attributes that motivate businesses to research and implement the technology for their own use. We believe that as time moves forward, we will see implementation of biometric technology continue to grow and be used in even more areas that touch our lives.

References used (avoid putting only links)

1. https://how2electronics.com/iot-biometric-fingerprint-attendance systemnodemcu/
2. https://www.researchgate.net/publication/340298787_Design Based_Fingerprint_Time_Attendance_System_Using_IOT_With_MCU_N ode_ESP8266
3. https://www.youtube.com/watch?v=v-t8AFjW08M

Contribution of each member in group

Roll Number	Name	Work Done for project
1913063	Hersh Vitekar	Report, Arduino code
1913072	Dhanshree Chavan	Report, Hardware implementation
1913078	Rahul Doshi	Report, Hardware implementation
1913100	Khushbu Parmar	Report, Arduino Code