A PROJECT REPORT ON
"SMART ATTENDANCE SYSTEM"

COURSE:
IT 478 : INTERNET OF THINGS

UNDER THE GUIDANCE OF
PROF. SANJAY SRIVASTAVA

Team Details

Name	ID
Riya Talwar	201501154
Skeny Patel	201501205

TABLE OF CONTENTS

1 Abstract

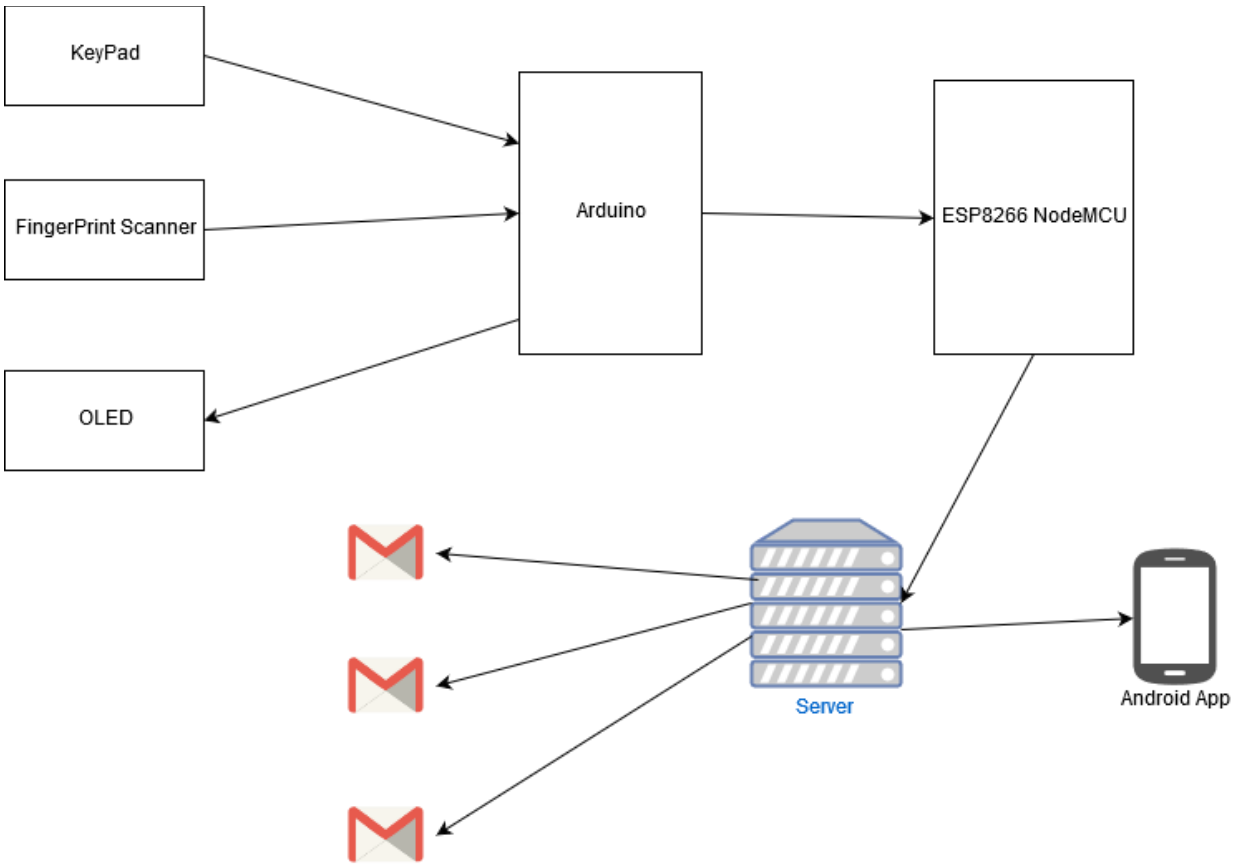
Attendance is one of the work ethics which is valued by most employers. In educational institutions also, attendance and academic success are directly related. Therefore, proper attendance management systems must be in place. Most of the educational institutions still use paper based attendance method to monitor the attendance. There is a need to replace these traditional methods of attendance recording with a more secure and robust system.

RFID based attendance systems were then introduced for attendance system. The problem here was that students tend to forget or misplace their cards. As a result, Fingerprint based automated identification system based are gaining popularity due to unique nature of fingerprints.

Through this project, we aim to build a *Smart Attendance System* which uses a fingerprint module to scan fingerprints of student in order to record their attendance. The attendance is taken course wise and as soon as the attendance is taken, it gets stored in the cloud database. This database is queried for fetching details of the student as well as the faculty. The fetched details are shown to the logged in user through an android application interface. Also, in case when a student takes a leave, he is notified regarding his absence through an email.

This Attendance System also tries to assure that students attending the lecture are punctual as once the person in charge for taking attendance presses the key denoting the end of attendance, all the students whose attendance was not marked are marked absent and immediately mails are sent to the students regarding their leave. Also, This Attendance System doesn't give unwanted results in certain unforeseen situations. Such as, in case when the person starts the attendance and then he realises that the lecture is cancelled, it is taken into consideration that none of the students receive mail for staying absent.

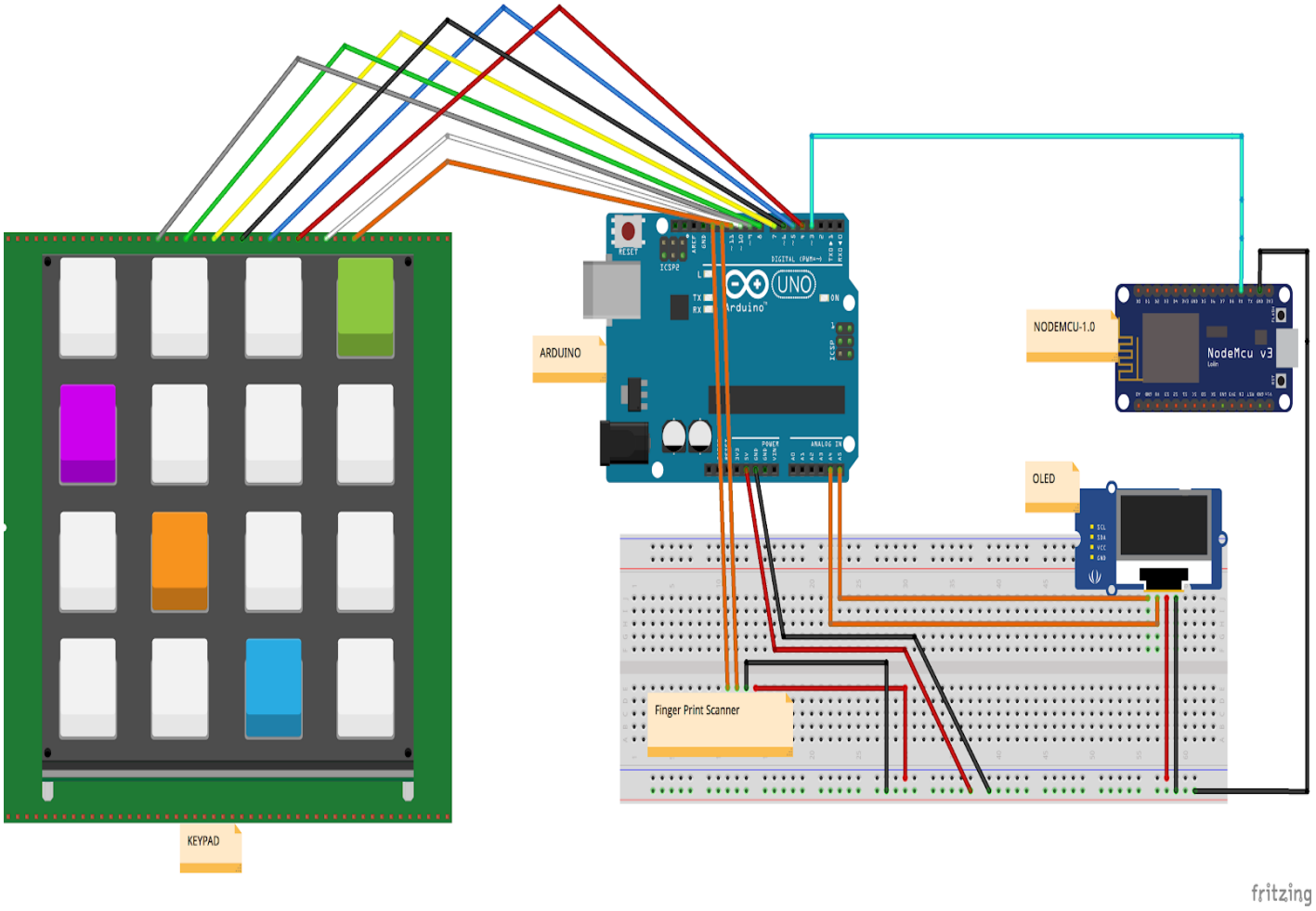
2 Block Diagram



3 Hardware Components

- Arduino Uno
- Finger Print Module
- NodeMCU-1.0 wifi module
- OLED
- Keypad
- Jumper wires
- Breadboard

4 Circuit Diagram



5 System Description

5.1 Software Design

- Arduino IDE
- phpMyAdmin Database
- Android Application
- Email

5.2 Prerequisites

- Active wifi network
- Student and Faculty details should be recorded in Database
- Student fingerprints should be enrolled in Fingerprint scanner module

5.3 Working

5.3.1 Arduino IDE codes

5.3.1.1 Arduino code

The following code is to be uploaded to arduino without connecting NodeMCU.

```
#include <Keypad.h>
#include <Adafruit_GFX.h>
#include <Wire.h>
#include <Adafruit_SSD1306.h>
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
SoftwareSerial mySerial(12, 13);
SoftwareSerial sw(2,3);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

#define OLED_RESET 4
```

```
Adafruit_SSD1306 display(OLED_RESET);

int c=0,c2=0;

const byte numRows= 4; //number of rows on the keypad
const byte numCols= 4; //number of columns on the keypad

/*keymap defines the key pressed according to the row and columns just as appears on the keypad*/
char keymap[numRows][numCols]={{'1','2','3','A'},{'4','5','6','B'},{'7','8','9','C'},{'*','0','#','D'}};

//Code that shows the the keypad connections to the arduino terminals
byte rowPins[numRows] = {9,8,7,6}; //Rows 0 to 3
byte colPins[numCols]= {5,4,10,11}; //Columns 0 to 3 3 changed 10, 2 to 11

String courseId="";
char keypressed;

//initializes an instance of the Keypad class
Keypad myKeypad= Keypad(makeKeymap(keymap), rowPins, colPins, numRows, numCols);

void setup()
{
 Serial.begin(115200);
 while (!Serial);  // For Yun/Leo/Micro/Zero/...
 delay(100);
 Serial.println("\n\nAdafruit finger detect test");

 // set the data rate for the sensor serial port
 finger.begin(57600);

 if (finger.verifyPassword()) {
  Serial.println("Found fingerprint sensor!");
 } else {
  Serial.println("Did not find fingerprint sensor :(");
  while (1) { delay(1); }
 }

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

  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);// initialize with the I2C addr 0x3C (for the 128x32)(initializing the
display)
```

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

}

//If key is pressed, this key is stored in keypressed variable
//If key is not equal to NO_KEY, then this key is printed out
//if count=17, then count is reset back to 0 (this means no key is pressed during the whole keypad scan process
void loop()
{
// display.print("Course ID:");
 //display.setCursor(0,10);

  keypressed=myKeypad.getKey();
  if(keypressed=='A')
  {
    display.clearDisplay();
    display.setCursor(0,0);
    display.print("Course ID:");
    display.setCursor(0,10);
    c2=1;
  }
   else if(keypressed=='#'&&c==0&&c2==1)
   {
    display.clearDisplay();
    display.setCursor(0,0);
    display.print("Start attendance");
    c=1;
   // while(1)
    //{
   // getFingerprintIDez();
   // }
   // delay(50);

   }
   else if(keypressed=='D')
   {
    display.clearDisplay();
    display.setCursor(0,0);
    display.print("Attendance over");
    c=0;
```

```
   mySerial.end();
   sw.begin(115200);
   String postdata="Dcourse_id=" + courseId ;
   Serial.println(postdata);
   sw.print(postdata);
   sw.println();
   sw.end();
   finger.begin(57600);
   courseId="";
  }
  else if(c==0&&c2==1)
  {
     keypadData(keypressed);
  }
 if(c==1)
 {
  // Serial.println(courseId);
  // Serial.println("startttttt");
   //display.clearDisplay();
   display.setCursor(0,10);
    int id=getFingerprintIDez();
    if(id!=-1)
    {
     mySerial.end();
      sw.begin(115200);
     String postdata="finger_id=" + (String)id + "&course_id=" + courseId +"&attendance="+ '1';
     Serial.println(postdata);
      sw.print(postdata);
     sw.println();
     sw.end();
     finger.begin(57600);
    }
    //delay(50);
 }
 display.display();
}

void keypadData(char keypressed)
{
 if(keypressed)
   display.print(keypressed);
  if(keypressed!=NO_KEY)
  {
```

```
if(keypressed=='0'||keypressed=='1'||keypressed=='2'||keypressed=='3'||keypressed=='4'||keypressed=='5'||keypressed=='6
'||keypressed=='7'||keypressed=='8'||keypressed=='9')
    courseId+=keypressed;
  }

}


uint8_t getFingerprintID() {
 Serial.println("hi");
 uint8_t p = finger.getImage();
 switch (p) {
  case FINGERPRINT_OK:
   Serial.println("Image taken");
   break;
  case FINGERPRINT_NOFINGER:
   Serial.println("No finger detected");
   return p;
  case FINGERPRINT_PACKETRECIEVEERR:
   Serial.println("Communication error");
   return p;
  case FINGERPRINT_IMAGEFAIL:
   Serial.println("Imaging error");
   return p;
  default:
   Serial.println("Unknown error");
   return p;
 }

 // OK success!

 p = finger.image2Tz();
 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!
  p = finger.fingerFastSearch();
  if (p == FINGERPRINT_OK) {
    Serial.println("Found a print match!");
  } else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    Serial.println("Communication error");
    return p;
  } else if (p == FINGERPRINT_NOTFOUND) {
    Serial.println("Did not find a match");
    return p;
  } else {
    Serial.println("Unknown error");
    return p;
  }

  // found a match!
  Serial.print("Found ID #"); Serial.print(finger.fingerID);
  Serial.print(" with confidence of "); Serial.println(finger.confidence);
  //delay(2000);
  return finger.fingerID;
}

// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez() {
  //Serial.println("ok");
  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)  return -1;

 // found a match!
   display.clearDisplay();
 display.setCursor(0,0);
 Serial.print("Found ID #"); Serial.print(finger.fingerID);
 Serial.print(" with confidence of "); Serial.println(finger.confidence);
 display.setCursor(0,10);
 display.print(finger.fingerID);
 display.print(" marked Present");
  //delay(5000);
 return finger.fingerID;
}
```

5.3.1.2 NodeMCU code

The following code is to be uploaded on NodeMCU without connecting Arduino.

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include<string.h>
#include <ESP8266HTTPClient.h>

// Update these with values suitable for your network.
const char* ssid = "Riya";
const char* password = "shreyariya";

int c=0;

WiFiClient client;
const char *host = "https://riyatalwar1697.000webhostapp.com";   // website or IP address of server

//PubSubClient client(wifiClient);

void setup_wifi() {
   delay(10);
   // We start by connecting to a WiFi network
   Serial.println();
   Serial.print("Connecting to ");
   Serial.println(ssid);
   WiFi.begin(ssid, password);
   while (WiFi.status() != WL_CONNECTED) {
     delay(500);
     Serial.print(".");
   }
```

```
  randomSeed(micros());
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void setup() {
 Serial.begin(115200);
 Serial.setTimeout(500);// Set time out for
 setup_wifi();
 //client.setServer(mqtt_server, mqtt_port);
 //reconnect();
}

void loop() {
  //client.loop();
  HTTPClient http;
  char bfr[101];
  memset(bfr,0, 101);
  Serial.readBytesUntil( '\n',bfr,100);

   if(bfr[0]=='D')
    {
       http.begin("http://riyatalwar1697.000webhostapp.com/testmail.php");            //Specify request destination
       http.addHeader("Content-Type", "application/x-www-form-urlencoded");
       String s="";
       for(int i=1;i<sizeof(bfr);i++)
       {
         s+=bfr[i];
       }
       Serial.println(s);
       int httpCode = http.POST(s);   //Send the request
       String payload = http.getString();
       Serial.println(httpCode);   //Print HTTP return code
       Serial.println(payload);
       http.end();  //Close connection
    }
    else
    {
      http.begin("http://riyatalwar1697.000webhostapp.com/postdemo.php");            //Specify request destination
      http.addHeader("Content-Type", "application/x-www-form-urlencoded");
      int httpCode = http.POST(bfr);   //Send the request
      String payload = http.getString();
      Serial.println(httpCode);   //Print HTTP return code
      Serial.println(payload);
      http.end();  //Close
      Serial.println(bfr);
    }
 }
```

5.3.2 Code Explanation

After making the initial hardware connections and uploading the respective codes to arduino and nodemcu, further connections are made which consists of connecting nodemcu with arduino. It is to be noted that the arduino to nodemcu serial connection is via serial ports 2 and 3. Whereas, the fingerprint sensor to arduino serial connection is via serial ports 12 and 13. And, at a time only one serial connection is active.

Initially, the wifi connection to the ssid mentioned is established. In this case, a mobile hotspot has been created with ssid =‘Riya’ and password =’shreyariya’. The serial connection corresponding to fingerprint sensor and arduino is active. The functioning of the attendance system starts when the person taking attendance presses the key ‘A’ on keypad. The instruction that follows this is that the person is prompted to enter the Course id whose attendance needs to be recorded. This prompt is displayed on the OLED display. Now, the person enters the course id using the keypad. After entering that, he needs to press the key ‘#’ on keypad which starts the attendance taking process. This is can be seen as there is a prompt displayed on the OLED display which says ‘Start Attendance’. Also, the fingerprint scanner module is now activated. So, now the person can start taking attendance of the students. The student’s fingerprint is taken by scanning the finger using fingerprint module. If the student had been previously enrolled in the fingerprint module database, a message is displayed on the OLED display saying ‘finger id marked present’ (the finger id is the unique fingerprint id stored in database). Now, this serial connection is ended and the serial connection between nodemcu and arduino is activated. The data consisting of the finger_id received, course_id whose attendance is taking place and attendance marked as 1 is sent to nodemcu via this serial connection. The nodemcu receives this data in bfr (buffer) and it initiates a http request to the postdemo.php script on server i.e. “http://riyatalwar1697.000webhost.com/postdemo.php ” . This bfr data is accepted in arguments of the php script. The php script has a query which inserts this data along with the timestamp in the course_attendance table. Now, the http link is closed using http.end() and as a result this serial connection also ends. Again the serial connection between arduino and fingerprint sensor is activated. This procedure of taking attendance and the henceforth set of actions continues till the person taking attendance presses key ‘D’ on the keypad which denotes the end of attendance. There is a message on the OLED display saying ‘Attendance over’. Also, this serial connection is closed and the serial connection between nodemcu and arduino is activated. The message

passed to nodemcu is a string consisting of ‘D’ (denoting attendance is over) and courseid. This message is sent because now we want to send emails to all those students who were absent when the attendance was taken and the lecture of this particular course was conducted. The message is stored in bfr (buffer). The courseid is extracted from bfr and stored in another string s. A http request is initiated to the postdemo.php script on server i.e. “http://riyatalwar1697.000webhost.com/testmail.php ”. This string s is accepted in argument of the php script. The php script has a query which extracts the details of students who have been marked absent for this course just now and email is sent to all these students stating there absence has been marked.

5.3.3 Database

Website name: riyatalwar1697.000webhostapp.com
Database name: id5262205_smartattendance

5.3.3.1 Php scripts

- conn.php - This is the script to connect the Android application to the database.

  Script:-

```
<?php

define('HOST_NAME','localhost');
define('USER_NAME','id5262205_riya');
define('PASSWORD','iotproject');
define('DATABASE','id5262205_smartattendance');

$con=mysqli_connect(HOST_NAME,USER_NAME,PASSWORD,DATABASE);

/*
if(!$con)
{
   echo "Database connection failed";
}
else
{
```

```
    echo "Database connection successful";
}
*/
?>
```

- alogin.php - This script is for checking the login of student. It takes the username and password as parameters,checks whether the username is there in the database and if the password is matched, it will take the student to his dashboard.

  Script:-

```
<?php

require_once 'conn.php';

$response = array();

//for login we need the username and password
if(isTheseParametersAvailable(array('username', 'password'))){
    //getting values
    $username = $_POST['username'];
    $password = $_POST['password'];

    //echo $username;
    //echo $password;

    //creating the query
    $stmt = $con->prepare("SELECT username,password FROM student WHERE username = ? AND password =
?");
    $stmt->bind_param("ss",$username, $password);
    $stmt->execute();
    $stmt->store_result();

    //if the user exist with given credentials
        if($stmt->num_rows > 0){
        $stmt->bind_result($username, $password);
        $stmt->fetch();
        $user = array(
        'username'=>$username,
        'password'=>$password,
            );
```

```
        $response['error'] = false;
        $response['message'] = 'Login successfull';
        $response['user'] = $user;
        }

        else{
            //if the user not found
            $response['error'] = false;
            $response['message'] = 'Invalid username or password';
        }
    }

echo json_encode($response);

function isTheseParametersAvailable($params){
foreach($params as $param){
if(!isset($_POST[$param])){
return false;
}
}
return true;
}
?>
```

- flogin.php - This script is similar to the alogin.php script, just that it is for the faculty login functionality.

  Script:-

```
<?php

require_once 'conn.php';

$response = array();

//for login we need the username and password
if(isTheseParametersAvailable(array('username', 'password'))){
    //getting values
    $username = $_POST['username'];
    $password = $_POST['password'];

    //creating the query
```

```
        $stmt = $con->prepare("SELECT username,password FROM faculty WHERE username = ? AND password =
?");
        $stmt->bind_param("ss",$username, $password);
        $stmt->execute();
        $stmt->store_result();

        //if the user exist with given credentials
                if($stmt->num_rows > 0){

           $stmt->bind_result($username, $password);
           $stmt->fetch();

           $user = array(
           'username'=>$username,
           'password'=>$password,
                   );

        $response['error'] = false;
        $response['message'] = 'Login successfull';
        $response['user'] = $user;
        }

        else{
           //if the user not found
           $response['error'] = false;
           $response['message'] = 'Invalid username or password';
        }
     }

  echo json_encode($response);

  function isTheseParametersAvailable($params){

  foreach($params as $param){
  if(!isset($_POST[$param])){
  return false;
  }
  }
  return true;
  }
  ?>
```

- fpage.php - This script generates the faculty page which shows the courses he teaches and the total attendance for his lectures for different dates.

Script:-

```
<?php

require_once 'conn.php';

    //getting values
    $username = $_POST['username'];
    //$password = $_POST['password'];
    //$username='crypto';
    //echo $username;

    $stmt = $con->prepare("SELECT course_id,course_name,total_attendance,students,date FROM (SELECT
course_id,username,course_name,name,total_attendance,students,date FROM (SELECT
course_id,cast(timestamp as date) as date,COUNT(attendance) as
total_attendance FROM course_attendance GROUP BY course_id,cast(timestamp
as date)) AS a NATURAL JOIN (SELECT * FROM faculty) AS b) AS k where username = '$username' ");

        //executing the query
        $stmt->execute();

        //binding results to the query
        $stmt->bind_result($course_id,$course_name,$attendance,$students,$date);

        $products = array();

        //traversing through all the result
        while($stmt->fetch()){
                $temp = array();
                $temp['course_id'] = $course_id;
                $temp['course_name'] = $course_name;
                //$temp['name'] = $name;
                $temp['attendance'] = $attendance;
                $temp['students'] = $students;
                $temp['date'] = $date;
                array_push($products, $temp);
}
 echo json_encode($products);

 ?>
```

- spage.php - This script will generate the student data which would include his subjects, number of lectures he/she attended and the total number of lectures.

Script:-

```
<?php

require_once 'conn.php';

    //getting values
    $username = $_POST['username'];

    $stmt = $con->prepare("SELECT course_id,course_name, total_attendance,lecs FROM (SELECT
course_id,course_name,finger_id, total_attendance,lecs
FROM (SELECT course_id,finger_id,COUNT(attendance) as total_attendance
FROM course_attendance GROUP BY course_id,finger_id) AS a NATURAL JOIN
(SELECT * FROM faculty) AS b) AS k NATURAL JOIN (SELECT name,s_id,finger_id
FROM student) AS l where s_id = '$username' ");

        //executing the query
        $stmt->execute();

        //binding results to the query
        $stmt->bind_result($course_id,$course_name,$attendance,$lecture);

        $products = array();

        //traversing through all the result
        while($stmt->fetch()){
                $temp = array();
                $temp['course_id'] = $course_id;
                $temp['course_name'] = $course_name;
                $temp['attendance'] = $attendance;
                $temp['lecture'] = $lecture;
                array_push($products, $temp);
}

$sql = $con->prepare("SELECT course_id,course_name,lecs FROM (SELECT * FROM (SELECT
course_id,finger_id FROM
course_taken AS x WHERE NOT EXISTS(SELECT
course_id,course_name,finger_id, total_attendance,lecs FROM (SELECT
```

```
course_id,finger_id,COUNT(attendance) as total_attendance FROM
course_attendance GROUP BY course_id,finger_id) AS a NATURAL JOIN (SELECT
* FROM faculty) AS b WHERE x.course_id=course_id and x.finger_id=finger_id))
AS y NATURAL JOIN (SELECT course_name,course_id,lecs FROM faculty) AS z) AS
k NATURAL JOIN (SELECT name,s_id,finger_id FROM student) AS l where s_id = '$username' ");

//executing the query
        $sql->execute();

        //binding results to the query
        $sql->bind_result($course_id,$course_name,$lecture);

        //traversing through all the result
        while($sql->fetch()){
                $temp = array();
                $temp['course_id'] = $course_id;
                $temp['course_name'] = $course_name;
                $temp['attendance'] = 0;
                $temp['lecture'] = $lecture;
                array_push($products, $temp);
}

 echo json_encode($products);

 ?>
```

- postdemo.php - This script is run when the request is sent from nodemcu. The main purpose of this script is to add a new tuple with finger_id, course_id, attendance received from arduino using esp wifi module into course_attendance table.

  Script:-

```
        <?php
//Creates new record as per request
   //Connect to database
define('HOST_NAME','localhost');
define('USER_NAME','id5262205_riya');
define('PASSWORD','iotproject');
define('DATABASE','id5262205_smartattendance');
   // Create connection
   $conn = new mysqli(HOST_NAME,USER_NAME,PASSWORD,DATABASE);
   // Check connection
```

```
if (!$conn) {
   die("Database Connection failed ");
}
else
{
   echo "ok";
}

//Get current date and time
//date_default_timezone_set('Asia/Kolkata');
// $d = date("Y-m-d");
//echo " Date:".$d."<BR>";
// $t = date("H:i:s");

if(!empty($_POST['finger_id']) && !empty($_POST['course_id']) && !empty($_POST['attendance']))
{
        $finger_id = $_POST['finger_id'];
        $course_id = $_POST['course_id'];
        $attendance = $_POST['attendance'];

          $sql = "INSERT INTO course_attendance (finger_id, course_Id, attendance)

                  VALUES ('".$finger_id."', '".$course_id."', '".$attendance."')";

                  if ($conn->query($sql) === TRUE) {
                     echo "OK";
                  } else {
                     echo "Error: " . $sql . "<br>" . $conn->error;
                  }
        }
        $conn->close();
?>
```

- testmail.php - This script is run when request is sent from nodemcu. The main purpose of this script is to send mails to all those students who had been absent in lecture of the course whose attendance had just been taken.

  Script:-

```
<?php

require_once 'conn.php';
```

```
if(isset($_POST["course_id"]))
{
$course_id = $_POST['course_id'];
echo $course_id;
}
else
{
   echo "data not received";
}
//$course_id='11478';
$date= date("Y-m-d");

//echo $date;

      $stmt = $con->prepare("SELECT s_id,course_id,course_name,emailid FROM (SELECT DISTINCT
      course_id,finger_id FROM (SELECT course_id,finger_id FROM (SELECT course_id FROM (SELECT
      finger_id,course_id,attendance,cast(timestamp as date) as date FROM course_attendance) AS a WHERE date='$date')
      AS b NATURAL JOIN (SELECT * FROM course_taken)AS c ) AS x WHERE NOT EXISTS (SELECT DISTINCT
      course_id,finger_id FROM (SELECT course_id,finger_id FROM (SELECT
      finger_id,course_id,attendance,cast(timestamp as date) as date FROM course_attendance) AS d WHERE date='$date')
      AS e WHERE x.course_id=course_id AND x.finger_id=finger_id)) AS y NATURAL JOIN (SELECT
      name,s_id,finger_id,emailid FROM student) AS z NATURAL JOIN (SELECT course_id,course_name FROM faculty)
      AS w where course_id='$course_id'");

      //executing the query
      $stmt->execute();

      //binding results to the query
      $stmt->bind_result($s_id,$course_id,$course_name, $emailid);

      //traversing through all the result
      while($stmt->fetch()){

      $msg = "This is to inform you that you have been absent for the $course_id - $course_name lecture on $date.";
      $msg = wordwrap($msg,70);
      //echo $msg;
      //echo $email;

      // send email
      mail($emailid,"IOT Smart Attendance System Test",$msg);
      }

      ?>
```

5.3.3.2 Tables

1. faculty

f_id	name	username	password	course_id	course_name	lecs	students
1	Sanjay Srivastava	iot	internetofthings	11478	IOT	1	10
2	Gagan Garg	crypto	cryptography	11325	Cryptography	2	10

2. student

finger_id	s_id	name	username	password	emailid
6	201501009	Prakhar	201501009	prakhar	prakhar@gmail.com
3	201501042	Vineeta	201501042	vineetameena	vineeta@gmail.com
8	201501104	Sravani	201501104	sravanikalangi	sravani@gmail.com
1	201501154	Riya	201501154	riyatalwar	riyatalwar1697@gmail.com
9	201501157	Nikisha	201501157	nikishapatel	nikisha@gmail.com
10	201501158	Brinda	201501158	brindajena	brinda@gmail.com
2	201501205	Skeny	201501205	skenypatel	skeny27@gmail.com
7	201501419	Abhin	201501419	abhinkakkad	abhinkakkad@gmail.com
4	201501420	Manthan	201501420	manthanmehta	manthan@gmail.com
5	201501454	Anushruti	201501454	anushrutipandey	annu@gmail.com

3. course_attendance

finger_id	course_id	attendance	timestamp
1	11325	1	2018-04-06 12:02:47
1	11325	1	2018-04-27 08:00:00
1	11478	1	2018-04-06 11:01:00
2	11325	1	2018-04-06 12:01:21
2	11478	1	2018-04-06 11:05:02
3	11325	1	2018-04-06 12:11:00
3	11478	1	2018-04-06 11:03:12
4	11325	1	2018-04-06 12:04:05
4	11478	1	2018-04-06 11:10:17
5	11478	1	2018-04-06 11:06:07
7	11325	1	2018-04-06 12:06:10
7	11478	1	2018-04-06 11:02:04
8	11325	1	2018-04-06 12:07:44
8	11478	1	2018-04-06 11:12:00
9	11325	1	2018-04-06 12:09:17
9	11478	1	2018-04-06 11:09:00
10	11478	1	2018-04-06 11:07:17

4. course_taken

finger_id	course_id
1	11325
1	11478
2	11325
2	11478
3	11325
3	11478
4	11325
4	11478
5	11325
5	11478
6	11325
6	11478
7	11325
7	11478
8	11325
8	11478
9	11325
9	11478
10	11325
10	11478

5.3.4 Android Application

The Android application is built for the purpose of viewing the status of the attendance for the student as well as the faculty. To view the corresponding logs, the users need to log in using their respective credentials. And, depending on who the user is, the system functionalities are delivered.

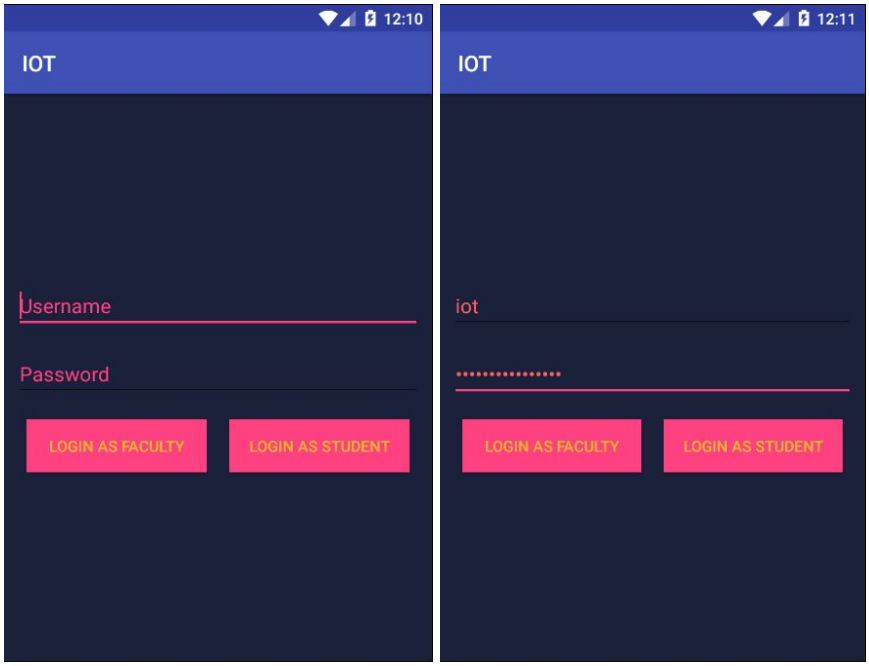
Android App Functionalities for the Student:

- Check his/her attendance for the courses that he has registered for.

Android App Functionalities for the Professor:

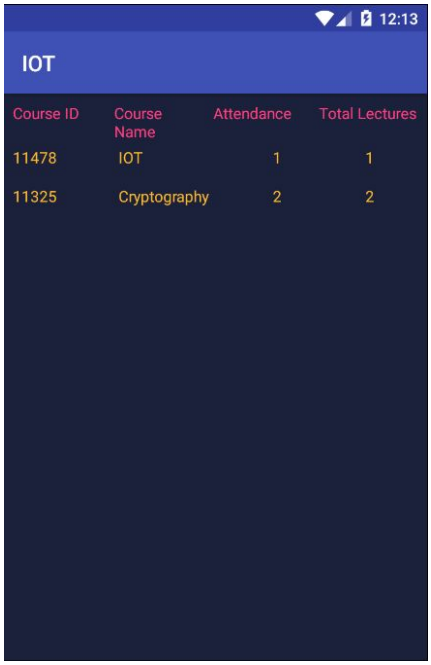
- Get the attendance statistics for each of the courses he teaches.
- The attendance is available datewise.

Login page



Faculty page

Student page



5.3.5 Email

Emails to all those students who have been absent for a lecture of a particular course are sent by running the testmail.php script of the server.
The format of the mail is given below:

Subject:- IOT Smart Attendance System Test

Mail Content:-
This is to inform you that you have been absent for the 11478 - IOT
lecture on 2018-05-02.
(In this content, the date, courseid and course name change depending on when and which lecture was conducted)

6 Future Scope

6.1 Hardware

- There is a possibility of forging fingerprints by using glue or latex. This can be avoided by developing a fingerprint reader with improved spoof-print detection. This reader is to be built using RaspiReader. Inside the RaspiReader's 3D-printed housing, LEDs shine light through an acrylic prism, on top of which the user rests their finger. The prism refracts the light so that the two Camera Modules can take images from different angles. The Pi receives these images via a Multi Camera Adapter Module feeding into the CSI port. Collecting two images means the researchers' spoof detection algorithm has more information to work with. (Link for this is mentioned in the references.)

6.2 Software

- Certain php scripts can be written to provide more functionalities to the users.
- A functionality can be added to provide faculty with a functionality to select a student from a given set of registered students for the course he is offering and then have a look at the to date attendance record.
- A functionality can be added to send a mail to a student if his attendance is below a certain acceptable level. This mail would be informing him about the consequences of having a low attendance.

7 References

- http://icircuit.net/arduino-connecting-arduino-uno-esp8266/2443
- https://circuits4you.com/2018/03/10/esp8266-nodemcu-post-request-data-to-website/
- https://create.arduino.cc/projecthub/tech-duino/using-4x4-keypad-with-arduino-2d22e9
- https://randomnerdtutorials.com/guide-for-oled-display-with-arduino/
- http://www.instructables.com/id/fingerprint-arduino-with-16x2-LCD/
- https://www.raspberrypi.org/blog/raspireader-fingerprint-scanner/

GITHUB LINK TO PROJECT:
https://github.com/RiyaTalwar/Smart-Attendance-System-IOT.git