SMART MANUFACTURING

PROJECT REPORT

SMART HEALTH MONITORING USING IOT

Group-5

K. Litheesh Kumar - S20190020218 (UG-3, ECE)

P. Govardhan Naik - S20190010135 (UG-3, CSE)

Srikanth Yadav - S20190010026 (UG-3, CSE)

K Rokesh Reddy - S20190010098 (UG-3, CSE)

K. Bhanu Chaitanya - S20190010097 (UG-3, CSE)

Abstract:

Internet of Things (IoT) based smart health monitoring system is a patient monitoring system in which a patient can be monitored 24 hours. In the present world, IoT is changing the infrastructure of technologies. Health monitoring systems are one of the most notable applications of IoT. Many types of designs and patterns have already been implemented to monitor a patient's health condition through IoT. In this paper, a review of IoT based smart health monitoring systems is presented. This review aims to highlight the common design and implementation patterns of intelligent IoT based smart health monitoring devices for patients.

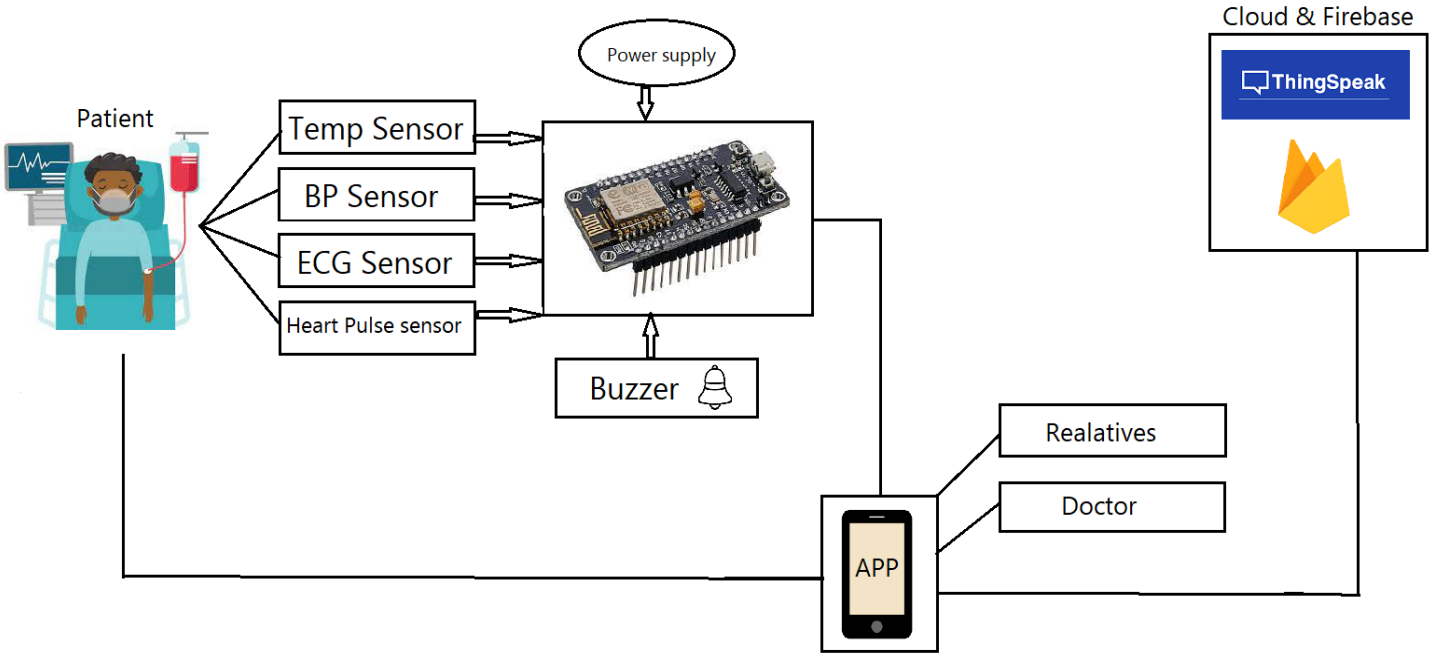
Continuously monitor the health i.e blood pressure, heart rate, ECG & temperature which will be collected using sensors. alert the user through a mobile app with respect to the collected health data using ML models. We build a health alert system which alerts the user about the health problems with the trained data set and deploy it.

Introduction:

IoT health monitoring has 4 sensors. First one is a temperature sensor, second is Heartbeat sensor and the third one is ECG sensor. We also measure the BP using sensors. This project is very useful since the doctor can monitor patient health parameters just by visiting a website or app. Nowadays many IoT apps are also being developed. So now the doctor or family members can monitor or track the patient's health through the app. To operate an IoT based health monitoring system project you need a Wi-Fi connection, the microcontroller or the Arduino board connects to the Wi-Fi network using a Wi-Fi module.

When an emergency situation occurs an alarm will be triggered to the nearby hospital. An alert call with a message will be sent to the hospital and immediate relatives of the patient who are registered in the app. So that in the absence of Wifi Network also this model will be helpful to indicate the emergency situation of the patient.

Block diagram of the project:



Phase wise implementation:

Phase-1:

- Connections of all sensors like Temperature sensors, BP Sensor etc.
- Sending values to MQTT cloud
- Building the basic UI interface of the app.

Phase-2:

- Comparing the various ML Algorithms error rate and implementing the algorithm with the least error rate.
- Alerting through Alarm to near Hospital
- Creating a App for video call and to check the readings of patient

Materials Used:

Hardware components:

- BP sensor
- Heart rate sensor
- ECG Sensor
- Temperature sensor
- Buzzer
- Ardiuno board
- Nodemcu

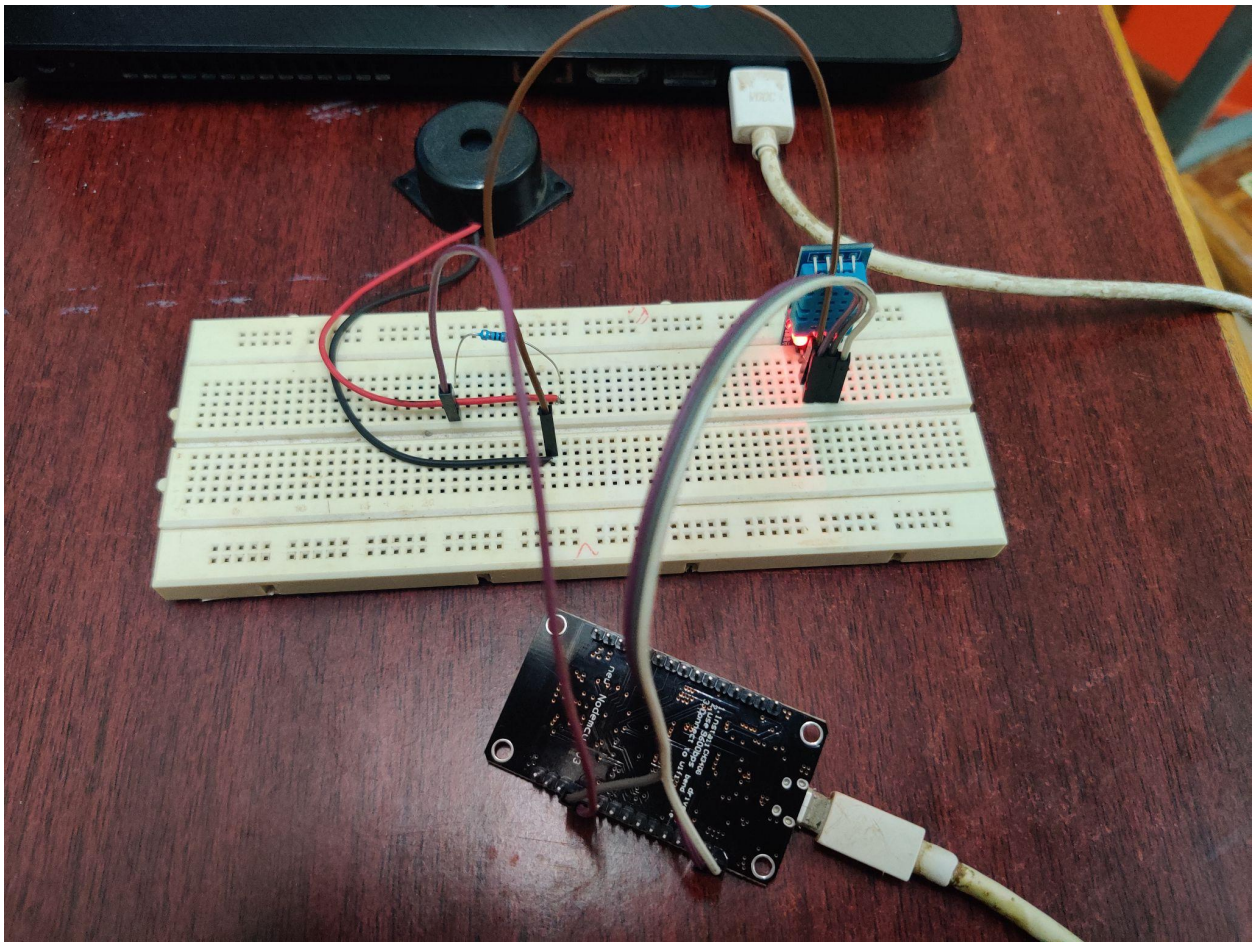
Software components:

- Google Collab / Jupyter Notebook
- Firebase
- Adafruit
- Fire Store
- Flutter(Dart)
- Arduino IDE

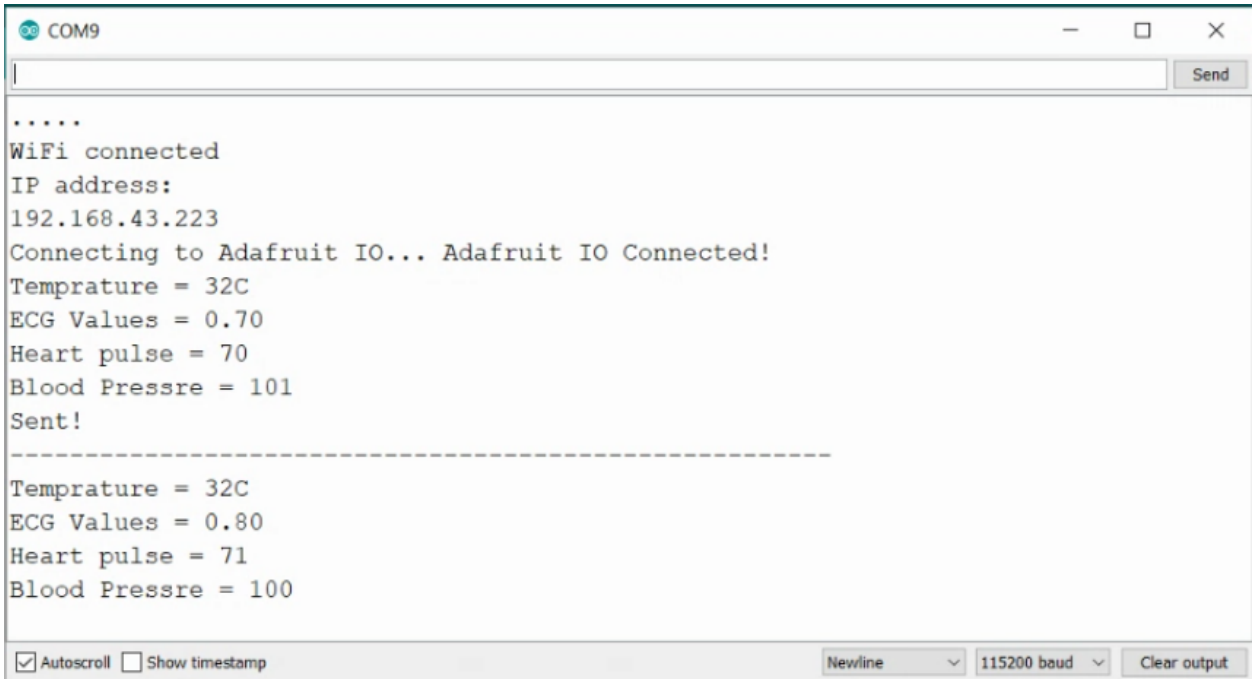
Procedure:

- We are collecting data from all the sensors and data will be sent to the MQTT cloud using NodeMCU.
- Machine Learning algorithms would be using the cloud data to filter the patient's data based on the severity level of the health data.
- Then the data is sent to Firestore to upload in the App.
- We are creating an App in which users can login with email id and password.
- In the app all the details will be displayed and when the emergency situation occurs, an alert call and notification will be sent to the doctor and immediate relations of the patient.
- A buzzer will be triggered at the nearby hospital to indicate the emergency of the patient.

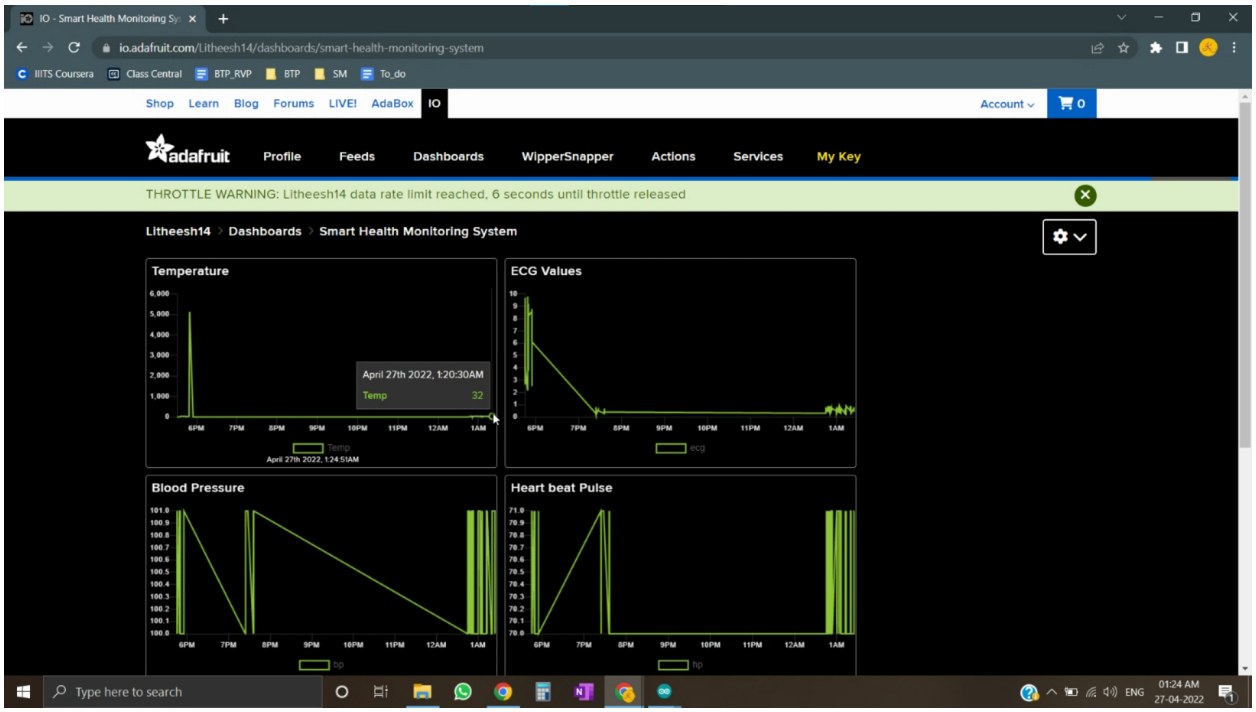
Screenshots of Hardware connections:



Serial Monitor Output:



MQTT Cloud output screenshot:

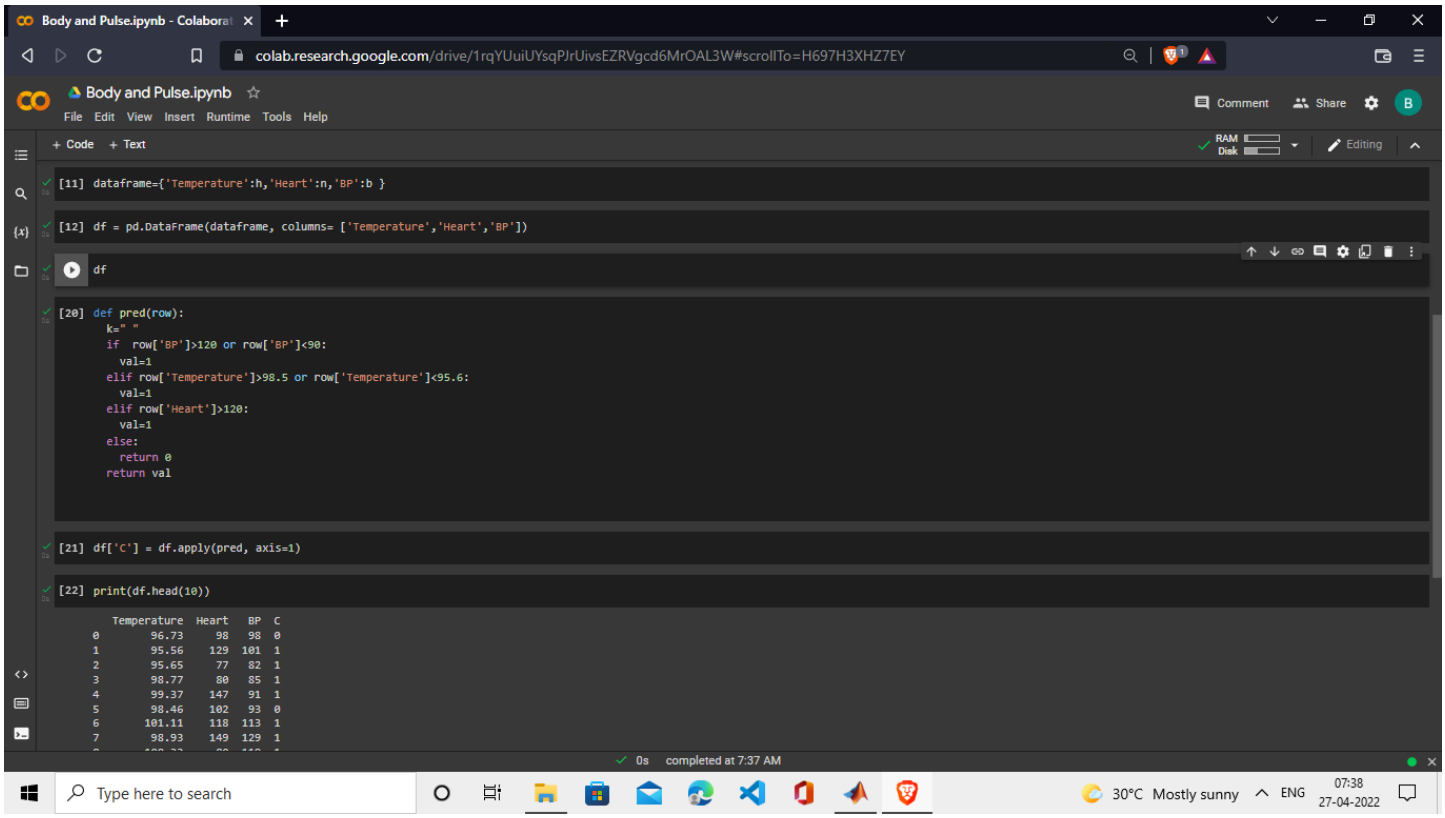


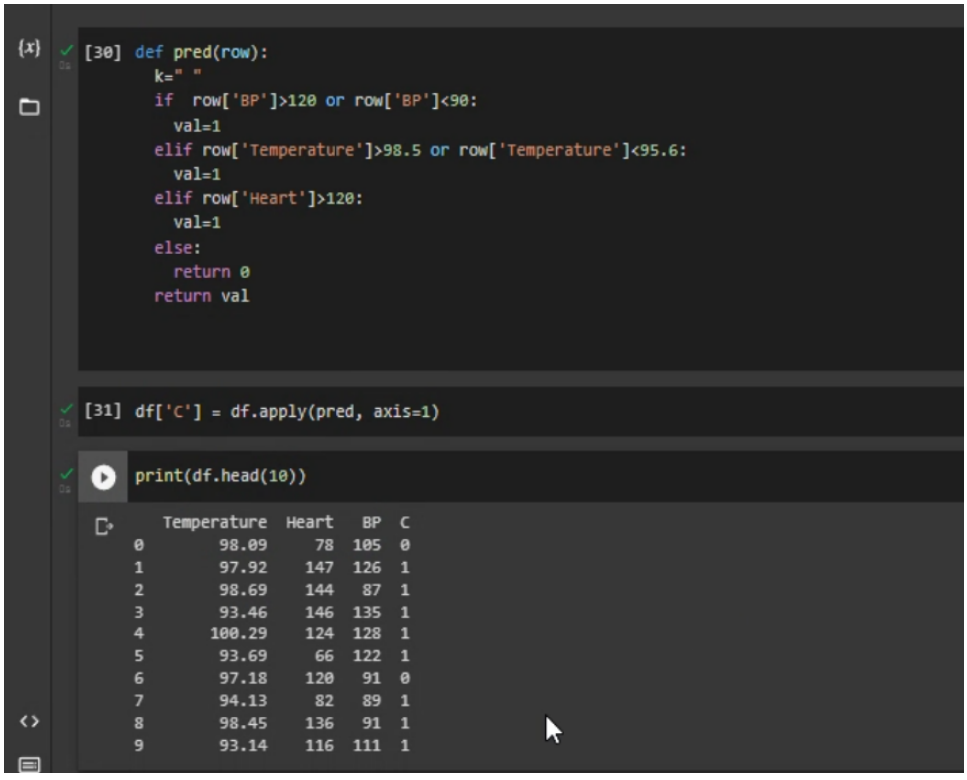
- In Adafruit cloud we created feeds for the output data and analyzed the conditions of patients from the output from sensors.
- We plotted them as graphs in the Smart Health Monitoring System Dashboard.

Code of software implementation:

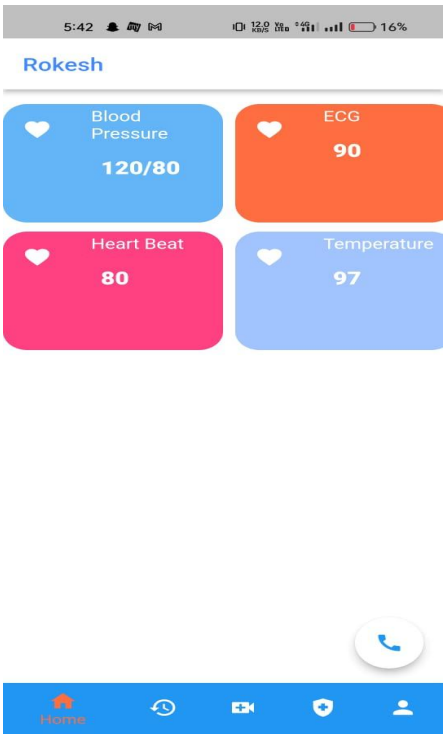
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import random
b=list()
for d in range(50):
  b.append(random.randint(80,135))
h=list()
for i in range(50):

h.append(round(random.uniform(93.00,101.12),2))
n=list()
for i in range(50):
  n.append(random.randint(65,150))
dataframe={'Temperature':h,'Heart':n,'BP':b }
df = pd.DataFrame(dataframe, columns=
['Temperature','Heart','BP'])
def pred(row):
      k=" "
  if  row['BP']>120 or row['BP']<90:
    val=1
  elif row['Temperature']>98.5 or
row['Temperature']<95.6:
    val=1
  elif row['Heart']>120:
    val=1
  else:
    return 0
  return val
df['C'] = df.apply(pred, axis=1)
print(df.head(10))
```

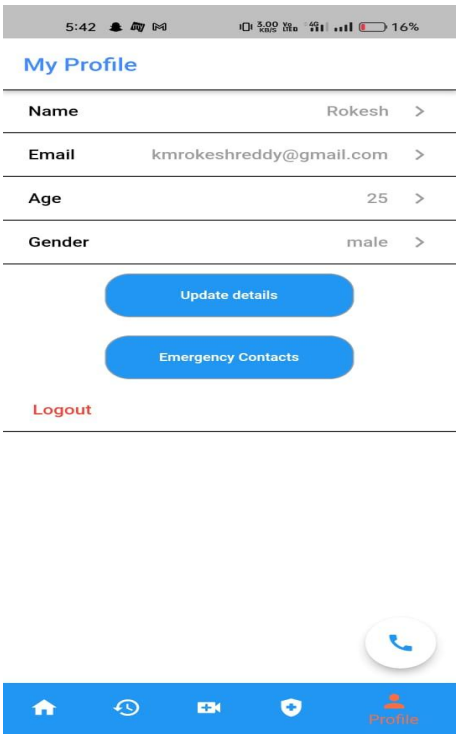
Screenshots of Software output:





Mobile APP screenshots:





- We made an App for the Smart Health monitoring System Where it has 5 subdivisions.
- First one is Profile where we can see the sensor readings of the patient.
- Second One is History data where we can see the patients health history.
- Video call option to doctor is given in the third subdivision part.
- Next, the doctor history is given in there the specialization of the doctor and the hospital career details are visible.
- Finally in the last part the emergency contacts of the patients who are registered for the patient emergency conditions.

Code for Hardware implementation:

```
#include <SimpleDHT.h>      // Data ---> D3
#include <ESP8266WiFi.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"

// WiFi parameters
#define WLAN_SSID       "Redmi 5A"
```

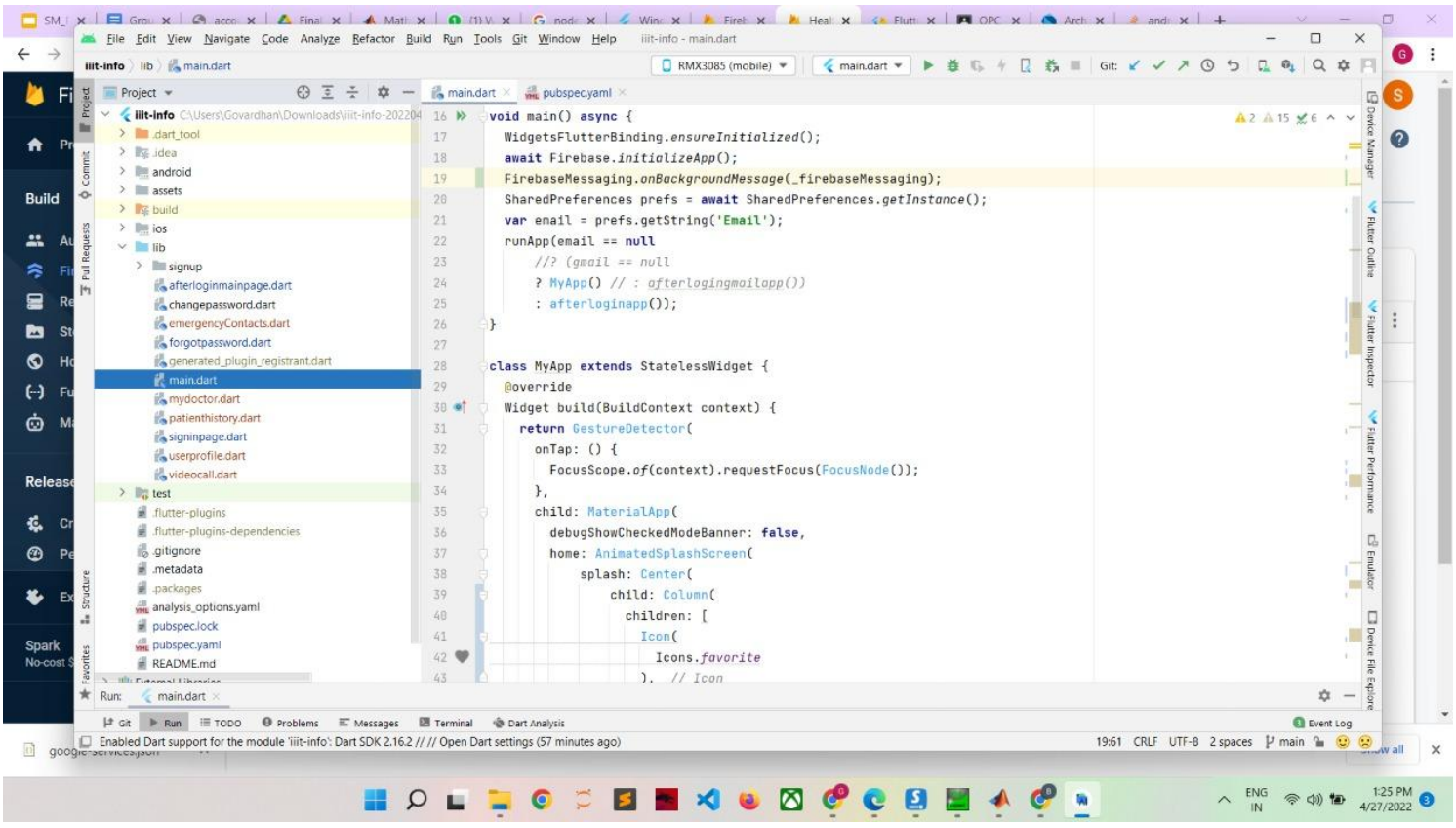
```
#define WLAN_PASS       "k.pallavi"

// Adafruit IO
#define AIO_SERVER      "io.adafruit.com"
#define AIO_SERVERPORT  1883
#define AIO_USERNAME    "Litheesh14"
#define AIO_KEY         "aio_PqSR55Pa9OVypEN4NBComN0TS1fu"
WiFiClient client;
#define Buzzer_pin 2

// Setup the MQTT client class by passing in the WiFi
client and MQTT server and login details.
Adafruit_MQTT_Client mqtt(&client, AIO_SERVER,
AIO_SERVERPORT, AIO_USERNAME, AIO_KEY);
Adafruit_MQTT_Publish Temp = Adafruit_MQTT_Publish(&mqtt,
AIO_USERNAME "/feeds/Temp");
Adafruit_MQTT_Publish Ecg = Adafruit_MQTT_Publish(&mqtt,
AIO_USERNAME "/feeds/ecg");
Adafruit_MQTT_Publish BP = Adafruit_MQTT_Publish(&mqtt,
AIO_USERNAME "/feeds/bp");
Adafruit_MQTT_Publish HP = Adafruit_MQTT_Publish(&mqtt,
AIO_USERNAME "/feeds/hp");

int pinDHT11 = 0;
SimpleDHT11 dht11(pinDHT11);
byte temp = 0; //Stores temperature value

// connect to adafruit io via MQTT
void connect()
{
    Serial.print(F("Connecting to Adafruit IO... "));
    int8_t ret;
    while ((ret = mqtt.connect()) != 0)
    {
        switch (ret)
        {
            case 1: Serial.println(F("Wrong protocol"));
break;
            case 2: Serial.println(F("ID rejected"));
break;
            case 3: Serial.println(F("Server unavail"));
break;
            case 4: Serial.println(F("Bad user/pass"));
break;
```

```
            case 5: Serial.println(F("Not authed")); break;
            case 6: Serial.println(F("Failed to
subscribe")); break;
            default: Serial.println(F("Connection
failed")); break;
        }
        if(ret >= 0)
            mqtt.disconnect();
        Serial.println(F("Retrying connection..."));
        delay(10000);
    }
    Serial.println(F("Adafruit IO Connected!"));
}

void setup()
{
    Serial.begin(115200);
    Serial.println(F("Adafruit IO Example"));
    // Connect to WiFi access point.
    Serial.println(); Serial.println();
    delay(10);
    Serial.print(F("Connecting to "));
    Serial.println(WLAN_SSID);
    WiFi.begin(WLAN_SSID, WLAN_PASS);
    while (WiFi.status() != WL_CONNECTED)
    {
        delay(500);
        Serial.print(F("."));
    }
    Serial.println();
    Serial.println(F("WiFi connected"));
    Serial.println(F("IP address: "));
    Serial.println(WiFi.localIP());

    // connect to adafruit io
    connect();
}

void loop()   // ping adafruit io a few times to make sure
we remain connected
{
    if(! mqtt.ping(3))    // reconnect to adafruit io
    {
        if(! mqtt.connected())
```

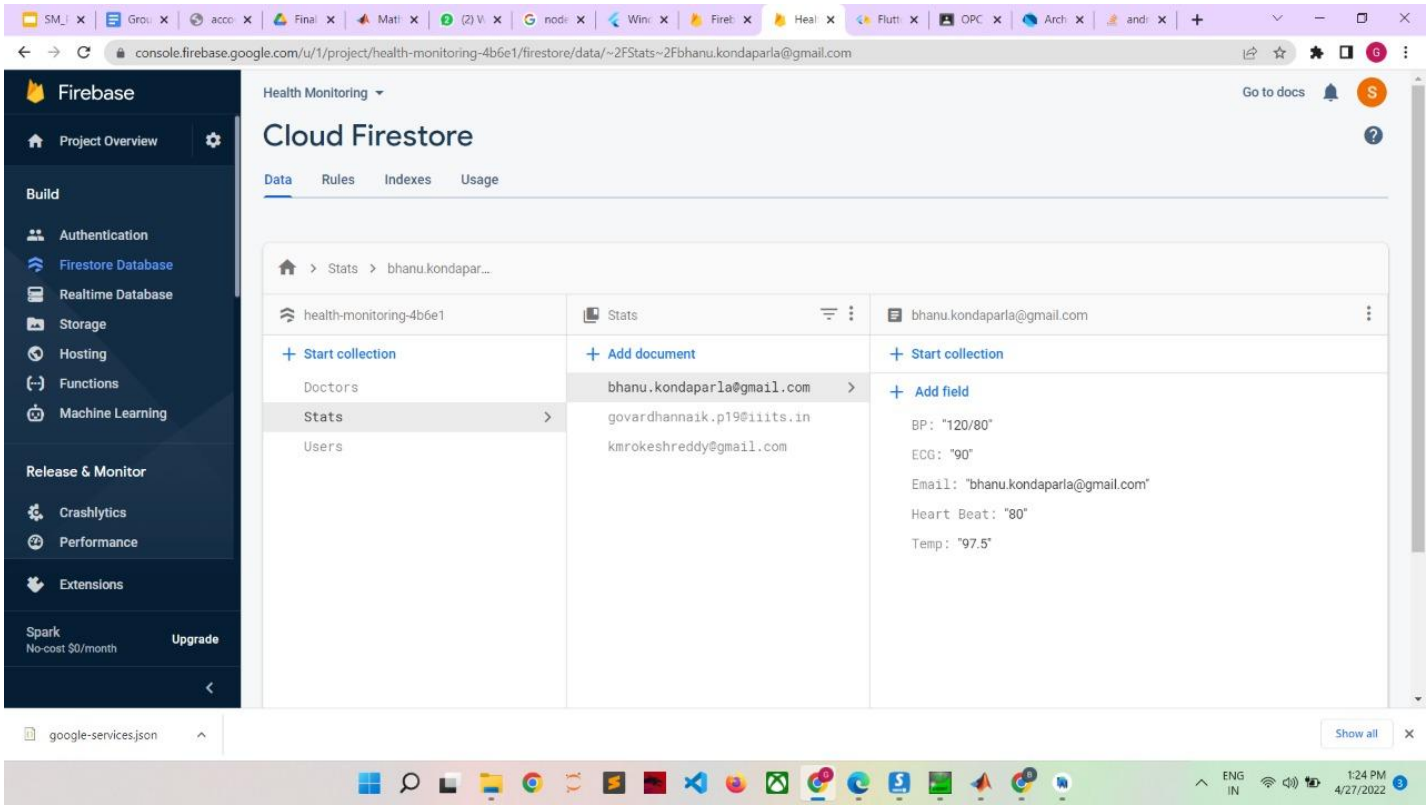
```
    connect();
}
dht11.read(&temp, NULL, NULL);
//temp = random(34, 37);
Serial.print("Temperature = ");
Serial.print((int)temp);
Serial.println("C");

float ecg = random(2, 10);
ecg = ecg/10;
Serial.print("ECG Values = ");
Serial.print((float)ecg);
Serial.println(" ");

int hp = random(70, 72);
Serial.print("Heart pulse = ");
Serial.print((int)hp);
Serial.println(" ");

int bp = random(100, 102);
Serial.print("Blood Pressure = ");
Serial.print((int)bp);
Serial.println(" ");
delay(5000);

if (! Temp.publish(temp))   //Publish to Adafruit
{
    Serial.println(F("Failed to publish Temperature"));
}
if (! Ecg.publish(ecg))       //Publish to Adafruit
{
    Serial.println(F("Failed to publish ECG"));
}
if (! BP.publish(bp))  //Publish to Adafruit
{
    Serial.println(F("Failed to publish Blood Pressure"));
}
if (! HP.publish(hp))  //Publish to Adafruit
{
    Serial.println(F("Failed to publish Heart pulse"));
}
else
{
    Serial.println(F("Sent!"));
```

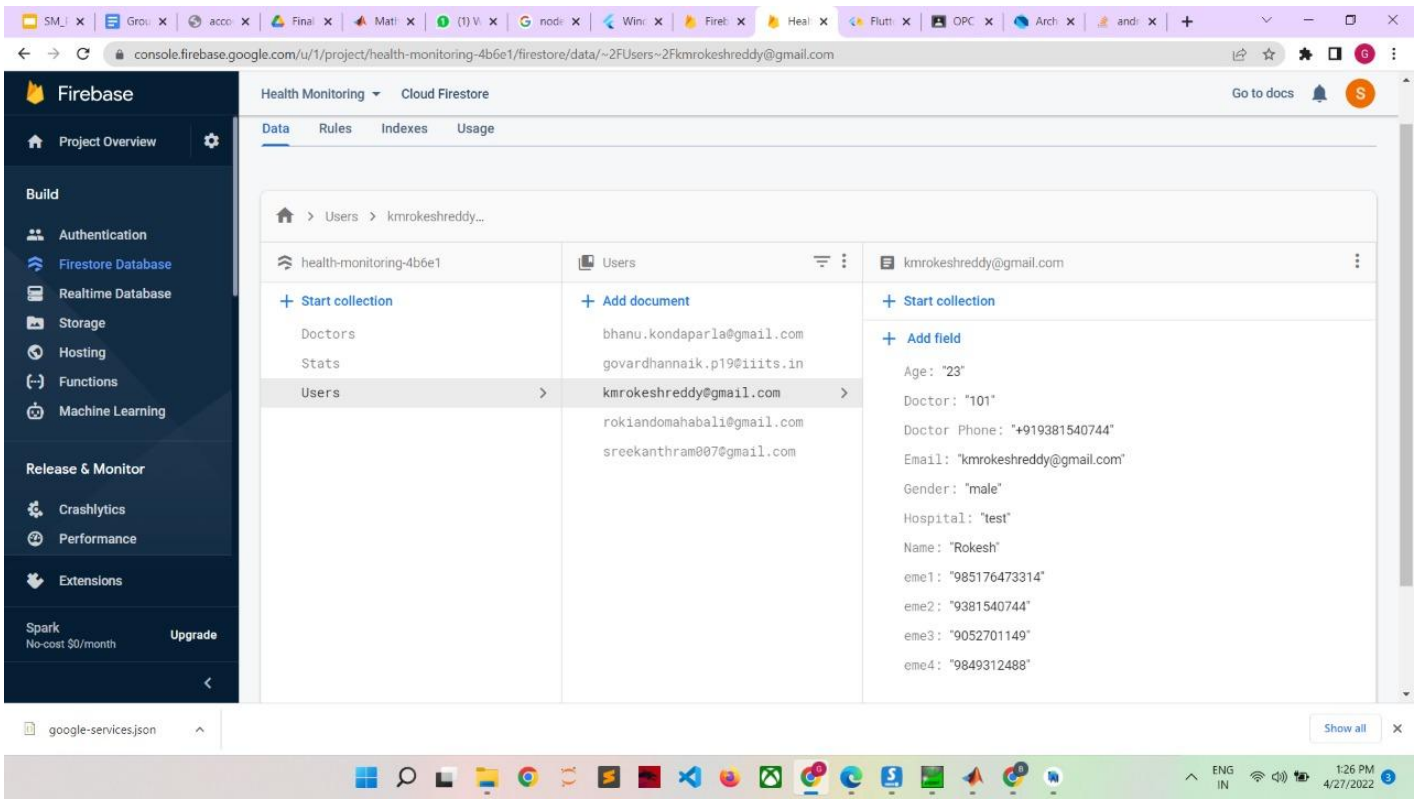
```
Serial.println("-----------------------------------------
------------");
    }

    if((int)temp < 30 || temp > 37 )
    {
        digitalWrite(Buzzer_pin, HIGH);
        delay(1000);
        digitalWrite(Buzzer_pin, LOW);
    }
}
```

Code for sending Notifications:



Firebase Screenshots:





Notifications from app:

Challenges:

- Faced difficulties in uploading the data to the cloud as publisher and subscribing it to the sensors.
- Getting a free paid MQTT Cloud is challenging.
- Sending the notifications to Immediate relatives of the patients through a mobile app.
- Video call option is not working properly.

Observations:

- We can know that ECG values of data of the patient will help us to understand the condition of the patient.
- An Effective Smart Health monitoring system is developed to monitor the up to date status of the patient irrespective of the presence of the doctor.
- The system collects the information like Temperature, Blood Pressure, Pulse rate, ECG Values of the patient and updates the same to the doctor.
- The doctor can monitor the progress of a patient's health now and then to advise about their health.

References:

1. Kolici, V., Spaho, E., Matsuo, K., Caballe, S., Barolli, L., Xhafa, F.: Implementation of a medical support system considering P2P and IoT technologies. In: Eighth International Conference on Complex, Intelligent and Software Intensive Systems, Birmingham, pp. 101– 106 (2014)

2. Yin, Y., Zeng, Y., Chen, X., Fan, Y.: The Internet of Things in healthcare: an overview.J. Ind. Inf. Integr. 1, 3–13 (2016)