## Abstract

Healthcare is given the extreme importance now a-days by each country with the advent of the novel corona virus. So in this aspect, an IoT based health monitoring system is the best solution for such an epidemic. IoT-enabled technologies enable the possibility of developing novel and noninvasive clinical support systems. This developed project presents the design and implementation of a health monitoring system using the Internet of Things (IoT) which will measure SpO2 (percentage of oxygen in the blood) and heart rate in BPM (Beat Per Minute) by interfacing MAX30100 Pulse Oximeter with NodeMCU ESP8266.To track the patient health, ESP8266 is interfaced to a LCD display and wi-fi connection to send the data to the web-server(wireless sensing node). Android app Blynk that will record and regularly update the data for both SPO2 & BPM. Even anyone can monitor the data from any part of the world as data are uploaded on server. As there is an availability of online data, so this project can be used to monitor the health of a patient in online. Thus Patient health monitoring system based on IoT uses internet to effectively monitor patient health and helps the user to save lives.

## Introduction

Health is always a major concern in every growth as the human race is advancing in terms of technology. Like the recent corona virus attack that has ruined the economy of China to an extent is an example how health care has become of major importance. In such areas where the epidemic is spread, it is always a better idea to monitor these patients using remote health monitoring technology. In todays social Health Insurance structure where patients stay at home after Operations they are monitored by a medical caretaker or a family member. Many people nowadays who work full time are facing a problem of monitoring their loved ones especially old age patients .

So to overcome this problem we are using this patient health monitoring system using IoT. Internet Of Things (IOT) based smart health monitoring system will help to measure various health-related parameters like body temperature, pulse, ECG, blood pressure etc. which will help to predict diseases. With tons of new healthcare technology start-ups, IoT is rapidly revolutionizing the healthcare industry. Keeping track of the health status of patient at home is a difficult task because of the busy schedules and our daily life work. Specially old age patients should be periodically monitored. So we propose an innovative system that automated this task with ease. Our system puts forward a smart patient health tracking system using Web Server so that the Patient health parameters like Heart Rate and Oxygen saturation can be monitored. This system can also be helpful to monitor the health of workers and people working at the merchant navy and to monitor the patients in the hospital who are kept under observation.

The core objective of this project is the design and implementation of a smart patient health tracking system that uses Sensors to track patient health and uses internet to inform their loved ones in case of any issues. The objective of developing monitoring systems is to reduce health care costs by reducing physician office visits, hospitalizations, and diagnostic testing procedure.

In this work, MAX30100 Pulse Oximeter is interfaced with NodeMCU ESP8266 . Bit Per Minute(bpm) and Oxygen saturation(SP02) is measured by the MAX30100 Pulse Oximeter Sensor. The code is processed by the ESP8266 and displayed on a I2C LCD display. The ESP8266 WiFi module connects to

WiFi and transfers data to a server for IoT device through Android app Blynk .Blynk is an application that runs over Android and IOS devices to control any IoT based application using Smartphones. It allows to create Graphical user interface for IoT application.

# Methodology

The goal of this project is to build a health monitoring system that can quickly measure a variety of health factors such as Heart beat (BPM) & oxygen saturation rate ($SPO_2$). This section is divided into four sections. In the very first section, the components used in the system are detailed . The system's block diagram is presented in the first subsection.

## 1. Hardware Components-

**ESP8266-** It is also called ESP8266 Wireless Transceiver which is a cost-effective, easy-to-operate, compact-sized & low-powered WiFi module, designed by Espressif Systems, supports both TCP/IP and Serial Protocol.It's normally used in IOT cloud based embedded projects and is considered the most widely used WiFi module beacuse of its low cost and small size.

Table1: Pins used in ESP8266 component -

| **Pin Name** | **Working** |
|---|---|
| D1 | Digital pin |
| D2 | Digital pin |
| 3v3 | External voltage source (3.3v) provided by this pin. |
| VV | Provide 5v output directly by USB |
| GND | Ground |

**Max30100 Pulse Oximeter-**It is a pulse oximetry and heart rate monitor sensor that can be powered by 1.8 V or 3.3 V power supply. A host microcontroller uses the I2C interface to communicate. It combines two LEDs, a photo detector, optimized optics, and low-noise analog signal processing to detect pulse oximetry and heart-rate signals. It is ultra-low-powered, making it suitable for systems operating on batteries.

Table2: Pins used in Max 30100 component -

| **Pin Name** | **Working** |
|---|---|
| VIN | Voltage input |
| GND | Ground pin |
| SCL | I2C – Serial Clock |
| SDA | I2C – Serial Data |

**I2C LCD display**- A 16 × 2 LCD display with I2C can show 16 characters in 2 lines. Four pins for the LCD display: VCC, GND, and SDA, as well as SCL.

Table3: Pins used in I2C LCD Display -

| **Pin Name** | **Working** |
|---|---|
| VCC | Supplies 5V power to LCD |
| GND | Ground pin |

| | |
|---|---|
| SCL | I2C – Serial Clock |
| SDA | I2C – Serial Data |
| LED | Header Pin which supplies power to backlight |

**LM 78705 Voltage Regulator**- The output voltage is kept constant using a voltage regulator integrated circuit (IC). A common voltage regulator integrated circuit is the 7805 Voltage Regulator, which is part of the 78xx series of fixed linear voltage regulators used to maintain such variations (IC).

Table 4: Pins used in LM 78705 Voltage Regulator -

| **Pin Name** | **Working** |
|---|---|
| INPUT | In this pin of the IC positive unregulated voltage is given in the regulation. |
| GND | This pin is neutral for equally the input and output. |
| OUTPUT | The output of the regulated 5V is taken out at this pin of the IC regulator. |

**Jumper wires**- Flexible wire with connectors on both ends can be linked to additional jumper wires or a pin header without the need for soldering.

**Bread board**- It's a construction base.

## 2. Circuit Diagram-

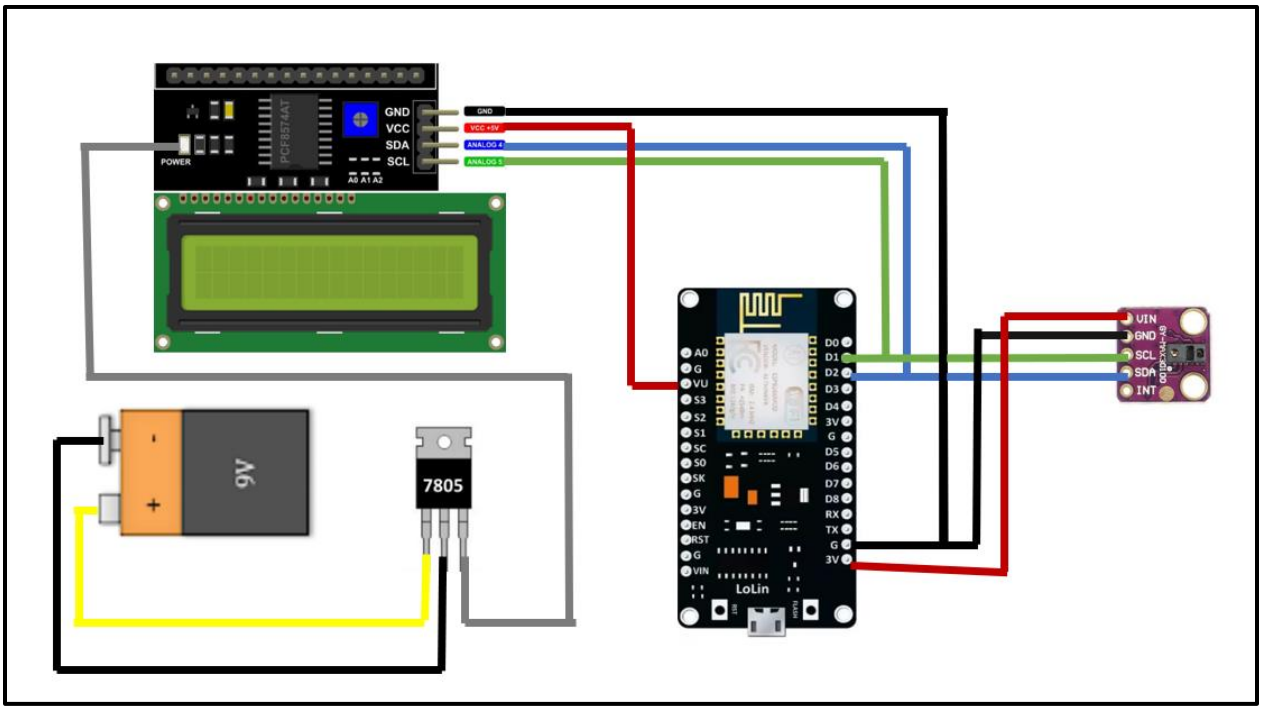


Figure1: Circuit diagram of health monitoring system

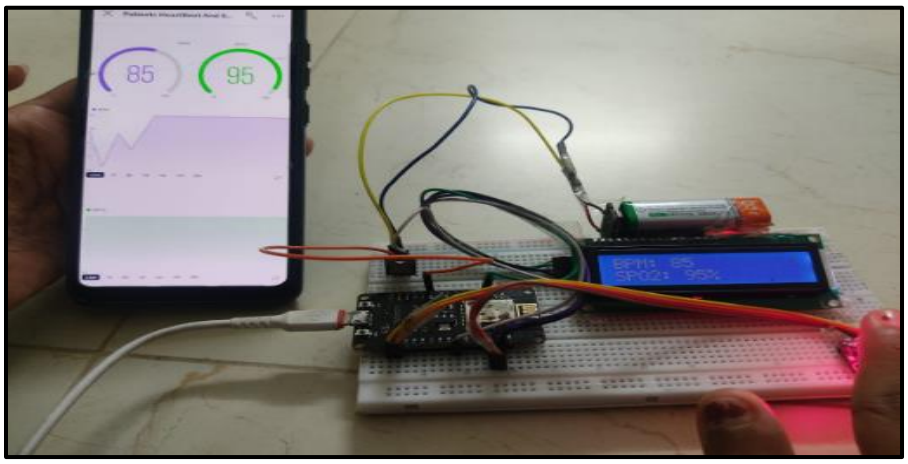


Figure 2: Prototype of the health monitoring system.

### 2.1 Connections-

1. MAX30100 to ESP8266-
   Vin to 3v3
   SCL to D1
   SDA to D2
   GND to GND
2. I2C LCD Display to LM 7805 Voltage Regulator-
   LED to Output
3. 9v Battery to LM 7805 Voltage Regulator-
   Posative terminal to Input
   Negative terminal to Output
4. I2C LCD Display to ESP8266-
   VCC to VV
   GND to GND
   SCL to D1
   SDA to D2

### 2.2 Explanation of IOT Layers –

**Perception-** MAX30100 sensors are used to sense the oxygen saturation level($SPO_2$) and Heart beat(BPM).Then , these data are transferred to ESP8266 by using serial communication . These values are visible on the LED display of the device.

**Communication-** ESP8266 is sent the data of heart beat rate and oxygen saturation over communication portal of blynk cloud using wifi module. These values will be displayed on blynk website which can be accessed by using IP address. Then, these values can be seen on the phone using blynk App.

**Application-** The data is sent over blynk cloud is controlled through a threshold condition that if the bpm rate is between 60-120 it is set to be the normal heart beat rate range. If the bpm rate is not in between the range a notification is delivered containing message " critical condition,take help" .

## 3. Simulation:

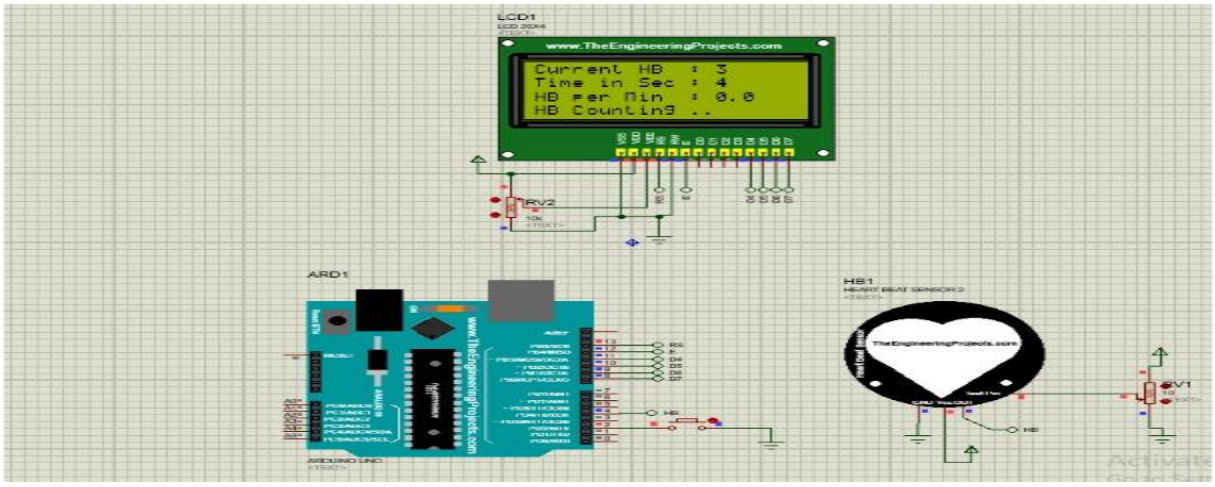


Figure 3 : Simulation of health monitoring system in Proteus software.

In proteus simulation here LCD display, heart beat sensor and aurduino Uno are taken for proceeding the simulation.In the simulation part, as ESP8266 module was not available in proteus software, the simulation has been conducted by using Aurduino uno.
By clicking the HB button it will start counting the HB as well as will count the Time in seconds. After ten seconds it will multiply the current heart rate with six and will give the Heart Beat Per Minute. The heart beat rate could not be send to the server due to not using ESP8266.Because integrated support for wifi network is in built in ESP8266 module which is not present in aurduino uno.

## 4. Code Explanation :

```
//Blynk Files
#define BLYNK_TEMPLATE_ID "TMPL_YZkiuXh"
#define BLYNK_DEVICE_NAME "Patients HeartBeat and SPO2"
#define BLYNK_AUTH_TOKEN "Xt9-gNZi9kyzMe-98_VY9Vv73Zy75dGX"
```

The following portion of the code is used to connect and communicate with the Blynk. AuthToken is a unique identifier generated by Blynk cloud. Device name can be set by the wiser's will.

```
//LCD
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);

//Headers
#include <Wire.h>
#include "MAX30100_PulseOximeter.h"
#define BLYNK_PRINT Serial
#include <Blynk.h>
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include "Wire.h"
```

This portion of the code is mainly about header files of different library. Wire.h and MAX30100_PulseOximeter.h will be required for the MAX30100 sensor's functionality.LiquidCrystal_I2C is used set the LCD address to 0x27 for a 16 chars and 2 line display.

```
//Wifi Data
char auth[] = "Xt9-gNZi9kyzMe-98_VY9Vv73Zy75dGX";
char ssid[] = "IOT CUET";
char pass[] = "C1234567";
```

This portion of code is used for wifi credential. This line helps to work with a WiFi-enabled device. These variables are used to set the password and ssid of personal wifi network.

```
PulseOximeter pox;
int BPM, SpO2;
uint32_t tsLastReport = 0;


//Times
#define REPORTING_PERIOD_MS 2000
```

In this portion, first line of the code indicates creating an object 'pox' of PulseOximeter. Two variables (BPM and SpO2) of type int is declared to hold the heart rate and SPO2 readings respectively. The variable (tsLastReport) holds the time when the last beat occurred. The REPORTING_PERIOD_MS defines the reporting time in milliseconds between the samples.

```
//Checking Conditions
#define check_status 5000
uint32_t tsLastCheck = 0;
int check_period=5;

//Alert Variables
int count=0;
#define bpm_low 60
#define bpm_high 100
#define spo2_low 92
```

Check_status is used to check the value for alert of patient condition in each five second.Initially tsLastCheck and check_period is declared 0 & 5 repectively.For the alerting condition bpm_low,bpm_high & spo2_low are set to 60,100 and 92 respectively.

```
void setup()
{
    Serial.begin(115200);
    Blynk.begin(auth, ssid, pass);

     lcd.init();
     lcd.setBacklight((uint8_t)1);
```

Inside the setup() function, the serial communication is opened at a baud rate of 115200.Blynk.begin() function is used to configure the blynk cloud by using ssid, authentication and the password.lcd.init() function is used to initialize the LCD display.setBacklight function is basically used to turn on the back light of LCD display.

```
void loop()
{
    pox.update();//Max30100 Updates
    Blynk.run();

    bpm_spo2();// Call Max30100 For data

    //Check Status After 5 second
     if (millis() - tsLastCheck > check_status)
    {
    alert();
    tsLastCheck = millis();
    }

}
```

Inside the loop() function, it will be read from the sensor using pox.update().Blynk. run() is a main Blynk routine responsible for keeping connection alive, sending data, receiving data, etc. Then bpm_spo2() function is used to call the sensor for data.By using the following condition,it checks the status of alert condition in every five second.

```
void bpm_spo2()
{
            BPM = pox.getHeartRate();
            SpO2 = pox.getSpO2();
    if (millis() - tsLastReport > REPORTING_PERIOD_MS)
    {
        Serial.print("Heart rate:");
        Serial.print(BPM);
        lcd.setCursor(0,0);
        lcd.print("BPM: ");
        lcd.print(BPM);
        lcd.print("    ");
        Serial.print(" SpO2:");
        Serial.print(SpO2);
        Serial.println(" %");

        lcd.setCursor(0,1);
        lcd.print("SPO2: ");
        lcd.print(SpO2);
        lcd.print("%");

        Blynk.virtualWrite(V5, BPM);
        Blynk.virtualWrite(V6, SpO2);

        tsLastReport = millis();
    }
}
```

Inside the bpm_spo2() function, the readings for the heart rate (saved in BPM) and the blood oxygen concentration (saved in SpO2) is obtained and print them in the serial monitor after two second. To obtain the heart rate reading, use the PulseOximeter object on the getHeartRate() method. Similarly, to obtain the blood oxygen concentration, use the PulseOximeter object on the getSpO2() method. The mills() function returns the number of milliseconds passed since running the current program. The esp8266 is used to send data to the Blynk cloud of V5,V6 pin using Blynk.virtualWrite(pin, value) function. Then tsLastReport is updated to the every last value of millis().

```
void alert()
{
  String body ="None";//Email Body
  String Message1=String("Alert..Your Patients is in Critical Condition!!! Here is Details: ");
  String Message2=String("\n\n\nNormal Values are: ");
  String Message3=String("\nBPM: ")+bpm_low+" To "+ bpm_high;
  String Message4=String("\nSPO2 Above: ")+spo2_low+"%";
  String Message6=String("\n\nTake Action Please...Check your App for more information.");


if(BPM<bpm_low && BPM!=0||BPM>bpm_high||SpO2<spo2_low && SpO2!=0)
{
     count=count+1;
if(count==check_period){

//Value to Send email
String BPM_Level=String("\nBPM: ")+BPM;
String SPO2_Level=String("\nSPO2: ")+SpO2+"%";

body = Message1 + BPM_Level +SPO2_Level+Message2+Message3+Message4+Message6;


    count=0;
}
}
```

Inside the alert function, these following message variables are basically used to show notification on the Blynk app. The data is sent over blynk if bpm and spo2 cross the range. Then a notification is delivered containing message " critical condition, take action" .

```
    if(body.equals("None")){Serial.println(body);}

    else
    {
      Serial.println(body);
      Blynk.logEvent("patients_condition",body);
      body ="None";
      lcd.clear();
      lcd.print("Please Relax!");

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

In the following portion, alert message will deliver to event called 'patients_condition' of Blynk server if the body condition is not satisfied.

## Result and Analysis

The system created for project is shown in this section, along with the results obtained by the system. The completed system consists of Max30100 pulse oximeter sensor that is connected to an ESP8266. It is connected to a device with the help of a USB, which will help power up the system. When we upload data to the Esp8266 from Arduino IDE , the system starts working, and the measurement data will be shown in the Liquid Crystal Display (LCD) display. When these sensor values are then sent to the Blynk server, these data can be accessed from the cloud by the authorized users using the IoT application platform. The sensor values of the patient are displayed on the app as shown in figure 3.

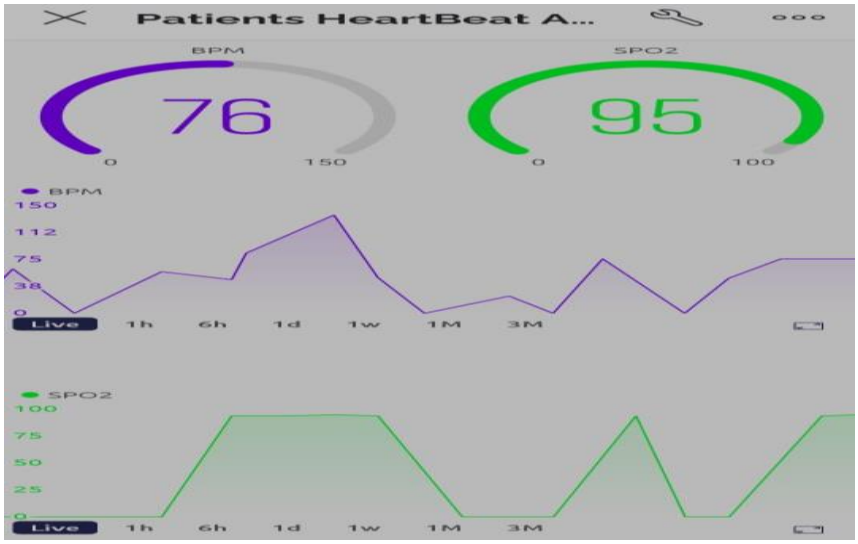


Figure 4: The collected value and graphical representation of BPM and SPO2 in Blynk android app

The case study is done by taking the readings from IOT system and then taking the readings of different person that are shown in the following table:

Table 5: Reading of Spo2 and Bpm of different persons.

| Person | Spo2(%) | BPM | Showing alert |
|---|---|---|---|
| Person1 | 94 | 51 | Yes |
| Person2 | 94 | 37 | Yes |
| Person3 | 95 | 76 | No |
| Person4 | 93 | 52 | Yes |
| Person5 | 95 | 45 | Yes |

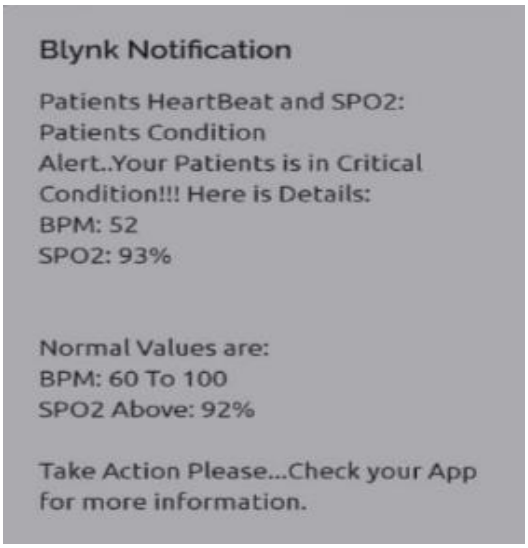


Figure 5 :Mobile app notification

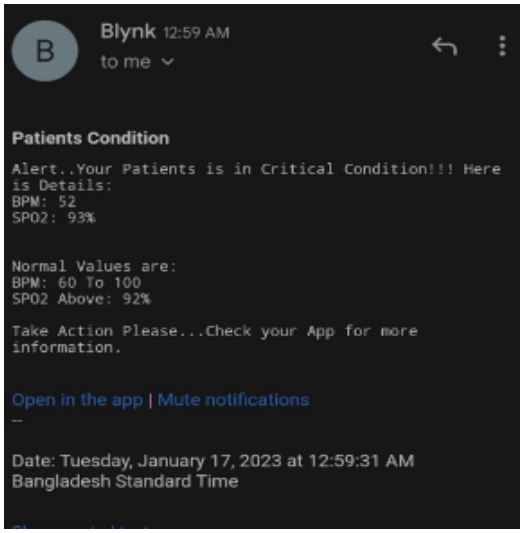


Figure 6: Mail notification

## CONCLUSION & FUTURE WORK

In this paper the health monitoring system was researched, designed and presented the concept of the Internet of things. In this project, the Internet of Things is used to build and deploy a health monitoring system (IoT). The system is created interfacing a MAX30100 pulse oximeter with a NodeMCU ESP8266 to measure SpO2 (the percentage of oxygen in the blood) and heart rate in BPM (Beats Per Minute).To track the patient's health, a ESP8266 is connected to an LCD display and a wireless network to transmit the information to a web server (wireless sensing node).These sensor values are then sent to via BlynkApp.The readings are collected in a simple cloud and can be viewed remotely by an app user or Healthcare giver.

Heartbeat sensor was found defective, which was a major complexity for this project. Defective sensor was showing fluctuating data. For this reason, bpm and oxygen saturation data cannot be displayed accurate for every cases. Power supply was a challenge for this project. 9v battery was used for LCD and power via USB cable is used for ESP8266.voltage regulator was used for 9v battery. For using these components the system looks somewhat bulky.

The system is getting only two parameters as input. Some more measures which are very significant to determine a patient’s condition like the level of diabetes, respiration monitoring, etc. can be addressed as future work .This system can be made more compact and user friendly.