National Textile University,

Faisalabad

Department of Computer Science

Name:	Noor Ul Huda Aqdas Fatima Areeba Mohsin
Class:	BSCS-B (5th Semester)
Registration No:	23-NTU-CS-1017 23-NTU-CS-1084 23-NTU-CS-1138
Course Code:	CSE-3080
Course Name:	Embedded IoT Systems
Submitted To:	Nasir Mahmood
Submission Date:	04 Jan, 2026

Table of Contents

PARAHEALTH – Patient Health Monitoring System

Overview:

PARAHEALTH is a small health monitoring system made using ESP32 and different sensors. The main idea of this project is to monitor a patient's health continuously and show the readings on a mobile phone using the Blynk application.

This system measures important health parameters like heart rate, oxygen level in blood (SpO_2), body temperature, and also checks room temperature and humidity.
If any value becomes dangerous, the system gives an alarm using a buzzer and also sends a message to the mobile app.

1.Problem Statement and Objectives:

Problem Statement:

In many homes and small healthcare setups, patient health is still monitored manually. This becomes difficult when continuous observation is required, especially for elderly or weak patients. Manual checking can cause delays, human errors, and sometimes critical health conditions are not noticed on time.

There is a need for a system that can:

- Monitor patient health continuously
- Alert immediately when something goes wrong
- Allow remote monitoring using a mobile phone

Objectives of the Project:

The main objectives of the PARAHEALTH project are:

- To measure heart rate and blood oxygen level using sensors
- To monitor body temperature, room temperature, and humidity
- To display all health data on a mobile application
- To provide an alarm system for abnormal health conditions
- To make a low-cost and easy-to-use health monitoring system

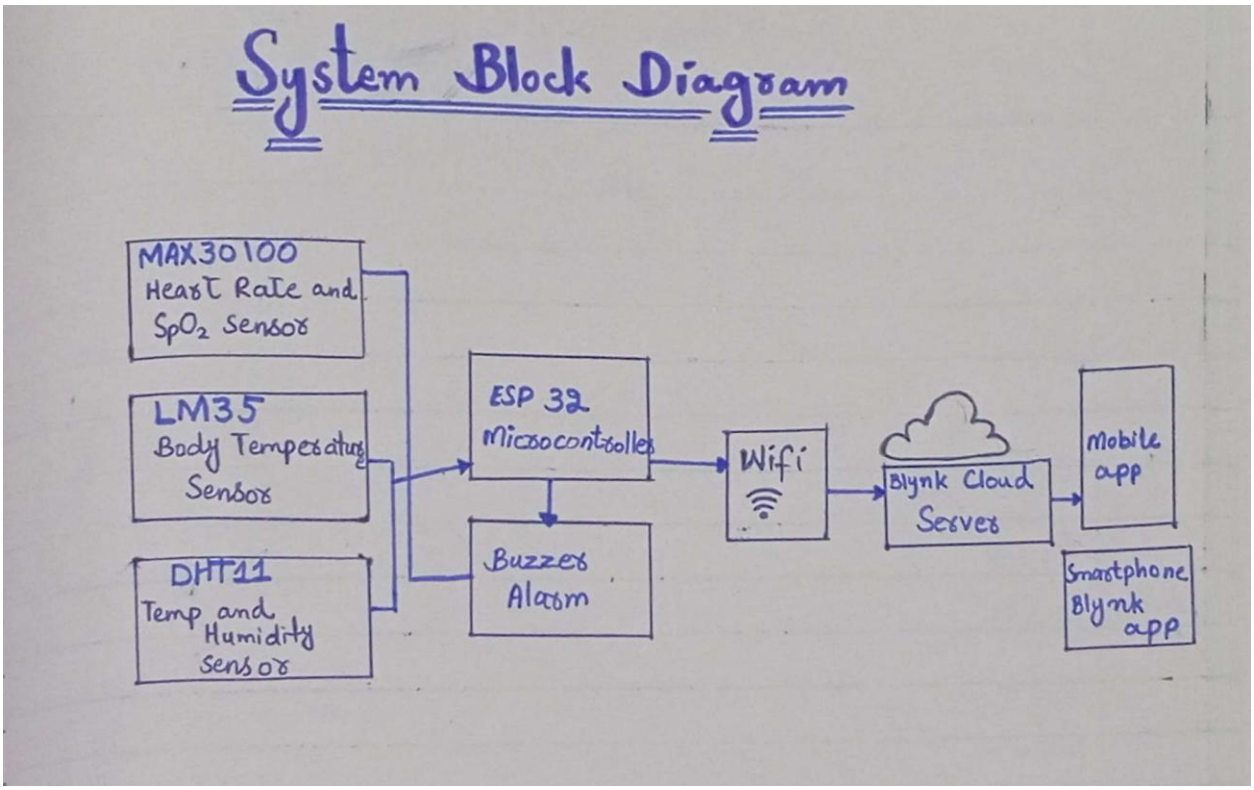
2. System Architecture / Block Diagram:

System Architecture Description:

The system is centered around the ESP32 microcontroller, which acts as the brain of the project.

- Sensors collect health and environmental data
- ESP32 processes the sensor readings
- Data is sent to the Blynk cloud through Wifi
- User can see data on the Blynk mobile app
- Buzzer gives alert during emergency conditions

Block Diagram :



3. Components Used:

Hardware Components:

- ESP32 microcontroller
- MAX30100 pulse oximeter sensor
- LM35 temperature sensor
- DHT11 temperature and humidity sensor
- Buzzer

- WiFi connection
- Mobile phone with Blynk app

Software Used:

- Arduino IDE
- Blynk IoT platform
- Required Arduino libraries for sensors and WiFi

4. Hardware and Software Description:

Hardware Description:

ESP32:
ESP32 is used as the main controller because it has built-in WiFi and good processing power.

MAX30100 Sensor:
This sensor is used to measure:

- Heart Rate (BPM)
- Blood Oxygen Level (SpO_2)

LM35 Sensor:
LM35 measures the body temperature of the patient in Celsius.

DHT11 Sensor:
It measures:

- Room temperature
- Humidity level

Buzzer:
The buzzer is used to alert when health parameters cross safe limits.

Software Description:

Arduino IDE:
Used to write, compile, and upload code to ESP32.

Blynk Application:
Used to display sensor data on a mobile phone and receive alerts.

Libraries Used:

- WiFi library for internet connection

- Blynk library for cloud communication
- MAX30100 library for heart rate and SpO_2
- DHT library for temperature and humidity

5. Working:

First, the ESP32 connects to the WiFi network.
After connecting to WiFi, it connects to the Blynk cloud using the authentication token.

Once the connection is successful, all sensors start working:

- MAX30100 measures heart rate and oxygen level
- LM35 measures body temperature
- DHT11 measures room temperature and humidity

All readings are:

- Shown on the Serial Monitor
- Sent to the Blynk mobile application

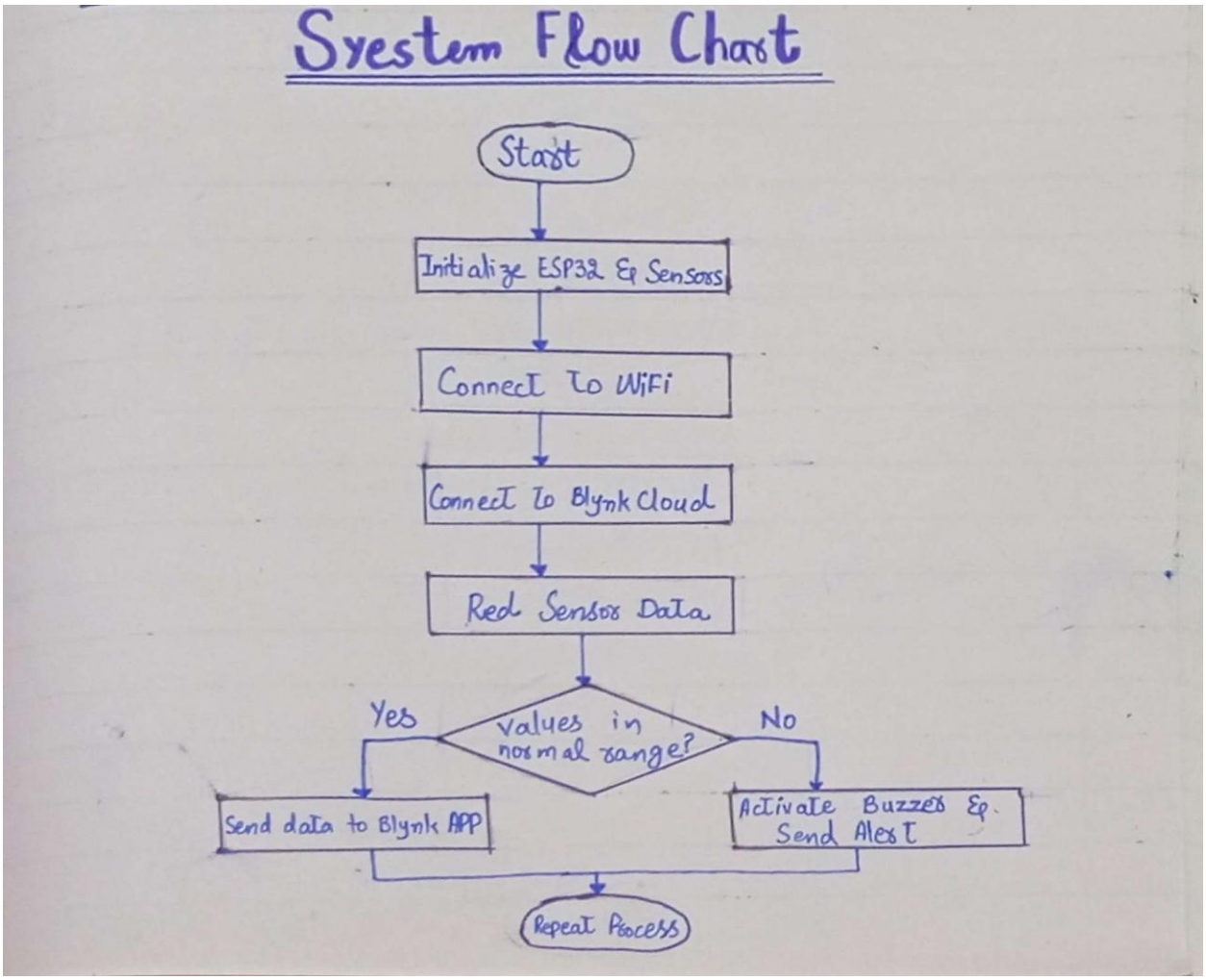
If any reading goes outside the safe range, the buzzer turns ON and an alert message is sent to the app.

6. Methodology and Flowchart:

Methodology:

1. Power ON the system
2. ESP32 connects to WiFi
3. ESP32 connects to Blynk cloud
4. Sensors start collecting data
5. ESP32 processes sensor values
6. Data is displayed on Blynk app
7. System checks for abnormal conditions
8. Buzzer turns ON if limits are crossed
9. User can reset alarm from app

Flowchart :



7. Code Explanation:

Blynk Setup:

The Blynk template ID, name, and authentication token are used to connect the ESP32 with the Blynk application. Without these values, data cannot be sent to the app.

WiFi Connection:

The WiFi name and password are defined in the code.
ESP32 tries to connect to the WiFi and waits until the connection is successful.

Pin Configuration:

Each sensor is connected to a specific pin of ESP32:

- MAX30100 uses I2C pins (SDA and SCL)
- DHT11 uses a digital pin

- LM35 uses an analog pin
- Buzzer uses a digital output pin

MAX30100 Sensor Task:

The MAX30100 sensor needs to be updated continuously, so a separate task is created using FreeRTOS.
This task runs independently and keeps updating heart rate and SpO_2 values without affecting other parts of the program.

Reading Sensors:

- DHT11 reads room temperature and humidity. If the reading fails, the value is ignored.
- LM35 reads body temperature by converting analog values into Celsius.
- MAX30100 gives heart rate and oxygen level.

Alarm Conditions:

Safe limits are defined in the code:

- Heart rate should be between normal range
- SpO_2 should not go below 90%
- Body temperature should stay within safe limits

If any value crosses these limits:

- The buzzer starts beeping
- A warning message appears on Blynk
- The alert is logged

Buzzer Operation:

The buzzer is used as a warning device. It turns ON when an abnormal condition is detected and turns OFF when readings return to normal or when manually reset.

Blynk App Controls:

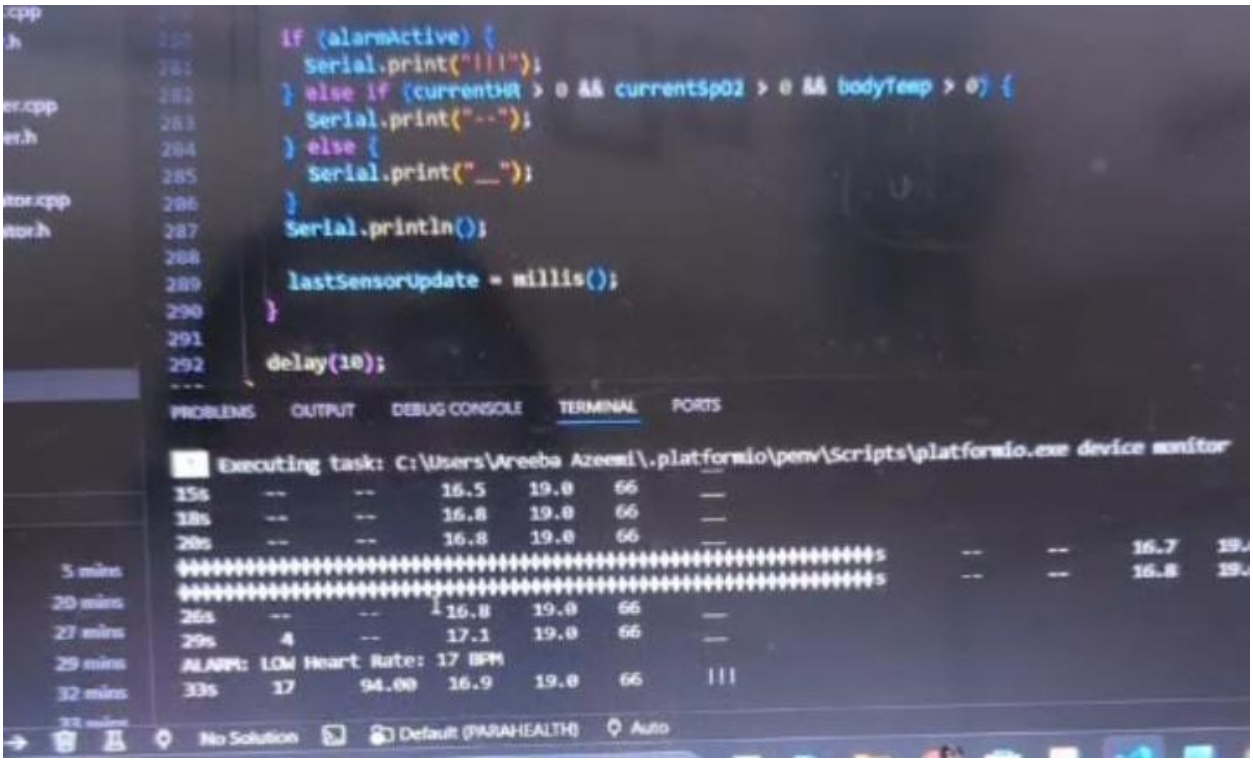
The Blynk app shows all sensor values using virtual pins.
It also has buttons to:

- Save readings manually
- Reset the alarm if needed

8. Screenshots of Output and Dashboards:

Serial Monitor Output:

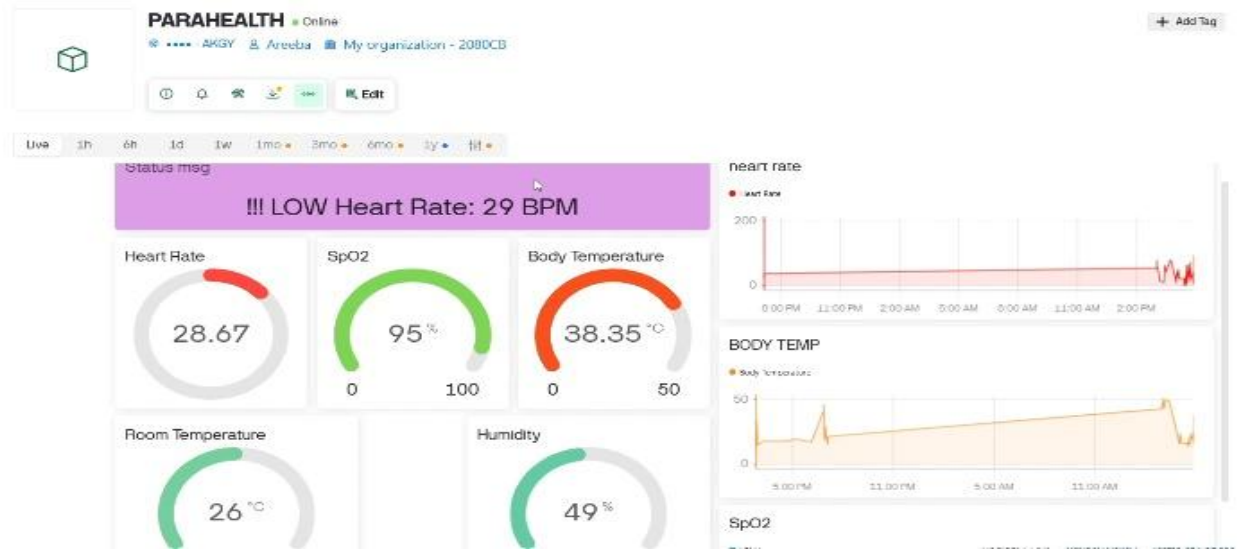
- Displays heart rate, SpO_2, body temperature, room temperature, and humidity
- Shows alarm status in real time
- Useful for debugging and monitoring



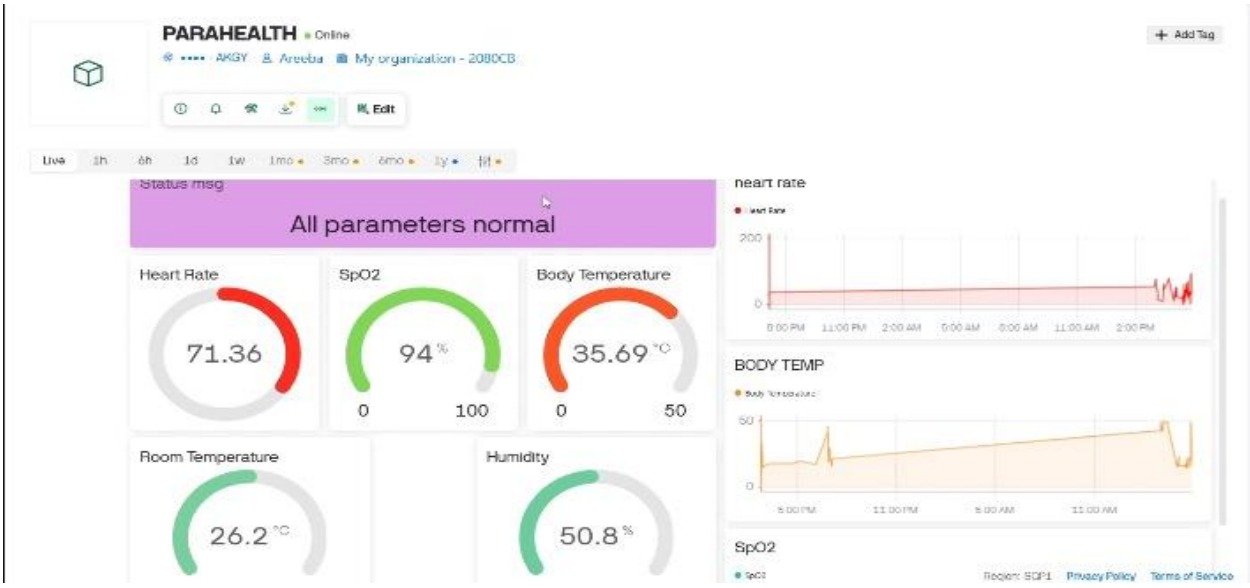
Blynk Dashboard:

The Blynk app shows:

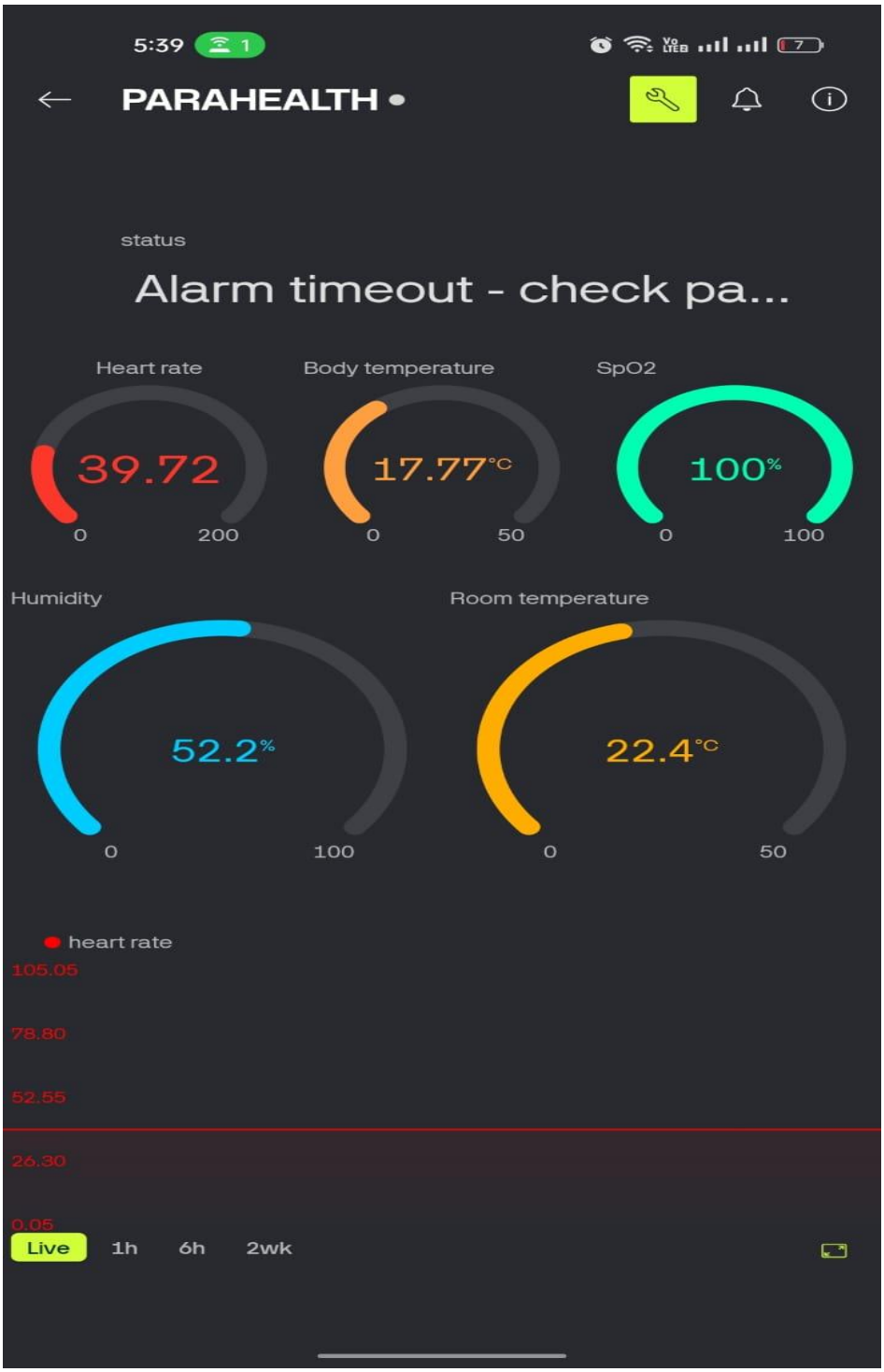
- Heart rate gauge
- SpO_2 percentage display
- Temperature readings
- Humidity level
- Alert messages
- Manual alarm reset button


PARAHEALTH
Online
AKGY
Areeba
My organization - 208DCB
Add Tag
Edit
Status msg
!!! LOW Heart Rate: 29 BPM
Heart Rate
28.67
SpO2
95 %
0
100
Body Temperature
38.35 °C
0
50
Room Temperature
26 °C
Humidity
49 %
heart rate
BODY TEMP
SpO2


PARAHEALTH
Online
AKGY
Areeba
My organization - 208DCB
Add Tag
Edit
Status msg
!!! LOW Temperature: 25.6°C
Heart Rate
53.88
SpO2
94 %
0
100
Body Temperature
25.59 °C
0
50
Room Temperature
26.3 °C
Humidity
46.1 %
heart rate
BODY TEMP
SpO2


PARAHEALTH
Online
AKGY
Areeba
My organization - 208DCB
Add Tag
Edit
Status msg
All parameters normal
Heart Rate
71.36
SpO2
94 %
0
100
Body Temperature
35.69 °C
0
50
Room Temperature
26.2 °C
Humidity
50.8 %
heart rate
BODY TEMP
SpO2
Region: SCP1
Privacy Policy
Terms of Service

On Mobile Blynk APP:



9. Advantages of the System:

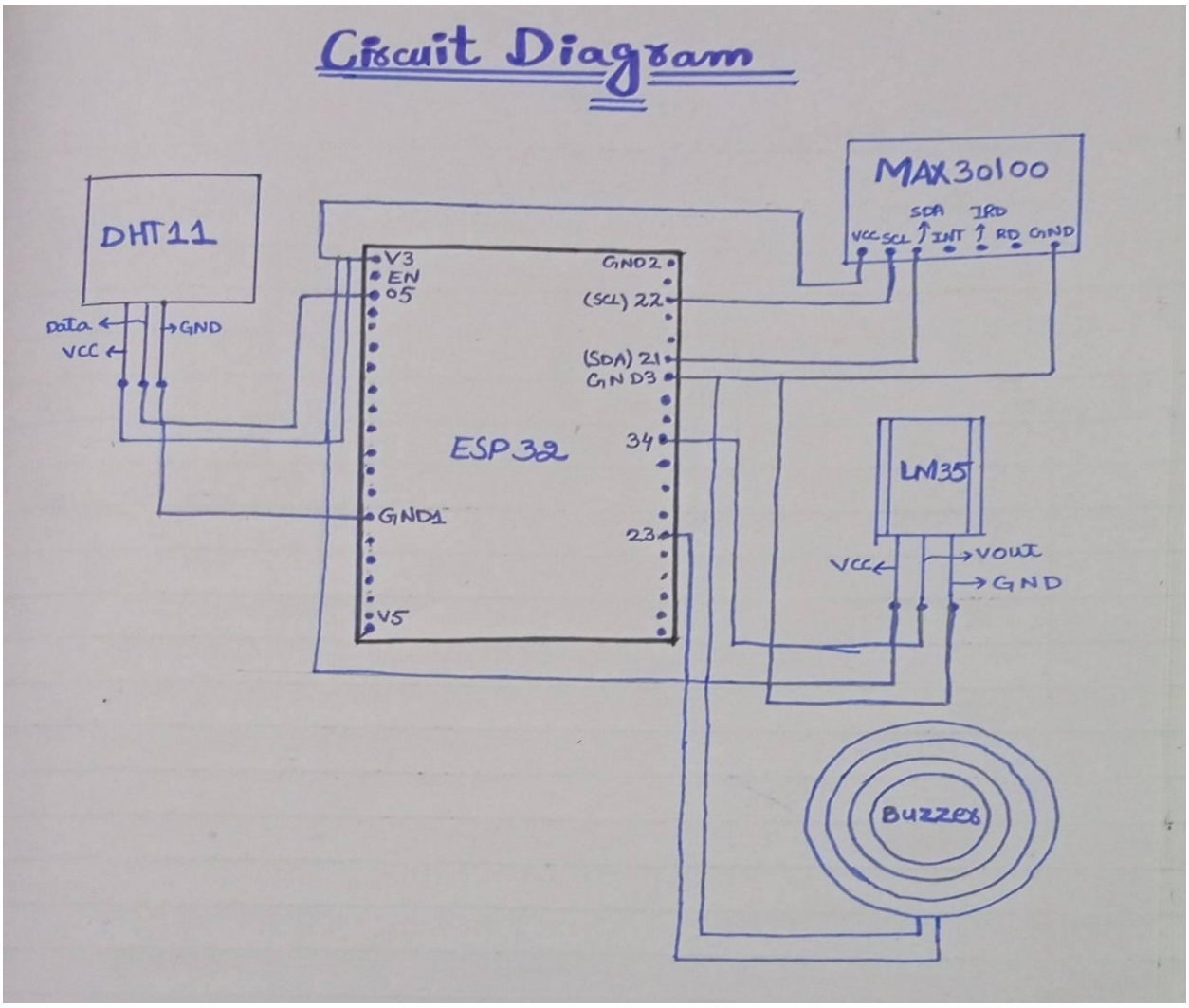
- Easy to use
- Low cost

- Real-time monitoring
- Remote access through mobile phone
- Immediate alert in emergency situations

10. Applications:

- Home patient monitoring
- Elderly care
- Small clinics
- Health monitoring projects
- IoT-based medical systems

11. Circuit Diagram:



12. Results:

The system worked successfully and provided accurate health readings.

- Heart rate and SpO_2 were updated continuously
- Temperature readings were stable
- Alerts were triggered correctly during abnormal conditions
- Blynk app displayed data in real time
- Alarm reset feature worked properly

13. Conclusion:

The PARAHEALTH project successfully demonstrates a real-time patient health monitoring system using IoT. It reduces the need for manual checking and provides instant alerts in case of emergencies. The system is reliable, low-cost, and easy to use, making it suitable for home and small healthcare environments.

14. Future Scope:

The project can be improved further by:

- Adding ECG sensor for more accurate monitoring
- Storing patient data in a database for history tracking
- Adding GPS for patient location tracking
- Using a mobile notification system instead of only buzzer
- Implementing AI for health prediction