

Integration Framework Solution for HealthCare Monitoring System

Proposed System & Architecture

The proposed system for the work used is shown in Figure 1. The system uses an ESP-32 processor to process all the functionalities. it is used with Heart rate, pulse oximeter sensor, temperature sensor, and ECEG Sensor to monitor various vital parameters for the patient. The Wi-Fi module connects this system remotely using mobile-based devices. The LCD is used to view the results of the system. The HTML is the server used to connect the processor with mobile devices.

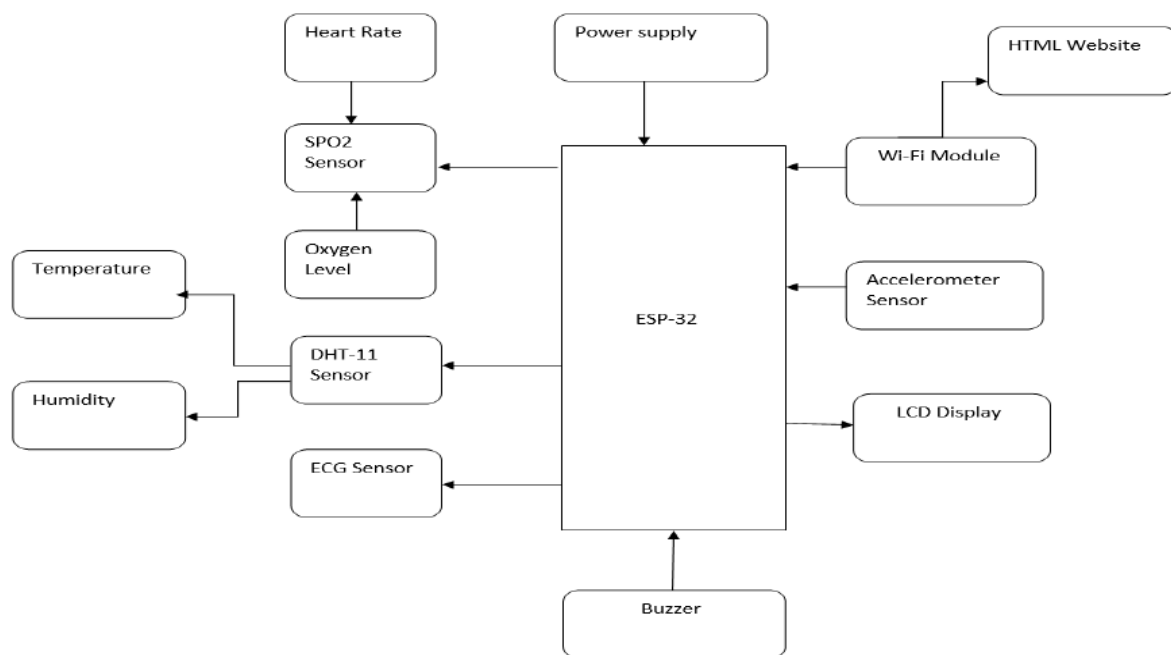


Figure 1. General view of healthcare monitoring

- The prototype used for the architecture is shown in Figure 2.
- A native Visual Studio Application that works on both windows as well as a smartphone that displays the patient's data with an emergency notification alert.
- The caretaker in charge gets an emergency notification in case any fluctuations are found in the patient's body.
- The discovery board has an LCD that displays a log of the patient's records kept in the microcontroller. The display is a touchscreen that allows us to travel through the patient's medical history.
- The data is pushed to the cloud, where it may be retrieved by the PC app and saved in a local database for display on the app. For the convenience of doctors, the app also allows them to update or delete a patient's history.

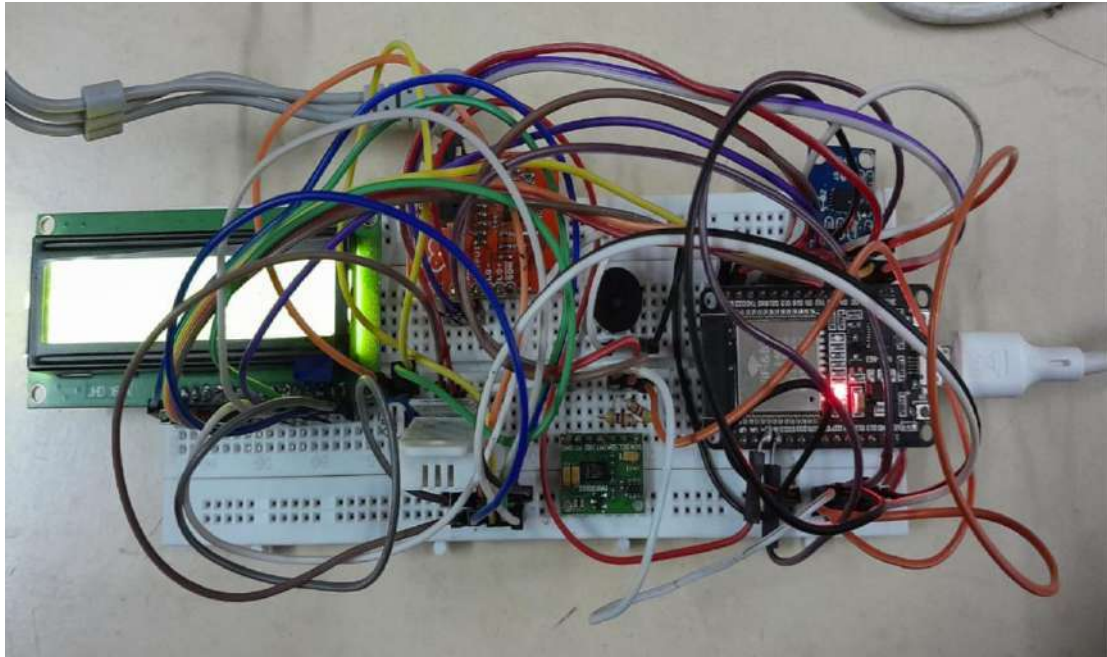


Figure 2. Prototype developed for interoperable health monitoring system

If the patient's pulse is low is life-threatening and requires the doctor's attention immediately. In that kind of scenario, the health monitoring system will need a common interface for that we are furnishing an interoperability solution. We used ESP-32 as the control board. To utilize ESP-32 as a server, we'll install XAMP. The oximeter will be attached to the pulse detection sensor, which will detect the pulses. Wi-Fi or a wired network can be used to connect to it. An LED display, with the help of detectors, will display the heart rate and breathing rate of the patient. We will set a threshold value, if the patient's heart rate is below 60 for the next three iterations, an alarm beep will sound, and an SMS and an email notification will be sent to the patient's family. In an LED display, there would be two lights red light and green light. Red light symbolizes the patient is critical and needs attention. Also, the green light symbolizes patient is healthy.

The system is integrated with the following three modules

- (a). Hardware integration to the controller.
- (b). XAMP server integration.
- (c). Server integration to hardware.

The circuit diagram for interfacing MAX30102 and DHT-11 with ESP-32 is given below in Figure 3.

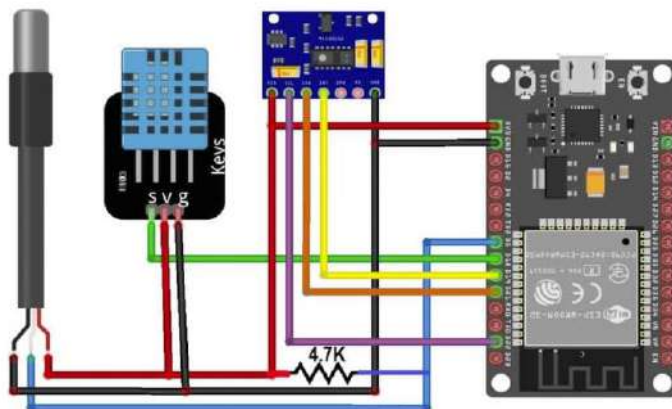


Figure 3. The system's schematic diagram of ESP-32 controller connecting the sensors

All of the sensors can work at 3.3V VCC. As a result, double-check that their VCC is linked to a 3.3V power source. GND and GND should be linked. Connect the MAX30100's SDA and SCL pins to GPIO21 and GPIO22, respectively, because it's an I2C sensor. Connect its INT pin to GPIO19 on the ESP32. The output pin of the DHT11 is connected to GPIO18 on the ESP32. Similarly, the output pin of the DS18B20 is connected to GPIO5 on the ESP32. A 4.7K pull-up resistor is located between the output pin and the VCC pin of the DS18B20.

If the patient is in a critical situation an SMS message will be sent to the family members to address the doctor immediately. The pulse sensor can be placed on a finger or earlobe, via a connection wire that can be connected to the ESP-32, to check the heart rate. To detect the presence of liquid, the pulse sensor employs a magnet and a reed switch. The reed switch is operated by a magnetic field and has its connections sealed in a glass tube containing inert gas. The pulse sensor's supply voltage is 3.3v -5v. The list of components required for the system used is shown in Table 1.

Table 1. The components required to develop the health monitoring system on the ESP board are as follows.

| Sl. No | Components Name | Description | Quantity |
|--------|-----------------------|---|----------|
| 1 | ESP-32 Board | Esp-32 Development Board. | 1 |
| 2 | DHT-11 Sensor | DHT-11 Digital humidity temperature sensor. | 1 |
| 3 | Pulse Oximeter Sensor | MAX30102 Pulse Oximeter Sensor | 1 |
| 4 | SPO2 Sensor | AD8232 Heart Monitor | 1 |
| 5 | ECG Sensor | AD8232 ECG module | 1 |
| 6 | Accelerometer Sensor | HW-013 Accelerometer sensor | 1 |
| 7 | Buzzer | - | 1 |
| 8 | Connecting Wires | Male to Male Jumper Wires | 15 |
| 9 | Bread Board | Solderless Breadboard Mini | 1 |
| 10 | Resistor | 4.7K | 1 |

Result and Discussion

Data gathering, data monitoring, and alerting for the user to communicate are the three categories of health monitoring systems.

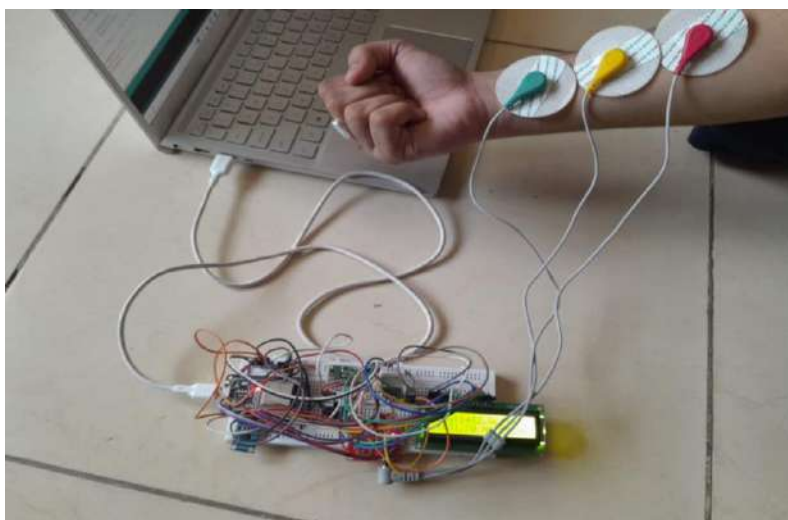


Figure 4. Patient utilizing the IOT device and given the test results instantly

- **Biometric Data Collection:** Various sensors affixed to the patient's body, as illustrated in Figures 8 & 9, can be used to gather data. A microcontroller platform can capture the data, which can then be mapped into the patient's records. Multiple sensors can be mapped to the patient's database initially. The patient other vital parameters of humidity and body temperature are verified by connecting DHT 11 sensor as shown in Figure 4. The results are taken by touching the sensor on the patients
- **Data processing:** A patient's bodily parameters can be tracked and sent to the cloud regularly. You can take advantage of a sensor platform.
- **User Interface:** The user interface's objective is to identify the parameters of the patient's body and alert the caregiver in the event of an emergency. A mobile app or a PC program with a navigational display is frequently used.

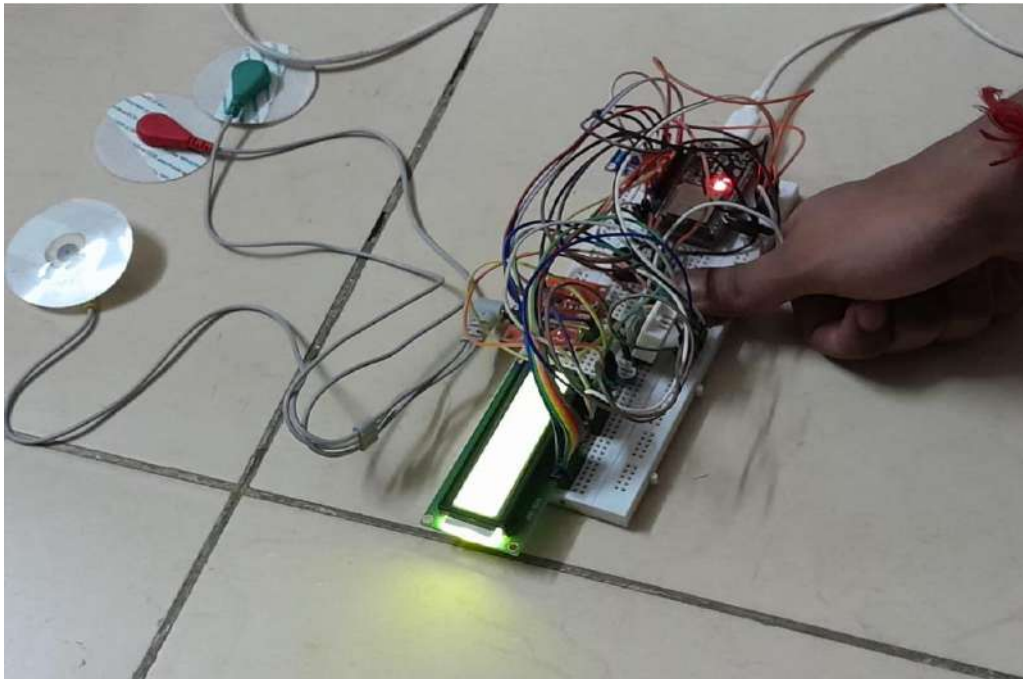


Figure 5. A patient inspecting body temperature

The outcomes of the system, as well as the system's results, are shown in the following section. Pulse rate, room temperature, ECG, humidity, and an accelerometer are all connected to the ESP-32 as part of the whole system. The ESP device is connected to the system through a USB connection, which allows us to power it up.



Figure 6. LCD display screening the values of ECG, accelerometer, Temperature in degree Celsius, and Heart rate

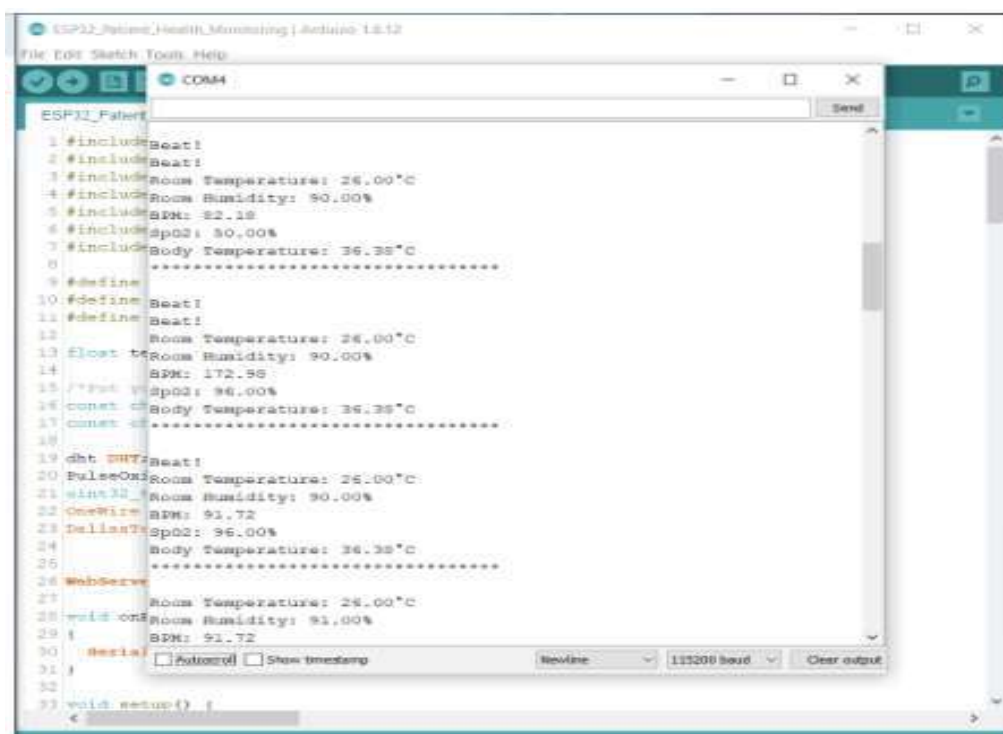


Figure 7. Diagram depicting the measurement on the Arduino IDE's serial monitor

When we upload data to the Arduino, the system begins to operate, and the measurement data is presented in the serial monitor of the Arduino Integrated Development Environment (IDE) and the Liquid Crystal Display (LCD) display, as shown in Figure 6, as well as on a mobile app through WiFi. Figure 7 illustrates the full system, including body temperature and ECG measurements shown in the Arduino IDE's serial monitor and the mobile app. Figure 7 shows the results of the measurements of all the parameters in the serial monitor of the Arduino IDE. The

MAX30102 and DHT-11 sensors provide the data value.

Once the code is uploaded to your respective board, you may view the program in action using the serial monitor when the code has been uploaded to your board. The Wi-Fi network will be activated by the Node MCU ESP32. It will show the ESP IP Address after it is connected. Use the ESP32 IP Address in the Web Browser.

The system's Wi-Fi may be switched on or off using the choices in the mobile app, as shown in Figure. For ECG, SpO2, and temperature measurement, a default symbol "--" meaning nil will appear on the screen before setting the Wi-Fi "On." When Wi-Fi is switched on, the figure illustrates a diagram of measurements on the mobile app. When we switch the Wi-Fi "on" in Figure, the data will be transferred into the IoT system we built via the Wi-Fi device. The acquired data will be accessible on the mobile application when it has been collected. Depending on the available data connection, it may take a few moments. On the mobile application, the measured heartbeat, Spo2, and temperature are shown in Figure 8.



Figure 8. Temperature in degrees Celsius, Fahrenheit, Humidity, and ECG are all shown on this health monitoring system.

In the figure, we observed an individual measure of six different patients in real-time their temperature, ECG, accelerometer, and humidity. Their current data is gathered and shown on the mobile app, as illustrated in Table 2. The data indicate that patient 1 is in good health and poses no threat. As a result, they aren't required to seek medical help.

Table 2: The suggested system's body temperature test results for three separate persons

| Body Temperature | Actual Data[deg F] | Observed Data[deg] | Error[%] |
|------------------|--------------------|--------------------|----------|
| Patient-1 | 97.4 | 97.9 | 0.52 |
| Patient-2 | 98.5 | 97.6 | 0.72 |
| Patient-3 | 98.2 | 98.9 | 0.51 |
| Patient-4 | 96.8 | 97.4 | 0.63 |
| Patient-5 | 97.6 | 97.2 | 0.52 |
| Patient-6 | 98.3 | 97.4 | 0.82 |

Patient 2's report is depicted in Table 2. Patient 2's health data was determined using a technique identical to that of Patient 1. Conceivably the report of patient 2 indicates that the body temperature is extremely high. The person may be suffering from a high temperature. As a result, this person may keep a close eye on his health and seek medical help in the days ahead. They can track their health in real-time and receive medical help. The system would share notifications to the doctors with the data if their cardiac rate gets high or low during testing.

We looked at the health data for patient 3 in Figure to see if it followed the same pattern. "Patient 3" has a quicker heartbeat. Over the following four days, they'll be able to track their health data and diagnose themselves with medical assistance.

In the table, you'll find the test results as well as an analysis of all the patient data. Two human test participants were employed in the study to test the following system. The findings of the measurements were utilized in the development of our newly manufactured technology. Real-world scenarios were used to test and implement the system. The real-world test results acquired with the systems are listed in the following tables in the order that they were obtained. The findings of three people's body temperature readings are shown in Table 3. The temperature findings are pretty close for each measurement setting of a test subject. The data for each test, however, differ significantly due to the variances in human bodies.

Table 3. ECG test results of three different people of the proposed system

| ECG | Patient-1 | Patient-2 | Patient-3 |
|--------|-----------|-----------|-----------|
| Take-1 | 96 | 92 | 98 |
| Take-2 | 96 | 92 | 98 |
| Take-3 | 96 | 92 | 98 |

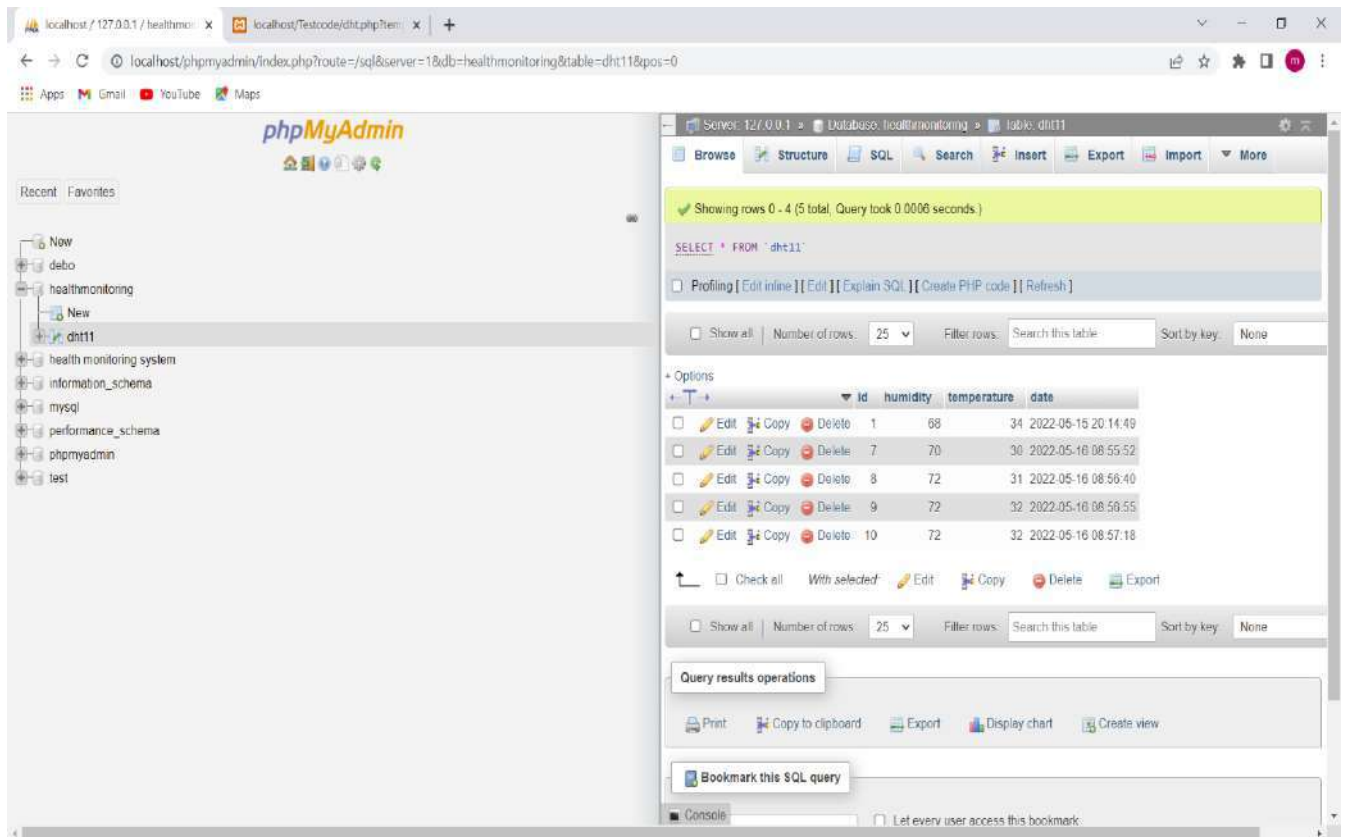


Figure 9. Health Monitoring System displaying temperature, and humidity through the XMPP server

XMPP is a widely used cross-platform web server that helps programmers to develop and test their applications on a native web server. It's an open-source web results package for a range of servers and command-line executables, as well as Apache, MariaDB, PHP, and Perl server modules, that comes with the Apache distribution.

XMPP allows the host or origin server to test their websites and visitors on PCs and laptops before publishing to the back-end server, as seen in Figure 13. It's a server-based platform for testing and verifying Apache, Perl, MySQL databases, and PHP-based systems. Perl is a web programming language, PHP is a backend scripting language, and MariaDB is a MySQL database server.

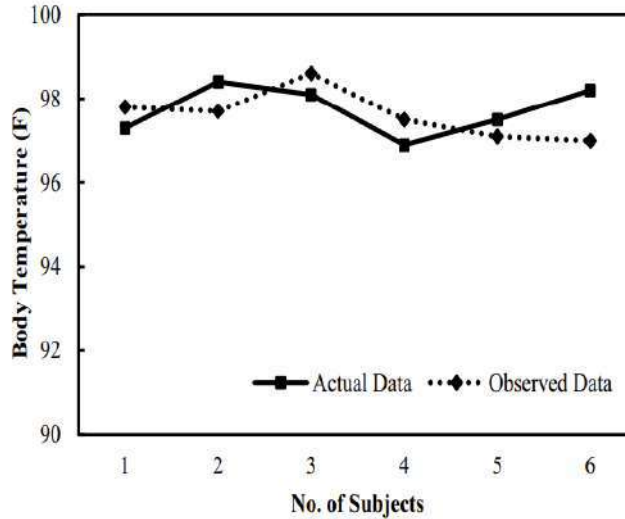


Figure 10. The test result of actual and calculated data for body temperature

To evaluate the proposed model developed the error accuracy metric is used to check its validity. The error percentage is calculated between two data namely, actual data of the patient and measured data from the system. the actual data for a different set of patients are taken without connecting to the system, and by using the integrated system same set of patients' measured data is noted. The error difference is calculated by using the formula shown in equation (1). This accuracy tells how much the measured data is nearer to the observed data. The same thing is plotted as a line chart graph and is shown in Figure 10.

$$\text{Error \%} = \frac{\text{Actual Data} - \text{Measured Data}}{\text{Measured Data}}$$

As the measured data closely matches the actual data, the integrated system can efficiently help in monitoring different vital parameters simultaneously and helps monitor the parameters remotely

Merits

- (a) Self-monitor and exchange critical physiological indicators in order to have a better understanding and control over their current health state.
- (b) Patients benefit from healthcare monitoring because it improves the quality and durability of care.
- (c) Real-time access to medical records, as well as their portability, may be ensured, and patients can receive high-quality medical care at any time.

Conclusion

The design and deployment of an Internet-of-Things-based health monitoring system are presented in this study. This Internet of Things (IoT)-based device allows users to identify their health indicators, which may help with long-term health management. If the necessity arises, the sufferers may eventually seek medical help. They might effortlessly communicate their health parameter data with

the doctor using a single application. The main purpose of healthcare interoperability is to provide a smooth flow of health-related data between caregivers and patients for clinical decision-making. The viability of employing the health monitoring system is demonstrated in this study. This platform may be used to integrate sophisticated analytics to improve the telemonitoring system's delicacy even further. The prototype can also be extended to include additional open and personal health rules. The system will transmit data regarding a patient's body temperature, heartbeat, and blood SpO2 levels to an app using Bluetooth. This information is also supplied to the patient's LCD, which allows them to check their current health state right away. Elderly patients, asthmatics, COPD patients, patients with chronic diseases, COVID-19 patients, and diabetic patients will be able to maintain their health over time with the help of the system we built.

Future work

The system may be updated and altered in the future in a variety of ways. The ESP-32 microcontroller in the system may be swapped out for a Raspberry Pi and used in several ways. The system's sensors may be updated when additional sensors are added, allowing us to track a wide range of health measures. For system security, new algorithms may be integrated with the entire system. For consumer safety and dependability, the healthcare industry's future depends more on health monitoring systems.