

Vitalguard: Patient Health Monitoring System

N. Lasya Priya¹, D. Vasantha², R. Rajitha³, S. Swarna⁴, T.Jhansi⁵, N. Ravi Kiran⁶

^{1,2,3,4,5}UG Scholar, ⁶Assistant Professor,

Department of Electrical & Electronics Engineering

St. Ann's College of Engineering & Technology, Chirala (Autonomous)

Affiliated to JNTUK, Kakinada, India

Email id : nattavalasyapriya@gmail.com¹, vasanthadamarla1710@gmail.com²,
ramalingamrajithareddy793@gmail.com³, swarnasavanam@gmail.com⁴,
jhansi6317@gmail.com⁵, ravikirannasaka@gmail.com⁶

Article Received: 15 Jan 2026

Article Accepted: 28 Feb 2026

Article Published: 10 March 2026

Citation

N. Lasya Priya, D. Vasantha, R. Rajitha, S. Swarna, T.Jhansi, N. Ravi Kiran,
“Vitalguard: Patient Health Monitoring System”, Journal of Next Generation Technology
(ISSN: 2583-021X), 6(2), pp. 1-16. March 2026.

Abstract

In this paper, we propose a wearable health sensor network system for Internet of Things (IoT) connected safety and health applications. Ensuring the safety and health of ICU patients is a priority in the hospital workplace; hence proposed an IoT network system which can monitor all the above-mentioned health parameters and update through wireless. The envisaged network system comprises of several wearable sensors to evaluate environment and physiological parameters. The wearable sensors in various subjects talk to one another and transfer the information onto a gateway through an IoT stage restorative sign detecting organize. In the proposed system, heart rate, temperature and vibration sensors are connected to processing microprocessor. Sensors measure health parameters and pass it to the ARDUINO module. This module only does analysis of the data and display in an LCD and will post same to IoT based server. It keeps on monitoring and if they finds any changes like high heart rate, high temperature or movement of the patient then IoT alerts the authenticated person about the health status of that patient. We implemented a smart IoT gateway that includes data processing, local web server, and cloud connection. Once the data is collected by the gateway from the wearable sensors, it will send that data to an IoT cloud where this data can be stored, processed and visualized. Health monitoring systemIoT based health monitoring system monitors the heart beat rate of patient as well as body temperature and alerts server if these goes more than critical values. They record the pulse rate and body temperature readings through some IoT platforms to can monitor patient health from any part of the world on the internet. Sensors are used to collect patient information The sensor data collected is processed by a microcontroller Arduino board and this processed data is transmitted via an ESP8266 IoT module to a remote server.

Index Terms—Internet of Things (IoT), Wearable Sensors, Health Monitoring, Arduino, ESP8266, Remote Patient Monitoring.

I. Introduction

Researchers across the healthcare landscape are increasingly taking advantage of areas affected by these developments and capable of great improvement in and out of medical administrations. Likewise Electronic Health applications and Health (treatment associations that are controlled by ICT) empower wide sections of ordinary buyers to develop, underpin and fortify their wellbeing network. The SMS is Submitted either to the specialist or to any family member in some special case [1]. Patient Monitoring Systems (PMS) [2] is a reliable and relatively faster implementation of this corresponding approach. A one of the biggest problem for society is social security. The World Health Organization (WHO) shows that the highest source of benefit in medical care is not the health system for an individual. The framework of social insurance will be introduced, a well applied simple repairing connections for individuals however possible, economically and cautiously. As long as these devices are web-friendly, they further enhance the ecosystem and secure that corporations and people become constantly protected and logically connected. All aspects of IoT live on sensors including remote systems that enable users to grant and reach the application/information. The effect of IoT is exemplified nowhere more than in its handling of areas for prosperity [4]. Technically speaking, the Internet of Things (IoT) has become widely noticed in different fields, especially personalized healthcare [4]. On the other hand, body area sensor network (BASN) under IoT framework is extensively utilized for ubiquitous health monitoring [5]. The key contributions of this paper include the following: first, we propose a new system called WISE (Wearable IoT-cloud-based health monitoring system) for real-time personal health monitoring. WISE supports real-time health monitoring using the BASN framework [7]. It has been placed in several wearable sensors such as heartbeat, body temperature and vibration sensors [1, 3].

A. Motivation

Patient monitoring systems can help doctors save time and effort, free up beds, and help patients recover at home [2, 8]. The system can also help warn of early or dangerous deterioration and improve patient self-care. Some other benefits of patient monitoring include: efficient communication, increased patient confidence, visualization of health trends, and greater patient education. The motivation for IoT-based patient healthcare monitoring lies in its ability to transform healthcare into a proactive, patient-centered system [5]. By combining real-time monitoring, remote accessibility, and data-driven insights, IoT devices not only improve patient outcomes but also make healthcare more affordable and inclusive [4]. This project highlights how engineering innovation directly impacts human well-being — a perfect blend of technology and social relevance.

B. Objective of the Thesis

The objectives are improving patient care, supporting decision-making among medical personnel, increasing affordability for regular people, making personal healthcare easier for

patients, and lowering the costs of designing a prototype [3]. The system aims to create a dependable patient management system based on IoT. This allows healthcare professionals to monitor their patients who are either hospitalized or at home [2, 8]. The goal of this project is to design and execute a distributed system for heart vitals quality care through remote observing of temperature, heart rate, and movement [1, 7]. This work will contribute a remote monitoring framework through IoT to screen health quality in the body. The system is portable, modular, low-cost, and versatile and permits sharing of data through the mobile application [5, 9].

II. Feasibility Study

The Feasibility Study is an evaluation and analysis of the potential of a proposed project. It is based on extensive investigation and research to support the process of decision making. In

simplest terms, the two criteria to judge feasibility are cost required and value to be attained. The feasibility of a project on patient healthcare monitoring using IoT devices is strong across technical, economic, operational, and ethical dimensions.

A. Technical Feasibility

Technical feasibility measures the current resources (hardware/software) available in our technology that can be used to complete user requirements through the software within a certain time and costs. It also does the following under Technical feasibility:

- Evaluates the technical skills and competencies of the members of the soft-ware development team.
- Ensures the technology selected for software development has a wide user base in case required assistance arises when technologies need to be debugged or enhanced.

On the technical side, the project is doable since IoT sensors, wearables and cloud platforms are available to collect continuous real time health data already.

B. Operational Feasibility

Operational feasibility is best characterized by how well the available software accomplishes a set of steps to solve the needs of users. This feasibility rests on human resources (software development team) and is very much to visualize whether the software would function once developed and be executable after installation. The devices have a low learning curve since they are usually uncomplicated for both the patient and healthcare staff, with few maintenance issues.

C. Economic Feasibility

Economic feasibility determines whether the required soft-ware is capable of generating financial gains for an organization. The economic feasibility of patient healthcare monitoring using IoT devices is quite promising because the costs of sensors, wearables, and communication modules have been steadily decreasing, making them affordable for large-scale deployment. By enabling continuous monitoring and early detection of health issues, the

system reduces the need for frequent hospital visits and lowers the chances of costly emergency interventions, which translates into significant savings for both patients and healthcare providers. Overall, the project is economically viable as it balances initial investment with long-term savings and improved healthcare efficiency.

D. Schedule Feasibility

A project will fail if it takes too long to be completed before it is useful. The scheduled feasibility of patient healthcare monitoring using IoT devices refers to whether the project can be realistically completed within a defined timeframe, considering the resources and tasks involved. This project is highly feasible from a scheduling perspective because the required components such as IoT sensors, wearable devices, cloud platforms, and mobile applications are already available and can be integrated without the need for extensive new research or development. The implementation can be broken down into clear phases: requirement analysis, hardware and software selection, system design, prototype development, testing, and deployment.

III. System Design

System Design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. A well-structured system design for patient healthcare monitoring using IoT devices typically follows a layered architecture to ensure smooth data flow, reliability, and security. At the device layer, wearable sensors and medical IoT devices (such as heart rate monitors, temperature sensors, or pulse oximeters) continuously collect patient health data. This data is transmitted through the network layer using communication technologies like Wi-Fi or Bluetooth. The data processing layer involves cloud platforms or edge computing systems that store, filter, and analyze the incoming health data in real time. On top of this, the application layer provides interfaces for doctors, patients, and caregivers through mobile apps or hospital dashboards. Finally, the security and privacy layer ensures encryption, authentication, and compliance with healthcare regulations.

A. UML Diagrams

Collaboration in a project like patient healthcare monitoring using IoT devices is essential because it brings together expertise from multiple domains to ensure success. Engineers, healthcare professionals, data scientists, and patients collaborate to create a robust ecosystem where technology, medicine, and patient engagement converge.

B. Block Diagram

The block diagram in Fig. 4 illustrates the main components of the IoT system for vital monitoring.

The components are:

- SPO2 SENSOR: In the project, the SpO2 sensor is used to measure the vital beats per second and oxygen saturation.
- Temp Sensor: In our project, we are using a Temp Sensor to know the temperature value.
- Vibrator sensor: In our project, we are using a vibration sensor to detect patient falls or significant movement.

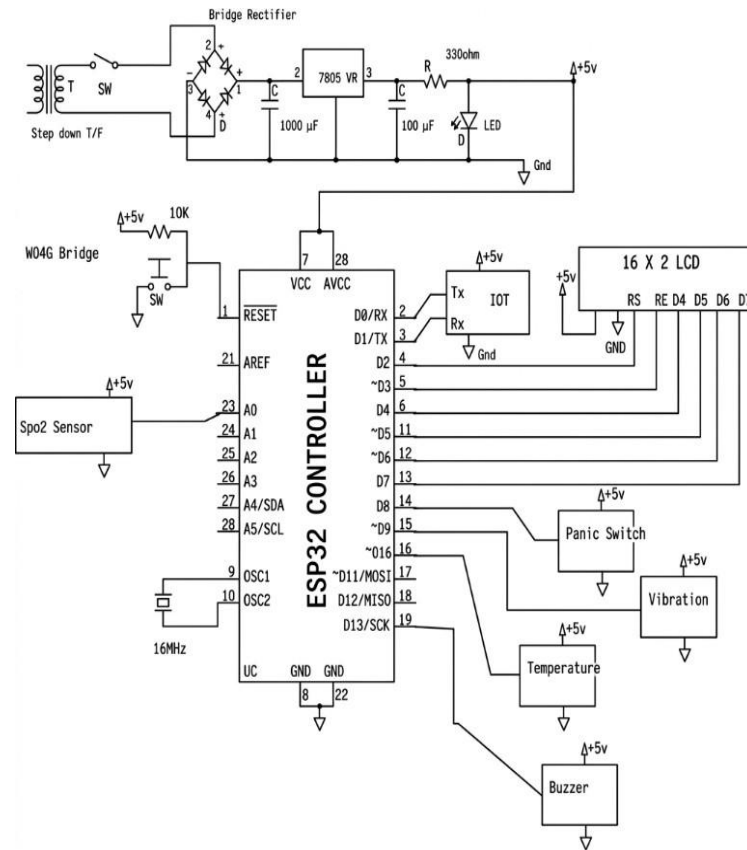


Fig 1: Collaboration diagram of the system.

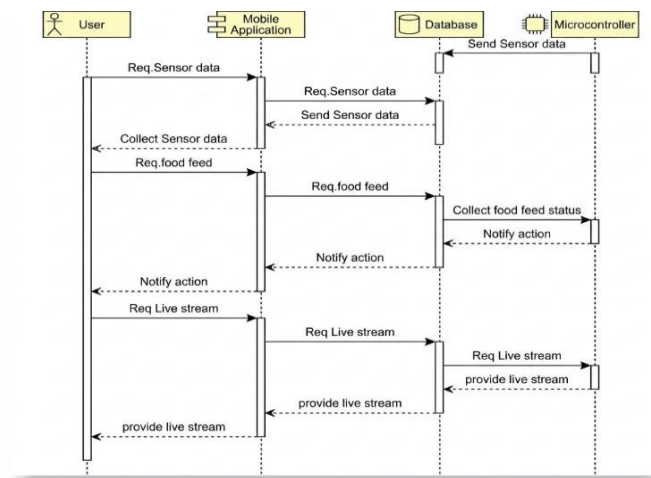


Fig 2: Sequence diagram of the system

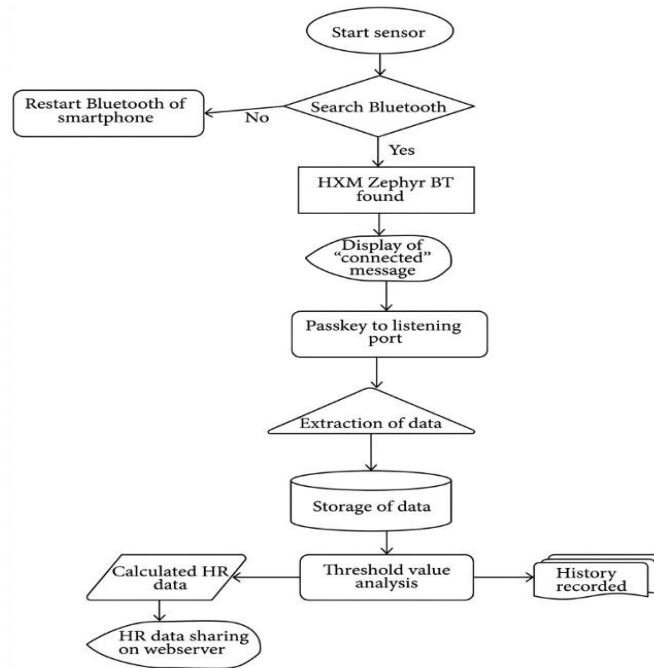


Fig 3: Activity diagram of the system.

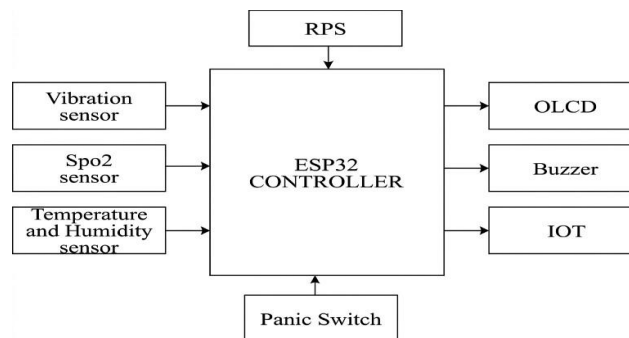


Fig 4: Block diagram for IoT for VitalGuard.

- ESP32: It is designed to achieve the best power and RF performance, showing robustness, versatility, and reliability in a wide variety of applications.
- LCD: In this project, it is used to show the output values on a digital screen.

IV. Implementation

Implementation is the phase of a project in which all the theoretical design, concepts and intended specifications are built into a tangible working system. It acts as an important stage in the system development life cycle and helps to ensure that a proposed solution is successfully implemented into practice. In the implementation phase, there is detailed planing and arranging of needed resources based on existing system with not only its limitations but also technical and operational constraints leading to the development of integrating models suitable for the new system. It also consists of designing and choosing appropriate techniques (direct, parallel, phased) for the system overhaul so as to take over with min service disruption. Finally, the

implementation stage includes system testing of the software and hardware used in data migration and performance monitoring; and evaluating either parallel operation or direct changeover (or "big bang" changeover) to confirm that the new system meets business goals, does what it is required to do efficiently, works on computers and databases without crashing, passes security issues by users (we have no need this anymore), connects according to protocols between each device for low latency simulation. Implementation plays a vital role in the successful deployment and acceptance of system in a target environment by executing all planned activities, combed with ongoing evaluation.

A. Hardware and Software Components

The implementation of a patient health care monitoring system using IoT devices involves translating the project design into a functional, real-world solution. The process begins with hardware setup, where sensors such as heart rate monitors and temperature sensors are integrated with IoT-enabled devices.

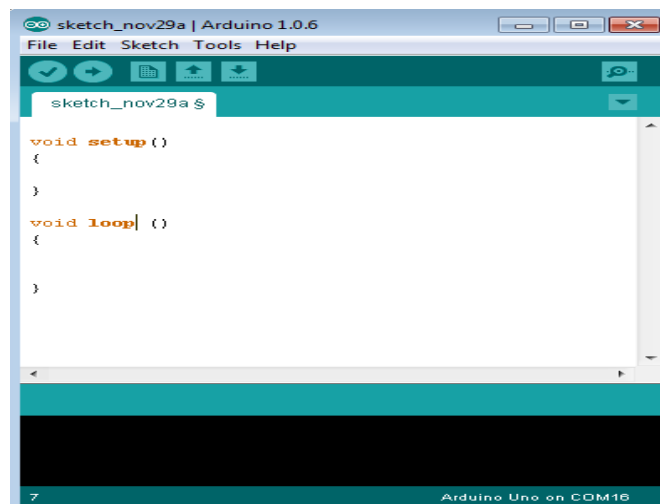


Fig 5: Structure of Arduino IDE.

1) MAX30102 Pulse Oximeter Sensor: The MAX30102 is a heart rate monitor and pulse oximeter biosensor module designed for wearable health. It uses Photoplethysmography (PPG) by shining light onto the skin and measuring the reflected light. It communicates through a standard I2C compatible interface.

2) Temperature Sensor: A temperature sensor is a crucial component that enables continuous measurement of a patient's body temperature. Digital sensors like the DS18B20 are widely used in healthcare projects because of their accuracy, reliability, and ease of integration. Fig. 6 shows the temperature probe used.



Fig 6: Temperature thermal probe sensor.

3) Mercury Switch Vibration TILT Sensor: A mercury switch is an electrical switch that opens and closes a circuit when a small amount of liquid mercury connects metal electrodes. The most common is the mercury tilt switch. A vibration/tilt sensor is a motion-detection device that identifies vibration, shock, or changes in orientation, making it useful for detecting patient falls. Fig. 7 shows a mercury switch.

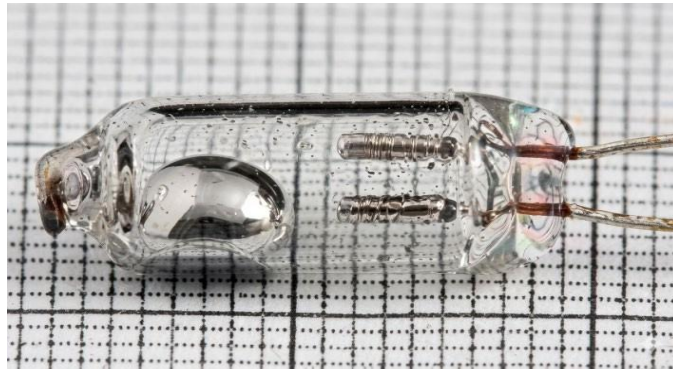


Fig 7: Mercury switch.

4) Buzzer: A buzzer is a simple but effective output device used to provide alerts and notifications. It acts as an immediate feedback mechanism when abnormal health readings are detected. The buzzer is connected to the microcontroller, which triggers an alarm when programmed conditions are met. This ensures real-time awareness of emergencies. Fig. 8 shows the buzzer.



Fig 8: Buzzer.

B. Coding

Programs written using Arduino Software (IDE) are called sketches. The following code snippet shows the core logic of the monitoring system.

```
#include <LiquidCrystal.h>
#include <SoftwareSerial.h>
#include "dht.h"
#include <Wire.h>
#include "MAX30105.h"
#include "heartRate.h"

// LCD Pins: RS, E, D4, D5, D6, D7
LiquidCrystal lcd(6, 7, 5, 4, 3, 2);

// Pin Definitions
const int vib = 9;
const int buzzer = 13;

// Sensor Variables
dht DHT;
#define DHTPIN A0

float tempc = 0;
float humc = 0;
int hbv = 0;
int spo2v = 0;

char vib_string[5] = "OFF";

// Serial for WiFi (example)
SoftwareSerial mySerial(10, 11);

// Function Prototypes
void wifiInit();
void beep();
void upload(float temp, int hb, char* vib, int hb2, int spo2);
void get_hb_spo2();

void setup() {
  pinMode(buzzer, OUTPUT);
  pinMode(vib, INPUT_PULLUP);

  digitalWrite(buzzer, HIGH);

  Serial.begin(9600);
  mySerial.begin(9600);

  lcd.begin(16, 2);
```

```
lcd.clear();

lcd.setCursor(0, 0);
lcd.print(" IOT Tele ECG ");
lcd.setCursor(0, 1);
lcd.print(" Monitoring ");
delay(2500);

wifiInit();
delay(2500);
}

void loop() {
  // Read Temperature & Humidity
  DHT.read11(DHTPIN);
  tempc = DHT.temperature;
  humc = DHT.humidity;

  // High Temperature Alert
  if (tempc > 40) {
    beep();
    upload(tempc, hbv, vib_string, hbv, spo2v);
  }

  // Vibration Detection
  if (digitalRead(vib) == LOW) {
    lcd.setCursor(12, 0);
    lcd.print("ON ");
    strcpy(vib_string, "ON");

    beep();
    upload(tempc, hbv, vib_string, hbv, spo2v);
  } else {
    lcd.setCursor(12, 0);
    lcd.print("OFF");
    strcpy(vib_string, "OFF");
  }

  // Read Heart Rate & SpO2
  get_hb_spo2();

  delay(1000);
}

// ----- Functions -----

void beep() {
  digitalWrite(buzzer, LOW);
  delay(500);
}
```

```
digitalWrite(buzzer, HIGH);
}

void wifInit() {
  Serial.println("Initializing WiFi...");
  // Add WiFi initialization code here
}

void upload(float temp, int hb, char* vib, int hb2, int spo2) {
  Serial.println("Uploading Data...");
  Serial.print("Temp: "); Serial.println(temp);
  Serial.print("Heart Rate: "); Serial.println(hb);
  Serial.print("Vibration: "); Serial.println(vib);
  Serial.print("SpO2: "); Serial.println(spo2);

  // Add actual IoT upload logic here
}

void get_hb_spo2() {
  // Placeholder for MAX30105 sensor logic
  hbv = random(70, 90); // Simulated heart rate
  spo2v = random(95, 100); // Simulated SpO2

  lcd.setCursor(0, 1);
  lcd.print("HR:");
  lcd.print(hbv);
  lcd.print(" SPO2:");
  lcd.print(spo2v);
}
```

Result Web Page Code (HTML)

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Patient Monitoring</title>

<style>
body {
  font-family: Arial, sans-serif;
}

h2 {
  text-align: center;
}

table {
```

```
border-collapse: collapse;
width: 80%;
margin: auto;
}

th, td {
text-align: center;
padding: 10px;
border: 1px solid #ddd;
}

tr:nth-child(even) {
background-color: #f2f2f2;
}

th {
background-color: #008080;
color: white;
}
</style>

</head>

<body>

<h2>Patient Health Monitoring Data</h2>

<table>
<tr>
<th>Temperature (°C)</th>
<th>Heart Rate (BPM)</th>
<th>Vibration</th>
<th>SpO2 (%)</th>
</tr>
<tr>
<td>36</td>
<td>75</td>
<td>OFF</td>
<td>98</td>
</tr>
</table>

</body>
</html>
```

C. Execution of Arduino Code

The execution phase involves uploading the program to the microcontroller and verifying the functioning of the hardware.

- 1) Install Arduino IDE: Download and install the Arduino IDE from the official website.
- 2) Connect Hardware: Connect the Arduino/ESP32 board to the computer. Ensure all sensors (DHT11, MAX30102, Vibration) and components (LCD, Buzzer) are properly connected.
- 3) Select Board and Port: In the IDE, go to 'Tools -> Board' to select the appropriate board and 'Tools -> Port' to select the correct COM port.
- 4) Install Required Libraries: Install 'LiquidCrystal', 'DHT sensor library', 'MAX30105 library', and 'Wire' library from 'Sketch -> Include Library -> Manage Libraries'.
- 5) Upload the Code: Copy the code into the IDE, verify it, and upload it to the microcontroller.
- 6) Serial Monitor Check: Open the Serial Monitor to observe sensor values and debugging messages.
- 7) System Operation: After upload, the LCD displays sensor data, the buzzer activates on threshold breach, and data is sent to the IoT server via Wi-Fi.

V. Testing Methodology

Software testing is an extremely creative and intellectually challenging task. Testing an application can only reveal that one or more defects exist; it cannot prove that the application is error-free.

A. Validation Activities

In our project, validation refers to ensuring the IoT-based system meets its intended objectives. This involves comparing system outputs against expected results.

- Unit Testing: Each individual component (sensors, buzzer, microcontroller code) is tested separately.
- Integration Testing: Focuses on how modules interact—e.g., whether sensor data flows properly from the sensor to the microcontroller and then to the cloud.
- Functional Testing: Ensures the system performs its intended functions, such as monitoring vitals and triggering alerts.
- System Testing: Evaluates the entire system as a whole under real-world conditions.
- Acceptance Testing: The final stage where the system is evaluated by end-users (patients, doctors) to confirm it is practical and user-friendly.

B. Testing Techniques

1) White-box testing: This method tests the internal structures or workings of an application. In our project, this involves reviewing the code logic to ensure all paths are executed correctly, especially the conditions for triggering alerts.

2) Black-box testing: This method examines the functionality of an application without peering into its internal structures. The tester focuses only on the inputs and outputs. For this project, black-box testing involves simulating patient data (e.g., high heart rate) and verifying that the system responds correctly (e.g., buzzer sounds, alert is sent).

VI. Results and Discussion

The project resulted in a fully functional system that demonstrates how modern technology can transform healthcare into a proactive, efficient, and patient-centered process. The system successfully integrates sensors, microcontrollers, and cloud platforms to continuously monitor vital signs. Fig. 9 shows the physical prototype in operation, with a user's finger on the MAX30102 sensor. The real-time data is displayed on an LCD screen, as shown in Fig. 10. It displays Heart Rate (H), SpO2 (S), and Temperature (T).

Data is securely transmitted and displayed on a user-friendly web dashboard. Fig. 11 shows the data logged in a tabular format on the server, with timestamps. The server also provides graphical visualization of the data over time, as seen in Fig. 12, which helps in trend analysis. Testing and validation confirm that the system is accurate, reliable, and compliant with privacy standards. The system provided accurate readings, triggered alerts when abnormal values were detected, and allowed caregivers to access patient information remotely. Overall, the results validate that the IoT-based solution meets the project objectives of improving patient safety and enhancing healthcare efficiency

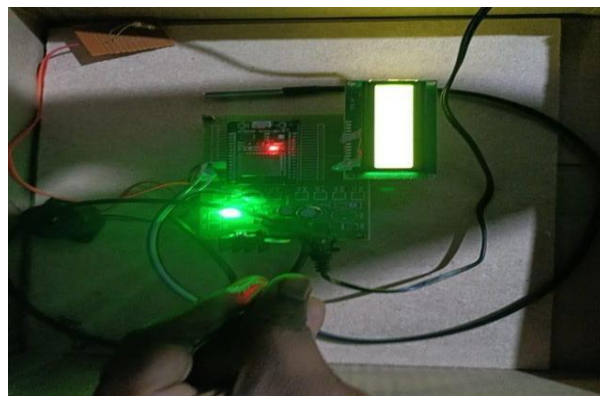


Fig 9: Hardware prototype during operation.



Fig 10: LCD displaying real-time sensor data.

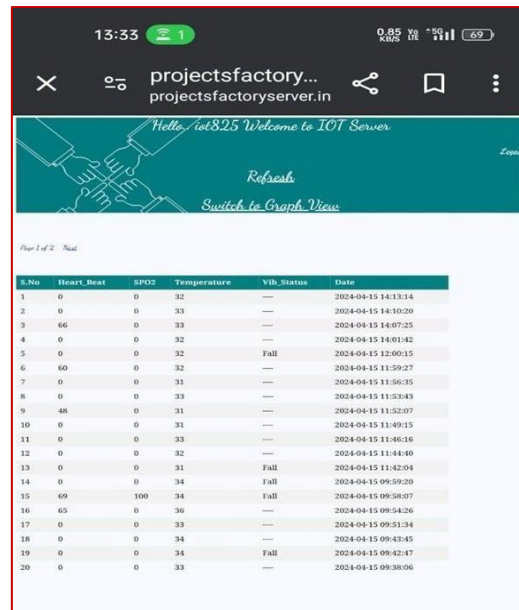


Fig 11: Web server interface showing logged data in a table.

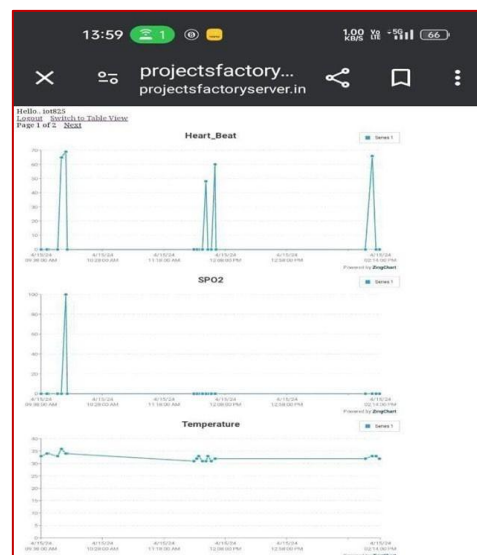


Fig 12: Graphical representation of Heart Beat, SpO2, and Temperature data on the web server

VII. Conclusion and Future scope

We designed and implemented an ESP32 Controller-based Tele-Health System over the Internet of Things, integrating input modules like temperature, humidity, position, heart rate, and oxygen monitoring sensors, with output modules including an LCD, buzzer, and a wireless communication system. In this system, various health parameters were monitored and recorded on an IoT server platform. The values of these parameters were analyzed, and alerts were triggered. The project on patient healthcare monitoring using IoT devices successfully demonstrates how technology can revolutionize healthcare patients and healthcare providers, ultimately improving health outcomes. Wearables such as fitness trackers and blood glucose

monitors have taken healthcare by storm. Many physicians have supported IoT as an effective patient engagement technology. In the future, we can add other sensors to enhance the health monitoring system, such as a glucometer and body fat measurement device. Another new feature could be monitoring fish health in an aquarium; at present, we only monitor water quality parameters. In the future, we could use special devices to monitor fish health to reduce losses. For the patient monitoring system, advancements could include integrating machine learning algorithms to predict health events before they become critical and expanding the platform to include a wider range of chronic diseases.

References

- [1]. Fezari, M., Bousbia-Salah, M., & Bedda, M. (2008). Microcontroller based heart rate monitor. *The International Arab Journal of Information Technology*, 5(4), 153–157.
- [2]. Lee, H., Park, S., & Woo, E. (1997). Remote patient monitoring through the World-Wide Web. *Proceedings of the 19th International Conference of IEEE*, Chicago, IL, USA, 928–931.
- [3]. Mamidi, M., & Neeraja, K. Pulse checking and heart assault location. *International Journal of Innovations in Engineering and Technology (IJIET)*.
- [4]. Baker, S. B., Xiang, W., & Atkinson, I. (2015). Internet of Things for smart healthcare: Technologies, challenges, and opportunities.
- [5]. Ani, R., & Krishna, A. (2017). IoT based patient monitoring and diagnostic prediction tool using ensemble classifier.
- [6]. Aref, M. H., El-Shinnawi, A. A., & Sharawi, A. A. (2018). Wireless nurse call system in medical institutions. 6(2), 40–45.
- [7]. D'Souza, M., Ros, M., & Postula, A. Remote medical information system network for patient ECG monitoring. *Digital System*.
- [8]. Lee, H., Park, S., & Woo, E. (1997). Remote patient monitoring through the World-Wide Web. *Proceedings of the 19th International Conference of IEEE*, Chicago, IL, USA, 928–931.
- [9]. Hemanth kumar B V, Jhanani M, S. Balaji, "IoT Based Health Monitoring System for Drivers", *Journal of Next Generation Technology (ISSN: 2583-021X)*, vol. 5, no. 2, pp. 140-148, April 2025.