iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** VI **Month of publication:** June 2025

DOI: https://doi.org/10.22214/ijraset.2025.72050

IoT-Based Patient Monitoring System

Shivani Chavan[1], Sakshi Mairal[2], Kalyani Chavan[3], Vijay Devkar[4]
Adsul Technical Campus, Chas, Ahilyanagar

Abstract: The IoT-Based Patient Health Monitoring System is an innovative approach to enhancing healthcare delivery through continuous, real-time tracking of critical health parameters. The system integrates advanced sensors to measure SpO2, heart rate, body temperature, room temperature, humidity, and ECG signals. These parameters are collected, processed, and transmitted to a web-based IoT platform, enabling caregivers and healthcare providers to monitor patient health remotely. The goal is to reduce response times in medical emergencies and provide effective remote healthcare solutions for patients in homes, rural areas, or understaffed medical facilities.

The system employs a combination of sensors, such as the MAX30100 for SpO2 and heart rate, the DS18B20 for body temperature, the DHT11 for room temperature and humidity, and an ECG sensor for cardiac monitoring. The ESP32 microcontroller serves as the central processing and communication unit, ensuring seamless integration and wireless data transfer. By leveraging IoT platforms, the system offers real-time data visualization and alert mechanisms to notify caregivers when health parameters cross critical thresholds. This ensures timely medical intervention and enhances patient safety.

This project not only improves the quality of patient care but also demonstrates the potential of IoT in revolutionizing the healthcare sector. It provides a scalable, cost-effective, and user-friendly solution for remote health monitoring. With its real-time capabilities and portability, the system is poised to transform how health data is collected, analyzed, and utilized, paving the way for smarter healthcare systems in the future.

I. INTRODUCTION

In today's fast-paced world, there is an increasing demand for accessible and efficient healthcare solutions, particularly for remote areas and patients requiring continuous monitoring. Traditional health monitoring methods often require frequent hospital visits, which can be both inconvenient and costly. To address these challenges, the integration of Internet of Things (IoT) technology in healthcare has paved the way for innovative solutions, enabling real-time monitoring and analysis of patient health parameters from virtually anywhere.

The IoT-Based Patient Health Monitoring System is a comprehensive solution designed to track vital health parameters such as blood oxygen saturation (SpO2), heart rate, body temperature, room temperature, humidity, and ECG signals. This system uses a range of sensors, including the MAX30100, DS18B20, DHT11, and an ECG sensor, to collect and process data. The ESP32 microcontroller acts as the central hub, transmitting the processed data to a cloud-based IoT platform. By providing healthcare professionals and caregivers with real-time data, the system ensures prompt action in the event of any health anomalies.

This project demonstrates how IoT technology can enhance the delivery of healthcare services, particularly for patients in rural or remote areas where access to medical facilities is limited. With its ability to provide continuous monitoring and remote data access, the system is a step forward in creating a more connected and efficient healthcare ecosystem, ultimately improving patient outcomes and reducing the burden on healthcare infrastructure.

II. OBJECTIVES

1) Real-Time Health Monitoring:

Develop a system capable of continuously measuring vital health parameters, including:

- Blood oxygen levels (SpO2) and heart rate using MAX30100.
- Body temperature using DS18B20.
- Room temperature and humidity using DHT11.
- ECG signals using an ECG sensor.

2) Wireless Data Transmission:

Utilize the ESP32 microcontroller to send collected health data wirelessly to an IoT platform for real-time visualization and analysis.

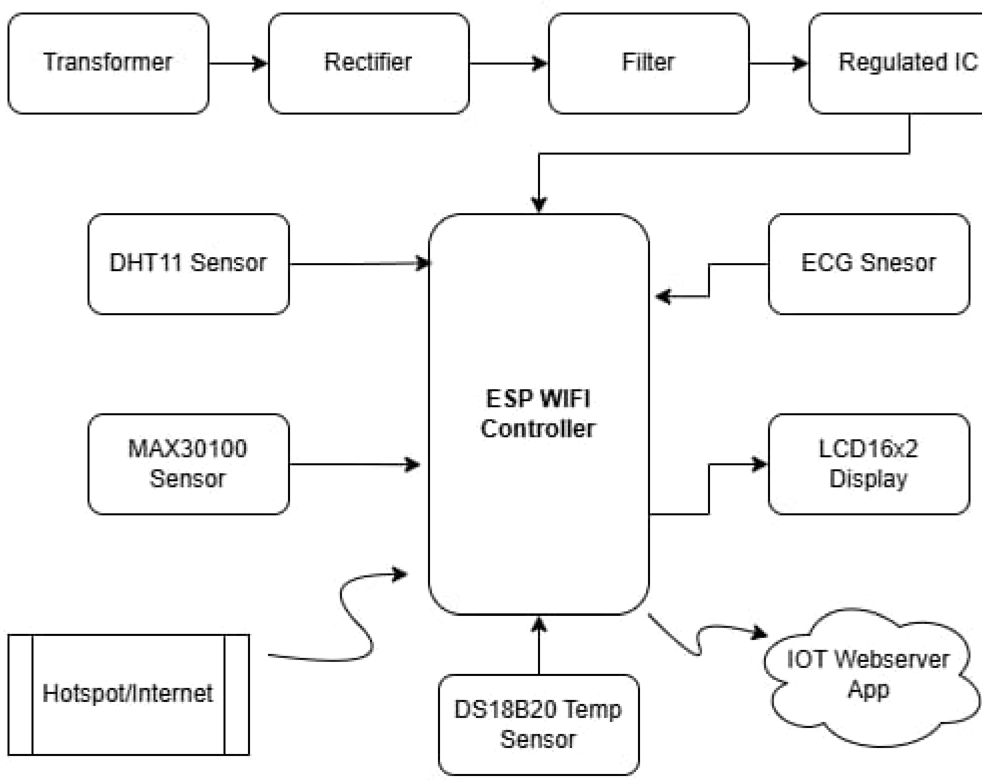
3) Alert System:
Implement a mechanism to generate alerts when health parameters deviate from predefined thresholds, enabling timely intervention.

4) User-Friendly Dashboard:
Create a user-friendly IoT web interface to display health data in an organized and easily interpretable format.

5) Scalability and Portability:
Design a scalable system that allows integration of additional sensors and portability for home and clinical use.

6) Cost-Effective Solution:
Build an affordable and reliable health monitoring system that can be used in both urban and rural healthcare setups.
This project aims to bridge the gap between patients and healthcare providers by leveraging IoT technologies for better health management and monitoring.

III. BLOCK DIAGRAM



Block Diagram Description
The system consists of a combination of sensors, a microcontroller (ESP WiFi Controller), power management circuitry, and a communication interface. It is designed to collect vital health data, process it, and transmit it to an IoT platform for real-time monitoring. Here's a detailed explanation of each block:

A. Power Supply Section
1) Transformer: Steps down the AC mains voltage to a suitable lower voltage.
2) Rectifier: Converts the AC voltage to DC voltage.

3) Filter: Removes any ripples from the rectified DC voltage, producing a steady DC output.
4) Regulated IC (e.g., LM7805): Ensures the voltage output is stable and suitable for powering the sensors, ESP WiFi Controller, and other components.

B. Sensors

1) MAX30100 Sensor:
- Measures SpO2 (oxygen saturation) and heart rate.
- Communicates with the ESP controller via I2C protocol.
- Provides critical real-time data for patient monitoring.

2) DS18B20 Temperature Sensor:
- Measures body temperature with high accuracy.
- Operates using the One-Wire communication protocol.
- Sends body temperature data to the ESP WiFi Controller.

3) DHT11 Sensor:
- Measures room temperature and humidity.
- Communicates with the ESP controller via a digital GPIO pin.
- Helps in monitoring the environmental conditions surrounding the patient.

4) ECG Sensor:
- Captures the electrical activity of the heart.
- Provides analog signals representing the ECG waveform.
- This data is fed to the ESP controller through its analog input for processing and transmission.

C. ESP WiFi Controller

- Acts as the central processing unit of the system.
- Sensor Data Collection:
  - Receives data from MAX30100, DS18B20, DHT11, and ECG sensors.
- Data Processing:
  - Filters and preprocesses the raw sensor data for accurate readings.
- WiFi Communication:
  - Establishes a connection with an IoT web server using WiFi.
  - Sends health parameters to the cloud for remote monitoring.
- Local Display:
  - Outputs critical health parameters to a connected 16x2 LCD for local monitoring.

D. LCD 16x2 Display

- Displays real-time health caregivers.data such as SpO2, heart rate, body temperature, and room conditions.
- Provides immediate visual feedback to users or

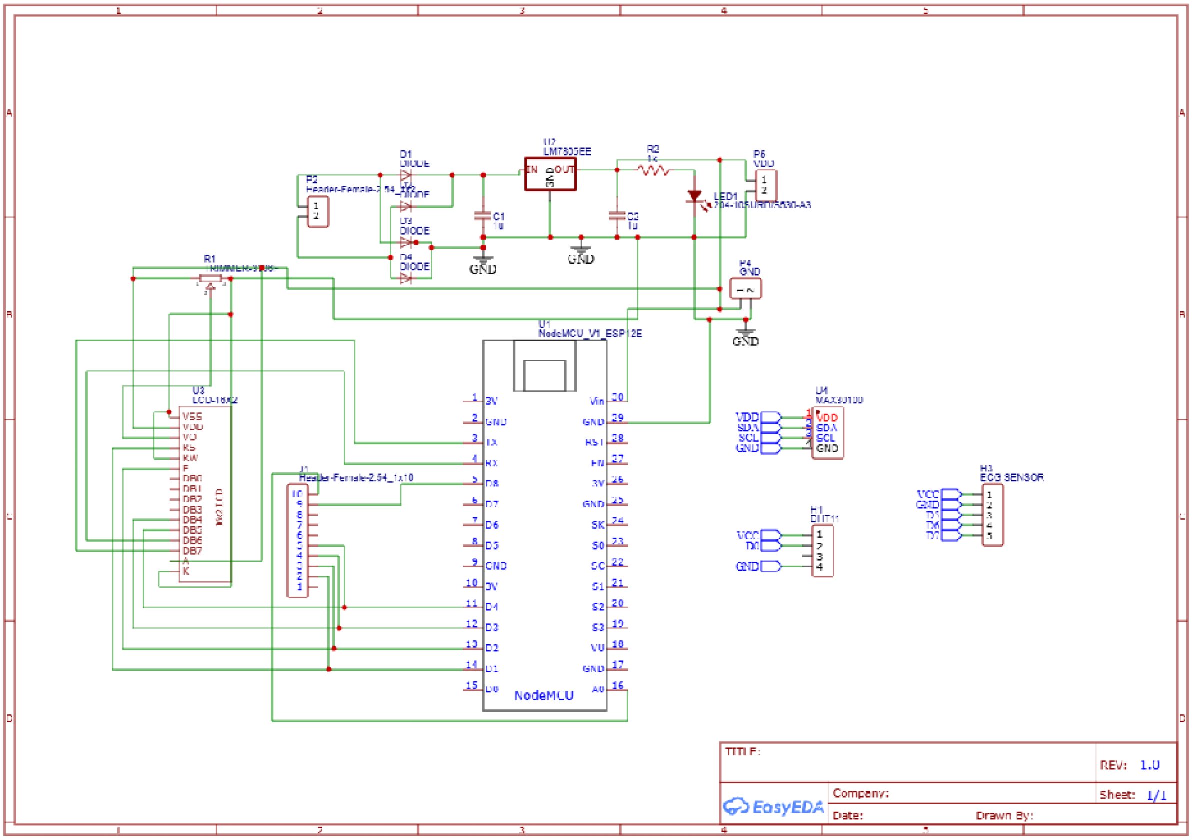
E. IoT Webserver App

- Receives sensor data from the ESP controller over the internet.
- Visualizes the data on a dashboard accessible from smartphones, tablets, or computers.
- Can trigger alerts when health parameters exceed predefined thresholds.

F. Internet Connectivity (Hotspot/Internet)

- Connects the ESP WiFi Controller to the IoT platform via a WiFi network.
- Enables data transmission from the controller to the IoT server for real-time access and analysis.

Circuit Diagram:



IV. METHODOLOGY

The methodology outlines the step-by-step approach used to design, develop, and implement the IoT-Based Patient Health Monitoring System. The process involves sensor integration, data processing, wireless communication, and IoT platform implementation. Below is a detailed description of each phase:

A. System Design and Planning

1) Objective Definition:

Define the primary goal of monitoring patient health parameters, such as SpO2, heart rate, body temperature, room temperature, humidity, and ECG signals.

2) Component Selection:

- Sensors: MAX30100 (SpO2 and heart rate), DS18B20 (body temperature), DHT11 (room temperature and humidity), and ECG sensor.
- Microcontroller: ESP32 is chosen for its built-in WiFi capabilities and support for multiple communication protocols (I2C, One-Wire, GPIO).
- IoT Platform: Select a cloud platform (e.g., ThingSpeak, Firebase, or Blynk) for data storage and visualization.
- Power Supply: Ensure stable power delivery to all components using a transformer, rectifier, filter, and regulator.

3) Block Diagram Development:

Create a high-level block diagram to represent the interconnection of components.

B. Hardware Development

The IoT-Based Patient Health Monitoring System incorporates a combination of sensors, a microcontroller, and peripheral devices to monitor health parameters and transmit them to an IoT platform for remote observation. Below is a detailed explanation of each component in paragraph format:

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

1) ESP32 WiFi Controller

The ESP32 WiFi Controller serves as the central processing unit of the system, integrating data from various sensors and enabling wireless communication. With its built-in WiFi capability, the ESP32 processes sensor data, displays the results on a 16x2 LCD, and transmits the information to an IoT web server. It plays a crucial role in ensuring real-time data monitoring and remote accessibility through the internet.

Key Specifications

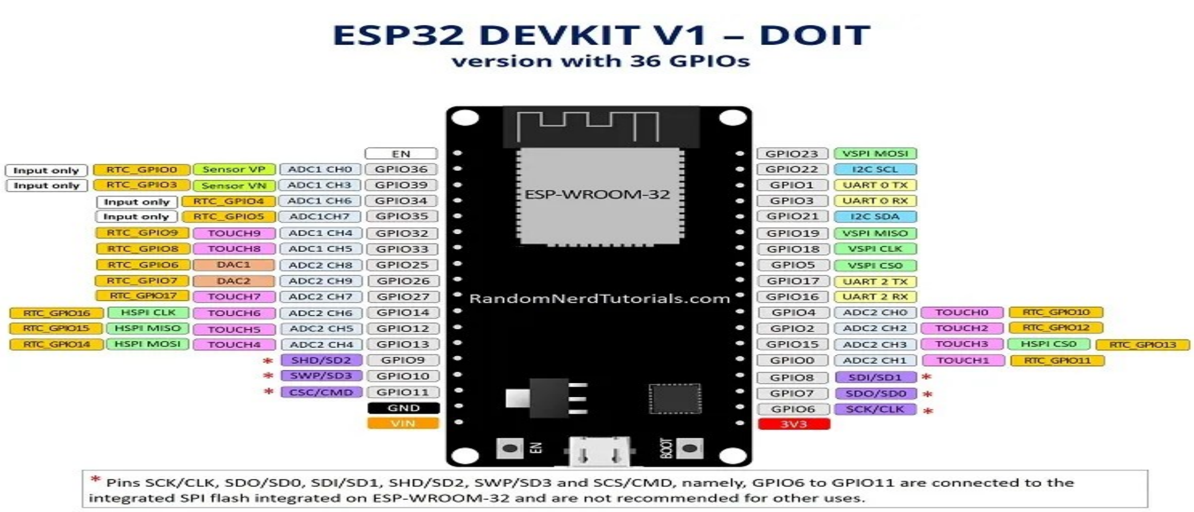
Feature	Details
Processor	Dual-core Tensilica Xtensa LX6 @ up to 240 MHz
Memory	520 KB SRAM, 16 KB RTC SRAM
Flash Memory	Typically 4 MB (external QSPI flash, varies by module)
WiFi	IEEE 802.11 b/g/n, 2.4 GHz
Bluetooth	Bluetooth v4.2 (Classic and BLE)
GPIO Pins	Up to 36 GPIO pins, multifunctional
ADC	12-bit ADC, up to 18 channels
DAC	2 × 8-bit DAC channels
Communication Interfaces	UART (3x), SPI (4x), I2C (2x), I2S (2x), CAN, PWM, SDIO
Timers	2 × 64-bit general-purpose timers, multiple PWM channels
Operating Voltage	3.3V
Power Modes	Active, Sleep, Deep Sleep
Operating Temperature	-40°C to +125°C
Package	QFN-48 (module), other variants available

Features

1) WiFi and Bluetooth Dual Mode: Offers seamless connectivity for IoT applications.
2) Low Power Consumption: Power-saving modes for battery-operated devices.
3) Wide Peripheral Support: Multiple ADC, DAC, UART, SPI, and I2C interfaces for sensor integration.
4) Real-Time Operating System (RTOS): Supports FreeRTOS for advanced multitasking.
5) High Processing Speed: Efficient for real-time data acquisition and IoT tasks.

Pinout Description

The ESP32 has a versatile pinout with up to 36 GPIOs, which can be configured for various purposes such as analog input, digital I/O, PWM, and communication protocols. Below is a summarized pinout description:



International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

Pin	Function	Notes
GPIO0	Input/Output, Boot Mode Select	Used for entering boot mode; pulled down internally
GPIO2	Input/Output, ADC, DAC, Touch	Multipurpose pin
GPIO4	Input/Output, ADC, Touch	Multipurpose pin
GPIO12	Input/Output, ADC, Touch, HSPI_MISO	Can be used as SPI, touch, or general-purpose pin
GPIO13	Input/Output, ADC, Touch, HSPI_MOSI	Can be used as SPI, touch, or general-purpose pin
GPIO14	Input/Output, ADC, Touch, HSPI_CLK	Can be used as SPI, touch, or general-purpose pin
GPIO15	Input/Output, ADC, Touch, HSPI_CS	Can be used as SPI, touch, or general-purpose pin
GPIO16	Input/Output	Used for general-purpose input/output
GPIO17	Input/Output	Used for general-purpose input/output
GPIO18	Input/Output, VSPI_CLK	Default SPI Clock
GPIO19	Input/Output, VSPI_MISO	Default SPI MISO
GPIO21	Input/Output, I2C SDA	Default I2C Data Line
GPIO22	Input/Output, I2C SCL	Default I2C Clock Line
GPIO23	Input/Output, VSPI_MOSI	Default SPI MOSI
GPIO25	Input/Output, ADC, DAC1	Can be used for DAC or ADC
GPIO26	Input/Output, ADC, DAC2	Can be used for DAC or ADC
GPIO27	Input/Output, ADC	General-purpose pin with ADC functionality
GPIO32	Input/Output, ADC, Touch	Multipurpose pin
GPIO33	Input/Output, ADC, Touch	Multipurpose pin
GPIO34-39	Input Only, ADC	These pins are analog inputs only, no digital I/O

Power and Reset Pins

- **VIN:** Input voltage (5V) for powering the ESP32 through an external regulator or USB.
- **3V3:** Output pin that provides a regulated 3.3V for external devices.
- **GND:** Ground connections.
- **EN (Enable):** Used to enable or reset the ESP32 module.
- **RST (Reset):** Active-low pin for resetting the ESP32.

Communication Interfaces

- **UART:** Used for debugging and serial communication (e.g., with PC or other microcontrollers).
- **SPI:** Interface for high-speed peripherals like sensors, DACs, or memory.
- **I2C:** Commonly used for interfacing with sensors like MAX30100 or DS18B20.
- **ADC/DAC:** Built-in ADC channels are used for analog sensors, and DAC channels can generate analog outputs.

2) MAX30100 Sensor

The **MAX30100 Sensor** is a combined pulse oximeter and heart rate monitor that uses red and infrared LEDs along with photodetectors to measure SpO2 (blood oxygen saturation) and heart rate. By analyzing the changes in light absorption caused by blood flow, it generates accurate and essential health metrics, which are transmitted to the ESP32 via the I2C communication protocol. This makes it ideal for non-invasive monitoring of cardiovascular health.

Description of MAX30100

MAX30100 is a multipurpose sensor used for multiple applications. It is a **heart rate monitoring** sensor along with a **pulse oximeter**. The sensor comprises two Light Emitting Diodes, a photodetector, and a series of low noise signal processing devices to detect heart rate and to perform pulse oximetry.

Features of MAX03100

Here are some of the features and specifications of the MAX03100 Heart Rate Oxygen pulse sensor.

- Operating Voltage - 1.8V to 3.3V
- Input Current - 20mA
- Integrated Ambient Light Cancellation
- High Sample Rate Capability
- Fast Data Output Capability

Pin Configuration of MAX30100 Oximeter Module

Below is the pin configuration of the MAX30100 module. It is a 7 pin sensor module with an enabled I2C communication protocol to interact with the microcontroller.

Pin Type	Pin Function
VIN	Voltage Input
SCL	I2C - Serial Clock
SDA	I2C - Serial Data
INT	Active low interrupt
IRD	IR LED Cathode and LED Driver Connection Point(Leave floating in the circuit)
RD	Red LED Cathode and LED Driver Connection Point(Leave floating in the circuit)
GND	Ground pin

Note: There are two module versions of the MAX30100 sensor module. The one which we have showcased is the 7-pin version, whereas there is a 5-pin version as well which does not have IRQ and RD pin.

3) DS18B20 Digital Temperature Sensor

The **DS18B20 Digital Temperature Sensor** is a highly accurate sensor that measures the patient's body temperature. Its unique One-Wire communication protocol simplifies its interface with the ESP32, requiring minimal GPIO pins for operation. The sensor outputs precise temperature readings in degrees Celsius, which are crucial for identifying fever or hypothermia in patients.

4) DHT11

The **DHT11 Temperature and Humidity Sensor** measures environmental conditions, including room temperature and humidity levels. It communicates with the ESP32 via a digital GPIO pin and provides additional context to the patient's health data by monitoring the surrounding environment. This helps in assessing how external factors may affect the patient's overall well-being.

5) ECG Sensor

The **ECG Sensor** is responsible for capturing the electrical activity of the heart to produce an electrocardiogram (ECG) signal. By detecting and amplifying the heart's electrical impulses through electrodes placed on the patient, the ECG sensor outputs analog signals to the ESP32. These signals are processed to generate an ECG waveform, providing critical insights into the heart's rhythm and electrical activity.

6) 16x2 LCD Display

The **16x2 LCD Display** acts as the local interface, presenting real-time data such as SpO2, heart rate, body temperature, and room temperature. This display provides immediate feedback to caregivers or patients without requiring a separate device for visualization. It uses either the I2C protocol or GPIO pins to interface with the ESP32.

7) Power Supply

The **Power Supply Unit** provides the required voltage and current for all components. It includes a transformer to step down AC mains voltage, a rectifier to convert AC to DC, and a voltage regulator (such as the LM7805) to deliver a stable 5V output. This ensures uninterrupted operation and protects sensitive components from voltage fluctuations.

Lastly, supporting components such as resistors, capacitors, and connecting wires ensure the proper functioning of the circuit by managing voltage levels, reducing noise, and connecting various modules. The breadboard or printed circuit board (PCB) organizes and secures all components, providing a reliable platform for the system.

This integration of components ensures accurate health monitoring, real-time data transmission, and ease of use, making the system an effective solution for modern healthcare challenges.

C. Software Development

1) Firmware Development for ESP32:

- Sensor Data Acquisition: Write code to read sensor data from MAX30100, DS18B20, DHT11, and the ECG sensor.
- Data Processing: Process raw sensor data into meaningful values (e.g., convert ADC values to temperature, heart rate, etc.).
- WiFi Configuration: Configure the ESP32 to connect to a WiFi network and establish communication with the IoT server.
- Data Transmission:Send processed data to the IoT webserver in real time using HTTP or MQTT protocols.
- Alert Mechanism: Implement threshold-based alerts in the ESP32 firmware to trigger warnings for critical health conditions.

2) IoT Web Server Configuration:

- Cloud Platform Setup: Set up an IoT platform to receive, store, and display health parameters.
- Dashboard Development: Design a user-friendly dashboard for visualizing health data in real time, including graphs and charts.
- Notification System: Configure the platform to send alerts (via email or SMS) when abnormal health values are detected.

D. Testing and Calibration

- Sensor Calibration: Test and calibrate each sensor to ensure accurate readings under various conditions.
- System Testing:
  - Validate communication between sensors, the ESP32, and the IoT server.
  - Test the display of real-time data on both the LCD and the IoT dashboard.

- Error Handling: Incorporate error detection and correction mechanisms for sensor faults or network failures.

E. Deployment and Real-Time Monitoring

- Prototype Deployment: Set up the system in a real-world environment to monitor patient health parameters.
- Data Analysis: Use the IoT platform to analyze trends in patient health data and identify anomalies.
- Feedback Loop: Gather user feedback to improve the system's functionality and usability.

F. Maintenance and Scalability

- Periodic Maintenance: Ensure regular updates to the firmware and calibration of sensors for long-term accuracy.
- Scalability: Enable the system to integrate additional sensors or features, such as GPS for location tracking or wearable device compatibility.