# SensoGuard: IoT-Based Wearable Health Monitoring System

## 1. Field of the Invention

The present invention relates to the field of wearable health monitoring devices and IoT-based medical systems. In particular, it discloses a compact, wearable device incorporating multiple physiological sensors (e.g. motion, pressure, temperature, $SpO_2$), a low-cost microcontroller (ESP8266/NodeMCU), local alerts, and a cloud-connected backend for real-time data analytics and automated alerting.

## 2. Background and Prior Art

Conventional health monitoring relies on periodic clinical check-ups or simple consumer wearables, which provide only episodic data and can miss critical events between measurements. Typical wearable trackers or smartwatches measure a single parameter (e.g. heart rate or steps) and often store data locally or sync only with proprietary smartphone apps. These systems suffer from high latency and limited data fusion – most commercial solutions remain *"siloed"*, offering narrow insights and lacking real-time connectivity to caregivers. Inaccessible or delayed health alerts can be life-threatening for elderly or chronically ill users. Some existing IoT health devices have combined basic sensors (e.g. pulse oximeters with a Wi-Fi link), but none integrates multiple complex sensors (motion/accelerometer, pressure/load, temperature, heart rate/$SpO_2$) in a single wearable with an automated cloud‑alert pipeline. Prior devices typically use either Bluetooth (requiring constant phone pairing) or only simple threshold alarms, and they often use expensive commercial hardware. For example, high-end medical wearables provide clinical data but are costly and not designed for continuous personal use. In contrast, low-cost trackers (like simple fitness bands) lack multi-parameter fusion and advanced alert logic. Thus there is a need for a low-cost, multi-sensor, IoT-enabled wearable that offers continuous monitoring and immediate alerts via cloud intelligence.

## 3. Problem Statement and Objectives

SensoGuard addresses these deficiencies by achieving the following objectives (derived from the problems above):

- **Full-stack, real-time fusion:** Design a continuous, multi-sensor monitoring system that overcomes passive/episodic monitoring gaps.
- **Sensor suite integration:** Incorporate a comprehensive set of physiological sensors on the wearable – including 3-axis motion/gyro (e.g. MPU-6050), weight/bed-presence (HX711 + load cell), body temperature (DS18B20), and pulse-oximetry (MAX30100/30102).
- **Hybrid edge-cloud architecture:** Use a low-cost ESP8266 microcontroller for local sensor polling and immediate feedback, while offloading complex analysis to a cloud server. Implement a robust cloud-native data pipeline (e.g. Supabase Postgres with HTTPS API) and global visualization (e.g. Vercel/Next.js dashboard).
- **Automated server-side alerts:** Develop an intelligent serverless backend (e.g. Supabase Edge Function) that is automatically triggered on each new data insert. The function must

compare new and previous values, detect significant anomalies (e.g. >20% spikes in weight, temperature, or sudden motion events), and dispatch immediate alerts (e.g. email/SMS) to caregivers.

- **Low-cost and scalable implementation:** Build the system using affordable, widely-available hardware (NodeMCU/ESP8266) and free-tier or open-source cloud services. Ensure the firmware is open and modular for customization. Aim for battery-efficient operation (e.g. deep-sleep modes) to enable wearable use in daily life.

## 4. Summary of the Invention

SensoGuard is a smart wearable health-monitoring device that continuously fuses data from multiple onboard sensors to provide real-time analytics and alerts. In one embodiment, SensoGuard comprises a NodeMCU (ESP8266) microcontroller mounted on a wristband or bed-pad, connected to: an **MPU-6050** 6-axis gyro/accelerometer, an **HX711** 24-bit ADC with load cell (for weight/bed-exit detection), a **DS18B20** digital temperature sensor, and a **MAX30100/30102** pulse-oximeter (heart rate and $SpO_2$). The device also includes a local **LCD display** and **buzzer** for immediate feedback. All these components are integrated on a compact PCB or breadboard, powered via USB or a small rechargeable battery.

The ESP8266 firmware (Arduino C/C++) runs a loop that polls each sensor non-blockingly (using libraries like HX711_ADC, DallasTemperature, etc.), updates the LCD with current readings, and sounds the buzzer if any local threshold is exceeded. The microcontroller establishes secure Wi-Fi (HTTPS) connectivity and periodically packages the sensor readings into a JSON payload. It then transmits this data over the Internet to a cloud-hosted backend (e.g. Supabase Postgres database).

On the cloud side, each incoming data record triggers a **serverless function** (Supabase Edge Function). This function retrieves the immediately prior reading for each sensor and compares values. If a significant anomaly is found (for example, a sudden 20% rise in weight indicating fall or fever spike, or a large motion spike), the function automatically sends an alert email via an API (e.g. Resend) to a caregiver or medical professional. Meanwhile, a front-end web dashboard (built in Next.js on Vercel) continuously fetches and displays the latest readings and trends in real time, allowing remote monitoring of the wearer.

The novel features of SensoGuard include its **hybrid edge-cloud architecture** (simple on-device acquisition + sophisticated serverless analytics), **full multi-sensor integration** on a single wearable, and use of **open-source, low-cost hardware**. Unlike prior systems, SensoGuard does not rely on continuous smartphone pairing or GPS. It leverages Wi-Fi for direct Internet connectivity and modular firmware, enabling flexible customization. This combination of features enables proactive, 24/7 health surveillance for vulnerable users, with immediate alerts that were not possible in earlier passive or siloed devices.

## 5. Detailed Description of the Invention

**Hardware Components**

- **Microcontroller (ESP8266/NodeMCU):** Acts as the central edge processor. Its role is to interface with all sensors, manage Wi-Fi communications, and execute the firmware logic. The NodeMCU board provides a 32-bit MCU with integrated Wi-Fi (802.11), serial USB interface for programming, and multiple GPIO pins (3.3V logic). In the prototype, the ESP8266 is powered via USB (5V) and regulated onboard to 3.3V for the MCU.
- **Motion Sensor (MPU-6050):** A 6-axis inertial measurement unit combining a 3-axis gyroscope and 3-axis accelerometer. Connected via I²C (SCL to GPIO5 (D1), SDA to GPIO4 (D2)), it provides real-time orientation and acceleration data. The MPU-6050 is used to detect user activity, posture changes, and sudden movements (e.g. falls or restlessness). In firmware, the MPU-6050 is initialized and updated each loop, and high accelerations or abrupt orientation changes are flagged as potential fall events.
- **Load Cell with HX711 ADC:** A resistive force-sensing load cell coupled with an HX711 analog-to-digital converter module measures weight or pressure. The HX711 module (VCC to 5V, GND common, DATA pin to a digital GPIO, SCK to another GPIO) provides high-resolution weight readings. We employ the *HX711_ADC* non-blocking library to continuously sample weight without blocking the main loop (avoiding I²C conflicts). After calibration, the load cell can accurately track user weight on a bed or chair. Significant drops or surges in weight (e.g. bed-exit, sudden impact) can be used to infer falls or changes in presence.
- **Temperature Sensor (DS18B20):** A digital thermometer mounted on the skin or clothing monitors body temperature. It is connected on a one-wire bus (DATA line to a GPIO pin and power to 3.3V). The firmware uses the DallasTemperature library to read the DS18B20. A rapid spike in measured temperature above a threshold can signal fever or overheating.
- **Pulse Oximeter (MAX30100/30102):** An optical sensor module measures heart rate (via photoplethysmography) and blood oxygen saturation ($SpO_2$). Connected via I²C on the same bus as the MPU-6050 and LCD (SCL/D2, SDA/D1), it provides periodic red/IR light measurements. In the prototype, this module is read in the main loop to compute heart rate; a falling $SpO_2$ or arrhythmic pulse can indicate potential emergencies (though such advanced analysis is available in future versions).
- **User Interface (I²C LCD and Buzzer):** A 16×2 character LCD (I²C interface address 0x27) provides local readouts. The firmware updates the LCD to show key metrics (e.g. “Temp: 37.2°C, HR: 76 bpm, Weight: 68 kg”). An active piezo buzzer (GPIO digital output) sounds an alert if any real-time threshold is exceeded (for example, if weight > 25g beyond baseline, or if temperature exceeds a limit). This gives immediate feedback even without the cloud connection.
- **Power and Connectivity:** In the prototype, all sensor modules (except the buzzer) are powered from the NodeMCU’s VIN (5V) input to ensure stable supply and prevent 3.3V sag. The ESP8266 establishes secure Wi-Fi (using WiFiClientSecure with certificate validation disabled on prototype) and communicates with the cloud over HTTPS.

**Firmware and Logic**

The device firmware is written in C++ (Arduino environment) and runs a continuous, non-blocking loop. Key functions include:

1. **Sensor Polling:** The ESP8266 sequentially reads each sensor using appropriate libraries. The MPU-6050 gyroscope is updated, the HX711 load cell is sampled via *HX711_ADC*, the DS18B20 temperature is requested, and the MAX30102 (if used) is read. Careful timing ensures the HX711 and I²C sensors do not interfere.
2. **Local Feedback:** After acquiring data, the firmware updates the LCD display and checks simple thresholds. If a threshold is crossed (e.g. sudden weight drop on load cell, or temperature > 38°C), it toggles the buzzer. This provides an on-device alert independent of the network.
3. **Data Packaging:** Sensor readings (gyro X/Y/Z, weight, temperature, heart rate, $SpO_2$, timestamp, etc.) are aggregated into a JSON object. The firmware uses ArduinoJson library to format the payload efficiently.
4. **Cloud Transmission:** Periodically (e.g. every few seconds or minute), the ESP8266 establishes an HTTP(S) connection and sends the JSON data via a POST request to a REST API endpoint (Supabase). All readings are time-stamped and include a unique device ID.
5. **Sleep Mode (Optional):** Between measurements the ESP8266 may enter a deep-sleep mode to conserve power (implementation not shown). In continuous monitoring scenarios, it remains awake with optimized code.

By offloading heavy analysis, the microcontroller remains focused on reliable data acquisition and transmission. Its limited RAM and CPU are thus not overburdened by computations.

**Cloud Backend and Analytics**

The invention's novel backend leverages free-tier, scalable services. In one embodiment, sensor data is stored in a Supabase (Postgres) database. A single **Edge Function** (serverless code) is configured to trigger on each new database insert. This function retrieves the newly inserted row and the immediately preceding row. It computes percentage changes for weight, temperature, etc., and checks for anomalies (for example, >20% spike or sustained outlier). If an anomaly is detected (e.g. sudden fall detected by 80% weight drop and high accel), the function automatically issues an alert (email or SMS) via an API such as Resend or Twilio. The alert message includes which sensor triggered it and the critical values. This fully-automated alert pipeline is a key differentiator: complex "if X, then alert" logic is offloaded to the cloud rather than embedded in the device.

Separately, a web dashboard (Next.js on Vercel) provides real-time visualization. It subscribes to Supabase's real-time updates to automatically refresh charts/tables whenever new data arrives. Thus, family members or doctors can log into the dashboard to see live vitals. The combined system therefore offers an end-to-end solution: **wearable data capture → cloud storage → analytics → alerting**.

## 6. Brief Description of Drawings

**1. FIGURE – Front View (Wearable Device)**

A clean, technical front view of a wearable IoT health monitoring device designed like a wristband. The casing is smooth, rectangular, and slightly curved to fit the wrist. The front surface shows a small LCD screen displaying vitals (heart rate, temperature, movement icons). Include button placeholders on the side. Style: patent drawing, monochrome, high clarity, engineering style.

**2. FIGURE – Top View**

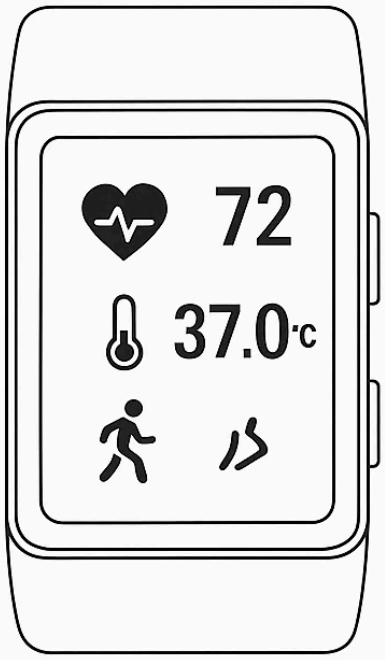
Top view of a compact wearable health monitoring device showing the entire flat surface of the upper casing. The LCD is centered. Vents or sensor windows are marked for temperature and optical sensors. Show strap connections on both ends. Style: patent drawing, clean lines, technical, monochrome.

**3. FIGURE – Bottom View**

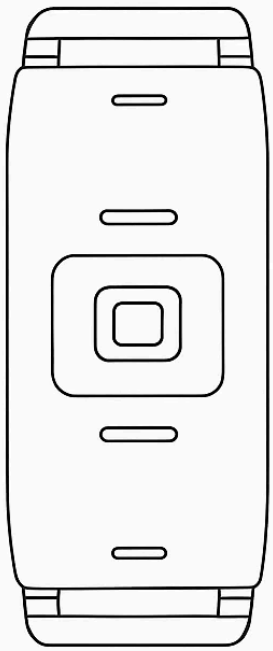
Bottom view of a wearable wristband-type device. The underside shows placement zones for sensors: load-cell contact points, MAX30100 optical sensor window touching skin, DS18B20 temperature contact pad. Include charging port or pins. Style: patent illustration, precise outlines, monochrome technical drawing.

**4. FIGURE – 3D Isometric View**

Three-dimensional isometric view of a wearable IoT health monitoring band. The image should emphasize the curved ergonomic design, the embedded sensors, LCD display, and strap. The 3D shape should show depth, slight shadowing, and engineering clarity. Style: patent-style 3D illustration, clean, professional, grayscale.

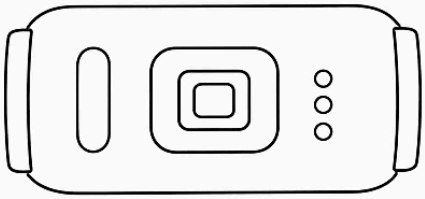



FRONT VIEW



TOP VIEW

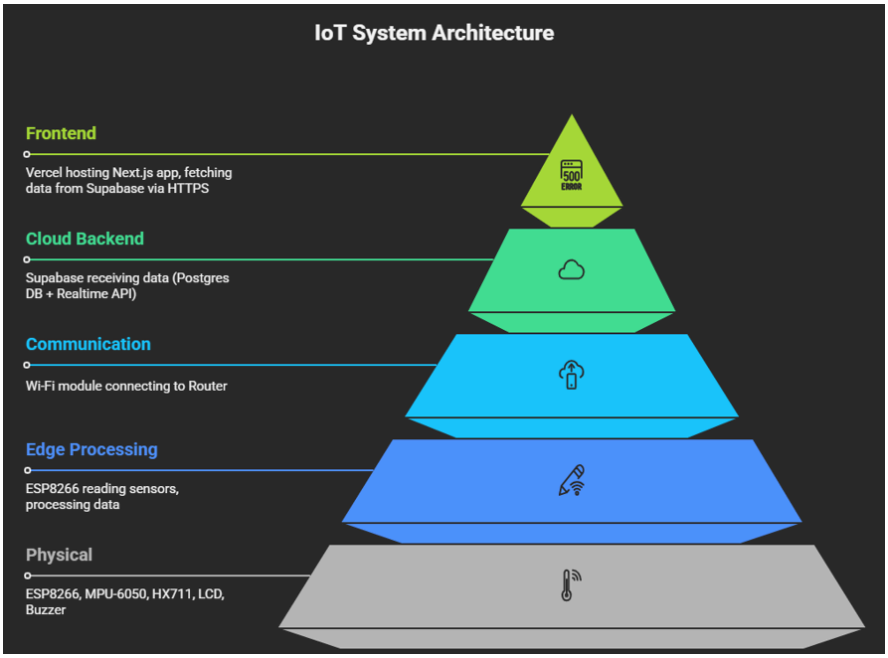


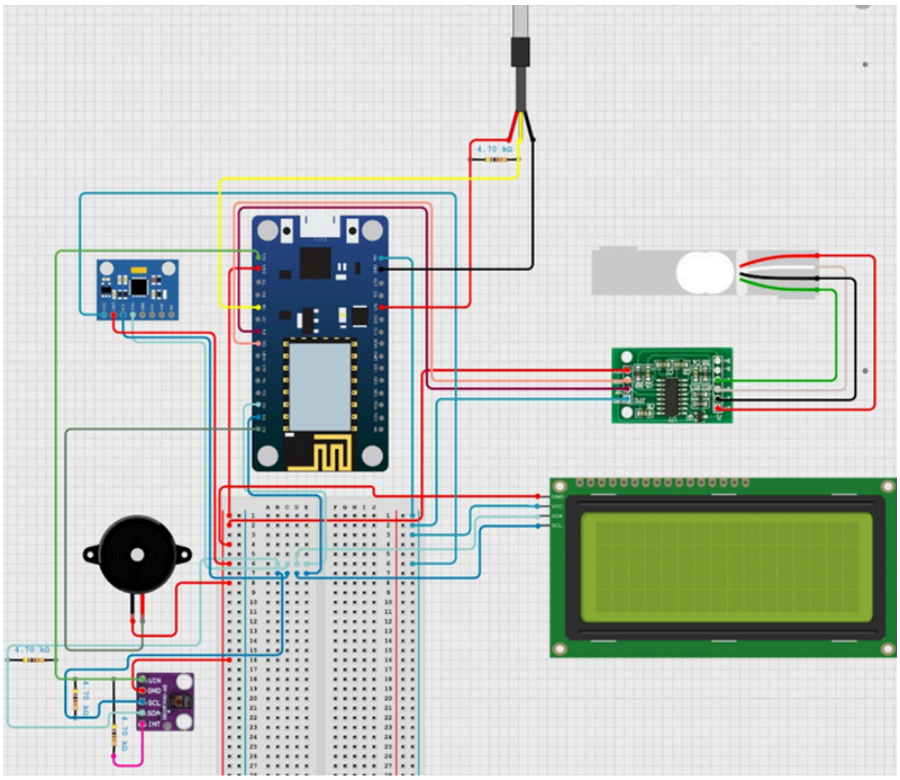



BOTTOM VIEW




3D ISOMETRIC VIEW


IoT System Architecture
Frontend
Vercel hosting Next.js app, fetching data from Supabase via HTTPS
Cloud Backend
Supabase receiving data (Postgres DB + Realtime API)
Communication
Wi-Fi module connecting to Router
Edge Processing
ESP8266 reading sensors, processing data
Physical
ESP8266, MPU-6050, HX711, LCD, Buzzer



## 7. Detailed Description of the Drawings

With reference to FIG. 1 and FIG. 2, the wearable device (100) integrates the following key hardware modules: the NodeMCU (ESP8266) microcontroller (101), motion sensor MPU-6050 (102), load cell with HX711 ADC (103), temperature sensor DS18B20 (104), pulse oximeter MAX30100/30102 (105), I²C LCD display (106), and buzzer (107). All grounds are common (GND), and power is supplied from a 5V source (e.g. USB/battery) with a 3.3V regulator for the MCU and DS18B20. The MPU-6050, MAX30102, and LCD share the I²C bus (SCL→ESP8266 D1, SDA→D2). The HX711 module uses two digital lines (DATA→ESP8266 D3, SCK→D4) with its own 5V supply. The DS18B20's data line connects to ESP8266 D5 with a 4.7k pull-up (not shown). FIG. 2 shows these connections and the on-device feedback elements.

FIG. 3 details the firmware sequence: on power-up, the system initializes all sensors and Wi-Fi. Then a repeating loop reads motion, load, temperature, and pulse; updates the LCD; checks thresholds (e.g. load cell >25g) to sound the buzzer; and composes a JSON payload. Every predefined interval (or continuously streaming), it posts this payload to the Supabase REST API.

FIG. 4 illustrates the cloud interaction: each data insert into the Supabase table (401) triggers an edge function (402). The edge function fetches the previous record for comparison. If (new_value > 1.2 * old_value) (i.e. >20% spike) for any parameter, it generates an alert. The function then calls an email API (403) to notify caregivers. Meanwhile, a Next.js front-end (404) is subscribed to the database and updates charts (450) and tables (451) live.

## 8. Claims

1. **A wearable health monitoring apparatus** comprising:
    - a microcontroller unit (e.g. ESP8266/NodeMCU) having wireless connectivity;
    - a plurality of physiological sensors interfaced to said microcontroller, including at least a motion sensor (MPU-6050), a pressure or weight sensor (load cell + HX711 ADC), a body temperature sensor (DS18B20), and a pulse oximeter (MAX30100/30102);
    - a local user interface comprising a display (LCD) and an alert actuator (buzzer);
    - wherein the microcontroller continuously polls all sensors, provides immediate feedback on the display and buzzer based on local thresholds, packages the sensor readings into data payloads, and transmits the payloads via a wireless protocol (e.g. Wi-Fi over HTTPS) to a remote server.
2. **The apparatus of claim 1**, wherein the microcontroller is programmed using the Arduino IDE and open-source firmware libraries (e.g. MPU6050, HX711_ADC, DallasTemperature, ArduinoJson, ESP8266WiFi).
3. **The apparatus of claim 1**, wherein the power module includes a replaceable or rechargeable battery (or USB input) and a voltage regulator, configured to supply 5V to the sensors (to avoid 3.3V sag) and regulated 3.3V to the microcontroller and temperature sensor.

4. **The apparatus of claim 1**, wherein the device is enclosed in a wearable form factor (e.g. wrist strap or band) suitable for personal health monitoring, and the display shows real-time values of the sensed parameters.
5. **The apparatus of claim 1**, wherein wireless communication uses HTTP(S) or MQTT protocols to send data to a cloud backend.
6. **A system for remote health monitoring**, comprising:
    - the apparatus of claim 1 (wearable device with sensors and MCU);
    - a cloud server with a database (e.g. Supabase Postgres) receiving the transmitted data;
    - a serverless analytics function triggered by each new database entry, programmed to compare the latest sensor readings with prior values and detect significant anomalies;
    - an alert module that, in response to the analytics function indicating an anomaly (e.g. a threshold breach), automatically sends a notification (email or SMS) to a designated caregiver.
7. **The system of claim 6**, wherein the analytics function is implemented as an edge function that performs anomaly detection (e.g. >20% increase in sensor values) and uses an API (e.g. Resend) to dispatch alerts.
8. **The system of claim 6**, further including a web-based dashboard (frontend) that retrieves data from the cloud database in real-time and displays physiological trends (charts and tables) for remote monitoring.
9. **A method for wearable health monitoring**, comprising the steps of:
    - continuously measuring multiple physiological parameters from a subject using on-body sensors (including motion, pressure/weight, temperature, and pulse/$SpO_2$ sensors);
    - in an edge microcontroller, generating alerts locally if immediate thresholds are exceeded;
    - sending the collected sensor data from the microcontroller to a remote server over a network;
    - analyzing the incoming data at the server by comparing each new measurement to stored prior measurements; and
    - automatically notifying a caregiver if the analysis detects a significant anomaly or emergency condition.
10. **The method of claim 9**, wherein the analyzing step is performed by a cloud-resident serverless function that is triggered upon each new data insert, and the notification step involves sending an email or SMS alert containing the parameter that breached its threshold.

(Additional claims may define specifics of threshold values, particular sensor arrangements, or methods of secure communication as needed.)

**9. Abstract (150 words max)**

SensoGuard is a wearable IoT health companion that continuously monitors multiple vital parameters and provides real-time alerts. The device integrates an ESP8266 microcontroller with an accelerometer/gyro (MPU-6050), load cell (HX711),

body-temperature sensor (DS18B20), and pulse-oximeter (MAX30100/30102), along with an LCD and buzzer for local feedback. It securely transmits time-stamped sensor data via Wi-Fi to a cloud backend. A serverless cloud function analyzes incoming data and automatically sends email alerts if it detects anomalous changes (e.g. sudden spikes in weight or temperature). A web dashboard visualizes the live data. SensoGuard's hybrid edge-cloud design and open-source firmware provide a low-cost, scalable solution for proactive health monitoring, enabling caregivers to receive immediate notifications of potential emergencies. (Words: 119)

**10. Advantages of the Invention**

- **Continuous, multi-parameter monitoring:** Unlike episodic check-ups, SensoGuard provides 24/7 tracking of motion, weight/pressure, temperature, and pulse/$SpO_2$.
- **Real-time alerts:** Automated cloud-based analytics issue instant alerts (email/SMS) on anomalies, enabling prompt intervention.
- **Low-cost hardware:** Uses inexpensive off-the-shelf components (ESP8266 module and commodity sensors) to keep unit cost low.
- **Hybrid edge-cloud design:** Local processing handles basic alerts and conserves bandwidth, while heavy analytics run in the cloud (serverless), balancing reliability with computational efficiency.
- **Wearable form factor:** Compact, ergonomic design (e.g. wristband or clip-on) suitable for daily wear by elderly or patients, with plug-and-play assembly.
- **Open-source and modular:** Firmware is based on open platforms (Arduino), allowing customization and extension.
- **Wireless connectivity:** Uses standard Wi-Fi and HTTPS/MQTT; no proprietary network or constant smartphone tethering is required.
- **Scalability:** Cloud backend can support multiple devices/users concurrently, with free-tier resources (Supabase, Vercel) to minimize recurring costs.
- **Battery efficiency:** Supports low-power modes (deep sleep) for long-term use with minimal charging.
- **Improved patient outcomes:** By shifting from reactive to proactive monitoring (from demonstration of focus in test results), SensoGuard can help prevent emergencies (falls, infections, cardiac events) through timely alerts.

**11. Applications:**

- **Elderly Fall Detection and Emergency Alerts**
  Continuous gyro + pressure fusion allows automated fall detection without requiring the user to press a button. Alerts are instantly sent to caregivers.
- **Post-Surgery Home Recovery Monitoring**
  Tracks temperature, mobility, and bed-presence to monitor post-operative patients for infection, inactivity, or unsafe movements.
- **Remote Chronic Disease Management**
  Continuous $SpO_2$, heart rate, and movement monitoring enables early detection of respiratory or cardiovascular deterioration.
- **Sleep Quality Analysis and Sleep Apnea Detection**

Load cell + motion + $SpO_2$ data provides a full sleep profile, including agitation patterns and potential apnea episodes.

- **ICU-to-Home Transition Support**
  Helps doctors remotely monitor discharged patients who still require high-frequency vital monitoring.
- **Dementia and Alzheimer's Wandering Prevention**
  Detects nighttime bed exit or unusual activity and instantly alerts caregivers.
- **Rehabilitation and Physiotherapy Tracking**
  Tracks movement intensity and patterns to assess patient progress in physical therapy routines.
- **Stress and Fatigue Monitoring**
  Combines heart rate variability, motion, and temperature to estimate physiological stress.
- **Athlete Training Load Monitoring**
  Provides real-time insights into workout intensity, body temperature elevation, and oxygen levels.
- **Remote Pediatric Monitoring (Children Monitoring)**
  Detects fever spikes, unusual movements, or sudden drops in $SpO_2$ for infants or children.
- **Occupational Safety Monitoring (Factories/Construction)**
  Detects falls, overheating, and abnormal physiological conditions of workers in hazardous environments.
- **Smart Bed / Smart Chair for Hospitals**
  Integrates with hospital beds to track patient presence, weight shifts, and vitals without additional equipment.
- **Sleep-Lab Replacement for Home Testing**
  Provides multi-sensor medical-grade data to clinicians for diagnosing sleep disorders remotely.
- **Remote Monitoring of COVID-like Symptoms**
  Tracks high temperature, low oxygen saturation, and fatigue for respiratory disease alerts.
- **Wheelchair / Bed-Bound Patient Monitoring**
  Detects repositioning needs, prevents bed sores, and monitors vitals in immobile patients.