Environmental Stress Monitoring System (ESMS)

Media Management- Project Report

Group Name: Team ESMS

Group Members:
- Dipendra Thapa
- Pritam Pramanik
- Shokhjahon Kodirov

Name- Pritam Pramanik
Semester- 3
Matriculation Number - 22401097

Submission Date: 20 January 2026

1.0 Introduction

Environmental stress is now a part of our lives. The human body and mind are greatly affected by environmental stress also. A person's internal stress level is determined by the total number of events in their environment. The most likely cause of negligent fabrication being that it is a rough aspect is due to a sudden drop in temperature or humidity spikes, noise, etc. Environmental stress will witness a complex physiological reaction in the body, which could help a variety of uses (Chrousos, 2009). In stressful cutting circumstances, the body's physiological functions may change through the hypothalamic-pituitary-adrenal response.Extreme temperatures that fail to trigger the proper working of the homoeostatic mechanism of the body can produce thermal stress and cause illness. Could lead to overheating, excessive cold or cardiac issues. Likewise, high humidity prevents the body from cooling down through sweating and prevents a high index. After that, physical aspects of noise are of great consequence since changes to the sound environment have social differences (Münzel et.al,2018). Unscreened and unmitigated incessant exposure to these environmental stressors produces cumulative physiological impairment, disease emergence, chronic hypertension, anxiety disorders, and devaluation.

2.0 Technical Solution and Architecture : ESMS

2.1 Technical Solution Overview

The Environmental Stress Monitoring System constantly monitors environmental as well as physiological parameters responsible for stress in human beings. It is a single embedded system. The technical solution is based on four sensor modules that are implemented using an Arduino microcontroller, which collects, analyzes, and processes data.It constantly observes environmental parameters such as temperature, humidity, noise, and motion, as well as a human being's stress response using his/her heartbeat in that environment.

The system architecture follows a modular design philosophy that emphasizes:

- Scalability: Additional sensors can be integrated without major architectural changes
- Real-time Processing: Continuous data acquisition with minimal latency
- Low Power Consumption: Suitable for prolonged monitoring applications
- Cost-effectiveness: Utilization of readily available, affordable components
- User Accessibility: Simple interface for data visualization and interpretation

2.2 System Architecture

The hardware architecture consists of three primary layers: the Sensor Layer, the Processing Layer, and the Interface Layer. The components used were the Central Processing Unit (Arduino Uno R3), DHT11 Temperature and Humidity Module, Sound Sensor Module, HC-SR501 PIR Motion Sensor, Heart Rate Monitor Module and Breadboard for circuit integration.I attached the DHT11, sound sensor, HC-SR501 PIR to the 5V Pin and GND. Join the DHT11 to Digital Pin 2, connect the Sound Sensor to either Analog A0 (or Digital 3) and the PIR to Digital Pin 4. The Heart Rate Module will work with the 3.3V Pin only so don't connect it to the 5V else it will be damaged. Attach the Heart Rate Sensor's signal wire to the A1 pin. To read data correctly, connect all negative pins to the common ground.

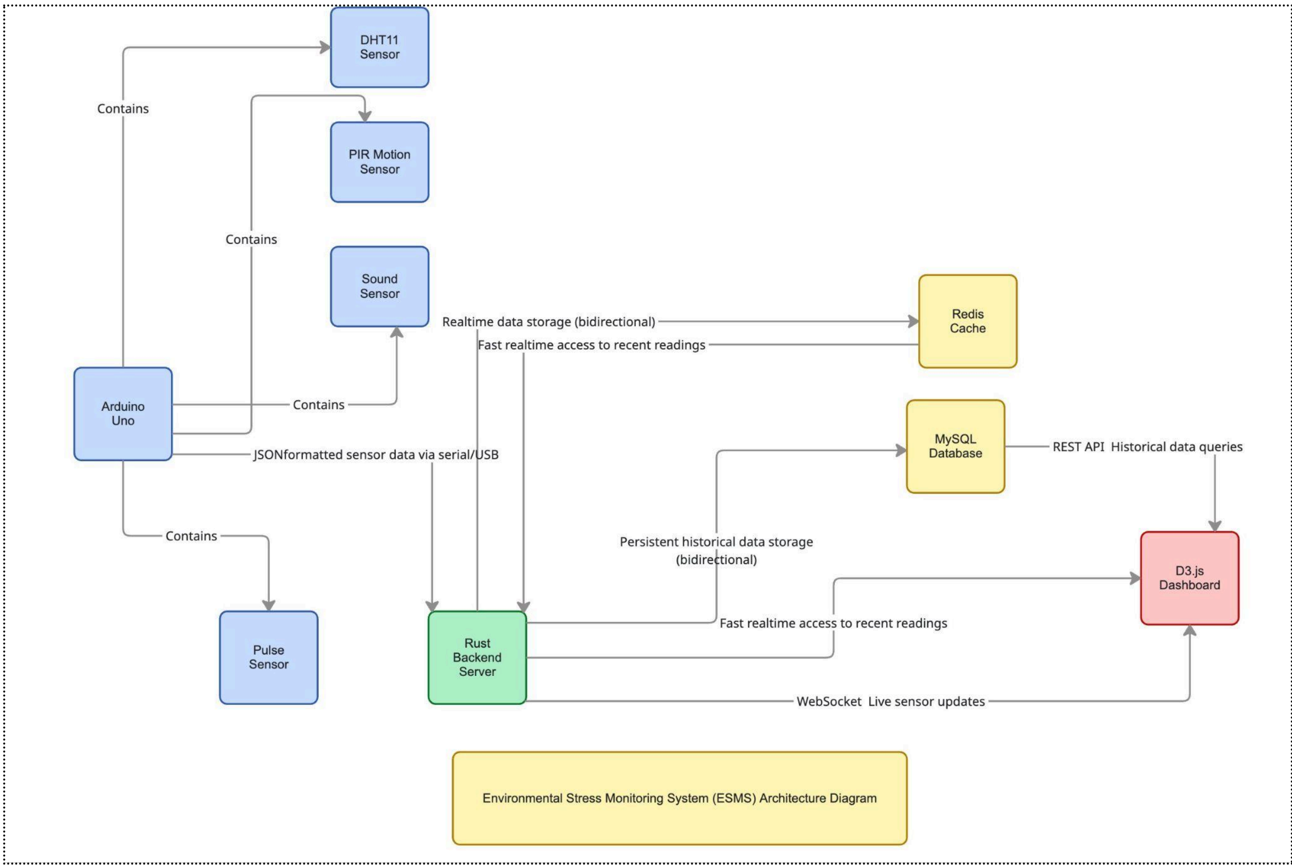

Figure 1: System architecture of ESMS showing sensors, backend (Rust), Redis, MySQL, and frontend visualization([D3.js](D3.js)).

From a hardware point of view, ESMS will incorporate environmental sensors like temperature, humidity and noise sensors. The sensors will measure and continually quantify the levels of environmental factors that are known to cause physiological stress. Furthermore, pulse rates will be recorded by the use of heart rate sensors. The essential element of the system is still this. To make the system contextually aware, the Passive Infrared (PIR) motion sensor shall be used. The motion detector is capable of detecting the existence of people and also the movement of the crowd. This helps the system to know if the time-periods are active/inactive. This is vital when interpreting heart rate data.

2.3 Data Processing Algorithm

The stress assessment algorithm evaluates environmental parameters against established comfort thresholds:

$$\text{Stress Index} = w_1 \cdot T_{norm} + w_2 \cdot H_{norm} + w_3 \cdot S_{norm} + w_4 \cdot M_{status} + w_5 \cdot HR_{norm}$$

Where:

- T_{norm} = Normalized temperature deviation from comfort range (20-26°C)
- H_{norm} = Normalized humidity deviation from comfort range (30–60%)
- S_{norm} = Normalized sound level relative to threshold (55 dB)
- M_{status} = Motion detection binary status
- HR_{norm} = Normalized heart rate deviation from resting baseline
- w_1, w_2, w_3, w_4, w_5 = Weighting coefficients

2.5 Software Architecture

The software architecture is built around an Arduino-based embedded program with two main parts: setup() and loop(). In setup(), the system initializes sensor pins, starts serial communication, and performs basic stabilization/calibration for reliable readings. In the continuous loop(), the program follows a cyclic data acquisition model: it reads inputs from the DHT11 (temperature/humidity), sound sensor, PIR motion sensor, and heart rate module, then applies simple filtering and threshold-based decision logic to interpret conditions. Finally, the processed results (raw values and stress indication) are output through the Serial Monitor for real-time monitoring and logging.

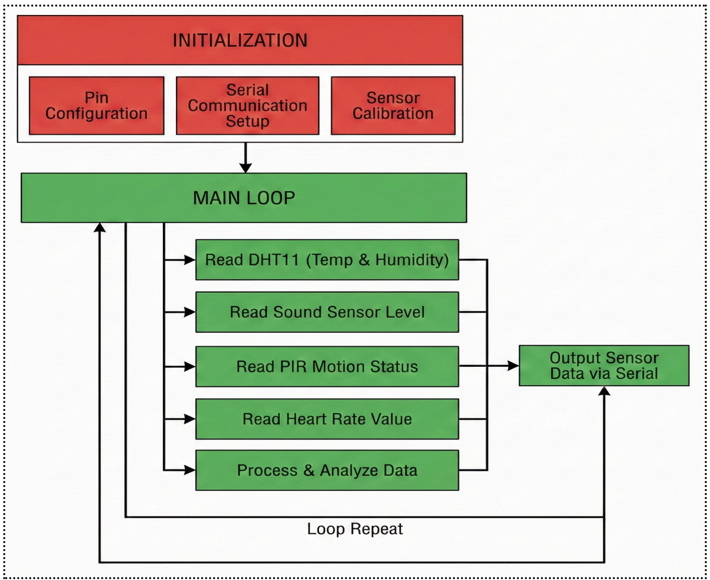

Figure 2: Software Architecture

The ESMS frontend is developed using HTML, CSS and JavaScript. It is bundled using Vite (a TypeScript-based configuration) for fast development and building in production. This technology chooses ESMS well where the dashboard must update and remain responsive while multiple live indicators must be rendered. When JavaScript is run in a browser, it can communicate with the backend REST APIs with less overhead. Vite enables fast hot-reload while in development mode and bundles efficiently during deployment.

The file index.html defines the structure of our dashboard (i.e. panels for live sensor values, trend graphs, and correlation/stress section). The file styles.css guarantees a clean readable layout with clear visual states (normal/warning/high). The file app.js drives real-time behavior. It gets data from the sensor, validates/formats it for display, and updates the charts and indicators on a fixed refresh cycle. The new modifications focus on the correlation with the physiological signal values by combining heart-rate trends with environmental and motion reading. So that the interface does not show only the raw values but interpretable ones e.g. heart rate rises when noise spikes and motion are reused often. This frontend stack is good for ESMS as it is light-weight, widely compatible, easy to extend and good for visualization and interactivity in real-time.

3.0 Results and Analysis incl. Interpretation

In all circumstances subjected to testing, environmental conditions had clear relationships with heart-rate activity. Most elevated temperature periods were likely to elevate heart rate and the effect of elevated temperature on heart rate was higher when humidity was also high. The trend in cardiovascular system activity is consistent with physiological thermoregulation. It occurs to support heat dissipation. In a similar vein, short-term heart-rate peaks occurred during high ambient noise levels, a typical stress-related arousal response.

Including the PIR motion sensor allowed us to better interpret heart-rate changes that were probably due to the environment as opposed to physical activity. An increase in heart rate was detected on motion indicating it was due to motion and not due to stress. Higher heart rate when the horse is moving little or not at all gives more power to suggest that environmental factors- heat, humidity and noise- should be involved in the interpretation.

The accompanying figure shows the relationship of heart rate with environmental variables (temperature, noise, etc; shaded regions indicate intervals with PIR-detected motion. The graph reveals that higher heart-rate values tend to appear during high-temperature and high-noise sections, and the motion shading helps to suggest whether the rise is likely activity-driven or environment-driven.

We notice similar patterns with stress monitors in the area. These suggest unfavourable conditions can have an impact on health. In isolation, heart rate is not a conclusive indicator of stress exposure. However, the conjunction of environmental readings, motion context and physiological response may offer a better approximation of stress-like conditions.

There are a few limitations. The system lacks consideration for individual differences (fitness level, baseline heart rate, and personal stress sensitivity), which limits generalization across users. Moreover, there may be variations and measurement noise due to the smaller observation period and sensor calibration constraints. In spite of these weaknesses, the findings indicate the technical feasibility and conceptual validity of ESMS as a multi-sensor approach to characterising stress-relevant environmental conditions.

4.0 Individual Responsibility, learning and challenges

My main role in the group was to work on the frontend dashboard of ESMS. I developed the UI that visualizes sensor data including environment temperature, humidity, noise and heart rate and other physiological data coming in from the sensors. Part of my job was to make sure that the display not only showed the values but also that they would not be misinterpreted, for instance, by expressing trends clearly and pointing out how physiology depends on the environment. I also worked on the integration of the frontend with the backend through data retrieval via REST-based methods, so that the interface consistently updated with new readings.

From this role, I learned how to build a more structured web application where the UI is based on live data instead of static content. I gained insights into structuring front-end code to deal with frequent updates, using short-term buffers, and making sure the rendered visible output corresponds to the latest reading. I also gained experience setting out how data would be presented as part of design. In other words, layout, labeling, and chart type all influence if a user can quickly see correlation. Working with multi-sensor information like a PIR motion sensor, I have become aware of context in interpreting physiological measurements.

A major difficulty is the visualization and interpretation in real time. The application should not freeze while charts are updated frequently. In other words, we had to juggle DOM updates with the rewriting of our data. The sensor streams may or may not arrive at the same time and again may have different sampling rates. This affects the ability to consistently display correlations between heart rate and the environmental variables. I had to ensure the UI was still clear and not cluttered. Showing too many variables at once lowers clarity. Overall, the challenges I faced led me to design features that are practical and user-focused to convey complex sensor data.