VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi – 590014.

Internship Report

On

"SMART HEALTHCARE MONITORING SYSTEM"

Submitted in partial fulfillment of the requirement for the award of degree of

BACHELOR OF ENGINEERING

In

INFORMATION SCIENCE AND ENGINEERING

&

ELECTRONICS AND COMMUNICATION ENGINEERING

By

Jyothsna S (1AP21IS013)

Shrusti B (1AP21IS031)

Shashidhar S (1AP21EC016)

2023 - 2024

A P S COLLEGE OF ENGINEERING

Anantha Gnana Gangothri,

NH-209, Kanakapura Road, Somanahalli, Bengaluru-560116.

Evaluation Sheet

Title of the Project: Smart Healthcare Monitoring System

Name of the Students: Jyothsna S (1AP21IS013)

Shrusti B (1AP21IS031)

Shashidhar S (1AP21EC016)

External Supervisor:

Internal Supervisor:

Date:

Place:

Project Completion Certificate

I, **JYOTHSNA S** (USN : 1AP21IS013), here by declare that the material presented in the Project Report titled **"SMART HEALTHCARE MONITORING SYSTEM"** represents original work carried out by me in the **Department of Information Science and Engineering** at the **APS college of Engineering, Bangalore** during the tenure **2 October, 2024 – 12, December, 2024**.

With My signature, I certify that:

- I have not manipulated any of the data or results.
- I have not committed any plagiarism of intellectual property and have clearly indicated and referenced the contributions of others.
- I have explicitly acknowledged all collaborative research and discussions.
- I understand that any false claim will result in severe disciplinary action.
- I understand that the work may be screened for any form of academic misconduct.

Date: **Student Signature:**

In my capacity as the supervisor of the above-mentioned work, I certify that the work presented in this report was carried out under my supervision and is worthy of consideration for the requirements of the B.Tech. Internship Work.

Advisor's Name:Dr Shivamurthaiah M **Guide Name:** Akhil Sai

Advisor's Signature **Guide Signature**

Project Completion Certificate

I, **SHRUSTI B** (USN : 1AP21IS031), here by declare that the material presented in the Project Report titled **"SMART HEALTHCARE MONITORING SYSTEM"** represents original work carried out by me in the **Department of Information Science and Engineering** at the **APS college of Engineering, Bangalore** during the tenure **2 October, 2024 – 12, December, 2024**.

With My signature, I certify that:

- I have not manipulated any of the data or results.
- I have not committed any plagiarism of intellectual property and have clearly indicated and referenced the contributions of others.
- I have explicitly acknowledged all collaborative research and discussions.
- I understand that any false claim will result in severe disciplinary action.
- I understand that the work may be screened for any form of academic misconduct.

Date: **Student Signature:**

In my capacity as the supervisor of the above-mentioned work, I certify that the work presented in this report was carried out under my supervision and is worthy of consideration for the requirements of the B.Tech. Internship Work.

Advisor's Name:Dr Shivamurthaiah M **Guide Name:** Akhil Sai

Advisor's Signature **Guide Signature**

Project Completion Certificate

I, **SHASHIDHAR S** (USN : 1AP21EC016), here by declare that the material presented in the Project Report titled **"SMART HEALTHCARE MONITORING SYSTEM"** represents original work carried out by me in the **Department of Electronics and Communication Engineering** at the **APS college of Engineering, Bangalore** during the tenure **2 October, 2024 – 12, December, 2024**.

With My signature, I certify that:

- I have not manipulated any of the data or results.
- I have not committed any plagiarism of intellectual property and have clearly indicated and referenced the contributions of others.
- I have explicitly acknowledged all collaborative research and discussions.
- I understand that any false claim will result in severe disciplinary action.
- I understand that the work may be screened for any form of academic misconduct.

Date: **Student Signature:**

In my capacity as the supervisor of the above-mentioned work, I certify that the work presented in this report was carried out under my supervision and is worthy of consideration for the requirements of the B.Tech. Internship Work.

Advisor's Name: Dr Prakash Jhadhav **Guide Name:** Akhil Sai

Advisor's Signature **Guide Signature**

INDEX

CHAPTER 1

ABSTRACT

The Smart Healthcare Monitoring System is an advanced, integrated solution that combines the capabilities of a Raspberry Pi, a variety of health sensors, and an intuitive web interface to provide real-time monitoring of a patient's vital signs. This system continuously tracks important health parameters, including heart rate, blood pressure, body temperature, and oxygen saturation levels, ensuring that healthcare providers have up-to-date information. The collected data is displayed both on a local LCD screen and a web dashboard, enabling easy access for medical staff or caregivers. Additionally, the system is equipped with a relay mechanism that can be activated in case of emergency, such as when any of the vital signs exceed predefined critical thresholds. By automating the monitoring process and alerting healthcare professionals to any anomalies, the system aims to improve patient care, enhance the accuracy of health assessments, and increase the efficiency of emergency response. This innovative approach to healthcare monitoring ensures that patients are consistently observed, and potential issues can be addressed promptly, ultimately contributing to better health outcomes and peace of mind for both patients and caregivers.

CHAPTER 2

INTRODUCTION

Efficient healthcare monitoring is critical, especially in emergencies, as it can make the difference between life and death. The Smart Healthcare Monitoring System utilizes Internet of Things (IoT) technology to continuously collect, analyze, and track vital signs, ensuring that health data is always available when needed most. The Raspberry Pi serves as the central hub, connecting with various sensors that measure vital signs such as heart rate, blood pressure, temperature, and oxygen saturation. It then relays this information to a web application, offering real-time updates and insights into a patient's condition. This system is especially beneficial in scenarios involving remote health monitoring or in critical care settings where constant supervision is necessary. By automating the monitoring process, the system enables timely interventions, improves accuracy, and reduces human error in health assessments.

2.1 Objective

The primary objective of this project is to develop an automated healthcare monitoring system that continuously tracks health parameters, provides real-time alerts during emergencies, and allows for remote supervision via a web application. The system aims to offer healthcare providers a reliable, efficient, and user-friendly tool to monitor patients' health remotely, reducing the burden of manual checks and ensuring that immediate action can be taken when necessary.

2.2 Problem Statement

Traditional healthcare monitoring methods often rely on manual checks, which can lead to delays in detecting health issues or responding to emergencies. These delays can be particularly dangerous in critical care or for patients with chronic conditions. This project addresses the pressing need for an automated, real-time monitoring solution that ensures constant surveillance of a patient's vital signs. By integrating sensors with a Raspberry Pi and leveraging IoT technology, the system minimizes human intervention, reducing response times and enhancing the effectiveness of healthcare management, especially in remote or emergency situations.

CHAPTER 3

APPLICATION

- Continuous and Real-Time Monitoring

  - 24/7 Monitoring of Vital Health Parameters: The system tracks critical health metrics such as blood pressure, heart rate, oxygen levels, and blood sugar levels at all times, ensuring that any abnormal patterns are identified promptly.

  - Early Detection of Health Issues: By continuously monitoring health indicators, the system helps detect early signs of health abnormalities, such as elevated blood pressure or fluctuating heart rates, which can trigger immediate alerts for timely intervention.

- Timely Medical Intervention

  - Immediate Alerts for Critical Changes: When abnormal readings are detected, alerts are triggered that notify the patient or their healthcare provider, enabling quick action to prevent health emergencies.

  - Prevention of Complications: Early identification of health concerns allows for prompt interventions, potentially preventing severe complications or health emergencies that may otherwise be irreversible.

- Support for Chronic Disease Management

  - Continuous Tracking for Chronic Conditions: Patients with chronic conditions such as hypertension, diabetes, and heart disease benefit from ongoing monitoring to ensure their vital statistics remain within healthy ranges.

  - Empowering Patients and Healthcare Providers: The system helps both patients and healthcare providers track progress, adjust treatment plans as necessary, and make more informed decisions about long-term care.

- Remote Monitoring and Accessibility

  - Remote Access for Patients and Healthcare Providers: Through a web interface, patients, caregivers, and healthcare professionals can access real-time health data from anywhere, making healthcare more accessible and timely.

  - Improved Access for Remote Areas: The system is particularly beneficial for individuals in rural or underserved regions, where access to traditional healthcare facilities may be limited, ensuring continuous monitoring without the need for frequent hospital visits.

- Cost-Effectiveness and Scalability

  - Affordable and Accessible Healthcare: The system's low-cost design makes it accessible to a wide range of users, from individuals managing their own health to clinics and hospitals monitoring large patient populations.

  - Scalable for Different Healthcare Settings: Whether for personal home use or integration into broader healthcare networks, the system is adaptable to various environments, providing flexibility and scalability.

- Proactive Healthcare for Individuals

  - Empowering Patients to Take Control: The system allows individuals to monitor their own health and receive alerts when necessary, giving them more control over their wellbeing.

  - Personalized Health Insights: By providing continuous feedback, the system enables patients to adjust their lifestyle or treatment plans in real-time, improving long-term health outcomes.

- Streamlining Healthcare for Medical Professionals

    - Reducing Workload for Healthcare Providers: The system automates the process of health monitoring, reducing the need for manual check-ups and allowing medical professionals to focus on more urgent patient needs.

    - Efficient Patient Monitoring: With real-time access to patient data, healthcare professionals can track multiple patients at once, improving the efficiency and quality of care.

- Improved Healthcare Delivery for Facilities

    - Managing Large Patient Populations: Healthcare facilities can leverage this system to monitor and manage a large number of patients efficiently, providing better care without needing additional staff or resources.

    - Higher Standards of Care: The system enhances the overall quality of healthcare by ensuring that patients receive continuous monitoring, resulting in more timely and accurate treatments.

CHAPTER 4

COMPONENTS

4.1 HARDWARE COMPONENTS

- Raspberry Pi:

Fig 4.1.1 : Raspberry pi

The Raspberry Pi is a single-board computer that acts as the central controller of the system. It processes data, manages sensor interactions, and runs the Flask server for the web-based monitoring application. It integrates with all connected sensors and enables remote access to health data through a web interface.

- LCD Display (16x2):

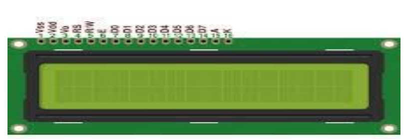


Fig 4.1.2: LCD Display(16*2)

The 16x2 LCD display is used to show health data such as temperature, heart rate, and oxygen saturation in real-time. It provides local, immediate feedback on vital parameters, allowing caregivers to monitor the patient's health without accessing the web interface.

- Relay Module:

Fig 4.1.3: Relay Module

The relay module controls external devices such as alarms or actuators based on system logic. When abnormal health readings are detected, it triggers alerts or activates devices like alarms, ensuring quick response in emergencies.

- GPIO Pins:

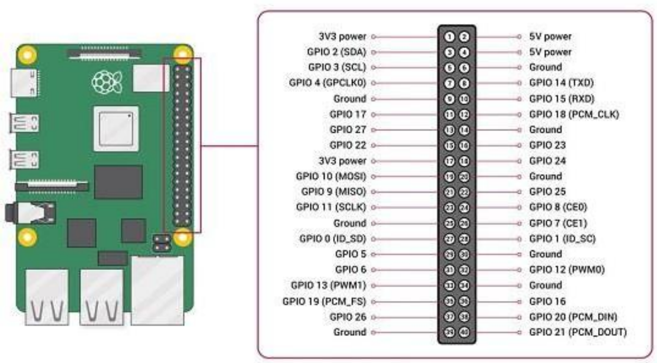


Fig 4.1.4: GPIO Pins

General Purpose Input/Output (GPIO) pins on the Raspberry Pi allow the system to interface with external components like sensors, relays, and the LCD display. These pins manage sensor data collection and control outputs like alarms or health monitoring displays.

- Power Supply:

  - The power supply ensures a stable 5V DC input to the Raspberry Pi and all connected components, ensuring uninterrupted operation. A consistent power source is essential for the proper functioning of the system.

- Temperature Sensor (Simulated):

  - The temperature sensor (such as DHT11 or a simulated version) measures the patient's body temperature. It alerts the system if the temperature is outside normal ranges, helping detect fever or potential infections. Simulated data allows testing even without physical sensors.

- Heart Rate and Blood Pressure Simulator:

  - This simulator generates data to mimic real heart rate and blood pressure readings, allowing the system to function without actual sensors. It helps test the system's response to different health conditions.

- Oxygen Saturation Simulator:

  - This simulator mimics oxygen saturation (SpO2) data, helping monitor respiratory health. It ensures the system can track and alert for abnormal oxygen levels, which may indicate issues like hypoxia.

4.2 SOFTWARE COMPONENTS

- Raspberry Pi OS:

  - Raspberry Pi OS is a Debian-based operating system that supports the necessary software libraries and frameworks for running the healthcare monitoring system. It provides a stable environment for managing hardware and running applications.

- Python Programming:

  - Python is used to interface with sensors, process data, and control the system’s logic. It handles the core functions like sensor data collection, relay control, web-based interface management, and alert processing.

- Flask Framework:

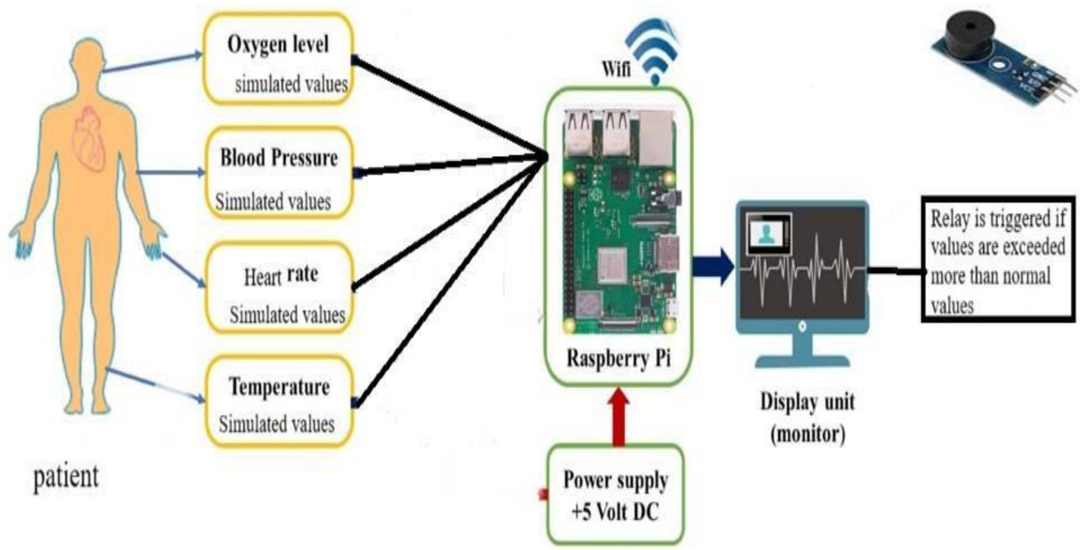
  - Flask is a lightweight Python framework that hosts the web-based monitoring interface. It serves the health data remotely, allowing users to access real-time updates through a web browser from anywhere with an internet connection.

- Webpage (HTML, CSS, JavaScript):

  - The webpage is built using HTML for structure, CSS for styling, and JavaScript for dynamic content updates. It displays live health data and allows users to interact with the system remotely, ensuring a responsive and user-friendly interface.

CHAPTER 5

FLOW CHART



The image represents a patient health monitoring system using a Raspberry Pi. It collects simulated values for oxygen levels, blood pressure, heart rate, and body temperature from the patient. The Raspberry Pi processes this data and sends it to a display unit (monitor) via WiFi. If any of the values exceed normal thresholds, a relay is triggered to alert medical personnel. The system is powered by a +5V DC power supply.

CHAPTER 6

CONCLUSION

The Smart Healthcare Monitoring System provides an innovative solution to real-time health monitoring, offering a cost-effective and scalable approach for both personal and clinical use. By leveraging Raspberry Pi, the system simulates and monitors vital parameters such as temperature, heart rate, blood pressure, and oxygen saturation, displaying these readings on both an LCD and a web interface. It ensures timely intervention by triggering alerts via a Relay Module when abnormal readings are detected.This system bridges the gap between traditional periodic health checks and continuous monitoring, enabling earlier detection of potential health issues. The cost-effectiveness of Raspberry Pi and open-source software ensures accessibility, particularly for those in resource-limited environments, while the system's scalability allows it to be deployed in both personal and clinical settings.By providing continuous, real-time health tracking, the system enhances healthcare accessibility and enables timely intervention, particularly for individuals with chronic conditions or those in remote areas. It improves health outcomes by detecting abnormalities early, reducing the risk of complications, and enabling remote monitoring by healthcare professionals.In essence, the Smart Healthcare Monitoring System offers a comprehensive solution for proactive healthcare, improving efficiency, accessibility, and outcomes, especially in settings where traditional healthcare infrastructure is limited.

CHAPTER 7

FUTURE WORK

The Smart Healthcare Monitoring System has significant potential for future enhancements that can increase its functionality and broaden its applicability in healthcare. One potential improvement is the integration of medical-grade sensors to provide precise, reliable health measurements, critical for professional healthcare settings. These sensors could monitor heart rate, blood pressure, temperature, and oxygen saturation, ensuring medical-standard readings.

Another development is the use of machine learning algorithms to analyze historical health data and detect patterns that may indicate future risks. For example, the system could predict heart attacks, strokes, or diabetic complications, allowing for earlier intervention and better patient outcomes. The system could also benefit from mobile app integration, allowing users to monitor their health data on-the-go, receive real-time updates, and track trends over time. Voice alerts would provide immediate notifications for abnormal readings, useful in emergencies or for those unable to interact with a screen. Emergency notifications could alert caregivers or healthcare providers when critical health changes occur, ensuring prompt attention. IoT integration could store health data in the cloud, enabling remote access by healthcare professionals and facilitating communication with other medical devices like smartwatches or glucose monitors.

Additionally, the system could be miniaturized and made wearable to provide continuous health monitoring in a more discreet manner. Wearables like smartwatches or fitness bands could track vital signs throughout the day, especially beneficial for individuals with chronic conditions or those seeking real-time health tracking.

These advancements would enhance the accuracy, convenience, and accessibility of the system, making it applicable for personal health tracking, chronic disease management, and remote care, particularly in underserved areas. By incorporating machine learning, IoT, mobile apps, and wearable devices, the system has the potential to transform healthcare, offering personalized and efficient care.

CHAPTER 8

APPENDIX

8.1 GPIO Pin Configuration

- LCD_RS (15): Register Select pin for LCD.
- LCD_E (12): Enable pin for LCD.
- LCD_D4 (23): Data pin 4 for LCD.
- LCD_D5 (21): Data pin 5 for LCD.
- LCD_D6 (19): Data pin 6 for LCD.
- LCD_D7 (24): Data pin 7 for LCD.
- RELAY_PIN (36): Controls the relay for external device management.
- SENSOR_PIN (4): Reads data from the DHT11 sensor.

8.2 APIs

- GET /api/readings

  Description: Retrieves the latest health readings from the system.

  Response:


```
{
  "heart_rate": 75,
  "systolic_bp": 120,
  "diastolic_bp": 80,
  "oxygen_saturation": 98,
  "temperature": 36.5
}
```

- GET /api/relay_status

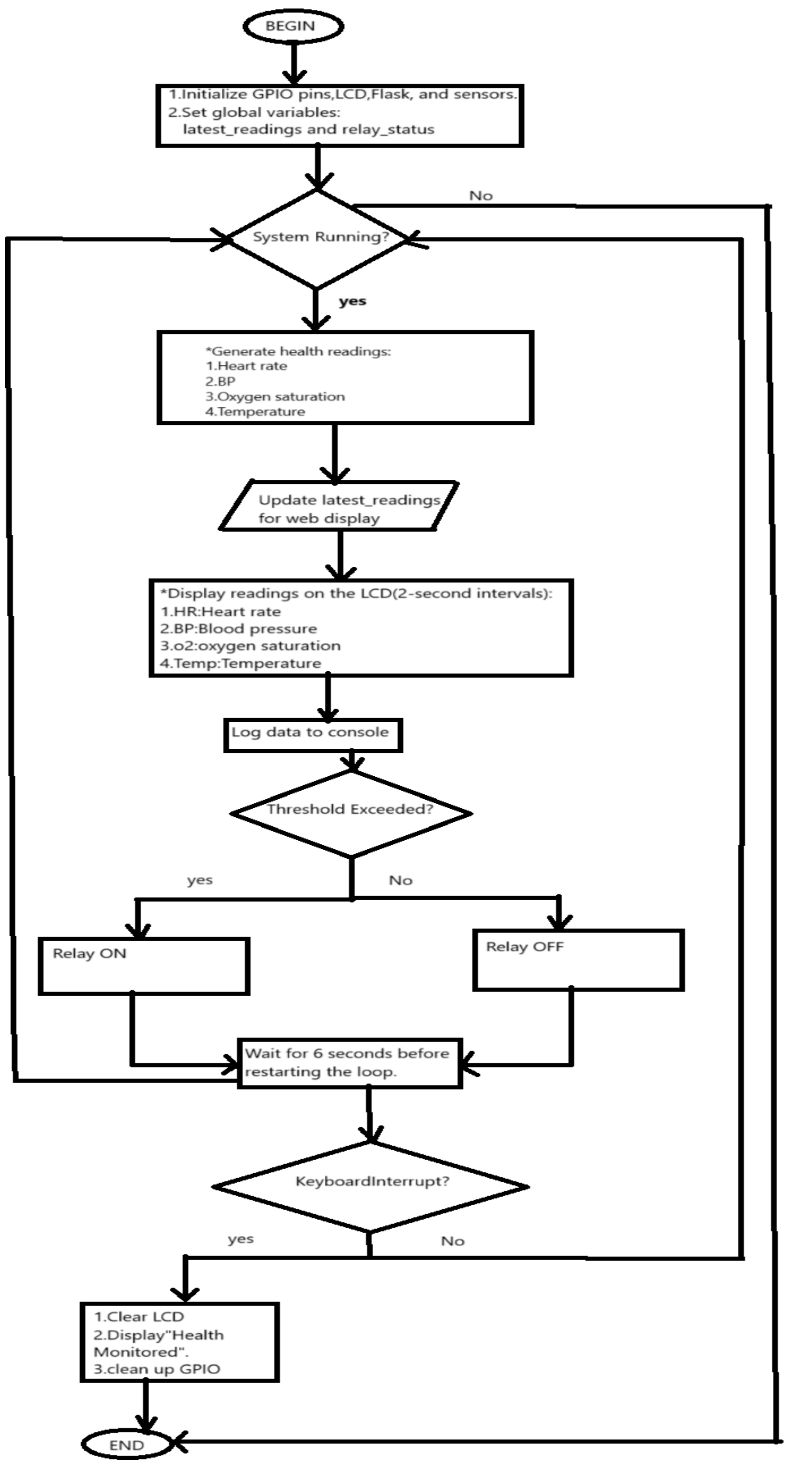
  Description: Fetches the current status of the relay.

  Response:


```
{
   "status": "ON",
   "trigger": "Temperature"
}
```

CHAPTER 9

PSEUDOCODE



```
BEGIN
    // Initialize hardware components
    Initialize GPIO pins for LCD and Relay
    Initialize LCD screen
    Initialize Flask web server
    Initialize DHT11 sensor for temperature and humidity readings

    // Global variables for latest health readings and relay status
latest_readings = {}          relay_status = {"status": "OFF",
"trigger": ""}

    // Main monitoring loop
    WHILE system is running
        // Generate real-time health readings
readings = generate_readings()
heart_rate = readings["heart_rate"]
systolic = readings["systolic_bp"]
diastolic = readings["diastolic_bp"]
 oxygen_saturation    =    readings["oxygen_saturation"]
temperature = readings["temperature"]

        // Update latest readings for web display
latest_readings.update({            "heart_rate":
heart_rate,
            "systolic_bp": systolic,
            "diastolic_bp": diastolic,
            "oxygen_saturation": oxygen_saturation,
            "temperature": temperature
        })
```

```
// Display readings on LCD screen
        DISPLAY "HR: " + heart_rate + " BPM" on LCD Line 1
        DISPLAY "BP: " + systolic + "/" + diastolic on LCD Line 2
        WAIT for 2 seconds

        DISPLAY "O2: " + oxygen_saturation + "%" on LCD Line 1
        DISPLAY "Temp: " + temperature + "°C" on LCD Line 2
        WAIT for 2 seconds

        // Log readings to console
        PRINT "Current Health Readings:"
        PRINT "  Heart Rate: " + heart_rate + " BPM"
        PRINT "  Blood Pressure: " + systolic + "/" + diastolic + " mmHg"
        PRINT "  Oxygen Saturation: " + oxygen_saturation + "%"
        PRINT "  Temperature: " + temperature + "°C"
        PRINT "******"

        // Control relay based on health thresholds         control_relay(temperature,
systolic, diastolic, heart_rate, oxygen_saturation)

        WAIT for 6 seconds  // Pause before the next reading
    END WHILE
    // Handle KeyboardInterrupt for cleanup
    ON KeyboardInterrupt
        CLEAR LCD display
        DISPLAY "Health Monitored" on LCD Line 1
        CLEAN UP GPIO pins
        PRINT "Program stopped by user"
    END ON
END
```

Explanation:

1. Initialization:
    - Set up GPIO pins for LCD and relay.
    - Initialize the LCD and Flask web server to handle the web interface.
    - Define global variables to store health readings and relay status.
2. Main Monitoring Loop:
    - Continuously generate random health readings for heart rate, blood pressure, oxygen saturation, and temperature.
    - Update the latest readings and display them on the LCD screen.
    - Log the readings to the console.
    - Control the relay based on health thresholds (if temperature, blood pressure, heart rate, or oxygen saturation exceeds a certain limit).
    - Pause for a set period before the next cycle of reading and displaying data.
3. Relay Control:
    - The system checks the generated readings against predefined thresholds and activates the relay if necessary (e.g., for high temperature, blood pressure, heart rate, or low oxygen saturation).
4. Graceful Exit:
    - On KeyboardInterrupt (e.g., when the program is stopped by the user), the system clears the LCD display, cleans up GPIO pins, and prints a message indicating that the program has been stopped.