


IOT & ML BASED **HEALTH MONITORING WEB APP**

**UNDER THE GUIDANCE OF
MR.T.SREEKANTH
(ASST.PROFESSOR,RGUKT ONGOLE)**

**PRESENTED BY
SHAIK ABDUL ILAHI (RO200249)
K. VENI GEETHIKA (RO200101)
K. TRESHIKA SHAILU (RO201081)
M. SIVA DURGA (RO200966)**



AGENDA

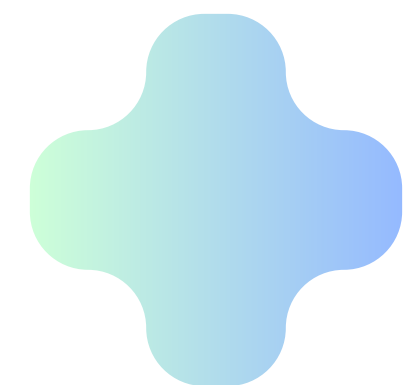
- Abstract
 - Introduction
 - Literature Review
 - Software and Hardware Requirements
 - Existed System and Proposed System
 - Objective and Scope of the Project
 - Architecture of the project
 - Algorithms Used and How it is working
 - Sample outputs and Graphs
 - Advantages
 - Disadvantages
 - Conclusion
 - Future Scope
 - References
- 

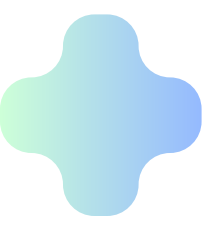
ABSTRACT

- This project presents a real-time health monitoring system integrating IoT, machine learning, and generative AI.
- An ESP32 microcontroller gathers vital signs:
Heart rate, SpO₂, and temperature.
- Data is served over a local Wi-Fi network via an onboard web server.
- A Streamlit web app:
Fetches sensor data.
Predicts health status using a Random Forest model.
Generates a personalized health report using Google Gemini API.
- Reports include AI-driven insights and can be downloaded as text file.
- The system is portable, low-cost, and ideal for home, rural, and eldercare monitoring.

INTRODUCTION

- Periodic health checkups delay early detection and intervention.
- This project offers a real-time monitoring solution using ESP32 and sensors for heart rate, SpO₂, and temperature.
- Data is shared via a local network and analyzed through a Streamlit app with a Random Forest model for health prediction.
- Google Gemini API generates an AI-based health report, downloadable as a text file.
- The system is compact, low-cost, and suited for remote, home, and rural healthcare.





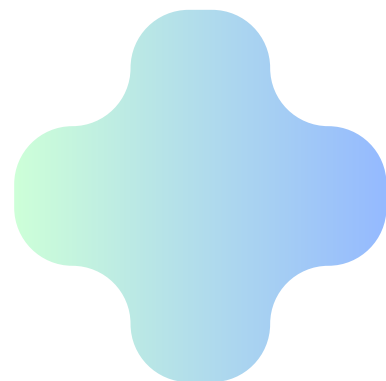
Literature Review

- IoT in healthcare enables real-time monitoring via sensor-based systems, improving early detection and patient management.
- Machine learning models, especially Random Forest, effectively predict health conditions from physiological data with high accuracy.
- Generative AI tools like Google Gemini enhance healthcare by generating personalized, easy-to-understand reports.
- This project builds on these advancements to create an intelligent, real-time health monitoring and diagnostic system.




Hardware Requirements

- ESP32 Development Board.
- MAX30100 Sensor
- DS18B20 Temperature Sensor
- OLED Display
- Buzzer (3–12V)
- Jumper Wires (Male–Male, Male–Female)
- Breadboard / PCB
- USB Cable (Micro-USB/Type-C)
- Power Supply (Battery/Power Bank/Adapter)
- Laptop
- Mouse






Software Requirements

- Arduino IDE
 - Python (3.8 or later)
 - ML Model
 - VS Code
 - Browser (Chrome/Firefox)
 - Jupyter Notebook
 - Requests
 - Streamlit
 - Pickle
 - Google Generative AI (Gemini API)
 - Scikit-learn
- 



Existed System

- Health monitoring is typically manual using separate devices for heart rate, SpO₂, and temperature.
 - No real-time data transmission or continuous tracking.
 - Diagnosis relies heavily on periodic hospital visits or physical consultations.
 - No integration of AI for personalized analysis or prescriptions.
 - Difficult for elderly, bedridden, or rural populations to access timely medical insights.
 - Health reports, if available, are generated manually and not instantly downloadable or shareable.
- 

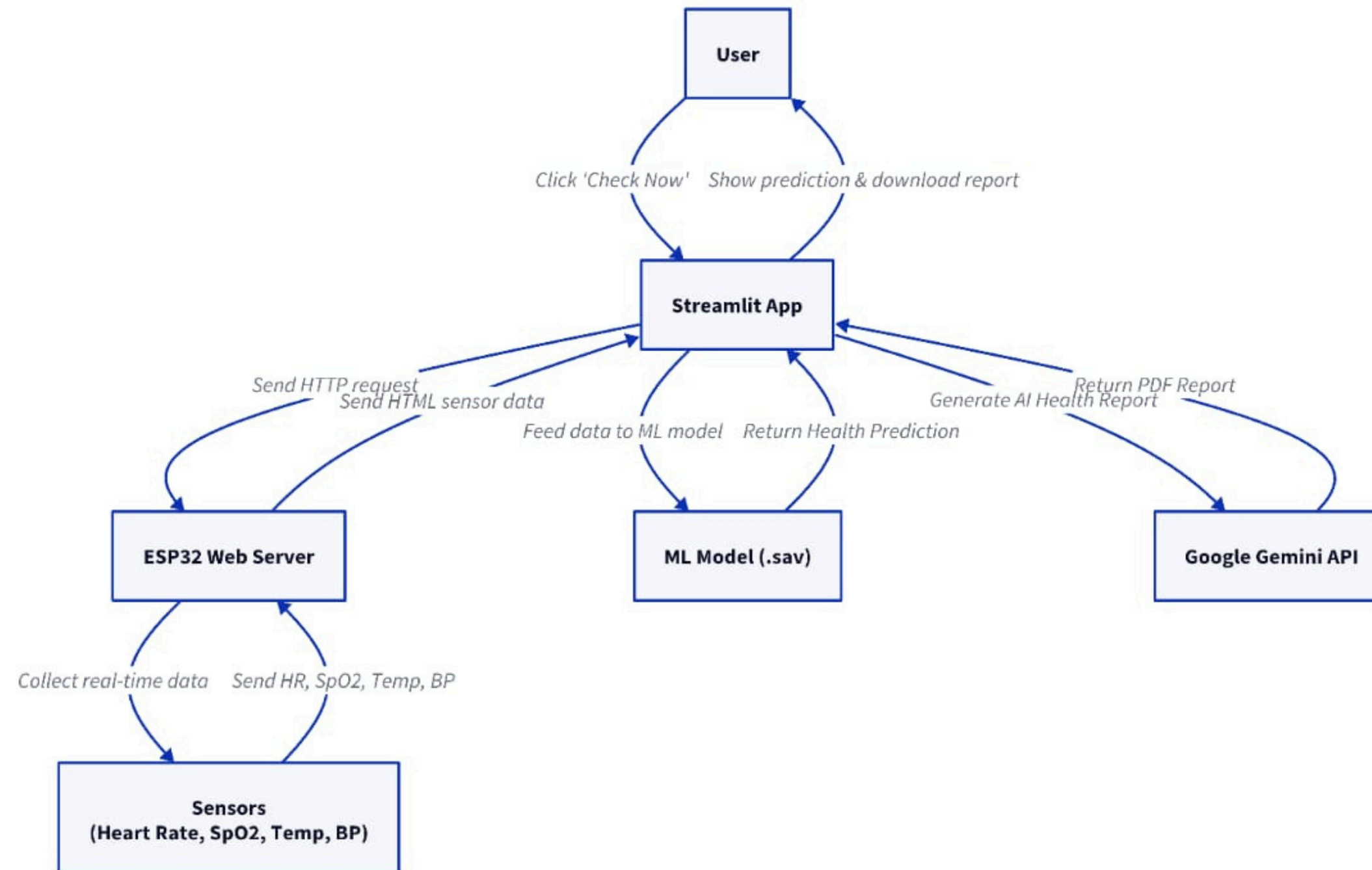
Proposed System

- Uses ESP32 and biomedical sensors to monitor health vitals in real-time.
- Hosts a local web server on ESP32 to transmit data wirelessly.
- A Streamlit-based web application fetches and displays sensor data dynamically.
- A trained Random Forest Classifier predicts whether the user is "Healthy" or "Unhealthy".
- Google Gemini API generates an AI-based, human-readable health report.
- Users can download the report as a text file with one click.
- Portable, low-cost, and ideal for home monitoring, rural clinics, and eldercare.
- Easily expandable to include more sensors and cloud/mobile integration in the future.

Objectives

- To monitor heart rate, SpO₂, and temperature in real-time using ESP32 and sensors.
- To display live health data through a user-friendly Streamlit web app.
- To predict health status using a trained Random Forest ML model.
- To generate AI-based health reports using Google Gemini API.
- To allow easy text file download of personalized reports.
- To offer a low-cost, portable solution for remote and home-based health monitoring.

System Architecture





Algorithms used and its working

1. Random Forest Classifier (Machine Learning)

Purpose:

To classify the user's health status as either "Healthy" or "Unhealthy" based on real-time vitals (heart rate, SpO₂, temperature).

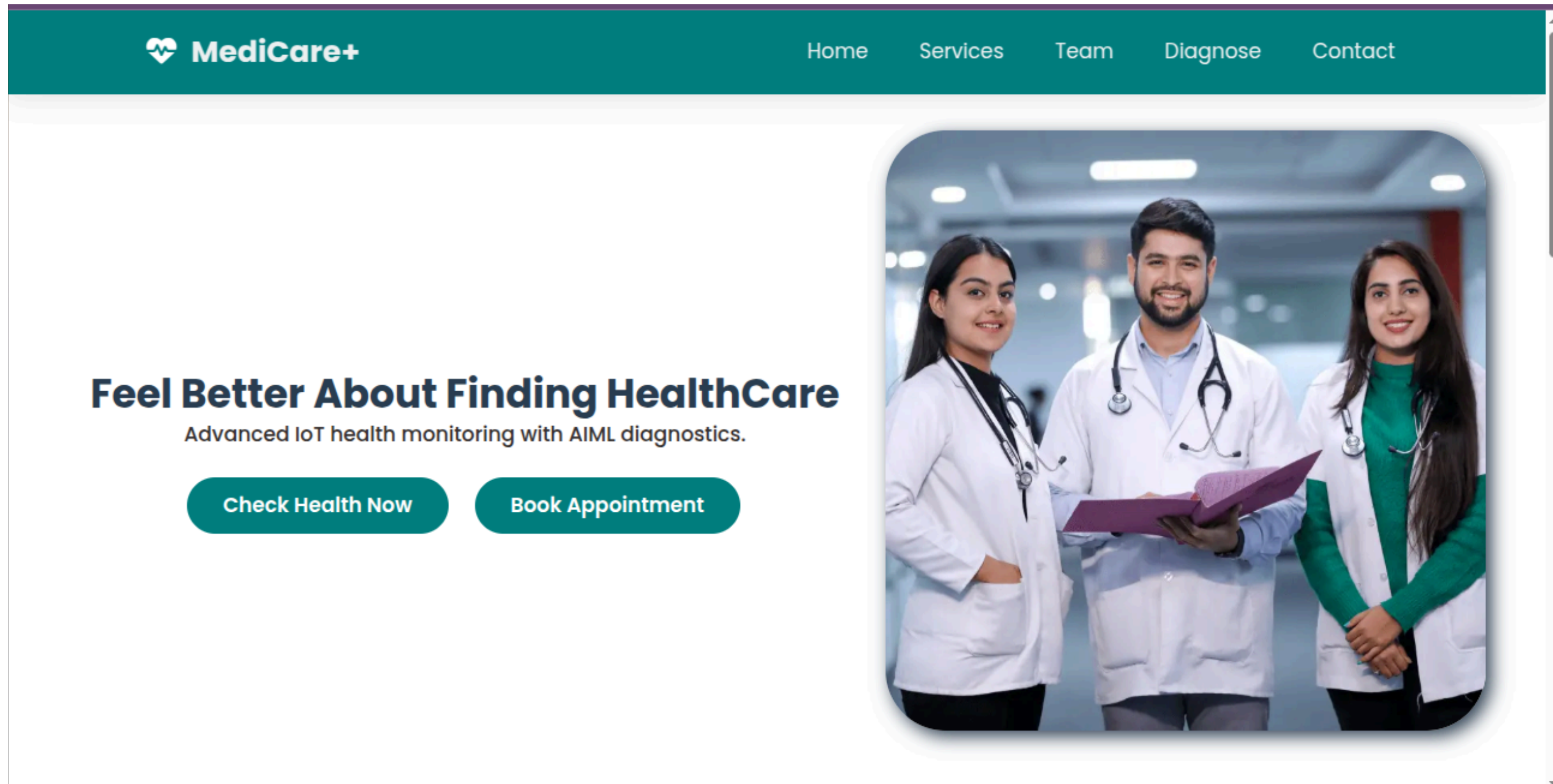
Working:

- The model is trained on a labeled dataset containing normal and abnormal vital sign values.
- It uses an ensemble of decision trees to make predictions.
- Each tree gives a classification, and the majority vote decides the final output.
- Offers high accuracy, handles missing/noisy data well, and avoids overfitting.



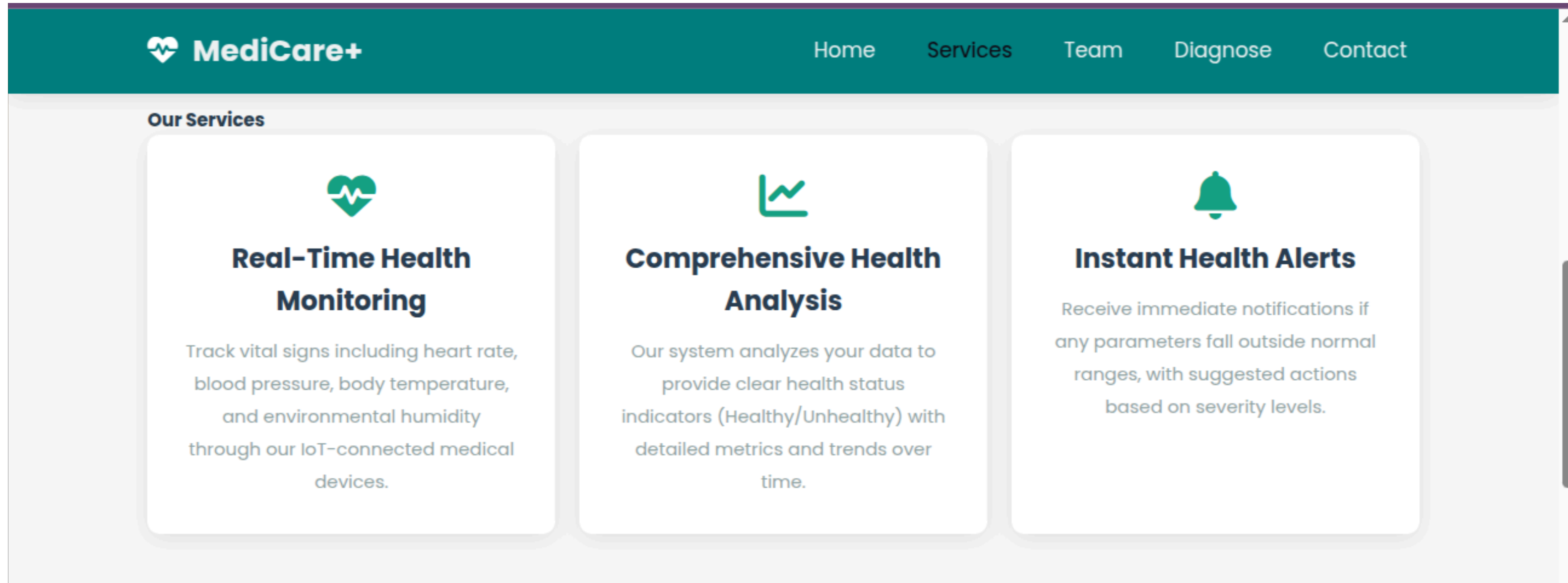
Sample Outputs

1.Home Page



Sample Outputs


2.Our Services



The screenshot displays the MediCare+ website's 'Our Services' section. The header is a dark teal bar with the MediCare+ logo on the left and navigation links (Home, Services, Team, Diagnose, Contact) on the right. Below the header, the 'Our Services' title is followed by three white service cards with rounded corners and soft shadows. Each card features a green icon, a bold title, and a descriptive paragraph. The first card has a heart with a pulse line icon and describes real-time health monitoring. The second card has a line graph icon and describes comprehensive health analysis. The third card has a bell icon and describes instant health alerts.


MediCare+ Home Services Team Diagnose Contact

Our Services




Real-Time Health Monitoring

Track vital signs including heart rate, blood pressure, body temperature, and environmental humidity through our IoT-connected medical devices.



Comprehensive Health Analysis

Our system analyzes your data to provide clear health status indicators (Healthy/Unhealthy) with detailed metrics and trends over time.



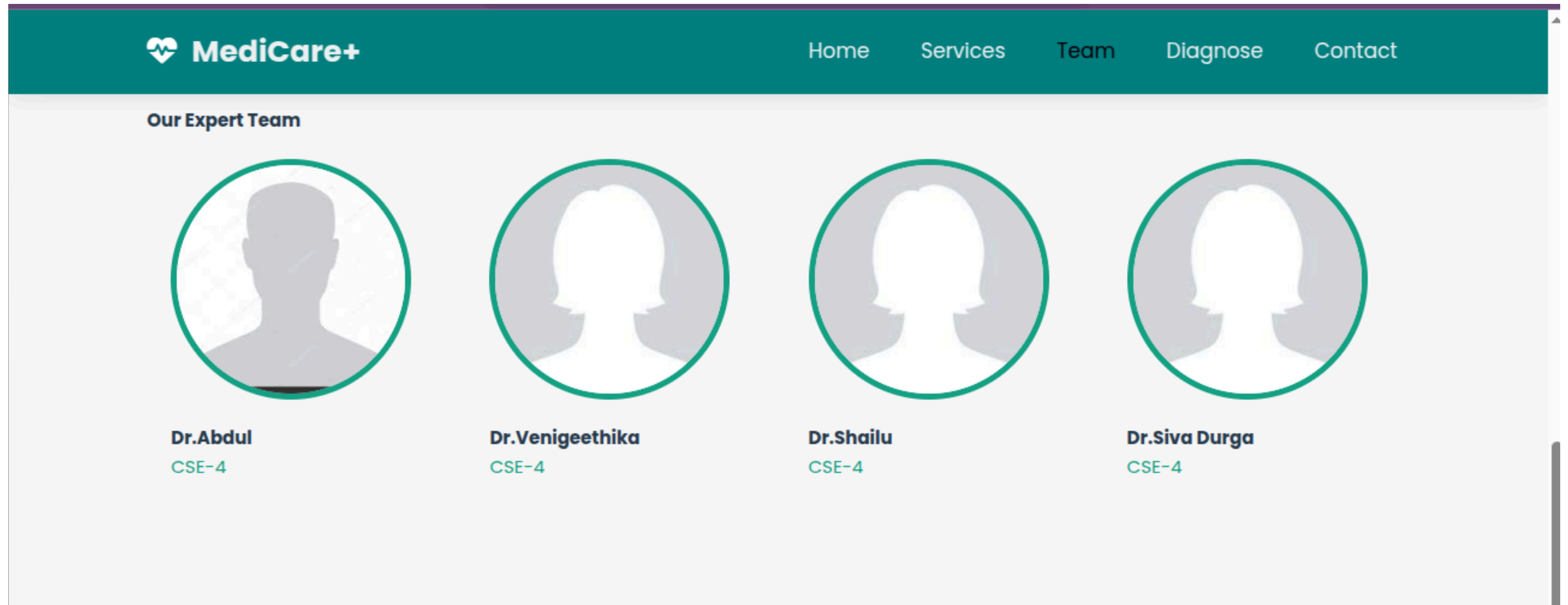
Instant Health Alerts

Receive immediate notifications if any parameters fall outside normal ranges, with suggested actions based on severity levels.



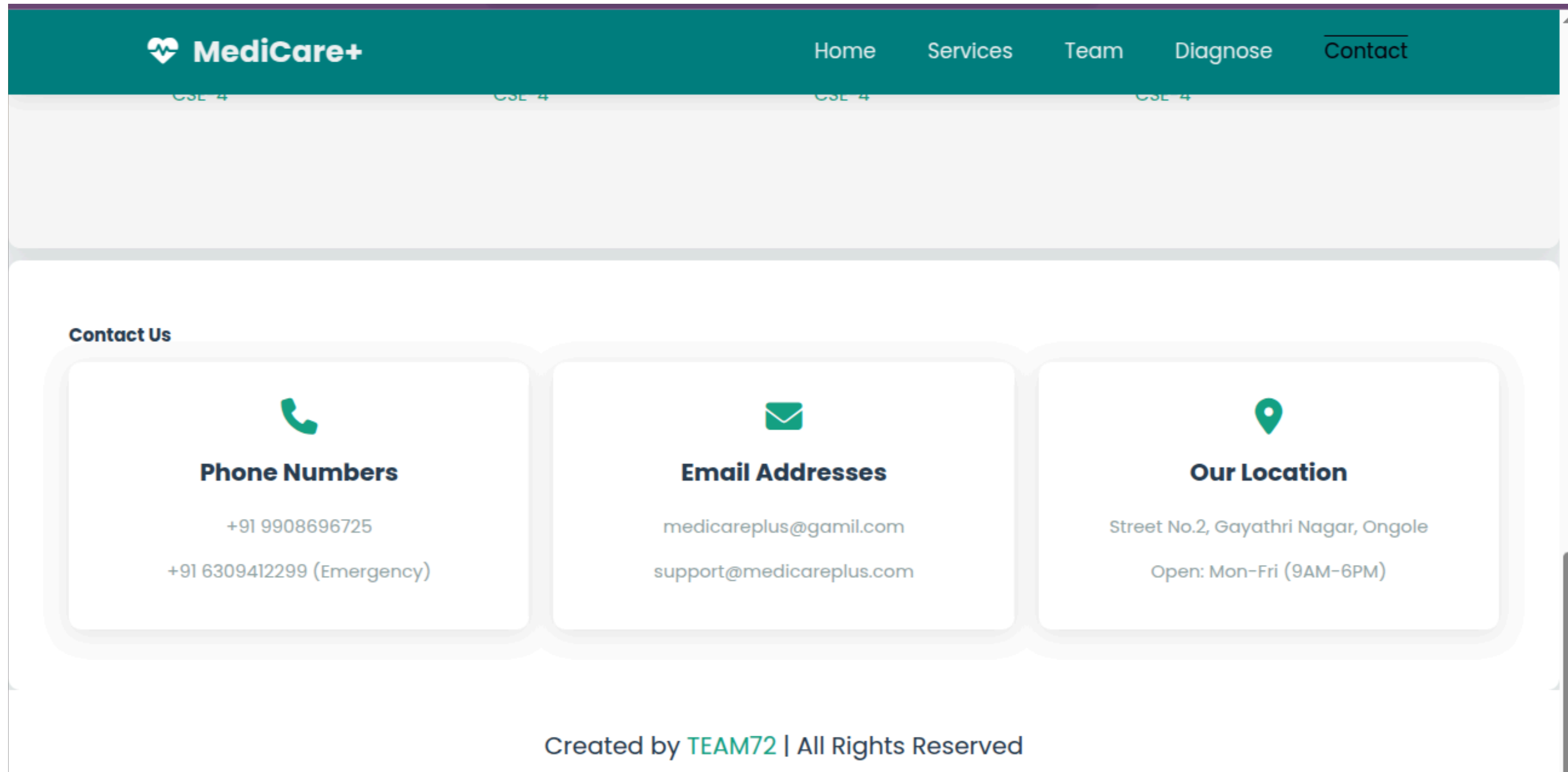
Sample Outputs

3.Team



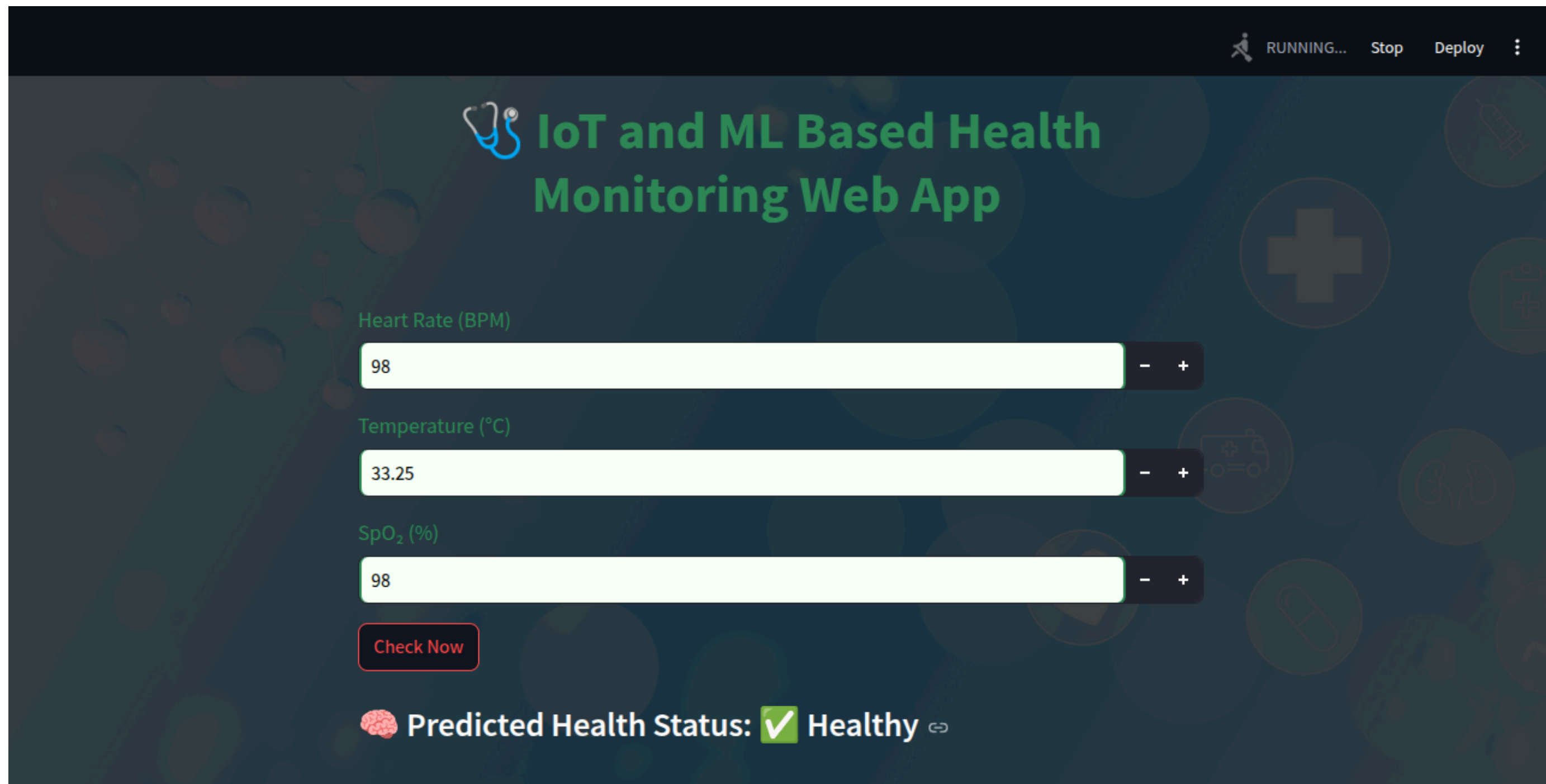
Sample Outputs

4. Contacts



Sample Outputs

5. Health values



Sample Outputs

6. Health Report



The screenshot shows a web application interface for a "Health Report". At the top right, there is a "Deploy" button and a menu icon. The main heading is "Health Report" with a document icon. Below this, the "Vitals:" section lists four items: Heart Rate: 98 BPM, Temperature: 33.25 °C (91.85 °F), SpO₂: 98%, and Reported Health Status: Healthy. The "Summary:" section contains a paragraph explaining that while the health status is "Healthy" and SpO₂ is excellent, the low body temperature (33.25°C) is a cause for concern, possibly indicating hypothermia. The "Advice:" section states that immediate medical attention is required due to the low body temperature. The "Recommendations:" section is partially visible at the bottom. The background features a dark blue gradient with faint medical icons like a cross, ambulance, and pills.

Deploy

Health Report

Vitals:

- Heart Rate: 98 BPM
- Temperature: 33.25 °C (91.85 °F)
- SpO₂: 98%
- Reported Health Status: Healthy

Summary:

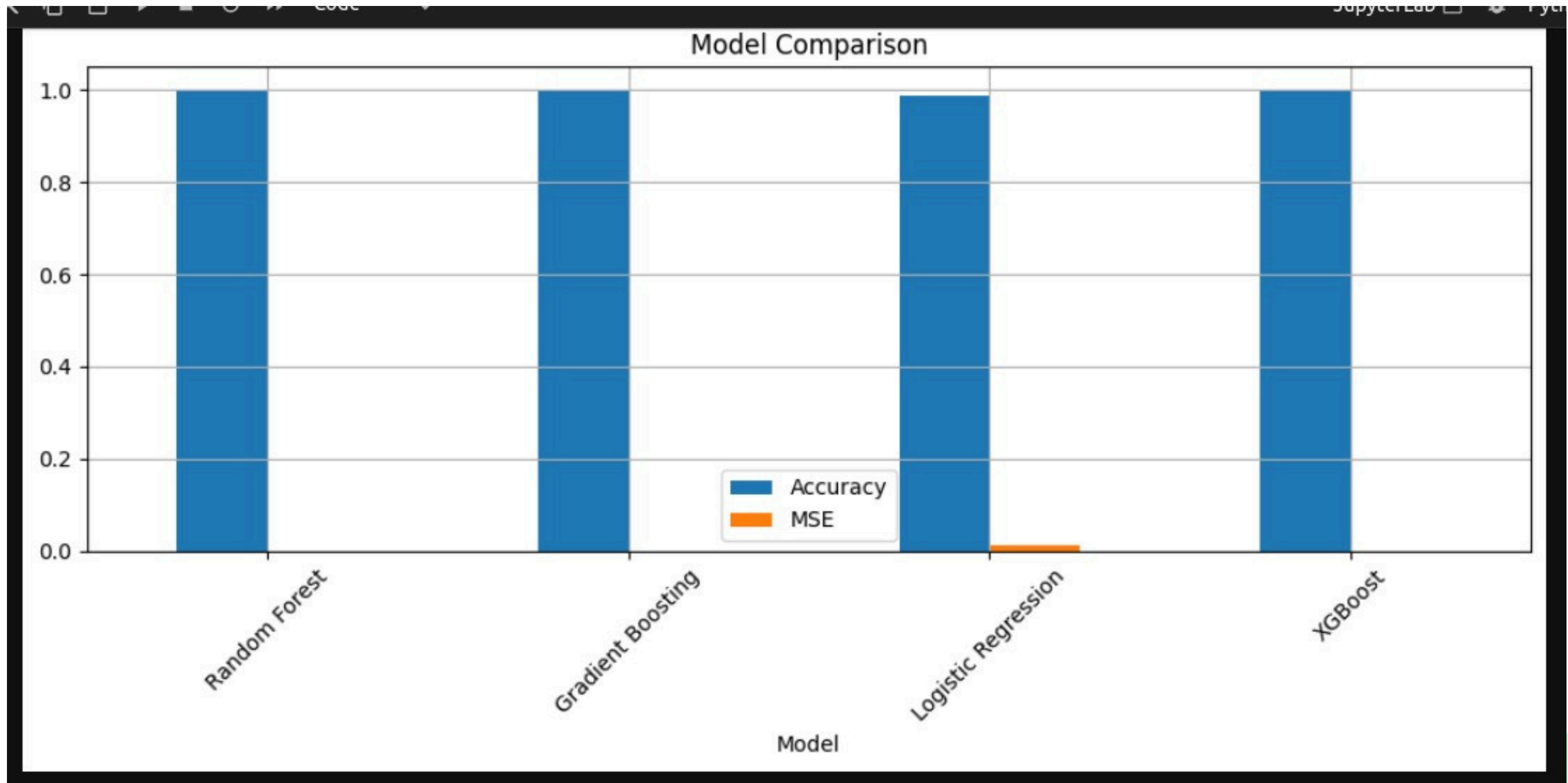
While the reported health status is "Healthy" and SpO₂ is excellent, the body temperature of 33.25°C is significantly below the normal range (36.5-37.5°C or 97.7-99.5°F). This is a cause for serious concern and could indicate hypothermia or another underlying medical condition. The slightly elevated heart rate might be a compensatory mechanism for the low body temperature.

Advice:

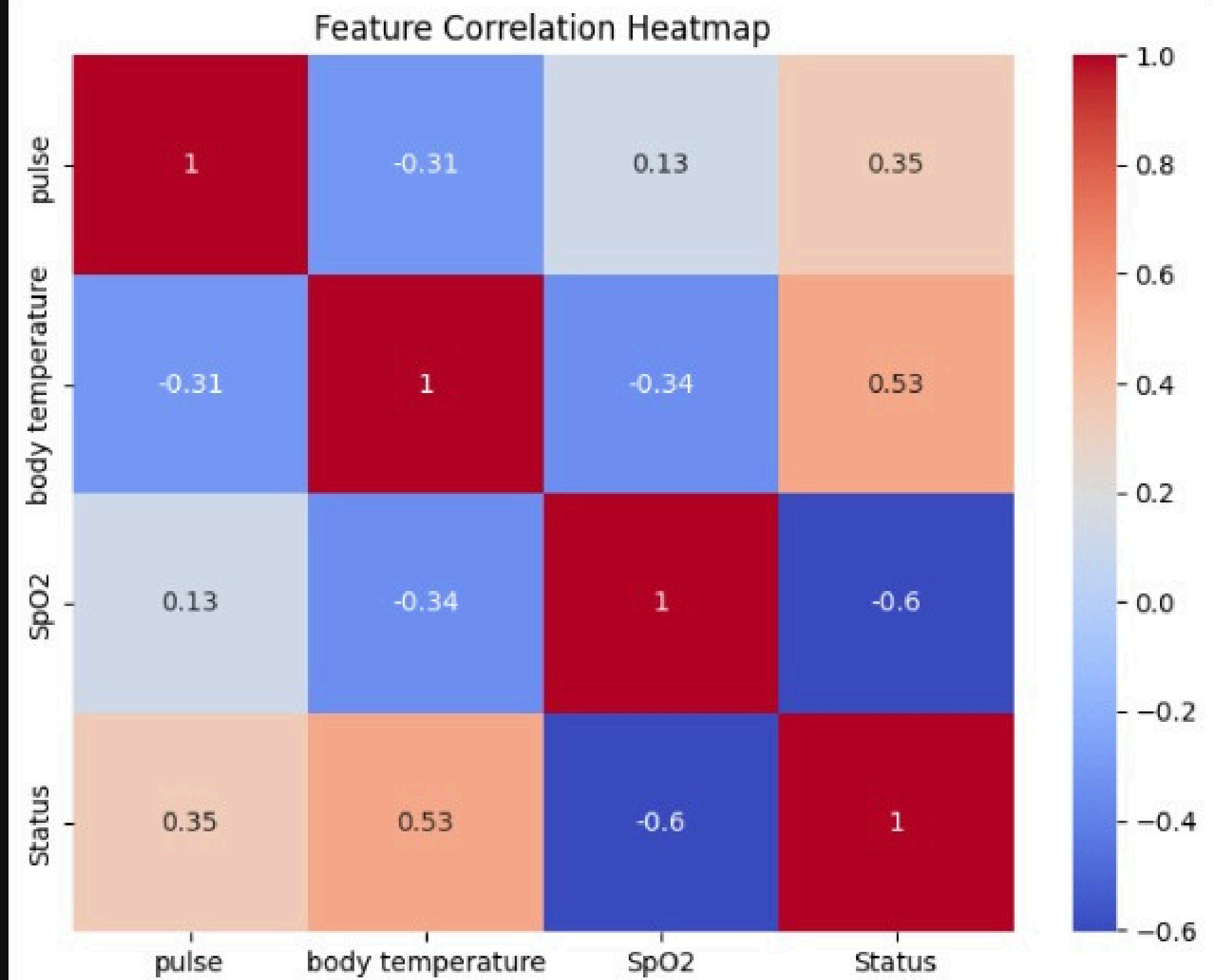
Immediate medical attention is required. This low body temperature is not normal and needs to be addressed urgently. Seek professional help immediately at a hospital or emergency clinic.

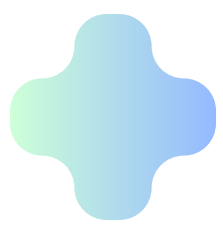
Recommendations:

Graphs



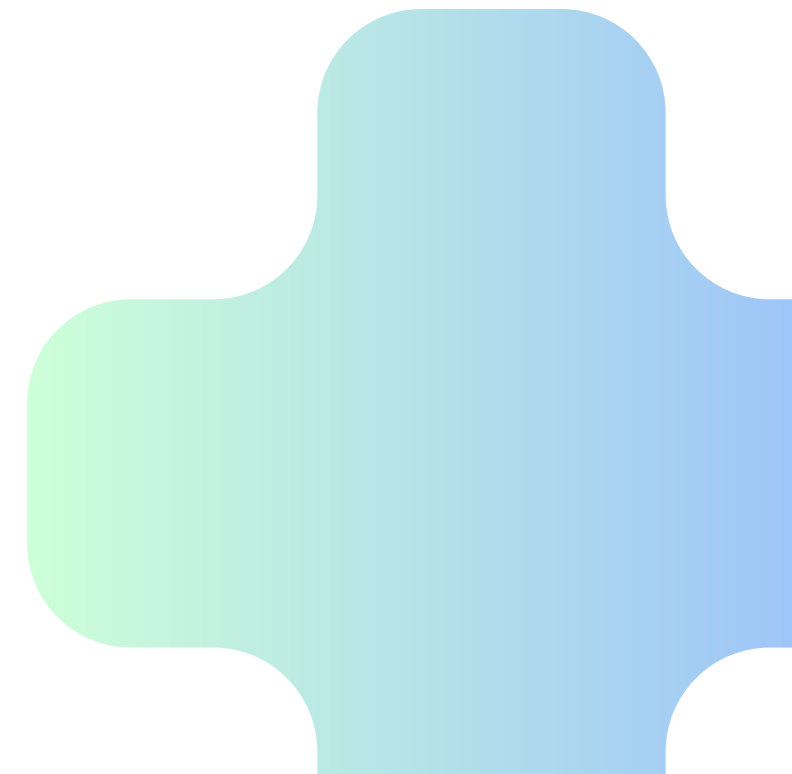
Graphs

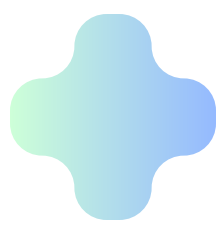




Advantages

- Real-time monitoring of vital signs (heart rate, SpO₂, temperature).
- AI-powered diagnosis using ML and Google Gemini.
- Works offline on a local network – no internet required.
- User-friendly dashboard via Streamlit web app.
- Portable and cost-effective – ideal for remote or home use.
- Auto-generated PDF health reports for easy sharing.

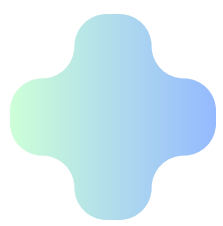




Disadvantages

- Sensor readings may lack clinical accuracy.
- No remote/cloud access by default.
- Accuracy depends on correct sensor placement.
- Requires basic technical knowledge for setup.
- No built-in alert system (SMS/email) in current version.





Conclusion

This project combines IoT, machine learning, and generative AI to create a smart, real-time health monitoring system. Using ESP32 and biomedical sensors, it tracks vital signs and predicts health status through a Random Forest model. The Streamlit web app provides live data visualization and generates personalized health reports using Google Gemini, which can be downloaded as PDFs. This low-cost, portable solution supports timely intervention and is ideal for remote, home, and rural healthcare applications.



Future Scope

- Cloud Integration: Extend data storage and access through platforms like Firebase or AWS.
- Emergency Alerts: Add SMS or email notifications for critical health readings.
- Sensor Expansion: Include additional sensors for ECG, BP, glucose, etc.
- Mobile App: Develop a companion app for better accessibility and user engagement.
- Telemedicine Integration: Connect with doctors for real-time consultations based on live vitals.

References

- [1] scikit-learn documentation: <https://scikit-learn.org/>
- [2] Google Generative AI: <https://ai.google.dev/>
- [3] Streamlit Docs: <https://docs.streamlit.io/>
- [4] ChatGPT: <https://chatgpt.com/>
- [5] Kaggle Datasets: <https://www.kaggle.com/datasets>
- [6] Random Forest Classifier:
<https://www.datacamp.com/tutorial/random-forests-classifier-python>
- [7] MAX30100 Pulse Sensor Datasheet
- [8] ESP32 Wi-Fi and Web Server Examples from Espressif
- [9] Research articles on IoT-based healthcare systems



Thank You

