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, SpO_2, 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, SpO_2, 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, SpO_2, 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.

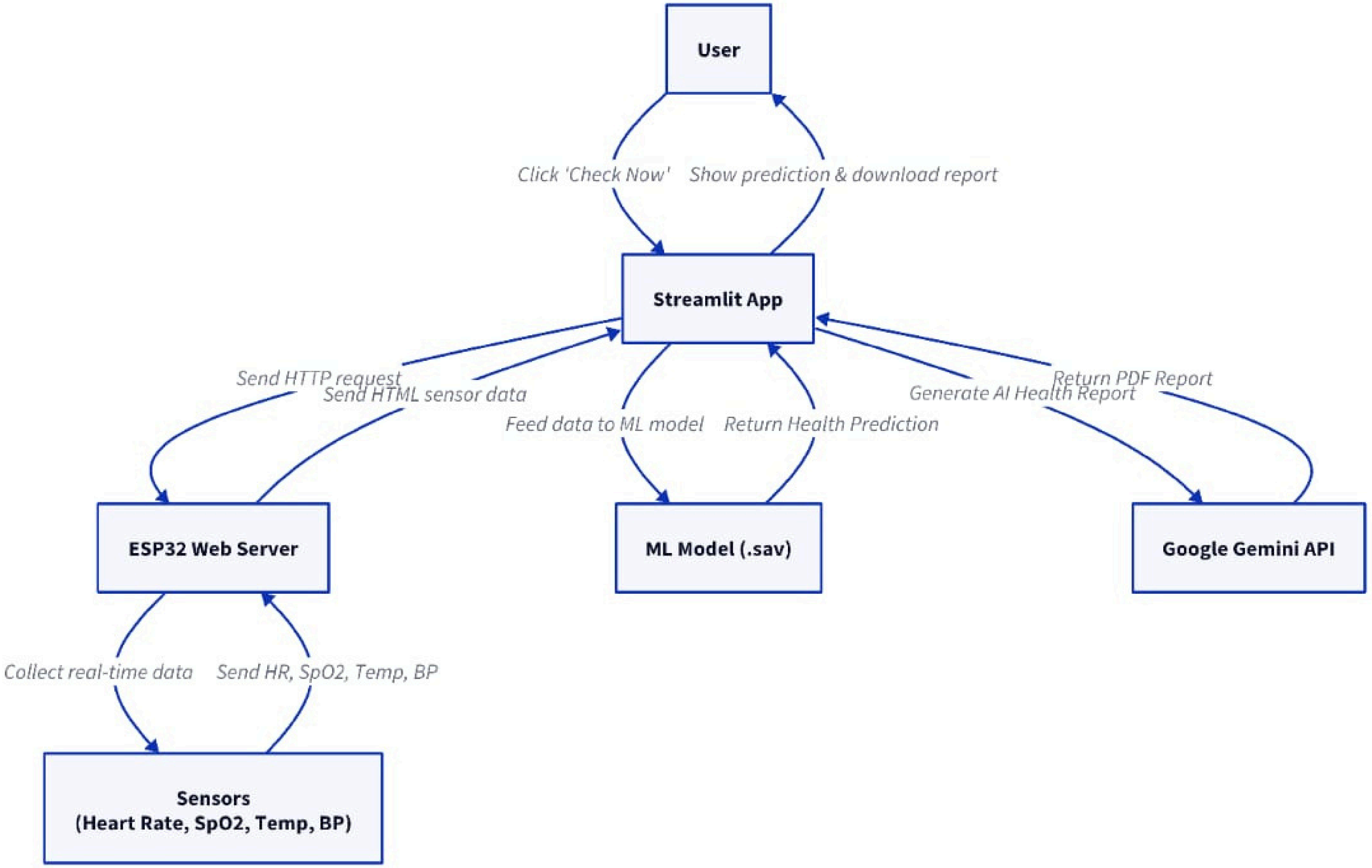
Proposed System

- 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, SpO_2, 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:

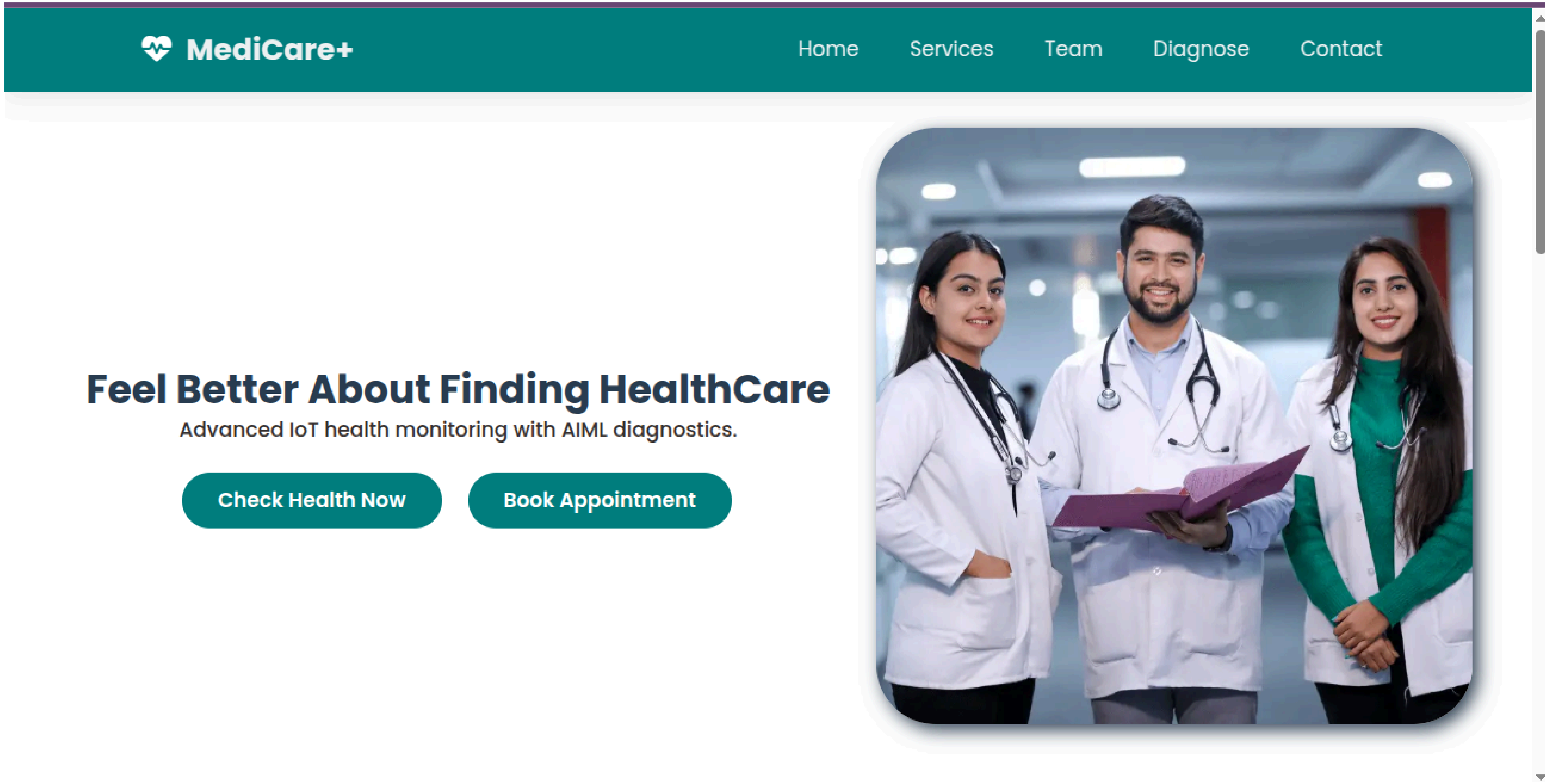
To classify the user's health status as either "Healthy" or "Unhealthy" based on real-time vitals (heart rate, SpO_2, temperature).

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.

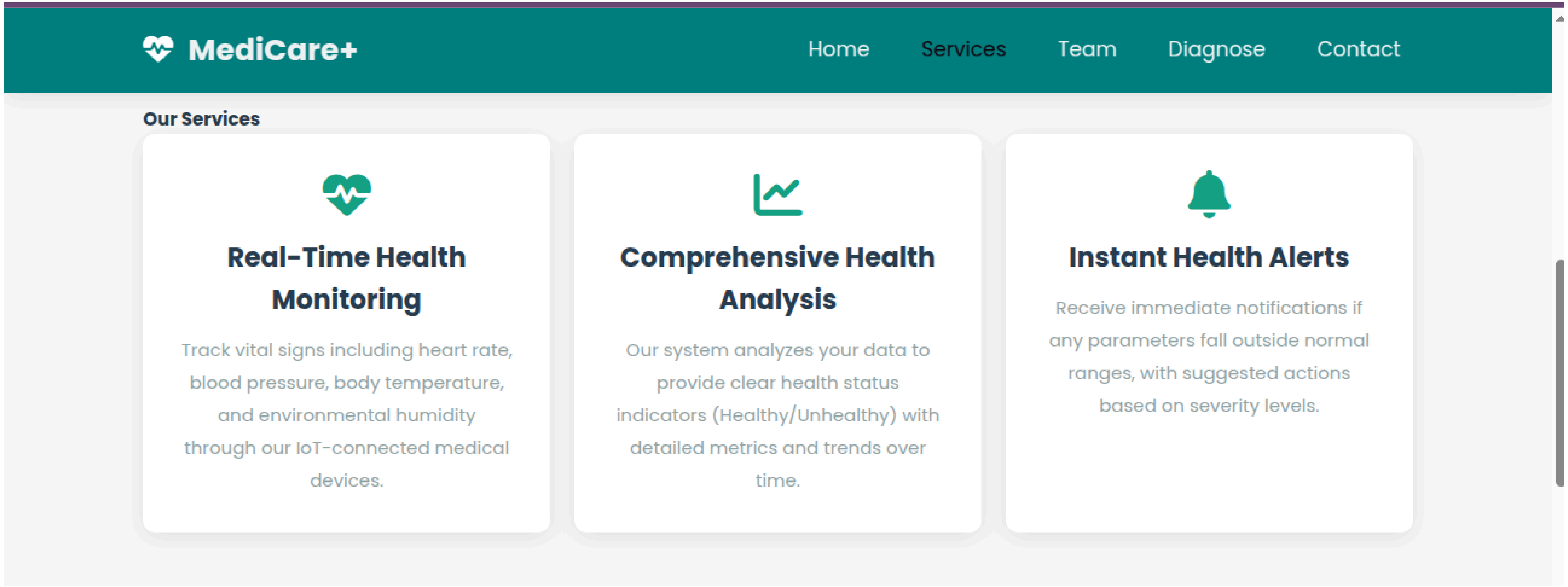
Sample Outputs

1.Home Page



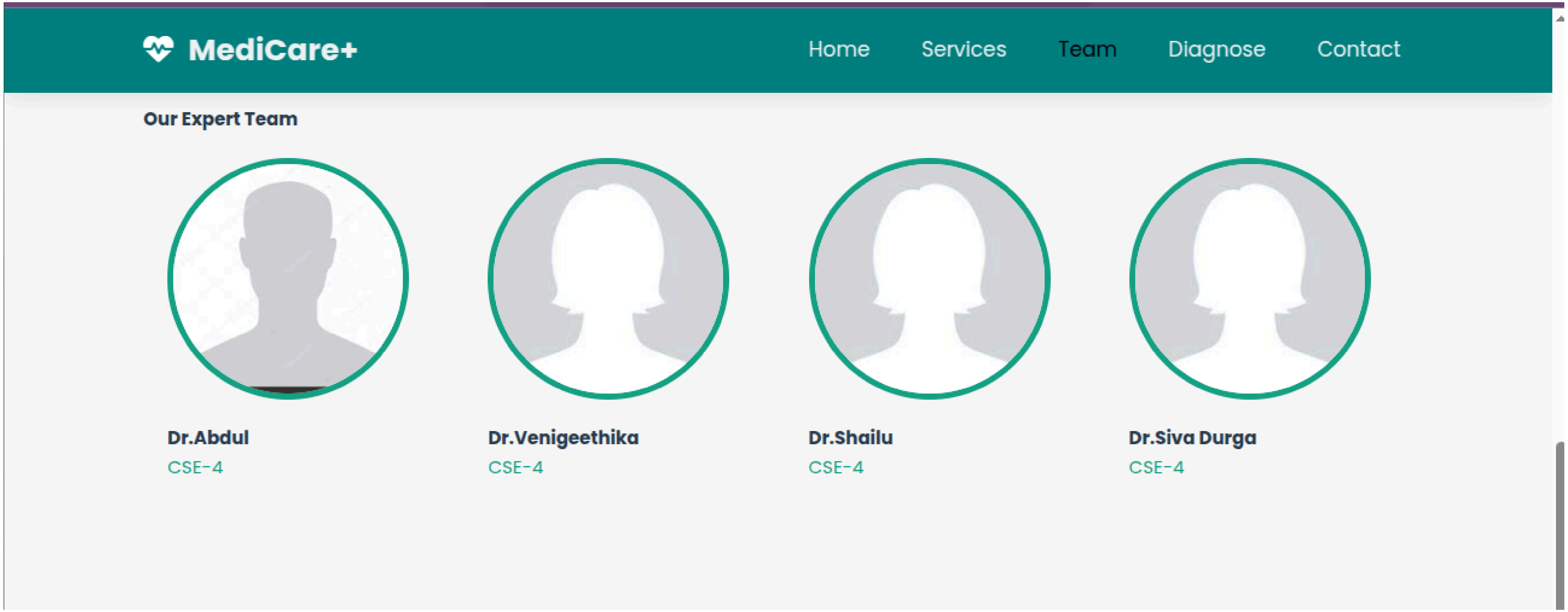
Sample Outputs

2.Our Services



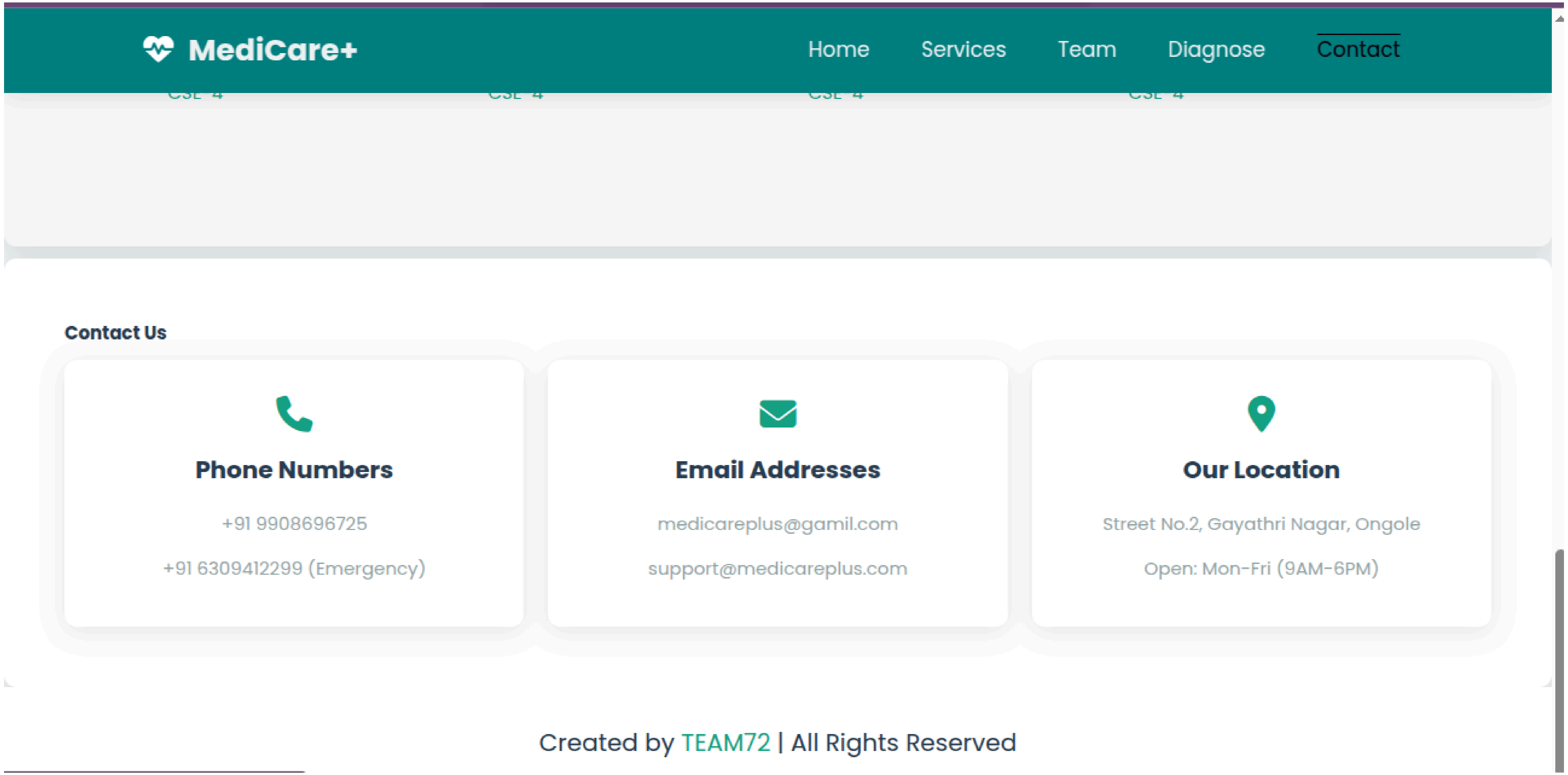
Sample Outputs

3.Team



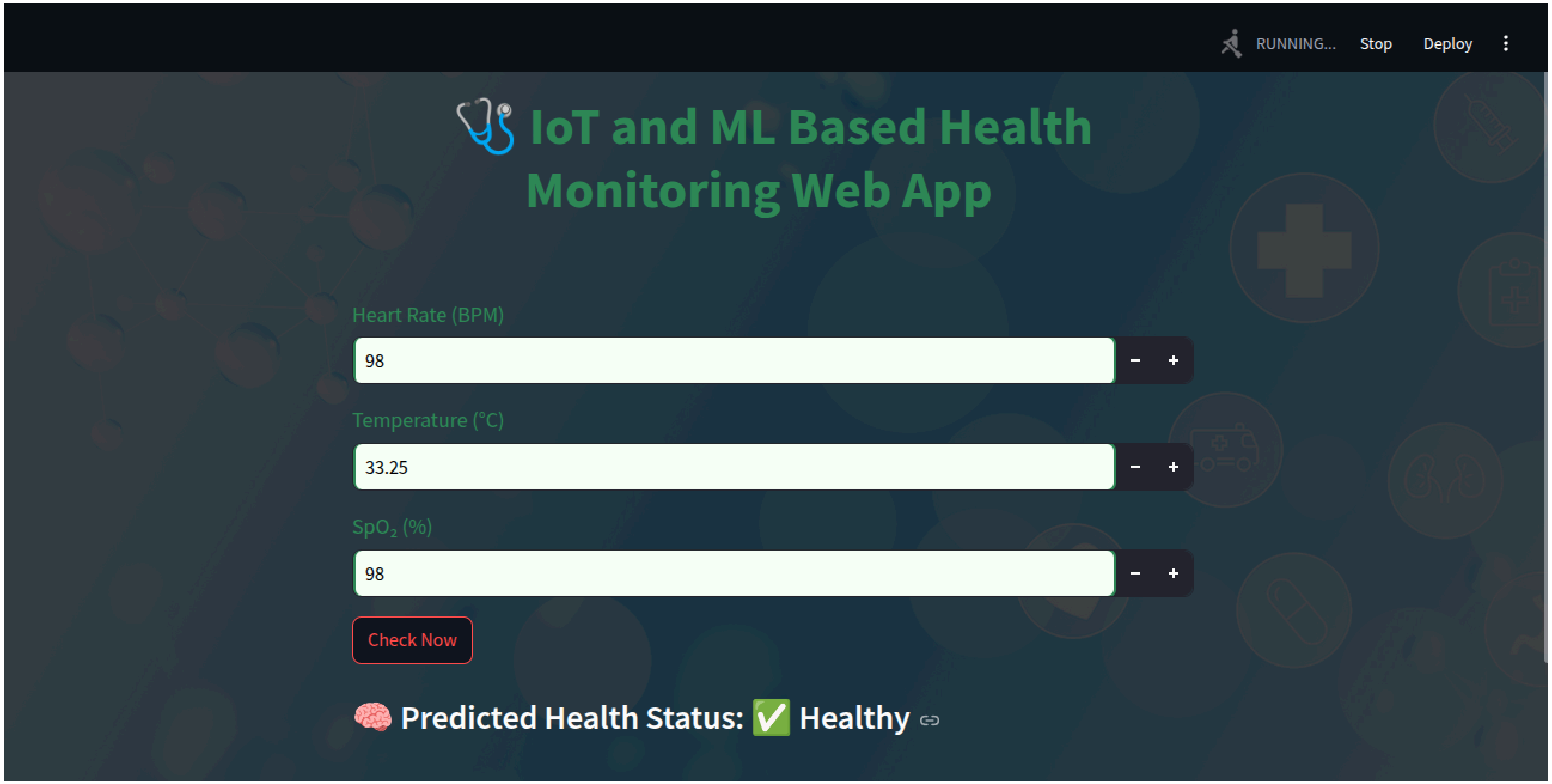
Sample Outputs

4.Contacts



Sample Outputs

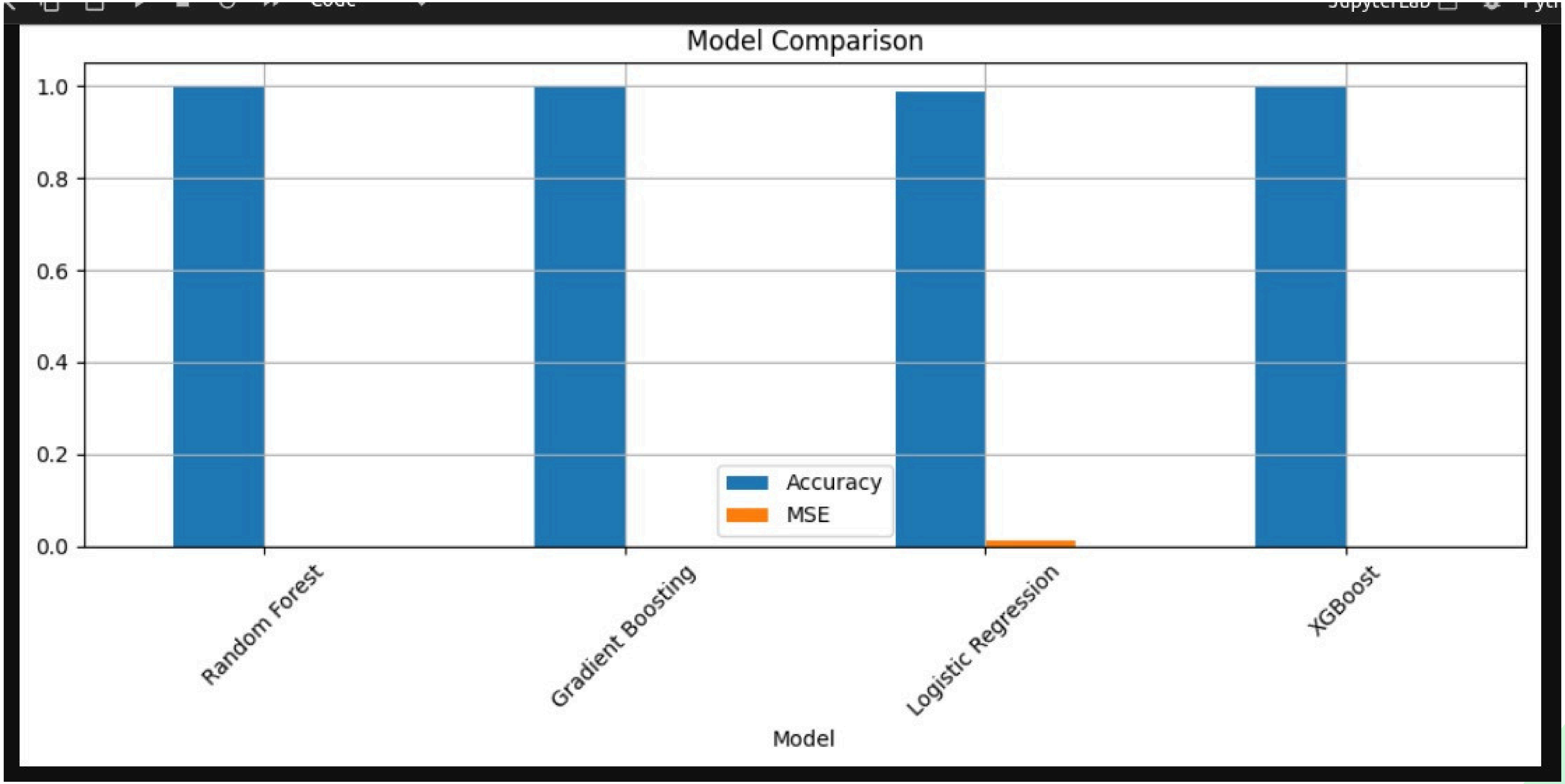
5.Health values



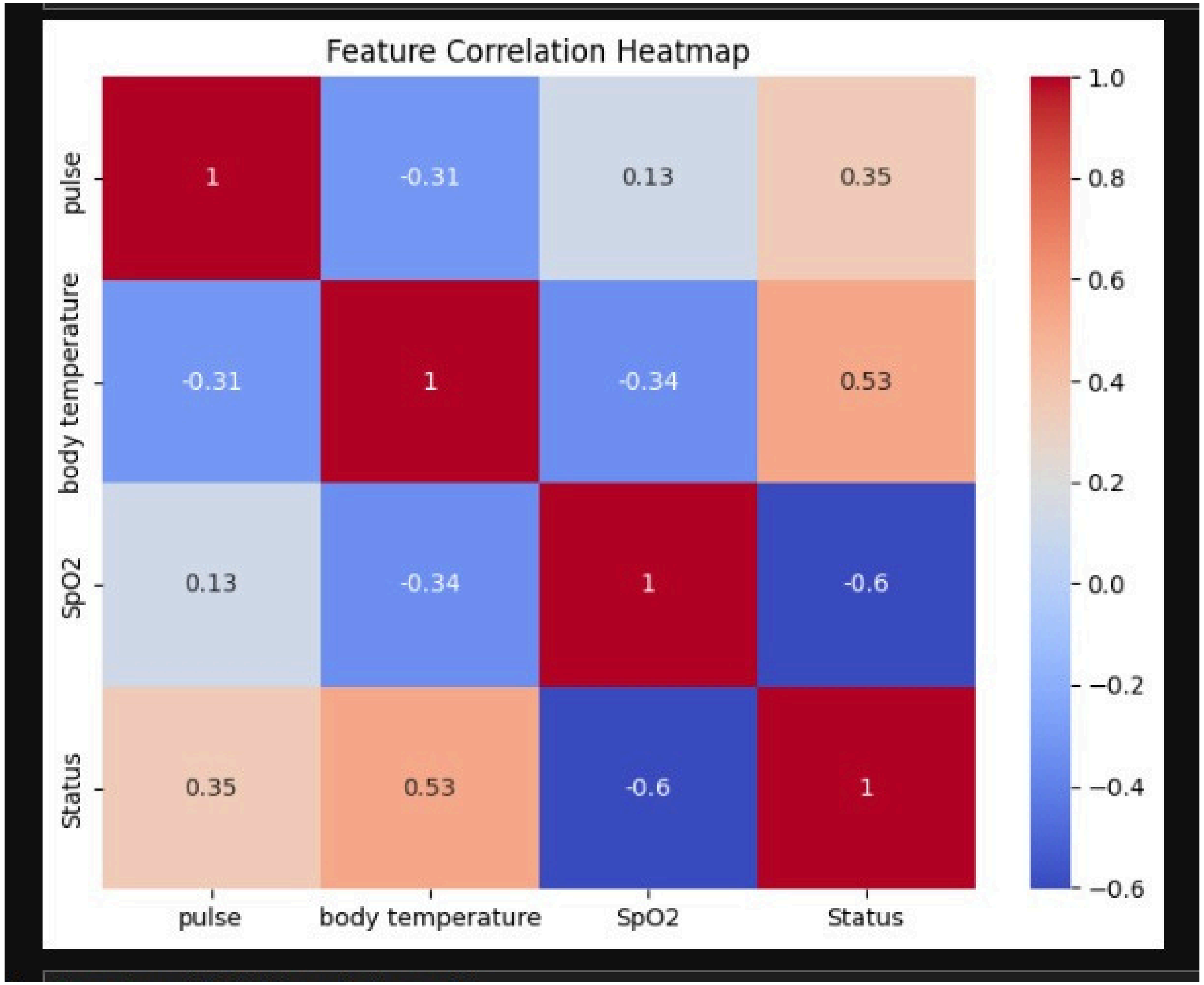
Sample Outputs

6.Health Report

Graphs


Model Comparison
1.0
0.8
0.6
0.4
0.2
0.0
Accuracy
MSE
Random Forest
Gradient Boosting
Logistic Regression
XGBoost
Model

Graphs


Feature Correlation Heatmap
pulse
body temperature
SpO2
Status
1
-0.31
0.13
0.35
-0.31
1
-0.34
0.53
0.13
-0.34
1
-0.6
0.35
0.53
-0.6
1
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

Advantages

- Real-time monitoring of vital signs (heart rate, SpO_2, 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