B-HEALTHY
(DOCUMENTATION)

Guide:
Akhil Sai

Team By:

Amrutha Bandi
Basha E
Harshita K

B-HEALTHY (DIGITAL HEALTH MONITORING UNIT)

The Digital Health Monitoring Unit is a Raspberry Pi-based project that monitors vital health parameters such as blood pressure, pulse rate, and body temperature. It provides real-time data on a web dashboard and sends alerts through SMS for abnormal readings. The system also integrates with a secure platform for data logging and uses a buzzer for immediate physical alerts.

Required Libraries

1. Backend Libraries

- **Flask:** Web framework to host the backend API.
  1. **Install:** pip install flask
- **Flask-StringIO** :To create in-memory file-like objects for reading/writing strings, used here for handling CSV data in memory
- **Adafruit_DHT:** Interface with the blood pressure,temperature and pulse sensor.
  1. **Install:** pip install Adafruit_DHT
- **Adafruit_GPIO :** Manage GPIO pins .
  1. **Install:** pip install Adafruit-GPIO Adafruit-MCP3008
- **RPi.GPIO:** GPIO library for Raspberry Pi
  1. **Install:** pip install RPi.GPIO
- **CSV**: A module for reading and writing, often used for logging or exporting data.
- **Requests:** Send HTTP requests to ThingSpeak.
  1. **Install:** pip install requests

Project Setup

1. Hardware Setup:

- Connect the BP sensor, pulse rate, Temperature sensor and buzzer to Raspberry Pi GPIO pins as specified in app.py.

2. Software Setup:

- Clone or place project files.
- **app.py**: Backend server code for managing and processing health monitoring system data.
- **Index.html:** Simple login page for user authentication and access control.
- **reading.html**: Displays real-time health parameters (blood pressure, pulse rate, temperature) with interactive charts.
- **Style.css :** Defines the styling and layout for the health monitoring web interface**.**
- **app.js**: Handles client-side interactions and communication with the backend for dynamic updates and responsive functionality.

3. Execution:

- Start the Flask server on the Raspberry Pi or the designated server device using: `python app.py`.
- On the **patient's device**, open the `index.html` file in a web browser to access the front-end.
- The **dashboard** displaying real-time sensor data will be directly visible by accessing the Flask server through its IP address.

4. Testing:

- Simulate alerts over hitting the range and buzzer functionality.
- Check whether Recommendations are updating according to the updating readings

SNAPSHOTS

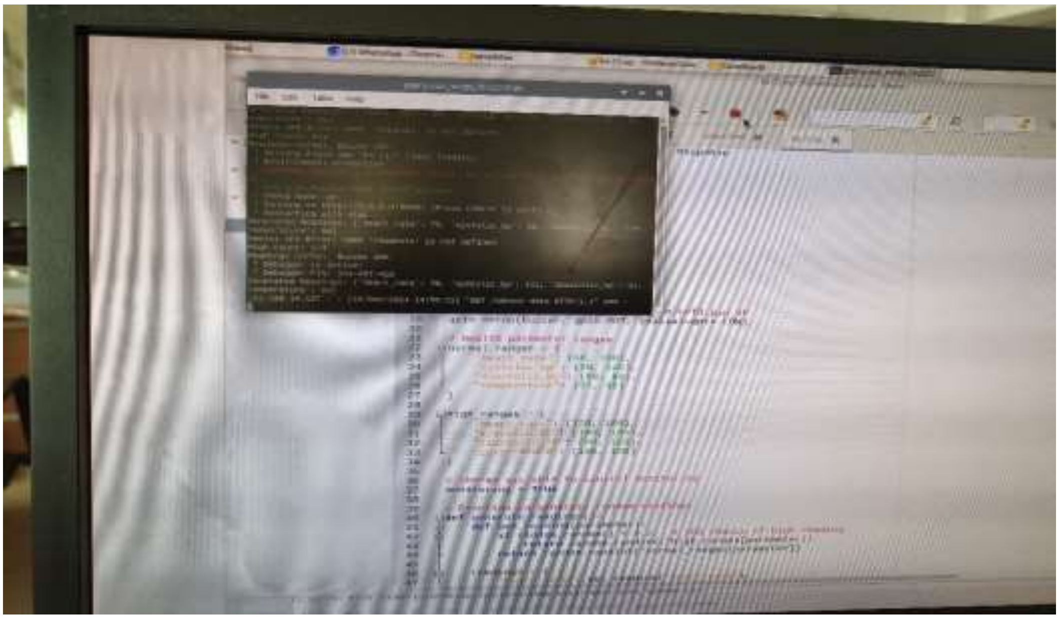
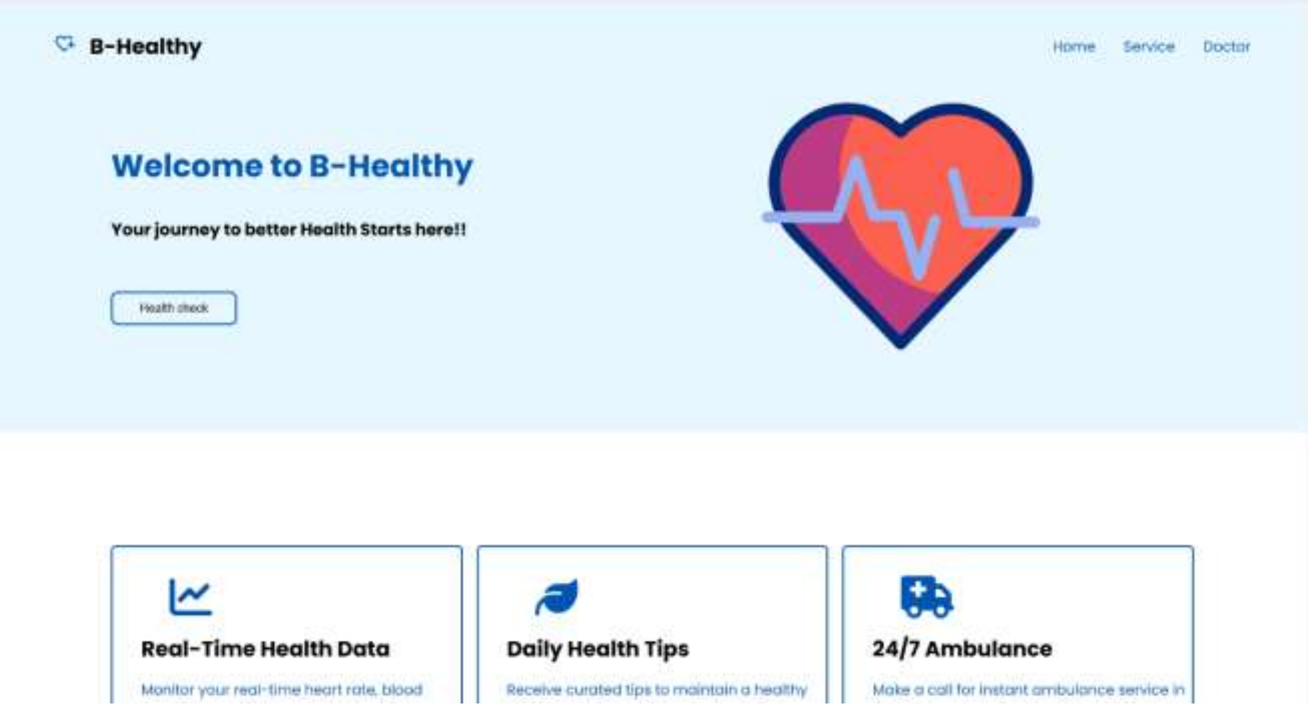


Figure1: Flask app running



Welcome to B-Healthy

Your journey to better Health Starts here!!

Real-Time Health Data

Monitor your real-time heart rate, blood

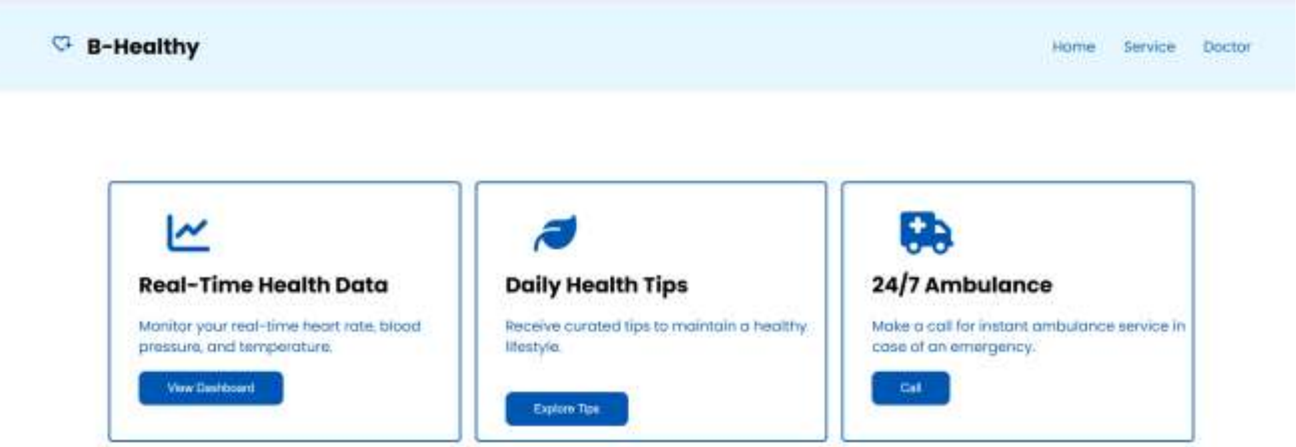
Daily Health Tips

Receive curated tips to maintain a healthy

24/7 Ambulance

Make a call for instant ambulance service in

Figure2: Homepage



B-Healthy

Real-Time Health Data

Monitor your real-time heart rate, blood pressure, and temperature.

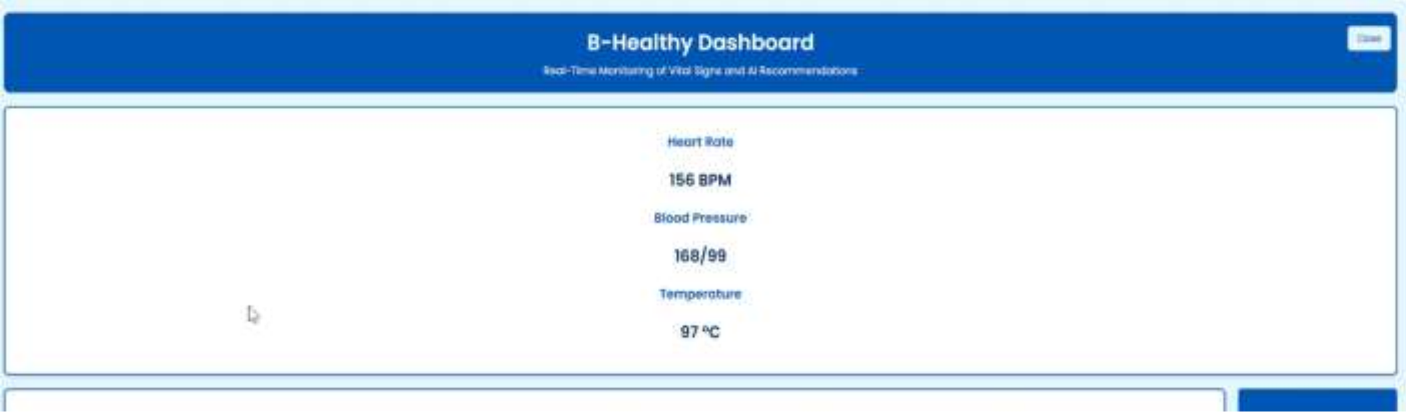
Daily Health Tips

Receive curated tips to maintain a healthy lifestyle.

24/7 Ambulance

Make a call for instant ambulance service in case of an emergency.

Figure3: Services



B-Healthy Dashboard

Real-Time Monitoring of Vital Signs and AI Recommendations

Heart Rate

156 BPM

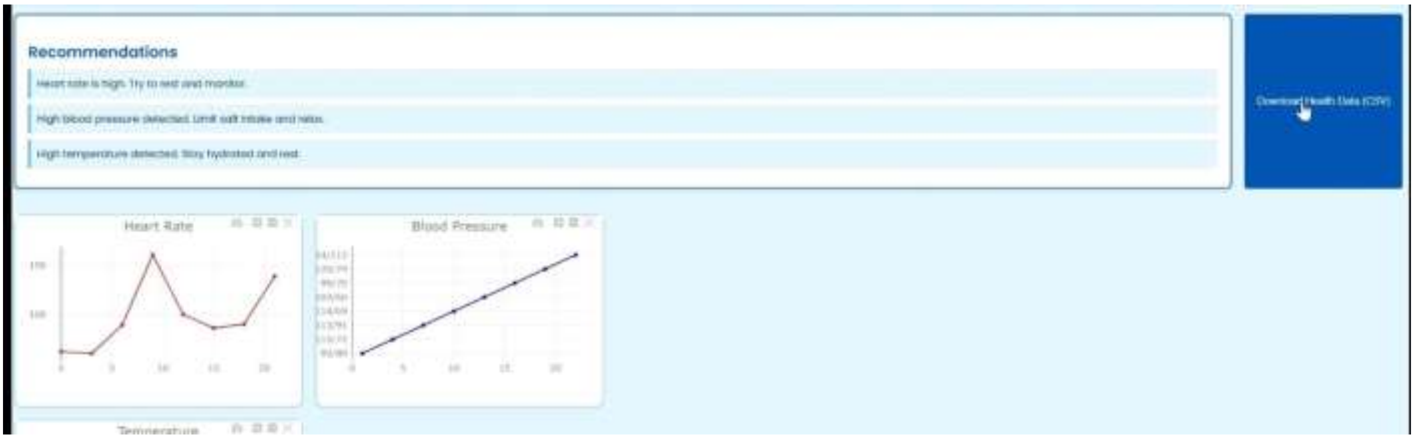
Blood Pressure

168/99

Temperature

97 °C

Figure 4: Dashboard



Recommendations

Heart rate is high. Try to rest and monitor.

High blood pressure detected. Limit salt intake and relax.

High temperature detected. Stay hydrated and rest.

Figure 5: Recommendations & History Chart

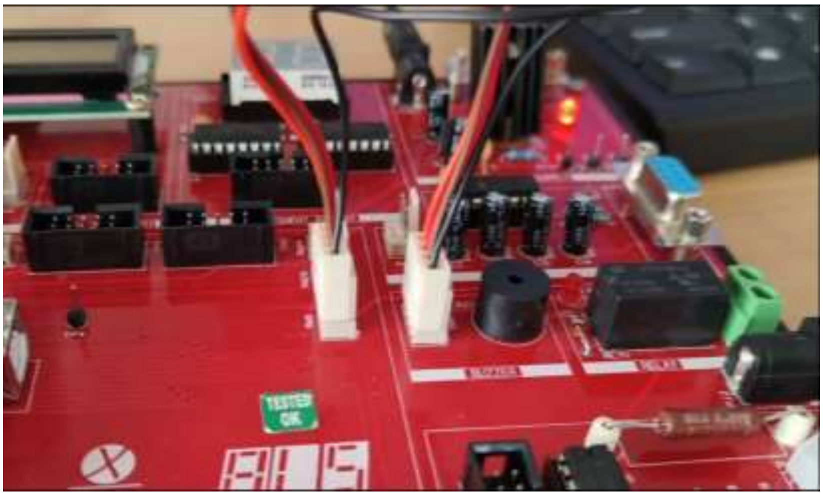


Figure 6: Buzzer Alert

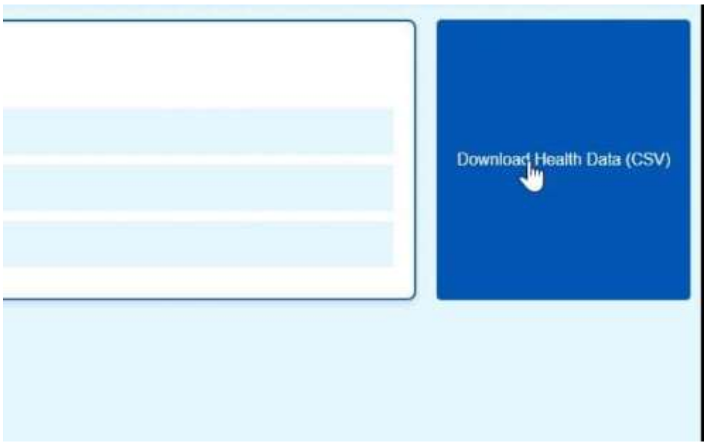


Figure 7: Download Health data(via CSV file)

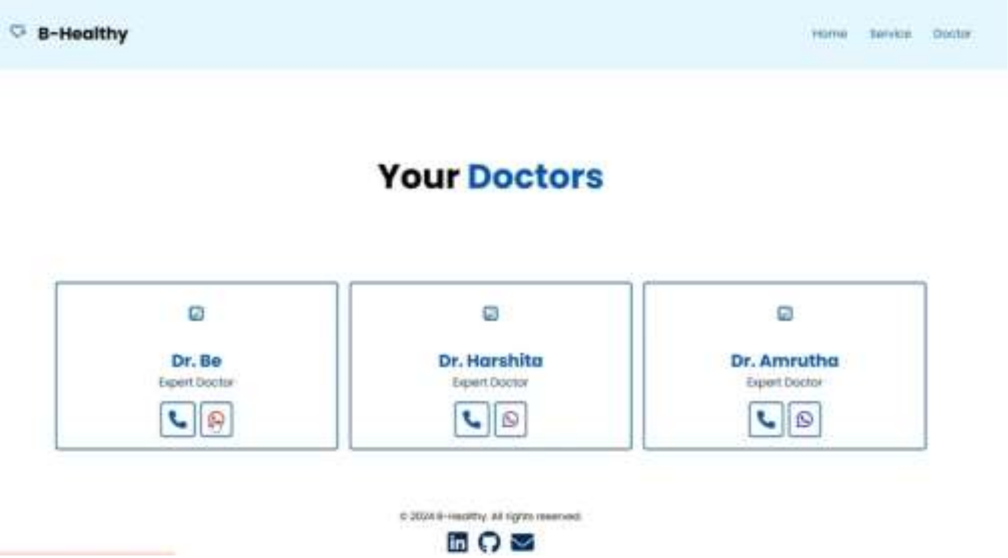


Figure 8: Redirecting to Doctor’s WhatsApp ChatBox

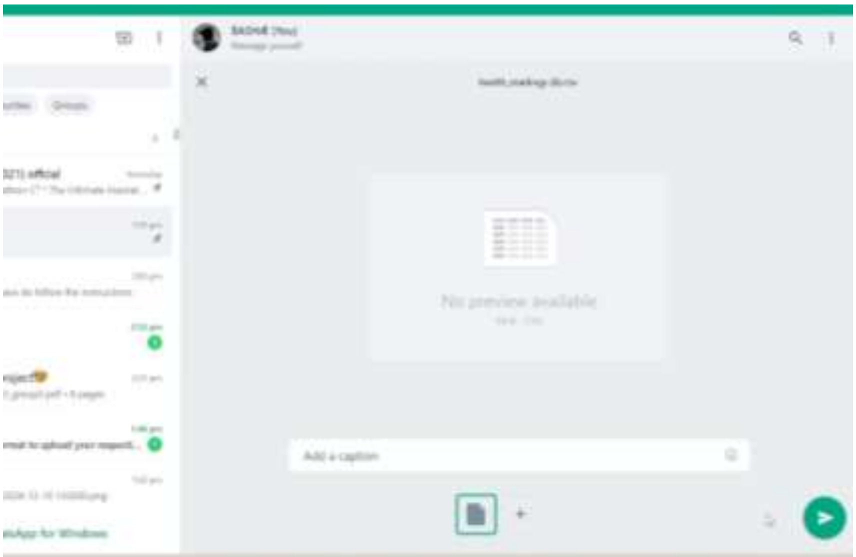


Figure 9: Communicating with Doctor through WhatsApp by sending the health data