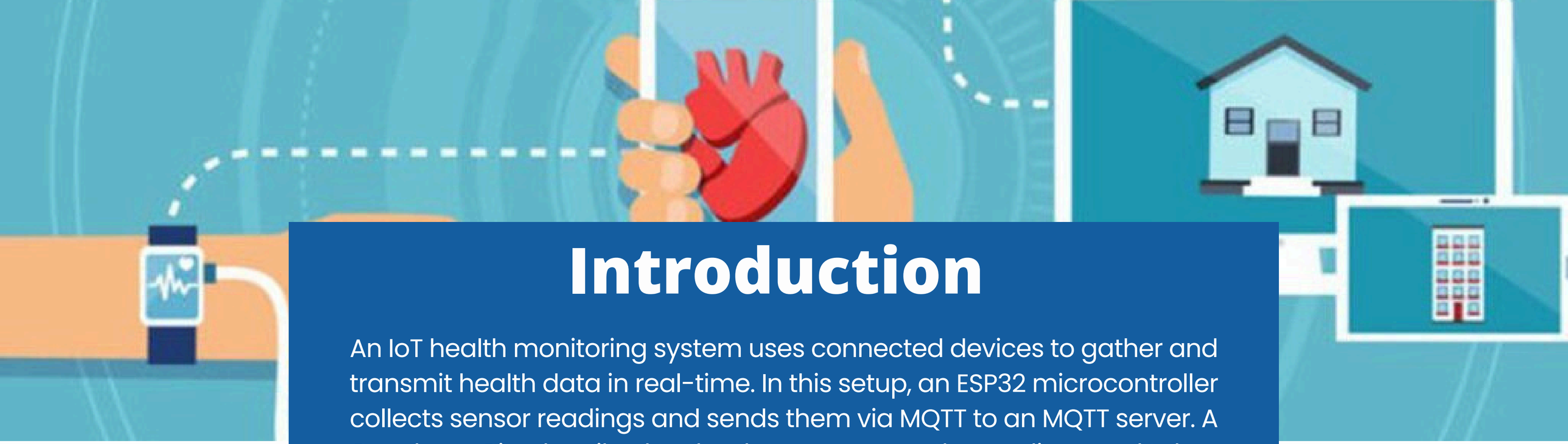
SAMSUNG

IoT Health Monitoring System

Overview

- Introduction
- Our Team
- About us
- Problems
- ESP32
- Raspberry PI
- Node Red
- Conclusion



Introduction

An IoT health monitoring system uses connected devices to gather and transmit health data in real-time. In this setup, an ESP32 microcontroller collects sensor readings and sends them via MQTT to an MQTT server. A Raspberry Pi, subscribed to the data, processes the readings and takes actions based on specific conditions, such as activating a camera. Captured images and data are then displayed on a Node-RED dashboard, allowing for real-time monitoring and response. This system enhances healthcare by enabling continuous, automated tracking of health metrics.

Our Team

Alaa Hamed

Helmy Ahmed

Abdulrahman Mohamed

Marwa Moniem

Yusuf Borham

Abdelrahman Mahmoud

About us

We are a passionate team of innovators, engineers, and problem-solvers driven by a shared goal of leveraging technology to improve lives.

With diverse expertise in IoT, software development, and systems integration, we work together to create smart solutions that address real-world challenges.

Our team thrives on collaboration, creativity, and a commitment to pushing the boundaries of what's possible. United by a dedication to excellence, we believe in the power of teamwork to deliver impactful, meaningful results.

Problems

During the COVID-19 pandemic, the need for continuous, remote health monitoring became crucial, especially for high-risk individuals. Traditional systems often fail to provide real-time data and timely responses, making it difficult for healthcare providers to track patients' health remotely. This project aims to address these challenges by developing an IoT-based system for automated health monitoring, enabling real-time data transmission and quick response actions, reducing the need for in-person contact while ensuring patient safety and timely care.

Problem 01

How can we make the device small enough for the patient to carry with them and integrate a camera so that we can monitor the patient from a distance? Additionally, how will the camera be able to detect if the patient has fallen or fainted?

Problem 02

What type of sensor will we use, how will it be integrated into the system, and what are the normal values for a healthy human, such as heart rate and temperature

Problem 03

How can we visualize the readings and the photo of the patient in a way that is easy to understand for any user, not just doctors

ESP32

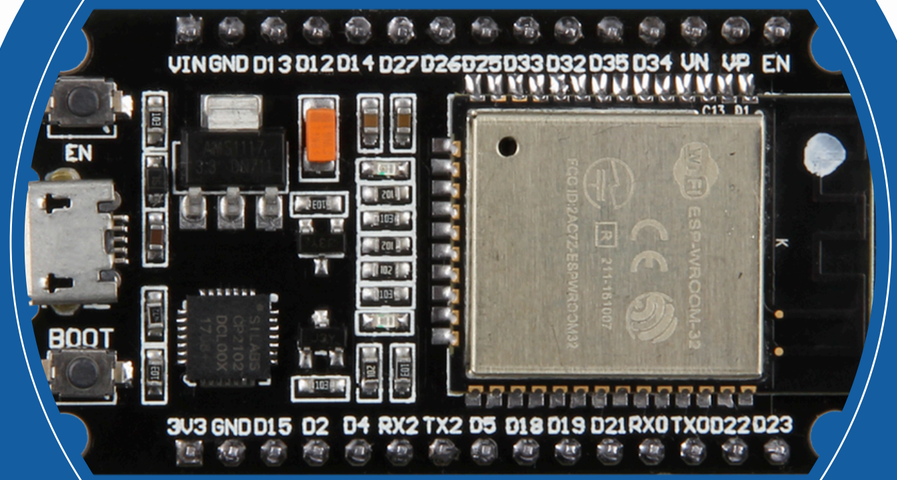
sensor connection & publish it to mqtt server

ESP32 connection

-We used esp32 to provide an analog to digital convert

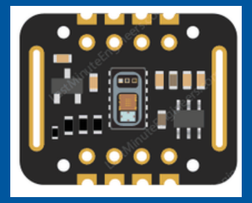
-All sensor are connected to it and it hang on paitent’s wrist like watch

-It publishes all sensor readings to an MQTT server



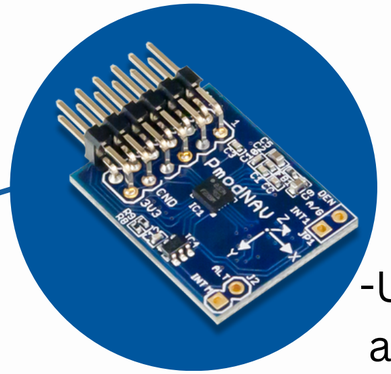
sensors

MAX30102



-Used to measure heart rate and oxygen saturation in blood

IMU



-Used to measure the acceleration and the amplitude in Z-axis to determine if patient fall or no

LM35

-Used to measure body temperature

RASPBERRY PI

connected to Raspberry Pi Camera and speaker

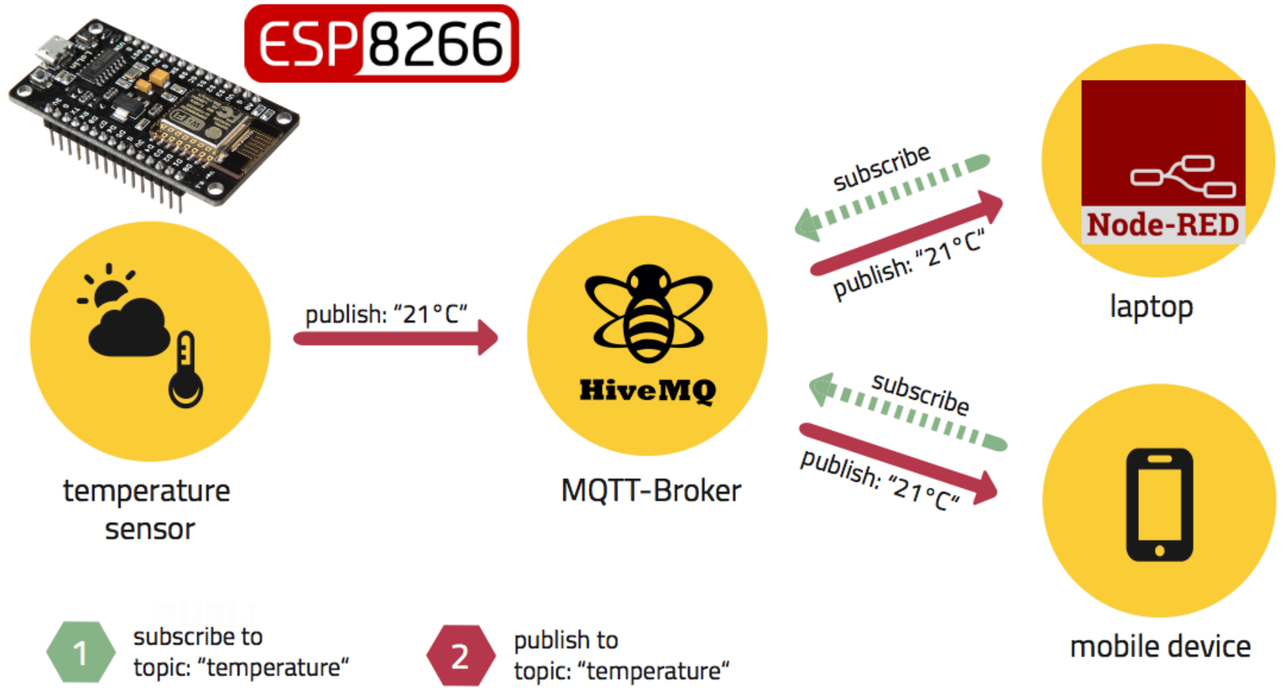
hivmq Broker

1. **Sensor Data Publishing:**

The ESP publishes sensor data to the HiveMQ broker, enabling real-time data transmission.

2. **Raspberry Pi Subscription:**

The Raspberry Pi subscribes to the broker, receiving and processing the sensor data for further actions.



Raspberry Pi Camera

01

Automatic Capture on Emergency Events:

- The camera automatically takes snapshots when an emergency reading is received via the HiveMQ subscription.

02

Integration with Node-RED via HiveMQ:

- The camera is connected to Node-RED through the HiveMQ broker, enabling real-time communication and control.

03

Manual Snapshot Triggering:

- Users can manually trigger the camera to take a snapshot through the Node-RED dashboard.

04

Remote Visualization:

- All captured images are sent to Node-RED for remote visualization, allowing real-time monitoring from anywhere.

Speaker System

Real-time Sensor Data Evaluation

- The code is designed to continuously monitor sensor readings from a system that tracks patient health metrics. It collects data from multiple sensors, including heart rate and oxygen levels (via the MAX30102 sensor), motion and orientation (via the IMU sensor).

Alert Mechanism

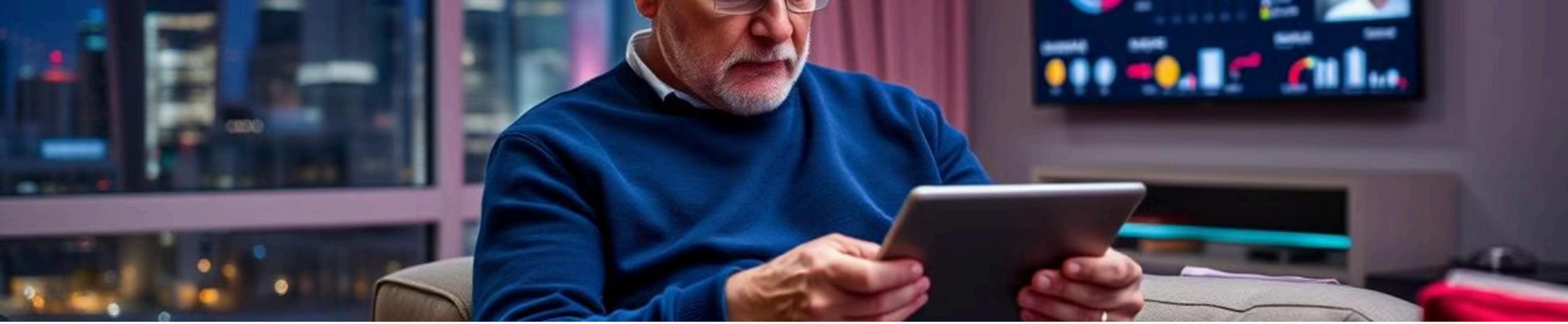
Based on the sensor data, the code determines whether the patient's condition is safe or unsafe. In the event of an unsafe condition, the code triggers an emergency response by sending an alert message and activating an alarm sound

Emergency Response

the system can quickly alert healthcare providers when a patient's vital signs indicate a potential emergency. This allows medical staff to take immediate action,

NODE RED

The UI and visualization of sensor readings



Node-RED’s Role :-

Data Acquisition

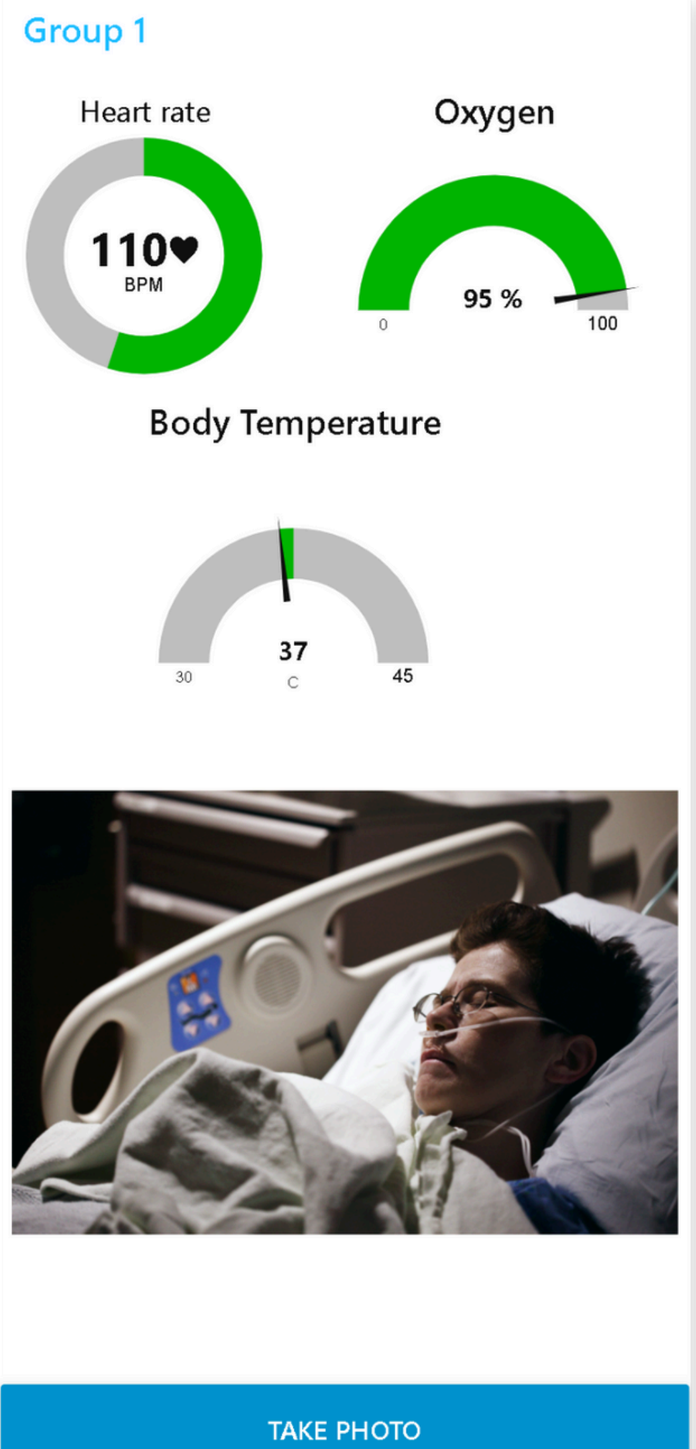
- **Receives sensor readings**
- **Processes data streams**
- **Triggers alerts when needed**
- **Manual image shots**

Visualization

- **Displays real-time data**
- **Provides interactive dashboards**
- **Display real-time imge**

Integration

- **Connects to various devices**
- **Facilitates data exchange**
- **Customizable workflow**



User Interface

An easy-to-use and clear interface that allows doctors and regular users to check on their patients without interacting with them

Heart Rate Monitoring :

A gauge that visualizes heart rate sensor readings.

Oxygen Saturation Level :

A gauge that visualizes oxygen in blood precentage

Body Temperature :

A gauge that visualizes body temperature in Celsius

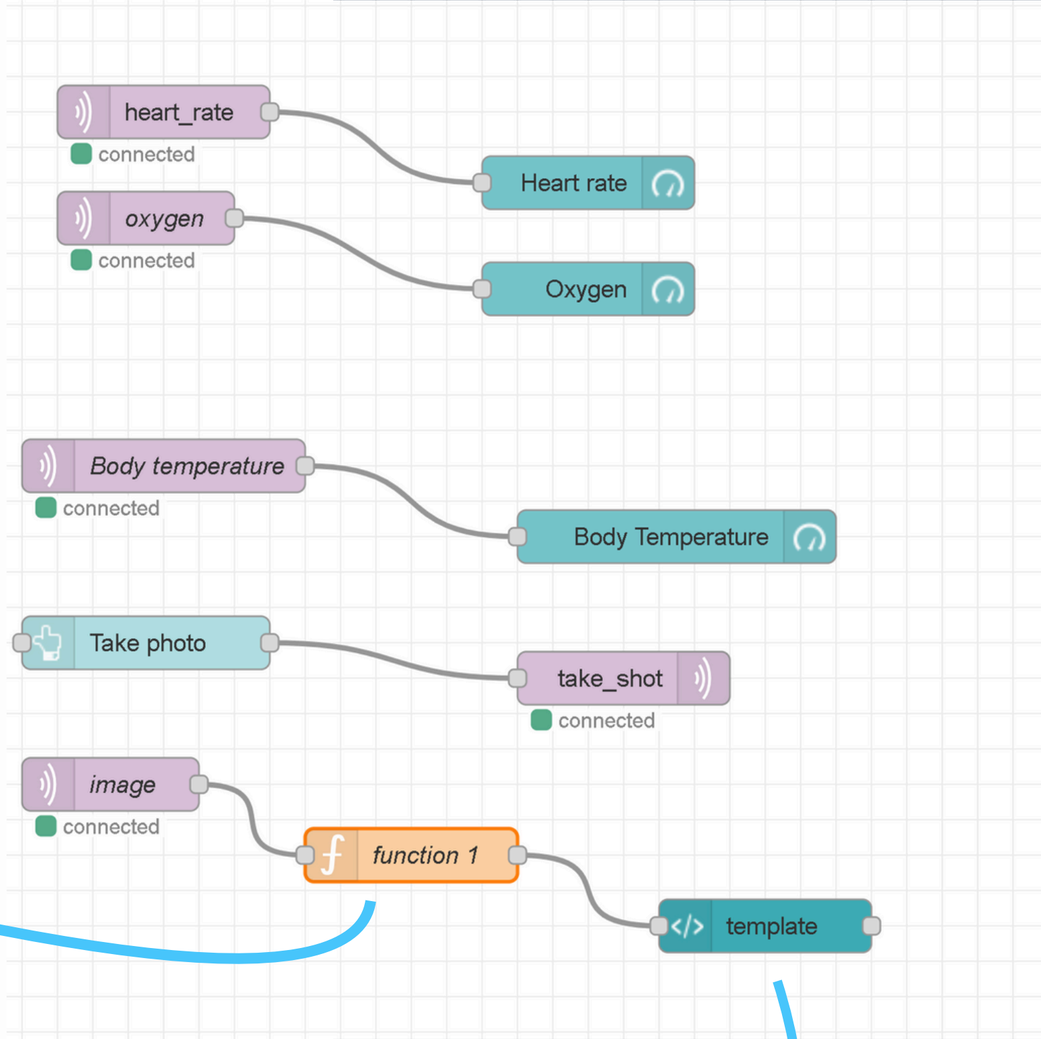
Patient Monitoring:

A real-time photo of the patient taken when the sensor readings are abnormal or by manual capture if necessary.

Node Red Flow

-MQTT in node used to subscribe to the sensor readings topic

-MQTT out node used to publish manual capture signal




```
let base64Image = msg.payload;

// Ensure that the Base64 string contains the appropriate header
if (!base64Image.startsWith('' + base64Image; // Ass
}

// Adjust the width and height in the img tag's style attribute
msg.payload = '<img width=\'100%\' src=\'' + base64Image + '\' /
return msg;
```

-The image received from MQTT server encoded as base64 and this function decoded it to image

-The template node uses HTML code to display image in dashboard

```
<div>
  <div ng-bind-html="msg.payload"></div>
</div>
```

Remote patient monitoring (RPM) are set to transform the healthcare industry in several ways over the coming years:

Early Detection & Prevention:

- Continuous monitoring for early detection of health issues.

Enhanced Chronic Disease Management:

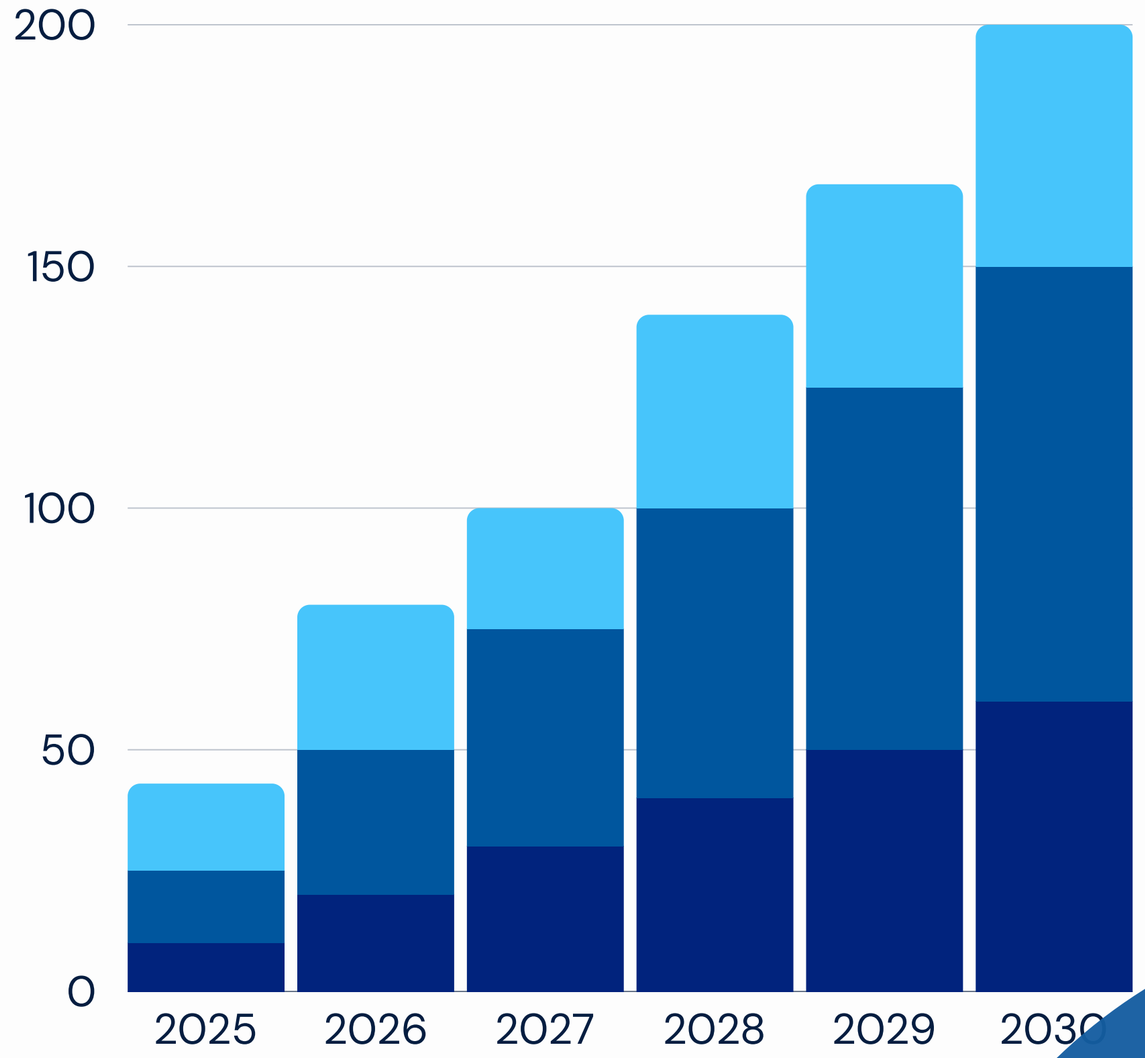
- Better management of chronic conditions like diabetes and hypertension.
- Reduced need for frequent hospital visits.

Cost-Effectiveness:

- Lower healthcare costs for patients and systems.

Real-Time Emergency Alerts:

- Immediate alerts for critical conditions.
- Faster emergency response and medical intervention.



THANK YOU!

SAMSUNG