Internet of Things

CrowdSecure bracelet

Attendance and security monitoring at crowded places

Team Members:

Lina-Chemeliine Titova

Ana Angosto

Mikhail Pilipov

Agenda

What to Know?

The main functions of the bracelet
General requirements from stakeholders
Key parameters that are monitored
Node-RED
Node-RED diagram of the project

The main functions of the bracelet

- Tracking GPS location
- Event handling when the alarm button is pressed
- Counting people in crowded places
- Monitoring health parameters

General Requirements from stakeholders

Name	Description
Durability and comfort of bracelet	The bracelet has to be made from hypoallergenic, lightweight and durable materials
Adjustability	The bracelet should fit vatious wrist sizes
Wireless technologies	The bracelet should be equipped by Bluetooth, GPS, Wi-Fi
Necessary sensors	The bracelet should be equipped by Heartrate and Saturation sensors
Display	The bracelet’s screen should be bright and clear such as OLED or AMOLED with responsive touchscreen

General Requirements from stakeholders

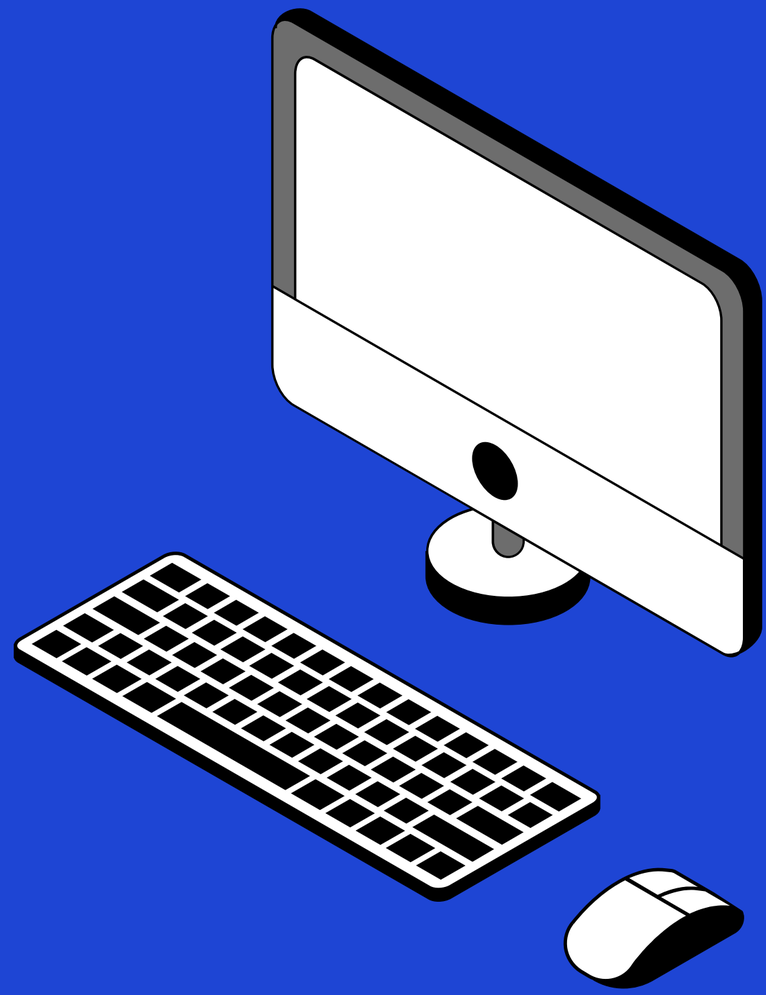
Name	Description
Durability and water persistence	The bracelet is worn on the arm and therefore should be used in withstand various conditions
Crowd management	The bracelet has to provide real-time tracking of the people and emergency alerts
Intuitive interface	The bracelet should be intuitively understandable with multilingual support

The key parameters to monitor

Heart Rate

Saturation - level of blood oxygen

GPS Location



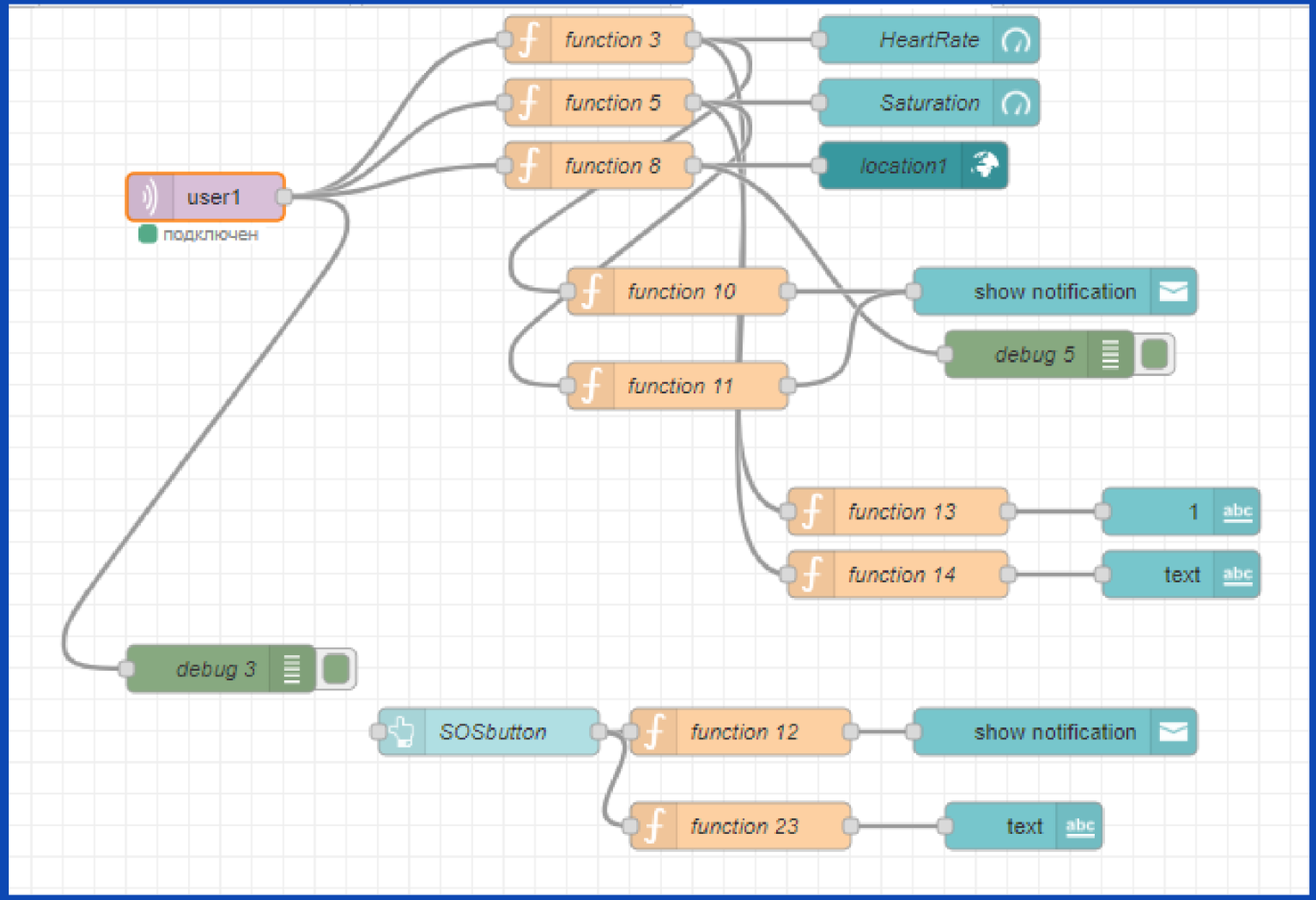
What is Node-RED?

Node-RED is a flow-based development tool for visual programming, designed for wiring together hardware devices, APIs, and online services as part of the IoT.

It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette, which can be deployed to its runtime in a single-click.

Bracelet scheme in Node-RED

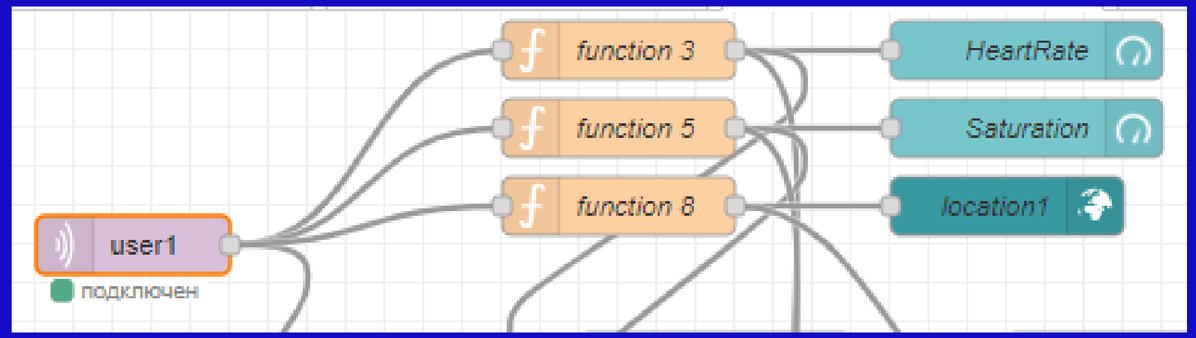
- MQTT Node Server: mqttserver@broker.emqx.io
- MQTT Node topics:
- user1, user2



How it works

- The bracelet receives Heart Rate, Saturation and Location data from the MQTT broker, the data is visible on the screen using the dashboard nodes: Heart Rate and Saturation in Gaude and Location on the online map.
- Also the presence of an SOS button, which sends a notification and is visible on the report.
- When the Heart Rate and Saturation indicators are in dangerous situations (Heart Rate>150, Heart Rate<40 and Saturation <94) - a notification is sent and displayed in the report

Bracelet scheme: extracting data from MQTT



Dashboard Nodes:

- The guide: Heart Rate
- The guide: Saturation
- The worldmap: location 1,2

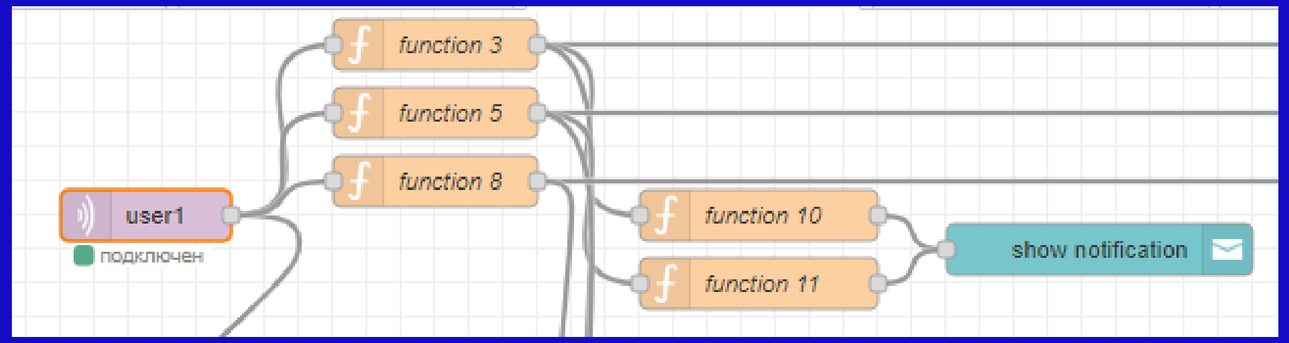
In the functions, data from the mqtt broker was assigned to the dashboard nodes

```
//function 3
//extracting Heart Rate from MQTT server
const data = msg.payload;

const HeartRate = { payload: data.HeartRate };

HeartRate.topic = "HeartRate";


return [HeartRate];
```

```
//function 5
//extracting Saturation from MQTT server
const data = msg.payload;

const Saturation = { payload: data.Saturation };

Saturation.topic = "Saturation";


return [Saturation];
```

```
//function 8
//extracting GPS location from MQTT server
var data = msg.payload;

const longitude = {payload: data.Location.longitude};
const latitude = {payload: data.Location.latitude};

const lat = parseFloat(latitude.payload);
const lon = parseFloat(longitude.payload);

msg.payload = {
    "name": "location1",
    "lat": lat,
    "lon": lon,
    "iconColor": "red"
};


return msg;
```

Bracelet scheme: notification



Dashboard Node:

- The notification

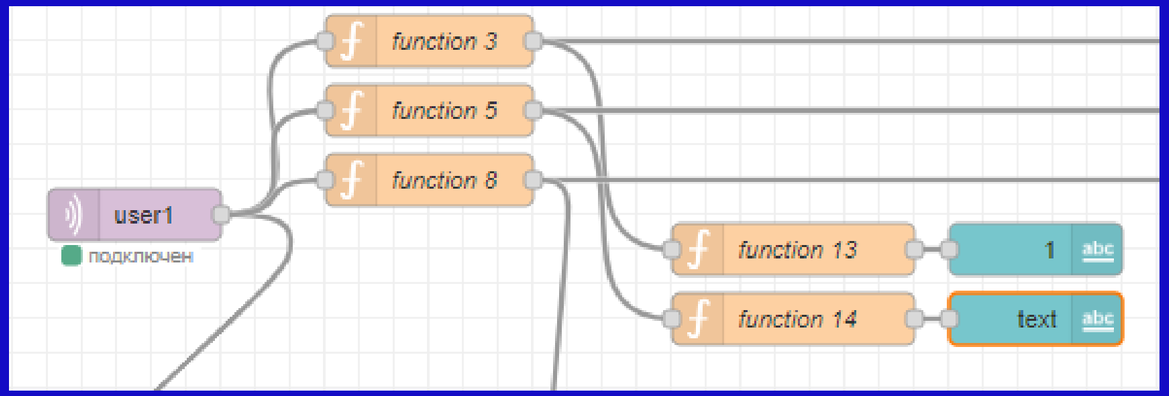
The functions monitor Heart Rate and

Saturation data extracted from the MQTT broker

```
//function 10
//Notification when Heart Rate below 40 or above 150
const HeartRate = msg.payload;

    if (HeartRate < 41){
      msg.payload = "Warning: HeartRate of user 1: " + HeartRate + " below 40!";
      return msg;
    } else if (HeartRate > 149) {
      msg.payload = "Warning: HeartRate of user 1: " + HeartRate + " above 150!";
      return msg;
    } else {
      return null;
    }
```

```
//function 11
//Notification when Saturation below 94
const Saturation = msg.payload;

    if (Saturation < 95) {
      msg.payload = "Warning: Saturation of user 1: " + Saturation + " below 94!";
      return msg;
    } else {
      return null;
    }
```

Bracelet scheme: report



Dashboard Node:

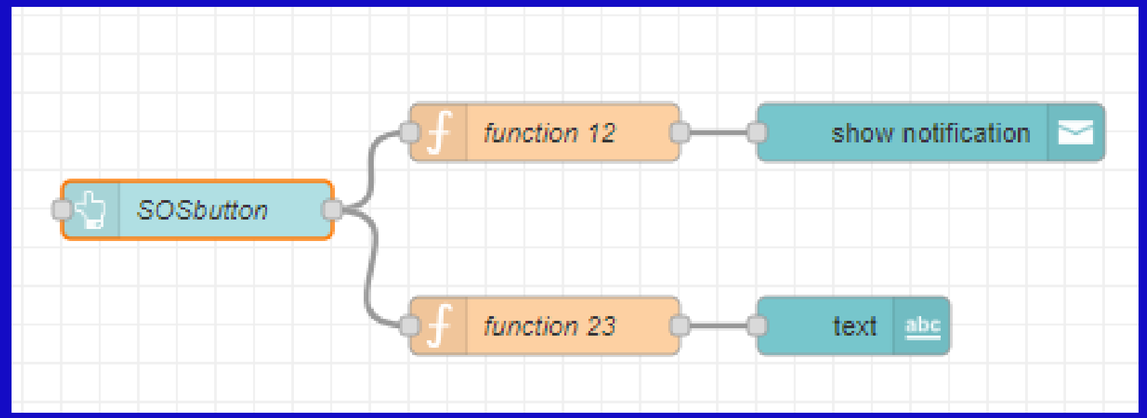
- The text

The functions show Heart Rate and Saturation reports

```
//function 13
//Heart Rate report
const HeartRate = msg.payload;

    if (HeartRate < 41){
      msg.payload = "HeartRate: " + HeartRate +
      ". Heart rate is lower than 40 beats per minute. Dangerously low";
      return msg;
    } else if (HeartRate > 149) {
      msg.payload = "HeartRate: " + HeartRate +
      ". Heart rate is higher than 150 beats per minute. Dangerously high";
      return msg;
    } else {
      msg.payload = "HeartRate: " + HeartRate + ". Heart rate is normal";
      return msg;
    }
```

```
//function 14
//Saturation report
const Saturation = msg.payload;

    if (Saturation < 95) {
      msg.payload = "Saturation: " + Saturation +
      ". Saturation is lower than 94 %. Dangerously low";
      return msg;
    } else {
      msg.payload = "Saturation: " + Saturation +
      ". Saturation is normal";
      return msg;
    }
```

Bracelet scheme: SOS button



Dashboard Node:

- The notification
- The text

The functions trigger notification and show on report if SOS button was pressed

```
//function 12
//Notofication when user press the button
msg.payload = "Warning: User 1 pressed SOS button!";
return msg;
```

```
//function 23
//SOS button report
msg.payload = "SOS button was passed";
return msg;
```

Generating data in a MQTT broker

Connection to the MQTT broker (JavaScript code)

```
'use strict';

const { connect } = require('mqtt');

const server = 'mqtt://Local_MQTT@broker.emqx.io';
const topic = 'bracelettpoicLAM';
const topic2 = 'user2';
const port = 1883;
const interval = 5000;
```

```
// Connection to the MQTT broker
const client = connect(server, {
    port: port
});

client.on('connect', () => {
    console.log('Connected to MQTT broker');
```

Generating data in a MQTT broker

```
//extraction data from topic user1
setInterval(() => {
    const data = generateData();
    client.publish(topic, JSON.stringify(data), { qos: 0, retain: false });
    console.log('Data published:', data);
}, interval);

//extraction data from topic user2
setInterval(() => {
    const data = generateData();
    client.publish(topic2, JSON.stringify(data), { qos: 0, retain: false });
    console.log('Data published:', data);
}, interval);
```

```
// Data simulation
function generateData() {
    return {
        HeartRate: Math.floor(Math.random() * (200 + 1)) + 0, // Heart rate
        Saturation: Math.floor(Math.random() * (100 - 90 + 1)) + 90, // Saturation
        Location: {
            latitude: (Math.random() * (47.092 - 35.492) + 35.492).toFixed(6), // Latitude
            longitude: (Math.random() * (18.521 - 6.627) + 6.627).toFixed(6) // Longitude
        }
    };
}
```

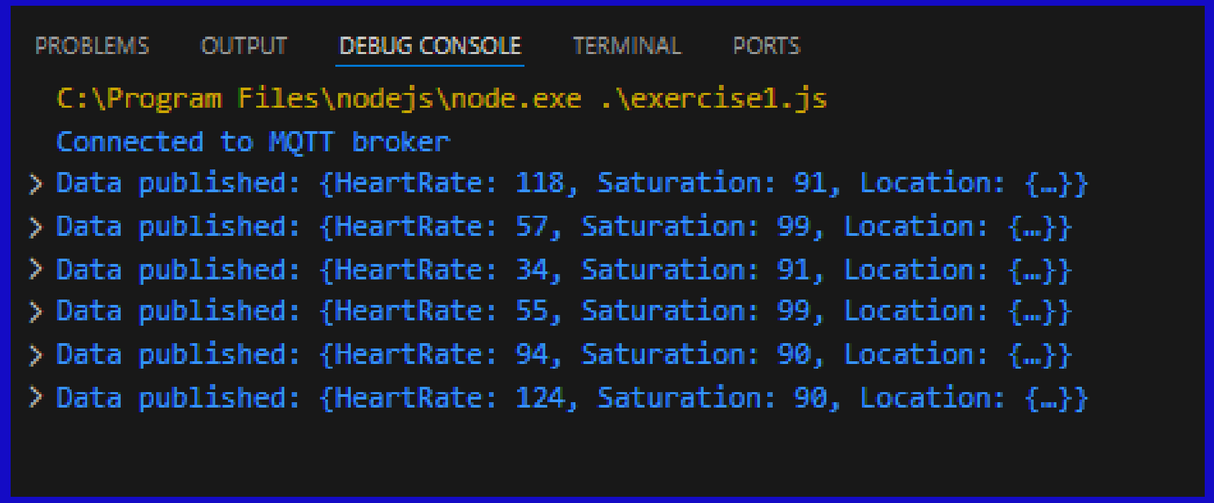
Generating Heart Rate, Saturation and Location in topics in a MQTT broker

Generating data in a MQTT broker

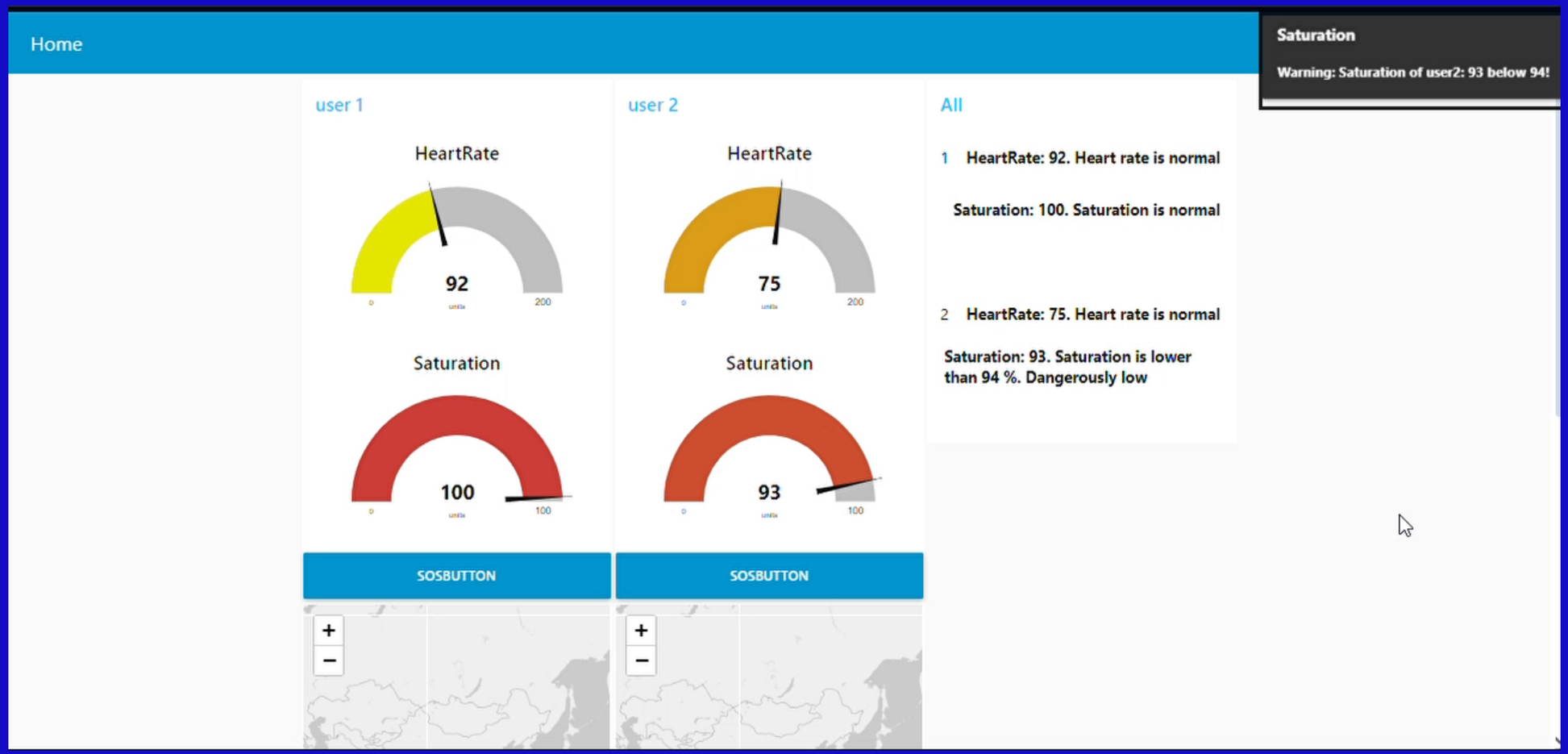
The end of the code

```
client.on('error', (err) => {
    console.error('Connection error:', err);
    client.end();
});

client.on('close', () => {
    console.log('Connection closed');
});
```

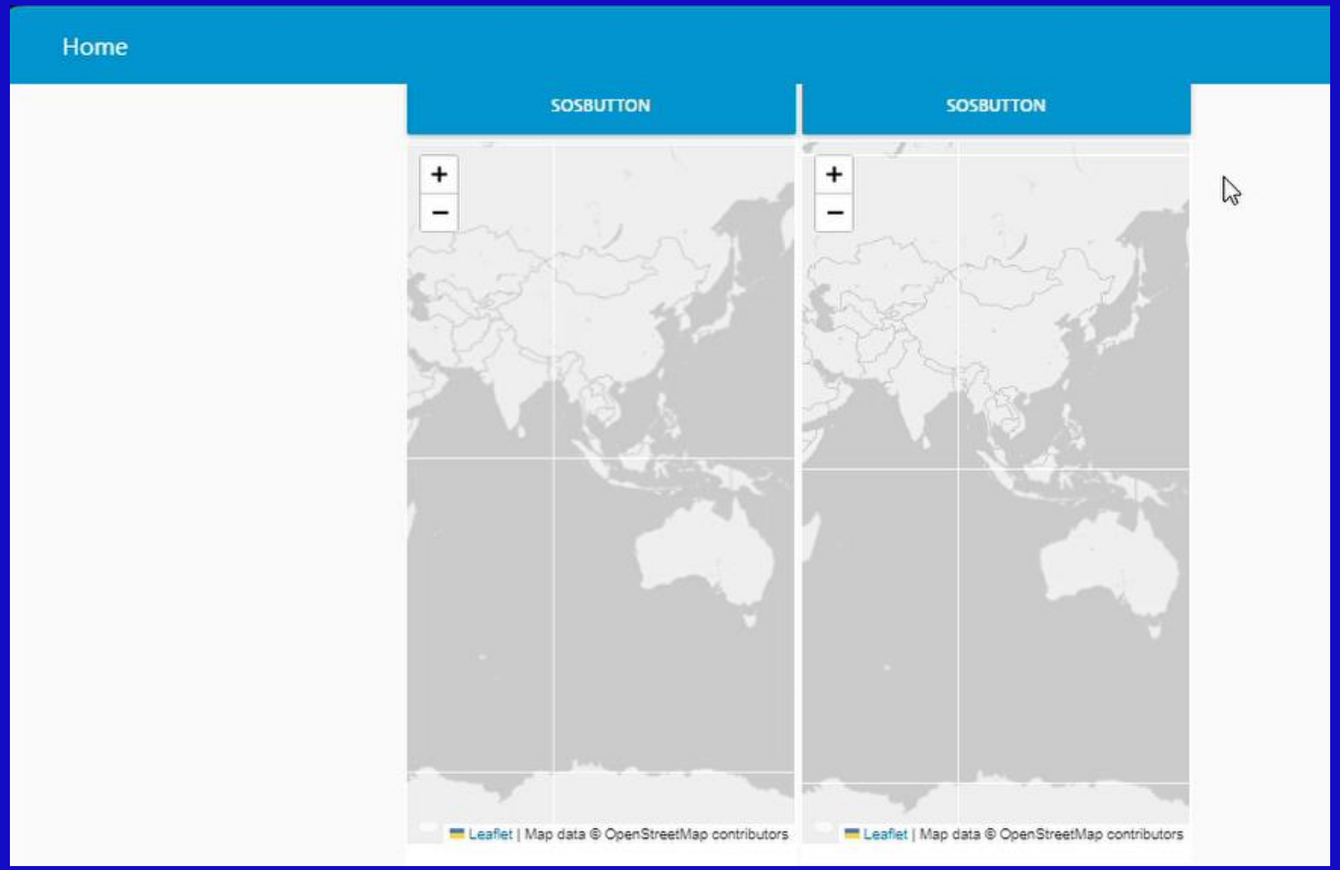
Generated data



Node-RED dashboard



Node-RED dashboard



Thank you for your attention!

References

Node-RED, flow-based programming for Internet of Things

Html.it| Article | Paolo Patierno | 2023

Learn: MQTT broker

Catchpoint.com| Web-site material | Catchpoint | 2021

What is MQTT and why it is important for the Internet of Things

Akenza.io | Article | Alexis Leibbrandt | 2023

MQTT Protocol Guide: Everything You Need to Know

Cedalo.com | Article | Tizian Prokosch| 2023

Tutorial: Connecting Devices to the IoT Platform with Node-RED & MQTT

Kaaiot.com | Web-site material | KaaIoT | 2023

Tutorial: Node-RED tutorial: How to get GPS coordinates with a Maps Widget

Industrialshields.com | Article | Fernandez Queralt Martinez | 2021

Q&A