Dharmik Jethva 202152309

MQTT

This HTML file sets up a web interface to display real-time temperature and humidity data using MQTT protocol. It utilizes Chart.js to visualize the data in a line chart format. The JavaScript code connects to an MQTT broker hosted on public.mqtthq.com and subscribes to two topics for temperature and humidity data. Upon receiving messages from these topics, it updates the chart with the latest data points. The accompanying Node.js script publishes randomized temperature and humidity data to the same MQTT topics at regular intervals, simulating sensor readings. This integration showcases the utilization of MQTT protocol for real-time data communication and visualization on web applications.

Index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>MQTT Protocol</title>
    <style>
        body {
            margin: 0;
            padding: 0;
            font-family: Arial, sans-serif;
        }

        .container {
            position: relative;
            width: 100%;
            height: 100vh;
            display: flex;
            justify-content: center;
            align-items: center;
        }

        .content {
            position: relative;
            padding: 20px;
            border-radius: 20px;
            background-color: rgba(244, 244, 246, 1);
        }
    </style>
</head>

<body>

    <div class="container">
        <div class="content-container">
            <h2>Temperature and Humidity</h2>
            <div class="content">
                <canvas id="line-chart" width="599" height="336"></canvas>
            </div>
        </div>
    </div>

    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/mqtt/5.3.6/mqtt.min.js"></script>
    <script>
        const line_chart = new Chart(document.getElementById("line-chart"), {
            type: 'line',
            data: {
                labels: [],
                datasets: [
                    {
                        data: [],
                        label: "Temperature",
                        borderColor: "#3e95cd",
                        fill: false
                    },
                    {
                        data: [],
                        label: "Humidity",
                        borderColor: "#8e5ea2",
                        fill: false,
                    }
                ]
            },
            options: {
                title: {
                    display: true,
                    text: 'Temperature and Humidity Data'
                },
                scales: {
                    y: {
                        title: {
                            display: true,
                            text: 'Temperature & Humidity'
                        },
                    },
                    x: {
                        title: {
                            display: true,
                            text: 'Time'
                        }
                    }
                },
                hover: {
                    mode: 'nearest',
                    intersect: true
                },
                tooltips: {
                    mode: 'index',
                    intersect: false
                }
            }
        });

        const client = mqtt.connect('ws://public.mqtthq.com:8083/mqtt');
        client.on('connect', function () {
            client.subscribe('ZvLVhkjaVu/temp', function (err) {
                if (!err) {
                    console.log('Subscribed to ZvLVhkjaVu/temp');
                }
            })
            client.subscribe('ZvLVhkjaVu/hum', function (err) {
                if (!err) {
                    console.log('Subscribed to ZvLVhkjaVu/hum');
                }
            })
        })
        client.on('message', function (topic, message) {
            const i = topic == 'ZvLVhkjaVu/temp' ? 0 : 1;
            const data = JSON.parse(message.toString());
            line_chart.data.labels.push(data.time.split(' ')[1]);
            line_chart.data.datasets[i].data.push(data.value);
            line_chart.update();
```

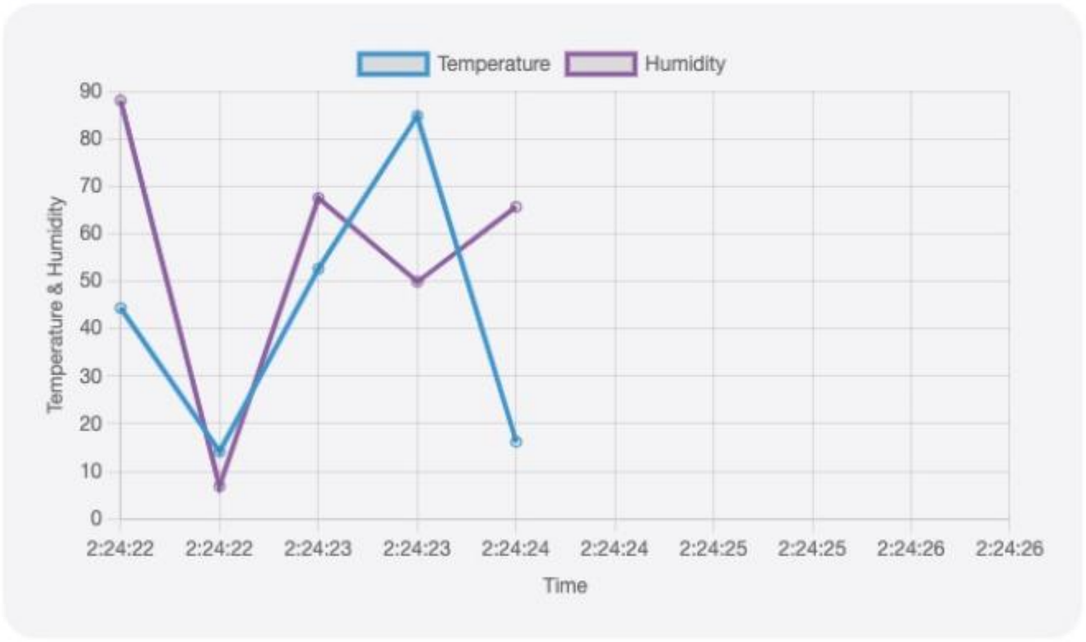
```
        })
    </script>
</body>

</html>
```

App.js

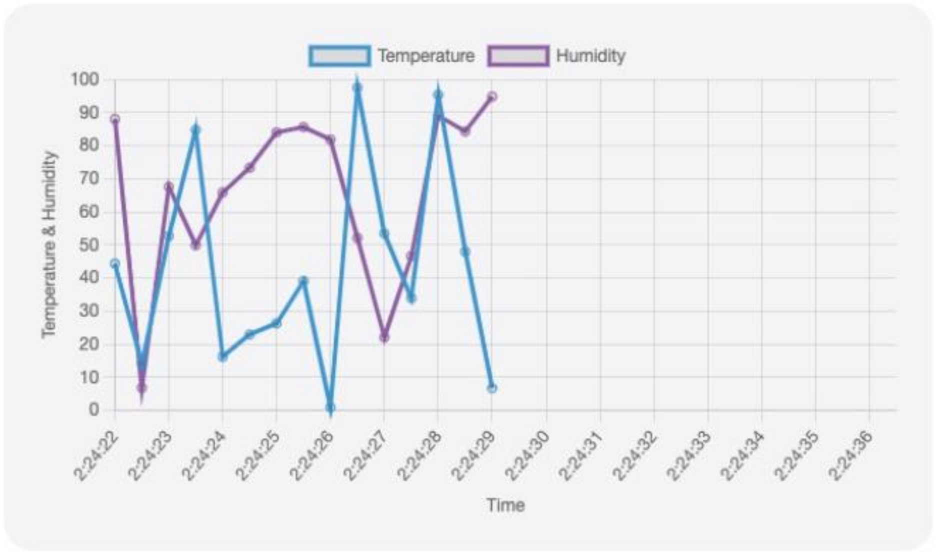
```
const mqtt = require('mqtt')

const client = mqtt.connect('mqtt://public.mqtthq.com:1883')

client.on('connect', () => {
    client.subscribe("ZvLVhkjaVu/temp", (err) => {
        if (!err) {
            console.log("Connected to MQTT broker")
        }
    })
})

setInterval(() => {
    const time = new Date().toLocaleString()
    const temp_data = {
        time: time,
        value: Math.random() * 100
    }
    const hum_data = {
        time: time,
        value: Math.random() * 100
    }
    client.publish("ZvLVhkjaVu/temp", JSON.stringify(temp_data))
    client.publish("ZvLVhkjaVu/hum", JSON.stringify(hum_data))
}, 1000)
```

```
{
  "dependencies": {
    "mqtt": "^5.3.6"
  }
}
```

Temperature and Humidity



Temperature and Humidity


Temperature
Humidity
Temperature & Humidity
2:24:22
2:24:23
2:24:24
2:24:25
2:24:26
2:24:27
2:24:28
2:24:29
2:24:30
2:24:31
2:24:32
2:24:33
2:24:34
2:24:35
2:24:36
Time