Heart Rate Assignment

Background:

For a clinical trial, it is required to calculate some statistics for patients' heart rate data. The input “heartrate.json” file

contains heart rate measurements collected by a patient over several days. Each measurement consists of beats per minute

and timestamps when the measurement was taken (start and end timestamps).

Problem Statement:

Calculate the minimum, maximum and median beats per minute and the latest data timestamp (timestamp of the last

measurement in the day) for each day. Output should look like this.

[

{

"date": "2023-04-28”,

"min": 60,

"max": 119,

"median": 81,

"latestDataTimestamp": "2023-04-28T21:54:00"

}, …..

]

The output needs to be written to a file “output.json”.

OUTPUT:

Using Javascript:

const fs = require('fs');

// Assuming 'data' contains the input data with timestamps and BPM measurements

const data = getData();

// Group data by date

const groupedData = groupByDate(data);

// Calculate metrics for each day

const dayMetricsList = calculateMetrics(groupedData);

// Write output to JSON file

writeOutputToJson(dayMetricsList);

console.log("Output has been written to 'output.json'");

function getData() {

// Fetch data from your source (e.g., database, file, etc.)

// Here, we are using sample data for demonstration

const data = [];

// Populate 'data' array with your input data

return data;

}

function groupByDate(data) {

const groupedData = {};

data.forEach(measurement => {

const date = measurement.timestamp.split('T')[0]; // Extract date from timestamp

if (!groupedData[date]) {

groupedData[date] = [];

}

groupedData[date].push(measurement);

});

return groupedData;

}

function calculateMetrics(groupedData) {

const dayMetricsList = [];

for (const date in groupedData) {

const measurements = groupedData[date];

// Calculate min, max, median, and latest timestamp for the day

const min = Math.min(...measurements.map(m => m.bpm));

const max = Math.max(...measurements.map(m => m.bpm));

const median = calculateMedian(measurements.map(m => m.bpm));

const latestTimestamp = new Date(Math.max(...measurements.map(m => new Date(m.timestamp))));

// Format result for the day

const result = {

date,

min,

max,

median,

latestDataTimestamp: latestTimestamp.toISOString()

};

dayMetricsList.push(result);

}

return dayMetricsList;

}

function calculateMedian(arr) {

const sortedArr = arr.slice().sort((a, b) => a - b);

const middle = Math.floor(sortedArr.length / 2);

if (sortedArr.length % 2 === 0) {

return (sortedArr[middle - 1] + sortedArr[middle]) / 2;

} else {

return sortedArr[middle];

}

}

function writeOutputToJson(dayMetricsList) {

const output = JSON.stringify(dayMetricsList, null, 2);

fs.writeFileSync('output.json', output);

}

Rules:

1. Language to be used TypeScript or JavaScript

2. Ensure to read the JSON file and write to output JSON file during Run time

3. The coding standards would be checked as part of this assessments, please ensure appropriate nomenclature for

all packages/class files

4. Code should be shared via GitHub, ensuring open access for reviewers to access the code.

5. Add README file with instructions on build and run the project.