Assignment: Temperature Monitoring

The assignment will have two main parts:

1. An Angular frontend web app.
2. A NodeJS backend which handles sensor data coming from a local process, and provides an API for communicating with the frontend.

Data source

Assume existence of an executable file called `measure_temp`. You don't need to care about where it is stored or whether it exists, just assume the following `nodejs` code will work

```
import { spawn } from 'node:child_process';

const measureTempProcess = spawn('measure_temp');
measureTempProcess.stdout.on('data', (buf: Buffer) => {
    console.log(`Subprocess outputted ${buf.length} bytes`);
});
```

You can chose to create a mock of this process (or the js code interacting with it) for your testing, or just skip testing entirely - it's your choice. Your approach to this "missing piece" issue is an important part of our screening process.

Binary data structure

As we saw, the nodeJS `Readable` which is accessed by `ChildProcess#stdout` gives a binary Buffer object. You can expect that the executable will, every few seconds or so, output 2-byte messages with the following data structure:

Bits	**0-1**	**2-7**	**8-15**
Meaning	A two bit code indicating the type of message. For our purposes, we only care about "type 2 message" - messages where these bits are `10`	A number from 0-63 indicating the sensorID which took the measurement	An 8-bit unsigned integer representing the sensor reading in degrees celsius

For example:

- The bytes `A1 10` represent the bits `1010 0001 0001 0000`, meaning: a message of type 2, from sensor number 33, with a reading of 16 degrees.

- The bytes `73 0F` represent the bits `0111 0011 0000 1111`. The first two bits tell us that this is a "type 1" message, and should thus be ignored by the system.

Frontend

You must build a web app which interacts with your backend, and displays in a user-readable format:

- The number of sensors active since the server was started.
- The average amount of messages received per minute.
- A table displaying the average temperature measured from each sensor.
- A way in which the user can select a single sensor (by ID) and see live updates every time a reading is received.
- You are welcome to add more stats and/or functionality if you so wish!
  If an error occurs when requesting data from the backend, the user should know about this and be prompted to refresh the page to try again.
  Note - your frontend should be usable and understandable. It doesn't need to be pretty, but we will definitely be impressed if it is pretty!
  Your frontend must display sensibly on both mobile and desktop.

Implementation notes

- You MUST implement the assignment in NodeJS and Angular v20.
- You can use third party libraries, as long as you include them properly and cleanly.
- Write clean and organized typescript. Focus on modularity and maintaining scalability. Specifically, your TS configuration should be strict and your code style consistent.
- Work in a git repository (or multiple repositories if you prefer).
- Use AI as much or as little as you like - we recommend using it only as much as you would use it in day-to-day work. However, remember that you will be asked how the code works and potentially asked to make small changes in real time. At our company, we expect developers to "own" their code and be able to explain why they approached a problem one way as opposed to another.

Final instructions

By our estimate, this assignment should take about 4 hours. We've given you 48 hours to complete it in your own time.
While we estimate this to be more than enough time, if you can't finish the entire thing, think about it from a product perspective: Life happens, and sometimes assignments are more complicated than we expect - but the customer still wants something to use. Focus on making a minimum viable app. Cut down on functionality if you need to, fulfilling the basic need of the app and delivering working code which could be expanded later.

Please upload your repository (or repositories) to github and email me a link to the code by the agreed upon time.

If you have any questions, feel free to contact me by email. I will try to answer as promptly as possible, but I cant guarantee.