Beev®

Technical Test

A solar farm wants to analyze their output according to the luminosity index that their sensors are reporting.
The values ranging from 0.01 to 19.99.

a. We need a simple dashboard with a button that will fetch the latest values and show them as follows:

Timestamp	13:04:43	…
Luminosity Index	14.22	…

Using the backend framework of your choice, implement the call that will fetch the latest values from the database and update the dashboard.
Attached is a script that will generate the values and store them in an sqlite database that you can interact with. Run it to simulate real time data output and fill your db.

b. To Optimize output forecast the solar farm manager needs to analyze the luminosity patterns and find the time of day with the best production potential.
   For that we need to find the highest value subset of 5 consecutive values and highlight them in the dashboard.

Add a second button that will call another route on the backend that will calculate the highest value subset from the data and return the values that will be highlighted on the dashboard.

c. We want to optimize the system to be able to handle more enriched data and real time updates and historic highest value subsets.Provide a high level system design that can easily scale and handle historical data, what do you think are the main challenges when implementing such a design?

```
const sqlite3 = require('sqlite3').verbose();

// Connect to the SQLite database
const db = new sqlite3.Database('database.db');

// Create the 'numbers' table if it doesn't exist
db.run(`CREATE TABLE IF NOT EXISTS luminosity (
 value REAL,
 timestamp DATETIME DEFAULT (DATETIME(CURRENT_TIMESTAMP, 'localtime'))
)`);

function generateRandomNumber(min, max) {
 return (Math.random() * (max - min) + min).toFixed(2);
}

setInterval(() => {
 const randomNumber = generateRandomNumber(0.01, 19.99);
 db.run('INSERT INTO luminosity (value) VALUES (?)', randomNumber, function (err) {
   if (err) {
     console.error('Error inserting value:', err);
   } else {
     console.log(`Inserted ${randomNumber} into the database.`);
   }
 });
}, 2000);
```