ZapZap Fleet Inc.

Introduction

Welcome to your work experience project! You will be taking on the role of a software developer working at “ZapZap Fleet Inc.”, a company specialising in managing and optimising a fleet of electric and hybrid vehicles. Your task is to develop a backend system to analyse some data collected from the vehicles onboard sensors.

The data from the vehicles is stored in a MongoDB database, and you will be using Node.js javascript to query and analyse this data.

Data Overview

The database contains two collections called “vehicle” and “vehicle_data_history” with documents that represent the vehicles and their associated frames of data collected over time. Each document has the following structure:

vehicle:

```
Unset
{
    "_id" : "string",
    "assetType" : "string",
    "make" : "string",
    "model" : "string",
    "name" : "string",
    "registration" : "string",
    "transferred" : "boolean",
    "visible" : "boolean",
    "status" : "string",
}
```

vehicle_data_history:

```
Unset
{
    "_id" : "string",
    "vehicle_id": "string",
    "timestamp" : "ISODate",
    "speed" : "number",
    "distance" : "number",
    "battery_level" : "number",
    "temperature" : "number",
}
```

Local Setup

1. Install and setup Git
    a. https://github.com/git-for-windows/git/releases/download/v2.45.2.windows.1/Git-2.45.2-64-bit.exe
    b. Continue through installer choosing all of the default settings
    c. Then navigate to https://github.com/Lachlancg/zapzapexercise and fork the project to your own Github account and clone it to your local machine
2. Install and setup dependencies
    a. VSCode https://code.visualstudio.com/docs/?dv=win64user
    b. Node.js 18 https://nodejs.org/dist/v18.20.3/node-v18.20.3-x64.msi
3. Download MongoDB
   https://fastdl.mongodb.org/windows/mongodb-windows-x86_64-6.0.15.zip
    a. Extract .zip contents to a new directory
    b. Unzip the data.zip from the git repo into the directory next to the bin directory
    c. Open a cmd terminal in the bin directory, and run the following command

```
Unset
mongod.exe --dbpath="..\data\db"
```

4. Open the *zapzapfleet* project using VSCode
5. Run the following commands in the terminal

```
Unset
//install required packages
npm i
//test everything is setup correctly
npm run test
```

Tasks

Below you have the tasks to be completed, you do not have to complete all of them, just see how far you get.

The first section of the tasks involves creating some functions to perform some actions on the data we have.

1. You have been given some code that reads in all of the cars from the DB, you must expand the existing code to print out each vehicle sorted alphabetically on the make field. (hint: be careful, some of the data might be corrupted)

```
Unset
npm run task1
```

    a. Fix the unit tests for task1 so that they start passing, you can run them using the command below.

```
Unset
npm run tests-task1
```

2. Next we need to create a new function which should return a document containing all the distinct values of “make” of vehicle with a count of the number of vehicles of that make in the form:
   {“Tesla” : 21, “Nissan” : 123, “<ANOTHER_MAKE>” : X...}
    a. Fix the unit tests for task2 so that they start passing.

3. Create a function which will modify each record where the “status” field is “INACTIVE” so that the “visible” field is changed from true to false. You should then return the result as a new array.
    a. Create unit tests for task3

4. Create a new function that reads in data from the vehicle_data_history collection and calculates overall distance and average speed travelled by each vehicle using the timestamp between frames, and returns it in JSON format.

5. Create a new function that calculates the battery consumption rate per km distance: BCR=Change in battery level/Distance travelled

6. Create a function that calculates how much further a vehicle can travel with the battery percentage it has left.

Now that we have created some functions the next section involves creating a small web app with a REST API backend.

We will be using express.js as the web application framework. There is an example of how to use the express.js framework in the “hello-world.js” file within the examples directory. It shows you how we

can define an app which listens on a certain port and then defines various REST endpoints of different types such as get, post, etc…

It also shows an example of how to serve a HTML page and populate a table with some data using an REST endpoint.

7. You should create a new .js file, and use the example to create an REST API with a GET endpoint which returns a string containing the phrase “Hello World”, the endpoint should have the path “/hello-world”.
    a. Start your new server using node, you can follow the example script in the package.json file. In your browser navigate to http://localhost:3000/hello-world and you should see the result of your new endpoint

8. Now you can have a go at creating your own basic website using the example as a base. Once you have the server running you should navigate to http://localhost:3000/

9. Next you could try adding some more columns from the data in the vehicle collection.

10. Now it would be cool to create a button to set the visibility field of the vehicles, you could even hide them/grey them out when visible is set to false.

11. After that you could try something a bit more complex and start to introduce new functionality using the functions you have previously created.

Now we have a working website that shows off some of the work you’ve done up to now, we could start to expand and add some more functionality, such as a way to edit the information for existing vehicles, a way to add new vehicles and a way to generate mock data for those vehicles. You could even look into displaying the historical data in a nice easily digestible format, such as a graph.