Car Management System

The **system** allows the user to **create** new cars, **update** and **view** the details of existing cars and **delete** existing cars.

The code for the system/project has **two** parts:

- **Client (Web Application)**

  It is the **end-user** facing part of the car management system.

  Technologies used: HTML, CSS, JavaScript, React, Redux Toolkit, NPM, Create React App and Node.js

  The code for the web application resides in the **client** folder. You can run the project by navigating into the folder. Install the NPM packages (***npm install***) and then, run the frontend web server (***npm start***). You can also, run the tests on the terminal (***npm test***).

  *** Also, you need to setup and run the **backend** application before you setup and run the web application as the web application is consuming API endpoints exposed by the backend application.

  *** Due to **time constraints** (**could be matter of subject in the interview**):

  - Couldn't write **integration tests** (the good approach) for Redux connected React components in the **features** folder. But there are some **unit tests** for some React components in the in the **components** folder.

  - Didn't **implement dynamic input suggestions for car make and model fields for a smoother user experience.**

  But all of the **other requirements** as specified in the **technical assignment** are covered in a possible and applicable manner.

- **Server (Backend)**

  It is the **core** of car management system. It exposes the required **API endpoints** for the web application.

  Technologies used: JavaScript, RESTful APIs, Node.js, Express.js, MongoDB and Postman

  The code for the backend application resides in the **server** folder. Please follow the **README** file located in the project directory in order to setup and run the backend. The code for the backend is written from scratch in order to expose API endpoints for the web application.

  The backend is using **MongoDB** as the **database** which is hosted on the **cloud**. So, there is no need to install the database locally. Also, there aren’t any **IPs** that are **blacklisted** on the **database server** which implies that it is **open** to **public,** at the moment.