Get Car manual

This manual is for get car website services and it contains information about using this website, this manual is divided into two parts as follows:

- *The user : contains information about which kinds of users and how this website can help them.*
- *Developer: contains information about the technologies that we used to build this website and about the content of the code of this website.*

The user:

This application helps two kinds of people:

***1. Car owner*:**

This application is an easy way to find the best deal to your car. At first, register in this app by “sign up” choice in the header of the homepage. After that you can upload your car photo and description by pressing the “add” button in the homepage. Then your car will be on our app and customers can see it and contact with you by comments or your email and phone. You can also add a car for renting and determine the car properties, the rent price for each day, place to pick up, return the car and the image as well.

2. *Customer*:

This app helps you find your suitable car according to your limitations. You can see all the cars we have in the app homepage and display the owner information to contact with him or post a comment to him. The easy way to find your target is the “filter” on the left of app homepage. You can fill your limitations and it will give you the specific cars that have the entered properties, you can see the rental cars and do comments or reviews for a certain car.

The developer:

We welcome you in this app and hope to enjoy working with us.

-Main information:

This app is programmed by using:

a. For back-end: Express.
b. For front-end: Angularjs.
c. For database: Mongoose.
d. For design: Bootstrap.

The homepage is the main part of the app. It has many **components**, **sign up**, **log in**, **filter** and all **cars** in our database. Also after you log in, you will have **log out** and **add** components. We add new components: Add rental car component, show and show rental components and profile for each user contain all the cars of this user. Each component has two files in "**client**" folder; the 'html' file in "*templates*" folder and the 'js' file in *components* folder. Also you will find in the *client* folder, '***headbar***' file which role is display the components on the homepage and '***show***' file which add each new added car on the homepage.

This app has two types of information that the user can insert. *The first one* is the inserted information to make new account in "**sign up**" which include *username, password*, *phone* and *email*. This information is saved in the **database** which is named as **db** file in the *db* folder. Then when the user **signs in**, the server will check its *username* and *password* information to be identical to the saved one in the database. *The other type* of inserted information is the car properties; which is *image*, *type*, *color*, *price*, *owner name, rental price, pickup place, return place, operation, availability* and his *phone number*. Which is saved in another database file '**carDB.js**' in *db* folder. The user has also a Schema for comments called comments Schema and all the comments will be saved there,also we have contact us schema which will save all the data from the contact us template in the data base.

In the **server** part, definitely, it has two actions, *get* and *post*. **Sign up**, **log in** and **add** uses *post request*. On the other hand, display **car** and **log out** uses *get request*. In the l**og in**, it is made with sessions and hashing as default for security and easy use. The add rental and rent parts are using a post/get methods to add and display the rental cars to the data base.

Dependencies: To make this app we need to install:
Bcrypt-nodejs, express, express-fileupload, express-session, live server, nodemon, mongoose, multer and path.