

Environment Setup

1. Backend

To run this program, the following things are required:

- PHP (we use PHP 7.2.12)
- Composer
- Any relational DB (ex: MySQL/PostgreSQL)

After cloning/ downloading the project, go to the project folder (carShareBackend) from cmd/terminal and run these following commands

```
composer install
php artisan key:generate
php artisan jwt:secret
```

A database is needed. Create your DB, but don't create any table. The tables will be migrated from the code.

To do that, you will need a .env file to connect to the db. Just change the .env.example file name in the root directory to .env

In the .env file, complete your DB credentials. Our team uses MySQL DB for this project.

```
DB_CONNECTION= the DB you use (mysql/pgsql/etc)
DB_HOST=127.0.0.1
DB_PORT= your DB port
DB_DATABASE= your DB name
DB_USERNAME= your DB username
DB_PASSWORD= your DB password
```

After that, run the following command to migrate the DB

```
php artisan migrate
```

If want to have some data in the DB table, we provide seeders. just run the following command to populate the DB.

```
php artisan DB:seed --class=UsersTableSeeder
php artisan DB:seed --class=CustomersTableSeeder
php artisan DB:seed --class=LocationsTableSeeder
php artisan DB:seed --class=CarsTableSeeder
php artisan DB:seed --class=BookingsTableSeeder
```

2. Frontend

Prerequisites:

- Nodejs
- NPM

After cloning the source code, run the following commands on a terminal session on the current folder

```
npm install
```

The command will download all dependencies required to run the frontend. No further setup is necessary

Running the Program

1. Backend

To run the program, use the following command

```
php artisan serve
```

If the command is successfully running, it will say that the program is running on port 8000 of your localhost (in windows, it might be different in other OS). Now you can test the APIs. to see the list of available APIs use the following command

```
php artisan route:list
```

Testing

Some test scripts are available inside the testing folder. We have both unit and feature testing. To run the test script use the following command

```
vendor\bin\phpunit tests\<Feature|Unit>\classname.php
```

2. Frontend

Development mode

To run the program in development mode, which allows hot reloading, and other development specific features, use the following command

```
npm run dev
```

If successful, the program will run in your localhost at port 4200.

Production mode

To run the program in production mode, use the following commands

```
npm run build
```

```
npm start
```

If successful, the program will run in your localhost at port 8080.

Important change logs

1. Sprint 1

Backend

Date	Changes
17/08/2019	finish and deploy all sprint 1 backend features including the related seeders and test scripts

Frontend

Date	Changes
16/08/2019	Implement and deploy sprint 1 frontend features.

2. Sprint 2

Backend

Date	Changes
26/08/2019	Modify DB design, add image_path to cars table
05/09/2019	Change the method to store image because heroku does not accept file upload, implementing base64 encoding in all functions related to image
11/09/2019	Modify DB design, change coordinate into latitude and longitude in locations table
15/09/2019	finish and deploy all sprint 2 backend features including the related seeders and test scripts

Frontend

Date	Changes
27/08/2019	File reorganization, set CI/CD on master branch
27/08/2019	Implement and deploy sprint 2 frontend features.

3. Sprint 3

Backend

Date	Changes
22/09/2019	Modify some models relationships

Frontend

Date	Changes
22/09/2019	Implement and deploy sprint 3 frontend features.

4. Sprint 4

Backend

Date	Changes
01/10/2019	Modify cars table because the price of each car was forgotten, add price_per_day column
02/10/2019	Add booking_status column to bookings table
10/10/2019	Modify how paypalController works so the frontend can be easier to implement
15/10/2019	Change from mailTrap to sendGrid for resetting password so the application can send email to client for real

Frontend

Date	Changes
15/10/2019	Add map display on add booking form. Reset function functionality.

Links

1. Github

Backend: <https://github.com/sefirakarina/vroom-backend>

Frontend: <https://github.com/stellarista/vroom-frontend>

2. Deployment Routes

a. Backend:

Most of the backend routes won't be available to open on its own (not from frontend) due to the middleware we apply on those routes.

AUTH

login: <https://powerful-sea-28932.herokuapp.com/api/auth/login>

logout: <https://powerful-sea-28932.herokuapp.com/api/auth/logout>

ADMIN

admin register: <https://powerful-sea-28932.herokuapp.com/api/users>

List of all admins: <https://powerful-sea-28932.herokuapp.com/api/users>

delete admin: <https://powerful-sea-28932.herokuapp.com/api/users>

CUSTOMER

register: <https://powerful-sea-28932.herokuapp.com/api/customers>

get all: <https://powerful-sea-28932.herokuapp.com/api/customers>

get active cust: <https://powerful-sea-28932.herokuapp.com/api/users/customers/active>

get inactive cust: <https://powerful-sea-28932.herokuapp.com/users/customers/inactive>

get by id: <https://powerful-sea-28932.herokuapp.com/api/customers/{id}>

activate status: <https://powerful-sea-28932.herokuapp.com/users/activate/{id}> (PUT METHOD)

update: <https://powerful-sea-28932.herokuapp.com/api/customers/{id}>

CAR

get all cars with their location details: <https://powerful-sea-28932.herokuapp.com/api/cars>

create: <https://powerful-sea-28932.herokuapp.com/api/cars/>

delete: <https://powerful-sea-28932.herokuapp.com/api/cars/{id}>

update: <https://powerful-sea-28932.herokuapp.com/api/cars/{id}>

show by availability: <https://powerful-sea-28932.herokuapp.com/cars/availability/{0/1}>

LOCATION

create: <https://powerful-sea-28932.herokuapp.com/api/locations/>

show: <https://powerful-sea-28932.herokuapp.com/api/locations/>

show by id: <https://powerful-sea-28932.herokuapp.com/api/locations/{id}>

delete: <https://powerful-sea-28932.herokuapp.com/api/locations/{id}>

update: <https://powerful-sea-28932.herokuapp.com/api/locations/{id}>

BOOKING (0->ongoing, 1->future booking)

create: <https://powerful-sea-28932.herokuapp.com/api/bookings/>

show: <https://powerful-sea-28932.herokuapp.com/api/bookings/>

show by id: <https://powerful-sea-28932.herokuapp.com/api/bookings/{id}>

update: <https://powerful-sea-28932.herokuapp.com/api/bookings/{id}>

delete: <https://powerful-sea-28932.herokuapp.com/api/bookings/{id}>

see booking by customer id: <https://powerful-sea-28932.herokuapp.com/api/bookings/customer/{id}>

see all booking by status: <https://powerful-sea-28932.herokuapp.com/api/bookings/status/{0/1}>

see customer's booking by status: <https://powerful-sea-28932.herokuapp.com/api/bookings/status/{id}/{status}>

show booking dates of a car (to mark the calendar): <https://powerful-sea-28932.herokuapp.com/api/bookings/cars/unavailableDates/{id}> - carId

activate a booking: <https://powerful-sea-28932.herokuapp.com/api/bookings/status/activate/>

deactivate a booking: <https://powerful-sea-28932.herokuapp.com/api/bookings/status/deactivate/>

change payment status: <https://powerful-sea-28932.herokuapp.com/api/bookings/paymentStatus/{id}/{status}>

HISTORY

create: <https://powerful-sea-28932.herokuapp.com/api/histories/>

show all customers' histories: <https://powerful-sea-28932.herokuapp.com/api/histories/>

show a customer's history: <https://powerful-sea-28932.herokuapp.com/api/history/{id}>
- cus id

RESET PASSWORD

request reset password email (POST): <https://powerful-sea-28932.herokuapp.com/api/password/create>

get the token to reset password (GET token is needed to reset password):
<https://powerful-sea-28932.herokuapp.com/api/password/find/{token}>

actual reset password (POST): <https://powerful-sea-28932.herokuapp.com/api/password/reset>

Paypal

create payment (POST): <https://powerful-sea-28932.herokuapp.com/api/payment/create/>

payment execution details (GET): <https://powerful-sea-28932.herokuapp.com/api/payment/execute>

History

create: <https://powerful-sea-28932.herokuapp.com/api/histories/>

show all customers' histories: <https://powerful-sea-28932.herokuapp.com/api/histories/>

show a customer's history: <https://powerful-sea-28932.herokuapp.com/api/history/{id}>
- cus id

b. **Frontend:**

Main page: <https://vroom-frontend.herokuapp.com/>

Cars list: <https://vroom-frontend.herokuapp.com/list>

Login page: <https://vroom-frontend.herokuapp.com/login>

Register page: <https://vroom-frontend.herokuapp.com/register>

Customer profile: <https://vroom-frontend.herokuapp.com/me>

Customer booking list: <https://vroom-frontend.herokuapp.com/booking/me>

Booking page: <https://vroom-frontend.herokuapp.com/booking?carId={id}>

Reset password page (send email): <https://vroom-frontend.herokuapp.com/reset/start>

Reset password page (new password): <https://vroom-frontend.herokuapp.com/reset/continue?{token}{email}>

Admin page: <https://vroom-frontend.herokuapp.com/admin-page>

Super admin page: <https://vroom-frontend.herokuapp.com/super-admin>