

Rating Application Documentation

Made by :

Sanaa BAKKAR

Soukaina Touzani

Supervisor:

Mr. Younes WADDAF

Table des matières

I.	Laravel Framework :.....	3
1.	Introduction :.....	3
2.	Installation :.....	3
i.	Server Requirements:	3
ii.	Download and install tools:	3
iii.	Import project:	5
II.	Application :.....	7
A.	Introduction :.....	7
B.	Operations:	7
a)	Login : « /login »	7
b)	User :	7
c)	Administrator:	8
C.	Browser Tests (Laravel Dusk) :.....	10
a.	Introduction :	10
b.	Installation:	10
c.	Tests:	11
Note :	11
❖	Webography:	11

I. Laravel Framework :

1. Introduction :

Laravel is free, open-source and one of the more popular PHP web framework based on the model-view-controller (MVC) architectural pattern. It is created by Taylor Otwell, intended to reduce the cost of initial development and improve quality of your code by defining industry standard design practices. Using Laravel, you can save hours of development time and reduce thousands of lines of code compared raw PHP. In our application, we have chosen Laravel 5.5 as the framework of our choice.

2. Installation :

i. Server Requirements:

The Laravel framework has a few system requirements. Of course, all of these requirements are satisfied by the Laravel Homestead virtual machine, so it's highly recommended that you use Homestead as your local Laravel development environment.

However, if you are not using Homestead, you will need to make sure your server meets the following requirements:

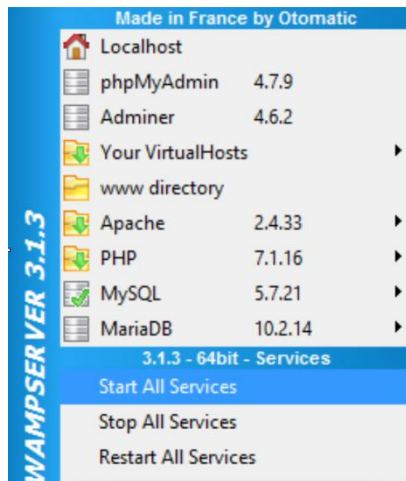
- PHP >= 7.0.0
- OpenSSL PHP extension
- PDO PHP extension
- Mbstring PHP extension
- Tokenizer PHP extension
- XML PHP extension

ii. Download and install tools:

❖ Download:

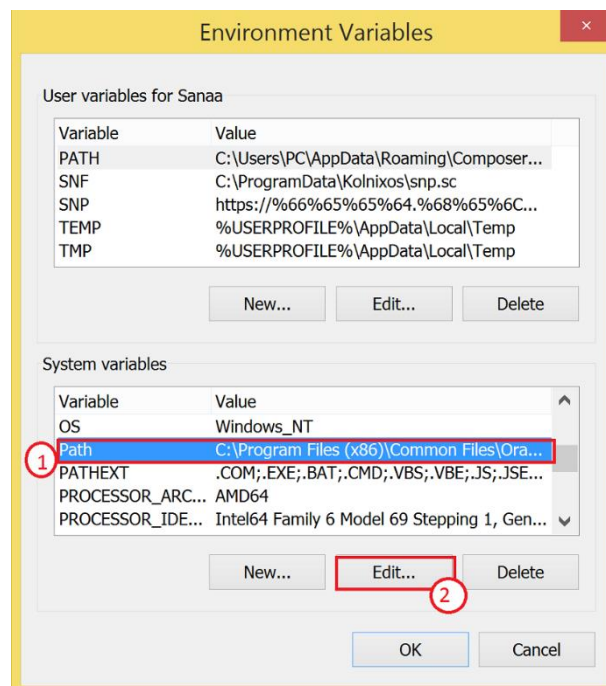
- WampServer 3.1.3 :
<https://sourceforge.net/projects/wampserver/files/WampServer%203/WampServer%203.0.0/>
- Composer : <https://getcomposer.org/download>
- Development tool of your choice, we used Sublime Text:
<https://www.sublimetext.com/3>
- Log Viewer:
<https://github.com/ARCANEDDEV/LogViewer/blob/master/docs/1.Installation-and-Setup.md>

To test the server, run WampServer and start all services.



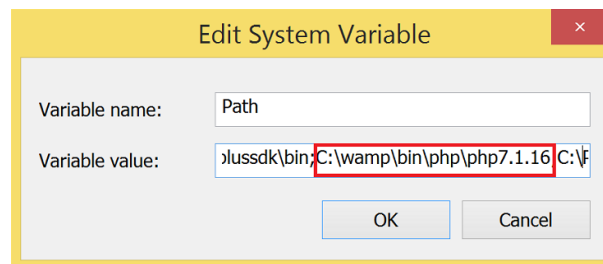
❖ Configuration:

we will need to use PHP on the command line. For Windows users, you have to change the environment variables. To do this, open the window below by making a search for "environment variables" (shortcut: home + S).



Click on the Path line and the Edit button.

Then put the path of your PHP installation in WampServer and validate:



❖ Information:

Here we present the general commands in Laravel 5:

- ✓ The help command is used to obtain information about a command, it also displays the possible parameters that can be passed to it: `php artisan help nameofCommand`
- ✓ List all the commands proposed by Laravel 5: `php artisan list`
- ✓ List all the routes in our application: `php artisan route:list`
- ✓ Delete the cache: `php artisan cache:clear`
- ✓ The serve command: `php artisan serve`

Moving to the commands dedicated to the database which you will need:

- ✓ Migration, modifies the database from the migration files: `php artisan migrate`
- ✓ Returns to the previous migration: `php artisan migrate:rollback`
- ✓ Update the database from our migrations: `php artisan migrate:refresh`

By migrating, rollbacking and changing the names of migration files, we may encounter an error due to the migration preload cache system. To resolve this problem, you must perform both of the following commands:

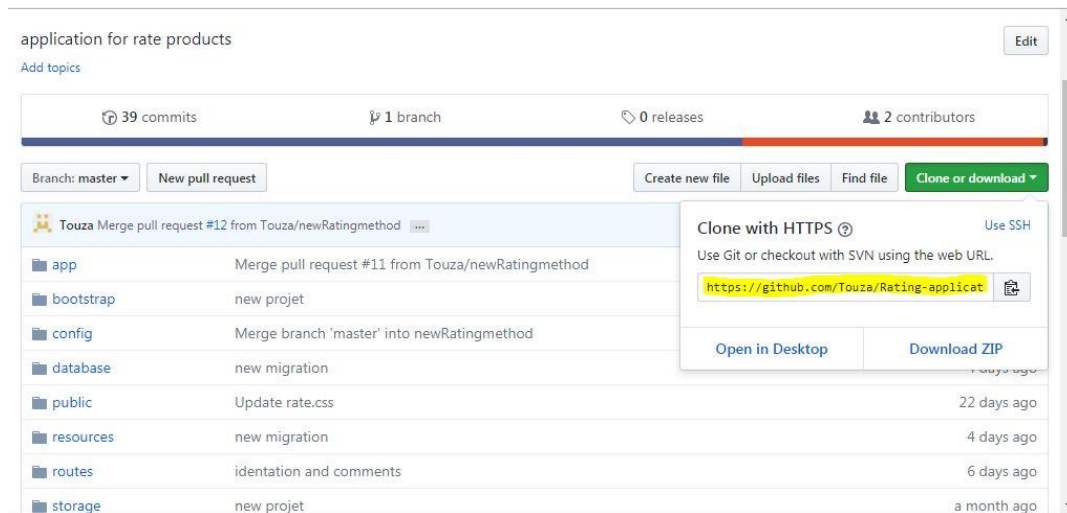
- ✓ `php artisan clear-compiled`
- ✓ `php artisan optimize`

If the problem persists, try `composer dump-autoload` between the two previous commands

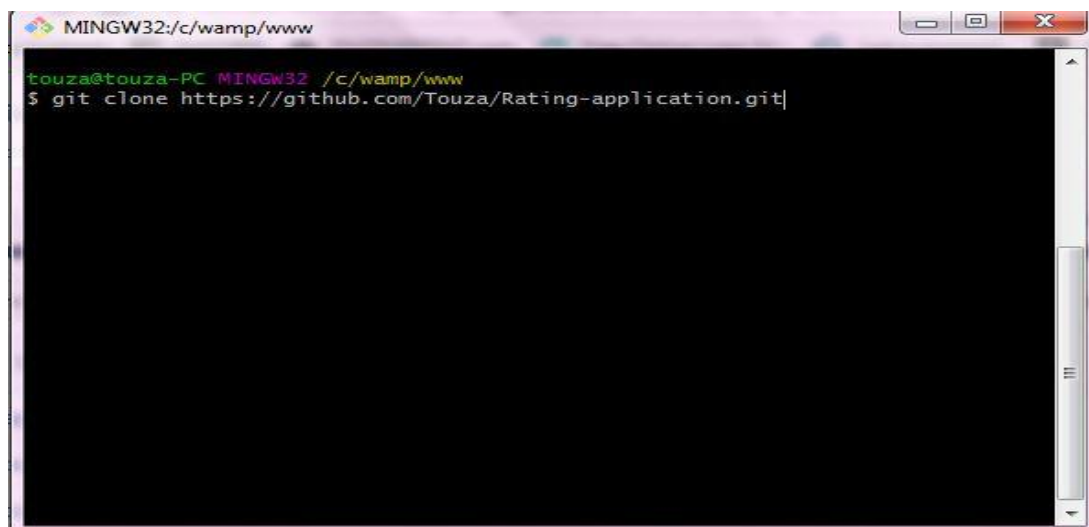
iii. Import project:

We publish the project as a repository in GitHub, so to import it just use the following line command: <https://github.com/Touza/Rating-application.git>

❖ Clone the project:



Copy the web URL from GitHub repository, from terminal or CMD using cd command go to the www directory for WampServer or httdocs for Xamp, type the command like the screenshot.



❖ Install dependencies:

- Go to the folder application using cd in CMD or terminal
- Run **composer install** on you CMD or terminal
- Copy **.env.example** file to **.env** file on the root folder, you can type **copy .env.example .env** if you using command prompt windows or **cp .env.example .env** if you using terminal.

- Open your `.env` file and change database name (`DB_DATABASE`) to whatever you have, username (`DB_USERNAME`) and password (`DB_PASSWORD`) field correspond to your configuration.
- Run `php artisan key:generate`.
- Run `php artisan migrate`.
- Run `php artisan serve`.

II. Application :

A. Introduction :

The idea to realize this application is to allow the different users of a company to evaluate a product according to different criteria in order to make an overall evaluation with each product.

Regarding admin role, he has the possibility of managing the products and keep track of each one of the logs files (alerts, errors, Info)

B. Operations:

a) Login : « /login »

❖ *Migrations files :*

Rating application has one page for login from where both admins and users can join the home page dedicated to each one, thanks to one attribute `admin` in users model.

```
$table->boolean('admin')->default(0);
```

For Users the admin attribute receives the default value 0, and 1 for Admins.

❖ *Redirection:*

In order to discern each home a method redirection is been defined.

```
if (Auth::user()->admin == 0) Redirect to "user home" else Redirect to "admin home".
```

b) User :

1) *Rating Product (car) : « /ratecar »*

i. *Dynamic dependent select box :*

To rate a product (a car in this case), a dynamic dropdown menu is done to select make of car and according to that choice user select a car model in order to make the rating operation easier.

CAR RATING

MAKE

-- Select Make --

MODEL

-- Select Model --

First, all makes data are displayed in first select box and when any make is selected from make dropdown then related model will be listed in second dropdown.

In dynamic dependent car controller, a method “find_model” is been defined to get a make of car and returns json response.

```
$car = Car::where('make_car',$make_car)->get();  
return response()->json($car);
```

In the views folder a script is created and using Ajax to get the returned response from method “find_model” and send the model corresponding to the make in the second dropdown.

ii. Get Rate Method:

After the user select make and model of a car, he sets rate according to different criteria:

Comfort:	★	★	★	★	★
Mecanism:	★	★	★	★	★
Price:	★	★	★	★	★
Design	★	★	★	★	★

[RATE PRODUCT](#)

After the rate is calculated and stocked in rating column in table “ratings” with id of product selected and id of user who rated the product.

After this operation we calculate the “General rate” is the summons of all rating column values divided with summons of users who rated the product.

At the end there is two possibilities:

- User select all criteria ➡ Redirect to home with success message.
- User did not select all criteria ➡ Redirect to rate product page with error message.

c) Administrator:

1. List of products (car in this case): “/listofcars”

The administrator is able to create, update and delete any product, can also supervise all the user’s activities due to Log Viewer.

List of cars show admin cars in the database and the actions that can do.

RATING APPLICATION				
CREATE A NEW PRODUCT ▾ JOIN LOG VIEWER LOGOUT				
id	Make	Model	Rate	action
7	Fiat	500	2.75	Update Delete

i. Log Viewer:”/Log-Viewer”

You can download and install Log Viewer from web URL in above.

Log Viewer allow admins to know the user and his rate and the product rated, with these lines of code putted in “getRate” method:

```
Log::info('Car:'.$_POST['make_car'].' '.$_POST['model_car'].' Rating :'.$avg_rate.' by user : '.Auth::user()->name);
```

Admins also receives alert if a user logged in the application:

```
Log::alert($user->email.' is logged out');
```

Log Viewer manage the errors and show them to the admin:

ENV	Level	Time	Header	Actions
local	Error	15:14:17	Undefined variable: errors (View: C:\wamp\www\blog - Copy\resources\views\ratephone.blade.php) {"exception":"[object] (ErrorException(code: 0): Undefined variable: errors (View: C:\wamp\www\blog - Copy\resources\views\atephone.blade.php) at C:\wamp\www\blog - Copy\storage\framework\views\44a02110e1228b3e46bc05ebc2a783a8070be58a.php:22, ErrorException(code: 0): Undefined variable: errors at C:\wamp\www\blog - Copy\storage\framework\views\44a02110e1228b3e46bc05ebc2a783a8070be58a.php:22)	Stack
local	Error	15:14:17	Undefined variable: errors (View: C:\wamp\www\blog - Copy\resources\views\ratecar.blade.php) {"exception":"[object] (ErrorException(code: 0): Undefined variable: errors (View: C:\wamp\www\blog - Copy\resources\views\atecar.blade.php) at C:\wamp\www\blog - Copy\storage\framework\views\023a5292c72bf3921b9f9456c550adc6740bcc07.php:22, ErrorException(code: 0): Undefined variable: errors at C:\wamp\www\blog - Copy\storage\framework\views\023a5292c72bf3921b9f9456c550adc6740bcc07.php:22)	Stack

ii. CRUD operations:

- Create a product (car): “/createcar”

Create a new car

Make

Model

CREATE

Admin sets a make and model of a car, and the id attribute is auto-increment rating attribute is nullable.

When create button is clicked there is two possibilities:

- Admin set a car already exist ➡ Redirect to list of cars page with error message "car already exists" and the car is not created.
- Admin set a new Car ➡ Redirect to list of cars page with success message "car created" and the new car in the list with rating attribute NULL.

➤ Update Product (car) : « /updatecar /{id} »

As create product admin update a car change make and model of car attributes if there's any mistake.



The form consists of two text input fields. The first field is labeled "Make" and the second field is labeled "Model". Below the "Model" field is a blue button with the text "UPDATE" in white capital letters.

Sets the correct values and clicks update buttons, the car will be updated in the database and the admin redirected to lists of cars with success message.

➤ Delete product (car): "/deletecar/{id}"

Admin choose a car and delete it, page refreshed and car is deleted from database and from the list of cars, and danger message "cars deleted".

C. Browser Tests (Laravel Dusk) :

a. Introduction :

Laravel Dusk provides an expressive, easy-to-use browser automation and testing API. By default, Dusk does not require you to install JDK or Selenium on your machine. Instead, Dusk uses a standalone ChromeDriver installation. However, you are free to utilize any other Selenium compatible driver you wish.

b. Installation:

To get started, you should add the Laravel/dusk Composer dependency to your project:

```
composer require --dev laravel/dusk:"^2.0"
```

Once Dusk is installed, you should register the `Laravel\Dusk\DuskServiceProvider` service provider. Typically, this will be done automatically via Laravel's automatic service provider registration.

If you are manually registering Dusk's service provider, you should never register it in your production environment, as doing so could lead to arbitrary users being able to authenticate with your application.

After installing the Dusk package, run the dusk:install Artisan command:

```
php artisan dusk:install
```

A Browser directory will be created within your tests directory and will contain an example test. Next, set the APP_URL environment variable in your .env file. This value should match the URL you use to access your application in a browser.

c. Tests:

In the test folder a new folder called browser will be created after install Laravel dusk, in this folder some files created to test login, admin site, and user site (rating operation).

To run these tests, use the dusk Artisan command. The dusk command accepts any argument that is also accepted by the phpunit command:

```
php artisan dusk
```

```
C:\wamp\www\Rating-application>php artisan dusk tests/Browser/AdminTest.php
Warning: TTY mode is not supported on Windows platform.
PHPUnit 5.7.27 by Sebastian Bergmann and contributors.

[640:5296:0912/205619.709:ERROR:install_util.cc(603)] Failed to read HKLM\SOFTWARE\Policies\Google\Chrome\MachineLevelUserCloudPolicyEnrollmentToken: The system cannot find the file specified. (0x2)

DevTools listening on ws://127.0.0.1:59823/devtools/browser/e90e2bf1-7bba-4b4e-bc34-d5d75440b07f
... [10028:9320:0912/205635.015:ERROR:mf_helpers.cc(14)] Error in dxva_video_decode_accelerator_win.cc on line 517
..... 12 / 12 (100%)

Time: 41.07 seconds, Memory: 12.00MB

OK (12 tests, 19 assertions)
```

All tests passed with success.

Note :

each type of product can be rated in such a way as car.

Webography:

Somme information is taken from:

<https://laravel.com/docs/5.4>