Installation guide and API information

You need to have set-up Laravel environment and installed all the dependencies to try this in your PC.

I am working on windows 10 and I have installed XAAMP (PHP and SQL), Laravel, composer and node.js I have used Tw-elements and you might need to install it in your Laravel project for styles to work properly I have also made use of highcharts.js to plot pie chart and bar chart

After setting up environment you can follow following steps to install this application in your device:

* After Extracting the zip file open terminal or command prompt (cmd)
* Go to the extracted project folder application using cd command on your cmd or terminal
* [Optional] Rename pharmachemistapp-master folder to desired Laravel project name or just remove -master
* On your cmd or terminal run > *composer install* (make sure you have installed the composer)
* Copy .env.example file to .env on the root folder. You can type copy .env.example .env if using command prompt Windows or cp .env.example .env if using terminal, Ubuntu
* Open your .env file and change the database name (DB\_DATABASE) to whatever you have, username (DB\_USERNAME) and password (DB\_PASSWORD) field correspond to your configuration. (I have XAAMP for managing local database)
* Run > *php artisan key:generate*
* Run > *npm install* && *npm run dev* [ you need npm installed for this I have it from installing NodeJs ] (run separately in windows)
* Run > *php artisan migrate:fresh --seed*
* Run > *php artisan serve*
* Go to http://localhost:8000/ in web browser (I use XAAMP for local hosting and accessing database)
* The database can be accessed from http://localhost/phpmyadmin/ if you install XAAMP

Alternatively,

If you want to clone the GitHub repo (recommended) please use this [link](https://github.com/VoidofLimbo/pharmachemistapp). Please read the readme file present there as well. The overall process is similar.

API Links

After cloning the project to test the api you can use the following links:

I am using XAAMP so my domain will be: <http://localhost:8000> , Yours can be different if you are testing this in different environment. If you have the same setup as me then this is usual the domain.

GET|HEAD api/carehomes......................... carehomes.index › Api\CarehomeApiController@index

POST api/carehomes ………...................... carehomes.store › Api\CarehomeApiController@store

GET|HEAD api/carehomes/{carehome} ..... carehomes.show › Api\CarehomeApiController@show

PUT|PATCH api/carehomes/{carehome} … carehomes.update › Api\CarehomeApiController@update

DELETE api/carehomes/{carehome} …... carehomes.destroy › Api\CarehomeApiController@destroy

GET|HEAD api/user ..........................................................................................................................

Examples:

<http://localhost:8000/api/carehomes> GET Get all the carehomes in json

<http://localhost:8000/api/carehomes/1> GET Get carehome with id 1

<http://localhost:8000/api/carehomes> POST Save given record in carehomes table

<http://localhost:8000/api/carehomes/5> PUT Updates data having provided id in carehomes table

<http://localhost:8000/api/carehomes/100> DELETE Deletes data with given id from carehomes table

Screenshots

I am using Postman to test the API in my local environment.

# Getting all the data

A screenshot of a computer

Description automatically generated

# Getting specific data

Graphical user interface, text, application

Description automatically generated

# Insert new Data in the collection

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

# Deleting the data from the table

Graphical user interface, text, application

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

The collection of requests for has been exported and included in this submission under name [Carehome API.postman\_collection.json](Carehome%20API.postman_collection.json) which can be imported and executed after the project has been successfully cloned.