***Laravel framework***

In a nutshell, a framework make you win time give the insurance to have trustable well coded components.

Laravel is a new, well furnished framework which is using the most recent possibilities offered by php , it is well coded and organized.

There is a lot of support and documentation for this framework

Laravel can work with the MVC scheme but it is not an obligation

<https://laravel-tricks.com/tricks>

use in combination with laragon

A Manager of dependencies is used to automate procedures.

<https://getcomposer.org/>

*In a nutshell*

to install and update LAravel, you have to use the manager of dependencies :  **composer**

The creation of a Laravel app is done through the console with a simple line of command

Laravel is organized through several maps

The public map is the only one accessible by the client

The environement is fixed thanks to the file « .env »

The componant « html » must be charged independantly.

*In a nutshell*

Laravel has « .htaccess » file to simplify the writing of the url.

The routage system is simple.

Its possible to plan parameters within the roads

We can force a parameter to match a regex

We can name a road to ease the url generation and redirections

All url must be dealt with, even those who have no road planed for them

Laravel has loads of « facades » which are easing the syntax

Helpers are also there to make the writing of the syntax easier.

*In a nutshell*

Laravel offer the possibility to create views

Its possible to onyl transmit parameters to the views

The tool « Blade » allow to create templates and optimise the code of the views and redirections can easily be made

*In a nutshell*

Controlers are used to catch the requests issued by the road and also to give a response to the client

**Artisan** is there to make the creation of a controler easier

Its easy to call a method of the controller from a route

We can name a road that point to the method of a controler.

*In a nutshell*

Laraven is generating a random token associated with the form so that when the form is submitted, the token is checked to ensure the origine of the form and that allows to prevent attacks like the Cross site request forgery CSRF one.

Laravel allow to create roads with different verbs such as get, post, …

A form can easily be made thanks to the class « form »

The entries of the client are stored in the request

*In a nutshell*

Validation is an essential step when it comes to check the entries of the client

There are lots of validation rules

The validator generates explicit errors to display to the client

To get the text of the erros in French, you must fetch traductions and place them in the right file

Laravel allow the simple send of emails

*In a nutshell*

Configurations files allow to memorize easily group of key values and are handled through the « Config » facade or the helper « config »

Sessions allow to memorize datas concerning a client and are easily manipulated with the face « Session » or thanks to the helper « session »

*In a nutshell*

A manager ( controler) must delegate every tasks that is not related to its domain of competencies.

The injection of dependencies allow to seperate tasks, to simplify the maintenance of the code and unitary tests

The providers allow to make initialisations, in particular links of dependencies between interfaces and classes

*In a nutshell*

Data bases should be configured to work with Laravel

Migrations can intervene on the scheme of the tables of the database

Eloquent can represent tables as objects to simplify the manipulation of the recordings

You must plan the handeling of the model in a class injected within the controler

The debug bar gives precious intels about requests

*In a nutshell*

When migrating, the constructor of the scheme can fix all the proprieties of the fields

A resource in Laravel is composed of a controller that has the 7 methods for a complete management.

Roads towards resources are created thanks to a simple line of code

For a resource, the validation is always different between the creation and the modification and you must adapt the code to take account of that .

*In a nutshell*

Create a repository independant from the manager to access the datas allow to have a clean and easy to maintain code.

Its important to think what will be done with your code after you created it

Views must use as moch as possible the possibilities offered by Blade, helpers and the Form class to be precise and readable

Handeling of the errors must no be ill considered. You should remove the debug mode on a production site and provide explicit messages for the users according to which erros is encountered

*In a nutshell*

Authentification is fully handled by Laravel

A middleware allow to make a treatment at the start of the request

We can use the middleware « authenticate » and « redirectIfAuthenticated » to allow or to forbid an access

Laravel also has a complete system to reniew the passwords

*In a nutshell*

A **1 :n** relation required the creation of a « foreigner key » from the « n » side.

It is possible to fill the registration tables with the population

A relation within the base requires that special methods are placed within the models

With the middlewares, it is easy to handle the access to methods and controlers

*In a nutshell*

A **n :n** type relation requires the creation of a « pivot table ».

Eloquent is handeling pivots tables with adaptated methods

We can create personalised rules of messages and validation

*In a nutshell*

**Artisan** allow a great set of tools of command to generate some code. Its possible to create new commands or to extend the possibilities of existing commands

The scheme designer allow, from a convivial visual interface, to generate migrations, models, populations, managers of views

*In a nutshell*

Practical tool to generate requests. Eloquent allow to makes lots of manipulations on the tables and is at ease when it comes to relations

Querry builder is the perfect mate for Eloquent

Sometimes, we have to use raw expressions in the requests but if its the case on should think to protect against SQL injections

*In a nutshell*

You should create environements to answer the different situations : local, far away, …

The deployement part is easy and fast if you have GIT or SSH

You can delete the file « public » even if you have a functional Laravel at the same time. You should just then take measures to ensure the safety

It possible to deploy an app with the FTP buts its not pleasant to do and updates should be carefuly handeled

*In a nutshell*

To use macros allow to have independant views regarding which framework is used

A service provider can make initialisations

The creation of templates can make the organisations of the views easier

The utilisation of maps to organise views is often necessary when there is a lot of views.

*In a nutshell*

The container of dependencies is the key component of Laravel

A component is register thanks to a service provider

We simplify the accessibility sintax with a class that has a facade

*In a nutshell*

Laravel has the basic tools for the localisation

You must create localisation files in accordance with the number of languages that there are to treat.

The localisation must be done at the level of a middleware

Formating of the dates can be done easily with a « presenter »

The linguistic version of the attributs must be planed

*In a nutshell*

Ajax is easy to use with Laravel

Sterilisation can put in shape the entries of a form

You can predict a specific middle ware for the ajax requests

**Middleware** acts as a bridge between a request and a response. It is a type of filtering mechanism

**Laravel** includes a **middleware** that verifies whether the user of the application is authenticated or not.

*In a nutshell*

Laravel is using PHPUnits to make unitary tests

There are helpers to integrate the tests within an application made with Laravel

The component **Mockery** allow to stimulate the behaviour of a class and so to isolate the tests

Laravel is offering funky commands to test an application

*In a nutshell*

Laravel has a complete and simple system to handle events

Laravel has a complete and simple system to handle the management of the authorisations

<https://github.com/bestmomo/laravel5-example>