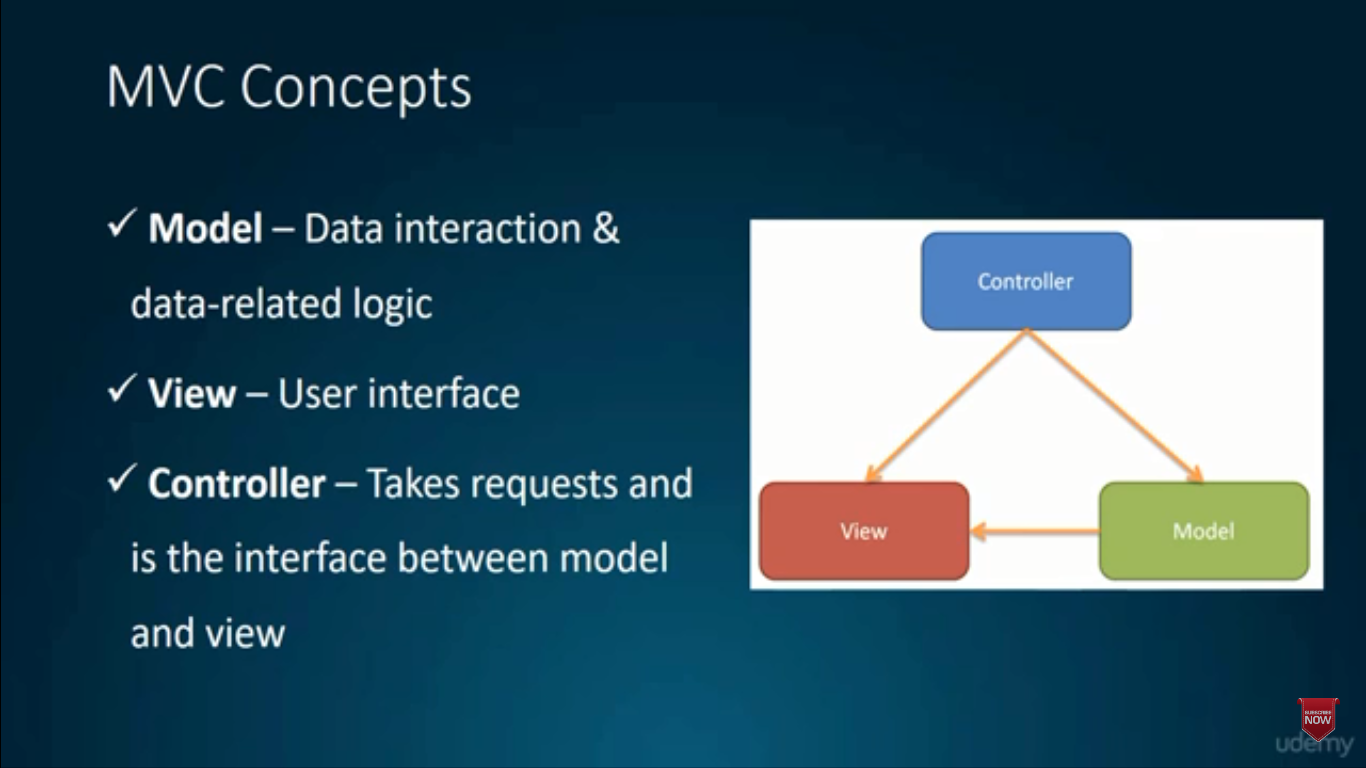
In this report, I tried to review what I knew about the Laravel framework, also learn something new. Additionally to share this with people who want to learn Laravel.

**Laravel**: is an open-source PHP web framework, which uses an MVC design pattern. It is the most popular PHP framework, and it is usually used to build database-driven dynamic websites and applications.

**MVC**: It is a design pattern as follows:



We install Laravel and build Laravel applications with **composer** which is a PHP package manager (similar to npm for nodejs).

>composer global require “laravel/installer” >Laravel new appName

**OR**

>composer create-project laravel/laravel appName

**Artisan CLI (Command Line Interface)**

This is a useful CLI, which we can make controllers, models, migrations with it. Also, we can run migration files with it (we don’t need to interact with databases). Tinker REPL is a tool that interacts with the app/database from CLI.

**Artisan CLI Command Examples**:

$php artisan list

$php artisan help migrate

$php artisan make:controller PostsController

$php artisan make:model Post –m

$php artisan make:migration add\_posts\_to\_db-table=posts

$php artisan migrate

$php artisan tinker

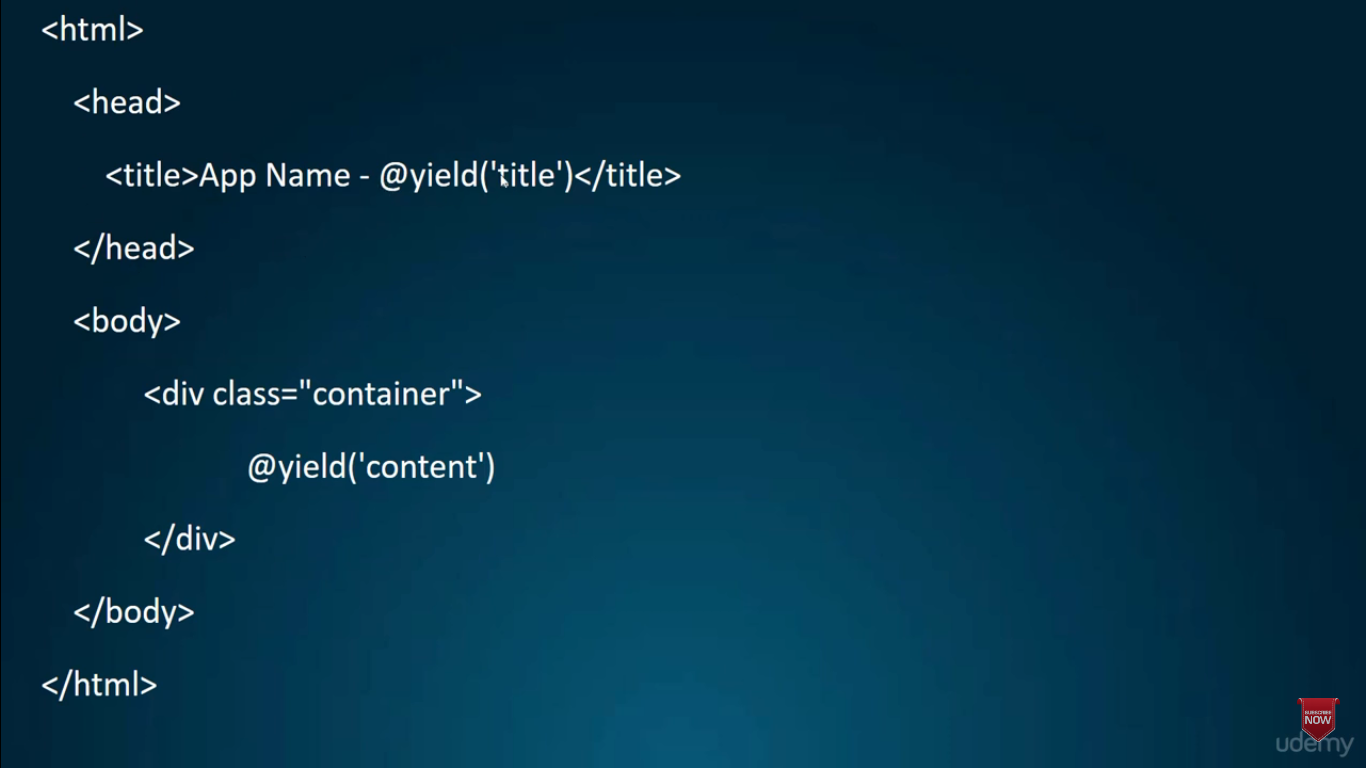
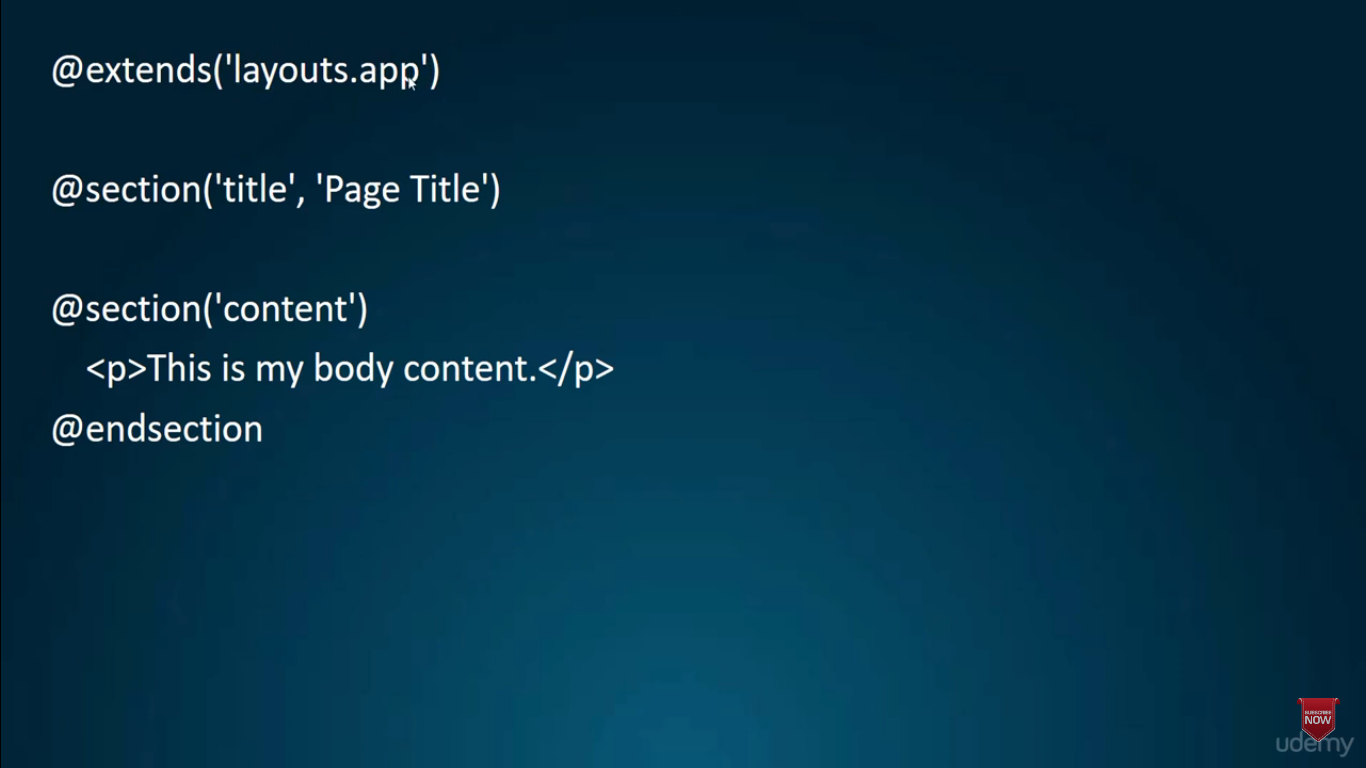
**Eloquent ORM (Object Relational Mapper/ AcrtiveRecord Implementation)**

It makes it very easy to work with the database.



**Blade templating**

It is a simple templating method but a powerful template engine. It does not restrict PHP in views, and it is compiled into plain PHP and it is cached with no overheads.

First, we have to install XAMPP which gives us an Apache server, MySQL, and PHP. Then, we just need to open XAMPP and turn on the Apache server and MySQL server to be able to access localhost/phpmyadmin.

**Note**: I had installed MySQL earlier while I was learning SQL, so I faced an error while trying to turn on MySQL. So, I turned off the MySQL service services.msc file.

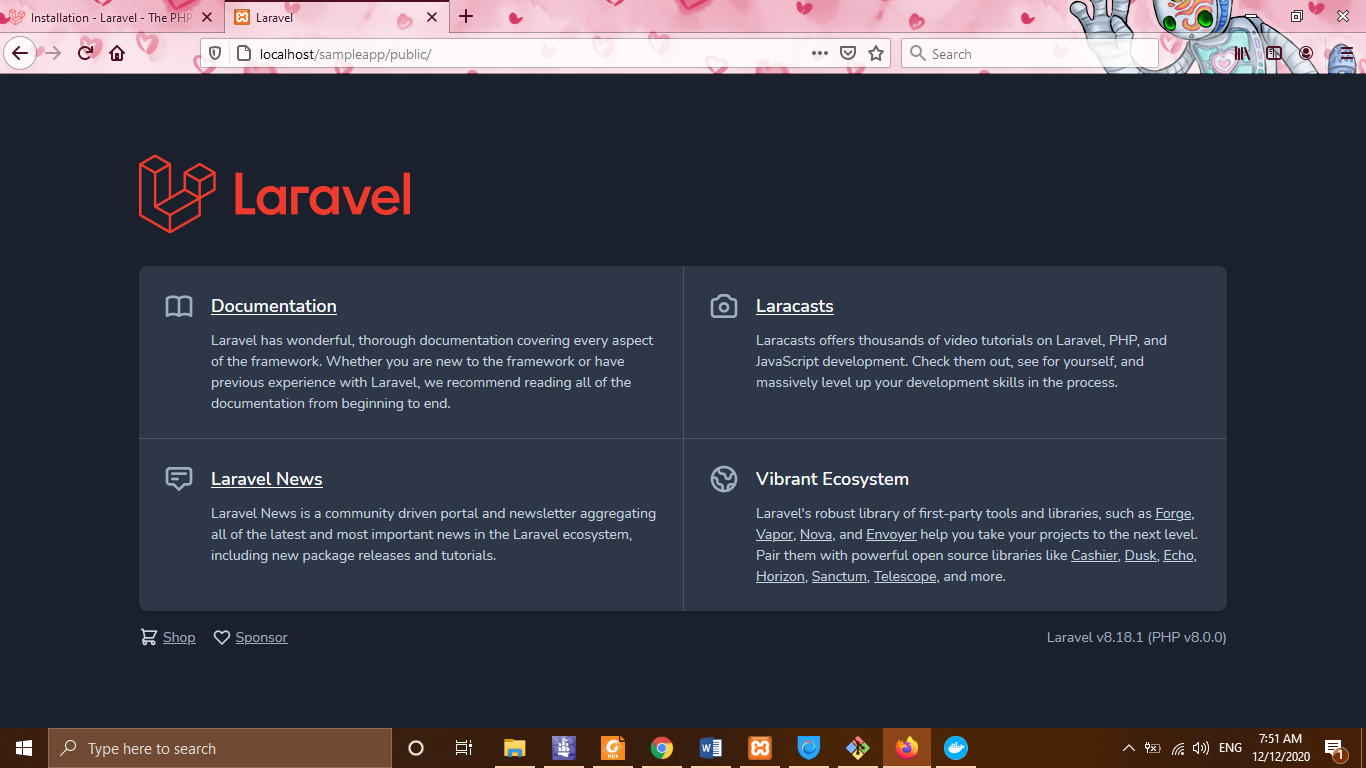
Now it is time to install Composer requirement manager.

Now, we can go and install Laravel and start developing what we like in this sweet framework. In Windows, we should be in the xampp/htdocs to run the following command.

>composer create-project laravel/laravel sampleApp

**Tip**: I faced another problem and I figured out by googling that I should have uncommented the line for extension=fileinfo in the php.ini file in the subfolder that I installed PHP.

After installing the first Laravel application, then I went to localhost/sampleApp, then clicked on public and the following image showed off.



Then, we can build our Laravel application where we like just try to run the application with the following command.

>php artisan serve

Another possibility is to install docker and use it easily without any configuration and extra work, just focus on developing the Laravel application.

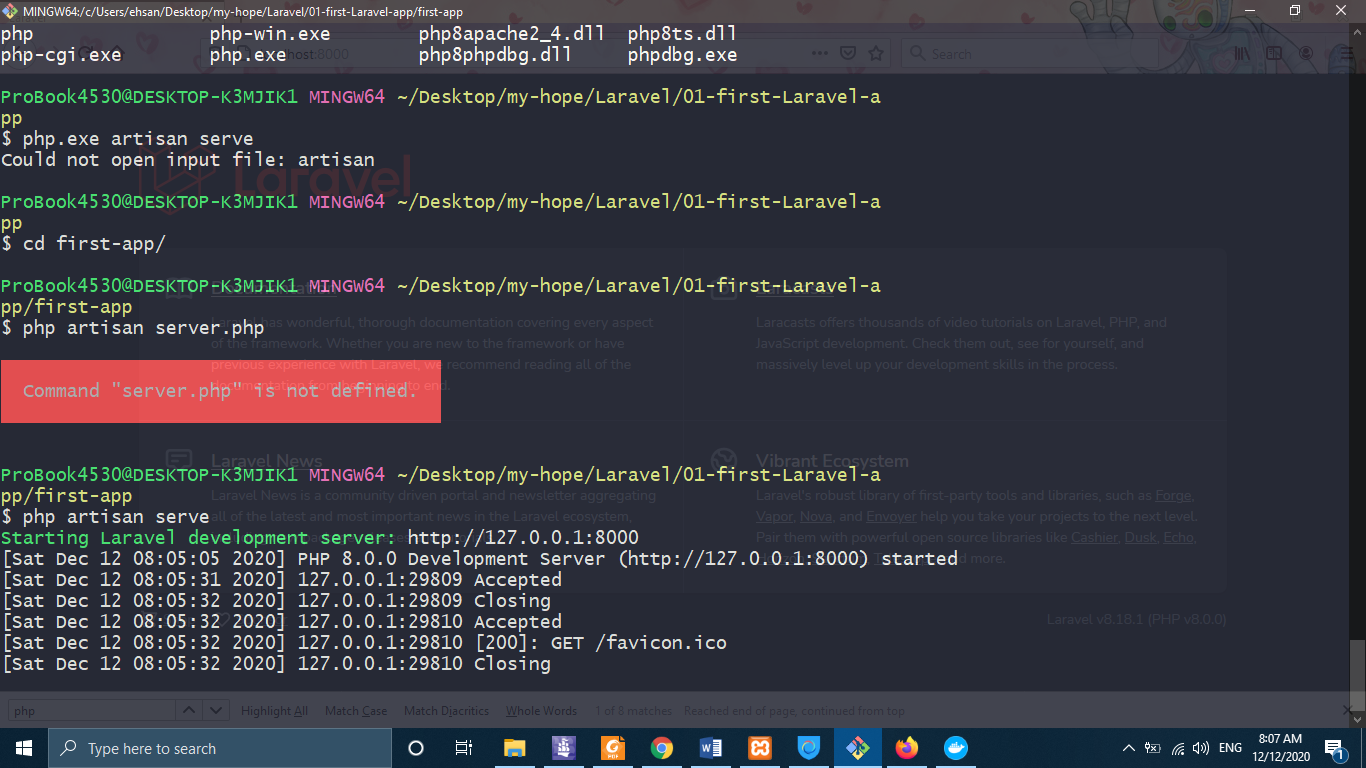
Also, we can install Laravel globally and just create projects locally without any internet connection, but this is not recommended because we have to update it while a new version comes, which is going to make requirement errors.

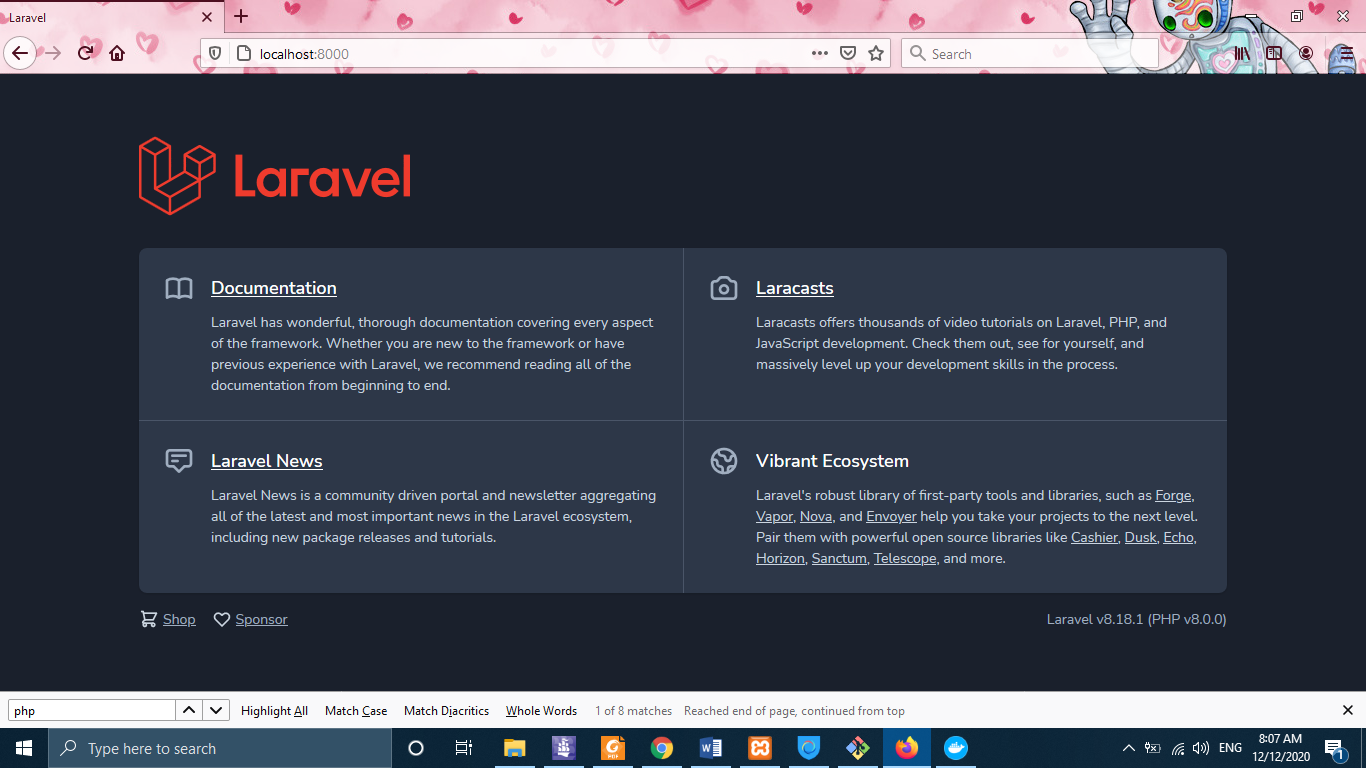
>composer global require laravel/installer

>laravel new example-app

>php artisan serve

I installed another application and run it with php artisan serve command which the following is what I can see now on my browser.





Now, we can go ahead and develop what we want easily with this powerful framework.

I started to develop a simple project on which to have a connection with the database. Through this project, I will review many concepts from route to model, etc.

After defining routes for about and contact, for avoiding code redundancy in the frontend, we build a folder layout that is going to keep the layout part of our project which is going to be the same for all pages of our project. We should keep the project modular to be able to debug it easily and with the least effort. Another example is using the structure specified by the SASS, for example, the sidebar is better to be located in the inc folder.

**Route concept**: Routing in Laravel allows you to route all your application requests to their appropriate controllers.

**View Concept**: Views hold the presentation logic of a Laravel application.

We use the following instruction to avoid code redundancy as explained above.

-------layout.app.blade.php---------

@yield(‘content’)

@include(‘inc.sidebar’)

-----about.blade.php-----

@extends(‘layout.app’)

@section(‘sidebar’)

@parent

<p>This is appended to the sidebar!</p>

@endsection

-----inc.sidebar.blade.php------

@section(‘sidebar’)

<div class=”sidebar”>

<h3> Sidebar </h3>

@show

</div>