**What is Laravel ?**

Laravel is framework of php

Framework provides its own structure

Framework provides more security

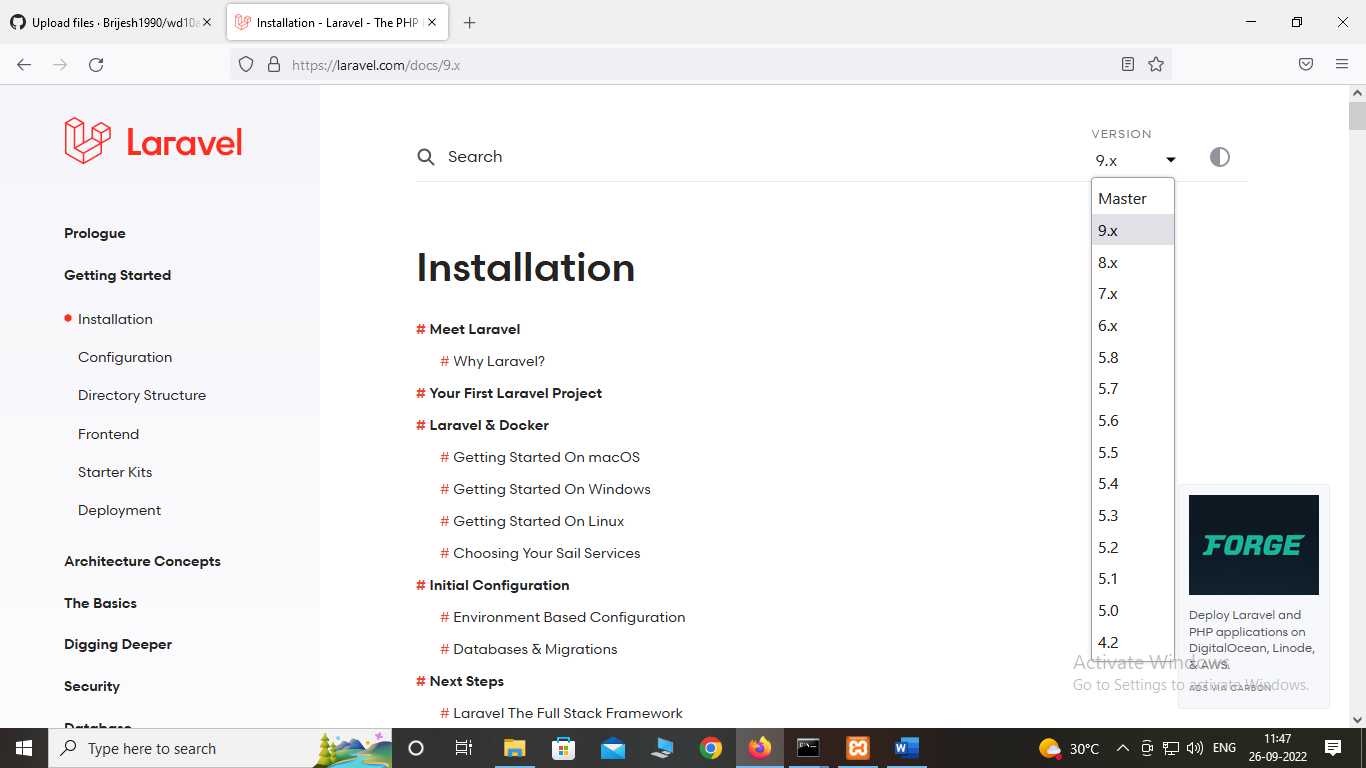
Laravel is used to completed large scale type of dynamic website or web app.

Laravel is used a MVC structure to create any website or web apps in php.

Laravel is support OOPS.

Laravel version :

Laravel latest version is 9.0



MVC structure :

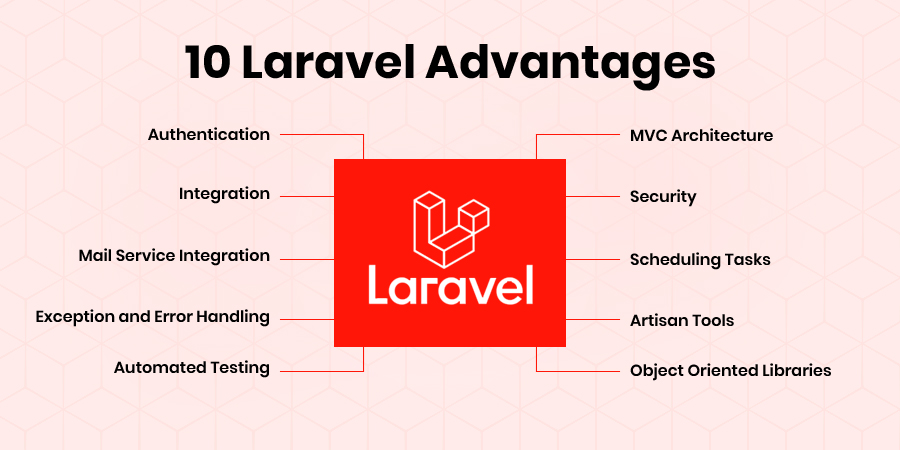
Model: model create a meber function and connect with Databas

View:view is used to load your view or UI part of applications

Controller :controller handle logic part of your applications.



Laravel advantage :



How to install Laravel :

Step 1:check your xampp or php version while used Laravel 9.0 its install on php8.1 version.

Localhost/xampp

<http://localhost/xampp/>

step 2: **Laravel install with composer**

**what is composer ?**

A composer is a dependency manager tools i.e used to install Laravel app.

A composer is install from

<https://getcomposer.org/download/>

step 3: **how to check a composer install or not in your computer**

**E:\xampp\htdocs>composer**

\_\_\_\_\_\_

/ \_\_\_\_/\_\_\_ \_\_\_\_ \_\_\_ \_\_\_\_ \_\_\_\_ \_\_\_\_\_\_\_\_ \_\_\_\_\_

/ / / \_\_ \/ \_\_ `\_\_ \/ \_\_ \/ \_\_ \/ \_\_\_/ \_ \/ \_\_\_/

/ /\_\_\_/ /\_/ / / / / / / /\_/ / /\_/ (\_\_ ) \_\_/ /

\\_\_\_\_/\\_\_\_\_/\_/ /\_/ /\_/ .\_\_\_/\\_\_\_\_/\_\_\_\_/\\_\_\_/\_/

/\_/

Composer version 2.4.2 2022-09-14 16:11:15

Step 4:how to create a Laravel app or Laravel project using composer

Cmd : composer create-project Laravel/Laravel my-laravel-crud

How to run Laravel on localhost

Public/index.php

Note: copy and paste index on main project directory open index.php and change path

<?php

use Illuminate\Contracts\Http\Kernel;

use Illuminate\Http\Request;

define('LARAVEL\_START', microtime(true));

/\*

|--------------------------------------------------------------------------

| Check If The Application Is Under Maintenance

|--------------------------------------------------------------------------

|

| If the application is in maintenance / demo mode via the "down" command

| we will load this file so that any pre-rendered content can be shown

| instead of starting the framework, which could cause an exception.

|

\*/

if (file\_exists($maintenance = \_\_DIR\_\_.'/../storage/framework/maintenance.php')) {

    require $maintenance;

}

/\*

|--------------------------------------------------------------------------

| Register The Auto Loader

|--------------------------------------------------------------------------

|

| Composer provides a convenient, automatically generated class loader for

| this application. We just need to utilize it! We'll simply require it

| into the script here so we don't need to manually load our classes.

|

\*/

require \_\_DIR\_\_.'../vendor/autoload.php';

/\*

|--------------------------------------------------------------------------

| Run The Application

|--------------------------------------------------------------------------

|

| Once we have the application, we can handle the incoming request using

| the application's HTTP kernel. Then, we will send the response back

| to this client's browser, allowing them to enjoy our application.

|

\*/

$app = require\_once \_\_DIR\_\_.'../bootstrap/app.php';

$kernel = $app->make(Kernel::class);

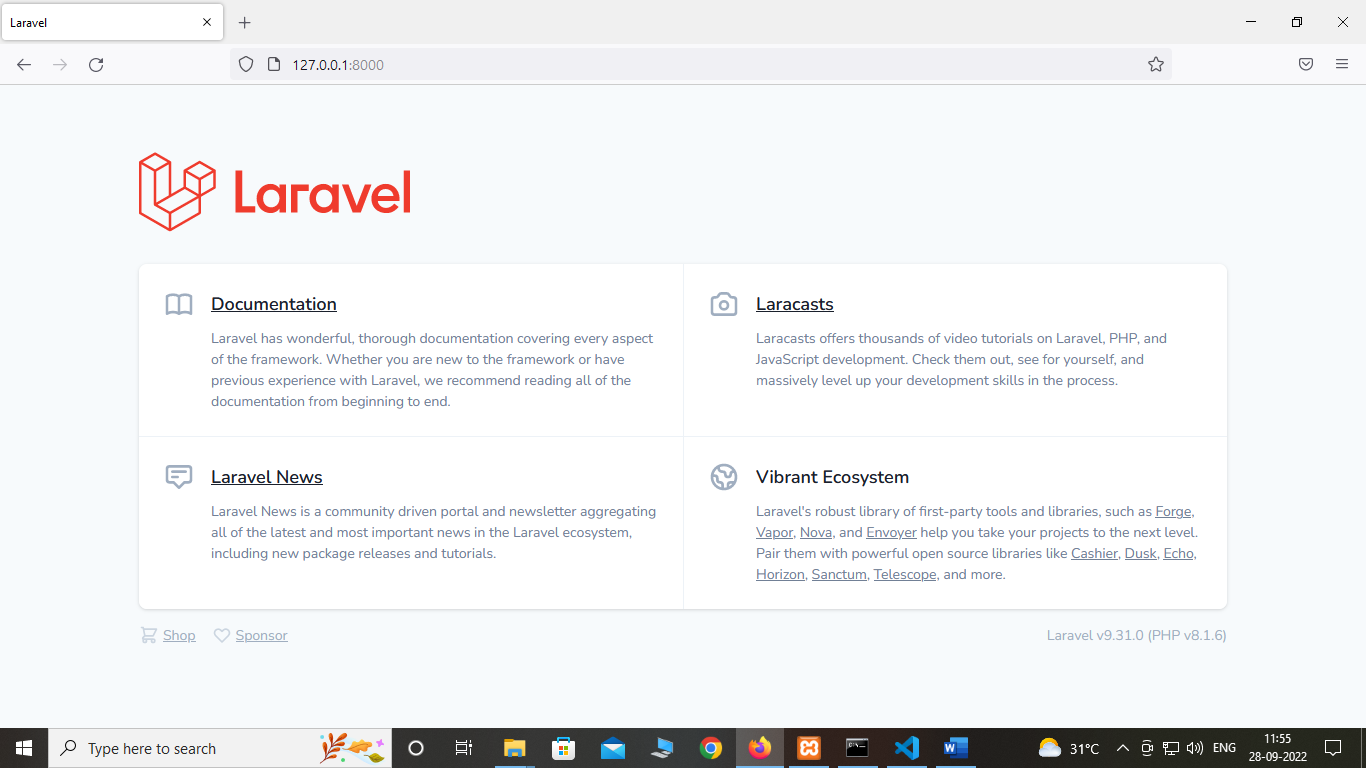
$response = $kernel->handle(

    $request = Request::capture()

)->send();

$kernel->terminate($request, $response);

Localhost/my-laravel-app



How to run Laravel on its server development url

Rename file

Index.php =>server.php

Run command : php artisan serve

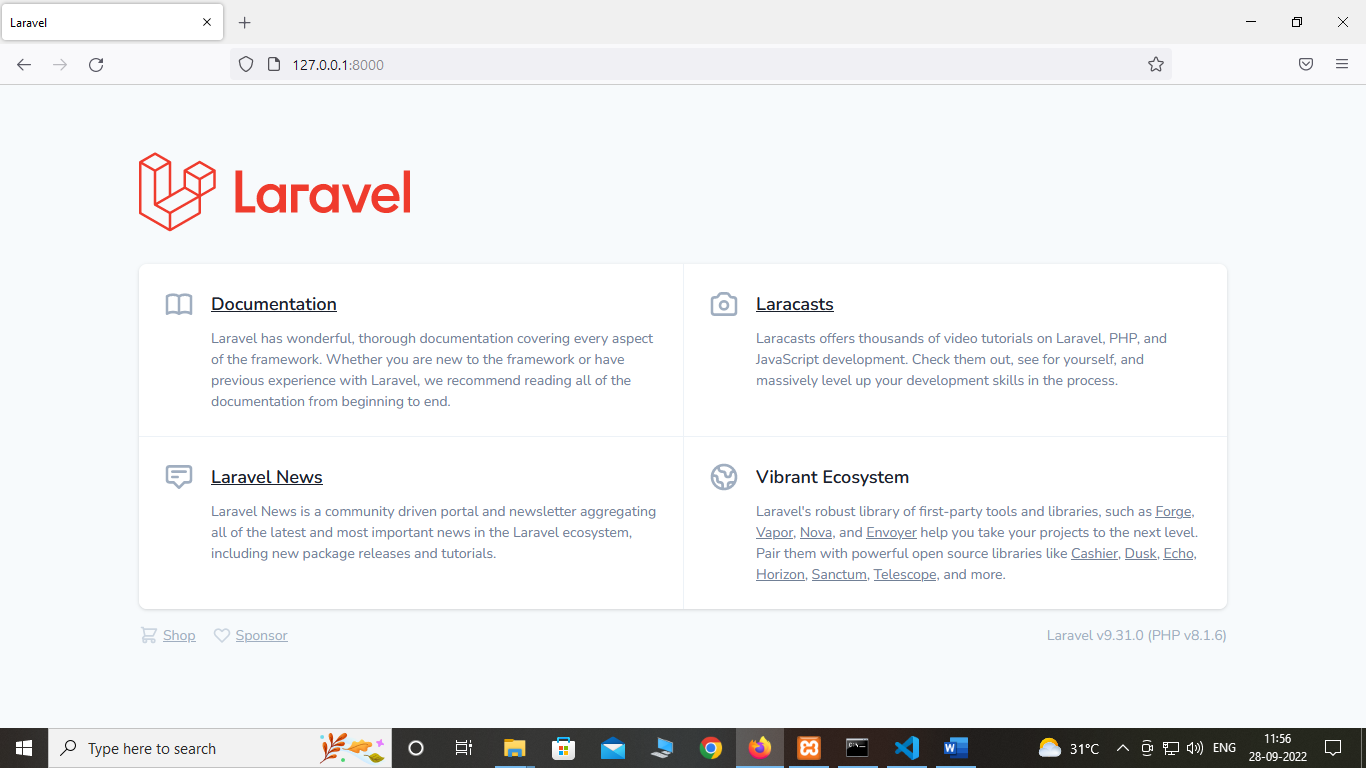
E:\xampp\htdocs\my-laravel-app>php artisan serve

INFO Server running on [http://127.0.0.1:8000].

Press Ctrl+C to stop the server

Run server development url in any browser

<http://127.0.0.1:8000>



How to done Laravel on maintenance mode :

E:\xampp\htdocs\my-laravel-app>php artisan down

INFO Application is now in maintenance mode.

503

Service Unavailable

E:\xampp\htdocs\my-laravel-app>php artisan up

E:\xampp\htdocs\my-laravel-app>php artisan up

INFO Application is now live.

How to connect Laravel app with database

Step-1 first open localhost/phpMyAdmin

Step-2 create database on mysql

my-laravel-app

step-3 open .env file inside of Laravel app

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=my-laravel-app

DB\_USERNAME=root

DB\_PASSWORD=

Step-4 run command

php artisan migrate

step -5 database database\migrations

migrations

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

return new class extends Migration

{

    /\*\*

     \* Run the migrations.

     \*

     \* @return void

     \*/

    public function up()

    {

        Schema::create('users', function (Blueprint $table) {

            $table->id();

            $table->string('name');

            $table->string('email')->unique();

            $table->timestamp('email\_verified\_at')->nullable();

            $table->string('password');

            $table->rememberToken();

            $table->timestamps();

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*

     \* @return void

     \*/

    public function down()

    {

        Schema::dropIfExists('users');

    }

};

How to create a table inside of Laravel

Step 1: create a command to create a table structure or table migration schema.

php artisan make:migration create\_customers\_table

step 2: write a command to migrate table in database

all command of migration

php artisan list

migrate

migrate:fresh Drop all tables and re-run all migrations

migrate:install Create the migration repository

migrate:refresh Reset and re-run all migrations

migrate:reset Rollback all database migrations

migrate:rollback Rollback the last database migration

migrate:status Show the status of each migration

how to create a foreign key in Laravel using database migration

step-1 php artisan make:migration create\_customers\_table

step-2 php artisan make:migration create\_countries\_table

step-3 php artisan make:migration create\_states\_table

step-4 php artisan make:migration create\_cities\_table

Note : we create a id as increments(“id”)

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

return new class extends Migration

{

    /\*\*

     \* Run the migrations.

     \*

     \* @return void

     \*/

    public function up()

    {

        Schema::create('countries', function (Blueprint $table) {

            $table->increments("id");

            $table->string("cname");

            $table->timestamps();

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*

     \* @return void

     \*/

    public function down()

    {

        Schema::dropIfExists('countries');

    }

};

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

return new class extends Migration

{

    /\*\*

     \* Run the migrations.

     \*

     \* @return void

     \*/

    public function up()

    {

        Schema::create('states', function (Blueprint $table) {

            $table->increments("id");

            $table->integer("cid")->unsigned();

            $table->foreign("cid")->references("id")->on("countries");

            $table->string("sname");

            $table->timestamps();

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*

     \* @return void

     \*/

    public function down()

    {

        Schema::dropIfExists('states');

    }

};

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

return new class extends Migration

{

    /\*\*

     \* Run the migrations.

     \*

     \* @return void

     \*/

    public function up()

    {

        Schema::create('cities', function (Blueprint $table) {

            $table->increments("id");

            $table->integer("sid")->unsigned();

            $table->foreign("sid")->references("id")->on("states");

            $table->string("ctname");

            $table->timestamps();

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*

     \* @return void

     \*/

    public function down()

    {

        Schema::dropIfExists('cities');

    }

};

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

return new class extends Migration

{

    /\*\*

     \* Run the migrations.

     \*

     \* @return void

     \*/

    public function up()

    {

        Schema::create('customers', function (Blueprint $table) {

            $table->increments("id");

            $table->string("name");

            $table->string("firstname");

            $table->string("lastname");

            $table->string("email");

            $table->string("password");

            $table->bigInteger("mobile");

            $table->string("gender");

            $table->string("hobby");

            $table->text("address");

            $table->string("upload\_photo");

            $table->integer("cid")->unsigned();

            $table->foreign("cid")->references("id")->on("countries");

            $table->integer("sid")->unsigned();

            $table->foreign("sid")->references("id")->on("states");

            $table->integer("ctid")->unsigned();

            $table->foreign("ctid")->references("id")->on("cities");

            $table->timestamps();

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*

     \* @return void

     \*/

    public function down()

    {

        Schema::dropIfExists('customers');

    }

};