Laravel

1. download and install XAMPP, Composer for PHP.
2. If error occurs when installing Composer packages: run following command in powershell

🡪 Run as administrator...

Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://php.new/install/windows/8.4'))

1. To create a project using any of this two
   1. Using Laravel Installer ( we can choose DB as any )

composer global require laravel/installer

laravel new example-app

cd example-app

npm install && npm run build

composer run dev

* 1. Using Composer Directly ( Default DB is sqlite )

choose location 🡪 cd:\xampp\hdocs OR any other location

composer create-project laravel/laravel project-name

1. start project
   1. php artisan serve 🡪 Your project will run at: <http://127.0.0.1:8000>
2. DB configuration

In your .env file, make sure the database settings are like this:

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=myapptest\_db

DB\_USERNAME=root

DB\_PASSWORD=

1. Optional commands
   1. php artisan list
   2. php artisan make
   3. php artisan make:controller –help
   4. php artisan config:cache
2. After DB sets in .env
   1. php artisan migrate

It will create default Laravel tables (like users, password\_resets, etc).

1. will see an error in loaded website (<http://127.0.0.1:8000/>)
   1. Fix ( got to app/ AppServiceProvider.php) :
      1. Add second line like this,

use Illuminate\Support\ServiceProvider;

use Illuminate\Support\Facades\Schema;

* + 1. In boot ()

Schema::defaultStringlength(191);

public function boot(): void

{

Schema::defaultStringlength(191);

}

* 1. Run again
     1. php artisan migrate
        + - will create three tables as user, passwords reset migration
     2. If you want to undo the last migration

php artisan migrate:rollback

Create tables in project

Go to terminal and run :

php artisan make:migration table\_name

(This will create a file in database/migrations/ folder.). Actually this is only for support no need manual process to create dbs with columns, just edit this file created in migration

Open that file that created in database\migration\ and there is a function called up() : void edit it like this

✅ (Note: id and timestamps() are usually already included.)

public function up(): void

    {

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

            $table->increments('id');

            $table->string('name');

            $table->string('color');

            $table->string('company');

            $table->timestamps();

        });

    }

Run 🡪

php artisan migrate

this will create table named 'cars' in actual mysql db & will add those attributes as columns into that table

Create Controllers & Models

php artisan make:controller CarController -r -m Car

* Check in app/Models/ there may be file name called Car,
* Chek in app/Http/ there may be file named CarController

Let creates routes

Got to routes open web.php initialize a route there are feew ways  
  
 Route::get('/cars', 'CarController@index'); // called Carcvontrolleed index method

Since we call index method go to CarController , then do something

Her we call blade.php file

Now crate it as index.blade.php at resources/view/cars/ folder

When we call [127.0.0.1:8000/cars](http://127.0.0.1:8000/cars) it call as follow files(methods)

Web.php(Route named ‘/cars’) ---> CarController (index method) ---> index.blade.php(in view/cars/)

Route::get('/cars', [CarController::class, 'index']);

public function index()

    {

        $name = 'Lmaborghini';

        return view('cars.index', );

    }

<h1>Hello world</h1>

This is the structure you can refer the code files

Data passing to blade.php

public function index()

    {

        $name = 'Lmaborghini';

        return view('cars.index', ['name'=> $name]);

    }

Values passing to url

Route::get('/cars/{carid}', [CarController::class, 'show']);

To create model controller migration in one command

php artisan make:model CustomUser -cm