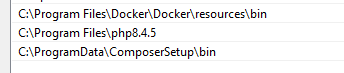
**Laravel workshop for SMCians day1 by 2019 batch:**

**Composer php and Laravel setup:**

If you have used nodejs we see package manager as npm for java maven for android java gradle and python has pip likewise we have composer in laravel working as a package manager

install php latest version from php official site set the system env variable as path of php and then install composer from getcomposer.org then put the environmental of the path your php and composer being installe my sample below☹

**(caution make a folder to do following like I made all the work in directory named \Web\laravelprojects)**



**Command prompt code**

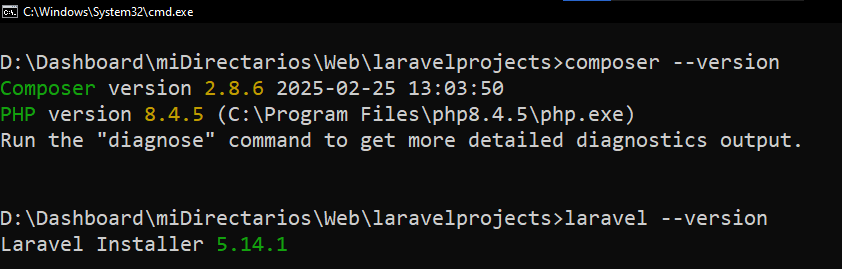
*>>composer global require laravel/installer*

After this check if Laravel and composer and php is installed or not which version installed will be shown if we type following command

*>>composer –version*

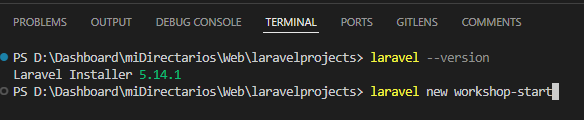
*>>Laravel –version*

*>>php –version*

If any of them shows not a internal or external command that means they are not installed or you didn’t set environment well or you messed up badly….

Something like this should show up

Then type following command



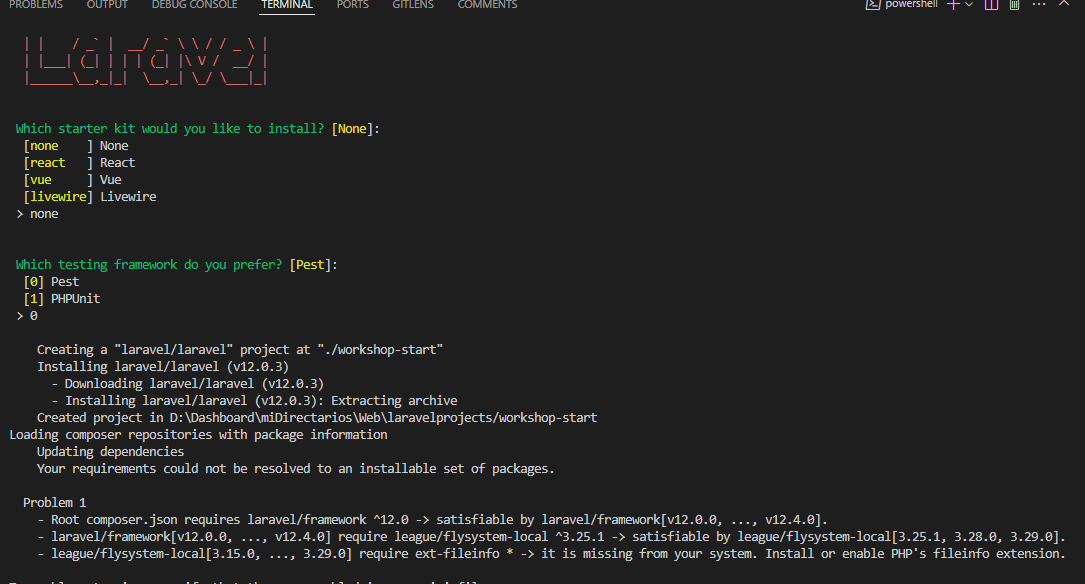
Php Laravel does not support the zipping directly so we need to paste the following code on the path **C:\Program Files\php8.4.5\php.ini** your path might be different in your system the path is where you have installed php copy paste the following code to your

Uncomment or remove ; from the the zip and filename and do such for database which one of them you want to use

;extension=pdo\_sqlite

;extension=sqlite3

Then do just like I did for in command prompt as below



Then it will give all the files required for your project setup in MVC pattern model being database file view being the html or React depends on you then controller being whatever backend code you write and route is provided for routing purpose

Then locate to the file and run

***Cd <project name>***

***Php artisan serve***

# Database connection table and attribute creation using laravel only no phpmyadmin

Go to .env file and find this give the databae name that has been created and give right credentials in following code

***DB\_CONNECTION=mysql***

***DB\_HOST=127.0.0.1***

***DB\_PORT=3306***

***DB\_DATABASE=laravel-crud***

***DB\_USERNAME=root***

***DB\_PASSWORD=***

To check if database is connected or not (actually this code executes all sql ddl queries that we made already by getting connection no connection means error)

***php artisan migrate***

this following will create table called products table and also a file in database>migrations

***php artisan make:migration create\_products\_table***

**locate the file named create\_products\_table.php that will have all the template already setup all we need to do will be putting attribute name and type that’s it**

following is the way we can create our schema table not directly putting code in sql running migration code after this will execute it and create table automatically

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

            $table->id();

            $table->string("pname");

            $table->integer("qty");

            $table->decimal("price");

            $table->text("description");

            $table->timestamps();

        });

**//then run following command in terminal new table product with our custom given attribute will be available with timestamps will be shown as default feature provided my laravel**

>>php artisan migrate

# Making model in Laravel

**In terminal type following note that name must be singular (like mine product)**

>>php artisan make:model Product

This will create model name product its template in model folder locate it and write following

class Product extends Model

{

    use HasFactory;

    // fillable represents the model blueprint/schema/template

    // that will be used to create the database table

    // and the data that will be stored in the database just kind of like placeholder that will get

    //indulged with databae to fill the data inserted in view

    protected $fillable = [

        'pname',

        'description',

        'price',

        'qty',

    ];

}

# Creating controller in Laravel using terminal

php artisan make:controller ProductController

inside ProductController.php

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class ProductController extends Controller

{

    //

    public function index(){

        return view('products.index');

    }

}

# Creating view in Laravel

Find views in resources/views file and create folder to place your view and make new file like index.blade.php and write following code in that file

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <h1>Product</h1>

    <div>index</div>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <h1>Product</h1>

    <div>index</div>

</body>

</html>

# Creating routes in Laravel

Routes/web.php

<?php

use Illuminate\Support\Facades\Route;

App\Http\Controllers\ProductController;

Route::get('/', function () {

    return view('welcome');

});

//the get request on /product endpoint

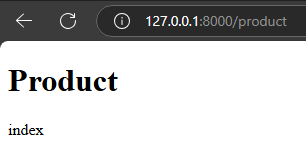
//parameter 2: loscation of controller

//function name of the controller

// with name of the route we specify only name of that route

Route::get('/product', [ProductController::class, 'index'])->name('products.index');

**After that you may get basic page as follows**

****

**If we are downloading the project from github we need to create new .env file taking reference to .env.example file already given and run following command in main folder**

*PS D:\Dashboard\miDirectarios\Web\laravelWorkshops\crud-app> php artisan key:generate*

*php artisan config:clear*

*php artisan serve*

# MVCR once again

**Step1: create file in /views/products/create.blade.php**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=`device-width`, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <h1>create a product</h1>

</body>

</html>

**Step 2; controller to pass control to view**

 public function create(){

        return view('products.create');

    }

**Step 3: route to hit to invoke that controller**

Route::get('/product/create', [ProductController::class, 'create'])->name('products.create');

# Finalized code to create new entry in the database using laravel

**Web.php(route)**

<?php

use Illuminate\Support\Facades\Route;

use App\Http\Controllers\ProductController;

Route::get('/', function () {

    return view('welcome');

});

//the get request on /product endpoint

//parameter 2: loscation of controller

//function name of the controller

// with name of the route we specify only name of that route

Route::get('/product', [ProductController::class, 'index'])->name('products.index');

Route::get('/product/create', [ProductController::class, 'create'])->name('products.create');

Route::post('/product', [ProductController::class, 'store'])->name('products.store');

**Product.php(model)**

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Product extends Model

{

    // fillable represents the model blueprint/schema/template

    // that will be used to create the database table

    // and the data that will be stored in the database just kind of like placeholder that will get

    //indulged with databae to fill the data inserted in view

    protected $fillable = [

        'pname',

        'description',

        'price',

        'qty',

    ];

}

**ProductController.php(controller)**

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

//importing the model  b

use App\Models\Product;

class ProductController extends Controller

{

    //

    public function index(){

        return view('products.index');

    }

    public function create(){

        return view('products.create');

    }

    public function store(Request $request){

        //dump data for debug puropose to see if data is coming or not coming it shows the gathered data

        // dd($request);

        $data = $request->validate([

            'pname' => 'required',

            'qty' => 'required|numeric',

            'price' => 'required|numeric|min:0',

            'description' => 'required'

        ]);

        $newProduct = Product::create($data);

        return redirect()->route('products.index');

    }

}

**Create.blade.php(view)**

    <!DOCTYPE html>

    <html lang="en">

    <head>

        <meta charset="UTF-8">

        <meta name="viewport" content="width=`device-width`, initial-scale=1.0">

        <title>Document</title>

    </head>

    <body style="font-family: Arial, sans-serif; background-color: #f4f4f9; padding: 20px;">

        <h2 style="text-align: center; color: #333;">Product Entry Form</h2>

        <!-- handling form in products.store controller writing them in action dows that by giving post request    -->

        <form method="post" action="{{ route('products.store') }}" style="max-width: 400px; margin: auto; padding: 20px; border: 1px solid #ccc; border-radius: 8px; background-color: #fff;">

            @csrf

            @method('POST')

            <label for="pname" style="display: block; margin-bottom: 10px; font-weight: bold; color: #555;">Product Name</label>

            <input type="text" id="pname" name="pname" style="width: 100%; padding: 8px; margin-bottom: 15px; border: 1px solid #ccc; border-radius: 4px;" placeholder="Enter product name" required>

            <label for="qty" style="display: block; margin-bottom: 10px; font-weight: bold; color: #555;">Quantity</label>

            <input type="number" id="qty" name="qty" style="width: 100%; padding: 8px; margin-bottom: 15px; border: 1px solid #ccc; border-radius: 4px;" placeholder="Enter quantity" required>

            <label for="price" style="display: block; margin-bottom: 10px; font-weight: bold; color: #555;">Price</label>

            <input type="number" step="0.01" id="price" name="price" style="width: 100%; padding: 8px; margin-bottom: 15px; border: 1px solid #ccc; border-radius: 4px;" placeholder="Enter price" required>

            <label for="description" style="display: block; margin-bottom: 10px; font-weight: bold; color: #555;">Description</label>

            <textarea id="description" name="description" style="width: 100%; padding: 8px; margin-bottom: 15px; border: 1px solid #ccc; border-radius: 4px;" placeholder="Enter description" required></textarea>

            <button type="submit" style="width: 100%; padding: 10px; background-color: #28a745; color: white; border: none; border-radius: 4px; font-size: 16px; cursor: pointer;">

                Submit

            </button>

        </form>

    </body>

    </html>