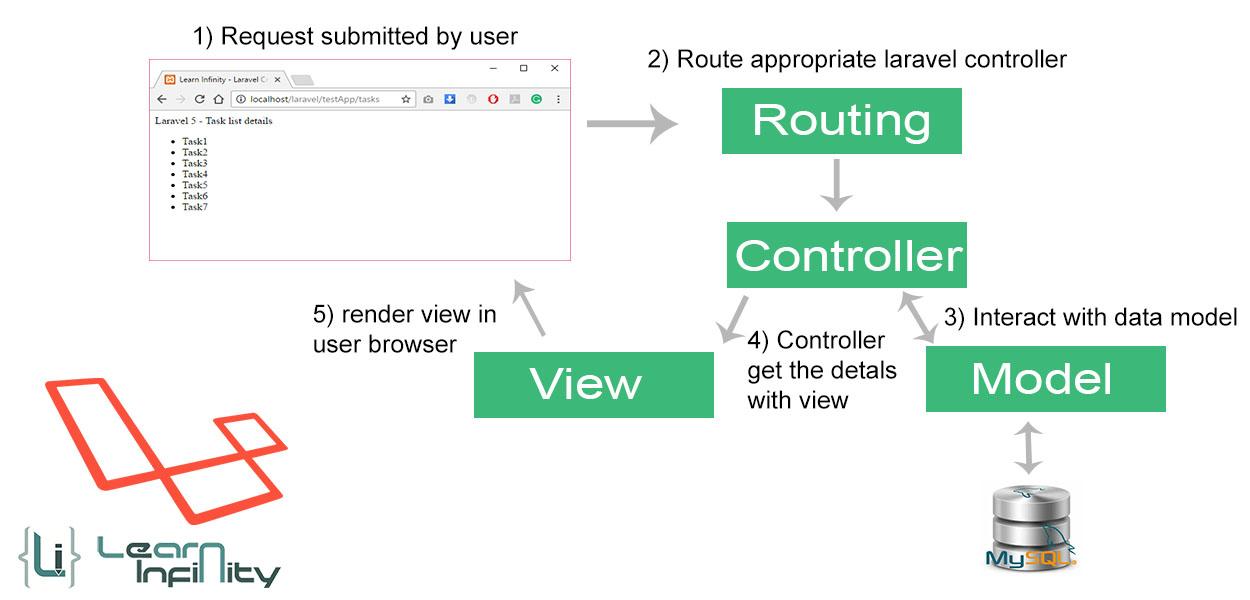
**Bài thực hành 02**

**Giới thiệu MVC trong Laravel**

***Mục tiêu***

* Nắm được các thành phần MVC trong laravel
* Khai báo và sử dụng route
* Tạo và sử dụng master view layout cho dự án
* Sử dụng một số lệnh command line cơ bản
* Tạo và sử dụng route get, post



* 1. ***Bài thực hành Step by Step***

# Bài 1: Tìm hiểu về Route::get & Route::post

Route::get chính là phương thức GET, cho phép truy cập các địa chỉ trang thông qua url trình duyệt

Route::post chính là phương thức POST, chỉ có thể thông qua submit form mới có thể truy cập tới

**Bước 1:** khai báo 2 route get

Mở file web.php trong thư mục routes và khai báo

Route::get('/', function () {

    return view('welcome');

});

Route::get('/demo', function () {

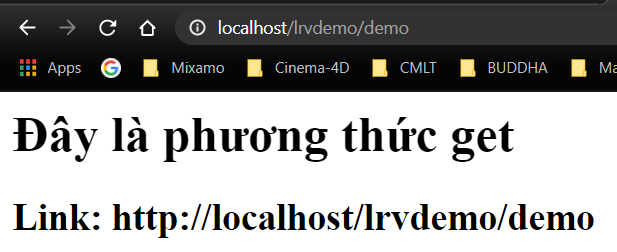
    echo '<h1>Đây là phương thức get</h1>';

    echo '<h2>Link: '.url('demo').'</h2>';

});

**Bước 2:** truy cập router demo theo link <http://localhost/lrvdemo/demo>

Kết quả có dạng sau



**Bước 3:** sửa lại route get thành post như sau

Route::get('/', function () {

    return view('welcome');

});

Route::**post**('/demo', function () {

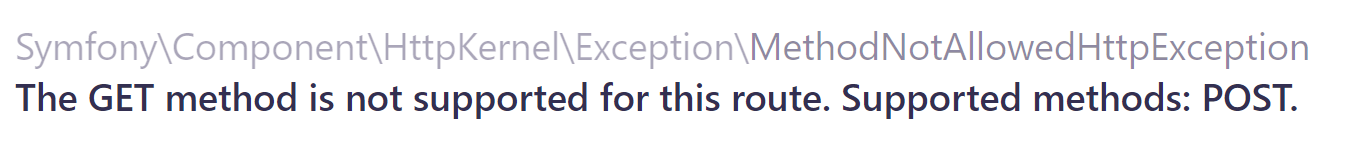
    echo '<h1>Đây là phương thức **post**</h1>';

    echo '<h2>Link: '.url('demo').'</h2>';

});

**Bước 4:** Vẫn tại link <http://localhost/lrvdemo/demo>

Hãy F5 lại sẽ có lỗi như sau



Nghĩa là phương thức GET không hỗ trợ cho route demo này, nó hỗ trợ phương thức POST

Vậy để có thể sử dụng phương thức post, hày tạo form với phương thức POST và có action tới route demo này

**Bước 5: Mở file welcome.blade.php** trong thư mục resources/views, xóa toàn bộ code cũ của nó và soạn lại code html mới với cấu trúc form như sau

<!DOCTYPE html>

<html>

<head>

    <meta charset="UTF-8">

    <title>Document</title>

</head>

<body>

    <form action="demo" method="post">

<!— Cần phải có @csrf tạo ra thẻ input ẩn chứa mã token bảo mật form -->

        @csrf

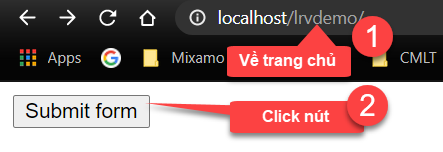
        <input type="submit" value="Submit form">

    </form>

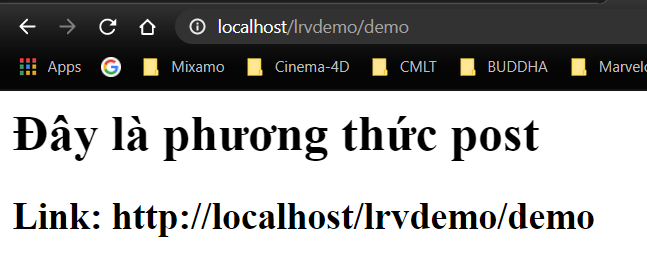
</body>

</html>

**Bước 6:** Hãy truy cập lại trang chủ của dự án và click vào nút bấm trên fỏm vừa thiết kế



Kết quả có dạng



**Chú ý:** Nếu trên form không có cú pháp **@csrf**, khi submit form sẽ có lỗi 419, lỗi này thông báo hết hạn vì mã token không phù hợp hoặc không có, mã token có tác dụng tránh giả mạo form khi gửi dữ liệu lên máy chủ

Ví dụ form không có **@csrf**

<!DOCTYPE html>

<html>

<body>

    <form action="demo" method="post">

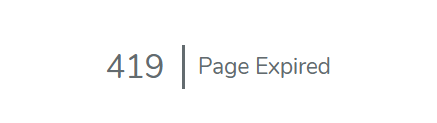
        <input type="submit" value="Submit form">

    </form>

</body>

</html>

Lỗi có dạng sau



# Bài 2: Route & controller

Controller là một class (lớp) gồm nhiều phương thức (action), và mỗi route có thể gọi đến một action này

**Bươc 1:** tạo một controller là HomeController bằng lệnh

**php artisan make:controller HomeController**

**Bước 2:** Sau đó mở HomeController lên và khai báo phương thức index có nội dung sau

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class HomeController extends Controller

{

    public function index()

    {

        return view('index');

    }

}

Như vậy trong code trên ta thấy, phương thức index này trả về một view tên là **index** (chính là file view nằm trong thư mục **resources/views**)

Bước 3: Tạo file **index.blade.php** trong thư mục resources/view với nội dung sau

<!DOCTYPE html>

<html lang="">

    <head>

        <meta charset="utf-8">

        <meta http-equiv="X-UA-Compatible" content="IE=edge">

        <meta name="viewport" content="width=device-width, initial-scale=1">

        <title>Title Page</title>

        <!-- Bootstrap CSS -->

        <link href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" rel="stylesheet">

    </head>

    <body>

        <div class="jumbotron">

            <div class="container">

                <h1>Hello, world!</h1>

                <p>Contents ...</p>

                <p>

                    <a class="btn btn-primary btn-lg">Learn more</a>

                </p>

            </div>

        </div>

        <!-- jQuery -->

        <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>

        <!-- Bootstrap JavaScript -->

        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>

    </body>

</html>

**Bước 3:** Mở lại **routes/web.php** và khai báo lại route như sau

<?php

use Illuminate\Support\Facades\Route;

use App\Http\Controllers\HomeController;

/\*

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

| Web Routes

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

|

| Here is where you can register web routes for your application. These

| routes are loaded by the RouteServiceProvider within a group which

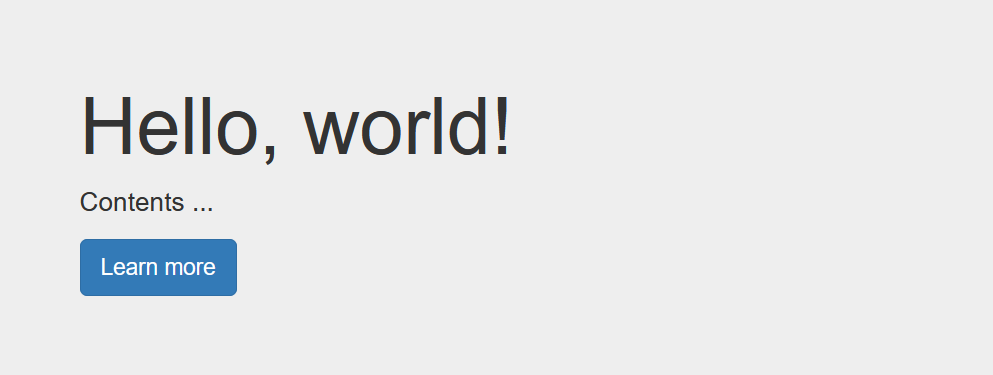
| contains the "web" middleware group. Now create something great!

|

\*/

Route::get('', [HomeController::class, 'index'])->name('home.index');

**Bước 4:** Chạy lai link: <http://localhost/lrvdemo>



# Bài 3: Hiển thị dữ liệu từ model lên view

**Bước 1:** Tạo model Category bằng lệnh cmd sau

**php artisan make:model Category**

Mở model Category lê và khai báo như sau

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class Category extends Model

{

    use HasFactory;

    /\*\*Phương thức trả về mảng danh mục gồm: id, name, status \*/

    public static function getAll()

    {

        return [

            ['id' =>  1, 'name' => 'Áo nam', 'status' => 1],

            ['id' =>  2, 'name' => 'Áo nữ', 'status' => 0],

            ['id' =>  3, 'name' => 'Quần bò', 'status' => 0],

            ['id' =>  4, 'name' => 'Túi xách', 'status' => 1],

            ['id' =>  5, 'name' => 'Ví da','status' => 1],

            ['id' =>  6, 'name' => 'Giày dép', 'status' => 1],

            ['id' =>  7, 'name' => 'Dây lưng','status' => 1],

            ['id' =>  8, 'name' => 'Đồng hồ','status' => 1]

        ];

    }

}

**Bước 2: Mở lại HmeController tại phương thức index code như sau**

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Models\Category;

class HomeController extends Controller

{

    public function index()

    {

        $categories =  Category::getAll();

        return view('index', compact('categories'));

    }

}

**Bước 3: Mở lại view index.blade.php và duyệt dữ liệu trên abngr html như sau**

<div class="jumbotron">

    <div class="container">

        <h1>Danh mục</h1>

        <table class="table table-hover">

            <thead>

                <tr>

                    <th>ID</th>

                    <th>Name</th>

                    <th>Status</th>

                </tr>

            </thead>

            <tbody>

            @foreach($categories as $cat)

                <tr>

                    <td>{{ $cat['id'] }}</td>

                    <td>{{ $cat['name'] }}</td>

                    <td>{{ $cat['status'] == 0 ? 'Tạm ẩn' : 'Hiển thị'  }}</td>

                </tr>

                @endforeach

            </tbody>

        </table>

        <hr>

        <p>

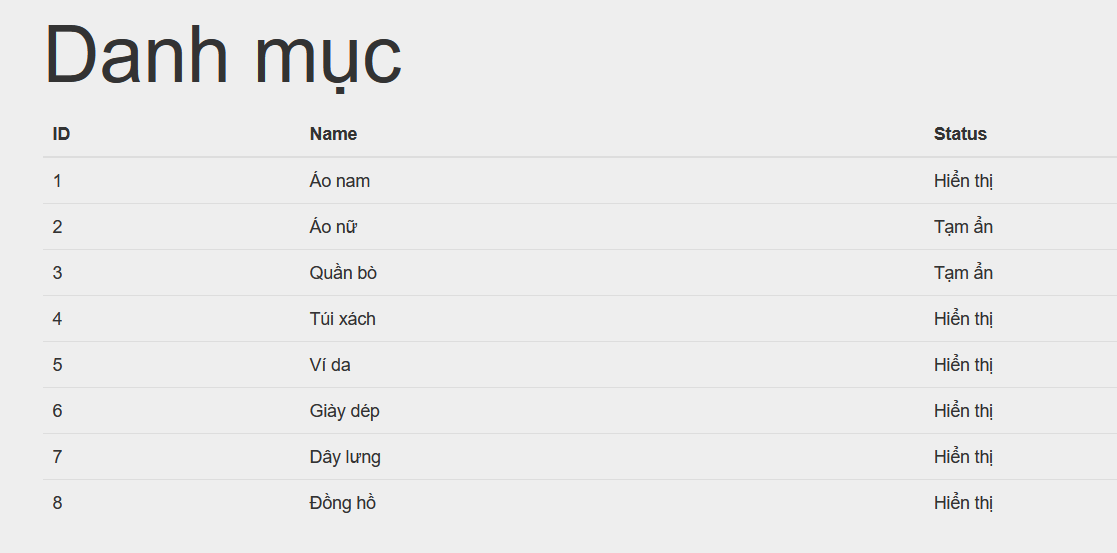
            <a class="btn btn-primary btn-lg">Learn more</a>

        </p>

    </div>

</div>

**Kết quả tại trang chủ hiển thị danh sách có dạng**



# Bài 4: Chuyển đổi mảng dữ liệu thông thường sang Collection đối tượng model hiện tại

**Bước 1:** Mở lại model Category và khai báo nhu sau

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class Category extends Model

{

    use HasFactory;

    // khai báo các trường dữ liệu

    protected $fillable = ['id','name','status'];

    /\*\*Phương thức trả về mảng danh mục gồm: id, name, status \*/

    Public static function getAll()

    {

        // mảng dữ liệu danh mục

        $data = [

            ['id' =>  1, 'name' => 'Áo nam', 'status' => 1],

            ['id' =>  2, 'name' => 'Áo nữ', 'status' => 0],

            ['id' =>  3, 'name' => 'Quần bò', 'status' => 0],

            ['id' =>  4, 'name' => 'Túi xách', 'status' => 1],

            ['id' =>  5, 'name' => 'Ví da','status' => 1],

            ['id' =>  6, 'name' => 'Giày dép', 'status' => 1],

            ['id' =>  7, 'name' => 'Dây lưng','status' => 1],

            ['id' =>  8, 'name' => 'Đồng hồ','status' => 1]

        ];

        $resault = [];

        foreach($data as $item) {

            $resault[] = (new static)->fill($item);

        }

        // trave một collection gồm các đối tượng

        return collect($resault);

    }

}

**Bước 2: Mở lại file resources/view/index.blade.php và sửa lại code như sau**

**Tìm tơi đoạn duyệt dữ liệu**

@foreach($categories as $cat)

    <tr>

        <td>{{ $cat->id }}</td>

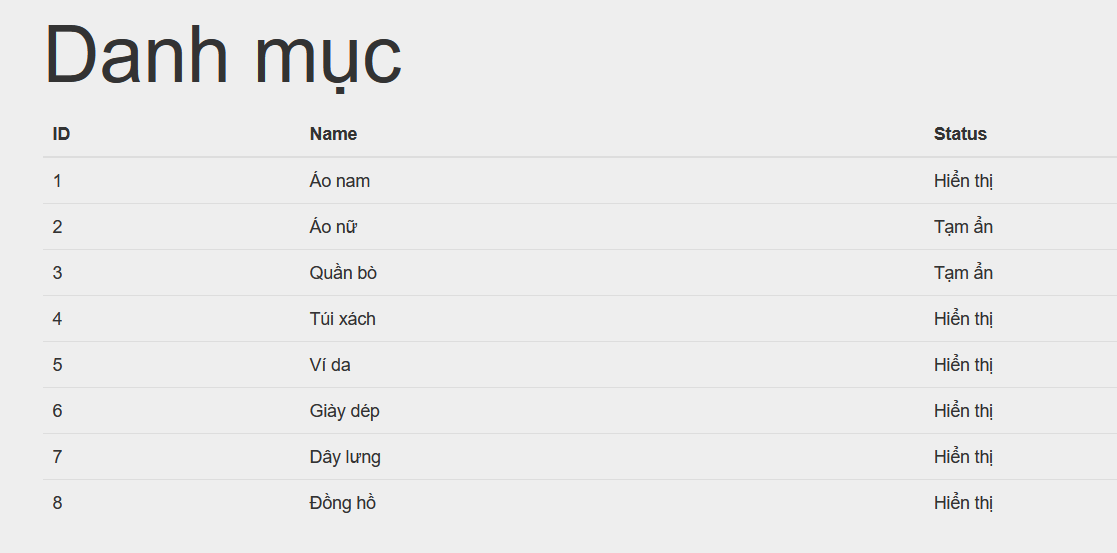
        <td>{{ $cat->name }}</td>

        <td>{{ $cat->status == 0 ? 'Tạm ẩn' : 'Hiển thị'  }}</td>

    </tr>

@endforeach

**Kết quả danh sách vẫn không thay đổi**



# Bài 5: Cấu hình phân trang từ mảng dữ liệu thông thường

**Bước 1: Mở lại model Category và thêm các phương thức sau**

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

// Phân trang

use Illuminate\Pagination\Paginator;

use Illuminate\Support\Collection;

use Illuminate\Pagination\LengthAwarePaginator;

class Category extends Model

{

    use HasFactory;

    // khai báo các trường dữ liệu

    protected $fillable = ['id','name','status'];

    /\*\*Phương thức trả về mảng danh mục gồm: id, name, status \*/

    public static function getAll()

    {

        // mảng dữ liệu danh mục

        $data = [

            ['id' =>  1, 'name' => 'Áo nam', 'status' => 1],

            ['id' =>  2, 'name' => 'Áo nữ', 'status' => 0],

            ['id' =>  3, 'name' => 'Quần bò', 'status' => 0],

            ['id' =>  4, 'name' => 'Túi xách', 'status' => 1],

            ['id' =>  5, 'name' => 'Ví da','status' => 1],

            ['id' =>  6, 'name' => 'Giày dép', 'status' => 1],

            ['id' =>  7, 'name' => 'Dây lưng','status' => 1],

            ['id' =>  8, 'name' => 'Đồng hồ','status' => 1]

        ];

        $resault = [];

        foreach($data as $item) {

            $resault[] = (new static)->fill($item);

        }

        // trave một collection gồm các đối tượng

        return collect($resault);

    }

    // Phân trang

    public static function paginate($perPage = 5, $page = null, $options = [])

    {

        $items = (new static)->getAll(); // láy dữ liệu từ phương thức getAll

        $page = $page ?: (Paginator::resolveCurrentPage() ?: 1);

        $items = $items instanceof Collection ? $items : Collection::make($items);

        return new LengthAwarePaginator($items->forPage($page, $perPage), $items->count(), $perPage, $page, $options);

    }

}

**Bước 2: Tại HomeController sử dụng phương thức pagination thay cho getAll() để lấy dữ liệu đã có phân trang**

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Models\Category;

class HomeController extends Controller

{

    public function index()

    {

        $categories = Category::paginate();

        return view('index', compact('categories'));

    }

}

**Bước 3: Tại view resources/views/index.blade.php**

<div class="container">

    <h1>Danh mục</h1>

    <table class="table table-hover">

        <thead>

            <tr>

                <th>ID</th>

                <th>Name</th>

                <th>Status</th>

            </tr>

        </thead>

        <tbody>

            @foreach($categories as $cat)

                <tr>

                    <td>{{ $cat->id }}</td>

                    <td>{{ $cat->name }}</td>

                    <td>{{ $cat->status == 0 ? 'Tạm ẩn' : 'Hiển thị'  }}</td>

                </tr>

            @endforeach

        </tbody>

    </table>

    <hr>

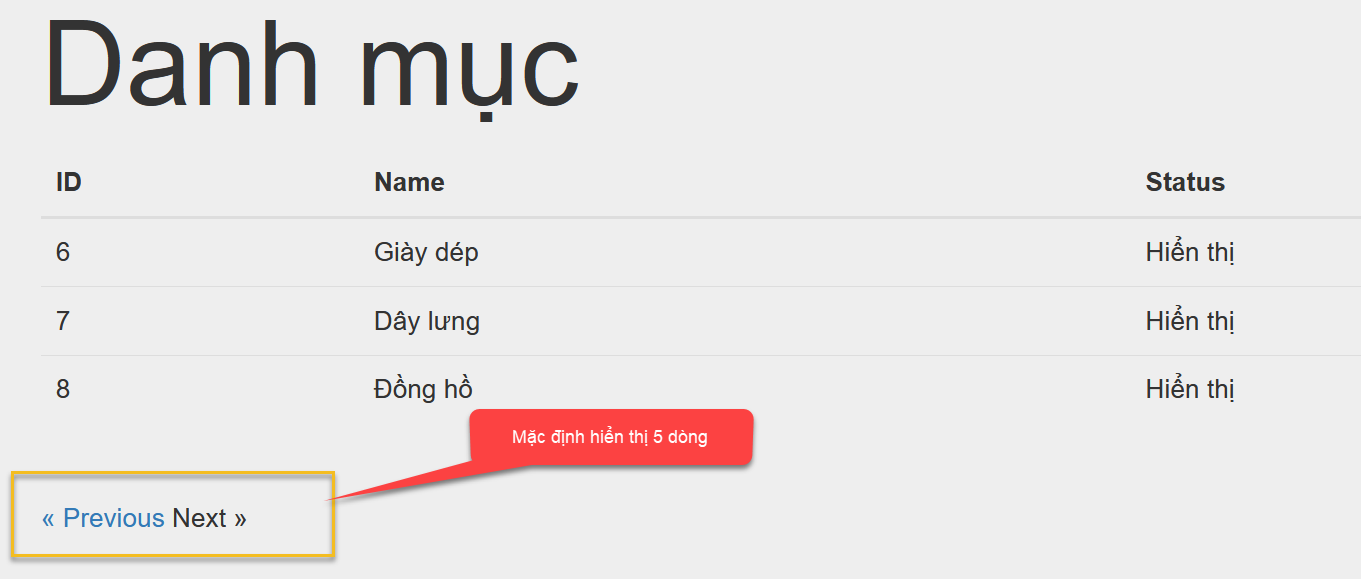
    <div>

        {{ $categories->links() }}

    </div>

</div>

**Kết quả ta có danh sách phân trang**



# Bài 5: Tạo và sử dụng master view cho dự án

**Bước 1:** Tạo master view

Tạo thêm thư mục **layouts** trong thư mục **resources/views**, thư mục này để chứa các file master view

Trong thư mục **layouts** vừa tạo, tạo thêm file master.blade.php

**master.blade.php** là file chứa cấu trung chung cho toàn bộ dự án, chứa các thành phần có thể tái sử dụng tại nhiều trang khác nhau như: header, menu, footer, menu trái, menu phải…

Ngoài ra nó còn link các thư viện css, js, …để cùng tái sử dụng chung với các thành phần trên khi các layout con kế thừa tới nó

Tùy vào cấu trúc giao điện đơn giản hay phức tạp để tạo file master.blade.php này

Các thành phần riêng của từng layout con sẽ được đặt vào một cú pháp **@yield(‘sectin\_name’)** trong đó section\_name là tên tùy chọn

Trong demo này **master.blade.php** có code sau

<!DOCTYPE html>

<html>

    <head>

        <meta charset="utf-8">

        <meta http-equiv="X-UA-Compatible" content="IE=edge">

        <meta name="viewport" content="width=device-width, initial-scale=1">

        <title>Title Page</title>

        <!-- Bootstrap CSS -->

        <link href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" rel="stylesheet">

    </head>

    <body>

        <nav class="navbar navbar-inverse">

            <div class="container">

                <ul class="nav navbar-nav">

                    <li class="active"><a href="#">Home</a></li>

                    <li><a href="#">Link</a></li>

                </ul>

            </div>

        </nav>

        <div class="container">

**@yield('main')**

        </div>

        <!-- jQuery -->

        <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>

        <!-- Bootstrap JavaScript -->

        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>

    </body>

</html>

**Bước 2:** Sử dụng **master.blade.php**

Mở file **resources/view/index.blade.php** trong thư mục resources/views, file welcome.blade.php kế thừa các thành phần của master.blade.php, code như sau

Các layout con khác muốn kế thừa master.blade.php thì cú pháp tương tự

@extends('layouts.master') <!-- Kế thừa master view -->

@section('main') <!-- main là giá trị tại @yield() trong master.blade.php -->

<div class="jumbotron">

    <div class="container">

        <h1>Hello, home page!</h1>

        <p>Contents ...</p>

        <p>

            <a class="btn btn-primar y btn-lg">Learn more</a>

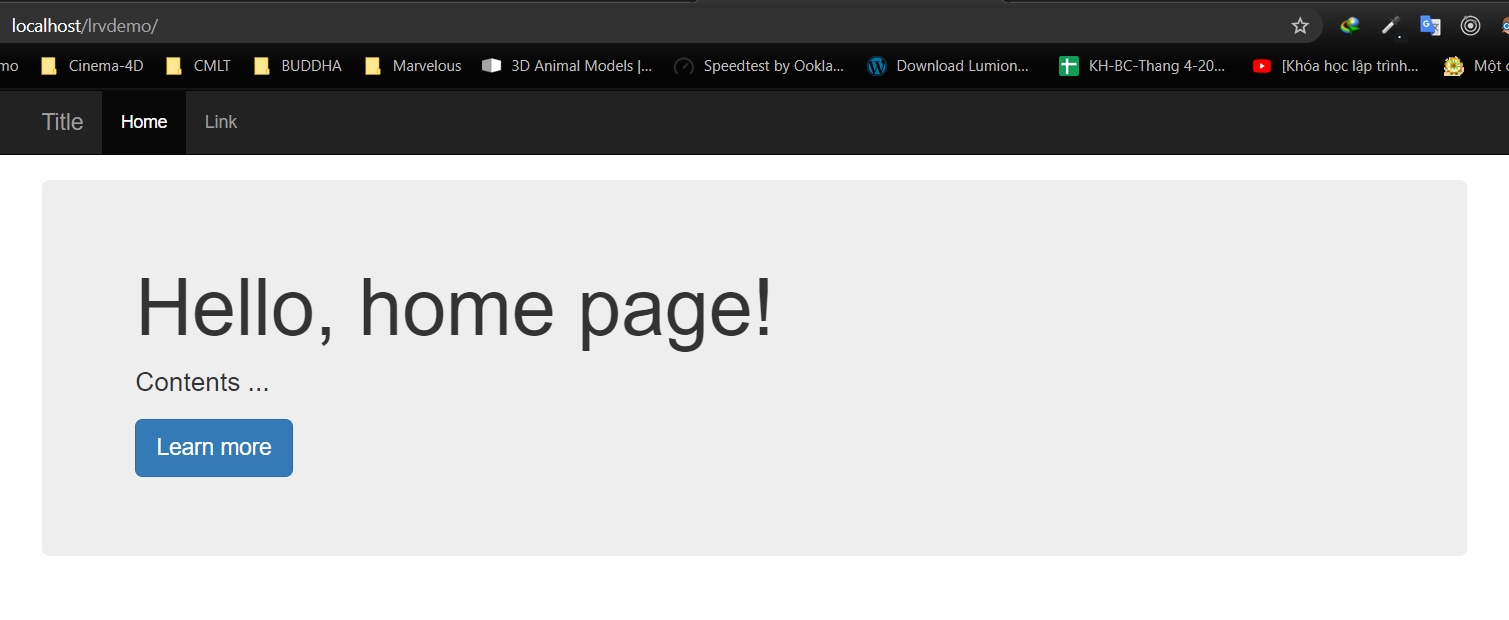
        </p>

    </div>

</div>

@stop() <!-- kết thúc một section -->

**Bước 3:** Truy cập trang chủ của dự án và xem lại kết quả

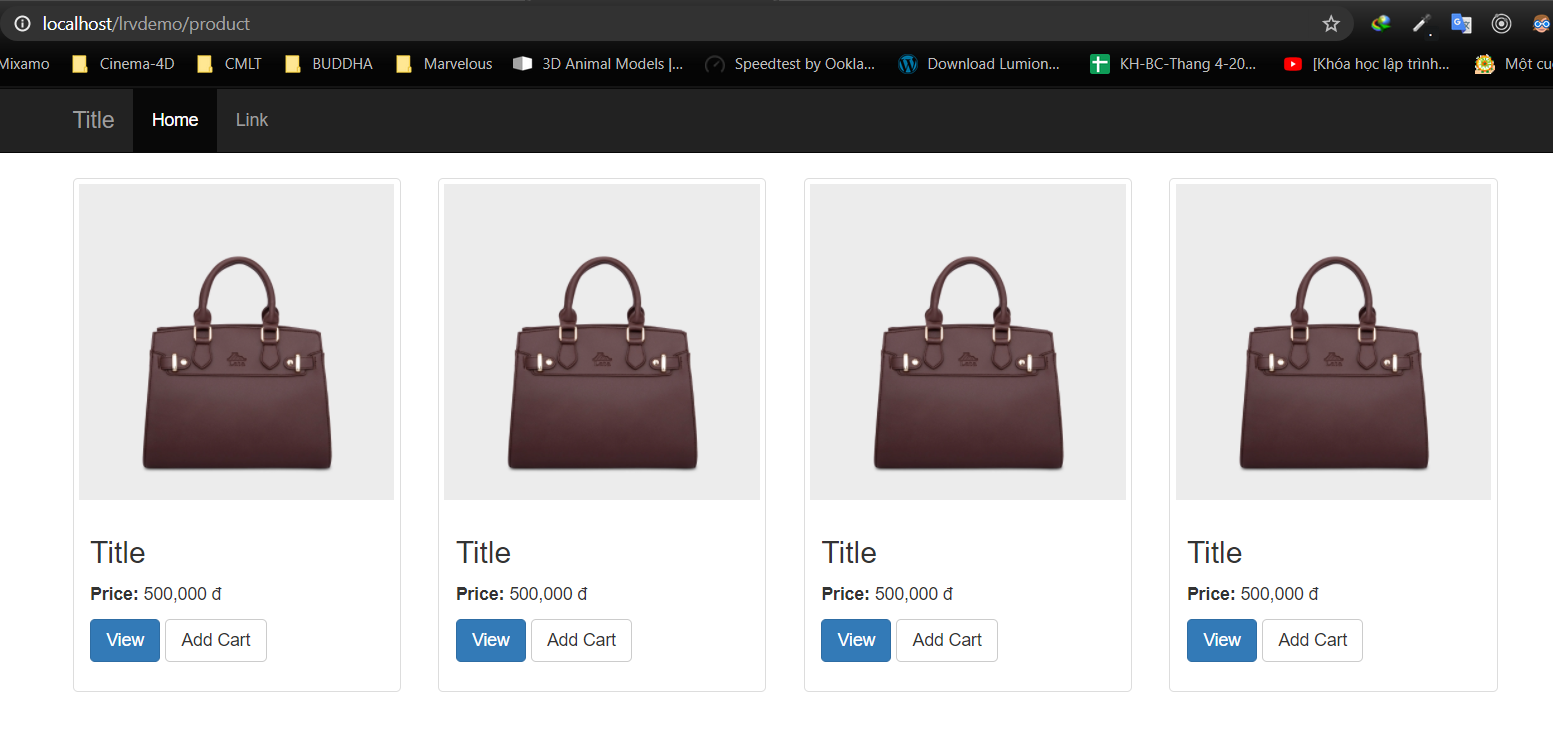


# Bài tập làm thêm

**Bài 1:** Tạo model Product trả về danh sách các sản phẩm gồm: name, image, price

Link: <http://localhost/lrvdemo/product>

Mà hình kết quả sau khi duyệt dữ liệu lên



**Bài 2: Tạo model Banner và trả về danh sách banner gồm, title, image**

Các banner sẽ hiển thị tại trang welcome.blade.php **theo cáu trúc carousel của bootstrap 3 hoặc 4, 5**

Màn hình có dạng sau

