



## **LARAVEL FRAMEWORK**

### **BÀI 2: MÔ HÌNH MVC LARAVEL FRAMEWORK**

- ⊙ Hiểu mô hình MVC
- ⊙ Sử dụng MVC trong Laravel



## Phần I: Tìm hiểu mô hình MVC

 Giới thiệu mô hình MVC

 MVC trong Laravel

 Route, Model, Controller, View

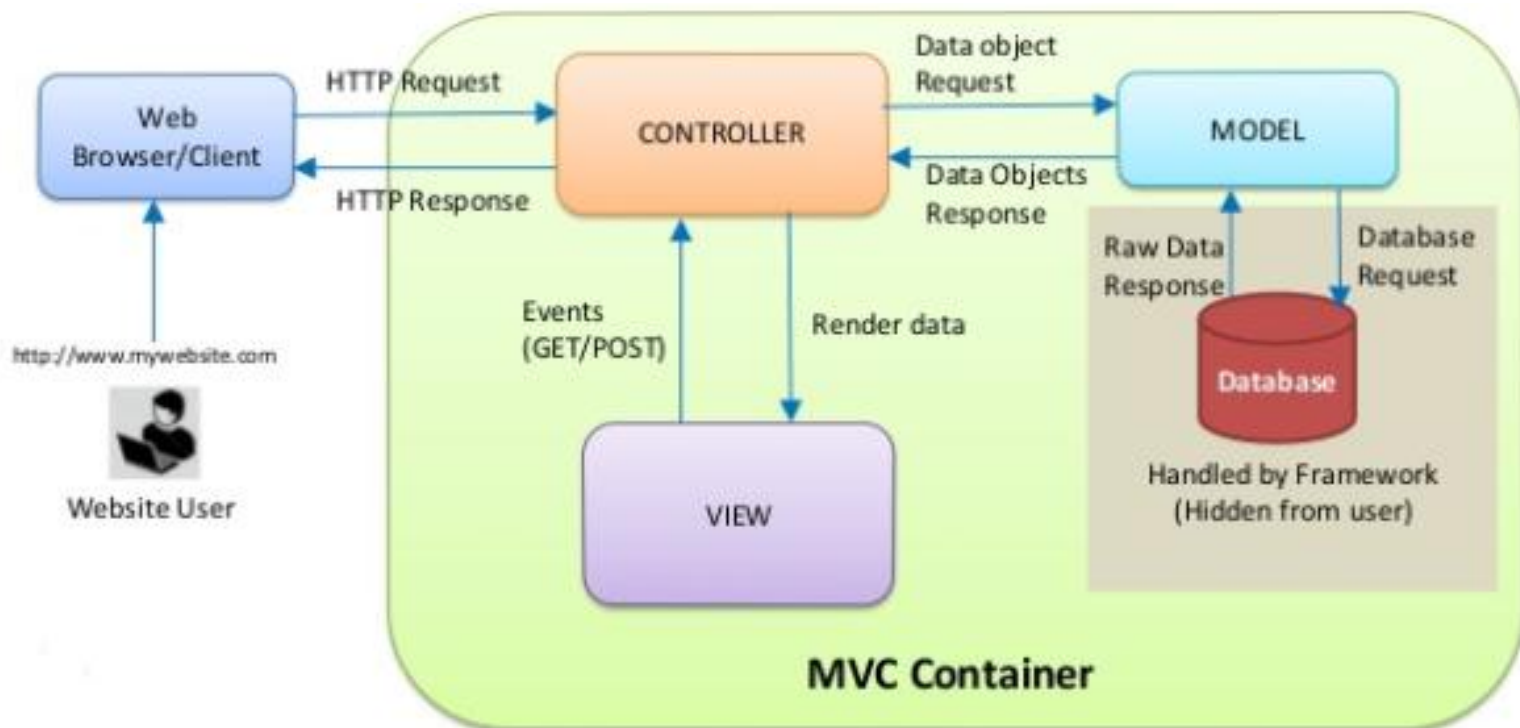
## Phần II: Ứng dụng MVC trong Laravel

 Lập trình MVC với Laravel

 Demo project



- ❑ Mô hình **MVC** (model, controller, view) là mô hình chuẩn cho ứng dụng web được sử dụng nhiều nhất ngày nay



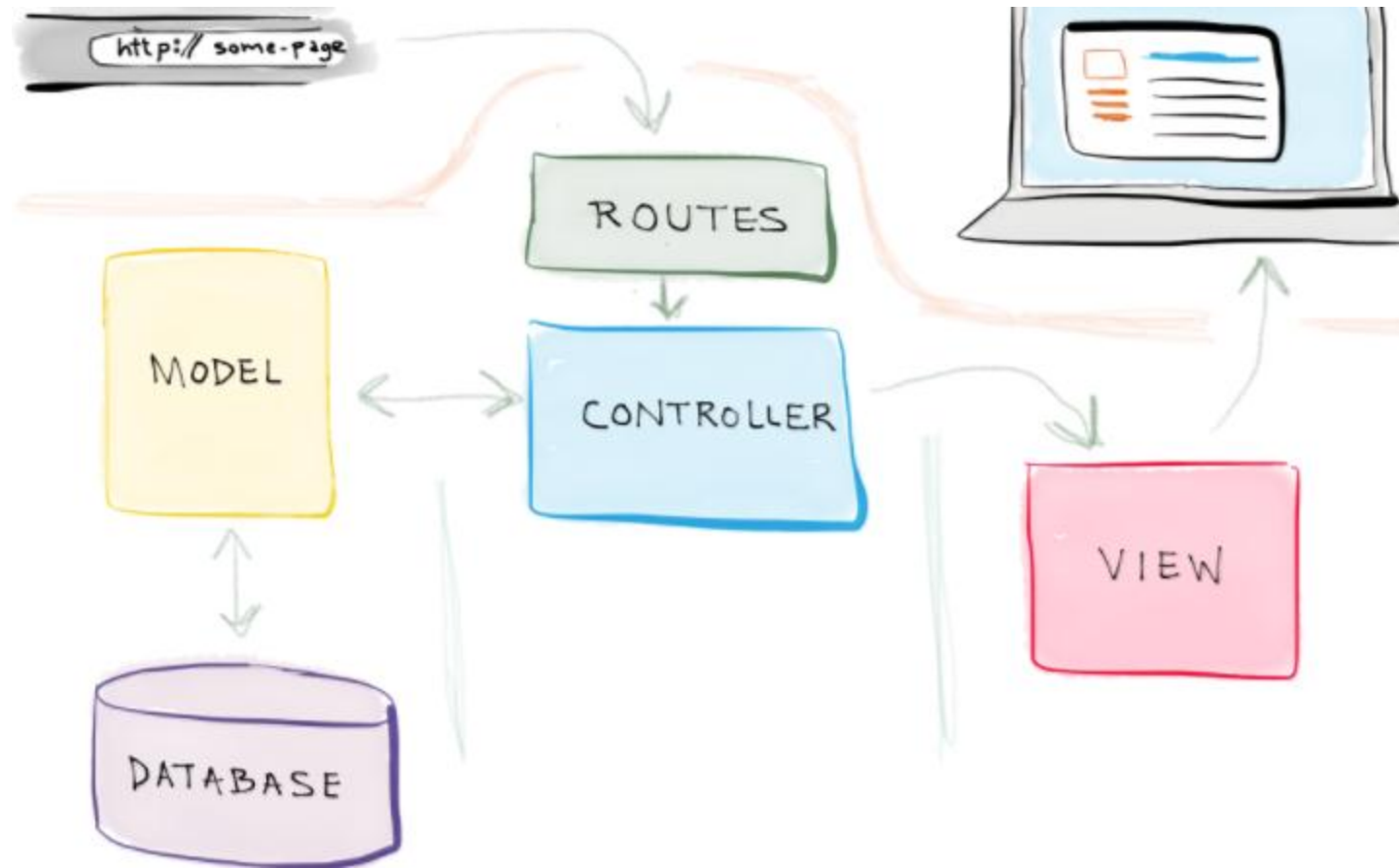
❑ Mô hình MVC là viết tắt của 3 chữ Model, View, Controller. Mô hình này tách một ứng dụng web ra làm 3 thành phần đảm nhiệm chức năng tách biệt, thuận tiện cho việc xử lý và bảo trì

❖ **Model** : Chịu trách nhiệm quản lý dữ liệu, nó lưu trữ và truy xuất các thực thể từ cơ sở dữ liệu như mysql, sql server, postgresSQL,... đồng thời chứa các logic được thực thi bởi ứng dụng

❖ **View** : Chịu trách nhiệm hiển thị dữ liệu đã được truy xuất từ model theo một format nào đó theo ý đồ của lập trình viên. Cách sử dụng của View tương tự như các module templates thường thấy trong các ứng dụng web phổ biến như WordPress, Joomla,...

- ❑ Mô hình MVC là viết tắt của 3 chữ Model, View, Controller. Mô hình này tách một ứng dụng web ra làm 3 thành phần đảm nhiệm chức năng tách biệt, thuận tiện cho việc xử lý và bảo trì
  - ❖ **Controller** : trung gian, làm nhiệm vụ xử lý cho model và view tương tác với nhau. Controller nhận request từ client, sau đó gọi các model để thực hiện các hoạt động được yêu cầu và gửi ra ngoài View. View sẽ chịu trách nhiệm format lại data từ controller gửi ra và trình bày dữ liệu theo 1 định dạng đầu ra (html).

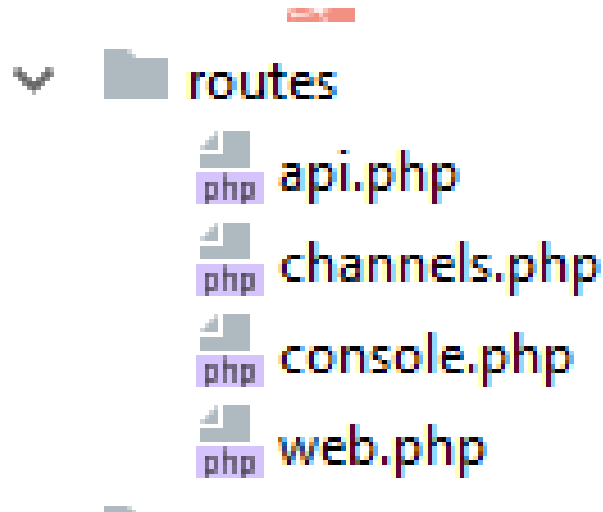
- ❑ Laravel Framework hỗ trợ lập trình theo mô hình MVC khá mạnh mẽ.
- ❑ Ngoài các thành phần chính Model, View, Controller thì Routes được sử dụng định tuyến người dùng theo đúng Urls.
- ❑ Cách thức hoạt động MVC trong Laravel như hình:





- ☐ User tạo ra một yêu cầu với URL dựa trên ứng dụng
- ☐ Xác định "route" tương ứng với URL của user, chuyển tới controller tương ứng
- ☐ Controller xử lý nghiệp vụ, nếu cần thiết thì truy vấn dữ liệu từ model và trả thông tin cho View
- ☐ View cung cấp thông tin trả về cho user

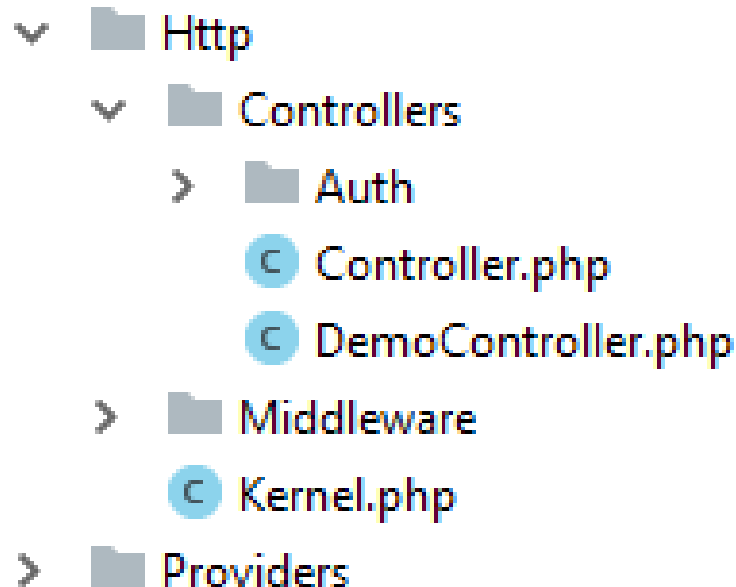
- ❑ Route được định nghĩa trong thư mục routes, được chia ra 2 file web.php và api.php giúp thêm tùy chọn phát triển web và api riêng.



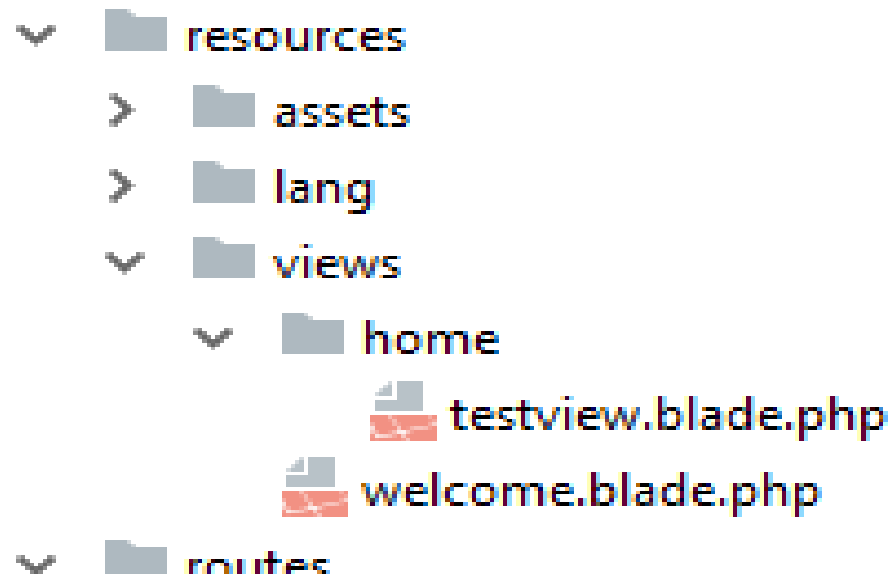
❑ Một số phương thức của route trong Laravel:

```
Route::get($uri, $callback);  
Route::post($uri, $callback);  
Route::put($uri, $callback);  
Route::patch($uri, $callback);  
Route::delete($uri, $callback);  
Route::options($uri, $callback);
```

- ❑ Trong Laravel thì controller được tạo ra trong thư mục Controllers, cụ thể là app/Http/Controllers



- ❑ Trong Laravel thì view được tạo ra trong thư mục resource/views
- ❑ View có thể là file .php hoặc .blade.php(blade template)



- ❑ Trong Laravel thì view được tạo ra trong thư mục resource/views
- ❑ View có thể là file .php hoặc .blade.php(blade template)



DEMO

Tạo Route,Controller,View



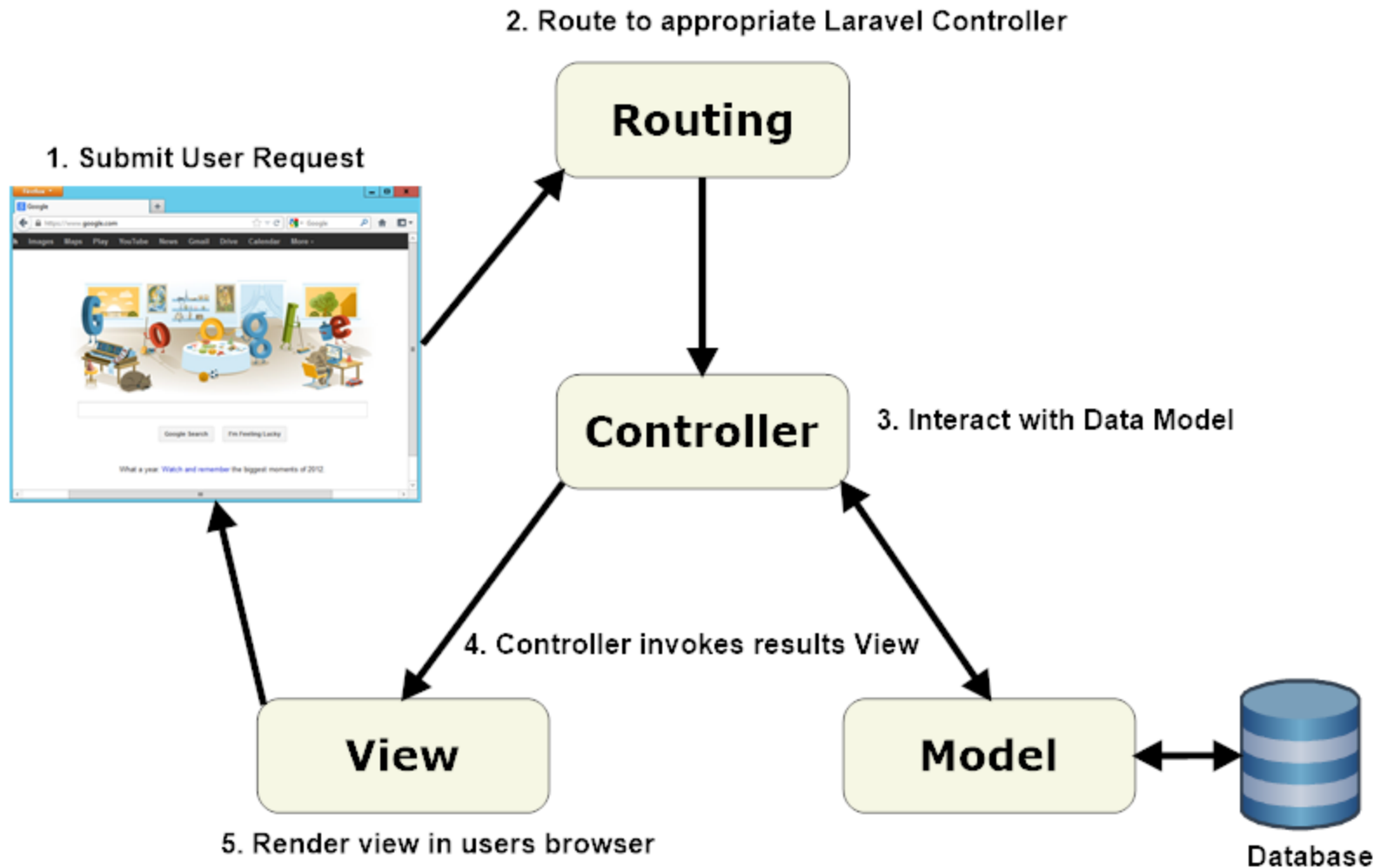


# **LARAVEL FRAMEWORK**

## **BÀI 2 (PHẦN 2)**



# WORKFLOW MVC TRONG LARAVEL



- ❑ Trong Laravel có nhiều loại route, 2 loại phổ biến được dùng nhiều là `Route::get` và `Route::post`
- ❑ `Route::get('/', 'WelcomeController@index');`
  - ❖ `'/'` thể hiện cho thư mục root (vd thư mục web là Poly, thì link sẽ là: `http://localhost/Poly`)
  - ❖ `WelcomeController` => là tên của Controller sử dụng
  - ❖ `index` => là phương thức trong Controller
  - ❖ Trên browser gõ `http://localhost/Poly`

```
Route::get('/name', function () {  
    return view('hello')->with('name', 'Warodom');  
});
```

## □ Basic Routing

```
Route::get('foo', function () {  
    return 'Hello World';  
});
```

## □ Redirect Routes

```
Route::redirect('/here', '/there', 301);
```

## □ View Routes

```
Route::view('/welcome', 'welcome');
```

```
Route::view('/welcome', 'welcome', ['name' => 'Taylor']);
```

## □ Route Parameters

```
Route::get('user/{id}', function ($id) {  
    return 'User '.$id;  
});
```

```
Route::get('posts/{post}/comments/{comment}', function ($postId, $commentId) {  
    //  
});
```

## □ Đặt tên cho route

```
// cach 1  
Route::get("/post", "TestController@test")->name('post');  
  
// cach 2  
Route::get("/post", ['as' => 'post', function() {  
    // do something in here  
}]);  
  
// cach 3  
Route::get("/post", [  
    'as' => 'post',  
    'uses' => 'TestController@test'  
]);
```

## ❑ Tạo model

- ❖ Sử dụng Cmd và đến thư mục gốc của project
- ❖ php artisan make:model Flight : tạo model tên Flight

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Flight extends Model
{
    //
}
```

## ❑ Tạo model

- ❖ Table Name: Khi tạo model nếu không chỉ định table nào thì Laravel lấy tên model thêm 's' vào cuối để làm tên table tương ứng
- ❖ Tạo model tên Flight và không chỉ định table thì mặc định model sẽ liên kết với table Flight**s**
- ❖ Có thể chỉ định table và PrimaryKey khi tạo model

```
protected $primaryKey = 'id';  
protected $table = 'categories';
```

## ❑ Tạo model

- ❖ Record Timestamps: Mặc định thêm 2 field vào table là: create\_at và update\_at.
- ❖ Nếu không muốn những field này tự động được quản lý bởi Eloquent, thiết lập thuộc tính \$timestamps thành false

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Flight extends Model
{
    /**
     * Indicates if the model should be timestamped.
     *
     * @var bool
     */
    public $timestamps = false;
}
```

- ❑ Tạo trực tiếp tại thư mục app/Http/Controller
- ❑ Tạo bằng artisan command

```
php artisan make:controller DemoController
```

```
namespace App\Http\Controllers;  
  
class DemoController extends Controller  
{  
    //  
}
```



## ❑ Tạo action trong Controller

```
class ShowProfile extends Controller
{
  /**
   * Show the profile for the given user.
   *
   * @param int $id
   * @return Response
   */
  public function __invoke($id)
  {
    return view('user.profile', ['user' => User::findOrFail($id)]);
  }
}
```

## ❑ Tạo Resource Controllers

```
php artisan make:controller PhotoController --resource
```

```
Route::resource('photos', 'PhotoController');
```

| Verb      | URI                  | Action  | Route Name     |
|-----------|----------------------|---------|----------------|
| GET       | /photos              | index   | photos.index   |
| GET       | /photos/create       | create  | photos.create  |
| POST      | /photos              | store   | photos.store   |
| GET       | /photos/{photo}      | show    | photos.show    |
| GET       | /photos/{photo}/edit | edit    | photos.edit    |
| PUT/PATCH | /photos/{photo}      | update  | photos.update  |
| DELETE    | /photos/{photo}      | destroy | photos.destroy |

## □ Tạo view

```
<!-- View stored in resources/views/greeting.blade.php -->

<html>
    <body>
        <h1>Hello, {{ $name }}</h1>
    </body>
</html>
```

## □ Tạo view

```
<!-- View stored in resources/views/greeting.blade.php -->

<html>

  <body>

    <h1>Hello, {{ $name }}</h1>

  </body>

</html>
```

## □ Gọi View

```
Route::get('/', function () {
    return view('welcome');
});

class DemoController extends Controller
{
    public function test1()
    {
        return view( view: 'home.testview' );
    }
}
```

## ❑ Truyền dữ liệu qua view

```
class HelloController extends BaseController {  
    public function showWelcome($name='stranger')  
    {  
        return View::make('hello', compact('name')) ;  
    }  
}
```

```
<html>  
    <body>  
        <p>Hello, {{{ $name }}}!</p>  
    </body>  
</html>
```



DEMO

Tạo web MVC Laravel



## Phần I: Tìm hiểu mô hình MVC

 Giới thiệu mô hình MVC

 MVC trong Laravel

 Route, Controller, View

## Phần II: Ứng dụng MVC trong Laravel

 Lập trình MVC với Laravel

 Demo project





**Cảm ơn**