

CSE485 – Công nghệ Web

dungkt@tlu.edu.vn



Back-end Tech Stack for Web Development

Programming languages



Web servers



Frameworks

django

for Python



for PHP



for JavaScript

Operating systems



Database languages



Bài 6. Laravel Framework (Phần 1)

NỘI DUNG

1. Giới thiệu về Laravel
2. Kiến trúc của Laravel và cài đặt
3. Cấu trúc dự án Laravel
4. Routing và controller
5. Quy trình tạo ứng dụng Laravel đơn giản



1. Giới thiệu về Laravel

- Laravel được tạo ra bởi Taylor Otwell và lần đầu tiên xuất hiện vào năm 2011
- Laravel là cung cấp một framework PHP có cú pháp sạch sẽ và đơn giản, giúp giảm thiểu công sức cần thiết cho các nhiệm vụ lập trình phổ biến.
- Phiên bản mới nhất hiện tại: 10.x (<https://laravel.com/docs/10.x>)
- Đặc điểm nổi bật:
 - **Eloquent ORM**: Tương tác với cơ sở dữ liệu trở nên dễ dàng và trực quan thông qua cú pháp hướng đối tượng
 - **Blade Template Engine**: Hệ thống template mạnh mẽ và linh hoạt, cho phép lập trình viên tạo ra các layout động một cách dễ dàng.
 - **Migrations và Seeders**: Quản lý cơ sở dữ liệu trở nên thuận tiện hơn với khả năng di chuyển (migrations) và cấy dữ liệu (seeders)
 - **Artisan CLI**: Giao diện dòng lệnh (CLI) mạnh mẽ, giúp tự động hóa nhiều tác vụ như tạo models, controllers, và thậm chí là cấu trúc cơ sở dữ liệu.
 - **Middleware, Routing, và Authentication**: các công cụ mạnh mẽ cho việc xác thực người dùng, điều hướng và middleware, cho phép lập trình viên dễ dàng quản lý và bảo mật ứng dụng của mình

1. Giới thiệu về Laravel



- Lý do phổ biến:
 - **Cộng đồng và Hỗ Trợ:** Laravel có một cộng đồng lập trình viên rất lớn và đa dạng, cung cấp nguồn lực học tập, hỗ trợ, và plugin phong phú.
 - **Hiệu suất và Tính năng:** Laravel không ngừng cập nhật với các tính năng mới, tối ưu về hiệu suất và bảo mật.
 - **Tài liệu và Học tập:** Laravel cung cấp tài liệu đầy đủ và dễ hiểu, giúp người mới bắt đầu có thể tiếp cận dễ dàng.
 - **Kiến trúc MVC:** Model-View-Controller (MVC) là kiến trúc phổ biến trong phát triển web, và Laravel triển khai kiến trúc này một cách hiệu quả, giúp quản lý mã nguồn trở nên rõ ràng và dễ dàng bảo trì.

2. Kiến trúc Laravel và cài đặt

- Kiến trúc Laravel:
 - Laravel sử dụng kiến trúc Model-View-Controller (MVC), một mô hình phổ biến trong phát triển ứng dụng web. Kiến trúc MVC giúp tách biệt logic ứng dụng, giao diện người dùng và tương tác với cơ sở dữ liệu, đem lại sự rõ ràng và dễ bảo trì cho mã nguồn.
 - **Model:** Trong Laravel, Model đại diện cho dữ liệu và chứa logic xử lý dữ liệu. Eloquent ORM của Laravel giúp làm việc với cơ sở dữ liệu trở nên trực quan và dễ dàng thông qua các lệnh truy vấn hướng đối tượng.
 - **View:** View là thành phần hiển thị thông tin và giao diện người dùng. Laravel sử dụng Blade, một hệ thống template mạnh mẽ, để quản lý các file giao diện. Blade cho phép bạn tạo các template kế thừa, section và component, giúp quản lý giao diện trở nên linh hoạt và dễ dàng.
 - **Controller:** Controller xử lý logic ứng dụng và tương tác người dùng. Controllers trong Laravel nhận yêu cầu, xử lý logic nghiệp vụ (thông qua Model nếu cần) và trả về response, thường là một View với dữ liệu.

2. Kiến trúc Laravel và cài đặt

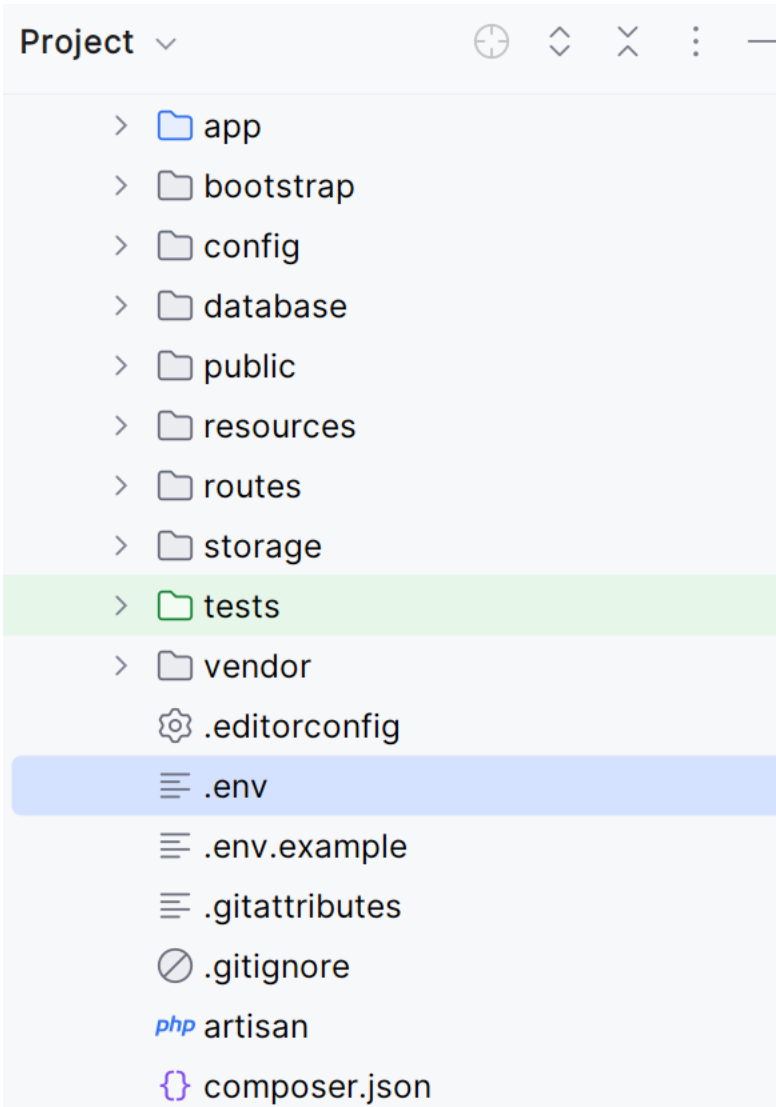


- Cài đặt Laravel:
 - Cài đặt composer: <https://getcomposer.org/>
 - Xampp: chỉ cần bật MySQL (ko cần bật Apache nữa)
 - Tạo dự án Laravel:
 - **Cách 1:** Tạo dự án Laravel trực tiếp:
 - `composer create-project --prefer-dist laravel/laravel ten-du-an`
 - **Cách 2:** Cài đặt thông qua Laravel Installer:
 - Cài đặt Laravel Installer: `composer global require laravel/installer`
 - Tạo dự án: `laravel new ten-du-an`
 - Cài đặt biến môi trường:
 - Sau khi tạo dự án, bạn cần cấu hình môi trường ứng dụng. Chỉnh sửa tệp **.env** trong thư mục gốc của dự án để thiết lập các thông tin cần thiết như kết nối cơ sở dữ liệu, mail server, v.v.
 - Chạy dự án:
 - `cd ten-du-an`
 - `php artisan serve`
 - Truy cập dự án: `http://localhost:8000`



```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=ten_csdldb
DB_USERNAME=nguoidung
DB_PASSWORD=matkhau
```

3. Cấu trúc dự án Laravel



/app: Chứa code của ứng dụng, bao gồm Models, Controllers, và các lớp PHP khác.
/Console: Chứa các lệnh Artisan do người dùng định nghĩa.
/Http: Chứa Controllers, Middleware và Request.
/Models: (từ Laravel 8 trở đi) Thư mục cho các Models Eloquent.
/bootstrap: Chứa tệp app.php khởi động framework và tệp cache dùng để tối ưu hiệu suất.
/config: Chứa các tệp cấu hình của ứng dụng.
/database: Chứa migrations, model factories, và seeds.
/public: Thư mục gốc của web server; chứa index.php, tệp tin này là điểm vào của ứng dụng.
/resources: Chứa các tài nguyên như Blade templates, CSS, JavaScript.
/routes: Chứa các tệp định tuyến cho ứng dụng.
/storage: Chứa các tệp được tạo ra trong quá trình chạy ứng dụng, như logs, tệp cache, và tệp được tải lên.
/tests: Chứa các bài kiểm thử tự động.
/vendor: Chứa các gói thư viện của Composer.

.env: Tệp cấu hình môi trường, nơi bạn đặt các thông tin như thông tin kết nối cơ sở dữ liệu, mail, v.v.

composer.json: Quản lý các phụ thuộc của Composer.



4. Routing và Controller

- **Routing:** Quá trình định nghĩa các đường dẫn (URLs) cho ứng dụng và xác định hành động sẽ được thực hiện khi yêu cầu đến một đường dẫn cụ thể.
- Routes được định nghĩa trong các tệp trong thư mục **routes**. Có một số loại routes cơ bản:
 - Web Routes (**routes/web.php**): Dùng cho ứng dụng web, hỗ trợ session state, CSRF protection, v.v.
 - API Routes (routes/api.php): Dùng cho API, không bao gồm state của session.

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

4. Routing và Controller

- **Controllers:** Trong Laravel xử lý logic của ứng dụng và được kết nối với các routes.
 - Tạo Controller: `php artisan make:controller ExampleController`
 - Định nghĩa Phương thức trong Controller:

```
namespace App\Http\Controllers;

class ExampleController extends Controller
{
    public function show()
    {
        return 'Hello from the Controller';
    }
}
```

- Kết nối Route với Controller

```
use App\Http\Controllers\ExampleController;
use Illuminate\Support\Facades\Route;

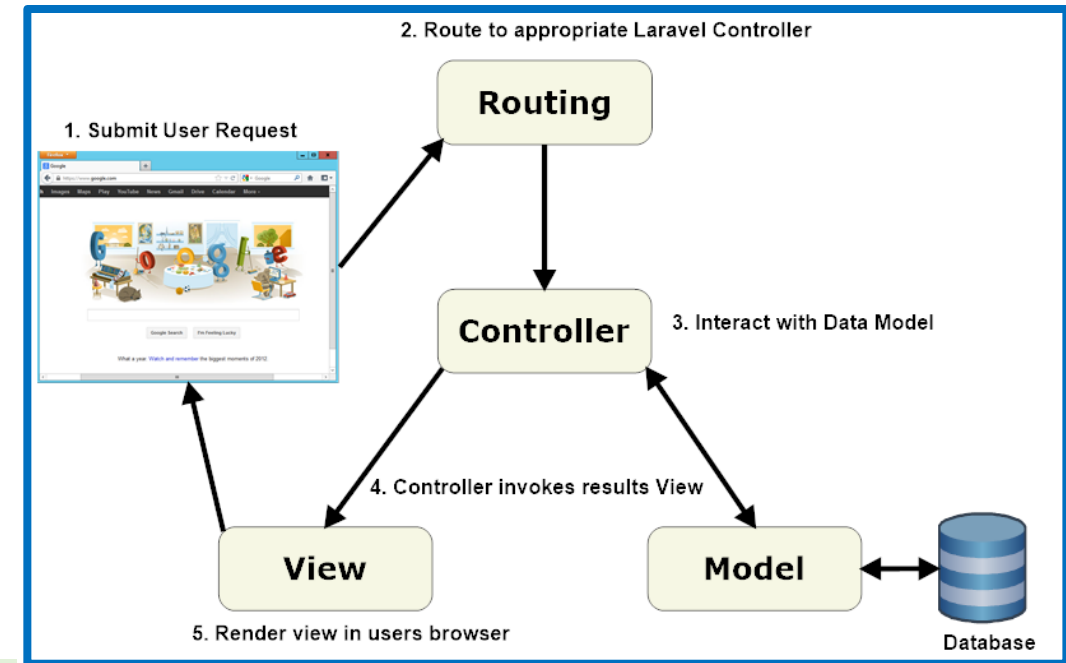
Route::get('/example', [ExampleController::class, 'show']);
```

5. Quy trình tạo ứng dụng Laravel đơn giản



- **Bước 1:** Tạo dự án Laravel
 - `laravel new lr-project`
- **Bước 2:** Tạo CSDL, chẳng hạn, `lr-project`
- **Bước 3:** Mở tệp tin `.env`, khai báo kết nối tới CSDL
- **Bước 4:** Tạo lần lượt các bảng, chẳng hạn bảng `posts`
 - `php artisan make:migration create_posts_table`
- **Bước 5:** Định nghĩa cấu trúc bảng `posts`:

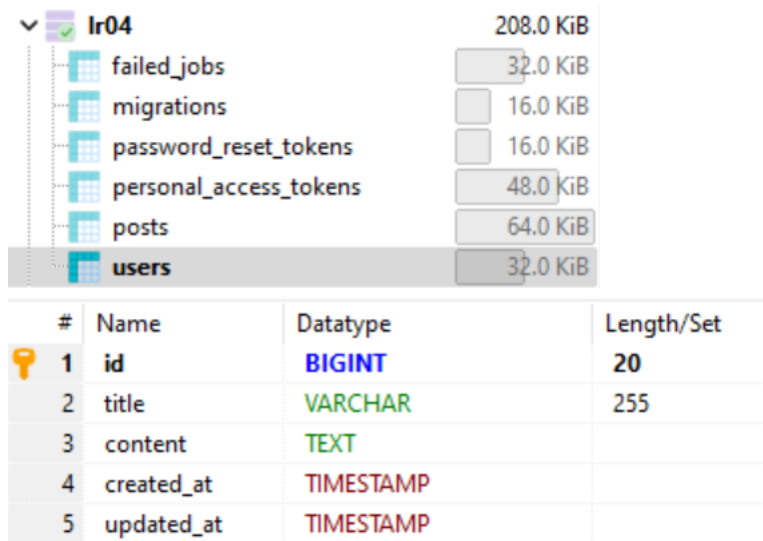
```
public function up(): void
{
    Schema::create('posts', function (Blueprint $table) {
        $table->id();
        $table->string("title");
        $table->text("content");
        $table->timestamps();
    });
}
```



5. Qui trình tạo ứng dụng Laravel đơn giản

- Bước 6:** Thực thi các tệp migration để tạo bảng

- `php artisan migrate`



#	Name	Datatype	Length/Set
1	id	BIGINT	20
2	title	VARCHAR	255
3	content	TEXT	
4	created_at	TIMESTAMP	
5	updated_at	TIMESTAMP	

- Bước 7:** Tạo Seeder để sinh dữ liệu minh họa

- `php artisan make:seeder PostsTableSeeder`
- `php artisan db:seed --class=PostsTableSeeder`

```
<?php

namespace Database\Seeders;

use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;
use Faker\Factory as Faker;
use Illuminate\Support\Facades\DB;
class PostsTableSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        $faker = Faker::create();

        for ($i = 0; $i < 100; $i++) {
            DB::table('posts')->insert([
                'title' => $faker->sentence,
                'content' => $faker->paragraph
            ]);
        }
    }
}
```

5. Qui trình tạo ứng dụng Laravel đơn giản

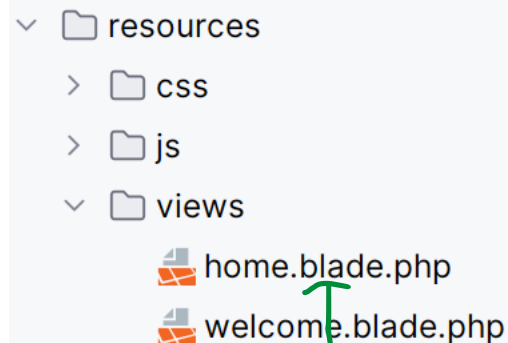


- **Bước 8:** Tạo model

- `php artisan make:model post`
- **Lưu ý:** Tên bảng nên là hình thức số nhiều **posts/categories** và tên model tương ứng là số ít **Post/Category** để Laravel tự map **Model - Table**

- **Bước 9:** Tạo Controller

- `php artisan make:controller PostController -r` (có hoặc không có `-r/-resource`)
- Laravel tự tạo 7 phương thức theo chuẩn, chẳng hạn nội dung của phương thức `index()`:



```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Post;
class HomeController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        $posts = Post::all();
        return view("home", compact("posts"));
    }
}
```

5. Qui trình tạo ứng dụng Laravel đơn giản

- **Bước 10:** Tạo Blade Template
 - php artisan make:**view** home
- **Bước 11:** Định nghĩa nội dung của [home.blade.php](#)

```
@foreach($posts as $post)
    <p>{{ $post->content }}</p>
@endforeach
```

5. Quy trình tạo ứng dụng Laravel đơn giản

- **Bước 12: Sửa đổi route**
 - Mở tệp routes/web.php

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\PostController;
use App\Http\Controllers\HomeController;

Route::get('/', [HomeController::class, "index"]);

Route::get("posts", [PostController::class, "index"]);
```

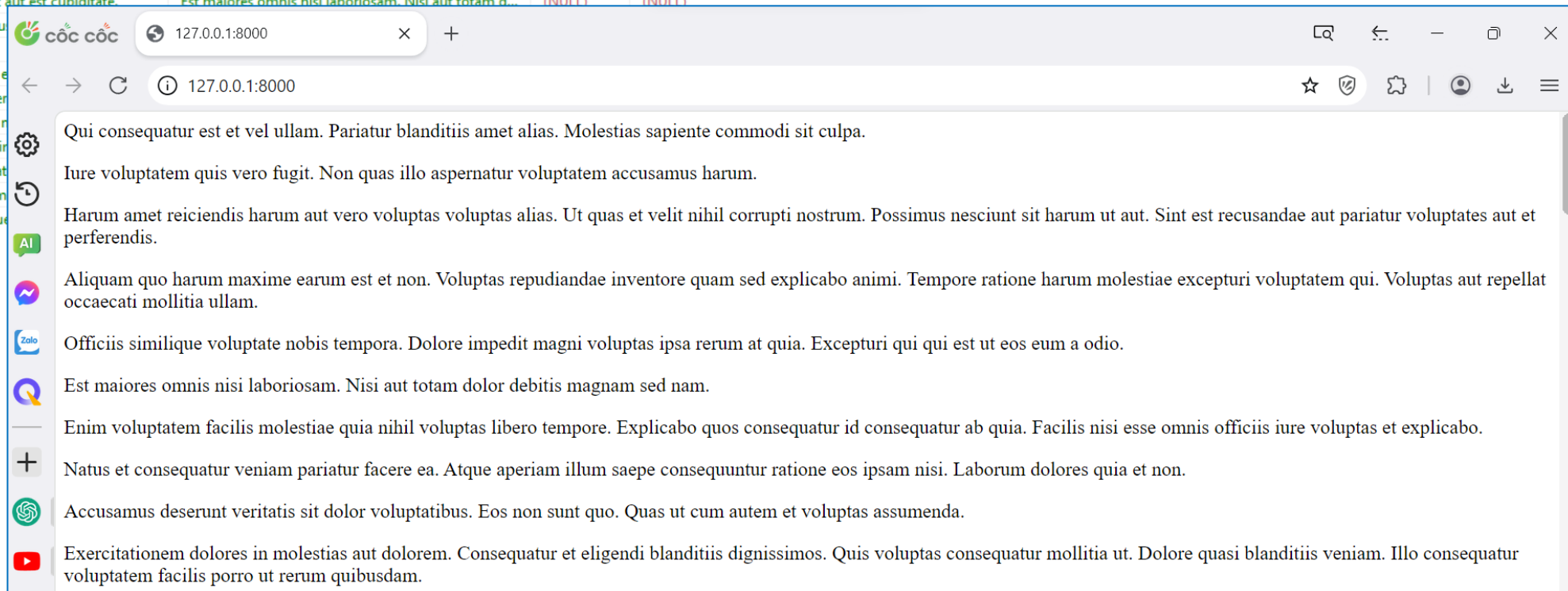
5. Qui trình tạo ứng dụng Laravel đơn giản



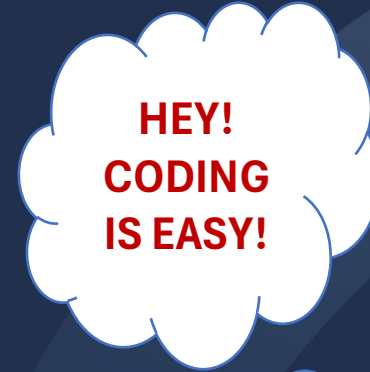
- **Kết quả:**

- **Chạy lệnh: `php artisan serve`**

id	title	content	created_at	updated_at
1	Aut voluptatem repudiandae maxime exercitatione...	Qui consequatur est et vel ullam. Pariatur blanditiis ...	(NULL)	(NULL)
2	Ut rerum laboriosam repellendus aliquam blanditiis.	Iure voluptatem quis vero fugit. Non quas illo asper...	(NULL)	(NULL)
3	Amet fuga magnam sequi repudiandae qui.	Harum amet reiciendis harum aut vero voluptas vol...	(NULL)	(NULL)
4	Beatae sequi dolor autem sed ullam nihil.	Aliquam quo harum maxime earum est et non. Volu...	(NULL)	(NULL)
5	Ut et neque sit consequatur.	Officiis similique voluptate nobis tempora. Dolore i...	(NULL)	(NULL)
6	Ut deleniti soluta at deserunt aut est cupiditate.	Est maiores omnis nisi laboriosam. Nisi aut totam d...	(NULL)	(NULL)
7	Consequuntur distinctio natu			
8	Qui earum minima qui.			
9	Repellendus qui voluptatem e			
10	Quia itaque perferendis qui er			
11	Dolorem temporibus debitis n			
12	Animi ut aut alias rerum ut sir			
13	Et aliquam excepturi aspernat			
14	Quia unde repellendus assum			
15	Nostrum autem fugit cumque			



“Câu hỏi & Thảo luận”



THE END!

