**Bài thực hành 10**

**Global function**

***Mục tiêu***

* Tìm hiểu các global function thường sử dụng
* Tự tạo hêm các glocal function cho dự án
* Global function là các hàm sử dụng toàn cục rong dự án Laravel
* Nó cung cấp giải pháp sử dụng các đối tượng của Laravel mà không cần hởi tạo hay include
* Có thể sử dugj bất cưa nơi đâu trong dự án Laravel
* Một số hàm thường dung như: auth(), dd(), url(), app\_part(), base\_part(), public\_part(), request(), response(), session(), Str::slug….
* Người dung có thể tạo hêm các global function để sử dụng tương tự trong dự án của mình

***Thực hành theo các bài tập***

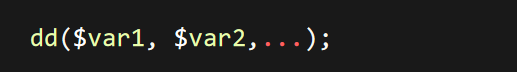
**Bài 1: Tìm hiểu về các global function thường sử dụng**

**Hàm dd():** Hàm này sử dụng khi muốn in ra dữ liệu của bất kỳ biến nào

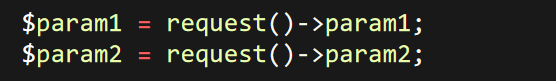
Sử dụng thay thế hàm **print\_r()** và **var\_dump()** của php thuần

Tham số truyền vào là một hoặc nhiều biến phân biệt bởi dấu phẩy

**Cú pháp** ( Bất kỳ chỗ nào bạn muốn kiểm tra dữ liệu của biến )



**Hàm request():** Hàm này thay thế cho đối tương Request, sử dụng bất cứ nơi nào khi bạn muốn nhận dữ liệu được gửi lên từ người dung dưới dạng các tham số url, tham số route. Cú pháp đơn giản là:



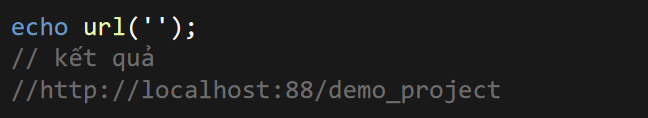
**Hàm url(path):** Hàm này trả về đường dẫn url gốc của dự án

Tham số path là bắt buộc, tham số thường là tên thư mục tiếp theo bạn muốn truy cập vào

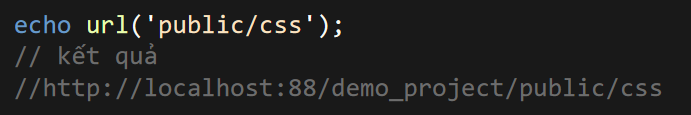
Nếu muốn lấy đường dẫn gốc của dự án thì tham số là 2 dấu nháy

VD dự án của bạn hiện tại đặt trong htdocs với tên là demo\_project

Nếu bạn sử dụng hàm url(‘’) thì kết quả có dạng



VD muốn đi vào thư mục public/css



Sư dụng để lấy các file css, image, js…

VD:



**Hàm base\_path(path):** Trả về đường dẫn tuyệt dối của dự án trên máy chủ

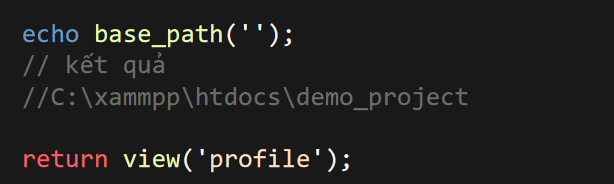
Thường sử dụng để lấy về đường dẫn tuyệt đối cho các trường hợp như aplaod file, truy cập đọc ghi file…

**Tham số path** thường là tên thư mục tiếp theo bạn muốn truy cập vào

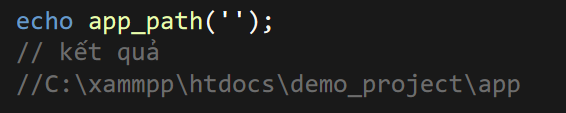
Nếu muốn lấy đường dẫn gốc của dự án thì tham số là 2 dấu nháy

VD dự án của bạn hiện tại đặt trong htdocs với tên là demo\_project

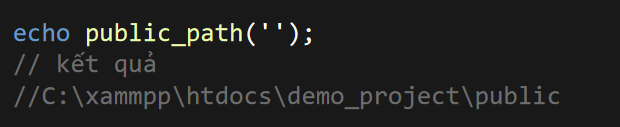
Nếu bạn sử dụng hàm url(‘’) thì kết quả có dạng



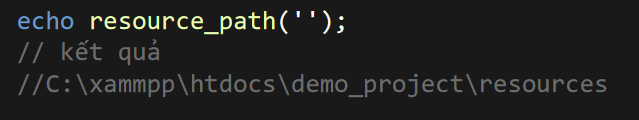
**Hàm app\_path(path): trả** về đường dẫn tuyệt đối của thư mục app



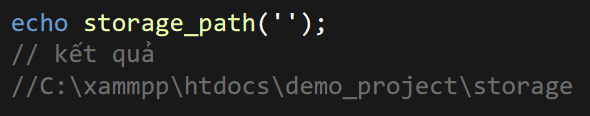
**Hàm public\_path(path):** trả về đường dẫn tuyệt đối tói thư mục public



**Hàm resource\_path(path):** trả về đường dẫn tuyệt đối tới thư mục resources



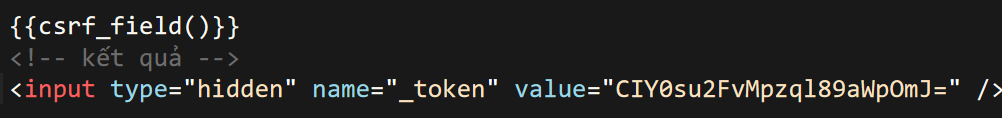
Hàm storage\_path(path): trả về đường dẫn tuyệt đối tới thư mục storage



**Hàm csrf\_token():** trả về mã token khi sử dụng form post, put..



**Hàm csrf\_field():** tạo thẻ input hidden \_token cho form token



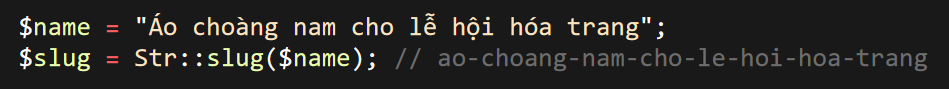
**Hàm Str::slug(string):** chuyển đổi chuỗi có dấu sang dạng slug

Thường dừng cho viecj tạo ra các url SEO

VD với tên sản phẩm là: **Áo choàng nam cho lễ hội hóa trang**

Thì slug có dạng: **ao-choang-nam-cho-le-hoi-hoa-trang**

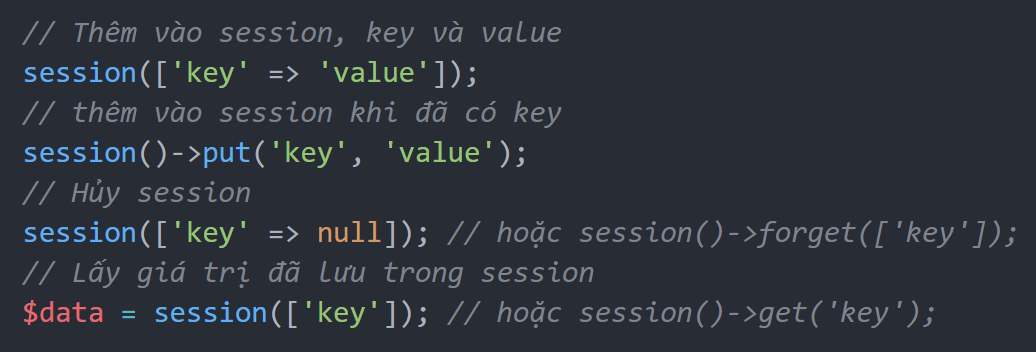
**Cú pháp sử dụng**



**Ngoài ra còn nhiều hàm khác, các bạn có thể tham khảo trên trang chủ**

[**https://laravel.com/docs/5.1/helpers#available-methods**](https://laravel.com/docs/5.1/helpers#available-methods)

**Hàm session():** Cho phép thực hiện với session dễ dàng hơn



**Bài 2: Tạo ứng dụng đăng nhập, đăng ký đơn giản sử dụng hàm session**

Bước 1: Tạo bảng **accounts** có cấu trúc sau

CREATE TABLE IF NOT EXISTS `accounts` (

  id INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,

  name VARCHAR(100) NOT NULL,

  email VARCHAR(100) NOT NULL UNIQUE,

  phone VARCHAR(50) NOT NULL,

  password VARCHAR(100) NOT NULL,

  address VARCHAR(200) NULL,

  gender tinyint(1) DEFAULT '1',

  last\_login date default NULL,

  created\_at date default CURRENT\_TIMESTAMP,

updated\_at date default NULL

);

**Bước 2:** Tạo model Account

Trong thư mục app\Models tạo thêm file **Account.php**

Sau đó copy toàn bộ code bên file app\User.php qua và sửa lại thành như sau

<?php

namespace App\Models;

use Illuminate\Contracts\Auth\MustVerifyEmail;

use Illuminate\Foundation\Auth\User as Authenticatable;

use Illuminate\Notifications\Notifiable;

class **Account** extends Authenticatable

{

    use Notifiable;

    /\*\*

     \* The attributes that are mass assignable.

     \*

     \* @var array

     \*/

    protected $fillable = [

        'name', 'email', 'password','phone','address','gender','last\_login'

    ];

    /\*\*

     \* The attributes that should be hidden for arrays.

     \*

     \* @var array

     \*/

    protected $hidden = [

        'password',

    ];

}

**Bước 3: Tạo route đăng nhập đăng ký**

Route::group( ['prefix' => 'account'], function () {

    // Phương thức get hiển thị form login

    Route::get('login', 'AccountController@login')->name('account.login');

    //Phương thức post để thực hiện login khi submit form

    Route::post('login', 'AccountController@post\_login');

    // Phương thức get hiển thị form register

    Route::get('register', 'AccountController@register')->name('account.register');

    //Phương thức post để thực hiện register khi submit form

    Route::post('register', 'AccountController@post\_register');

});

**Bước 4:** Tạo controller **AccountController** và các phương thức tương ứng như đã khai báo trên các route

<?php

namespace App\Http\Controllers;

use Auth;

use Illuminate\Http\Request;

class AccountController extends Controller

{

    public function login()

    {

        // gọi view hiện hị form đăng nhập

        return view('account.login');

    }

    public function post\_login()

    {

       // code thực hiện đăng nhập

    }

    public function register()

    {

        // gọi view hiện hị form đăng ký

        return view('account. register);

    }

    public function post\_register()

    {

       // code thực hiện đăng ký

    }

}

**Bước 5: Tạo view account/login**

Tạo thêm file **resources/views/account/login.blade.php**

@extends('layouts.main')

@section('content')

<div class="container">

    <div class="row justify-content-center">

        <div class="col-md-8">

            <div class="card">

                <div class="card-header">{{ \_\_('Account Login') }}</div>

                <div class="card-body">

                    <form method="POST" action="{{ route('account.login') }}">

                        @csrf

                        <div class="form-group row">

                            <label for="email" class="col-md-4 col-form-label text-md-right">{{ \_\_('E-Mail Address') }}</label>

                            <div class="col-md-6">

                                <input id="email" type="email" class="form-control @error('email') is-invalid @enderror" name="email" value="{{ old('email') }}" required autocomplete="email" autofocus>

                                @error('email')

                                    <span class="invalid-feedback" role="alert">

                                        <strong>{{ $message }}</strong>

                                    </span>

                                @enderror

                            </div>

                        </div>

                        <div class="form-group row">

                            <label for="password" class="col-md-4 col-form-label text-md-right">{{ \_\_('Password') }}</label>

                            <div class="col-md-6">

                                <input id="password" type="password" class="form-control @error('password') is-invalid @enderror" name="password" required autocomplete="current-password">

                                @error('password')

                                    <span class="invalid-feedback" role="alert">

                                        <strong>{{ $message }}</strong>

                                    </span>

                                @enderror

                            </div>

                        </div>

                        <div class="form-group row">

                            <div class="col-md-6 offset-md-4">

                                <div class="form-check">

                                    <input class="form-check-input" type="checkbox" name="remember" id="remember" {{ old('remember') ? 'checked' : '' }}>

                                    <label class="form-check-label" for="remember">

                                        {{ \_\_('Remember Me') }}

                                    </label>

                                </div>

                            </div>

                        </div>

                        <div class="form-group row mb-0">

                            <div class="col-md-8 offset-md-4">

                                <button type="submit" class="btn btn-primary">

                                    {{ \_\_('Login') }}

                                </button>

                            </div>

                        </div>

                    </form>

                </div>

            </div>

        </div>

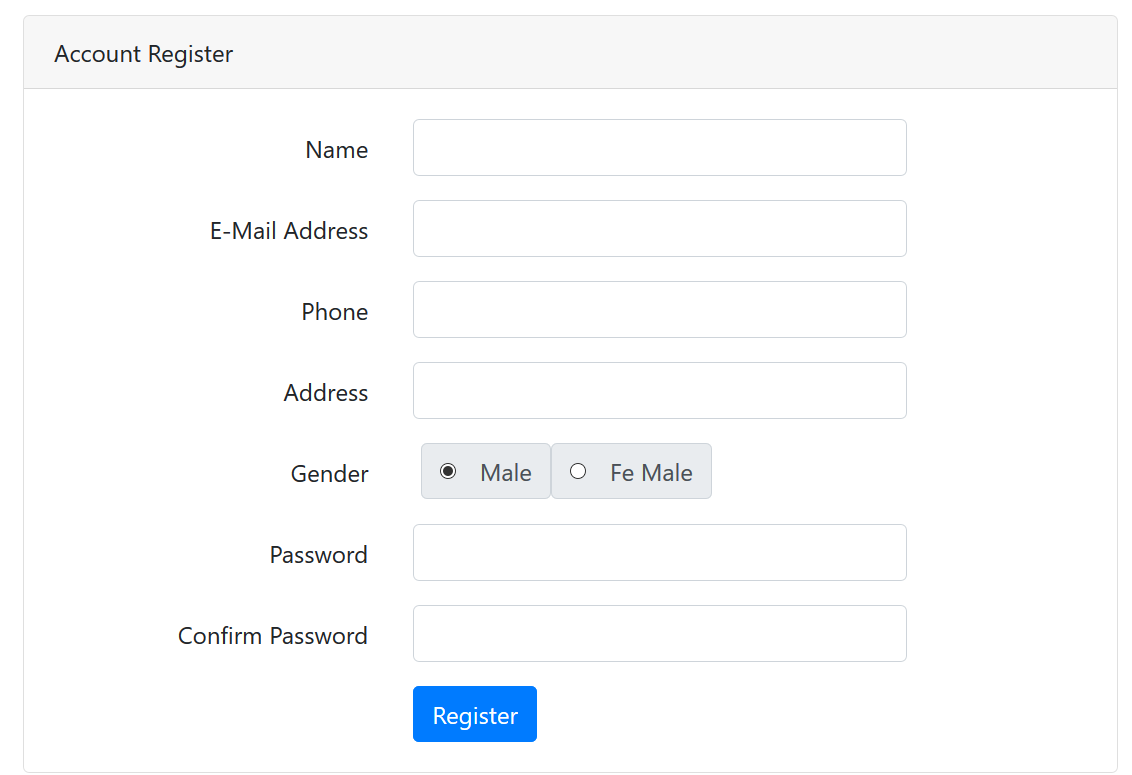
    </div>

</div>

@endsection

View trên trình duyệt có dạng

<http://localhost/project/account/login>



**Bước 7: Tạo view account/register**

Tạo file **resources/views/account/register.blade.php** và soạn code tạo form register như sau như sau

@extends('layouts.main')

@section('content')

<div class="container">

    <div class="row justify-content-center">

        <div class="col-md-8">

            <div class="card">

                <div class="card-header">{{ \_\_('Customer Register') }}</div>

                <div class="card-body">

                    <form method="POST" action="{{ route('account.register') }}">

                        @csrf

                        <div class="form-group row">

                            <label for="name" class="col-md-4 col-form-label text-md-right">{{ \_\_('Name') }}</label>

                            <div class="col-md-6">

                                <input id="name" type="text" class="form-control @error('name') is-invalid @enderror"

                                    name="name" value="{{ old('name') }}" required />

                                @error('name')

                                <span class="invalid-feedback" role="alert">

                                    <strong>{{ $message }}</strong>

                                </span>

                                @enderror

                            </div>

                        </div>

                        <div class="form-group row">

                            <label for="email"

                                class="col-md-4 col-form-label text-md-right">{{ \_\_('E-Mail Address') }}</label>

                            <div class="col-md-6">

                                <input id="email" type="email" class="form-control @error('email') is-invalid @enderror"

                                    name="email" value="{{ old('email') }}" required />

                                @error('email')

                                <span class="invalid-feedback" role="alert">

                                    <strong>{{ $message }}</strong>

                                </span>

                                @enderror

                            </div>

                        </div>

                        <div class="form-group row">

                            <label for="address"

                                class="col-md-4 col-form-label text-md-right">{{ \_\_('Address') }}</label>

                            <div class="col-md-6">

                                <input id="address" type="text"

                                    class="form-control @error('address') is-invalid @enderror" name="address"

                                    value="{{ old('address') }}" required  />

                                @error('address')

                                <span class="invalid-feedback" role="alert">

                                    <strong>{{ $message }}</strong>

                                </span>

                                @enderror

                            </div>

                        </div>

                        <div class="form-group row">

                            <label class="col-md-4 col-form-label text-md-right">{{ \_\_('Gender') }}</label>

                            <div class="form-check">

                                <div class="input-group">

                                    <div class="input-group-text">

                                        <input type="radio" checked class="mr-3" name="gender" value="1" id="male">

                                        <label class="form-check-label" for="male"> Male </label>

                                    </div>

                                    <div class="input-group-text">

                                        <input type="radio" class="mr-3" name="gender" value="0" id-="female">

                                        <label class="form-check-label" for="female"> Fe Male </label>

                                    </div>

                                </div>

                            </div>

                        </div>

                        <div class="form-group row">

                            <label for="password"

                                class="col-md-4 col-form-label text-md-right">{{ \_\_('Password') }}</label>

                            <div class="col-md-6">

                                <input id="password" type="password"

                                    class="form-control @error('password') is-invalid @enderror" name="password"

                                    required  />

                                @error('password')

                                <span class="invalid-feedback" role="alert">

                                    <strong>{{ $message }}</strong>

                                </span>

                                @enderror

                            </div>

                        </div>

                        <div class="form-group row">

                            <label for="password-confirm"

                                class="col-md-4 col-form-label text-md-right">{{ \_\_('Confirm Password') }}</label>

                            <div class="col-md-6">

                                <input id="password-confirm" type="password" class="form-control"

                                    name="password\_confirmation" required />

 @error('password\_confirmation')

                                <span class="invalid-feedback" role="alert">

                                    <strong>{{ $message }}</strong>

                                </span>

                                @enderror

                            </div>

                        </div>

                        <div class="form-group row mb-0">

                            <div class="col-md-6 offset-md-4">

                                <button type="submit" class="btn btn-primary">

                                    {{ \_\_('Register') }}

                                </button>

                            </div>

                        </div>

                    </form>

                </div>

            </div>

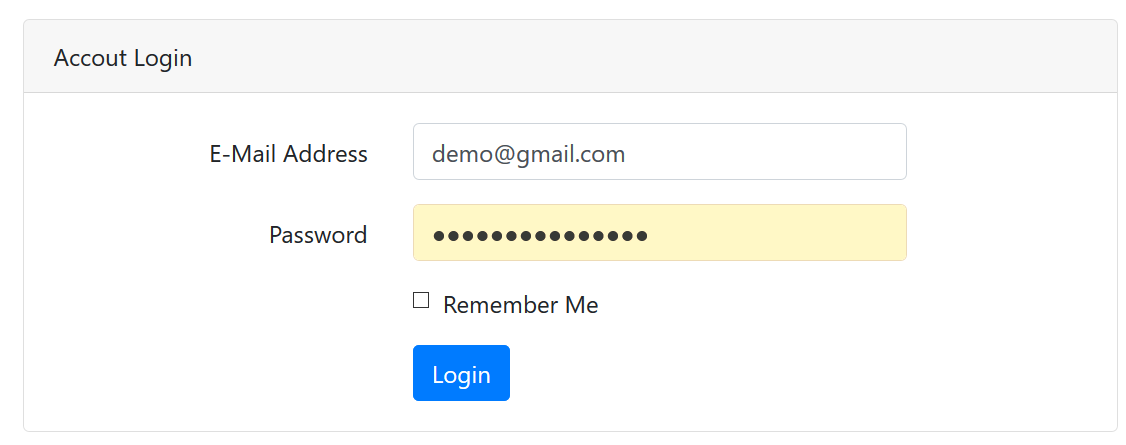
        </div>

    </div>

</div>

@endsection

Màn hình có dạng sau



**Bước 8: Thực hiện đăng ký cho account**

Khi người dùng submit form register thì sẽ thực hiện nhận request post\_register và lưu vào bảng accounts

Mở file **AccountControler** và tìm đến **post\_regiser** code như sau

public function post\_register(Request $request)

{

    // validae dữ liệu trên form

    $rules = [

        'name' => 'required|max:100',

        'email' => 'required|unique:account|max:100',

        'phone' => 'required|max:50',

        'address' => 'required|max:200',

        'password' => 'required|min:6|max:12',

        'password\_confirmation' => 'required|same:password',

    ];

    $message = [

        // 'name.required' => 'Vui lòng nhập họ tên'

    ];

    $request->validate($rules,$message);

    // Lưu thông in vào bảng accounts

    $add = Account::create([

        'name' => $request->name,

        'email' => $request->email,

        'phone' => $request->phone,

        'address' => $request->address,

        'gender' => $request->gender,

        'password' => bcrypt($request->password)

    ]);

    // kiểm tra thêm mới thành công hay không

    if($add){

        return redirect()->back()->with('error','Đăng ký không thành công vui lòng thử lại');

    }

    return redirect()->route('account.login');

}

Hãy truy cập <http://localhost/blog/account/register> nhập thông tin và test, và kiểm tra trong bảng acounts xem có dữ liệu hay chưa trước khi sang phàn đăng nhập

**Bước 12: Thực hiện đăng nhập và lưu session khi form login được submit**

Mở file **AccountControler** và tìm đến phương thức post\_login và soạn code như sau

public function post\_login(Request $request)

{

*//$login\_data = $request->only('email');*

    $login\_data = [

        'email' => $request->email

    ];

*// TRUY VẤN THÔNG IN TÀI KHOẢN VỚI EMAIL*

    $check\_login = Account::wehere($login\_data)->first();

*// NẾU TÒN TẠI EMAIL VÀ MẬT KHẨU TRÙNG KHỚP*

    if($check\_login && \Hash::check($request->password, $check\_login->password)){

*// TIẾN HÀNH LƯU THÔNG IN ACCOUNT VÀO SESION*

        session(['account\_login' => $check\_login]);

*// CHUYỂN HƯỚNG VỀ TRANG CHỦ*

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

    }

*// NẾU ĐỊA CHỈ EMAIL KHÔNG TỒN TẠI TRONG BẢNG SẼ BÁO LỖI*

    return redirect()->back()->with('error','Đăng nhập không thành công vui lòng thử lại');

}

Hãy nhập thông tin đăng nhập và test thử

<http://localhost/blog/customer/login>

**Bài 3:** **Tạo custum global function – các hàm global do người dung tạo**

framework Laravel cho phép người dung tạo ra các global function để có thể sử dụng toàn cụ thuận tiện hơn và ngắn gọn hơn

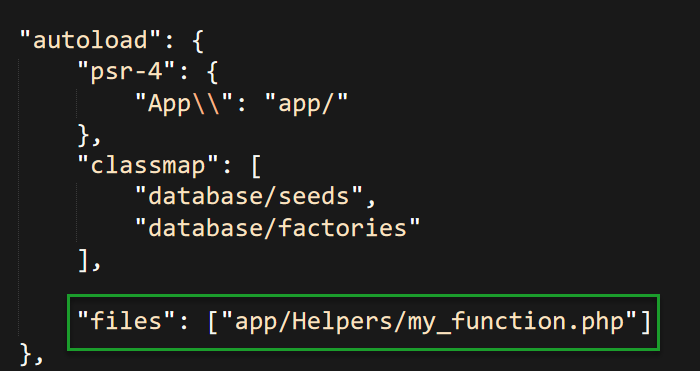
**Bước 1: T**ạo một thư mục mới trong thư mục app hoặc bất kỳ thư mục nào trong thư mục gốc của dự án

VD ở đây sẽ tạo thư mục **app/Helpers**

**Bước 2:** trong thư mục **Helpers** vửa tạo, tạo thêm file php bất kỳ

VD ở đây tạo thêm file **my\_functions.php**

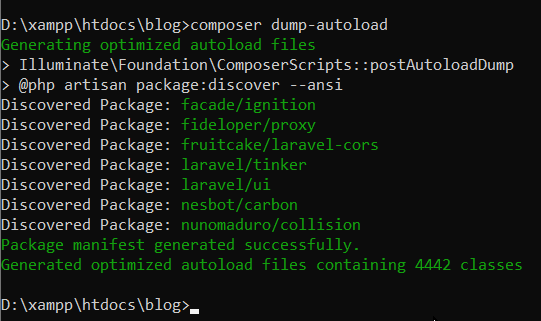
**Bước 3**: mở file **composer.json** tìm đến đoạn **autoload** và khai báo thêm như sau



**Bước 4:** mở cmd tại thư mục dự án và chạy lệnh sau

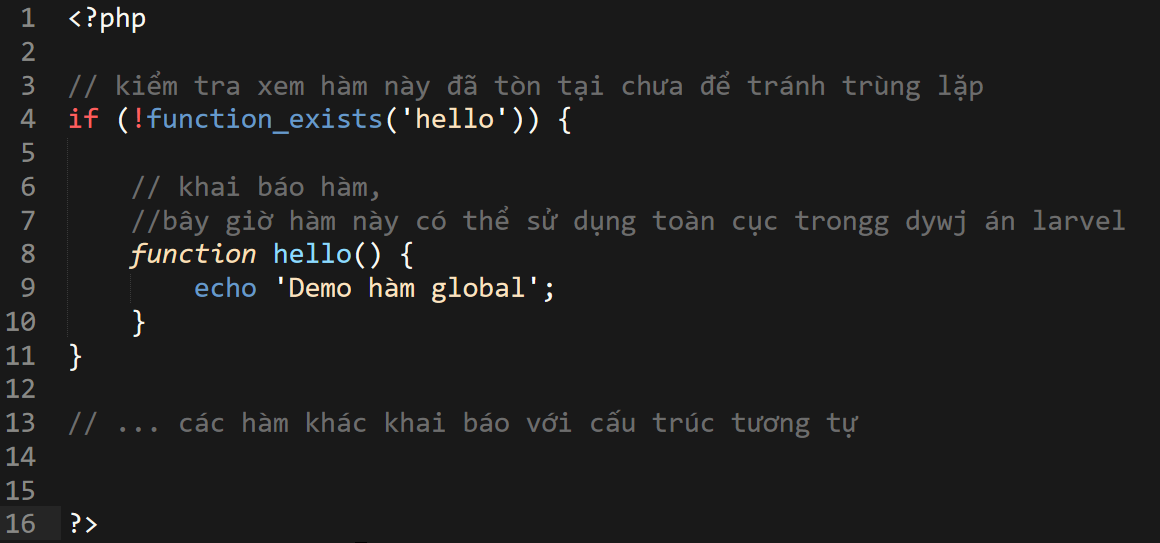
Composer dump-autoload

Lệnh này giúp trình quản lý composer cập nhật lại các thiết lập các file



**Bước 5:** Bây giờ bạn có thể khai báo bất ký hàm nào để có thể sử dụng toàn cục như các hàm đã tham khảo bài 1

Mở file **my\_functions.php** lên và khai báo các hàm có dạng sau



***Bài tập tự làm***

**Bài 1:** Hãy sử dụng hàm Str::slug để tạo đường dẫn tối ưu SEO cho các sản phẩm và tin tức hoặc danh mục

**Bài 2:** Hãy tạo một global function thực hiện công việc upload ảnh có thể tái sử dụng ở bất kỳ trường hợp nào , hàm chỉ rả về tên file hoặc mảng các tên file khi upload thành công

**Bài 3:** Hãy hiển thị thông tin đăng nhập của account rong session như bài tập tự làm số 2 lên giao diện

**Bài 4:** Tìm hiểu các hàm xử lý **session()** để tạo chức năng giỏ hang