**Bài thực hành 04**

**Kết nối CSDL, truy vấn dữ liệu**

***Mục tiêu***

* Cấu hình kết nối CSDL
* Truy vấn dữ liệu cơ bản với class DB
* Validate dữ liệu

***Thực hành theo các bài tập***

# Bài 1: Thực hành kết nối CSDL

**Bước 1:** Tạo CSDL demo\_lrv7 và bảng categories như sau

Vào phpmyadmin và chạy đoạn code sau

CREATE DATABASE demo\_db DEFAULT CHARACTER SET utf8 COLLATE utf8\_unicode\_ci;

use demo\_lrv7;

CREATE TABLE categories (

    id INT UNSIGNED PRIMARY KEY AUTO\_INCREMENT,

    name VARCHAR(100) NOT NULL UNIQUE,

    status tinyint DEFAULT '1',

    created\_at date DEFAULT NOW(),

    updated\_at date NULL

)

**Bước 2**: Cấu hình kết nối CSDL **demo\_db**

Mở file config/database.php, sau đó tìm cấu hình kết nối mysql như sau

'mysql' => [

    'driver' => 'mysql',

    'url' => env('DATABASE\_URL'),

    'host' => 'localhost',

    'port' => '3306'),

    'database' => 'demo\_db'),

    'username' => 'root',

    'password' => '',

    'unix\_socket' => '',

    'charset' => 'utf8',

    'collation' => 'utf8\_unicode\_ci',

    'prefix' => '',

    'prefix\_indexes' => true,

    'strict' => true,

    'engine' => null,

    'options' => extension\_loaded('pdo\_mysql') ? array\_filter([

        PDO::MYSQL\_ATTR\_SSL\_CA => env('MYSQL\_ATTR\_SSL\_CA'),

    ]) : [],

],

# Bài 2: Thục hiện chức năng Danh sách cho bảng cateogories

**Bước 1: Tạo CategoryController**

Trong thư mục **app\Http\Controllers** tạo file **CategoryController.php** và soạn code như sau (Tên file và tên class phải trùng nhau)

<?php

namespace App\Http\Controllers;

use DB;

use Illuminate\Http\Request;

class CategoryController extends Controller

{

    public function index(){

        $cats = DB::table('categories')->paginate(3);

        return view('category.index',compact('cats'));

    }

}

**Bước 2:** Tạo view và hiển thị dữ liệu có phân trang

Trong thư mục resources tạo thêm thư mục category

Trong thư mục category vừa tạo, thao file index.blade.php có nội dung tương tự sau

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

@section('main')

<h2>Danh sách danh mục</h2>

<table class="table table-bordered">

    <thead>

        <tr>

            <th>ID</th>

            <th>NAME</th>

            <th>STATUS</th>

        </tr>

    </thead>

    <tbody>

        <!-- cú pháp của blade view -->

        @foreach($cats as $cat)

        <tr>

            <td>{{$cat->id}}</td>

            <td>{{$cat->name}}</td>

            <td>{{$cat->status}}</td>

        </tr>

        @endforeach

    </tbody>

</table>

<!-- Hiển thị phân trang -->

{{$cats->links()}}

@stop()

**Bước 3** Tạo route để truy cập vào danh sách danh mục

Mở file web.php trong thư mục routes va khai báo route như sau

Route::get('/', function () {

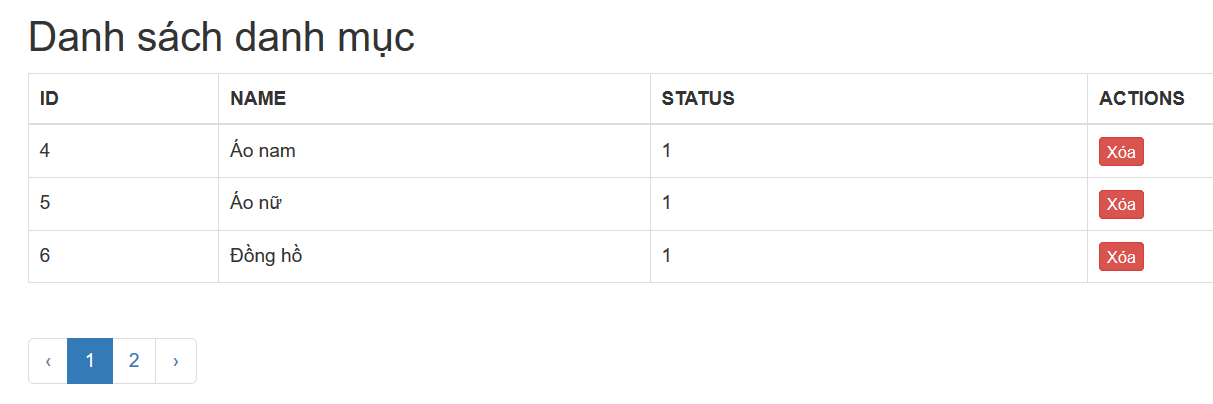
    return view('welcome');

});

Route::get('categorry','CategoryController@index')->name(‘category.index);

**Bước 6:** Truy cập vào link đã tạo theo đường dẫn

<http://localhost/lrvdemo/categorry>



Bạn có thể chèn vài dòn dữ liệu để xem kết quả

# Bài 3: Tạo form thêm mới có validate dữ liệu cho bảng categories

**Bước 1:** Tạo thêm các route để hiển thị form thêm mới

**GET** **categorry/create** để hiển thị form nhập liệu

**POST** **categorry** để nhận dữ liệu và lưu vào bảng category khi submit form

Mở file **routes/web.php** lên và them route như sau

Route::get('/', function () {

    return view('welcome');

});

// GET categorry để hiển thị danh sách categories

Route::get('categorry','CategoryController@index')->name('category.index');

// GET categorry/create để hiển thị form nhập dữ liệu

Route::get('categorry/create','CategoryController@create')->name('category.create');

// POST categorry để nhận dữ liệu khi submit form

Route::post('categorry','CategoryController@store')->name('category.store');

Bạn thấy route category/create sử dụng phương thức create của CategoryController

**Bước 2**: Mở file **CategoryController** thêm các phương thưc create và store với code có dạng sau

<?php

namespace App\Http\Controllers;

use DB;

use Illuminate\Http\Request;

class CategoryController extends Controller

{

    public function index(){

        $cats = DB::table('categories')->paginate(3);

        return view('category.index',compact('cats'));

    }

     /\*\*

     \* Phương thức create hiển thị form nhập dữ liệu

     \*/

    public function create(){

        return view('category.create');

    }

    /\*\*

     \* Phương thức store để nhận và lưu dữ liệu vào bảng

     \*/

    public function store(Request $request){

        // $request->only() lấy dữ liệu từ form giống với $\_POST

        //Category::create(); // như lệnh INSERT INTO category

        DB::table('categories')->insert($request->only('name','status'));

        return redirect()->route('category.index'); // chuyển hướng về danh sách

    }

}

**Bước 3:** tào file **create.blade.php** trong thư mục **resources/views/category** và code như sau

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

@section('main')

<h2>Thêm mới danh mục danh mục</h2>

<form action**="{{route('category.store')}}"** method="POST" role="form">

**@csrf**

    <div class="form-group">

        <label for="">Tên danh mục</label>

        <input class="form-control" name="name" placeholder="Ten danh mục">

    </div>

    <div class="form-group">

        <label for="">Trạng thái</label>

        <div class="radio">

            <label>

                <input type="radio" name="status" value="1" checked>

                Hiển thị

            </label>

            <label>

                <input type="radio" name="status" value="0">

                Ẩn

            </label>

        </div>

    </div>

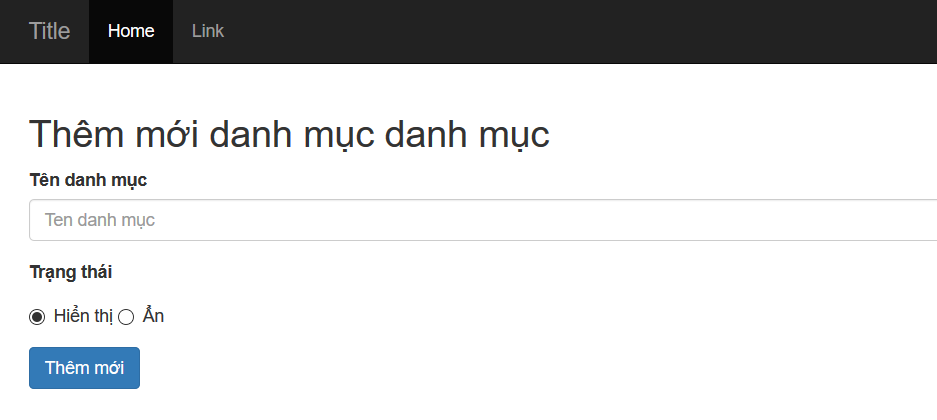
    <button type="submit" class="btn btn-primary">Thêm mới</button>

</form>

@stop()

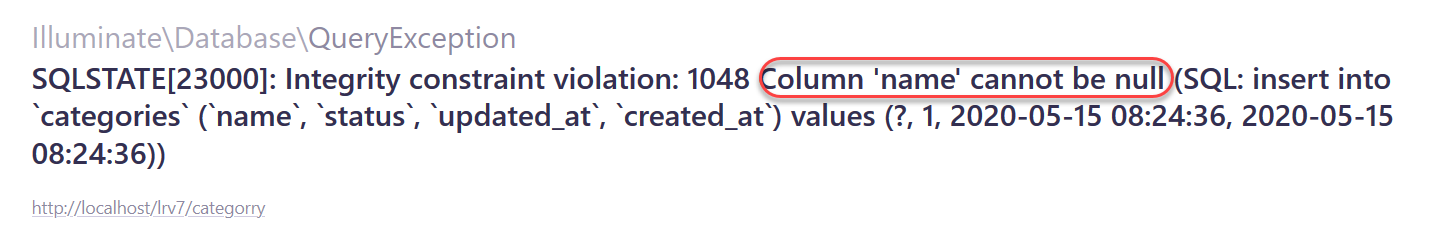
**Bước 4: truy cập đường dẫn sau để xem kết quả**

Link có dạng: <http://localhost/lrv7/categorry/create>



**Bước 5:** Nhập dữ liệu và nhấn nút thêm mớ để test thử

**Bước 6:** Không nhập dữ liệu và click vào nút thêm mới, sẽ có lỗi như sau



Do chưa validate form, nên người dung có thể không nhập mà click nút bấm

**Bước 7: Validate dữ liệu cho phù hợp với cấu trúc và ràng buộc của bảng categories**

Bảng categories có trường **name là not null unique** nên code validate có dạng sau

**Tại CategoryController@store**

public function store(Request $request){

    $rules = [

        'name' => 'required|unique:categories'

    ];

    $messages = [

        'name.required' => 'Tên danh mục không để trống',

        'name.unique' => 'Tên danh mục đã được sử dụng',

    ];

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

    DB::table('categories')->($request->only('name','status'));

    return redirect()->route('category.index'); // chuyển hướng về danh sách

}

**Trên form, hiển thị lỗi tại các input như sau**

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

@section('main')

<h2>Thêm mới danh mục danh mục</h2>

<form action="{{route('category.store')}}" method="POST" role="form">

    @csrf

    <div class="form-group">

        <label for="">Tên danh mục</label>

        <input class="form-control" name="name" placeholder="Ten danh mục">

        @error('name')

        <small class="help-block text-danger">{{$message}}</small>

        @enderror

    </div>

    <div class="form-group">

        <label for="">Trạng thái</label>

        <div class="radio">

            <label>

                <input type="radio" name="status" value="1" checked>

                Hiển thị

            </label>

            <label>

                <input type="radio" name="status" value="0">

                Ẩn

            </label>

        </div>

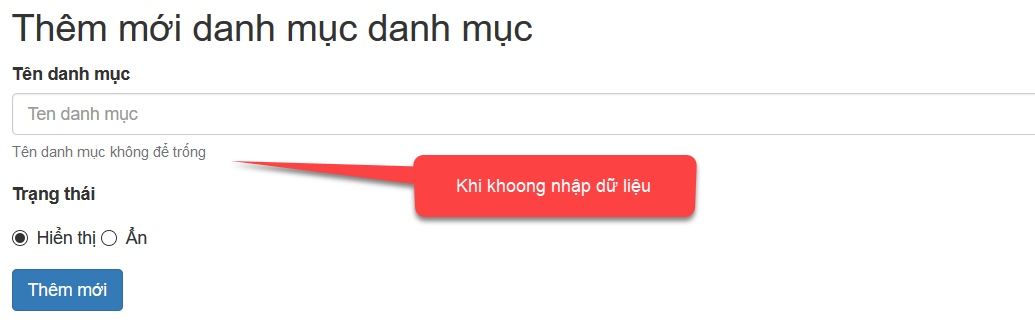
    </div>

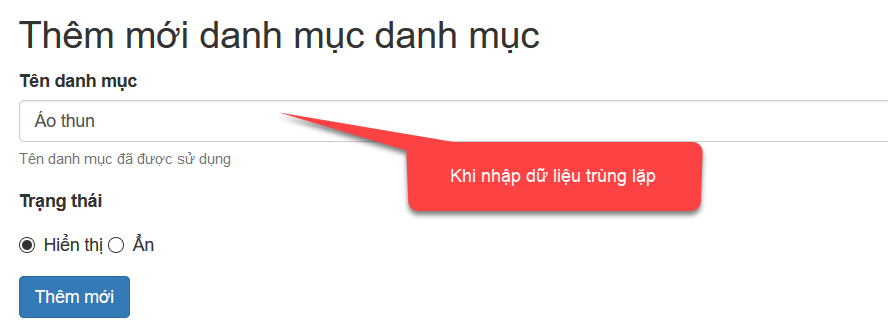
    <button type="submit" class="btn btn-primary">Thêm mới</button>

</form>

@stop()

**Test lại kết quả như sau**





# Bài 4: Tạo chức năng xóa dữ liệu theo id

**Bước 1:** Tạo route category/delete có ham số id như sau

//DELETE category/{id} phương thức DELETE, khác với GET, hay POST

Route::delete('categorry/{id}','CategoryController@delete')->name('category.delete');

**Bước 2: Tại danh sách danh mục tạo nút xóa**

Mở file **resources/views/category/index.blade.php** và tạo them cột và nú xóa như sau

**Chú ý:** do các phương thức DELETE, HEAD, PUT không hỗ trợ trực tiếp thông qua http request

Vì vậy Laravel xử lý bằng cách, sử dụng form post có thêm input ẩn có name là **\_method** có giá trị là DELEE. PUT, HEAD để xử lý phương thức DELEE. PUT, HEAD

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

@section('main')

<h2>Danh sách danh mục</h2>

<table class="table table-bordered">

    <thead>

        <tr>

            <th>ID</th>

            <th>NAME</th>

            <th>STATUS</th>

            <th>ACTIONS</th>

        </tr>

    </thead>

    <tbody>

        <!-- cú pháp của blade view -->

        @foreach($cats as $cat)

        <tr>

            <td>{{$cat->id}}</td>

            <td>{{$cat->name}}</td>

            <td>{{$cat->status}}</td>

            <td>

                <form action="{{route('category.delete',$cat->id)}}" method="POST">

                    @method('DELETE') @csrf

                    <button class="btn btn-xs btn-danger">Xóa</button>

                </form>

            </td>

        </tr>

        @endforeach

    </tbody>

</table>

<!-- Hiển thị phân trang -->

{{$cats->links()}}

@stop()

**Bước 3: Mở CateoryController và tạo phương thức delete để xử lý xóa như sau**

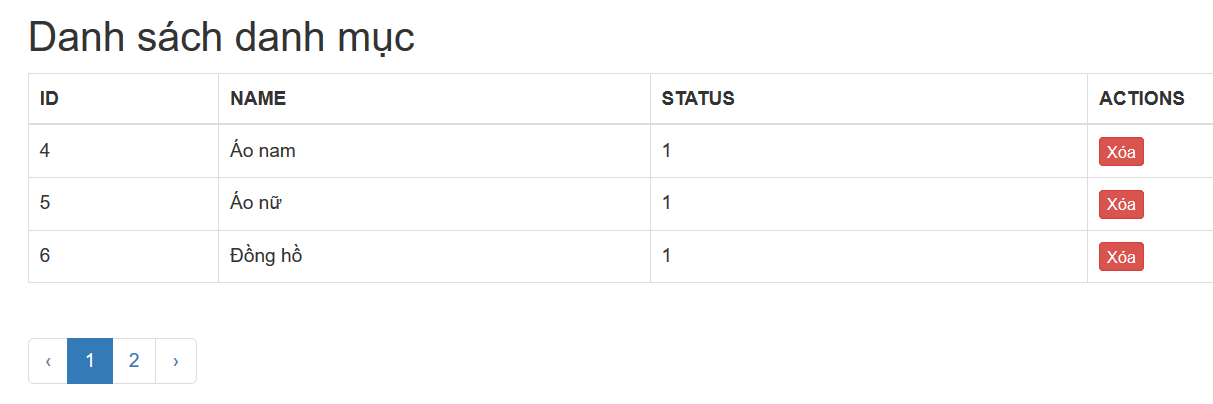
public function delete($id){

    DB::table('categories')->where('id',$id)->delete(); // return true, false

    return redirect()->route('category.index'); // chuyển hướng về danh sách

}

**Bước 4: quay lại trang danh sách danh mục và click nút xóa để test thử**

**Màn hình có dạng** 

# Bài 5: Tạo chức năng sửa danh mục

**Chức năng này cần 2 route là:**

**GET edit/{id} để hiển thị dữ liệu lên form khi click vào nút sửa tại danh sách**

**PUT update/{id} để nhận vào cà cập nhật dữ liệu**

**Bước 1: Tạo route GET và PUT như sau**

Mở file **routes/web.php** và soạn code sau

Route::get('categorry/{id}/edit','CategoryController@edit')->name('category.edit');

Route::put('categorry/{id}','CategoryController@update')->name('category.update');

**Bước 2: Tạo link click sang màn hình sửa**

Mở lại file danh sách danh mục **resources/views/category/index.blade.php** tạo thêm nút sửa ngay cạnh nút xóa như sau

<!-- cú pháp của blade view -->

@foreach($cats as $cat)

<tr>

    <td>{{$cat->id}}</td>

    <td>{{$cat->name}}</td>

    <td>{{$cat->status}}</td>

    <td>

        <form action="{{route('category.delete',$cat->id)}}" method="POST">

            @method('DELETE') @csrf

            <button class="btn btn-xs btn-danger">Xóa</button>

            <a href="{{route('category.edit',$cat->id)}}" class="btn btn-xs btn-primary">Sửa</a>

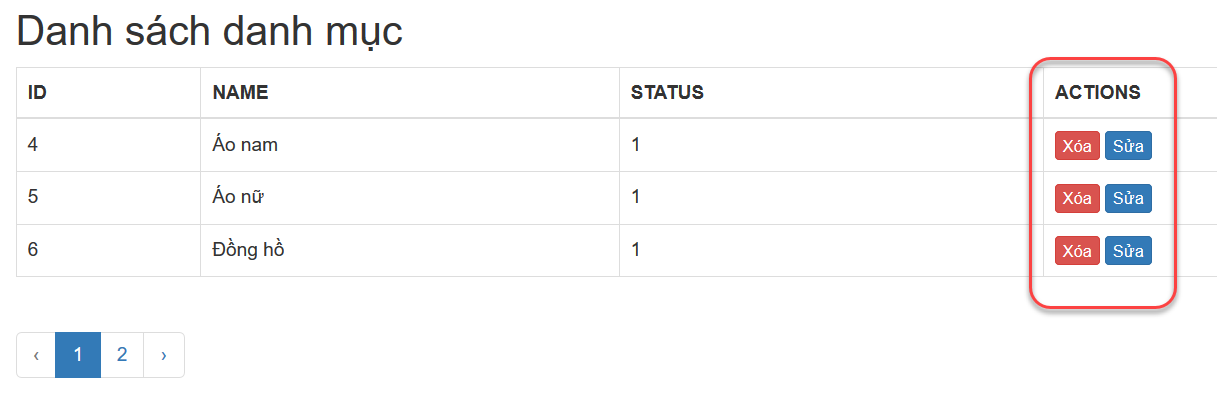
        </form>

    </td>

</tr>

@endforeach

Màn hình danh sách bây giờ có dạng



**Bước 3:** Mở lại file CategoryController them 2 phương thưc tương ứng với 2 route như sau

/\*\* phương thức edit hiển thị dữ liệu trên form theo id \*/

public function edit($id){

    /\*\* SELECT \* FROM categories WHERE id = $id \*/

    $cat = DB::table('categories')->find($id);

    /\*\* Gửi dữ liệu qua form edit.blade.php\*/

    return view('category.edit',compact('cat'));

}

/\*\* Phương thức update để nhận và lưu dữ liệu vào bảng \*/

public function update($id,Request $request){

    $rules = [

        'name' => 'required|unique:categories'

    ];

    $messages = [

        'name.required' => 'Tên danh mục không để trống',

        'name.unique' => 'Tên danh mục đã được sử dụng',

    ];

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

    DB::table('categories')->where('id',$id)->update($request->only('name','status'));

    return redirect()->route('category.index'); // chuyển hướng về danh sách

}

**Bước 4: Tạo from sửa dữ liệu**

Tạo file **resources/views/category/edit.blade.php** và soạn code sau (Form giống form them mới)

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

@section('main')

<h2>Chỉnh sửa danh mục danh mục: {{$cat->name}}</h2>

<form action="{{route('category.update',$cat->id)}}" method="POST" role="form">

    @csrf @method('PUT')

    <div class="form-group">

        <label for="">Tên danh mục</label>

        <input class="form-control" name="name" value="{{$cat->name}}">

        @error('name')

        <small class="help-block text-danger">{{$message}}</small>

        @enderror

    </div>

    <div class="form-group">

        <label for="">Trạng thái</label>

        <div class="radio">

            <label>

                <input type="radio" name="status" value="1" checked>

                Hiển thị

            </label>

            <label>

                <input type="radio" name="status" value="0">

                Ẩn

            </label>

        </div>

    </div>

    <button type="submit" class="btn btn-primary">Thêm mới</button>

</form>

@stop()

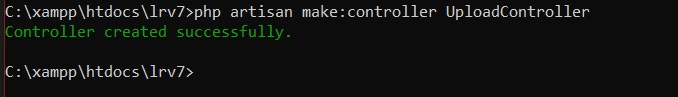
**Bước 5: Test kết quả, khi click vào nút sử, màn hình form có dạng**



# Bài 6: Tạo form upload ảnh cơ bản

**Bước 1:** Tạo UploadController bằng cách mở cmd tại thư mục dự án và chạy lệnh

php artisan make:controller UploadController



**Bước 2:** Mở file **routes/web.php** và tạo 2 route đại diện cho 2 phương thức

**GET upload** để hiển thị form cho phép chọn ảnh đẻ upload

**POST upload** nhận dữ liệu từ form và upload ảnh vào thư mục public/uploads

// GET upload

Route::get('upload','UploadController@form')->name('upload.form');

// POST upload

Route::post('upload','UploadController@upload')->name('upload.upload');

**Bước 3:** Mở file app\Http\Contronllers\UploadController và tạo 2 hàm đại diện cho 2 phương thức

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class UploadController extends Controller

{

    /\*\*Hienr thị form chọn ản \*/

    public function form(){

        return view('upload.form');

    }

    /\*\* Nhận dữ liệu và thực hiện upload \*/

    public function upload(Request $request){

        $rules = ['upload' => 'required:mimes:jpeg,png,gif'];

        $mesages = [

            'upload.required' => 'Vui lòng chọn một file',

            'upload.mimes' => 'Định dạng file cho phép là: jpg, png, gif',

        ];

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

        $file\_name = $request->upload->getClientOriginalName();

        $request->upload->move(public\_path('uploads'),$file\_name);

        // quay lại form upload và gửi kèm một flash message

        return redirect()->back()->with('success',"Upload ảnh thành công");

    }

}

?>

Bước 4: Tạo file **resources/views/upload/form.blade.php** và soạn code tạo form

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

@section('main')

<h2>Form Upload</h2>

<!-- Hiển thị flash message khi upload thành công -->

@if(Session::has('success'))

<div class="alert alert-success">

    <button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>

    {{Session::get('success')}}

</div>

@endif

<form action="" method="POST" enctype="multipart/form-data">

    @csrf

    <div class="form-group">

        <label for="">Chọn ảnh</label>

        <input type="file" class="form-control" name="upload" placeholder="Ten danh mục">

        @error('upload')

        <small class="help-block text-danger">{{$message}}</small>

        @enderror

    </div>

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

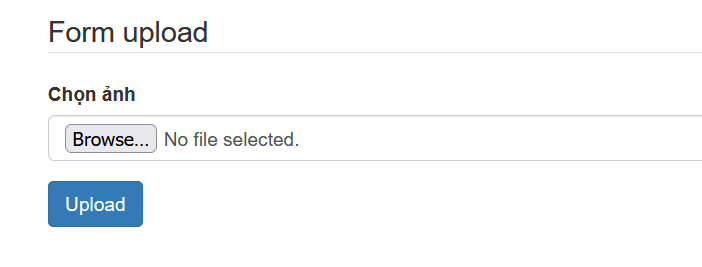
</form>

@stop()

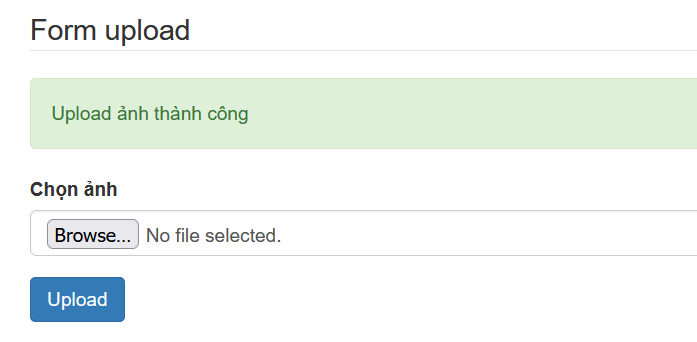
**Bước 3: Truy cập link và test kết quả**

Link có dạng

<http://localhost/lrv7/upload>



Sau khi chọn file và upload thành công màn hình có dạng



Kiểm tra file upload trong thư mục **public/uploads**

* 1. ***Bài tập tự làm***

Kế thừa bài hướng dẫn trên, tiếp tục làm các yêu cầu sau

Cho bảng products như sau

| **Tên cột** | **Kiểu dữ liệu** | **Độ dài** | **Rang buộc** |
| --- | --- | --- | --- |
| id | Int |  | Khóa chính tự động tang |
| name | varchar | 150 | Không rỗng |
| price | float | 9,3 | Khong rỗng |
| category\_id | Int |  | Khong rỗng khóa ngoại tới bảng **categories** |
| image | varchar | 150 | Không rỗng |
| created\_at | Date |  |  |
| updated\_at |  |  |  |

**Yêu cầu:** Tạo chức năng **CRUD** cho bảng products trong đó chú ý

Danh sách sản phẩm có hiển thị thông tin gồ: id, name, price, image

Danh sách có phân trang

Màn hình thêm mới và chỉnh sửa products cho phép chọn danh mục tại select box

Các chức năng thêm mới và sửa cần có upload ảnh

**Validate:** Dữ liệu cột cần sử dụng laravel để validate như sau

name: bắt buộc nhập, không quá 150 ký tự

category\_id: bắt buộc chọn

miage bắt buộc chọn và định dạng cho phép là: jpg, png, gif

price: Bắt buộc nhập và chỉ cho nhập số