QUY TRÌNH LÀM DỰ ÁN LARAVEL NHIỀU BẢNG:

Bước1: Tạo dự án:

composer create-project laravel/laravel <Project Name>

Bước 2: Config tên database trong tệp .Env

Bước 3: Đọc đề, kiểm tra xem Bảng nào liên kết với bảng nào, yêu tiên tạo bảng ko có thuộc tính của bảng khác ở trong.

VD: Có 3 bảng : Tác giả, Thể loại, Bài viết. Trong đó bảng Bài viết lấy khóa chính của 2 bảng còn lại làm khóa ngoại. Vì thế ta tiến thành tạo -MCR của 2 bảng Tác giả và thể loại trước rồi mới thực hiện tạo của bảng Bài viết.

Lệnh tạo tệp migration, model, controller cùng 1 lúc:  
 php artisan make:model + model\_name -mcr

VD: php artisan make:model Author  
 php artisan make:model Category

Bước 4: Tiến thành tạo bảng và điền các thuộc tính của bảng trong thư mục migrations:

VD: thiết lập các thuộc tính cho bảng Author:

public function up(): void

    {

        Schema::create('authors', function (Blueprint $table) {

            $table->integer('author\_id', true, true);

            $table->string('author\_name');

            $table->string('author\_img');

        });

    }

* Lưu ý: tại thuộc tính ID, phải dùng integer(‘id’, true, true) : để khớp kiểu dữ liệu và ko bị lỗi

Bước 5: Đối với bảng có ràng buộc khóa chính khóa ngoại:

public function up(): void

    {

        Schema::create('posts', function (Blueprint $table) {

            $table->integer('post\_id', true, true);

            $table->unsignedInteger('author\_id');

            $table->unsignedInteger('category\_id');

            $table->foreign('author\_id')->references('author\_id')->on('authors')->onDelete('cascade');

            $table->foreign('category\_id')->references('category\_id')->on('categories')->onDelete('cascade');

        });

    }

* Lưu ý: khi khai báo 2 thuộc tính khóa ngoại phải có kiểu dữ liệu là ‘unsigneInteger’

Bước 6: Tiến hành thực hiện làm model.

Làm model cho các thành phần như bình thường, bao gồm:  
 protected $fillable = [‘tên các thuộc tính của bảng’, ..];

public $primaryKey = ‘tên khóa chính’

public $timestamps = false;

Bước 7: Đọc kĩ đề bài:

Nếu đề bài yêu cầu thêm xóa sửa ở bảng có chức khóa ngoại thì trong Model của nó phải khai báo thêm ràng buộc quan hệ ( một – một, một – nhiều)

VD: trong đề bài yêu cầu thêm xóa sửa tại bài viết, thì trong Model Post, cần khai báo ràng buộc quan hệ như sau:

public function author()

    {

        return $this->belongsTo(Author::class, 'author\_id');

    }

    public function category()

    {

        return $this->belongsTo(Category::class, 'category\_id');

    }

* Lưu ý: belongsTo sẽ dùng khi liên kết của 2 bảng là liên kết 1 – nhiều
* Sử dụng belongsTo ở bên trong Model của bảng có chứa khóa ngoại.
* Nếu liên kết là 1-1 thì thay belongsTo = hasOne.
* Ta sẽ chỉ sử dụng hasMany khi bảng hiện tại không chứa khóa chính của bảng được liên kết cùng ( tức là bên 1, bên kia là N)

VD: thêm xóa sửa Author:

Thì Model của Author sẽ như sau:

public function post()

    {

        return $this->hasMany(Post::class, 'author\_id');

    }

* Giải thích : post(), author(), category() ở trên là các hàm để lấy dữ liệu của bài viết thông qua bảng được thêm sửa xóa. ( trong hàm index sẽ rõ)

Sau khi hoàn tất thực hiện tạo bảng với lệnh:

php artisan migrate

Bước 8: Thực hiện fake dữ liệu với seeder:

Lệnh tạo seeder:

php artisan make:seeder + “tên bảng seeder”

Trong tệp seeder vừa tạo:

Khai báo thư viện faker:

use Faker\Factory as Faker;

Nhớ phải khai báo Model: use App\Models\ Tên Model

VD: về thực hiện faker dữ liệu:

public function run(): void

    {

        $faker = Faker::create();

        for($i=0; $i <5; $i++){

            Author::create([

                'author\_id' => $i+1,

                'author\_name' => $faker->name,

                'author\_email' => $faker->email,

            ]);

        }

    }

* $faker->word(3,true) : sử dụng khi cần faker 1 chữ có 3 kí tự
* $faker->sentence(3,true): sử dụng khi cần faker 1 câu có 3 chữ
* …

Lệnh chạy seeder:

php artisan db:seed –class=<tên bảng seeder>

Bước 9: Kiểm tra trên csdl

Bước 10: thực hiện điền routes cho dự án:

// POST ROUTES

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

Route::get('/post/create', [PostController::class,'create']);

Route::post('/post/create', [PostController::class,'store']);

Route::get('/post/edit/{post\_id}', [PostController::class,'edit']);

Route::post('/post/edit/{post\_id}', [PostController::class,'update']);

Route::get('/post/detail/{post\_id}', [PostController::class,'show']);

Route::get('/post/delete/{post\_id}', [PostController::class, 'destroy']);

Trên đây là ví dụ tạo routes cho bảng post, dự vào đây để sửa đổi sang cho phù hợp

Bước 11: thực hiện tạo trang home ( có thể đổi tên welcome thành home)

Trang home.blade.php:

<!doctype html>

<html lang="en">

<head>

    <meta charset="utf-8">

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

    <title>Home Page</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet"

        integrity="sha384-9ndCyUaIbzAi2FUVXJi0CjmCapSmO7SnpJef0486qhLnuZ2cdeRhO02iuK6FUUVM" crossorigin="anonymous">

    <!-- Option 1: Include in HTML -->

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.3.0/font/bootstrap-icons.css">

</head>

<body>

    <header class="header">

        <nav class="navbar bg-success">

            <div class="container-fluid">

                <a class="navbar-brand text-white">Navbar</a>

                <form class="d-flex" role="search">

                    <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">

                    <button class="btn  text-white" type="submit">Search</button>

                </form>

            </div>

        </nav>

    </header>

    <div class="container">

        <div class="container-head d-flex justify-content-between align-items-center p-3">

            <h1>List of departments</h1>

            <a href="/department/create"><button class="btn btn-success">Add new</button></a>

        </div>

        <table class="table table-striped">

            <thead>

                <tr>

                    <th scope="col">ID</th>

                    <th scope="col">Name</th>

                    <th scope="col">Location</th>

                    <th scope="col">Action</th>

                </tr>

            </thead>

            <tbody>

                @foreach($departments as $department)

                <tr>

                    <th scope="row">{{$department->department\_id}}</th>

                    <td>{{$department->department\_name}}</td>

                    <td>{{$department->locations->location\_name}}</td>

                    <td>

                        <a class="text-decoration-none m-2" href="/department/show/{{$department->department\_id}}">

                            <i class="bi bi-eye"></i>

                        </a>

                        <a class="text-decoration-none m-2" href="/department/edit/{{$department->department\_id}}">

                            <i class="bi bi-pencil"></i>

                        </a>

                        <a class="text-decoration-none m-2 link"

                            href="/department/delete/{{$department->department\_id}}" id="deleteDepartment"><i

                                class="bi bi-trash"></i></a>

                    </td>

                </tr>

                @endforeach

            </tbody>

        </table>

    </div>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"

        integrity="sha384-geWF76RCwLtnZ8qwWowPQNguL3RmwHVBC9FhGdlKrxdiJJigb/j/68SIy3Te4Bkz" crossorigin="anonymous">

    </script>

    <script>

    document.addEventListener('DOMContentLoaded', function() {

        const deleteLinks = document.querySelectorAll('#deleteDepartment');

        deleteLinks.forEach(function(link) {

            link.addEventListener('click', function(event) {

                event.preventDefault();

                const confirmation = confirm(

                    'Are you sure you want to delete this department?');

                if (confirmation) {

                    window.location.href = this.href;

                }

            });

        });

    });

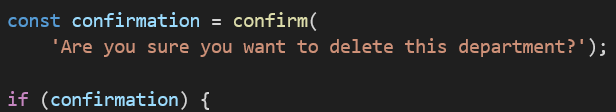
    </script>

</body>

</html>

- Nếu sử dụng trang này lưu ý chỉnh sửa tên biến sao cho phù hợp  
- Trường ID ở thẻ <a> phải trùng vơi ID trong thẻ <script> ở dưới cùng:

 và 

Đổi tên đâu confirm cho phù hợp: 

Trang create.blade.php:

<!doctype html>

<html lang="en">

<head>

    <meta charset="utf-8">

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

    <title>Add Department</title>

    <link rel="stylesheet" href="{{ asset('style.css')}}">

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.3.0/font/bootstrap-icons.css">

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet"

        integrity="sha384-9ndCyUaIbzAi2FUVXJi0CjmCapSmO7SnpJef0486qhLnuZ2cdeRhO02iuK6FUUVM" crossorigin="anonymous">

</head>

<body>

    <div class="app">

        <header class="header">

            <nav class="navbar bg-success">

                <div class="container-fluid">

                    <a class="navbar-brand text-white">Navbar</a>

                    <form class="d-flex" role="search">

                        <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">

                        <button class="btn  text-white" type="submit">Search</button>

                    </form>

                </div>

            </nav>

        </header>

        <div class="container">

            <div class="container-head d-flex p-3 justify-content-between align-items-center">

                <h1>Add a department</h1>

                <a href="/" class="text-white text-decoration-none"><button class="btn btn-warning">Back</button>

                </a>

            </div>

            <form class="form-create p-3 border" action="/department/create" method="POST">

                @csrf

                <div class=" mb-3">

                    <label for="location\_id" class="form-label combo-box"><strong>Location:</strong></label>

                    <select name="location\_id" id="location\_id">

                        @foreach($locations as $location)

                        <option value="{{$location->location\_id}}">{{$location->location\_name}}</option>

                        @endforeach

                    </select>

                    @error('location\_id')

                    <span class="text-danger">{{ $message }}</span>

                    @enderror

                </div>

                <div class="mb-3">

                    <label for="project\_id" class="form-label combo-box"><strong>Project:</strong></label>

                    <select name="project\_id" id="project\_id">

                        @foreach($projects as $project)

                        <option value="{{$project->project\_id}}">{{$project->project\_name}}</option>

                        @endforeach

                    </select>

                    @error('project\_id')

                    <span class="text-danger">{{ $message }}</span>

                    @enderror

                </div>

                <div class="mb-3">

                    <label for="department\_name" class="form-label"><strong>Department name</strong></label>

                    <input type="text" class="form-control" id="department\_name" name="department\_name">

                    @error('department\_name')

                    <span class="text-danger">{{ $message }}</span>

                    @enderror

                </div>

                <button type="submit" class="btn btn-success">Submit</button>

            </form>

        </div>

    </div>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"

        integrity="sha384-geWF76RCwLtnZ8qwWowPQNguL3RmwHVBC9FhGdlKrxdiJJigb/j/68SIy3Te4Bkz" crossorigin="anonymous">

    </script>

</body>

</html>

Trang edit.blade.php:

<!doctype html>

<html lang="en">

<head>

    <meta charset="utf-8">

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

    <title>Add Department</title>

    <link rel="stylesheet" href="{{ asset('style.css')}}">

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.3.0/font/bootstrap-icons.css">

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet"

        integrity="sha384-9ndCyUaIbzAi2FUVXJi0CjmCapSmO7SnpJef0486qhLnuZ2cdeRhO02iuK6FUUVM" crossorigin="anonymous">

</head>

<body>

    <div class="app">

        <header class="header">

            <nav class="navbar bg-success">

                <div class="container-fluid">

                    <a class="navbar-brand text-white">Navbar</a>

                    <form class="d-flex" role="search">

                        <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">

                        <button class="btn  text-white" type="submit">Search</button>

                    </form>

                </div>

            </nav>

        </header>

        <div class="container">

            <div class="container-head d-flex p-3 justify-content-between align-items-center">

                <h1>Edit a department</h1>

                <a href="/" class="text-white text-decoration-none"><button class="btn btn-warning">Back</button>

                </a>

            </div>

            <form class="form-create p-3 border" action="/department/edit/{{$department->department\_id}}" method="POST">

                @csrf

                <div class=" mb-3">

                    <label for="location\_id" class="form-label combo-box"><strong>Location:</strong></label>

                    <select name="location\_id" id="location\_id">

                        @foreach($locations as $location)

                        <option value="{{$location->location\_id}}">{{$location->location\_name}}</option>

                        @endforeach

                    </select>

                    @error('location\_id')

                    <span class="text-danger">{{ $message }}</span>

                    @enderror

                </div>

                <div class="mb-3">

                    <label for="project\_id" class="form-label combo-box"><strong>Project:</strong></label>

                    <select name="project\_id" id="project\_id">

                        @foreach($projects as $project)

                        <option value="{{$project->project\_id}}">{{$project->project\_name}}</option>

                        @endforeach

                    </select>

                    @error('project\_id')

                    <span class="text-danger">{{ $message }}</span>

                    @enderror

                </div>

                <div class="mb-3">

                    <label for="department\_name" class="form-label"><strong>Department name</strong></label>

                    <input type="text" class="form-control" id="department\_name" name="department\_name"

                        value="{{$department->department\_name}}">

                    @error('department\_name')

                    <span class="text-danger">{{ $message }}</span>

                    @enderror

                </div>

                <button type="submit" class="btn btn-success">Submit</button>

            </form>

        </div>

    </div>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"

        integrity="sha384-geWF76RCwLtnZ8qwWowPQNguL3RmwHVBC9FhGdlKrxdiJJigb/j/68SIy3Te4Bkz" crossorigin="anonymous">

    </script>

</body>

</html>

Trang show.blade.php:

<!doctype html>

<html lang="en">

<head>

    <meta charset="utf-8">

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

    <title>Detail Page</title>

    <link rel="stylesheet" href="{{ asset('style.css')}}">

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.3.0/font/bootstrap-icons.css">

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet"

        integrity="sha384-9ndCyUaIbzAi2FUVXJi0CjmCapSmO7SnpJef0486qhLnuZ2cdeRhO02iuK6FUUVM" crossorigin="anonymous">

</head>

<body>

    <div class="app">

        <header class="header">

            <nav class="navbar bg-success">

                <div class="container-fluid">

                    <a class="navbar-brand text-white">Navbar</a>

                    <form class="d-flex" role="search">

                        <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">

                        <button class="btn  text-white" type="submit">Search</button>

                    </form>

                </div>

            </nav>

        </header>

        <div class="container">

            <div class="container-head d-flex p-3 justify-content-between align-items-center">

                <h1>Detail of department</h1>

                <a href="/" class="text-white text-decoration-none"><button class="btn btn-warning">Back</button>

                </a>

            </div>

            <div class="container-content border">

                <ul class="post-detail">

                    <li><strong>Name:</strong>

                        {{ $department->department\_name }}

                    </li>

                    <li><strong>Location:</strong>

                        {{ $department->locations->location\_name }}</li>

                    <li><strong>Project:</strong>

                        {{ $department->projects->project\_name }}</li>

                </ul>

            </div>

        </div>

    </div>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"

        integrity="sha384-geWF76RCwLtnZ8qwWowPQNguL3RmwHVBC9FhGdlKrxdiJJigb/j/68SIy3Te4Bkz" crossorigin="anonymous">

    </script>

</body>

</html>

Bước 12: Thực hiện xử lý các hàm trong Controller:

File Controller mẫu: giải thích các hàm ở dưới sau:

<?php

namespace App\Http\Controllers;

use App\Models\Department;

use Illuminate\Http\Request;

use App\Models\Location;

use App\Models\Project;

use Illuminate\Support\Facades\Validator;

class DepartmentController extends Controller

{

    /\*\*

     \* Display a listing of the resource.

     \*/

    public function index()

    {

        $departments = Department::with('locations')->get();

        return view('home', compact('departments'));

    }

    /\*\*

     \* Show the form for creating a new resource.

     \*/

    public function create()

    {

        $locations = Location::all();

        $projects = Project::all();

        return view('department.create', compact('locations', 'projects'));

    }

    /\*\*

     \* Store a newly created resource in storage.

     \*/

    public function store(Request $request)

    {

        $validator = Validator::make($request->all(), [

            'location\_id' => 'required',

            'project\_id' => 'required',

            'department\_name' => 'required|unique:departments',

        ]);

        if ($validator->fails()) {

            return redirect('/department/create')->withErrors($validator)->withInput();

        }

        $input = $request->all();

        Department::create($input);

        return redirect('/');

    }

    /\*\*

     \* Display the specified resource.

     \*/

    public function show(Department $department, $department\_id)

    {

        $department = Department::findOrFail($department\_id);

        return view('department.show', compact('department'));

    }

    /\*\*

     \* Show the form for editing the specified resource.

     \*/

    public function edit(Department $department , $department\_id)

    {

        $locations = Location::all();

        $projects = Project::all();

        $department = Department::findOrFail($department\_id);

        return view('department.edit', compact('locations', 'projects', 'department'));

    }

    /\*\*

     \* Update the specified resource in storage.

     \*/

    public function update(Request $request, Department $department, $department\_id)

    {

        $department = Department::findOrFail($department\_id);

        $validator = Validator::make($request->all(), [

            'location\_id' => 'required',

            'project\_id' => 'required',

            'department\_name' => 'required|unique:departments',

        ]);

        if ($validator->fails()) {

            return redirect('/department/edit'. $department\_id)->withErrors($validator)->withInput();

        }

        $input = $request->all();

        $department->update($input);

        return redirect('/');

    }

    /\*\*

     \* Remove the specified resource from storage.

     \*/

    public function destroy(Department $department, $department\_id)

    {

        $department = Department::findOrFail($department\_id);

        $department->delete();

        return redirect('/');

    }

}

1. Hàm index():

public function index()

    {

        $departments = Department::with('locations')->get();

        return view('home', compact('departments'));

    }

Khai báo 1 biến để lưu toàn bộ dữ liệu,

ở đây dùng ::with(‘locations’) là để có thể lấy được location\_name thông qua hàm locations đã khai báo trong model của department ( giống như with(‘author’) là lấy author\_name thông qua hàm author khai báo trong post)

return ra view home kèm theo biến departments.

2. Hàm create():

public function create()

    {

        $locations = Location::all();

        $projects = Project::all();

        return view('department.create', compact('locations', 'projects'));

    }

Để có thể hiển thị select box với cái option thì ta phải lưu các giá trị của 2 bảng được liên kết với bảng hiện tại đang được thêm xóa sửa.

Ở đây đang lưu locations và projects vào 2 biến qua phương thức all()

Lưu ý import 2 model Location và Project ( thay đổi tên này sao cho phù hợp đề bài của mình)

Return ra view kèm theo 2 biến đó, vì trong create.blade.php có sử dụng vòng foreach để load lên các option để lựa chọn

3. Hàm store():

public function store(Request $request)

    {

        $validator = Validator::make($request->all(), [

            'location\_id' => 'required',

            'project\_id' => 'required',

            'department\_name' => 'required|unique:departments',

        ]);

        if ($validator->fails()) {

            return redirect('/department/create')->withErrors($validator)->withInput();

        }

        $input = $request->all();

        Department::create($input);

        return redirect('/');

    }

* Lưu ý phải khai báo thư viện Validator:
  + Lệnh như sau:

use Illuminate\Support\Facades\Validator;

* Ràng buộc ‘required’ là không được để trống
* ‘unique: departments’ là department\_name ko được giống nhau ( nếu thêm mới 1 department mới có tên giống department cũ thì sẽ báo lỗi tên đã tồn tại)

Cú pháp: ‘unique: tên bảng’

Nếu $validator fail thì sẽ hiển thị lại trang thêm mới: redirect(‘/department/create’)

Còn không có lỗi sẽ tạo mới và hiển thị về trang chủ:

4. Hàm show():

public function show(Department $department, $department\_id)

    {

        $department = Department::findOrFail($department\_id);

        return view('department.show', compact('department'));

    }

* Nhớ phải truyền vào $id ở tham sô đầu vao của hàm show
* Hàm này thực hiện tìm phần tử có $id được truyền vào và lưu vào biến $department sau đó return ra view show kèm theo biến đó

5. Hàm edit():

public function edit(Department $department , $department\_id)

    {

        $locations = Location::all();

        $projects = Project::all();

        $department = Department::findOrFail($department\_id);

        return view('department.edit', compact('locations', 'projects', 'department'));

    }

* Nhớ truyền vào $id ở tham số đầu vào của hàm edit
* Việc tạo 2 biến locations và projects cũng giống như ở hàm create() là để hiển thị được selected box với các option thông qua việc duyệt vòng foreach
* Sau đó tìm theo $id và return ra view edit kèm theo 3 biến trên

6. Hàm update():

public function update(Request $request, Department $department, $department\_id)

    {

        $department = Department::findOrFail($department\_id);

        $validator = Validator::make($request->all(), [

            'location\_id' => 'required',

            'project\_id' => 'required',

            'department\_name' => 'required|unique:departments',

        ]);

        if ($validator->fails()) {

            return redirect('/department/edit'. $department\_id)->withErrors($validator)->withInput();

        }

        $input = $request->all();

        $department->update($input);

        return redirect('/');

    }

* Nhớ truyền vào $id ở tham số đầu vào của hàm update()

7. Hàm destroy() – xóa:

public function destroy(Department $department, $department\_id)

    {

        $department = Department::findOrFail($department\_id);

        $department->delete();

        return redirect('/');

    }

* Nhớ truyền vào $id ở tham số đầu vào của hàm destroy()

Như vậy là đã hoàn thành, lưu ý thực hiện từng chức năng và kiểm tra ngay kết quả, lưu ý ở các trang edit, home, show, create các trường name, id, for phải giống nhau tránh nhầm lẫn và phải tìm lỗi sau này

LÀM ĐẾN ĐÂU CHẮC CHẮN ĐẾN ĐÓ.

GOOD LUCK!