

# Xây dựng chức năng CRUD cho dự án

Công cụ sử dụng:

- Máy chủ local: Apache hoặc tương đương
- Hệ quản trị cơ sở dữ liệu: MySQL hoặc MySQL Workbench hoặc tương đương
- Công cụ soạn thảo mã nguồn: Visual Studio Code hoặc tương đương.

Các bảng dữ liệu cơ bản cần thiết cho dự án thương mại điện tử bao gồm:

1. `categories`:

- Chức năng: Lưu trữ danh mục của sản phẩm. Mỗi danh mục có thể được liên kết với nhiều sản phẩm.

2. `products` :

- Chức năng: Lưu trữ thông tin về sản phẩm.

3. `users` :

- Chức năng: Lưu trữ thông tin về người dùng của hệ thống, bao gồm cả quản trị viên và khách hàng.

4. `orders`:

- Chức năng: Lưu trữ thông tin về các đơn hàng.

# ĐỒ MẪNH

## Đ

5. ``order_items``:

- Chức năng: Lưu trữ chi tiết về các sản phẩm trong mỗi đơn hàng.

6. ``payments``:

Chức năng: Lưu trữ thông tin về các khoản thanh toán cho đơn hàng.

**Chuẩn bị:**

**Trình tự thực hiện để xây dựng chức năng  
CRUD**

1. Mở hệ quản trị cơ sở dữ liệu tạo cơ sở dữ liệu, giả sử là `ecommerce2024`:

```
create database ecommerce2024;
```

2. Dùng Visual Studio Code mở thư mục dự án Laravel đã tạo:

EXPLORER

> app

```
> bootstrap
```

```
> config
```



#### 4. Khởi động máy chủ local apache

Thực hiện:

lar

# Ma

1. Cập nhật kết nối đến cơ sở dữ liệu trong file .envn:

```
> tests
```

```
> vendor
```

```
  .editorconfig
```

```
  .env
```

```
  .env.example
```

```
  .gitattributes
```

```
10 DB_CONNECTION=mysql
11
12 DB_HOST=localhost
13
14 DB_PORT=3306
```

# powershell + X

```
14 DB_DATABASE=e-commerce2024
15 DB_USERNAME=root
16
17 DB_PASSWORD=mot234567tam
```

**Lưu ý:** Các dòng 13..16 điền theo cấu hình máy tính của người thực hiện.

2. Tạo Migration cho Categories:

Trong Terminal, soạn thảo câu lệnh:

```
php artisan make:migration create_categories_table --create-categories
```

Trong lần tạo bảng đầu tiên, cập nhật migration để tạo bảng categories:

~LAR\_VIDU1

> app

> bootstrap

> config

V database

G

database > migrations>2024\_08\_12\_055742\_create\_categories\_table.php > £s CreateCategories Table >

1

<?php

2

3

use Illuminate\Database\Migrations\Migration;

4

use Illuminate\Database\Schema\Blueprint;

5

use Illuminate\Support\Facades\Schema;

up >

Closure

> factories

✓ migrations

2014\_10\_12\_000000\_create\_users\_table.php

2014\_10\_12\_100000\_create\_password\_resets\_table.php

6

7

0

class CreateCategoriesTable extends Migration

8

{

9

/\*\*

10

\*Run the migrations.

11

,

12

\*

2019\_08\_19\_000000\_create\_failed\_jobs\_table.php

2019\_12\_14\_000001\_create\_personal\_access\_tokens\_table.php

2024\_08\_12\_055742\_create\_categories\_table.php

> seeders



Thực thi migration:

```
php artisan migrate
```

g  
categoric  
S

Nếu thực thi thành công, trong cơ sở dữ liệu sẽ có bảng categories có cấu trúc:

```
categories x

Don't Limit

10 SELECT
FROM
ecommerce2024.categories;
```

The diagram illustrates a 'Result Grid' interface. It features a table with four columns: 'id', 'name', 'created\_at', and 'updated\_at'. Above the table, there is a 'Filter Rows:' label and an 'Edit:' button. The table itself contains three rows of data. The first row has values '1', 'John', '2013-01-01 12:00:00', and '2013-01-01 12:00:00'. The second row has values '2', 'Jane', '2013-01-01 12:00:00', and '2013-01-01 12:00:00'. The third row has values '3', 'John', '2013-01-01 12:00:00', and '2013-01-01 12:00:00'. The 'id' column is highlighted in yellow. The 'name' column is highlighted in light blue. The 'created\_at' column is highlighted in light green. The 'updated\_at' column is highlighted in light orange. The 'Filter Rows:' button is highlighted in light blue. The 'Edit:' button is highlighted in light orange.

| id | name | created_at          | updated_at          |
|----|------|---------------------|---------------------|
| 1  | John | 2013-01-01 12:00:00 | 2013-01-01 12:00:00 |
| 2  | Jane | 2013-01-01 12:00:00 | 2013-01-01 12:00:00 |
| 3  | John | 2013-01-01 12:00:00 | 2013-01-01 12:00:00 |

### 3. Tạo modal cho Categories:

Thực thi lệnh tạo modal:

php artisan make: model Category -m

Cập nhật model  
(app/Models/Category.php):





```

6
7
8 class Category extends Model
9
10     use HasFactory;
11
12     protected $fillable = ['name'];

```

#### 4. Tạo Controller:

Thực hiện lệnh tạo controller CategoryController:



# Da

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

Cập nhật CategoryController với các chức năng hiển thị, thêm, sửa, xóa (CRUD):

```

// app/Http/Controllers/CategoryController.php
namespace App\Http\Controllers;

use App\Models\Category;
use Illuminate\Http\Request;

class CategoryController extends Controller
{
    {
        public function index()

```

```

    $categories
        Category::all();
    return view('categories.index', compact('categories'));
}

public function create()
{
    return view('categories.create');
}

public function store (Request $request)
{
    $request->validate([

    ]);
    'name' => 'required|string|max:255',

    Category::create($request->all());
    return redirect()->route('categories.index')
        ->with('success', 'Category created
successfully.');
```

```

    }

    public function show (Category $category)
    {

    }

    return view('categories.show', compact('category'));

    public function edit (Category $category)
    {

```

```

    }

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

public function update (Request $request, Category $category)
{
    $request->validate([

    ]);

    'name' => 'required|string|max: 255',

    $category->update($request->all());
    return redirect()->route('categories.index')

successfully.');
```

->with('success', 'Category updated

```

    }

public function destroy (Category $category)
{
    $category->delete();
    return redirect()->route('categories.index')

successfully.');
```

->with('success', 'Category deleted

```

    }

}
```

##### 5. Tao Routes:

Cập nhật file  
routes/web.php:

```
use App\Http\Controllers\CategoryController;
```

```
Route::resource('categories', CategoryController::class);
```

Route::resource('categories', CategoryController::class); tự động tạo ra các route cho các

hành động resourceful trong CategoryController, bao gồm:

GET /categories (index) - Hiển thị danh sách các categories.

GET /categories/create (create) - Hiển thị form để tạo mới category. POST /categories (store) - Xử lý việc lưu category mới.

GET /categories/{category} (show) - Hiển thị chi tiết category. GET /categories/{category}/edit (edit) - Hiển thị form để chỉnh sửa category. PUT/PATCH /categories/{category} (update) - Xử lý việc cập nhật category. DELETE /categories/{category} (destroy) - Xử lý việc xóa category.

## 6. Tạo Views:

Các trang hiển thị được đặt trong thư mục  
`resources/views`

# Dat

Các trang hiển thị liên quan đến danh mục được đặt trong thư mục  
`categories`

- Trang chủ của categories:

```
<!-- resources/views/categories/index.blade.php
@extends ('layouts.app')
@section('content')
```

```

<div class="container">
  <div class="row">
    <div class="col-lg-12 margin-tb">
      <div class="pull-left">
        <h2>Categories</h2>
      </div>
      <div class="pull-right">
        ->

        <a class="btn btn-success" href="{
route('categories.create') }}"> Create New Category</a>
      </div>
    </div>
  </div>
  <div class="alert alert-success">
    <p>{{ $message }}</p>
  </div>
</div>

@if ($message
    Session::get('success'))

    <div class="alert alert-success">
      <p>{{ $message }}</p>
    </div>
@endif

<table class="table table-bordered">
  <thead>
    <tr>
      <th>ID</th>
      <th>Name</th>
      <th>Action</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>Category 1</td>
      <td><a href="#">Edit</a> <a href="#">Delete</a></td>
    </tr>
    <tr>
      <td>2</td>
      <td>Category 2</td>
      <td><a href="#">Edit</a> <a href="#">Delete</a></td>
    </tr>
  </tbody>
</table>

```

```

        </thead>
        <tbody>
            @foreach ($categories as $category)
                <tr>
                    <td>{{ $category->id }}</td> <td>{{ $category->name
                        }}</td>
                    <td>
                        <a class="btn btn-info" href="{{
route('categories.show', $category->id) }}">Show</a>
                        <a class="btn btn-primary" href="{{
route('categories.edit', $category->id) }}">Edit</a>
                        <form action="{{ route('categories.destroy',
$category->id) }}" method="POST" style="display:inline">
                            danger">Delete</button>
                            @csrf
                            @method('DELETE')
                            <button type="submit" class="btn btn-
                                </form>
                            </td>
                        </tr>
                    @endforeach
                </tbody>
            </table>
        </div>
    @endsection

```

- Trang thêm danh mục (Create Categories):

```

<!-- resources/views/categories/create.blade.php -->
@extends ('layouts.app')

@section('content')

```

```

<div class="container">
  <div class="row">
    <div class="col-lg-12 margin-tb">
      <div class="pull-left">
        <h2>Create New Category</h2>
      </div>
      <div class="pull-right">
        <a class="btn btn-primary" href="{{
route('categories.index') }}"> Back</a>
      </div>
    </div>
  </div>
  <form action="{{ route('categories.store') }}" method="POST">
    @csrf
    <div class="row">
      <div class="col-xs-12 col-sm-12 col-md-12">
        <div class="form-group">
          placeholder="Name">
            <strong>Name:</strong>
            <input type="text" name="name" class="form-control"

          </div>
        </div>
        <div class="col-xs-12 col-sm-12 col-md-12 text-center">
          <button type="submit" class="btn btn-
primary">Submit</button>
        </div>
      </div>
    </form>
  </div>
@endsection

```

- Trang sửa danh mục (Update Categories):

```

<!-- resources/views/categories/edit.blade.php -->
@extends ('layouts.app')

@section('content')
<div class="container">
    <div class="row">
        <div class="col-lg-12 margin-tb">
            <div class="pull-left">
                <h2>Edit Category</h2>
            </div>
            <div class="pull-right">
                <a class="btn btn-primary" href="{{
route('categories.index') }}"> Back</a>
            </div>
        </div>
    </div>

    <form action="{{ route('categories.update', $category->id) }}"
method="POST">
        @csrf
        @method('PUT')
        <div class="row">
            <div class="col-xs-12 col-sm-12 col-md-12">
                <div class="form-group">
                    <strong>Name:</strong>
                    <input type="text" name="name" value="{{ $category-
>name }}" class="form-control" placeholder="Name">
                </div>
            </div>
            <div class="col-xs-12 col-sm-12 col-md-12 text-center">
                <button type="submit" class="btn btn-
primary">Submit</button>
            </div>
        </div>
    </div>

```



```

        </form>
    </div>
@endsection

```

- Trang xem chi tiết thông tin 1 danh mục:

```

<!-- resources/views/categories/show.blade.php -->
@extends ('layouts.app')

@section('content')
    <div class="container">
        <div class="row">
            <div class="col-lg-12 margin-tb">
                <div class="pull-left">
                    <h2>Show Category</h2>
                </div>
                <div class="pull-right">
                    <a class="btn btn-primary" href="{{
route ('categories.index') }}"> Back</a>
                </div>
            </div>
        </div>
        <div class="row">
            <div class="col-xs-12 col-sm-12 col-md-12">
                <div class="form-group">
                    <strong>Name: </strong>
                    {{ $category->name }}
                </div>
            </div>
        </div>
    </div>
@endsection

```

- Tạo layout gốc: Trong thư **resources/views** mục tạo thư mục **layouts** sau đó tạo file

layout app.blade.php:

```
<!-- layouts/app.blade.php -->
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Categories</title>

    <!-- Bao gồm Bootstrap CSS hoặc các file CSS khác -->
    <link

href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css

    rel="stylesheet">
</head>
<body>
    <div class="container">

        @yield('content')

    </div>

    <!-- Bao gồm các file JS cần thiết -->
    <script src="https://code.jquery.com/jquery-
3.2.1.slim.min.js"></script>

    <script

src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.11.0/umd/popper.
min

.js"></script>

    <script

src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
><

/script>
</body>
</html>
```

Khởi chạy dự án bằng câu lệnh:

Khi

php artisan serve

Khi đó, nếu thành công thì URL của dự án sẽ xuất hiện:

```
PS C:\xampp\htdocs\lar_vidu1> php artisan serve  
Starting Laravel development server: http://127.0.0.1:8000
```

Trong trường hợp này thì URL gốc của dự án là <http://127.0.0.1:8000>

URL của categories là <http://127.0.0.1:8000/categories>