

Câu 03. Tạo ứng dụng web Laravel. Thao tác **tìm kiếm** thông tin sản phẩm, **thống kê** số lượng sản phẩm theo danh mục và **kiểm tra sự hợp lệ** dữ liệu với bảng Categories và Products của cơ sở dữ liệu **Lab14**.

3.1. Xây dựng chức năng thống kê số lượng sản phẩm theo danh mục. Với giao diện ứng dụng web như sau:

The screenshot shows a web browser displaying a Laravel application titled "My Laptop Shop". On the left, there is a sidebar with links for "Products List" and "Categories". The main content area displays a table titled "Statistic of products" with three rows: Dell (Quantity: 6), Asus (Quantity: 2), and Hp (Quantity: 2).

Category Name	Quantity
Dell	6
Asus	2
Hp	2

© 2020 My Laptop Shop.

Gợi ý:

- **B1.** Cấu hình Laravel

- Truy cập vào project Laravel
 - Chép 2 file **index.php** và **.htaccess** trong folder public ra ngoài thư mục gốc.
 - Chỉnh sửa lại file **index.php**
- ```
require __DIR__.'/vendor/autoload.php';
$app = require_once __DIR__.'/bootstrap/app.php';
```

- **B2.** Viết đoạn mã script trong MySQL để xử lý thống kê số lượng sản phẩm theo danh mục.

**CREATE VIEW v\_quantity**

**AS**

**SELECT**

categories.categoryName AS categoryName,

Count( products.productID ) AS quantity

**FROM**

categories

INNER JOIN products ON products.categoryID = categories.categoryID

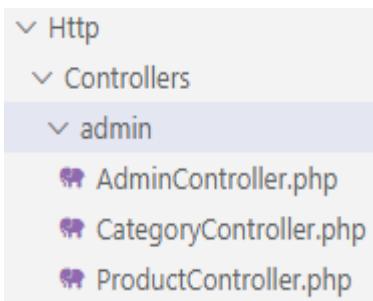
**GROUP BY**

categoryName;

- **B3.** Tạo file controller: **admin\AdminController.php** và viết code cho phương thức **index()** để xử lý thống kê dữ liệu.

```
PS C:\xampp\htdocs\Lab14> php artisan make:controller admin\AdminController
Controller created successfully.
```

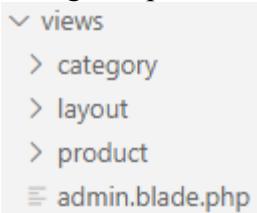
=> Sau khi tạo:



+ Mở file **admin\AdminController.php** và viết code xử lý nghiệp vụ thống kê dữ liệu cho phương thức **index()**

```
<?php
namespace App\Http\Controllers\admin;
use App\Http\Controllers\Controller;
use App\Product;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
class AdminController extends Controller
{
 public function index()
 {
 //c1. Sử dụng view
 $sql = "select * from v_quantity";
 $quantity = DB::select($sql);
 //c2. Sử dụng Query Builder
 /*
 $quantity = DB::table('categories')
 ->join('products','categories.categoryID','=', 'products.categoryID')
 ->select('categoryName',DB::raw('count(products.productID) as quantity'))
 ->groupby('categoryName')
 ->get();
 */
 return view('admin',['quantity'=>$quantity]);
 }
}>
```

- B3. Tạo file **views\admin.blade.php**, xây dựng giao diện cho chức năng thống kê số lượng sản phẩm theo danh mục.



### 3.1. File views\layout\admin\left.blade.php

```
<h4>Products List</h4>
```

```
<h5>Categories</h5>
```

### 3.2. File views\layout\admin\main.blade.php

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <meta http-equiv="X-UA-Compatible" content="ie=edge">
 <title>My Laptop Shop</title>
 <link rel="stylesheet" href="{{ asset('public/css/bootstrap-4.3.1.min.css') }}>
 <style>
 h4, h5 {
 color:#CF8507;
 }
 h4 a, h5 a{
 color:#CF8507;
 text-decoration: underline;
 }
 h4 a:hover, h5 a:hover{
 color:#CF8507;
 text-decoration: none;
 }
 </style>
</head>
<body>
 <div class="container mt-3">
 <div class="row">
 <div class="col-sm-12">
```

```

 @include('layout.admin.header')

 </div><!--header-->
</div>
<div class="row">
 <div class="col-sm-3">
 @include('layout.admin.left')
 </div><!--left-->
 <div class="col-sm-9">
 {{-- @yield: display content of section --}}
 @yield('content')
 {{-- or use code below: @section('content') @show --}}
 </div><!--content-->
</div>
<div class="row">
 <div class="col-sm-12">
 @include('layout.admin.footer')
 </div><!--footer-->
</div>
</div>
</body>
</html>

```

### 3.3. File views\admin.blade.php

```

@extends('layout.admin.main')
@section('content')
<div class="container">
 <div class="row">
 <div class="col-sm-8">
 <h5>Statistic of products</h5>
 <table class="table table-bordered table-sm">
 <thead class="thead-light">
 <th>Category Name</th>
 <th>Quantity</th>
 </thead>
 <tbody>
 @foreach($quantity as $row)

```

```

<tr>
 <td>{{ $row->categoryName }}</td>
 <td>{{ $row->quantity }}</td>
</tr>
@endforeach
</tbody>
</table>
</div>
</div>
@endsection

```

- B4. Định nghĩa Route để định tuyến đến phương index() của admin\AdminController.php

```

Route::group(['prefix' => 'admin'], function () {
 Route::group(['namespace' => 'admin'], function () {
 Route::resources([
 'category'=>'CategoryController',
 'product'=>'ProductController'
]);
 Route::get('/', 'AdminController@index')->name('admin.index');
 });
});

```

=> Lưu ý: định nghĩa lại Route như trên, cũng thay đổi cách định tuyến đến các chức năng category và product.

## My Laptop Shop

[Products List](#)
[List Categories](#)
[Categories](#)

| ID | Category Name | Action                                                           |
|----|---------------|------------------------------------------------------------------|
| 1  | Dell          | <a href="#">Show</a> <a href="#">Edit</a> <a href="#">Delete</a> |
| 2  | Asus          | <a href="#">Show</a> <a href="#">Edit</a> <a href="#">Delete</a> |
| 3  | Hp            | <a href="#">Show</a> <a href="#">Edit</a> <a href="#">Delete</a> |

< 1 2 >

[Add Category](#)

**3.2. Xây dựng chức năng tìm kiếm thông tin sản phẩm.** Với giao diện ứng dụng web như sau:

C ⓘ localhost:8080/Cau03/admin/find ⋮

## My Laptop Shop

 Search

### Products List

#### Categories

### List of products

| Category Name | Product Code | Product Name            | List Price |
|---------------|--------------|-------------------------|------------|
| Hp            | DW0061TU     | HP Pavilion 14 DW0061TU | 590.00     |
| Hp            | BA0046TU     | HP ENVY BA0046TU        | 890.00     |

© 2020 My Laptop Shop.

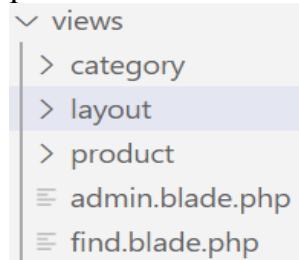
#### Gợi ý:

- B1. Sửa nội dung file `views\layout\admin\header.blade.php`, để thêm thành phần tìm kiếm sản phẩm như hình trên.

```
▽ layout
 ▽ admin
 footer.blade.php
 header.blade.php
```

```
<header>
 <h4>My Laptop Shop</h4>
 <div class="container">
 <div class="row">
 <div class="col-sm-5 offset-7">
 <form style="margin:0px;" action="{{ route('admin.find') }}" method="post" >
 {{ csrf_field() }}
 <input type="text" name="search" class="form-control-sm">
 <input type="submit" value="Search" class="btn btn-info btn-sm mb-1">
 </form>
 </div>
 </div>
 </div>
</header>
<hr class="border-dark">
```

- **B2.** Tạo file **views\find.blade.php**, xây dựng giao diện cho chức năng tìm kiếm sản phẩm.



```
@extends('layout.admin.main')

@section('content')
<div class="container">
 <div class="row">
 <div class="col-sm-10">
 <h5>List of products</h5>
 <table class="table table-bordered table-sm">
 <thead class="thead-light">
 <th>Category Name</th>
 <th>Product Code</th>
 <th>Product Name</th>
 <th>List Price</th>
 </thead>
 <tbody>
 @foreach($result as $row)
 <tr>
 <td>{{ $row->categoryName }}</td>
 <td>{{ $row->productCode }}</td>
 <td>{{ $row->productName }}</td>
 <td>{{ $row->listPrice }}</td>
 </tr>
 @endforeach
 </tbody>
 </table>
 </div>
 </div>
</div>
```

```

</div>
</div>
@endsection

```

- **B3.** Mở file **admin\AdminController.php** và viết code xử lý nghiệp vụ tìm kiếm sản phẩm cho phương thức **find()**

```

public function find(Request $request)
{
 //c1. Sử dụng store procedure
 /*
 $sql = "call timkiem(:condition)";
 $params = ['condition'=>$request->search];
 $result = DB::select($sql,$params);
 */

 //c2. Sử dụng Query Builder
 if(!is_numeric($request->search))
 {
 $result = DB::table('categories')
 ->join('products','categories.categoryID','=','products.categoryID')
 ->select('categoryName','productCode','productName','listPrice')
 ->orwhere('categoryName','like', '%' . $request->search . '%')
 ->orwhere('productCode','like', '%' . $request->search . '%')
 ->orwhere('productName','like', '%' . $request->search . '%')
 ->get();
 }
 else{
 $result = DB::table('categories')
 ->join('products','categories.categoryID','=','products.categoryID')
 ->select('categoryName','productCode','productName','listPrice')
 ->where('listPrice','>', $request->search)
 ->get();
 }
}

```

```

 return view('find',['result'=>$result]);
}

```

- B4. Định nghĩa Route để định tuyến đến phương find() của admin\AdminController.php

```

Route::group(['prefix' => 'admin'], function () {
 Route::group(['namespace' => 'admin'], function () {
 Route::resources([
 'category'=>'CategoryController',
 'product'=>'ProductController'
]);
 Route::get('/', 'AdminController@index')->name('admin.index');
 Route::post('find', 'AdminController@find')->name('admin.find');
 });
});

```

### 3.3. Xây dựng chức năng kiểm tra sự hợp lệ dữ liệu với bảng Categories và Products.

3.3.1. Kiểm tra sự hợp lệ dữ liệu với bảng Categories. Với giao diện ứng dụng web như sau:

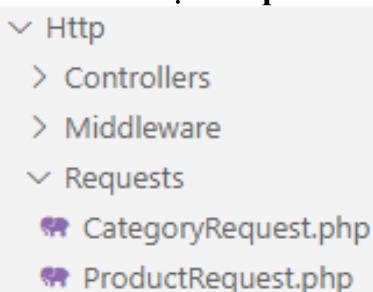
The screenshot shows a web application interface for managing categories. At the top, there's a header with the title "My Laptop Shop". Below it is a search bar with a "Search" button. There are two tabs: "Products List" (underlined) and "Categories". The "Categories" tab is currently selected. On the right, there's a form for adding a new category. It has a text input field labeled "Category Name:" with the placeholder text "Vui lòng nhập danh mục sản phẩm !". Below the input field is a blue "Create Category" button. The footer of the page says "© 2020 My Laptop Shop."

#### Gợi ý:

- B1. Tạo CategoryRequest và ProductRequest để xử lý kiểm tra hợp lệ dữ liệu cho form thêm danh mục sản phẩm và sản phẩm (tương tự với form cập nhật dữ liệu danh mục sản phẩm và sản phẩm)

```
PS C:\xampp\htdocs\Cau03> php artisan make:request CategoryRequest
Request created successfully.
PS C:\xampp\htdocs\Cau03> php artisan make:request ProductRequest
Request created successfully.
```

=> Sau khi tạo Request:



- B2. Mở file Controllers\admin\CategoryController.php, sửa lại mã cho phương thức store() như sau, để xử lý request dữ liệu trên form khi thêm danh mục sản phẩm.

```
namespace App\Http\Controllers\Admin;
use App\Http\Controllers\Controller;
use App\Http\Requests\CategoryRequest;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
class CategoryController extends Controller
{
 public function store(CategoryRequest $request)
 {
 $params = [
 'categoryName'=>$request->categoryName,
 'created_at' => \Carbon\Carbon::now(),
 'updated_at' => \Carbon\Carbon::now()
];
 DB::table('categories')->insert($params);
 return redirect()->route('category.index');
 }
}
```

- B3. Mở file Requests\CategoryRequest.php và viết code để kiểm tra sự hợp lệ dữ liệu trên form.

```
<?php
namespace App\Http\Requests;
use Illuminate\Foundation\Http\FormRequest;
```

```

class CategoryRequest extends FormRequest
{
 public function authorize()
 {
 //Determine if the user is authorized to make this request.
 return true;
 }
 public function rules()
 {
 //Get the validation rules that apply to the request.
 return [
 'categoryName'=>'required'
];
 }
 public function messages()
 {
 return [
 'categoryName.required'=>'Vui lòng nhập danh mục sản phẩm !'
];
 }
}
?>

```

- B4. Mở file views\category\create.blade.php bổ sung mã hiển thị thông báo lỗi khi xảy ra lỗi kiểm tra hợp lệ dữ liệu trên form thêm danh mục sản phẩm.

```

@extends('layout.admin.main')
@section('content')
<h5 style="font-weight: bold">Add Category</h5>
<div class="container">
 <div class="row">
 <div class="col-sm-10">
 <form action="{{ route('category.store') }}" method="post">
 {{ csrf_field() }}
 <div class="form-group">
 <label for="categoryname" style="font-weight: bold">Category Name:</label>
 <input type="text" name="categoryName" id="categoryname" class="form-control">

```

```

{{-- Hien thi thong bao loi --}}
@if ($errors->has('categoryName'))
<p>{{ $errors->first('categoryName') }}</p>
@endif
{{-- --}}
</div>
<div class="form-group">
<input class="btn btn-info btn-sm" type="submit" value="Create Category">
</div>
</form>
</div>
</div>
</div>
@endsection

```

**3.3.2. Kiểm tra sự hợp lệ dữ liệu với bảng Products.** Với giao diện ứng dụng web như sau:

The screenshot shows a browser window with the URL `localhost:8080/Cau03/admin/product/create`. The page title is "My Laptop Shop". On the left, there are two links: "Products List" and "Categories". On the right, there is a form titled "Add Product". The form fields include "Category" (set to "Dell"), "Code" (empty), "Name" (empty), and "List Price" (empty). A red callout box highlights an error message: "Lỗi:" followed by three bullet points: "Vui lòng nhập mã sản phẩm!", "Vui lòng nhập tên sản phẩm!", and "Vui lòng nhập giá sản phẩm!". Below the form is a "Create Product" button.

- **B1.** Mở file Controllers\admin\ProductController.php, sửa lại mã cho phương thức store() như sau, để xử lý request dữ liệu trên form khi thêm sản phẩm.

```
namespace App\Http\Controllers\admin;
use App\Http\Controllers\Controller;
use App\Category;
use App\Http\Requests\ProductRequest;
use App\Product;
use Illuminate\Http\Request;
class ProductController extends Controller
{
 public function store(ProductRequest $request)
 {
 $product = new Product();
 $product->categoryID = $request->categoryID;
 $product->productCode = $request->productCode;
 $product->productName = $request->productName;
 $product->listPrice = $request->listPrice;
 $product->save();
 return redirect()->route('product.index');
 }
}
```

- **B2.** Mở file Requests\ProductRequest.php và viết code để kiểm tra sự hợp lệ dữ liệu trên form.

```
<?php
namespace App\Http\Requests;
use Illuminate\Foundation\Http\FormRequest;
class ProductRequest extends FormRequest
{
 public function authorize()
 {
 return true;
 }
 public function rules()
 {
 //Get the validation rules that apply to the request.
 }
}
```

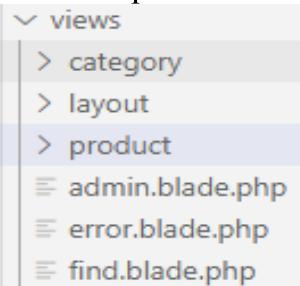
```

 return [
 'productCode'=>'required',
 'productName'=>'required',
 'listPrice'=>'required|numeric'
];
}

public function messages()
{
 return [
 'productCode.required'=>'Vui lòng nhập mã sản phẩm !',
 'productName.required'=>'Vui lòng nhập tên sản phẩm !',
 'listPrice.required'=>'Vui lòng nhập giá sản phẩm !',
 'listPrice.numeric'=>'Giá sản phẩm phải kiểu số !'
];
}
}

```

- **B3.** Tạo file **error.blade.php** để hiển thị tất cả các thông báo lỗi phát sinh trên form thêm sản phẩm.



```

@if(count($errors) > 0)
{{-- Danh sách lỗi --}}

Lỗi:

@foreach ($errors->all() as $error)
- {{ $error }}

@endforeach

@endif

```

- B4. Mở file views\product\create.blade.php bổ sung mã hiển thị thông báo lỗi khi xảy ra lỗi kiểm tra hợp lệ dữ liệu trên form thêm sản phẩm.

```

@extends('layout.admin.main')
@section('content')
<h5 style="font-weight: bold">Add Product</h5>
{{-- Hien thi noi dung loi --}}
<div class="card-body">
 @include('error')
</div>
{{-- --}}
<div class="container">
 <div class="row">
 <div class="col-sm-10">
 <form action="{{ route('product.store') }}" method="post">
 {{ csrf_field() }}
 <div class="form-group">
 <label for="categoryID" style="font-weight: bold">Category:</label>
 <select name="categoryID" id="categoryID" class="form-control">
 @foreach($categories as $category)
 <option value="{{ $category->categoryID }}>
 {{ $category->categoryName }}
 </option>
 @endforeach
 </select>
 </div>
 <div class="form-group">
 <label for="productCode" style="font-weight: bold">Code:</label>
 <input type="text" name="productCode" id="productCode" class="form-control">
 </div>
 <div class="form-group">
 <label for="productName" style="font-weight: bold">Name:</label>
 <input type="text" name="productName" id="productName" class="form-control">
 </div>
 </form>
 </div>
 </div>
</div>

```

```
<label for="listPrice" style="font-weight: bold">List Price:</label>
<input type="text" name="listPrice" id="listPrice" class="form-control">
</div>
<div class="form-group">
 <input class="btn btn-info btn-sm" type="submit" value="Create Product">
</div>
</form>
</div>
</div>
@endsection
```