**LAB 8 MÔN LẬP TRÌNH PHP3: RESTFUL API WITH LARAVEL**

## Bài 1

1. Cài Laravel vào folder **la8**
2. Tạo database có tên **la8** và mở file .env cấu hình kết nối đến database vừa tạo.

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

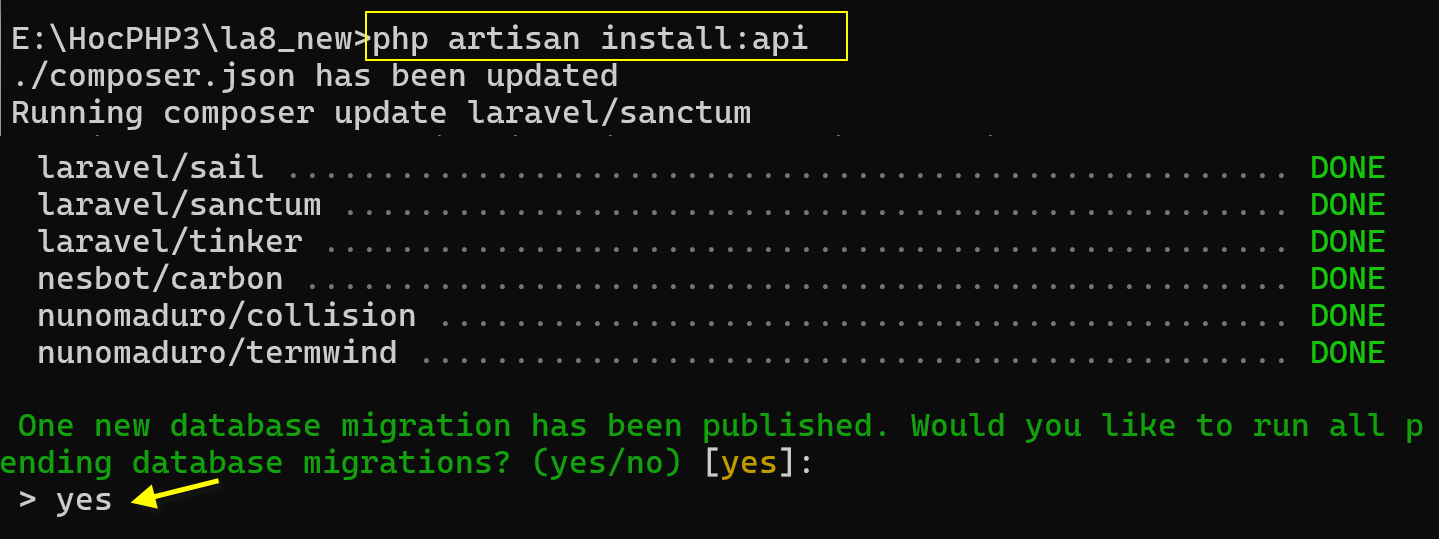
DB\_PORT=3306

DB\_DATABASE=la8

DB\_USERNAME=root

DB\_PASSWORD=

1. Cài đặt api cho ứng dụng : php artisan install:api



1. Cài đặt postman ở địa chỉ <https://www.postman.com/downloads/>
2. Tạo 2 table **loai** và **san\_pham**

Tạo file migration và code hàm up, down

public function up(): void {

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

        $table->id();

        $table->string('ten\_loai');

        $table->integer('thu\_tu')->default(0);

        $table->boolean('an\_hien')->default(0);

        $table->timestamps();

    });

    Schema::create('san\_pham', function (Blueprint $table) {

        $table->id();

        $table->string('ten\_sp');

        $table->integer('gia');

        $table->integer('id\_loai');

        $table->timestamps();

    });

}

public function down(): void {

    Schema::dropIfExists('loai');

    Schema::dropIfExists('san\_pham');

}

* Chạy migration vừa thực hiện để có 2 table

1. Tạo 2 model **loai** và **san\_pham**

* Chạy lệnh tạo model loai và code khai báo thông số :

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class loai extends Model {

    use HasFactory;

    protected $table = 'loai';

    public $primaryKey = 'id';

    protected $fillable = ['ten\_loai', 'thu\_tu', 'an\_hien'];

}

* Chạy lệnh tạo model san\_pham và code khai báo thông số :

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class san\_pham extends Model {

    use HasFactory;

    protected $table = 'san\_pham';

    public $primaryKey = 'id';

    protected $attributes = [];

    protected $dates = [];

    protected $fillable = ['ten\_sp', 'gia', 'id\_loai'];

}

1. Nhập dữ liệu vào 2 table **loai** và **san\_pham**

* Mở DatabaseSeeder.php, import 2 model ở đầu:

use App\Models\san\_pham;

use App\Models\loai;

* Code trong hàm run trong DatabaseSeeder.php

public function run(): void {

    loai::create(["ten\_loai" => "Asus", "thu\_tu" =>"1" , "an\_hien" => "1"]);

    loai::create(["ten\_loai" => "Acer", "thu\_tu" =>"2" , "an\_hien" => "1"]);

    loai::create(["ten\_loai" => "Lenovo", "thu\_tu" =>"3" , "an\_hien" => "1"]);

    loai::create(["ten\_loai" => "MSI", "thu\_tu" =>"4" , "an\_hien" => "1"]);

    $sp\_arr = [

        ["ten\_sp"=>"Asus TUF Gaming 7535HS", "gia"=>18490000, "id\_loai"=>"1"],

        ["ten\_sp"=>"Acer Gaming Aspire A15", "gia"=>16990000, "id\_loai"=>"2"],

        ["ten\_sp"=>"MSI Gaming GF63 12450H", "gia"=>16990000, "id\_loai"=>"4"],

        ["ten\_sp"=>"Lenovo Ideapad 5500H", "gia"=>15990000, "id\_loai"=>"3"],

        ["ten\_sp"=>"Acer Gaming Nitro i7", "gia"=>21990000, "id\_loai"=>"2"],

        ["ten\_sp"=>"Asus Gaming R5 7535HS", "gia"=>19290000, "id\_loai"=>"1"],

        ["ten\_sp"=>"Asus Air 13 M3", "gia"=>38490000, "id\_loai"=>"1"],

        ["ten\_sp"=>"Asus fq5229TU i3 215U", "gia"=>10790000, "id\_loai"=>"1"],

        ["ten\_sp"=>"Acer Aspire 51M i5", "gia"=>11490000, "id\_loai"=>"2"],

        ["ten\_sp"=>"MSI Gaming Thin 12450H", "gia"=>16990000, "id\_loai"=>"4"],

        ["ten\_sp"=>"MSI GF63 12UCX i5", "gia"=>17290000, "id\_loai"=>"4"],

        ["ten\_sp"=>"MSI Sword HX B14VFKG", "gia"=>37990000, "id\_loai"=>"4"],

        ["ten\_sp"=>"MSI HX B14VFKG i7 14700HX ", "gia"=>36590000, "id\_loai"=>"4"],

        ["ten\_sp"=>"Lenovo LOQ i5 12450HX", "gia"=>21490000, "id\_loai"=>"3"],

        ["ten\_sp"=>"Lenovo Gaming 15IAX9 i5", "gia"=>20790000, "id\_loai"=>"3"],

    ];

    for($i=0; $i<count($sp\_arr); $i++) san\_pham::create($sp\_arr[$i]);

}//run

* Chạy seeder để chèn dữ liệu.

## Bài 2: Tạo api lấy tài nguyên

1. **Tạo resource san\_pham để khai báo cấu trúc của từng san\_pham**

* Chạy lệnh : php artisan make:resource san\_pham
* Mở file resource mới tạo (trong app/http/Resources) , khai báo

<?php

namespace App\Http\Resources;

use Illuminate\Http\Request;

use Illuminate\Http\Resources\Json\JsonResource;

class san\_pham extends JsonResource {

    public function toArray(Request $request): array  {

        return [

        'id' => $this->id,

        'ten\_sp' => $this->ten\_sp,

        'gia\_sp' => $this->gia,

        'id\_loai' => $this->id\_loai,

        'cap\_nhat' => $this->updated\_at->format('d/m/Y')

        ];

    }

}

1. **Tạo controller và các action**

* Chạy lệnh: php artisan make:controller SanPhamController
* Code trong controller tạo các action

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Models\san\_pham;

use App\Http\Resources\san\_pham as SanPhamResource;

class SanPhamController extends Controller {

    function index(){

        $listsp = san\_pham::all();

        $data = SanPhamResource::collection($listsp);

        return response()->json($data);

    }

    function chi\_tiet($id = 0){

        $sp = san\_pham::findOrFail($id);

        $data = new SanPhamResource($sp);

        return response()->json($data);

    }

    function sp\_trong\_loai($id=0){

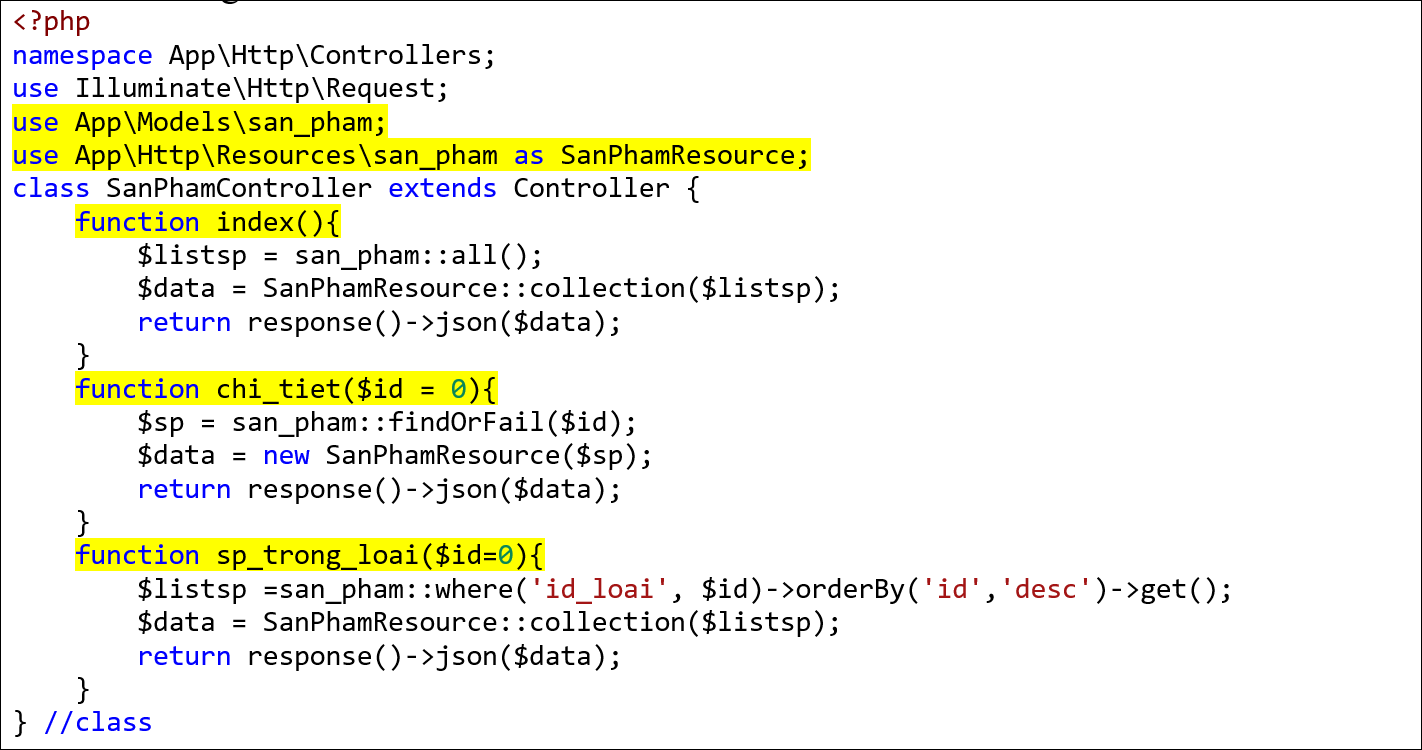
        $listsp =san\_pham::where('id\_loai', $id)->orderBy('id','desc')->get();

        $data = SanPhamResource::collection($listsp);

        return response()->json($data);

    }

} //class



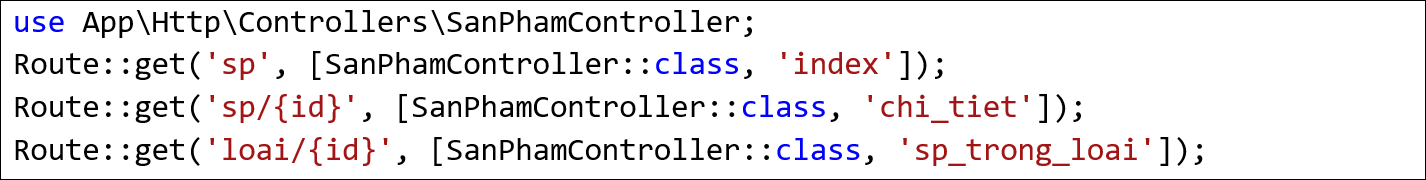
1. **Routing api:** Mở file routes/api.php và code

use App\Http\Controllers\SanPhamController;

Route::get('sp', [SanPhamController::class, 'index']);

Route::get('sp/{id}', [SanPhamController::class, 'chi\_tiet']);

Route::get('loai/{id}', [SanPhamController::class, 'sp\_trong\_loai']);



* Test Routing



## Bài 3: Api quản lý tài nguyên

### Tạo controller api

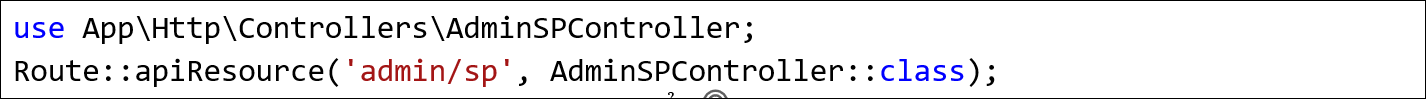
* Chạy lệnh : php artisan make:controller AdminSPController --api

### Routing api

* Code trong file routes/api.php

use App\Http\Controllers\AdminSPController;

Route::apiResource('admin/sp', AdminSPController::class);



* Chạy lệnh **php artisan route:list** để xem các route mới



### Code xử lý thêm, sửa xóa trong controller

* Mở AdminSPController.php và code

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Models\san\_pham;

use App\Http\Resources\san\_pham as SanPhamResource;

class AdminSPController extends Controller {

    public function index() {   }

    public function store(Request $request) {

        $input = $request->all();

        $sp = san\_pham::create($input);

        $data = new SanPhamResource($sp);

        return response()->json($data, 201);

    }

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

        $input = $request->all();

        $sp = san\_pham::find($id);

        $sp->ten\_sp = $input['ten\_sp'];

        $sp->gia = $input['gia'];

        $sp->id\_loai = $input['id\_loai'];

        $sp->save();

        $data = new SanPhamResource($sp);

        return response()->json($data, 200);

    }

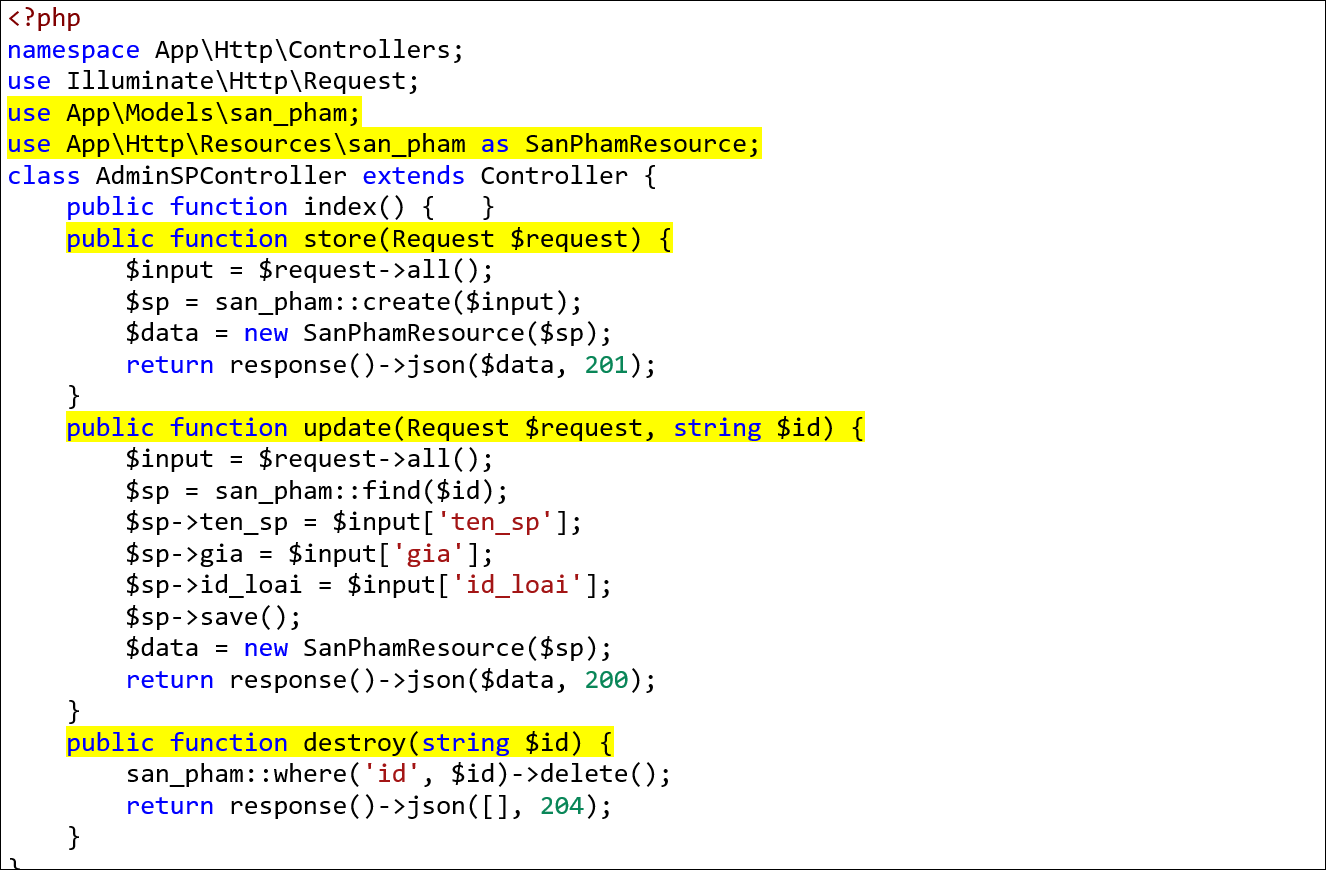
    public function destroy(string $id) {

        san\_pham::where('id', $id)->delete();

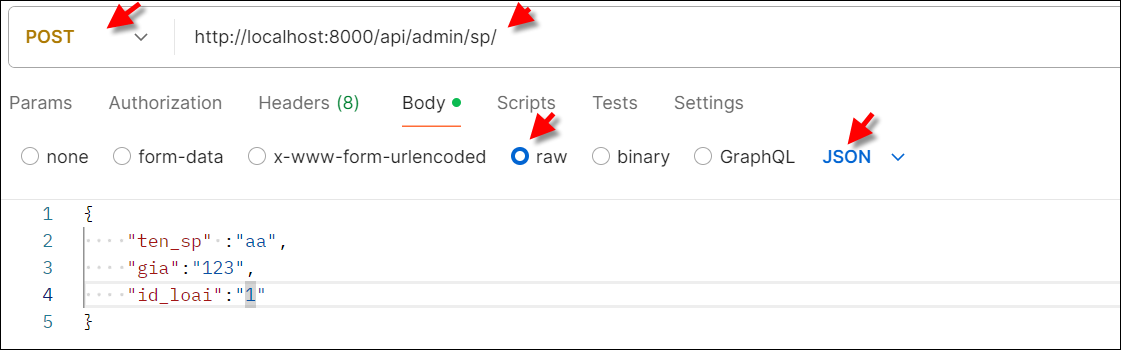
        return response()->json([], 204);

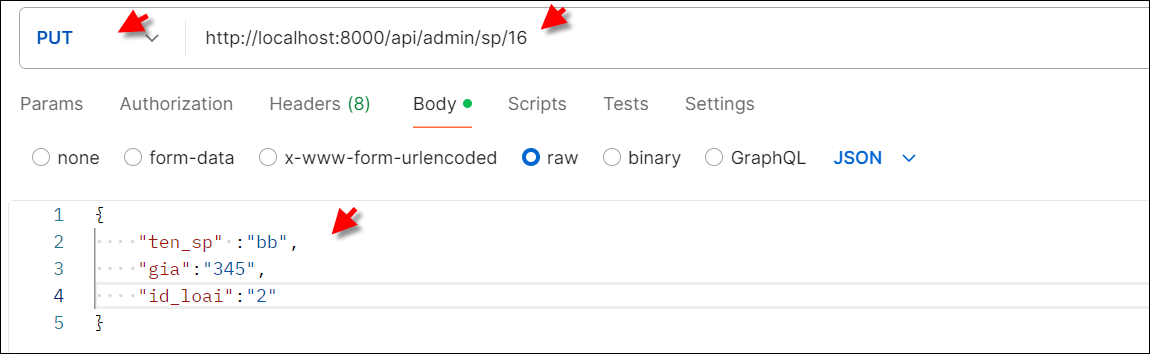
    }

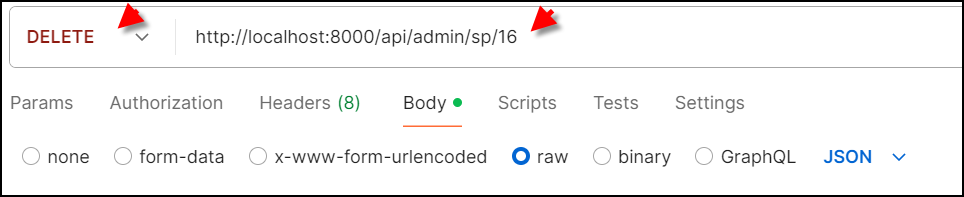
}



* Test với Postman : thêm sửa xóa







## Bài 4: Thực hiện front end: Tương tác api từ trang web

**Tạo folder la8\_frontend**

### Lấy dữ liệu (get) từ api server

* Tạo file index.html trong folder mới tạo và code

<head>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel= "stylesheet" >

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>

<div class="container">

<div id="listsp" class="d-flex flex-wrap"></div>

</div>

</body>

* Code tiếp tục ở cuối file để lấy dữ liệu từ server

<script>

fetch("http://localhost:8000/api/sp")

.then(function(response) {

    console.log("response=", response);

    return response.json();

})

.then(function(data) {

    console.log("data=", data);

    let str =``;

    data.forEach( sp => str+=hien1sp(sp) );

    document.querySelector("#listsp").innerHTML=str;

})

.catch(function(error) {

    console.log('Co loi : \n', error);

});

function hien1sp(sp){

    return `<div class="col-md-3">

        <div class="sp border border-primary m-2 p-2 text-center fs-5">

            <h3 class="fs-5" style="height:40px">${sp.ten\_sp}</h3>

            <p>${sp.gia\_sp.toLocaleString("vi")} VNĐ</p>

            <p>${sp.cap\_nhat}</p>

        </div>

    </div>`;

}

</script>

### Gửi dữ liệu (post) lên server để thêm mới

Tạo file themsp.html có form để user nhập dữ liệu, xong sẽ gửi lên server để lưu. Xem hướng dẫn như sau:

* Tạo file themsp.html và code

<head>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel= "stylesheet" >

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>

<div class="container">

    <div class="col-md-8 m-auto border border-primary px-3 mt-3 shadow-lg">

    <h3 class="text-center">Thêm sản phẩm</h3>

    <div class="mb-3"> Tên sản phẩm

      <input type="text" id="ten\_sp" class="form-control shadow-none border-primary">

    </div>

    <div class="mb-3">Giá sản phẩm

      <input type="number" id="gia" class="form-control shadow-none border-primary">

    </div>

    <div class="mb-3">Loại

      <input type="number" id="id\_loai" class="form-control shadow-none border-primary">

    </div>

    <div class="mb-3">

      <button type="button" onclick="guiSP()" class="btn btn-primary px-3" >Thêm</button>

    </div>

    </div>

</div>

</body>

* Code tiếp tục ở cuối file để gửi dữ liệu lên server

<script>

function guiSP(){

    let url=`http://localhost:8000/api/admin/sp`;

    let ten\_sp = document.querySelector("#ten\_sp").value;

    let gia = document.querySelector("#gia").value;

    let id\_loai = document.querySelector("#id\_loai").value;

    let data = {ten\_sp: ten\_sp, gia: gia, id\_loai:id\_loai}

    let options = {

        method: "POST",

        body: JSON.stringify(data),

        headers: {"Content-type": "application/json; charset=UTF-8"}

    };

    fetch( url ,options )

    .then(function(res) {

        if (!res.ok) throw Error(res.statusText);

        return res.json();  //dùng text() nếu dạng text

    })

    .then(function(data) {

        console.log("Response từ server= ", data);

        alert("Đã thêm xong")

    })

    .catch(function(error) {

        console.log('Có lỗi nè : \n', error);

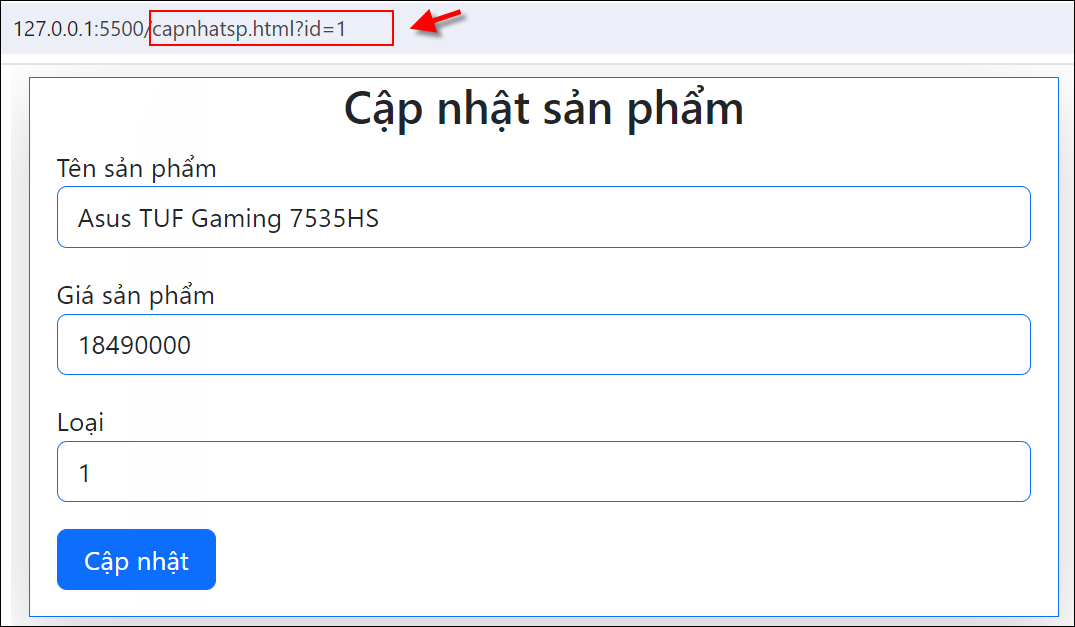
    });

}

</script>

### Gửi dữ liệu (put) lên server để thêm cập nhật

Tạo file **capnhatsp.html** có tham số là **id** , hiện ra form có thông tin sản phẩm để user nhập dữ liệu mới, xong sẽ gửi lên server. Xem hướng dẫn như sau:



* Tạo file capnhatsp.html và code

<head>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel= "stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"> </script>

</head>

<body>

<div class="container">

    <div class="col-md-8 m-auto border border-primary px-3 mt-3 shadow-lg">

    <h3 class="text-center">Cập nhật sản phẩm</h3>

    <div class="mb-3">Tên sản phẩm

        <input type="text" id="ten\_sp" class="form-control shadow-none border-primary">

    </div>

    <div class="mb-3">Giá sản phẩm

       <input type="number" id="gia" class="form-control shadow-none border-primary">

    </div>

    <div class="mb-3">Loại

       <input type="number" id="id\_loai" class="form-control shadow-none border-primary">

    </div>

    <div class="mb-3">

     <button type="button" onclick="guiSP()" class="btn btn-primary">Cập nhật</button>

    </div>

    </div>

</div>

</body>

* Code tiếp tục ở cuối file để lấy sản phẩm về hiện ra trong form

<script>

const params = new URLSearchParams(location.search);

var id = Number(params.get('id'));

let url=`http://localhost:8000/api/sp/${id}`;

fetch(url)

.then(function(response) {

    console.log("sp=", response);

    return response.json();

})

.then(function(sp) {

    console.log("data=", sp);

    document.querySelector("#ten\_sp").value=sp.ten\_sp;

    document.querySelector("#gia").value=sp.gia\_sp;

    document.querySelector("#id\_loai").value=sp.id\_loai;

})

.catch(function(error) {

    console.log('Co loi : \n', error);

});

</script>

* Code gửi sp lên server

function guiSP(){

    let url=`http://localhost:8000/api/admin/sp/${id}`;

    let ten\_sp = document.querySelector("#ten\_sp").value;

    let gia = document.querySelector("#gia").value;

    let id\_loai = document.querySelector("#id\_loai").value;

    let data = {id:id, ten\_sp: ten\_sp, gia: gia, id\_loai:id\_loai}

    let options = {

        method: "PUT",

        body: JSON.stringify(data),

        headers: {"Content-type": "application/json; charset=UTF-8"}

    };

    fetch( url ,options )

    .then(function(res) {

        if (!res.ok) throw Error(res.statusText);

        return res.json();

    })

    .then(function(data) {

        console.log("Response từ server= ", data);

        alert("Đã cập nhật xong")

    })

    .catch(function(error) {

        console.log('Có lỗi nè : \n', error);

    });

}

### Tạo trang lấy danh sách sản phẩm để quản trị



* Code trong hàm index của AdminSPController

public function index() {

    $listsp = san\_pham::all();

    $data = SanPhamResource::collection($listsp);

    return response()->json($data);

}

* Tạo trang danhsachsp.html ở phía frontend và code

<head>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">

<script src= "https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"> </script>

</head>

<body>

<div class="container">

    <div id="listsp"></div>

</div>

</body>

* Code tiếp theo để lấy dữ liệu

<script>

fetch("http://localhost:8000/api/sp")

.then(response => response.json())

.then( data => {

    let str =`<div class="d-flex justify-content-between fs-6">

      <b class="p-2 border col-md-5">Ten SP</b>

      <b class="p-2 border col-md-2">Giá</b>

      <b class="p-2 border col-md-2 text-end">Ngày cập nhật</b>

      <b class="p-2 border col-md-3 text-end">Thao tác</b>

    </div>`;

    data.forEach( sp => str+=hien1sp(sp) );

    document.querySelector("#listsp").innerHTML=str;

})

.catch( error => console.log('Co loi : \n', error));

function hien1sp(sp){

    return `<div class="d-flex justify-content-between fs-6">

      <span class="p-2 border col-md-5">${sp.ten\_sp}</span>

      <span class="p-2 border col-md-2">${sp.gia\_sp.toLocaleString("vi")} VNĐ</span>

      <span class="p-2 border col-md-2 text-end">${sp.cap\_nhat}</span>

      <span class="p-2 border col-md-3 text-end">

          <a href="#">Sửa</a>

          <a href="#" onclick="xoasp(${sp.id})">Xóa</a>

      </span>

    </div>`;

}

</script>

### Gửi yêu cầu xóa (delete) lên server

* Code lại cho link Xóa trong file danhsachsp.html

<a href="#" onclick="xoasp(${sp.id})">Xóa</a>

* Code hàm xoasp ở cuối file để request xóa sản phẩm

<script>

function xoasp(id){

    if (confirm("Xóa sản phẩm này thật không bạn yêu ơi")==false)  return;

    fetch(`http://localhost:8000/api/admin/sp/${id}`, {"method":"delete"})

    .then(response => response.text())

    .then( data => {

        alert("Đã xóa");

        location.reload();

    })

    .catch( error => console.log('Co loi : \n', error));

}

</script>