

Laravel

Installation laravel:

- Créer un projet.
- Configuration de base donnée.
- Créer model.
- Configurer les Tables.
- Configurer les Model.
- Migration.

View Controller:

- Controller.index - View.index.
- Controller.destroy - View.index.
- Controller.show - View.show.
- Controller.create - View.create.
- Controller.store - View.create.
- Controller.edit - View.edit.
- Controller.update - View.edit.

Créer un middleware :

Function Import file (image, file, video, ect ...) :

Factorys & Seeders :

Authentication :

Routage file web.php :

Installation de projet laravel

Créer de project:

- `composer create-project laravel/laravel Name_projet`
- `composer create-project --prefer-dist laravel/laravel project-name "10.*"` version 10

Configuration de base donnee:

File .env : laravel version 10

- ligne 14 : `DB_DATABASE=Name_DB`

Créer model:

- `php artisan make:model Client --all`
- `php artisan make:model Product --all`
- `php artisan make:model Order --all`

Configury les Tables:

```
public function up(): void {  
    Schema::create('clients', function (Blueprint $table) {  
        $table->id();  
        $table->string('name');  
        $table->string('phone');  
        $table->string('email');  
        $table->timestamps();  
    });  
}  
  
public function up(): void {  
    Schema::create('orders', function (Blueprint $table) {  
        $table->id();  
        $table->foreignId('client_id')->constrained();  
        $table->foreignId('product_id')->constrained();  
        $table->timestamps();  
    });  
}
```

Configury Model:

File model client.php

```
use App\Models\Order;

class Client extends Model
{
    use HasFactory;

    protected $fillable = [
        "name",
        "phone",
        "email",
    ];

    public function orders(){
        return $this->hasMany(Order::class, 'clients');
    }
}
```

File model order.php

```
use App\Models\Client;

class Order extends Model
{
    use HasFactory;

    protected $fillable = [
        "client_id",
        "product_id",
    ];

    public function clients(){
        return $this->belongsTo(Client::class, 'client_id', 'id');
    }
}
```

migration:

- php artisan migrate

Views - Controllers

Controller.index

```
public function index(){  
    $clients = Client::all();  
    return view("clients.index",["clients" => $clients]);  
}
```

View . index

```
<table>  
    <thead>  
        <tr class="text-center">  
            <th>id</th>  
            <th>name</th>  
            <th>phone</th>  
            <th>email</th>  
            <th>action</th>  
        </tr>  
    </thead>  
    <tbody>  
        @foreach($clients as $index => $a)  
            <tr class="text-center">  
                <td>{{ $a->id }}</td>  
                <td>{{ $a->name }}</td>  
                <td>{{ $a->phone }}</td>  
                <td>{{ $a->email }}</td>  
                <td>  
                    <form action="{{ url('clients/'. $a->id. '/delete') }}" method="POST">  
                        @csrf  
                        <a href="{{url('clients/'. $a->id) }}"> show</a>  
                        <a href="{{url('clients/'. $a->id . '/edit') }}">edit</a>  
                        <button type="submit" onclick="return confirm('Are You Sure ??')">delete</button>  
                    </form>  
                </td>  
            </tr>  
        @endforeach  
    </tbody>  
</table>
```

Controller . destroy

```
public function destroy(Request $request)
{
    $data = Order::find($request->id);
    $data->delete();
    return redirect("orders");
}
```

Controller . show

```
public function show(Request $request) {
    $data = Order::findOrFail($request->id);
    return view("orders.show",["order" => $data]);
}
```

View . show

```
<div align="center">
    <h1>detail order</h1>
    <b>client name</b><p>{{ $order->clients->name}}</p>
    <b>client phone</b><p>{{ $order->clients->phone}}</p>
    <b>client email</b><p>{{ $order->clients->email}}</p>
    <b>product name</b><p>{{ $order->products->name}}</p>
    <b>product price</b><p>{{ $order->products->price}}</p>
    <b>date creation</b><p>{{ $order->created_at}}</p>
</div>
```

Controller. Create

```
public function create() {  
    $clients = Client::all();  
    $products = Product::all();  
    return view("orders.create",["products" => $products,"clients" => $clients]);  
}
```

View . Create

```
<form action="{{ url('orders') }}" method="POST">  
    @csrf  
    <div class="form-group mb-3">  
        <label>Clients:</label>  
        <select name="client_id">  
            <option selected disabled >----Choisissez Client----</option>  
            @foreach ($clients as $e)  
                option value={{ $e->id }}>{{ $e->id }} - {{ $e->name }}</option>  
            @endforeach  
        </select>  
    </div>  
    <div>  
        <label >Products:</label>  
        <select name="product_id" >  
            <option selected>----Choisissez Product----</option>  
            @foreach ($products as $e)  
                <option value={{ $e->id }}>{{ $e->id }} - {{ $e->name }} </option>  
            @endforeach  
        </select>  
    </div>  
    <button type="submit" >Enregister</button>  
</form>
```

Controller . Store

```
public function store(Request $request) {  
    Order::create($request->all());  
    return redirect("orders");  
}
```

Controller . edit

```
public function edit(Request $request) {  
    $clients = Client::all();  
    $products = Product::all();  
    $data = Order::findOrFail($request->id);  
    return view("orders.edit",["order" => $data, "products" => $products, "clients" => $clients]);  
}
```

View . Edit

```
<form action="{{ url('orders/' . $order->id) }}" method="POST">  
    @csrf  
    <div>  
        <label>Clients:</label>  
        <select name="client_id" >  
            <option selected disabled value="{{ $order->client_id }}">{{ $order->clients->name }}</option>  
            @foreach ($clients as $e)  
                <option value="{{ $e->id }}">{{ $e->id }} - {{ $e->name }}</option>  
            @endforeach  
        </select>  
    </div>  
    <div>  
        <label>Products:</label>  
        <select name="product_id" >  
            <option selected value="{{ $order->product_id }}">{{ $order->products->name }}</option>  
            @foreach ($products as $e)  
                <option value="{{ $e->id }}">{{ $e->id }} - {{ $e->name }}</option>  
            @endforeach  
        </select>  
    </div>  
    <button type="submit" >Enregister</button>  
</form>
```

Controller . update

```
public function update(Request $request) {  
    $data = Order::find($request->id);  
    $data->update($request->all());  
    return redirect("orders");  
}
```

middleware

Créer un middleware:

- `php artisan make:middleware CheckProductClient`

File middleware CheckProductClient.php :

```
use App\Models\Product;
```

```
use App\Models\Client;
```

```
class CheckProductClient
```

```
{  
    public function handle(Request $request, Closure $next): Response  
    {  
        $products = Product::all();  
        $clients = Client::all();  
        if (count($products) == 0) {  
            return redirect('products');  
        }  
        if (count($clients) == 0) {  
            return redirect('clients');  
        }  
        return $next($request);  
    }  
}
```

Add middleware to App/http/kernel.php:

```
protected $middlewareAliases = [  
    'check.products.clients' => \App\Http\Middleware\CheckProductClient::class,  
];
```

Check middleware to web.php:

- `Route::get('/orders', 'index') -> middleware('check.products.clients');`

Import File (image)

Function import file:

```
public function store(Request $request)
{
    $data = new Product();
    $data->name = $request->name;
    $data->price = $request->price;
    if($request->hasfile('image')){
        $image = $request->file('image');
        $path = $image->store('Images','public');
        $data->image = $path;
    }
    $data->save();
    return redirect("products");
}
```

```
public function update(Request $request)
{
    $data = Product::find($request->id);
    $data->name = $request->name;
    $data->price = $request->price;
    if($request->hasfile('image')){
        File::delete(storage_path('app/public/' . $data->image));
        $image = $request->file('image');
        $path = $image->store('Images','public');
        $data->image = $path;
    }
    $data->update();
    return redirect("products");
}
```

Storage link public:

- php artisan storage:link

Factory & Seeders

Créer Factory:

- `php artisan make:factory Clientfactory`

```
class ClientFactory extends Factory
{
    public function definition(): array
    {
        return [
            'name' => fake()->name(),
            'phone' => fake()->phoneNumber(),
            'email' => fake()->unique()->safeEmail(),
        ];
    }
}
```

Créer Seeder:

- `php artisan make:seeder ClientSeeder`

```
class ClientSeeder extends Seeder
{
    public function run(): void
    {
        Client::factory(10)->create([]);
    }
}
```

Run Seeder:

- `php artisan db:seed ClientSeeder`

Authentication

Install Laravel Breeze:

- `composer require laravel/breeze --dev`
- `php artisan breeze:install`

Run Migrations:

- `php artisan migrate`

Routes file web.php

```
use App\Http\Controllers\OrderController;
```

```
Route::controller(OrderController::class)->group(function () {  
    Route::get('/orders', 'index');  
    Route::get('/orders/create', 'create');  
    Route::get('/orders/{id}', 'show');  
    Route::get('/orders/{id}/edit', 'edit');  
    Route::post('/orders', 'store');  
    Route::post('/orders/{id}', 'update');  
    Route::post('/orders/{id}/delete', 'destroy');  
});
```