**Insertar imágenes en Laravel**

Paso 1

Lo primero que debemos hacer es permitir él envió de archivos desde el formulario, vamos a resources->products->create.blade.php y agregamos lo siguiente:

Código original. Activamos la funcionalidad de file con la siguiente línea 'file' => true

          @include('products.fragment.error')

          {!! Form::open(['route' => 'products.store']) !!}

@include('products.fragment.form')

{!! Form::close() !!}

Código agregado.

          @include('products.fragment.error')

          {!! Form::open(['route' => 'products.store', 'file' => true]) !!}

@include('products.fragment.form')

{!! Form::close() !!}

Igual hacemos con edit

Código original

@include('products.fragment.error')

{!! Form::model($product, ['route' => ['products.update', $product->id], 'method' => 'PUT']) !!}

@include('products.fragment.form')

{!! Form::close() !!}

Código agregado

@include('products.fragment.error')

{!! Form::model($product, ['route' => ['products.update', $product->id], 'method' => 'PUT', 'file' => true]) !!}

@include('products.fragment.form')

{!! Form::close() !!}

Con esto le decimos que este formulario va a enviar archivos y Laravel lo va a permitir.

Paso 2

Colocamos el campo de la imagen en el formulario común para crear y editar ubicado en:

**resources->views->fragment->form.blade.php**

Colocamos el campo de primero

<div class="form-group">

     {{ Form::label('file', 'Imagen') }}

     {{ Form::file('file') }}

</div>

<div class="form-group">

     {!! Form::label('name', 'Nombre del producto') !!}

     {!! Form::text('name', null, ['class' => 'form-control']) !!}

</div>

<div class="form-group">

     {!! Form::label('short', 'Dscripción breve del producto') !!}

     {!! Form::text('short', null, ['class' => 'form-control']) !!}

</div>

<div class="form-group">

     {!! Form::label('body', 'Dscripción del producto') !!}

     {!! Form::textarea('body', null, ['class' => 'form-control']) !!}

</div>

<div class="form-group">

     {!! Form::submit('ENVIAR', ['class' => 'btn btn-primary']) !!}

</div>

Paso 3

Vamos a config->filesystem.php vamos a la funcionalidad de disk, vamos a salvar un archivo en el disco, Laravel por defecto usa la carpeta storage para almacenar archivos, pero nosotros vamos a utilizar la carpeta public así que nos ubicamos disk y modificamos la línea 53

51 'public' => [

52 'driver' => 'local',

53 'root' => storage\_path('app/public'),

54 'url' => env('APP\_URL').'/storage',

55 'visibility' => 'public',

56 ],

Vemos la modificación usamos el helper public\_path(),

51 'public' => [

52 'driver' => 'local',

53 'root' => public\_path(),

54 'url' => env('APP\_URL').'/storage',

55 'visibility' => 'public',

56 ],

Paso 4 vamos al controlador

Usamos la clase de almacenado Storege

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Product;

use App\Http\Requests\ProductRequest;

use Illuminate\Support\Facades\Storage;

Vamos al método Storage y colocamos la acción para agregar la imagen debajo de $product->save();

public function store(ProductRequest $request)

{

$product = new Product;

$product->file = $request->file;

$product->name = $request->name;

$product->short = $request->short;

$product->body = $request->body;

$product->save();

//IMAGE

if($request->file('file')){

$path = Storage::disk('public')->put('image', $request->file('file'));

$product->fill(['file' => asset($path)])->save();

}

return redirect()->route('products.index')->with('info', 'El producto fue guardado');

}

if($request->file('file')){

$path = Storage::disk('public')->put('image', $request->file('file'));

$product->fill(['file' => asset($path)])->save();

}

Preguntamos si tenemos un archivo en el campo file, que corresponde al campo que tenemos en el formulario de resurce->products->fragment->form.blade.php. Si se cumple la condición creamos la ruta donde vamos a guardar esa imagen, $path = Storage::disk(‘public’)->put(‘imagen’, $request->file(‘file’)). $ path es igual a decir ruta a almacenar en el disco sección public la que configuramos en filesystem, vamos a subir una imagen y la vamos a subir en una carpeta que se va a llamar image, allí vamos a guardar la imagen que tenemos en el campo file. Nota: si la carpeta image no existe Laravel la va a crear, luego actualizo la ruta que se ha creado, el archivo se llama file y con el helper asset($path) se crea la ruta completa, esa ruta es la que se va a guardar en la base de datos.

Redireccionamos a índex y mandamos un mensaje “El producto fue guardado”.

**Arreglamos la vista**

Nota: la migración debe estar con el campo correspondiente para la imagen, nosotros creamos una migración sin ese campo vamos a agregar ese campo ahora que estamos en la fase de desarrollo y no estamos en producción. La regla es que se planifique la estructura de la migración muy bien para evitar tener que hacer arreglos de última hora estando ya en producción.

Vamos a la carpeta database->migrations->2018\_07\_29\_145155\_create\_products\_table.php y agregamos la línea para el campo de imagen. Si tenemos un Factories por usar los Seeders vamos a la carpeta database->factories->UserFactory.php y agregamos la línea para el campo imagen.

Migrations

2018\_07\_29\_145155\_create\_products\_table

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

$table->increments('id');

$table->string('name');

$table->mediumText('short');

$table->text('body');

$table->string('file')->nullable()->default('sin-imagen.jpg');

$table->timestamps();

});

Factories

$factory->define(App\Product::class, function (Faker $faker) {

return [

'file' => $faker->imageUrl($width = 50, $height=100),

'name' => $faker->sentence(2),

'short' => $faker->text(140),

'body' => $faker->text(600),

];

});

Y el archivo index que está ubicado en resources->views->products->index.blade.php y agregamos la columna correspondiente para almacenar la imagen. Debemos tener una vista semejante a la siguiente.

