convencion de anotacion: base nivel 1 y #numeracion y code nivel2 nivel3 error notas(~ significa renglon siguiente)

1º) crear las siete rutas necesarias para hacer nuestro crud

A-en admin.php creamos la ruta a) la configuramos en admilte.php

B-php artisan make:controller Admin/TagController -r

C- importamos ese controlador a nuestro archivo de rutas, luego le asignó el control de esa ruta a ese controlador, y como nombre Admin.tag

vamos a el controlador e (importamos el modelo TAG)

D-cambiamos el parametro id por (Tag $tag)

E-return c/u de los metodos index,create,show,edit (return view('admin.tag.index',xxx); xxx=compact('tag') para edit y show

F-creamos las vistas show,edit,create,index

\*\*\* nota OJO en vez de copiar-- un paso a paso pa segirlo y reproducir el resultado

G-EXTENDEMOS implamentamos la plantilla .:. la copiamos en c/u esta en resources\views\admin\index.blade.php

H) en adminlte.php configura rutas,iconos, (se borro otros links(multi level)

\*\*\* nota en index.blade.php deberiamos mostrar los post del usuario que esta autenticado y porveer una maner da busqueda (con un plugin da jsquery datable

I- nota trabajaremos con livewire para lo que en adminlte.php al final 'livewire' => false, a true

a-creamos un new componente php artisan make:livewire Admin/PostsIndex recordemos que este rendrizara a livewire.admin.posts-index'

b- pa poder utilizarlo en admin.blade.php en @section('content') @livewire('admin.posts-index')

c. en PostsIndex .php llamamos a use App\Models\Post; vamos a toma a PostsIndex.php como si fuera un clontrolador y

y recuperamos la INFO ->

seria 2º) $posts = Post::where('user\_id', auth()->user()->id)->paginate(); //pedimos que filtre de acuerdo al campo user-id

2º)-recuperamos INFO… en TagController.php en index( ) --$tag = Tag::all( ); // recupera datos -en

3º) pero en posts-index'.blade.php -> no hay que borrar… y creamos los div.card …etc según 3º y va con la mismo code … div.card~div.card-body… etc

Nota\*\*\*como el uesr de id=1 no tenia post no le aparecio nada - ->cambiamos en seeders pa que a menos user le toque mas posts y php artisan migrate:fresh –seed

b-queeremos q la paginacion aparesca al final -> debemos import (use Livewire\WithPagination;) y adentro use WithPagination; y en la vista debajo del card-body div.card-footer -- error pagina feo xq qfalta protected $paginationTheme = "bootstrap"; r 12

c-queremo un input search arriba pa filtrar la info -> en posts-index'.blade.php r4 un div.card-header~input.(form-control, placeholder)

d-queremos a medida que se escribe se filtre - ahi r13 public $search; -- y pedimos que se sincronice con esta c) cuando vamos al ultimo reg y lo metemos en search no aparece -> debemos add public function updatingSearch(){$this->resetPage();}--se activa solo cuando la propiedad search cambie

e- cuando no alla registros -> letrero quelo diga -> con un if encerramos toda la tabla

error mostraba el input pegado xq mal card estaba crad

3ª) queremo que los ultimos post aparesca de 1ero

3º)-en index.blade.php borramos las 2 ultimas seccione de las 5 y en @content --.card ~ table class table ~ table-striped ~ thead.tr.th\*3

(1ID,2A,NAME,3colspoan2) y tbody.@foreach.($tags as $tag).tr^td{{$tag->id}}-td{{$tag->name}}-td.<a class="btn btn-primary btn-sm"

href="{{route('a.t.edit',$tag)}}">editar<>-form action="{{route('a.t.s',$tag)}}" method="POST">-@csrf-@method('delete')-buttom

class="btn btn-danger btn=sm" type="submit">Eliminar< \*\*\*nota pa que amvos este a la derecha en td widtn="10px"

\*\*\*nota error1 undefine $tag cambio a tags pero igual RTO error1a falta ,compact('tags')) error2 boton editar no aparece RTO faltan class="btn..." error3 no color po fila rta a)tenia car es card-body b) el @csrf y @method(..) no estaban dentro del form c) tenia stripe y es striped

A- sidebar se mueva en adminlte.php r 88 'layout\_fixed\_sidebar' => null, a true (pa que no se mueve sidebar con contenido)

B-boton crear colocamos un enlace con forma de boton en la cabeza pa q nos redireccione a admon.tag.crear etiquetas

en index.blade.php @section('content-header') <a.class="btn...float-right" href="{{r('a.t.c',)}}>nueva etiqueta<a/>"

\*\*\*nota en category este boton se puso en @section('content') sin un card-header

4º)-en create.blade.php en @section('conten) card~card-body~ (L-collec o (L-c)) form::open(['route'=>'a.t.store'])~div class=form-group~form::leabel('name','Nombre')~ ~form::text('name',null,'class'=>'form-control','placeholder'=>'ingerese el nombre de la etiqueta...')~form::colse( )

\*\*\*L-collective a) el mande el @csrf b)si no le decimos el method x defauld es POST

B-SLUG duplicamos el div interior pa el slug .:. queda el div~div~form::close ...cambiamos 3 slug por tag y al final q diga slug

a-pegamos el plugin conversor de slug

D-en el input slug (,'readonly')evitar que el usuario pueda cambiar el slug

E-mostrar un select desplegable que permita al user escoger un color -> etiquetas (selct y option)

div.class="form-group"~label.for>color:<>~select.name="color" id=" class="form-control>~option\*3>a,b,c<

F-boton enviar -> a) form::submit. 'crear etiqueta,['class'=>'...} b) en TagController.php store( ) return $request->all( );

le pedimos q nos retur lo que se esta mandando por el formulario

G-select con collective form::label.'color','color:'~form::select.('color',$colors,null,['class'=>'form-control'] yyyy en TagController.php en create( ) $colors=['red'=>'color rojo',....] y ,compact('color'));

Cambia xq escogemos es categories PostController.php -use Categories- en create() { $cate=Categories::pluck('name');

en create copiar el select de tags

H- textArea pa colocar el extracto del post div.form-group~form::label('extract','extracto)~form::texarea('extract',null,'class=>form-control)~~~se repi lo anterior y pa body ~ form::submit('crear post',['class'=>'primary] \*\*\*nota la idea es permitir escribir texto enriquecido ahi ->Plugin(CKEditor)->bajamos el cdn e indicamos a cual tex area se aplica

i-@19:49 status opcion agregar (borrador, publicado) input radio div.form-group~<p>~label~from::radio

V- un boton al clic redirect a blog.text/admin/post/create y alli muestre un form -> en index.blade.php r5 @section('content\_header') add <a.btn.flow-right href="{{route(admin.posts.create en create.blade.php div.card~div.card-body~ un desplegable pa escoger cartegoria en PostController.php -use Categories- en create() { $cate=Categories::pluck('name'); en create copiar el select de tags 14:32 textArea pa colocar el extracto del post div.form-group~form::label('extract','extracto)~form::texarea('extract',null,'class=>form-control)~~~se repi lo anterior y

pa body ~ form::submit('crear post',['class'=>'primary] \*\*\*nota la idea es permitir escribir texto enriquecido ahi ->Plugin(CKEditor)->bajamos el cdn e indicamos a cual tex area se aplica

A-eEstado @19:49 status opcion agregar (borrador, publicado) input radio div.form-group~<p>~label~from::radio

si el usuario envia el form y ha marcado BORRADOR (que se le permita estar vacia(s)etiquetas, extracto, body si PUBLICADO -> todo lleno .:. dos validaciones

a) db null a extrac ^ body ^ etiquetas -> en database\migrations\2021\_04\_13\_232155\_create\_posts\_table.php b) las reglas las podriamos incluir en el controlador pero como sin un poco complejas xq depende de status usaremos un formRequest

c) creamos php artisan make:request StorePostRequest limitar a user el ingreso a cierta seccion si cumple cierta condicion ej: admini bloger BUTBUT esto lo manejaremos son L jpermmission -=> true ahi y en la RULES()

d) en postcontroller.php use el form request d1) en create.blade.php imprimir los mensajes de validacion @error ... --probamos y bien las 2 valida

e) dentro del form debemos integrar el id del user actual (pa relacionarlos) con un imput oculto hidde --un problema el usuario podria reemplazar el valor en el inspector pa impedirlo vovlemos a; formreques SotorePo.... si utilizaremos el authoriza()

f) guarda la info en db g) habilitar asignacion masiva dos maneras co fillable [los que queremos] y con guarder(colocamos los campos que queremos evitar que se llenen por asignaicon masiba)

\*\*\*nota cuando guardamos una parte se guardara en la tabla post pero otra va en la tabla relacion post\_tag

-> en PostController.php y ponemos una condicional si enviamos infor de etiquetas una vez que recupero la relacion ->attach([])

Vi- permitir subir una imagen y una vez alli relacionarla con el post que crearemos BUTBUT si probamos subir un post ahora sin imagen da un error xq la plantilla en la parte que muestra la imagen dice: (accede a la relacion que hay entre post e imagenes) -- style="background-image: url({{Storage ::url($post->imagen->url) y como no hay imagen da null y de ahi el error la RTO add condicional

b- que se muestre una imagen por defecto en el caso anterior pixabay.com copiamos un url y en un @else la ponemos bi) cuando filtramos por etiquetas da el mismo error en la vista de etiquetas y en show

2o) volvemos a form de crear new post add una opcion pq el usuario pueda seleccion imagenes // encima de extrac r64 add GRID asi: div.row~div(x c/U column) .:. div\*2(class=col el

i| pa la ima default (queremos solo ocupe la 1/2 del ancho y 16:9 .:. un div.image-wrapper

ii| el acceso file pa la imag y se podra escribir un texto relacionado

A) previsualizar la imag selcciona en el div 1, con js i) identificamos <ima id="picture" ii) en @section('js') el escript...,

r) el code se mantendra a la escucha de cualquier cambio que hagamos con el id file,

rr) si selcc un archivo activa una funcioin que tenemos cambiarimagen() esta transforma la ima en base 64 y busca picture y reemplaza el valor del atribu src por la imag

B) en admin/postcontroller.php en store() q me retorne lo que se esta mandando desde el form r r->all() y ensayamos enviar una imagen ,

i) en el campo file no se esta mandando una imagen sino un string "avatar.jpg",

ii) se debe habilitar el envio de archivos 'file'=>true indica envio de file dentro del form

C) c/v que enviemos una ima quedara temp en una carpeta tal (en el servidor)

ii) debemos mover la imag a una carp de nuestro proy desde la cual podremos acceder luego a img desde el navegador

iii) moverla a public/storage con la ayuda de facade de lavaravel STORAGE -> postController.php ...use

iv) ya vimos que en {config\filesystems.php 'default' => env('FILESYSTEM\_DRIVER', 'public'),} al pasarlo a public logramos que la imagenes se suban

dentro de app/public/storage/nuestro y asi acceder desde app/public/ acceso storage (\*\*\*nota pa generar el acceso dire a storage es con php artisan storage:link

v) debemos guardar esta url en la tabla image y relacionarlo con el post que vamos a crear

vi) como el envio de imagen no va hacer obligatorio colocare una condicional y ahi pregunto si se esta enviadno una imagen desde el form si si

mover la ima de la temp a la carp storage/file y guargar esto en $url

b)->quiero generar un new registro en la tabla image y ese relacionarlo con el post que acabo se crear

c) r49 $post->image()->create(['url' => $url asi logramos que el campo imageabla type e imagiable\_id tomen la informacion del id de este post

y de modelo de este post

d) habilitar la asiganacion masiva en el modelo image app\Models\Image.php protected $fillable ['url'];

3o) add reglas de validacion para las imagenes a) en el boto de carga de file q solo deje ver imagenes a) en create.blade.php botn file 'accept'=>'image/\* y b) en en e

formrequest StorePostRequest.php r36 'file' => 'image'c) imprimimos el error