1.File Storage

Go to .env

FILSESYSTEM\_DRIVER=public

We will store all in strage facade cause its a simple way to handle files

2.File Upload Form

Go to create.blade.php make th form multi enctype

also need in edit.blade.php

3.Handling File Uploads

in PostController in store method

$hasFile=$request->hasFile('thumbnail');

dump($hasFile);

if ($hasFile) {

$file = $request->file('thumbnail');

dump($file);

dump($file->getClientMimeType());-image/jpeg

dump($file->getClientOriginalExtension());-.jpg

dump($file->store('thumbails'));-Thumbnail folders in created in app/store folder

}

die;

4.Using Storage Facades to create files

$hasFile=$request->hasFile('thumbnail');

dump($hasFile);

if ($hasFile) {

$file = $request->file('thumbnail');

dump($file);

dump($file->getClientMimeType());

dump($file->getClientOriginalExtension());

dump($file->store('thumbails'));

//this is same as upper code in a broad form

dump(Storage::disk('public')->put('thumbails', $file));

//You can also write

dump($file->storeAs('thumbails', $blogPost->id . '.'. $file->guessExtension()));

//in local storage not working

// dump(Storage::disk('local')->put('thumbails', $file, $blogPost->id . '.' . $file->guessExtension()));

}

5.Getting url of storedFile

//There was a problem getting URL

$hasFile = $request->hasFile('thumbnail');

// dump($hasFile);

if ($hasFile) {

$file = $request->file('thumbnail');

dump($file);

dump($file->getClientMimeType());

dump($file->getClientOriginalExtension());

dump($file->store('thumbails'));

dump(Storage::disk('public')->put('thumbails', $file));

$name1 = $file->storeAs('thumbails', $blogPost->id . '.'. $file->getClientOriginalExtension());

dump(Storage::url($name1));

// }

die;

**6.Image model,1 to 1 relations and Migrations**

At first create a IMAge model with migration then slowly

in Blogpost.php

public function image(){

return $this->belongTo(BlogPost::class);

}

public function BlogPost(){

return $this->One(Image::class);

}