Extensões úteis para VC

- PHP IntelliSense
- PHP Namespace Resolver
- Laravel-blade
- Laravel Blade Snippets
- Laravel goto view
- Laravel extension pack
- Laravel go-to-components
- Laravel Extra Intellisense

Comandos Gerais:

- php artisan serve: inicia o Servidor
- php artisan route:list : lista as rotas que temos registadas
- Route::get('/home_contacts', [HomeController::class, 'index'])->name('index.contacts'): exemplo de rota com Get
- php artisan make:resource TaskResourceCollection –collection: criar recurso de colecção

Views:

- Mostrar Mensagem da Sessão

  if(session('message'))

  session('message')

  @endif

- asset('caminhodentrodapastapublic/meuficheiro'): Chamar um asset da pasta Public (imagens, CSS, JS,..)
- onclick/onsubmit="confirm('Tem a certeza?')" ; confirmar certa acção, por exemplo Delete
- Para adicionar imagens ao form

  enctype="multipart/form-data"

  No input:

  type="file"

  accept="image/*"

Controller:

- php artisan make:controller DashboardController: cria um controller
- php artisan make:controller UserController –r : cria um controller com recursos
- return view ('nameofview', compact('contacts') : retorna uma view com a variável/array contacts

- return back(): volta para a rota anterior
- return redirect()->route('nomeDaRota')->with('message', 'envio de mensagem'): redireccionar para a rota especifica com mensagem
- request->has('email') -> isset;

Migrações e Base de Dados

- mysql --version
- mysql -u root -p (entrar como root)
- CREATE DATABASE sd_app;

- php artisan make:migration nomeDaMigração: cria uma migração
- php artisan migrate: corre as migrações novas
- php artisan migrate:status
- php artisan migrate:rollback: reverte a migração
- php artisan make:migration add_user_type_to_users_table --table=users: adiciona uma migração para adicionar uma nova coluna à tabela de users

Facades para o use

- use Illuminate\Support\Facades\DB: Facade para uso do Query Builder
- use Illuminate\Support\Facades\Auth: Facade para uso da Autenticação
- use Illuminate\Support\Facades\Hash: encriptar a password

OUTRAS UTILIDADES:

Debug:

- var_dump($var)
- dd($var);
- file_put_contents("test.txt","Hello World. Testing!"): uma variável
- file_put_contents("output.txt", print_r($user, true)): array ou objecto:
- ->toSql(): debug de queries

Dar permissões a uma pasta

- sudo chmod -R 777 nossaPasta

File Storage:

- Storage::url(‘aminhaimagem.img’) -> retorna o caminho através do servidor.
- Storage::path(‘aminhaimagem.img’) -> retorna o caminho na nossa aplicação.
- Storage:: exists(‘aminhaimagem.img’) -> retorna true ou false conforme exista ou não
- Storage:: size(‘aminhaimagem.img’) -> retorna o tamanho da imagem em bytes
- Storage:: lastModified(‘aminhaimagem.img’) -> retorna a data em que foi modificado
- Storage:: download(‘aminhaimagem.img’) -> retorna o download
- Storage::put('localização relativa ao disco', 'Conteúdo do ficheiro'); -> guardar ficheiros
- Storage::prepend('localização relativa ao disco', 'Conteúdo do ficheiro Adicionado'); -> Adicionar conteúdo ao ficheiro
- Storage::apend('localização relativa ao disco', 'Conteúdo do ficheiro Adicionado'); -> Adicionar conteúdo ao ficheiro
- Storage::putFile('pasta para o ficheiro', 'objecto do ficheiro'); -> Adicionar imagem a uma pasta
- Storage::putFileAs('pasta para o ficheiro', 'objecto do ficheiro', 'nome do ficheiro'); -> Adicionar imagem com nome específico a uma pasta