&nbsp cách chữ

setTimeout(function(){

window.location.reload();

}, 800);

**text-justify**

1) URL post: update profile của KH ở bảng m\_customer

//2) Thêm select box chọn School từ bảng m\_school

3) URL post: update thông tin child của KH ở bảng m\_child

4) URL post: delete child

5) URL post: add child

Note:

Thông báo: search lobibox

Chú ý refference các ID cho đúng

getID hiện tại của user đang đăng nhập : Auth::user()

Sử dụng Query builder để get dữ liệu

Validate input

Sử dụng DB Transaction

cus[{}{}{}]

\public function homepage\_view\_pdf(Request $request)

base64\_decode

$filename = base64\_decode(basename($request->filename, '.pdf')).'.pdf';

$sorts\_query\_low = $request->fullUrlWithQuery(['sorts' => 'low']);

$provider = \Laravel\Socialite\Two\FacebookProvider::class;

* **One to One** : Một Một.

return $this->hasOne('App\FeaturedImages','forigen\_key');

return $this->hasOne('App\FeaturedImages','forigen\_key','local\_key');

Đảo ngược

return $this->belongsto('App\Posts','forigen\_key','local\_key');

* **One to Many**: Một nhiều.

return $this->hasMany('App\Posts','author\_id','local\_key');

Đảo ngược

return $this->belongsto('App\Posts','forigen\_key','local\_key');

* **Many to Many**: Nhiều nhiều.

return $this->belongstoMany('App\category','table\_medium');

Đảo ngược

return $this->belongstoMany('App\Posts');

<http://localhost/vi/login>

<http://localhost/api/get/test1>

php artisan make:migration create\_users\_table --create=users

php artisan migrate

php artisan make:seeder UsersTableSeeder

php artisan db:seed

php artisan db:seed --class=UsersTableSeeder

php artisan migrate:refresh --seed

public function run()

{

$this->call([

UsersTableSeeder::class,

PostsTableSeeder::class,

CommentsTableSeeder::class,

]);

}

AdminLTE

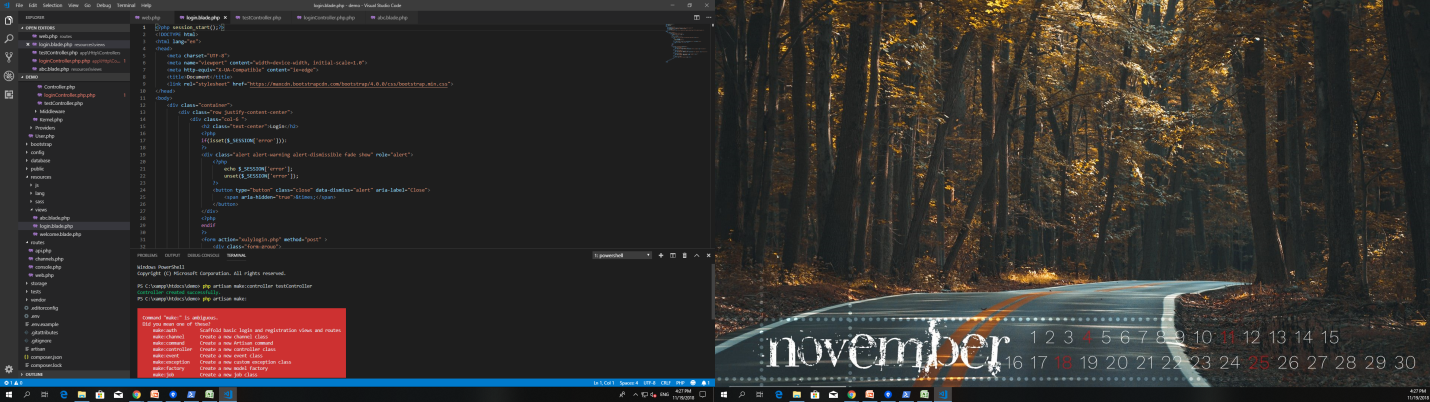
#refs

Commit -> fetch- >pull ->về nhánh –rebase dev -> lên dev merge nhánh -> push

Database: Migrations

**Database: Seeding**

**Authentication**



Nên sử dụng bản lavarel 5.6

Gọi phương thức ::

<https://allaravel.com/laravel-tutorials/laravel-eloquent-orm-phan-1-thao-tac-voi-database-qua-eloquent-model/>

view

khai báo { }

<?php ?>

@php @endphp

@if (isset($a)) // <?php if ?>

{{$a}}

@else

<h1>aaaaaaa</h1>

@endif

{{$status}} //<?=?>

{!!$user!!} //<?=?> chạy,đọc hiểu html css các thẻ <h1> <a> …

{{$user->username}} // <?=$user->name

<?php echo $user->raw\_password; ?>

@php echo $user->password; @endphp

WEB.php

Route::get('redirect/{id}','MyController@redirect'); //truyền dữ liệu lên

Redirect/3 truyền id= 3 lên

'MyController có function redirect($id){} biến truyền vô là id

CONTROLLER

Luôn nằm trong try catch để không xuất hiện lỗi

Bên view kiểm tra lỗi bên server cũng phải kiểm tra lỗi lại.

Kiểm tra username chưa điền , bên server kiểm tra lại

public function redirect($id){

try{

$user = User::where('id',$id)->first(); //lấy user có id = $id đầu

if($user){

return view('view\_demo',['b'=>$user]); // trả về người dùng’

//dữ liệu trong mảng vs key = b

}else{

}

}catch(\Exception $ex){

return "Xay ra loi trong qua trinh xu ly";

}

}

return view('view\_demo')->with('status','Tai khoan khong ton tai'); sử dụng with tương tự như session

redirect('redirect/3'); // điều hướng

redirect('login'); // đăng nhập không thành công thì quay về login

return view('view\_demo',["user"=>"<h1>Tai khoan khong ton tai!</h1>"]); trả về dữ liệu chứa html -> bên view sử dụng {!!$user!!} để đọc được html

Cấu trúc một controller

$rules

$Validator = Validator::make($request->all(),$rules);chạy kiểm tra

public function submit\_login(Request $request){

$rules = [

'username' => 'required|min:5|max:20',

'password' => 'required|min:6|max:30'

];

$validator = Validator::make($request->all(),$rules);

if($validator->fails()){

return $validator->errors()->first(); //trả về lỗi đầu tiên, trả tất cả sử dụng

$validator->errors()->all();

}else{

try{

$username = $request->username;

$pass = $request->password;

if(Auth::attempt(['username' => $username, 'password' => $pass])){

return view('wrapper');

}else{

$\_SESSION['error'] = "Sai thông tin đăng nhập";

return view('login.login');

// return "Tai khoan khong chinh xac!";

};

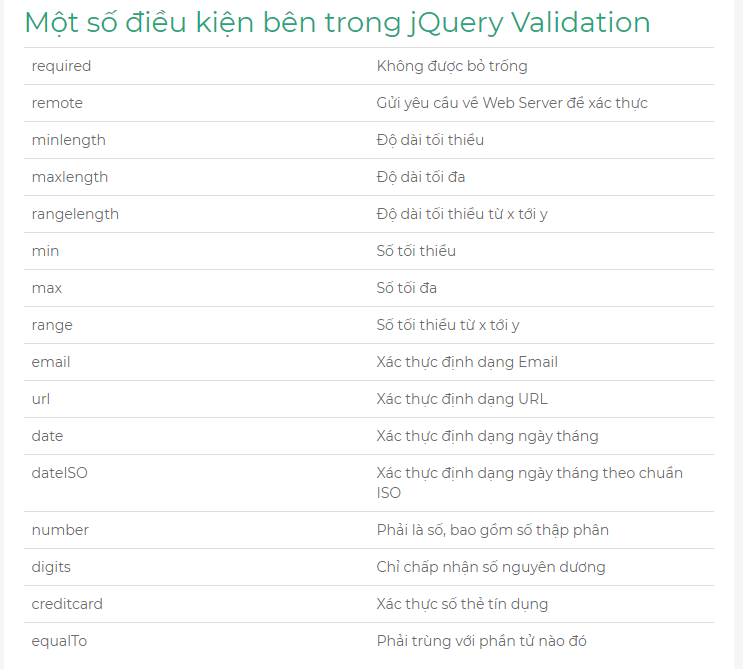
}catch(\Exception $ex){

}

}

datatable jquery

jquery-validation



<https://thienanblog.com/javascript/jquery/huong-dan-su-dung-jquery-validation/>

custom jquery validation

https://viblo.asia/p/tong-quan-ve-datatable-jquery-QWkwGnBnG75g