1 -----Update laravel 6.0 version-----

Step no 01:

open the file composer.jason and write the text "5.1\*" to "6.0.\*"

Step no 02:

open the terminal and type "composer update" and wait till update the laravel version

Step no 03:

open the terminal and type "php artisan serve" and wait

"now the laravel version will be successfull updated"

2 -----Remove laravel-----

composer global remove laravel/installer

3 -----create laravel framework-----

composer create-project laravel/laravel laravel "6.0.\*"

4 -----query builder-----

first add the libarary "use Illuminate\Support\Facades\DB;"

view the data

$user=db::table('table\_name')->get();

print\_r($user);

insert the data

$user=db::table('table\_name')->insert([ 'name'=>'ali kamran','address'=>'gujranwala']);

print\_r($user);

update the data

$user=db::table('table\_name')->where('id','1')->update('name','athar');

print\_r($user);

delete the data

$user=db::table('table\_name')->where('id','1')->delete();

print\_r($user);

5 -----fetch data using with model-----

first make the model in singular and add this model in controller like "use App\model\_name"

if database table prular

then write this code in controller function "return model\_name::all();

if database table name different then write this code in model not controller " protected $table='table\_name' "

6 -----Create login page (auth)-----

first run this command on terminal and install the packages "composer require laravel/ui"

second run this command on terminal "php artisan ui vue --auth"

after this run this command "npm install" and then "npm run dev"

7 -----Migrate the tables-----

first config the .env file and write the database name

then add this file in app->provider->appservicesprovider add this library "use Illuminate\Support\Facades\Schema;"

then write this line in boot function "Schema::defaultStringLength(191);"

after this run the command "php artisan migrate"

8 -----create master page-----

step no 01: first create master.blade.php page in view and design the header and footer with the help of html:5

step no 02: the create home or other page and access the master page this command @extends('path\_master\_page')

and also write this command top of the page

step no 03: then in master page type this command in body tage @yield('type\_any\_name')

step no 04: then in child page write this command after @extends command @section('type\_yield\_name')

step no 05: then write the child page code and last write this command @endsection

9. -----Generate pdf file-----

step no 01. first of all install pdf packages, write this code in terminal and enter

"composer require barryvdh/laravel-dompdf"

step no 02. After updating composer, add this file in config->app.php file and past it in "providers" last

"Barryvdh\DomPDF\ServiceProvider::class,"

step no 03. add this file in controller "use PDF" as a library of controller

Step no 04. write this code in controller function

$data=[];

$pdf = PDF::loadView('file\_name', $data);

return $pdf->stream('file\_name.pdf');

10. ------insert data into database using model----

step no 01: create form into view and add "@csrf" token after starting form tab. action is="any\_name" and this name

is also write in web.php in controller route::post(type\_here);

step no 02: make controller and make a function and write this code in controller function

function save(Request $req){

$any\_name = new student;

$kamboh->table\_column\_name= $req->form\_input\_name;

$any\_name->table\_column\_name= $req->form\_input\_name;

$any\_name->table\_column\_name= $req->form\_input\_name;

$any\_name->table\_column\_name= $req->form\_input\_name;

$any\_name->save();

}

}

step no 03: make model and add this model in controller using this command "use App\model\_name"

if you do not make created\_at and updated\_at coloum in sql db then write this command in model

"public $timestamps = false;"

if table name is different in sql db then we access the data using model in this command and type

this command in model

"protected $table='table\_name';"

11. ------fetch data into database using model----

step no 01: make controller and make a function and write this code in controller function

public function save(){

$any\_variable= model\_name::all();

return view('page\_name\_in\_view\_where show the data',['any\_key\_name'=>$varivable\_name]);

}

step no 02: make view page and write this code here

<div>

<ul>

@foreach($variable\_name\_same\_as\_controller\_variable as $any\_variable)

<li>

{{$variable\_name->column name which column is required to fetch the data}}

</li>

@endforeach

</ul>

</div>

step no 03: make model and add this model in controller using this command "use App\model\_name"

if table name is different in sql db then we access the data using model in this command and type

this command in model

"protected $table='table\_name';"

12. -----update data-----

step no 01: make view page and create form like insert

step no 02: make controller and type code in controller function

this code is update address where name is type in textbox

public function save(Request $req){

model\_name::where('name',$req->name)

->update(['address'=>$req->address]);

}

}

this is route example of update data

Route::view('form','fetchdata');

Route::post('update','ali@save');

13. -----delete data-----

same as update data just write this code in controller

public function save(Request $req){

model\_name::where('name',$req->name)

->delete(['name']);

}

}

another delete data with select and option method

---view page form code---

<form action="update" method="POST">

@csrf

<select name="id" >

<option value="6">ali kamran</option>

<option value="7">mozam</option>

<option value="9">ahmad</option>

<option value="10">athar</option>

<option value="11">usman</option>

</select>

<button type="update">delete</button>

</form>

----controller page function code----

public function save(Request $req){

$ali=student::find($req->id);

$ali->delete();

}

14. -----fetch data from database with pdf-----

1.view page formate

@foreach($students as $student)

<tr>

<td>{{$student->id}}</td>

<td>{{$student->name}}</td>

<td>{{$student->address}}</td>

<td>{{$student->cast}}</td>

<td>{{$student->grade}}</td>

</tr>

@endforeach

2. controller function formate

public function pdf(){

$students=student::all();

$pdf = PDF::loadView('view',['students'=>$students]);

return $pdf->stream('customer.pdf');

}

15. ----send email----

1. type this code in web.php

Route::get('/', function () {

// return view('welcome');

$to\_name='ali kamran';

$to\_email="alikamrankamboh@gmail.com";

$data=array('name'=>'type here message tittle','body'=>"type here your message body");

Mail::send('mail',$data,function($message)use ($to\_name,$to\_email){

$message->to($to\_email)->subject('Web Testing mail');

});

2. and then create the view page name mail.blade.php and type this code

<strong>{{$name}}</strong>

<p>{{$body}}</p>

16. -----Migration command-------

type this command in terminal and then enter "php artisan make:migration migration\_name --create=migration\_name"

17. ---------------validate function---------------

this code is in the controller function

$request->validate([

'address' => 'require',

email=> email

]);

-----show error message top of the page------

@if($errors->any())

@foreach($errors->all() as $error)

<ul>

<li> {{$error}} </li>

</ul>

@endif

@endforeach

-----show error message bottom every input type------

type this code every input field bottom

@error('address')

{{$message}}

@enderror

18. ------------create login page----------

this code is written in controller function

$email=$req->email;

$password=$req->password;

$dat=DB::select('select \* from useres where email=? and password=?',[$email,$password]);

if(count($dat))

{

$req->session()->put('data',$dat);

return redirect('homepage');

}

else{

return redirect('/userlogin')->withErrors(['Wrong G Email OR Password']);

}

19. --------convert data into excel sheet-----------

first install this package in terminal and enter "composer require maatwebsite/excel"

second add this file extension in config/app.php last in the files " Maatwebsite\Excel\ExcelServiceProvider::class,"

third add this file extension in same file last in alliss " 'Excel' => Maatwebsite\Excel\Facades\Excel::class, "

forth add this files in controller "use Excel" & "use Maatwebsite/Excel/Concerns/FromCollection"

five write this code in controller

public function exportData(){

return Excel::download(new DataExport,'table\_name.xlsx');

}

}

six this class add end the controller function and end of the class braces

class DataExport implements FromCollection{

function collection(){

return Model\_name::all();

}

}

20. -------------component in laravel 7 (reuse data)-----------

first of all make the component using this command "php artisan make:component component\_name"

this component show in resource->view->component folder. In this folder all the component reside,and type any html text in this component how reuse this any view file

second fetch the component data just add this link in view file <x-component\_name /> after this reuse the component data in any view file

21. ------session in laravel---------

first make a view page and make a input field(email,password) and make route this form in controller index function and write this code in controller

public function index(Request $request){

$request->session()->put('data',$request->input());

if($request->session()->has('data')){

return redirect('practiceProfile');

}

second make user profile view page and write this code user profile

<h1>user profile</h1>

<h2>welcome {{session('data')['name']}} </h2> // access the user name in user profile page

<a href="practiceLogout">logout</a> // delete the session in this link

third this is the web.php route of session

route::view('hi','practiceLogin'); // access form page where user enter your name and password

route::post('login','practiceLogin@index'); // form page redirect this controller

Route::get('/practiceProfile', function () { // access user profile page using condition

if(!session()->has('data')){ // if session not found then redirect login page

return redirect('hi');}

return view('practiceProfile'); // else redirect user profile page

});

Route::get('/practiceLogout', function () { // delete the session

session()->forget('data');

return redirect('hi');

});

}

22. ---seeder----- Enter dumy data into database

1. Enter this command in terminal and hit "php artisan make:seeder seeder\_name"

to see this seeder open database folder and then seeder folder

2. Add this two library in seeder file

use Illuminate\Support\Facades\DB;

use Illuminate\Support\Str;

3. Write this code in run function of seeder

DB::table('table\_name')->insert([

'name' =>Str::random(8),

'email' =>Str::random(10).'@gmail.com',

'address' =>Str::random(8)

]);

4. Write this command in terminal and enter random data into database

php artisan db:seed --class=seeder\_name

23. ---------upload file------------- laravel 8 short code

1.this code paste in controller

function fileupload(Request $req){

return $req->file('fileName')->store('anyFolderName'); // stroage->app view save pic

}

2. this is view code

<form action="upload" method="POST" enctype="multipart/form-data">

@csrf

<input type="file" name="pic"><br>

<button type="submit">upload</button>

</form>

24. ------ Accessors--------- show data through model with editing

controller code

use App\Models\student; // add library

function fileupload(){

return student::all();

}

model code

function getNameAttribute($value){ // capital first letter

return ucFirst($value);

}

function getEmailAttribute($value){

return ucFirst($value);

}

function getAddressAttribute($value){ // add some values

return $value.",Pakistan";

}

25. ---------Mutator----------- // save the data with extra words user not enter this word

paste this code in model function

public $timestamps=false;

public function setNameAttribute($value){

return $this->attributes['name']= 'Mr. '.$value;

}

function setEmailAttribute($value){

return $this->attributes['email']= $value."@hotmail.com";

}

26. ----------Multiple DB Connection------------

step no 01

open .env file and past it in double like this and change name

DB\_CONNECTION\_SECOND=mysql

DB\_HOST\_SECOND=127.0.0.1

DB\_PORT\_SECOND=3306

DB\_DATABASE\_SECOND=test2

DB\_USERNAME\_SECOND=root

DB\_PASSWORD\_SECOND=

step no 02

open config->database.php file and paste it double and change name like this

'mysql2' => [

'driver' => 'mysql',

'url' => env('DATABASE\_URL'),

'host' => env('DB\_HOST\_SECOND', '127.0.0.1'),

'port' => env('DB\_PORT\_SECOND', '3306'),

'database' => env('DB\_DATABASE\_SECOND', 'forge'),

'username' => env('DB\_USERNAME\_SECOND', 'forge'),

'password' => env('DB\_PASSWORD\_SECOND', ''),

'unix\_socket' => env('DB\_SOCKET', ''),

'charset' => 'utf8mb4',

'collation' => 'utf8mb4\_unicode\_ci',

'prefix' => '',

'prefix\_indexes' => true,

'strict' => true,

'engine' => null,

'options' => extension\_loaded('pdo\_mysql') ? array\_filter([

PDO::MYSQL\_ATTR\_SSL\_CA => env('MYSQL\_ATTR\_SSL\_CA'),

]) : [],

],

step no 03

add this library into controller using query builder to fetch the data

use Illuminate\Support\Facades\DB;

step no 04

add this code in controller function

return DB::connection('second\_db\_name')->table('second\_table\_name')->get(); // fetch another db data

return DB::table('table\_name')->get(); // fetch same db data

---using model to fetch data--------

simply add this code in model and fetch data

public $connection="db\_name";

and fetch the data using model through this code

return model\_name::all();

27. ----------Depended Select Box----------

https://investmentnovel.com/laravel-dependent-dropdown-tutorial-with-example/