**PHP**

**VIEWS**

Views are located at resources –views

So we create the views app – resources- view rightclick –php file – filename.**blade**.php(with extension) (where html codes are written for the view of the page)- ok

Create a route for the view in the routes folder – route::get(‘/filename’,controllermethod@contact);

So the method has to be defined in the controller which has a php file.

**Public function contact()**

**{**

**Return view (‘contact’);**

**}**

How to pass data to the views:

* **Blade engine is a templating engine that laravel uses to show php execution in a very simple way**
* **We can chain the functions or the method with the parameter using (“->with” keyword) it takes two parameters ,variable name and variable value itself.**

**Eg: public function show\_post(id)**

**{**

**return view (‘post’)->with(id, $id);**

**}**

* **And make changes in the post blade view page by entering curly braces**

**eg: <h1> post {{$id}}</h1>**

* **There is a native function called** *compact*, **as we can pass many parameters that has same name and variable it will convert it into the variable.**
* **Eg: public function show-post($id, $name, $password)**

**{**

**Retrun view (‘post’) compact {‘id’ , ‘name, ‘ password’);**

**}**

Blade is a php templating agent .

**Database**

**Command:** php artisan make:migration add \_is\_admin\_column \_to\_posts\_table ---table =”posts”(table name)

Path **: App – database- migration.**

To refresh and to get updated one click on synchronize which is there at rightclick of the migrations. **Command :** php artisan migrate to migrate the table

In order to create the new migration file

Command : php artisan make:migration create\_posts\_table –create(flag whih comes with 2 --) = “posts”.

No sensitive imformation has to be kept n the .env(environmental )file because chances that git bash ignores.

When timestrap is used in the creation of the database it generates the **“created at” and “updated** **at”** in the database table

Command: **php artisan migrate:reset** it will rollback all the previous migrations/delete all the migrations

Command: **php artisan migrate:refresh** it will automatically rollback all the tables and migrate them.

**RAW SQL**

Route::get(‘/update’,function(){

$update =DB::update(‘update post set title =”update the title” where id= ?’,[1]);

Return $updated;

}

Route ::set(‘/delete’, function()

{

$delete = DB::delete(‘delete from Posts where id-?’[1]);

Return $deleted;

}

Route::get(‘/insert’, function()

{

DB::insert(‘insert into posts(posts here is a tablename)(title, content) values(?,?)’,[‘PHP with Laravel’,’Laravel is a best thing]);

}

Route::get(‘/read’, function()

{

$results = DB::select(“select \* from Posts where id=?’,[1]);

Return $results

});

**Var\_dump** – displays the content with its type and number of characters in it.

Route:;get(‘/update’, function()

{

$updated =DB::update(update posts set title =”update new title”where id=’?’,[1]);

Return $updated;

});

Create a page:

Route::get(‘/create’, function()

{

Post::create([‘title’=>’create method’, ‘content’ => ‘I am learning’]);

}

);

Route::get(‘/forcedelete’, function()

{

Post::withtrashed()->where(is admin ,0)->forcedelete);

}

**); It deletes the entire method**

Route::get(‘/delete’, function()

{

$post= Post::find(3) ; **[here we can pass parameters to delete the specific id or name etc]**

$post->delete();

});

**Withtrashed() is a method to delete the items which are present in the method along with the trash items.**

**Onlytrashed() is a method to delete the items which are there in the trash.**

Route::get(‘/delete’, function()

{

Post::destroy([3,4]); **(insert al the ids that has to be deleted, we can delete multiple methods)**

});

**Basic insert:**

Route::get(‘/basicinsert’,function()

{

$post **= new** post**;…..[if u wanna insert at specific location or file then use $post = post::find[2]; ]**

$post->title=”my new post”;

$post->content = “ ksdjflsughmcBV.VFSLJGH;jfduol”;

$post->save();

});

Route::get(‘/findmore’, function()

{

$post = post::findorfail( it is a predefined method which takes id’s as the parameters,2 );

Return $post;

});

OR

Route::get(‘/findmore’, function()

{

$post=post::where(‘user\_count’,’<’,50)->firstorfail();

});

**To delete a record**

Route::get(‘/deleted’,function()

{

Sdeleted=DB::delete(delete from posts where id=’?’, 2);

});

**Command :** php artisan make:model filename ( it creates a model)

Models will be there in the apps folder

Route::get(‘/findwhere’ , function()

{

$post =post::where(‘id’, 2)->orderBy(‘id’, ‘desc’) ->take(1)->get();

Return post;

});

// **here B should be in capital for** orderBy **clause. And ‘**desc**’ is descending order.**

**To restore a record which we have deleted, inorder to do that first we need to find them.**

Route::get(‘/restore’, function()

{

Post::withtrashed()->where(‘is\_admin’,0)->restore();

});