**প্রয়োজনীয় website:**

<https://laravelcollective.com/>

<https://laracasts.com/>

<http://laravel-recipes.com/>

**Laravel Installation and Configaration**

1. Composer Download then setup to **xampp>php>php.exe**
2. Laravel installation using **composer global require "laravel/installer"**
3. Now open command window using **ctrl+shift+right click to the htdocs folder** then paste the 2 no code.
4. Now we have to search laravel then ctrl+shift+right click and open command window and write **php artisan serve** for server start
5. **Routing**: হচ্ছে URL; কোন পেজ view করার জন্য কোন নাম define করে দেওয়াই হচ্ছে Routing এর কাজ। i.e. localhost/**contact**, localhost/**about** etc.
6. Routes.php page এ যে get method টা আছে এটি দুটি parameter নেয়; একটি string and anonymous function
7. **View** থাকে হচ্ছে Resource folderএর ভিতর।
8. **Controller** থাকে হচ্ছে App>HTTP>Controllers.
9. View এর প্রতিটি পেজকে আমরা Route থেকে Call করতে পারি কিন্তু এটি কোন সঠিক পদ্ধতি না; সেজন্য আমরা View এর পেজ সমূহকে Controller এর মাধ্যমে Call করব।
10. Controller তৈরি করার জন্য Command হচ্ছে **php artisan make:controller page\_name**
11. প্রতিটি পেজে master page কে import বা inherit করে কাজ করতে হয় (এটি parent class এবং এক একটি page হচ্ছে object. Master পেজ create করতে হয় View এ।
12. **কিভাবে page, css, script, image, link etc. dynamic করতে হয় (using laravel 5)**:

এজন্য ২টি কাজ করতে হয়:

১. Page গুলোকে পোজেক্ট ফোল্ডারে রাখতে হয়

২. CSS ও অন্যান্য Assets গুলোকে Public ফোল্ডারে রাখতে হয়।

**তারপর**

HTML::script() is used for .js files and HTML::style() is used for css.

অথবা, <link href="{{ URL::asset('css/masterCss.css') }}" rel="stylesheet">

Or

<link href="/css/master.css" rel="stylesheet">

[http://laravel.com/docs/5.1/upgrade#upgrade-5.0](http://laravel.com/docs/5.1/upgrade" \l "upgrade-5.0)

1. **Migrating one version to another:**

In cmd run command:

composer require laravelcollective/html

In your composer.json file in "require" group should be added this line:

"laravelcollective/html": "5.2.\*"

Update composer with command:

**composer update**

In config/app.php add this lines:

in providers group:

Collective\Html\HtmlServiceProvider::class,

in aliases group:

'Form' => Collective\Html\FormFacade::class,

'Html' => Collective\Html\HtmlFacade::class,

Now we can dynamic our Link, CSS, Images, Script like this:

**{!! Html::style('assets/bootstrap/css/bootstrap.css')!!}**

1. Database এর সাথে Connection:

Vendor>env এই env ফাইলটাই Database Connection ফাইল।

Database তৈরির Code:

**In command line:**

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

তাহলে user table তৈরি হবে।

Table এ ফিল্ড add করার পরে Run করতে হবে তা না হলে ফিল্ড add হবে না।

**Command Line:**

php artisan migrate

এবং Rollback বা আগের অবস্থায় ফিরে আসার জন্য বা এক স্টেপ পিছেনে আশার জন্য **Command line**:

php artisan migrate:rollback

1. **Restfull Resource Controller:**

Restful Resource Controller তৈরি করলে Route.php পেইজ এ Route::get এর পরিবর্তে Route::resource ব্যবহার করে Controller টিকে include করতে হবে যাতে সহজে Image বা অন্যান্য ফাইল নিয়ে কাজ করা যায় এবং ফাংশনটা হবে নিচের মত:

Route::resource('registration','dbController');//Only Controller Load হবে, কোন ফাংশন হবে না।

1. {{ Form::open(array('route' => 'registration.store')) }}

//Html Form

{!! Form::close() !!}

এটার মাধ্যমে URL এ ডাটা পাঠানোর জন্য **page\_name.store** ('registration.store')

অথবা নরমালভাবেও পাঠানো যায় এক্ষেত্রে ফর্ম এর নিচে {{Form::token()}} এটা দিতে হবে।

1. **Accessing The Request** or **Input request** (Form Data controller থেকে দেখা):

public function store(Request $request){

$input=$request->all();

return $input;

}

1. **Mass Assignment** for getting data to **model** from controller for storing to database: Model তৈরির জন্য **Class** এ নিচের Code লিখতে হবে।

protected $table = '$users';

protected $primaryKey='$user\_id';

protected $fillable = ['id', 'password'];

1. **Connection between Controller and Model (Controller to Model):**

public function store(Request $request)

{

$input=$request->all();

//return $input;

myFirstModel::create($input);

return redirect('success');

}

url:localhost:8000/registration/create

1. **Model to Controller (Retrieving Multiple Models):**

public function index()

{

$allDBdata = myFirstModel::all();

return $allDBdata;

}

url:localhost:8000/registration

1. **Pagination (Paginating Eloquent):**

public function index()

{

$allDBdata = myFirstModel::paginate(1); //all() **ফাংশনটি দেওয়ার দরকার নাই**

return $allDBdata;

}

1. **Displaying Results In A View (Controller to View):** compact or with **ফাংশনের মাধ্যমে পাঠাতে হবে**

public function index()

{

$allDBdata = myFirstModel::paginate(1);

return view('dbTest.index', compact('allDBdata'));

//return $allDBdata;

}

url:localhost:8000/registration

1. Insert, update, delete oparations:

**Edit or Update (View to URL): URL এ আমরা ID পাঠালাম বা Using Link**

<td><a class="btn btn-info" href="{!! Route('reg.edit',$singleDBdata->user\_id) !!}">Edit</a></td>

**Query in Controller by ID and Controller to view:**

public function edit($id)

{

$allDBdata=myFirstModel::findOrFail($id);

return view('dbTest.edit',compact('allDBdata'));

}

**In View after Update and Data will be sent from View to Controller:**

{!! Form::open(array('route' => ['reg.update', $allDBdata->user\_id],'method'=>'PUT')) !!}

**In controller:**

public function update(Request $request, $id)

{

$inputData=$request->all();

$dbQueryData=myFirstModel::findOrfail($id);

$dbQueryData->update($inputData);

return redirect('reg');

}

**Delete:**

**In View:** Using Form

<td>

{!! Form::open(array('route' => ['reg.destroy', $singleDBdata->user\_id],'method'=>'delete')) !!}

{!! Form::hidden($singleDBdata->user\_id) !!} <!-- this line is not needed -->

{!! Form::submit('Delete', ['class'=>'btn btn-danger']) !!}

{!! Form::close() !!}

</td>

**In Controller:**

public function destroy($id)

{

$dbQueryData=myFirstModel::findOrfail($id);

$dbQueryData->delete($dbQueryData);

return redirect('reg');

}

1. Searching Data from Database:

**In View:**

{!! Form::open(['route' => 'reg.index','method' => 'GET']) !!}

{{ Form::text('user\_name')}}

{{ Form::submit('Search')}}

{!! Form::close() !!}

**In Controller:**

public function index(Request $request)

{

$count=1;

$allDBdata = myFirstModel::orderBy('user\_name');

$user\_name=$request->input('user\_name');

if(!empty($user\_name)){

$allDBdata->where('user\_name', 'like', '%'.$user\_name.'%');

}

$allDBdata = $allDBdata->paginate(3);

return view('dbTest.index', compact('allDBdata','count'));

//return $allDBdata;

}

**For Two search boxes:**

**In view:**

{!! Form::open(['route' => 'reg.index','method' => 'GET']) !!}

<lavel>User Full name: </lavel>{{ Form::text('user\_fullname')}}

<lavel>Username: </lavel>{{ Form::text('user\_name')}}

{{ Form::submit('Search')}}

{!! Form::close() !!}

**In Controller:**

public function index(Request $request)

{

$count=1;

$allDBdata = myFirstModel::orderBy('user\_name');

$user\_fullname=$request->input('user\_fullname');

$user\_name=$request->input('user\_name');

if(!empty($user\_fullname)){

$allDBdata->where('user\_fullname', 'like', '%'.$user\_fullname.'%');

}

if(!empty($user\_name)){

$allDBdata->where('user\_name', 'like', '%'.$user\_name.'%');

}

$allDBdata = $allDBdata->paginate(3);

return view('dbTest.index', compact('allDBdata','count'));

//return $allDBdata;

}

1. Authentication or Login/Logout System: