***LARAVEL***

\*We will need Composer install first

\*Composer = PHP dependency Manager.

\*Composer install next Laravel install.

\*Use command line $ composer create-project laravel/laravel project-name.(“llc\*laravel”)

\*Namespace PHP version 5.3.0 from available.

\*Same names multiline class use for namespace.

\*PHP server open by command line $php artisan serve

\*env=Enverment

\*.env file a something change korte hobe for database connection;

\*config/app.php something change has been coming;

\*<?php echo ‘ ’;?> of instead we use by {{config(‘ ’)}}

\*WE need a controller create,use by command line $php artisan make:controller HomeController

\*Table foreign key define $table->foreign(‘user-id’)->references(‘id’)->(‘users’);

\*table for a column add :

Schema::table(‘users’,function(Blueprint $table){

$table->string (‘email’)->nullable()->after(‘name’);

});

\*Column Delete:

Schema::table(‘users’,function(Blueprint $table){

$table->dropColumn(‘name’);

});

\*Create a database use by command line $create database llc\_ecommerce;

\*table create then this table run db use by command line$php artisan migrate;

\*table show database use command line $Mysql –uroot

$use llc\_ecommerce;

$show tables;

$describe users;

\*mysql table data show:$select \* from users;

\*show migration table:$select \* from migrations;

\*table dropdown or rollback:$php artisan migrate:rollback;

\*Migration table create:$php artisan make:migration create\_orders\_table;

\*Model create:$php artisan make:model Models\Order;

\*Seeder create:$php artisan make:seeder UserstablesSeeder;

\* database drop:$drop database llc\_ecommerce;

\*Something delete from laravel; do you run command line$composer dampautoload ;

\*Migration and seed part with run$php artisan migrate –seed;

\*Migratation refresh $php artisan migrate:refresh –seed;

\*When we one seeder run so that we run$php artisan db:seed --class=UserTableSeeder

\*When we all seeder run so that we run$php artisan db:seed

\*Route connect by homecontroller: Route::get(‘/’, ’HomeController@showHome’);

\*Route page connection:

\*Route::get(‘contact’,’FrontContact@showContact’)->(‘contact’);

\*<a href=”{{route(‘contact’)}}”>Contact</a>

\*Route a same names,example:admin/order;admin/user;admin/login;another page thakte pare so we are create this route;

Route::group([‘prefix’=>’admin’],function(){

Route::get(‘/order’,’ FrontContact@showContact’);

Route::get(‘/user,’ FrontContact@showContact’);

Route::get(‘/login,’ FrontContact@showContact’);

});

\*Shortcart namespace connection by route:

Route::group(['namespace'=>'Frontend','name'=>'frontend.'],*function* (){  
 Route::get('/', 'Homecontroller@showIndex')->name('index');  
 Route::get('/about', 'Homecontroller@showAbout')->name('users');  
});

\*All files namespace must be right don’t forget it;

\*Users/id/name khuje ber korte hole route jevabe create korte hobe:

Route::get('/users/{id}/{name}', 'Homecontroller@showUser');

\*Best practice:

Route::get('/users/{id}/{name?}', 'Homecontroller@showUser')->where(['id'=>'[0-9]+','name'=>'[a-z]+']);

\*R controller create for example:

*public function* showUser($id,$name=’’){  
 *echo* $id.' '. $name;  
}

\*When create a register form we use by @csrf=Cross-site request forgery token;

\*Backend or Frontend ar vitor for a cotroller create use command line:

$php artisan make:controller Backend\\UsersController;

\*Route show example use command line:$php artisan route:list;

\*process register first validation then insert into database then redirect;

\*Resource controller create$php artisan make:controller Frontend\\postcontroller –resource;

\*r atar jonno jevabe route create korte hobe

Route::resource(‘posts’,’postcontroller’)->except([‘destory’]);

\*Ro php <?php echo “Hello”;?> \*Laravel blade {{(‘’Hello”)}} this is the better;

\*Laravel ifelse statement: first controllers file create then blade.php file a statement create;

*class* Homecontroller *extends* Controller  
{  
 *public function* Index(){  
 $data=[];  
 $data['day']='Saturday';  
 *return* view('about',$data);  
 }

}

<div *class=*"title m-b-md">  
 *@if(*$day==='Friday'*)* Today is Holiday!  
 *@elseif(*$day==='Saturday'*)* Today is half work day!  
 *@else* Today is works day!  
 *@endif*</div>

\*unless statement: *@unless(*$day==='Friday'*)* Today is Work day  
 *@endunless*

\*Loop Variable statement foreach links work:

$data=[];  
 $data['links']=[  
 'facebook'=>'https://facebook.com',  
 'twitter'=>'https://twitter.com',  
 'google'=>'https://google.com',  
];

<div *class=*"links">  
 *@foreach(*$links *as* $title=>$url*)* <a *href=*"{{$url}}">  
 {{$title}} {{$loop->iteration}}  
 </a>  
 *@endforeach* </div>

\*Blade templets master file a @yield(‘content’)use kore side file a @section(‘content’)HTML@stop ;

\*Master file side file jevabe declear kora hoy @extends(‘master’) & others partials files declear @include(‘\_header’);

\**public function* showprocessRegistration(Request $request){  
 *return* $request->all();  
}

\*$request->all(); give us all data;

\*If any data don’t use:

*public function* showprocessRegistration(Request $request){  
 *return* $request->except(['\_token']);  
 }

\*Register form for error returns: *@if (*$errors->any()*)* <div *class=*"alert alert-danger">  
 *@if(*$errors->count()===1*)* {{$errors->first()}}  
 *@else* <ul>  
 *@foreach (*$errors->all() *as* $error*)* <li>{{ $error }}</li>  
 *@endforeach* </ul>  
 *@endif* </div>  
*@endif*

\*Customize validation: $validator = Validator::make($request->all(), [  
 'full\_name' => 'required',  
 'email' => 'required|email|unique:users,email',  
 'phone\_number' => 'required|min:6|max:13|unique:users,phone\_number',  
 'password' => 'required|min:6|confirmed',  
]);  
*if* ($validator->fails()) {  
 *return* redirect()->back()->withErrors($validator)->withInput();  
}

\*When a html from for uploads file create so <from *enctype=*"multipart/form-data> declear ;

\*Image uploads time config/filesystem.php something changing;

\*File uploads testing dd($request->file('photo'));

\*Laravel file uploaded coding:

*public function* showprocessRegistration(Request $request)  
{  
 $this->validate($request,[  
 'photo' =>'required|image|max:10240',  
 ]);  
 $photo = $request->file('photo');  
 $file\_name=uniqid('photo\_',*true*).str\_random(10).'.'.$photo >getClientOriginalExtension();  
 *if*($photo->isValid()){  
 $photo->storeAs('images',$file\_name);  
 }

\*Laravel login form : *public function* processLogin(Request $request)  
{  
 *// validation* $this->validate($request, [  
 'email' => 'required|email',  
 'password' => 'required|min:6',  
 ]);  
 $credentials = $request->except(['\_token']);  
 *if* (auth()->attempt($credentials)) {  
 *return* redirect()->route('home');  
 }  
 $this->setErrorMessage('Invalid credentials.');  
 *return* redirect()->back();  
}

\*Laravel database/factories uses huse data insert database;

\*Github Sumon vai <https://github.com/SumonMSelim/llc05-laravel>;

\*User email id & image other pages show: *public function* showHome()  
{  
 $data['user']=auth()->user();  
 *return* view('home',$data);  
}

& Home page : <h4>Logged in successfully as {{optional($user)->email}}</h4>  
<p>  
 <img *src=*"{{url('uploads/images',optional($user)->photo)}}" *class=*"img-thumbnail">  
</p>

\*Pagination Controller: *public function* showindex(){  
 $data['categories']=Category::select('id','name','slug','status')->paginate(3);  
 *return* view('backend.category.index',$data);  
}

\*Index:

<div *class=*"pagination justify-content-center">  
 {!! $categories->links() !!}  
</div>

\*Laravel post a kaj korar somay resource use korte hobe,controller create:

$php artisan make:controller Bakend\PostController –resource;

\*And route defined Route::resource('/posts', 'Backend\PostController');

\*Route show:$php artisan route:list;

\*Relational operation Models/Post.php in:

*public function* category()  
{  
 *return* $this->belongsTo(Category::*class*);  
}

\*Webmix file a change asle command line$npm run development -- --watch;

\*Controller all method create how$php artisan make:controller StudentsController –r;

\*model,migration,controller with create$php artisan make:model Models\Product –mc;

\*webpack.mix a change korle commandline run$yarn run watch

\*Image manipulation $composer require intervention/image;

\*image insert for form in use *enctype=*"multipart/form-data";

\*Instalation done,next config folder er app.php te provider include Intervention\Image\ImageServiceProvider::*class*,

then aliases in include 'Image'=>Intervention\Image\Facades\Image::*class*,

\*When laravel development time we install$composer require barryvdh/laravel-debugbar –dev

\*Then this package link config/app.php providers;

\*Relational join,ak table ar data others table a include.

//example post table include hobe others tables data so

PostController ar moddhe

$data['posts'] = Post::with('category', 'user')->select

('id', 'title', 'user\_id', 'category\_id', 'status')->paginate(10);

And Post Model a jei jei table include hobe oi table ar name bole dite hobe//example

*public function* category()  
{  
 *return* $this->belongsTo(Category::*class*);  
}  
  
*public function* user()  
{  
 *return* $this->belongsTo(User::*class*);  
}

\*category ar moddhe a post ar relation first category model a->

//Example

*public function* posts()  
{  
 *return* $this->hasMany(Post::*class*);  
}

\*next category controller a

//Example *$data['category'] = Category::with('posts')->select('id', 'name', 'slug', 'status', 'created\_at')->find($id);*

\*Cache use ar jonno,config file ar cache.php file a uses'default' => env('CACHE\_DRIVER', 'redis'),

\*Use Redis connection// first redis install, then config/cache.php file a something change,then

.env file change asbe, then terminal run $composer require predis/predis. Then j controller a cache use hobe oi file a cache function use korte hobe..example//

$data['articles'] = cache('articles',*function* (){  
 *return* Post::with('user', 'category')->orderBy('created\_at', 'desc')->take(50)->get();  
});

Then Events file create.ki ki events file use hote pare oi events file create korte hobe.Then listeners file create,, and listeners file a

*public function* handle($event)  
{  
 cache()->forget('articles');  
 $posts = Post::with('user', 'category')->orderBy('created\_at', 'desc')->take(50)->get();  
 cache()->forever('articles', $posts);  
}

Then provider/EventserviceProvider.php file a something change asbe..Example//

*protected* $listen = [  
 PostCreated::*class* => [  
 PostCacheListener::*class*,  
 ],  
 PostUpdated::*class* => [  
 PostCacheListener::*class*,  
 ],  
 PostDeleted::*class* => [  
 PostCacheListener::*class*,  
 ],  
];

\*Mailer create $php artisan make:mail VerificationEmail ,then verification.php file a kon file theke mail aste ace oi file bole dite hobe,controller a register a somay mail option ta create korte hobe example//

*Mail::to($user->email)->send(new VerificationEmail($user));*

.env file a mail ar host,name,username,password bole dite hobe, config/mail.php file somthig change.

\*Email use ar jonno QUEUE use korte hobe,,ar jonno .env file queue-driver=redis use korte hobe,,controller ar register mail ar moddhe

*Mail::to($user->email)->queue(new VerificationEmail($user));*

And registration ar age ba pore command line a $php artisan queue:work; run korte hobe

\*Notification use kore mail pas ar jonno first notification class creat korte hobe example//$php artisan make:notification VerifyEmail,then j notification ta create holo oi notification a system moto sajate hobe, notification a queue use korte hole ta define kore dite hobe implements ShouldQueue,controller register ar moddhe use,,example

$user->notify(*new* VerifyEmail($user));

\*SMS Notification ar jonno first run$composer require nexmo/client,next webserver nexmo login then nexmo-key and nexmo-secret config/service.php te include korte hobe example//

'nexmo' => [  
 'key' => env('NEXMO\_KEY','b4797050'),  
 'secret' => env('NEXMO\_SECRET','Dvuaju089JR8VoLh'),  
 'sms\_from' => '15556666666',  
],

Nexmomessage link korar jonno run $composer require laravel/nexmo-notification-channel

Then notification file a verifyEmail.php te return[‘mail’,’nexmo’] define korte hobe then sms asar jonno ai file class create korte hobe Example//

*public function* toNexmo($notifiable)  
{  
 *return* (*new* NexmoMessage())  
 ->content('Dear ' . $this->user->full\_name . '. Your account is registered. - LLC Team');  
}

\*Database notification use ar jonno run $composer artisan notifications:table when table create successful then run$php artisan migrate;then notification folder ar moddhe akta notifyadmin.php create then ai file ar verify and controller ar moddhe example//

$user->notify(*new* VerifyEmail($user));  
$admin = User::find(4);  
$admin->notify(*new* NotifyAdmin($user));

\*And notification database ar data show korar jonno create example//

*@if(*$user->id ===1*)* <div *class=*"well">  
 <ul>  
 *@foreach (*$user->unreadNotifications *as* $notification*)* <li>{{ $notification->data['full\_name'] }} just registered</li>  
 *@php* $notification->markAsRead(); *@endphp  
 @endforeach* </ul>  
 </div>  
*@endif*

\*BROADCAST use…..first event file create na thakle ta create kore nite hobe then broadcaston class ar moddhe kaj korte hobe,then config/broadcast.php file a config kore nite hobe,pusher use korle pusher or redis use korle redis;then puser ar pakeg run korte hobe $composer require pusher/pusher-php-server ” ~3.0 ”,,even postcreate file a implements ShouldBroadcast korte hobe, pusher.com a akti account create korte hobe then account login dashboard ar id-no,key-no,secret-key-no .env pusher ar moddhe define korte hobe,,, .env pusher-driver a pusher ar name define korte hobe.

\*Broadcast pusher notify use kore application kono new post add hole ta notify korar jonno ,

event/postcreate ar moddhe define

*public function* broadcastOn()  
{  
 *return new* Channel('post.created');  
}

then master.blade file js code likhte hobe notification ar jonno example//

<script>  
 Echo.public('post.created')  
 .listen('PostCreated', (e) => {  
 $.notify(e.post.title + ' has been published now');  
 });  
</script>

And ar jonno bootstrap-notify.min.js file ta js file create korte hobe;

\*PHP artisan laravel command create, first $php artisan make:command EmailInactive@users;then this created file verify user k notification deoar jonno je dhorener class create korte hobe ta ai file a define korbo,then mail notify’r jonno notification folder ar moddhe NotifyInactiveUser name akta notification create korbo and message ar logic likhbo;command line diay check korar jonno $php artisan email:inactive-users;

\*Task schedule use kore uporer command laravel je vabe user ark ache notification dy schedule use kore nirdisto time por por user ark ache email pathano jay,ar jonno consol/karnel.php file a kichu kaj korte hoy and email pathanor jonno $php artisan schedule:run korte hoy;

\*API = Application Programing Interface

\*RDBMS : Relational database manage system.

\*MVC : Model view Controller

\*laravel use ar jonno must use laravel debugger this run command line that project--

composer require barryvdh/laravel-debugbar –dev

\*laravel a ide helper use better so we run$composer require –dev barryydh/laravel-ide-helper

Then its config run $ php artisan vendor:publish --provider="Barryvdh\LaravelIdeHelper\IdeHelperServiceProvider" --tag=config;

Then ide-helper.php file config.then 2 ta command run php artisan ide-helper:generate;

Then run composer require --dev doctrine/dbal;then run php artisan ide-helper:models

//holder js use korar jonno run $yarn run holderjs.then $yarn run dev

**Git**

\*git init

\*git status

\*git add .

\*git commit –m “Laravel setup done, added redme instruction”

Brance create

\*git branch

\*git checkout –b dev //dev create

\*git remote add origin <https://github.com/saiful912/laravel_project.git>

\*git push origin dev //dev a push korar jonno

\*git checkout master

\*git push origin master //master a push korar jonno

.git ak branch theke r ak branch a aste hole ,git checkout dev,then oi branch ar data ai branch a ante hole run git pull,then git merge saiful-dev,then git a push;

// .env file copy korar jonno run$ cp .env.example .env

\*\*Laravel a image use korar jonno package use kora jay,ar jonno run

composer require "spatie/laravel-medialibrary

then package migration init korar jonno run

php artisan vendor:publish –

provider="Spatie\MediaLibrary\MediaLibraryServiceProvider" --tag="migrations"

then config file publish korar jonno run

php artisan vendor:publish --provider="Spatie\MediaLibrary\MediaLibraryServiceProvider" --tag="config"

[\\media](file:///\\media) file model a use korar jonno first oi model ar class ar moddhe **implements HasMedis** then function ar moddhe **use HasmediaTrait**

//laravel a mix file jokhon plain css add korbo tokhon jevabe add korte hobe//

mix.styles([

'public/css/vendor/normalize.css',

'public/css/vendor/videojs.css'

], 'public/css/all.css');

//laravel a categories ar view onno page bar bar dekhate hole shorthcart a ja kora jay appService provider ar boot class kichu configar korte hobe;then jei home page a loop use kore anbo;

//laravel a success message or error message dekhano jonno controller.php file a

*protected function* setSuccess($message)  
{  
 session()->flash('type','success');  
 session()->flash('message',$message);  
}

and je file a message dekhabo oi file a

$this->setSuccess('Account Registered');

[\\notification](file:///\\notification) mail use a should key use korle $php artisan queue:work run korte hobe;

[\\laravel](file:///\\laravel) authentication ar jonno default laravel auth deoaa ace,ata bebohar korte hole first amader run $php artisan make:auth;

[\\laravel](file:///\\laravel) admin a kaj korte hole first admin table model create then config/auth.php file config kore nite hobe,then app\exception\handler.php file a exception create korte hobe;then middleware/redirectAuhtentication.php switch class create;

[\\admin](file:///\\admin) register ar jonno $php artisan thinker use kora jete pare;then $admin=new Admin();$admin->name=”saiful”.$admin->email=[saiful@gmail.com](mailto:saiful@gmail.com).$admin->password=Hash::make(“1234”).$admin->save();

[\\AJAX](file:///\\AJAX) = Asyncrousas Javascript and XML. Ajax use javascript ar jonno bebohito hoy.kono kichur id auto select ar jonno ajax code use hoy.amra register form a ar bebohar dekheci ,select line ar akti id niaychi then form ar nicy js code likheci,then jquery-3.4.2.min.js download kore ta include kora and route akta route function create korechi

Route::get('/get-districts/{id}', *function* ($id) {  
 *return* json\_encode(App\Models\District::where('division\_id', $id)->get());  
});

[\\button](file:///\\button) diay kono kichu add korar somay tate js use kora jay,button a

*onclick=*"addToCart({{$product->id}})

bebohar korte hobe then script tag use kore tar moddhe function bebohar korte hobe,cart ar moddhe jquery notify use korte hole api\cartController a store ar ses a

*return* json\_encode(['status'=>'success','Message'=>'Item added to cart','totalItems'=>Cart::totalItems()]);

then js function a code likhte hobe,api route ar moddhe carts route create

<script>  
 $.ajaxSetup({  
 headers: {  
 'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')  
 }  
 });  
  
 *function* addToCart(product\_id) {  
 $.post('https://monir\_laravel.test/api/carts/store',  
 {  
 product\_id: product\_id  
 })  
 .done(*function* (data) {  
 data = JSON.parse(data);  
 *if* (data.status === 'success') {  
 *//toast css js include baki ace* alertify.set('notifier',position('top-center'));  
 alertify.success('Item added to cart successfully || total Items: '+data.totalItems+'</br> to checkout ' +  
 '<a href="{{route('carts')}}">Go to checkout page</a>');  
 $('#totalItems').html(data.totalItems);  
 }  
 });  
 }  
</script> and toast css js include korte hobe

[\\laravel](file:///\\laravel) customer pdf file create first run

composer require barryvdh/laravel-dompdf

then config/app.php file third party package

Barryvdh\DomPDF\ServiceProvider::class,and aliases

'PDF' => Barryvdh\DomPDF\Facade::class,